Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electronic Systems

Simen Håbet Ødegård

Wireless Video Capsule Endoscopy Prototype Utilizing a Backscatter Transceiver for High Data Rate Transmission

Master's thesis in Electronic Systems Design and Innovation
Supervisor: Ilangko Balasingham
Co-supervisor: Ali Khaleghi
February 2022

NTNU
Norwegian University of Science and Technology

Simen Håbet Ødegård

Wireless Video Capsule Endoscopy Prototype Utilizing a Backscatter Transceiver for High Data Rate Transmission

Master's thesis in Electronic Systems Design and Innovation
Supervisor: Ilangko Balasingham
Co-supervisor: Ali Khaleghi
February 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electronic Systems

NTNU
Norwegian University of
Science and Technology

Abstract

Capsule endoscopy is an important medical application for safer, pain-free, and possibly more accurate detection of diseases in the gastrointestinal tract, such as cancerous polyps in the colon. Wireless Capsule Endoscopy (WCE) became available for clinical use in 2001. WCEs are powered by batteries and typically need to last more than 8-10 hours, depending on the application. The active components in a WCE are the camera, a processing unit, LED lights, and a transceiver. The WCE transmits a video stream to the equipment outside the body. The video stream consists of 2-30 frames per second, depending on the application. Power consumption is the main limiting factor to increasing the frame rate and image quality. The clinical value of the video stream requires high image quality which depends on the pixel depth, resolution, and illumination conditions.

It is proposed to implement the backscatter system developed by A. Khaleghi et al. [1] to improve the power efficiency of a WCE, potentially reducing the transceiver's power consumption from 20-45 mW to only 250 nW. In addition, different processing units and cameras will be considered to improve the WCE's power efficiency. A capsule-size prototype is developed to assess the impact of these improvements on the WCE's total power efficiency.

The designed capsule-size prototype is comprised of three PCBs and three batteries. Assembled, it measures 25.14 mm without the power switch and 27.65 mm with the power switch. The diameter is 9.5 mm. The backscatter switch ADG902 and the camera module NanEyeC from ams AG, with 320x320 resolution, are implemented. Due to the chip shortage, the capsule size prototype uses an STM32L051 MCU. Based on the specifications and power simulations, we wanted to use STM32L432, as this supports 12-bit SPI and has RAM large enough to receive camera data in the minimum number of operations. Three Renata SR754SW 80 mAh, 1.55 V batteries are used. The capsule-size prototype has two 2 mA 0201 LEDs to keep the impact of the LEDs as low as possible for the prototype. The realistic LED implementations are reviewed and analyzed.

The data transfer rate in interfacing with the backscatter system is 12 Mbps. The best software implementation, 12-bit SPI with DMA, running on an L4 MCU with 24 MHz core frequency, enables 8 FPS. The total power consumption for this configuration is 71.0 mW, resulting in a lifetime of 3 hours and 23 minutes. By reducing the frame rate to 2 FPS and implementing an adaptive LED scheme, only flashing the LEDs during the exposure time, the lifetime is increased to 7 hours and 15 minutes. The power analysis shows that the implementation of backscattering as the transceiver, reduces the total power consumption by 20-28 %. Giving this WCE implementation a huge advantage to the existing WCEs. The most significant impact on the combined power consumption is the 3V3 LDO, responsible for 57 % of the total power consumption.

Sammendrag

Kapselendoskopi er en viktig medisinsk applikasjon for sikrere, smertefri og muligens mer nøyaktig påvisning av sykdommer i mage-tarmkanalen, som kreftpolypper i tykktarmen. Trådløs Video Kapselendoskopi (WCE) ble tilgjengelig for klinisk bruk i 2001. WCE-er drives av batterier og trenger vanligvis å vare i mer enn 8-10 timer, avhengig av bruken. De aktive komponentene i en WCE er kameraet, en prosesseringsenhet, LED-lys og en sender/mottaker. WCE sender en videostrøm til utstyret utenfor kroppen. Videostrømmen består av 2-30 bilder per sekund, avhengig av applikasjonen. Strømforbruk er den viktigste begrensende faktoren for å øke bildefrekvensen og bildekvaliteten. Den kliniske verdien av videostrømmen krever høy bildekvalitet som avhenger av pikseldybden, oppløsningen og belysningsforholdene.

Det foreslås å bruke backscatter-systemet utviklet av A. Khaleghi et al. [1] for å forbedre strøm-effektiviteten til en WCE, og potensielt redusere senderen/mottakerens strømforbruk fra 20-45 mW til bare 250 nW. I tillegg vil ulike prosesseringsenheter og kameraer bli vurdert for å forbedre strøm-effektiviteten til WCE-er. En prototype i kapselstørrelse er utviklet for å vurdere effekten av forbedringene på den totale strøm-effektivitet til WCE-er.

Prototypen i kapselstørrelse består av tre kretskort og tre batterier. Montert måler prototypen uten strømbryter 25,14 mm og 27,65 mm med strømbryter. Diameteren er 9,5 mm. Backscatter-bryteren ADG902 og kameramodulen NanEyeC fra ams AG, med 320x320 oppløsning, er brukt. På grunn av brikkemangel, bruker kapselstørrelsesprototypen en STM32L051 mikrokontroller-enhet. Basert på spesifikasjonene og effektsimuleringene ønsket vi å bruke STM32L432, da denne støtter 12-bits SPI og har RAM som er stor nok til å motta kameradata i så få operasjoner som mulig. Det brukes tre Renata SR754SW 80 mAh, 1,55 V batterier. Prototypen i kapselstørrelse har to 2 mA 0201 LED-lys for å minimere LED-lysenes innvirkning på det totale strømforbruket. De aktuelle implementeringene av LED-lysene for et endeprodukt blir gjennomgått og analysert.

Dataoverføringshastigheten i systemet er 12 Mbps. Den beste programvareimplementeringen, 12-bits SPI med DMA, kjører på en L4 mikrokontroller-enhet med 24 MHz kjernefrekvens, som muliggjør 8 bilder i sekundet. Det totale strømforbruket for denne konfigurasjonen er 71,0 mW, noe som gir en levetid på 3 timer og 23 minutter. Ved å redusere bildefrekvensen til 2 bilder i sekundet og implementere en adaptivt LED-lys kontroll, gjennom å bare bruke LED-lysene under eksponeringstiden, økes levetiden til 7 timer og 15 minutter. Effektanalysen viser tydelig at implementering av backscatter som sender/mottaker reduserer det totale strømforbruket med 20-28 %. Den største delen av det kombinerte strømforbruket er 3V3 LDOen, som utgjør 57 % av det totale strømforbruket.

Preface

I am a student at the integrated master's program, Electronics Systems Design and Innovation, at the Norwegian University of Science and Technology, specializing in embedded engineering. In addition, I have a considerable interest in the field of medicine. Therefore, I have taken an additional profile in my studies "ICT in Health", which have allowed me to combine my interest in electronics and medicine. This is my master thesis with supervisor Ilangko Balasingham and co-supervisor Ali Khaleghi.

The extensive literature search gave a good understanding of the WCE application and the vast potential for future applications. It has been fascinating to see how large the WCE research field is, and I feel privileged to be part of the development of such an important application.

The whole process has been highly educational. This project has provided me with more valuable knowledge than most subjects throughout my studies. The project provided knowledge about researching and choosing the best development platform and hardware specifications, communication protocols, camera interfacing, imaging theory, PCB design, and manufacturing. The coding has been varied with C programming at all levels: assembly, register, low, and high levels. I have learned a lot about debugging and have seen the importance of good documentation and obtaining all available documentation. I have even read the manual for the oscilloscope, which gave me a good understanding of how to utilize the oscilloscope as a debugging tool properly. Learning Altium Designer for PCB design from scratch has been challenging, with a steep learning curve. I am grateful for this knowledge and know it will be valuable when starting my career in the industry.

The Covid-19 situation and the considerable chip shortage have significantly impacted this work. I have spent countless hours searching for components and finding replacements due to size limitations and out-of-stock components and learned the hard way that components had to be ordered on the fly as soon as we decided to use them. I have spent much time discussing with PCB manufacturers, doing several revisions, and placing orders. This process has given me valuable knowledge about PCB manufacturing, and I have seen firsthand how important it is to follow the specifications of the PCB manufacturer precisely. The current market situation is unfavorable for the "little guy," only ordering a few PCBs. Communication with the PCB manufacturers has required much nagging, and we were not a priority. The consequence was longer delivery times and much wasted time. The ordering processes even forced me to make an industrial account with UPS and handle the export of components. I must say I am looking forward to being part of a larger company in terms of PCB ordering.

I want to thank Nilojan Gnanakulasekaran for his technical support and advice in the startup phase and Amin Hasanvand for the suggestions during the coding process and continuous support during the debugging process of the software. Additionally, I would like to thank Magnus Kofoed, Thomas Mandal Nilsen, and Ingulf Helland, experts in hardware and PCB design. They provided valuable consultation and are referred to as hardware and PCB experts in the report, where their knowledge has been used. Finally, I am incredibly grateful for the continuous support, advice, and guidance throughout the thesis work provided by my supervisors, Ilangko Balasingham and Ali Khaleghi.

Contents

Figures

Tables

Code Listings

Acronyms

ACK Acknowledge. 23

ADC Analog-to-Digital Converter. 12, 13, 40, 41, 55, 80

AR Annular Ring. 29, 31

ASIC Application Specific Integrated Circuit. 1, 14–16, 40

BGA Ball Grid Array. 14, 66, 76, 145

BJT Bipolar Junction Transistor. 33, 34, 57, 139, 140, 151

BOM Bill of Materials. 39, 174, 176

BSRR Bit Set/Reset Register. 105–107

CAD Computer-Aided Design and Drafting. 68

CCD Charge-Coupled Device. 12, 13

CDR Clock Data Recovery. 21

CMOS Complementary Metal–Oxide–Semiconductor. 1, 9, 12–15, 17, 40, 41, 151

CMSIS Common Microcontroller Software Interface Standard. 45

CPU Central Processing Unit. 15, 23, 25, 40, 98, 105, 115

CRC Colorectal Cancer. 5, 6

CS Chip Select. 24, 40, 98, 147

DC Direct Current. 21, 55, 81

DFN Dual Flat Non-leaded. 35, 36, 84

DMA Direct Memory Access. 2, 3, 23, 25, 98, 99, 101, 102, 114–116, 138–141, 143, 147–150, 152, 153, 155–157, 159

DSP Digital Signal Processing. 12

DUT Device Under Test. 136

EMC Electromagnetic Compatibility. 27, 53, 55, 66, 80, 99

EMI Electromagnetic Interference. 27

ESD Electrostatic Discharge. 27, 55, 92

FET Field Effect Transistor. 33

FIFO First In, First Out. 148

FPGA Field-Programmable Gate Array. 1, 14, 16, 17, 22, 28, 40, 130, 147, 157, 159

FPS Frames Per Second. 9, 40, 44, 103, 116, 147, 148, 150, 152, 153, 156, 157

FSK Frequency-Shift Keying. 19

GI gastrointestinal. 1, 5–8, 22

GPIO General-Purpose Input/Output. 25, 48, 53, 54, 57, 79, 80, 99, 101, 104–106, 108, 138, 139, 148, 149, 155

GUI Graphical User Interface. 99

HAL Hardware Abstraction Layer. 45, 101, 104, 115, 116, 148

HDI High-Density Interconnect. 31, 66

I In-Phase. 21

I/O Input/Output. 13, 55

I2C Inter Integrated Circuit. 14, 24, 25

IAR Inner Annular Ring. 31

IC Integrated Circuit. 32, 33

IF Intermediate Frequency. 21

LAN Local Area Network. 22, 130

LDO Low Dropout Regulator. 35–37, 60–62, 69, 71, 72, 81, 84, 85, 92, 135, 141, 142, 150–153, 156, 157, 160

LED Light-Emitting Diode. 1, 2, 9–12, 37, 38, 48, 56–59, 61, 62, 69, 71, 73, 76, 79, 89, 98, 108, 135, 137, 139–142, 146, 147, 151–153, 155, 156, 160

LL Low-Level. 45, 103, 148

LNA Low-Noise Amplifier. 17, 21, 130

LP Low-Pass. 21

LQFP Low-profile Quad Flat Package. 36, 45, 47, 53

LSB Least Significant Bit. 24

LVDS Low-Voltage Differential Signaling. 40

MCO Microcontroller Clock Output. 104, 124, 127, 138

MCU Microcontroller Unit. 1, 2, 13–16, 24, 25, 28, 32, 36–41, 43, 45–49, 52–57, 61, 62, 66, 68–72, 74, 76, 79, 80, 92, 95, 98, 101, 102, 104, 107–110, 116–119, 124, 125, 128, 130, 135, 137–141, 143, 145–153, 155–157, 159, 160

MISO Master Input/Slave Output. 24

MLCC Multilayer Ceramic Capacitors. 32

MOSFET Metal Oxide Semiconductor Field Effect Transistor. 33

MOSI Master Output/Slave Input. 24, 41

MPU Microprocessor Unit. 16, 45

MSB Most Significant Bit. 24

NOP No Operation. 105–107, 128, 148

NSS Negative Slave Select. 24

OAR Outer Annular Ring. 29, 31

PA Power Amplifier. 17

PC Personal Computer. 7, 8, 22, 45, 85, 109, 114, 130–132

PCB Printed Circuit Board. 2, 3, 10, 11, 23, 26–32, 37–39, 48, 49, 52–54, 60, 63, 65, 66, 68–71, 73–76, 79–81, 86, 89, 90, 92, 95–97, 111, 112, 137, 145–147, 156, 159, 160, 176

PLL Phase-Locked Loop. 55, 104

PP Pixel Period. 41–44, 99, 101, 107, 116, 118–120, 122–124

PP Pad to Pad. 29, 31

PWM Pulse-Width Modulation. 108, 139

Q Quadrature. 21

QFN Quad Flat Non-leaded. 35

QFNP Quad Flat Non-leaded Package. 36, 45–47, 146, 157

RAM Random-Access Memory. 46, 56, 98, 102, 103, 116, 117, 148

REQ Request. 23

RF Radio-Frequency. 17, 18, 21, 22, 81

RFID Radio-Frequency Identification. 18, 19

RX Receiver. 17, 21, 130

SAW Surface Acoustic Wave. 21

SCCB Serial Camera Control Bus. 14

SCLK Synchronization Clock. 24

SDK Software Development Kit. 16, 45

SEIM Single Ended Interface Mode. 40–42, 69, 98, 114, 124, 126, 137, 140, 147, 150, 157

SMD Surface Mount Device. 3, 10, 23, 29, 32–34

SME Solder Mask Expansion. 31, 52, 79, 85

SMPS Switched Mode Power Supply. 47, 160

SMT Surface Mount Technology. 32, 36

SNR Signal-to-Noise Ratio. 21

SOT Small Outline Transistor. 34, 35

SPI Serial Peripheral Interface. 2, 3, 23–25, 40, 41, 43, 44, 46, 47, 56, 57, 79, 98, 99, 101–104, 114–120, 122, 138–141, 143, 147–150, 152, 153, 155–157, 159

SS Slave Select. 24

SWD Serial Wire Debug Interface. 55, 56, 79

SWO Serial Wire Output. 55

TP Trace to Pad. 29, 31

TT Trace to Trace. 29, 31

TW Trace Width. 29, 31

TX Transmitter. 17, 21, 130

UART Universal Asynchronous Receiver-Transmitter. 24, 25, 56

UGI upper GI. 1, 9

WCE Wireless Capsule Endoscopy. 1, 2, 6–10, 12–18, 20, 22, 45, 55, 57, 61, 68, 71, 85, 93, 147, 148, 151, 152, 155

WLCSP Wafer Level Chip Scale Packaging. 36, 45, 47, 66, 70, 146, 156, 157, 159, 160

Chapter 1

Introduction

1.1 Motivation

Capsule endoscopy is an important medical application for safer, pain-free, and possibly more accurate detection of diseases in the gastrointestinal (GI) tract, such as cancerous polyps in the colon. Wireless Capsule Endoscopy (WCE) became available for clinical use in 2001 [2]. Papers [2–4] reviews the state of the art, future applications of WCE and the challenges. WCE is considered to be the best diagnostic tool in the small bowel, but for the upper GI (UGI) tract and the colon, traditional flexible endoscopy and colonoscopy are still the preferred methods due to cases with too low specificity rates for WCEs. The low rates are caused by limitations in image quality and the lack of active steering. The active steering is out of scope for this paper. The main limiting factor in increasing the image quality is power consumption. WCEs are powered by batteries and typically needs to last more than 8-10 hours depending on the application. The active components in a WCE are the camera, a processing unit, Light-Emitting Diode (LED) lights, and a transceiver. The WCE transmits a video stream to the equipment outside the body. This video stream consists of 2-30 frames per second, depending on the application. WCE typically use two 3 V, 55 mAh batteries that deliver an average of 20 mW for 8 hours operating time [5].

The literature proposes several techniques to reduce the power usage of WCEs. The LEDs typically use 25-40 % of the total power consumption. It is proposed to use only 40 % brightness and perform application-specific post-processing of the frames to compensate for lower lighting. Another technique that is already in use is only to flash the LEDs exactly when capturing a frame. For the camera chip, Complementary Metal–Oxide–Semiconductor (CMOS) technology has become the standard due to its ultra-low power features [6]. The processing unit are responsible for all system functions and can be realized with a Microcontroller Unit (MCU), Field-Programmable Gate Array (FPGA) or Application Specific Integrated Circuit (ASIC). From the literature ASICs have the best performance in terms of processing power per watt. MCU has the worst energy efficiency but the development time and cost are significantly lower than both ASIC and FPGA [7].

The commercially available capsules use active transceivers. Active transceivers typically use 20-45 mW [8]. This is a significant part of the total power consumption. The lowest power transceiver for WCEs is proposed by A. Khalegi et al. [1]. The transceiver uses the backscattering technique. This is a semi-passive transceiver, where the power usage is reduced to only a micro-/nanowatt switch [8, 9]. All of the mentioned power-saving techniques will free up power that can be used to increase the image quality.

1.2 Objectives

The main objective of this project is to make a capsule-size prototype of a WCE and interface it with the backscatter system developed by A. Khaleghi et al. [1]. The capsule-size prototype should implement the backscatter switch used to modulate the incoming carrier wave from the backscatter reader. The backscatter reader and visualization of the images have been developed previously and are not a part of this work.

The design of a capsule-size prototype requires many considerations to be taken into account for hardware, software, and especially mechanical limitations. The goal is to fit the complete assembled capsule-size prototype in a capsule with outer measurements; 26 mm length and 11 mm diameter. In total, six Printed Circuit Boards (PCBs) will been designed. The first step is a custom evaluation board containing everything part of the capsule-size prototype. Additionally, a breakout board for the NanEyeC camera module will be designed to aid in the software development of the interfacing between the camera module and MCU, using a NUCLEO-L4R5ZI development board. The designed capsule-size prototype will consist of three PCBs and a programming board used to program the MCU before assembly.

The power consumption in WCEs are critical. Power analysis and execution time measurements will be conducted to gauge the energy efficiency and asses which hardware and software implementations result in the best performance of the capsule-size prototype.

1.3 Organization of the Report

The report is divided into eight chapters. Chapter 2 starts by introducing WCEs as an application. Further, all the active parts of a WCE are presented, including an extensive literature review aimed towards energy efficiency and a review of the camera solutions and LEDs available in the market. Finally, the backscatter system is presented.

Chapter 3 presents three of the design methods used for embedded programming: Direct Memory Access (DMA), Serial Peripheral Interface (SPI) and bit banging. Followed by an introduction to the design tool Altium Designer, with a brief overview of common manufacturing limitations that must be taken into

account in PCB design. Finally, some properties of the various Surface Mount Device (SMD) components used in this design are introduced.

Chapter 4 argues for the choice of the development platform and camera module. The rest of the chapter contains the implementation for the six PCBs designed in this work: custom evaluation board, capsule-size prototype (3 PCBs), NanEyeC breakout board, and the programming board. This is followed by the software development of SPI with and without DMA and pure bit banging.

The results are presented in Chapter 5, containing verification and measurements of the performance for the software implementations and the interfacing with the backscatter system. Additionally, the total power consumption for various implementations and configurations is included. The results and implementation are discussed in Chapter 6. Finally we conclude the work and present the future work in Chapter 7 and 8, respectively.

Chapter 2

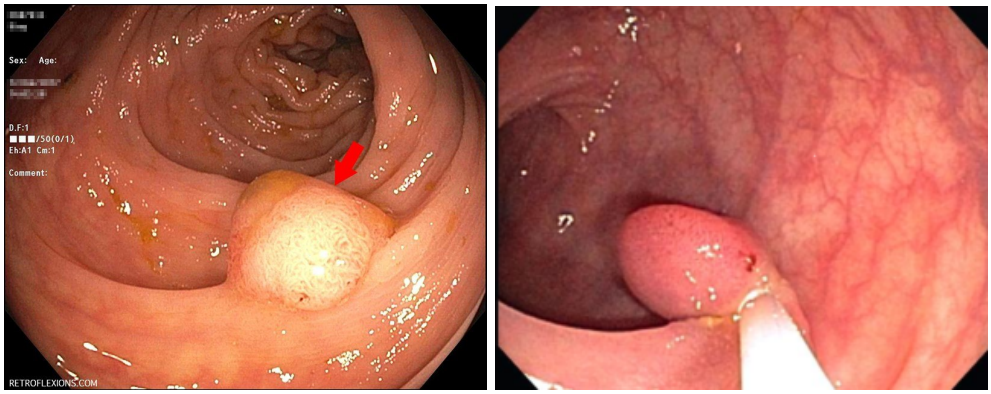
Background and Related Works

A large number of conditions and cancers are today diagnosed using various types of endoscopic procedures [10]. Only in the United States of America 75 million endoscopies were performed in 2017 [11]. This number includes diagnostic, biopsy, and surgical endoscopic procedures. Around 50 million of the procedures were GI endoscopies, and most of them were performed under anesthesia. According to a study analyzing the anesthesia administration in the endoscopy unit, there has been an increase in both the use of endoscopic procedures and the use of anesthesia in endoscopic procedures to ensure patient comfort and satisfaction [12]. They have observed an increasing number of complications and longer observation times due to the increasing use of anesthesia. Endoscopic procedures performed without general anesthesia are uncomfortable for the patient [10].

In an interview in Dagens Medisin with Bjørn Gustafsson, the Norwegian Gastroenterology Association leader, stated that around 90 % of the patients manage with only local anesthesia in the throat. Patients have reported high discomfort with the procedure, and a quote reads: "the tube from hell." He says that in Trondheim, 20-30 GI endoscopic procedures are performed every day. If more patients were to get general anesthesia, they would not have the capacity to perform this number of procedures. This is due to the required surveillance after having undergone general anesthesia. The waiting lists for gastroscopy are already long [13].

The primary use of endoscopic diagnostics is to detect cancer. The most frequent cancer types in the GI tract are colon cancer and rectal cancer. They are often grouped as Colorectal Cancer (CRC) due to similar properties. Colon cancer is the most common and requires a colonoscopy for diagnostic and biopsy. The cancer is visible as polyps, which are growths on the inside of the colon. Polyps are not uncommon, but most polyps are benign. When a patient has polyps larger than 1 cm, or three or more polyps are located in relatively close proximity, the patient is considered at risk of developing CRC. Polyps can be removed with a biopsy and studied under a microscope by a specialist to determine if the polyps are cancerous or benign [14, 15]. Figure 2.1 shows two images of polyps captured with an endoscope during a colonoscopy.

CRC has the third-highest incident rate and the second-highest mortality of



(a) **(b)**

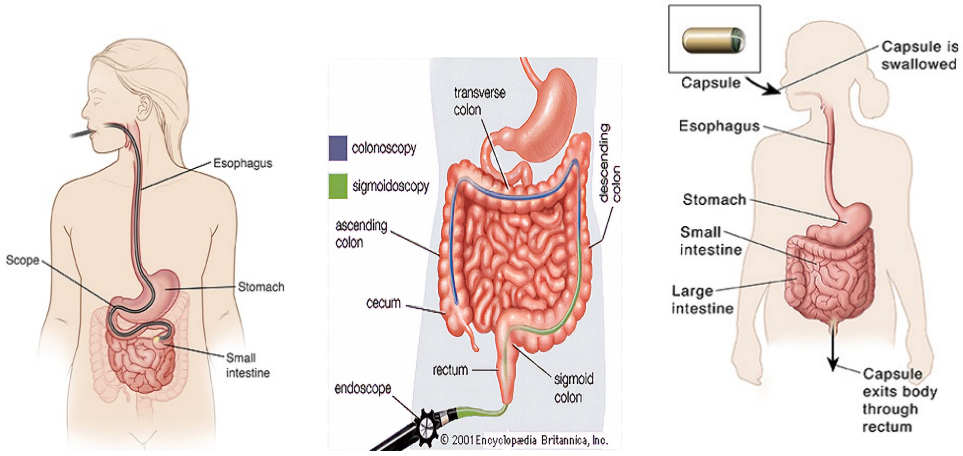
Figure 2.1: Two examples of polyps in the colon. The images are captured with an endoscope. In (b) they are preparing for a biopsy, using the cold snare technique. Image (a) is sourced from [16] and (b) is sourced from [17].

all cancer types, only beat by lung cancer [18]. According to the World Health Organization, more than 1.9 million people were diagnosed with CRC in 2020. The high CRC mortality rate caused more than 0.9 million fatalities in 2020 [19]. Studies show that these numbers will only increase, and it is estimated that the rate of incidences will increase by 40 % from 2020 to 2040 [18]. According to the American Cancer Society, screening is one of the most efficient ways to catch CRC at the early stages. They recommend an endoscopic check every ten years for everyone above 45 years at risk of developing CRC [20].

Another type of cancer is small intestine cancer. While CRC is responsible for 8.3 % of all cancer fatalities in USA, only 0.3 % is caused by small bowel cancer [21, 22]. The small intestine is the part of the GI tract that is hardest to reach. Only a few institutions have the equipment and expertise to perform small bowel endoscopy, also called balloon-assisted endoscopy, where they enter a long endoscope through the upper GI tract all the way through the small bowel. They use two balloons at the end of the endoscope to thread it slowly through the small bowel. This procedure requires full anesthesia of the patient. Due to the severity of the procedure and the low incidence rate of small bowel cancer, it is mainly used for biopsy and treatment. For diagnostic of the small bowel, WCE is today considered to be a requirement [23].

Figure 2.2 shows illustrations of gastroscopy, colonoscopy and WCE. Figure 2.2a shows an illustration of a gastroscopy, where the endoscope enters through the mount and ends at the start of the small intestine. Figure 2.2b shows both a colonoscopy and a sigmoidoscopy. Figure 2.2c shows a WCE and illustrates the path it travels, all the way through the GI tract.

It is clear that there is a need for WCEs, and that WCE has the potential to become the gold standard for diagnostic and screening purposes. As described above, the traditional wired endoscopy procedures can feel very invasive for the



(a) **(b)** **(c)**

Figure 2.2: (a) and (b) shows illustrations of the most common endoscopic procedures. (a) shows an illustration of a gastroscopy, where the endoscope enter through the mount and ends at the start of the small intestine. (b) shows both a colonoscopy and a sigmoidoscopy. (c) shows a WCE and illustrates the path it travels, all the way through the GI tract. The images are sourced from [24–26] respectively. Figure (b) has been slightly modified.

patient and might require general anesthesia. WCE is less invasive for the patient and does not require any anesthesia by removing the wired connection. In addition, the small capsule size makes WCEs beneficial for the patient's comfort and safety, as the capsule mitigates the risk of lesions compared to the traditional wired endoscopy procedure. The current product range of PillCam™capsules has an 11 mm outer diameter and varying lengths from 26-32 mm [27].

More widespread use of WCE will lighten the load on the hospitals and decrease the waiting time for necessary endoscopic procedures. Figure 2.3 shows an example of a typical WCE system suitable for home use. The system consists of an internal and external part. The internal part (a) is the capsule that is swallowed and transmits images of the GI tract. The external part (b) consists of the antennas placed on the abdomen and a recorder device. The antennas receive the transmitted images stored in the recorder device with the proper time stamps. The recorder typically comes with a battery pack and a carrying belt. In the depicted system, images are stored on a memory card and can be displayed on a Personal Computer (PC).

Ideally, a patient could pick up the equipment at a pharmacy or hospital. The patient could then bring the equipment home and swallow the capsule in the comfort of their own home. Finally, the patient would deliver the external equipment and memory card for analysis. This ease of use will likely result in higher participation in screening programs.

In screening programs and diagnostic procedures WCE removes the need of the health personnel performing the endoscopy and the use of anesthetics with

the required observation of the patient after the procedure. The only remaining personnel is the specialist that has to analyze the large number of images captured by the WCE. This person's workload can be substantially reduced by implementing automatic detection algorithms that analyze the video in real-time. These algorithms can be trained to have better error rates than real-life specialists. The algorithms can highlight possible cancerous polyps, which the specialist can review [4].

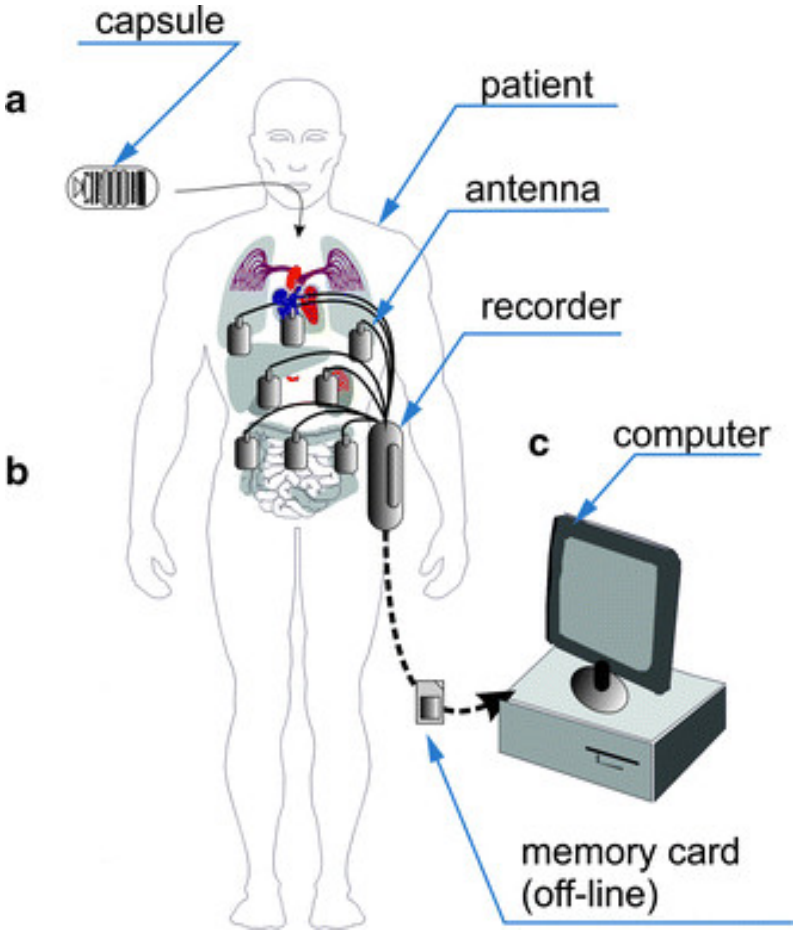


Figure 2.3: A typical WCE system for home use. The system consists of an internal and external part. The internal part (a) is the capsule that is swallowed and transmits images of the GI tract. The external part (b) consists of the antennas placed on the abdomen and a recorder device. The antennas receive the transmitted images stored in the recorder device with the proper time stamps. The recorder typically comes with a battery pack and a carrying belt. In the depicted system, images are stored on a memory card and can be displayed on a PC. The image is sourced from [4].

2.1 Wireless Video Capsule Endoscopy (WCE) Review and Technical Details

Since its release in 2001, WCEs has continuously improved in conjunction with advancements in technology. There are several different types of WCEs available today. Paper [2] reviews the state of the art WCEs. They differ mainly by the number of cameras and Frames Per Second (FPS). For the UGI tract, dual cameras are used with higher FPS as the capsule moves fast over a shorter distance. PillCam™UGI capsule has 18-35 FPS and 90 minutes operating time [28]. WCEs in the small bowel use a single camera with lower FPS and higher operating time as the capsule moves slower over a longer distance. PillCam™SB3, has 2-6 FPS and more than 8 hours operating time [29]. In the colon WCEs use dual cameras with a wide range of FPS and long operating time. PillCam™Crohn's Capsule has 4 FPS in stasis and up to 35 FPS in motion with a minimum operating time of 10 hours.

Figure 2.4 shows an expanded view of a generic WCE. The passive parts are the external casing, optical dome, and lens. The dome and external casing protects the electronics and give the capsule a smooth surface for safe passing. The optical dome spread the light from the LEDs and set the viewing angle, which also depends on the lens [4]. The active components are the LED-array, CMOS image sensor, MCU, transceiver, and batteries.

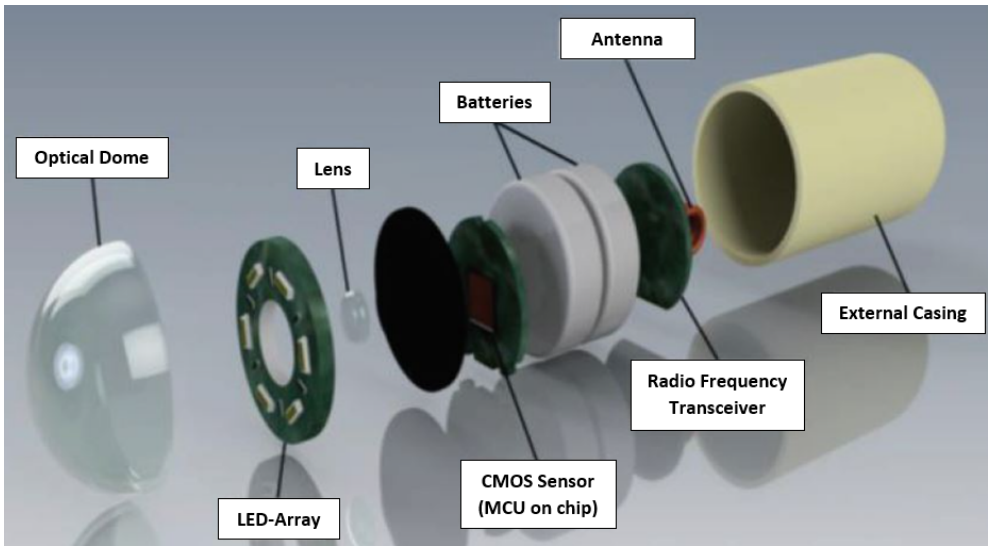


Figure 2.4: Expanded view of a generic WCE, showing the internal components. Modified version of figure 2 in [27].

2.1.1 LED

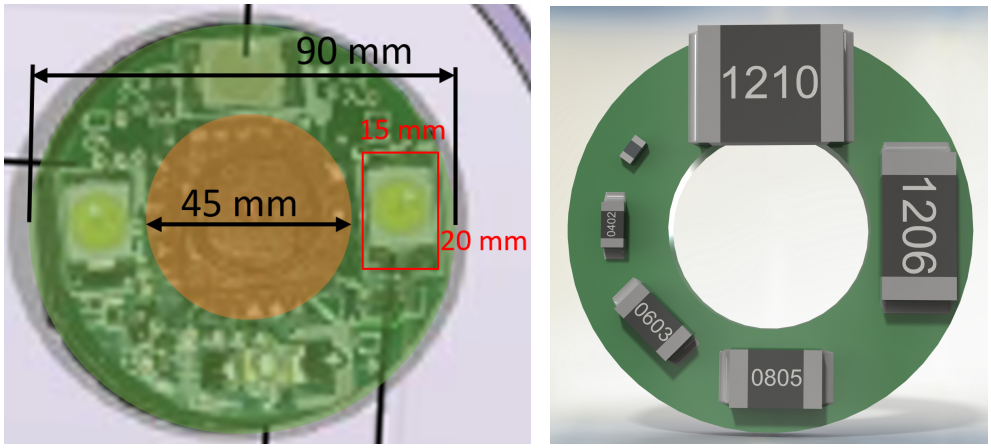
There are several different types of LEDs available for PCB mounting. Important parameters are size, luminosity, radiation field, electromagnetic spectrum, and power usage. The small size and limited power supply set strict boundaries for the choice of LEDs. The SMD LEDs comes in different sizes and efficiency (luminosity/watt). The most optimal result needs a balance between size, the number of LEDs and the efficiency. It's common to use 4, 6 or 8 LEDs [27].

The LEDs are only needed to provide good lighting conditions when capturing an image. A technique for saving power is only to flash the LEDs exactly when capturing an image [30]. Another power-saving technique is not to use maximum brightness at all times. In [6] they implemented and tested an adaptive algorithm that controls the brightness of the LEDs in real-time. The computations are made on the computer side, limiting extra power usage by the capsule and giving access to faster hardware. Good quality images were achieved at just 40 % brightness levels, reducing the total current for the illumination system by approximately 4 mA. The illumination hardware is a big factor in the total power consumption of the WCE, typically 25-40 % according to [6]. They also measured the total current used by the entire capsule and the current used by four LEDs. For 100 % brightness the LEDs used 24 % of the total current, but at 50 % brightness it used only 14 % of the total current. They measured the whole system at 100 % brightness to use an average of 41.5 mA while capturing 21 frames using a 5 V supply voltage. The resulting total average power consumption of the WCE was 207.5 mW, where the four LEDs at 100 % brightness used 50 mW.

Review of LEDs Available in the Market

The LED sizes used in commercial WCEs was not possible to find in the literature. For the sake of argument, the area of the illumination PCB was estimated from an image of the illumination PCB used in the WCE prototype in [6], this looks similar to the commercially available WCEs using four LED. The assumptions made are that the external casing has a diameter of 11 mm and a thickness of 1 mm. Hence the maximum PCB diameter is 9 mm. The area covered by the image sensor and optics is around 4.5 mm in diameter, illustrated by the orange circle in Figure 2.5b. The LEDs are estimated to be 0805 (20x15 mm). The imperial codes for SMD sizes will be used further, as it is common practice.

Figure 2.5b shows that 0201, 0402, 0603, 0805 and 1206 fits the ring, while 1210 is to large. A LED in each size is chosen for comparison on how the number and size of the LEDs affect the power usage. The choices are based on a search of available white LEDs in online stores and their respective datasheets. The LEDs with the best luminosity to power rating for each size was chosen. Due to the power limitations, 5 mA forward current was chosen and not the more common 20 mA forward current. For comparison, LEDs with the same max luminosity intensity was chosen. There is one exception to this, for the 0201 LED the only white LED available at the time has a 2 mA forward current and naturally a lower lumi-



(a) **(b)**

Figure 2.5: Estimation of the illumination PCB size. (a) shows an image of the illumination PCB from the WCE prototype in [6], with size estimates added on top of the image. (b) Shows a 3D-illustration of the estimated illumination PCB with 0201, 0402, 0603, 0805, 1206 and 1210 LEDs placed on the PCB.

Table 2.1: Power, luminosity intensity and max efficiency (max luminosity intensity/ max power) for white SMD LEDs in five different sizes

Size	Power [mW]		Luminosity Intensity [mcd]		Max Efficiency
	Min	Max	Min	Max	[Lumen/Watt]
0201	5.6	6.4	60	60	10.7
0402	13.25	15.25	45	180	11.8
0603	13.50	15.75	45	180	11.4
0805	14	16	56	180	10.6
1206	14.75	16	71	180	10.6

nosity intensity. This LED is part of the QuasarBrite™product series developed by Lumex, specifically for small ultra-low-power applications [31]. The low current and forward voltage of 2.8 V makes it substantially more low powered than the larger LEDs in this comparison.

Table 2.1 lists min and max values for power and luminosity intensity and finally the max efficiency for each size. All values in the table are only valid for a 5 mA forward current, except the 0201 LED. The data is retrieved from the respective datasheets [32–36]. The last LED size 1210 in the figure is too large for the PCB and not included in the table.

The table and datasheets show that the differences are minor between the four larger sizes. The 0402 and 0603 LEDs are the most efficient in terms of max luminosity per watt. The luminosity intensity range is the same for both. However, the 0402 has 0.5 mW lower power usage at max luminosity intensity. The two largest LEDs, 0805 and 1206, have the same max efficiency, slightly worse than

Table 2.2: Total power and luminosity intensity for 0402, 0602, 0805 and 1206 white SMD LEDs, for four, six and eight LEDs

#LEDs	Total Max Power [mW]				Total Max
	0402	0602	0805	1206	Luminosity Intensity [mcd]
4	61	63	64	64	720
6	91.5	94.5	96	96	1080
8	122	126	128	128	1440

the 0402 and 0603. The difference in max power is 0.25 mW compared to 0603 and 0.75 mW compared to 0402. The 0805 and 1206 LEDs has a higher minimum luminosity intensity. However, it is unlikely to use brightnesses below 40 %, based on the literature review. A significant difference between the LED sizes is the graph in the datasheets that plots Relative Luminous Intensity vs. Angular Displacement. These graphs show that 0402 has a more narrow radiation field than the larger LEDs, meaning that 0402 is more bright in one direction. The same is true for 1206 that has a wider radiation field than the smaller LEDs, meaning that larger LEDs spreads the light over a larger area.

Table 2.2 shows how the number of LEDs affects the total LED power usage, for the four largest LEDs that has the same max luminosity. The difference in power gets more significant between the LEDs for larger number of LEDs. For four LEDs the two largest use 1 mW and 3 mW more than 0602 and 0402 respectively. For six LEDs the two larges use 1.5 mW and 4.5 mW more than 0602 and 0402 respectively. Lastly for eight LEDs the two larges use 2 mW and 6 mW more than 0602 and 0402 respectively. The 0201 LED, not included in the table as the max luminosity is a third of the other LEDs. The corresponding max total power consumption for four, six and eight LEDs are 25.6, 38.4 and 51.2 mW.

2.1.2 Camera

Two camera technologies are used in WCEs, CMOS and Charge-Coupled Device (CCD). In the early years of WCE, CMOS image sensors were associated with noise and poor sensitivity compared to CCD image sensors that provided high quality images. The benefits of CMOS over CCD was lower power consumption and smaller implementation size, making them interesting for small wireless applications. In the last 15-20 years CMOS imaging technology have improved significantly faster than CCD. The size reduction in CMOS-technology gave room for on-chip amplifiers for all pixel-elements, reducing the noise and sensitivity issues previously associated with CMOS image sensors. These improvements made the image quality and resolution of CMOS image sensors comparable to CCD. The implementation of CMOS-cameras are simpler and smaller compared to CCD-cameras. CMOS image sensors nature opens the possibility of manufacturing a complete camera on a single chip - including Analog-to-Digital Converter (ADC), memory and Digital Signal Processing (DSP). The camera can then operate on the

same voltage as the MCU and simplifies interfacing with only digital Input/Output (I/O). CCD on the other hand, requires several independent components to make a complete camera, and operates at higher voltages (5-10 V). The advantages of using CCD is slightly better image quality, due to less noise and lower dependency on uniform illumination compared to CMOS. This is partly due to CCDs ability to expose all pixels at the same time, where CMOS use a rolling shutter. From the literature, CMOS imaging is the obvious choice for small and ultra-low power wireless applications like WCE [5, 37–40]. In [5] they compared nine commercial available WCEs. Only one of the WCEs use CCD technology - the EndoCapsule™ from Olympus™. One of the leading manufacturers PillCam™ use CMOS in all WCEs [41]. The EndoCapsule™ was compared with PillCam™SB in a 2007 study. From the specifications the main differences are the image sensor technology and resolution. PillCam™SB use CMOS (256x256) and EndoCapsule™ use CCD (1920x1080). The study concluded that the image quality from a clinical perspective was 69 % better for PillCam™SB [42]. This shows the importance of how the available power is used and that higher resolution doesn't always mean better image quality. This favors the CMOS image sensor. There isn't many camera chips available, designed specifically for WCE. According to their website, OmniVision is the leading manufacturer of CMOS image sensors for medical use. However, no camera chips are marketed specifically towards WCE. The closest are camera chips for wired endoscopy. In the literature camera chips designed specifically for WCEs are proposed, such as [39].

Review of Cameras Available in the Market

According to a set of market analyses of CMOS imaging sensors, the major companies are OmniVision Technologies, ams AG, Fujikura Sony, Canon, Himax Technologies, and Samsung [43, 44]. We did a search for each company, and the smallest CMOS chip from each company is summarized in Table 2.3 with information gathered from the respective datasheets [45–51].

The leading manufacturer and supplier of CMOS image sensors for medical use is OmniVision with a 70 % market share [52]. They have a large number of various CMOS image sensors developed specifically for medical equipment like endoscopy. They currently hold the world record for the smallest CMOS image sensor available commercially. In addition, OmniVision provides wafer-level camera modules, which significantly simplifies the implementation and mounting of the camera in applications. The smallest OmniVision camera modules do not include an ADC, meaning that the output is analog, requiring extra hardware before feeding the image data to an MCU [53]. ams AG, on the other hand, only has one camera chip for each application. For medical use, they currently have three products: a CMOS image sensor and two complete camera modules, one for surface mounting and one for cable mounting. The NanEyeC camera module for surface mounting is the smallest complete camera module on the market. This module even has a built-in ADC, giving a serializes digital output [54]. Fujikura has only

Table 2.3: Comparison of the smallest CMOS camera module from the major companies in CMOS sensor imaging. Only ams AG, OmniVision and Fujikura have products specifically designed for medical products like endoscopy. For the rest the smallest available CMOS image sensors was chosen for comparison.

Manufacturer	Size [mm]	Pixel size [µm]	Resolution [pixel]	Power [mW]
ams AG	1.00 x 1.00	3.00	250 x 250	3.1
OmniVision	0.575 x 0.575	1.75	200 x 200	25
Fujikura	1.65 x 1.65	1.75	400 x 400	N/A
Sony	6.22 x 6.22	2.74	2464 x 2056	N/A
Canon	8.80 x 7.00	3.40	2592 x 2056	N/A
Himax Tech.	2.30 x 1.55	1.12	1200 x 720	40
Samsung	5.00 x 3.73	1.12	2592 x 1944	N/A

one CMOS image sensor for medical applications. However, they only sell it as a complete camera module for wire mounting, to be used for endoscopy. For Sony, we did not find any public information about their CMOS image sensor developed for the medical market. They have some information on the professional product site, but nothing that can be used for comparison. The same goes for Samsung, Canon, and Himax Technologies. The smallest available CMOS image sensors were chosen for all of them. The table shows that the four latter have very high resolutions compared to ams AG, OmniVision, and Fujikura, which are made specifically for medical equipment, such as endoscopy. During the reviewing process, it is clear that ams AG definitely is best on public documentation of their products. OmniVision has a short product brief for each product, and public documentation is only available for the older products. The rest were even worse, and most did not even list the power consumption. It is worth noting that no one is selling camera solutions marketed directly to WCE today.

From our review, ams AG and OmniVision are the only companies providing camera modules for surface mounting. For an application like WCE, we require digital data to be able to process the data using an MCU. The complete camera module, NanEyeC, from ams AG, has this feature built-in. In contrast, OmniVision has an additional ASIC to bridge an MCU or FPGA with the camera module. This ASIC called OV426 is a 64-pin Ball Grid Array (BGA) chip measuring 6x6 mm. The input to the chip is the analog signal from the OV6946 or OV6948 CMOS image sensors. The chip has registers and memory that can be programmed by the user with the Serial Camera Control Bus (SCCB) interface, OmniVisions version of Inter Integrated Circuit (I2C). The chip can then perform post-processing based on the users' preference to improve the image quality. The output from the chip is a 10-bit digital video port, which means that 10-bits are transmitted at the same time in parallel [55]. The most important specifications for the complete camera module from ams AG, NanEyeC, and the two camera modules from OmniVision, OVM6946 and OVM6948, are summarized in Table 2.4. The power consumption

Table 2.4: Comparison of the camera module NanEyeC from ams AG and the two camera modules OVM6946 and OVM6948 from OmniVision. NanEyeC has digital output, while OVM6946 and OVM6948 has analog input and requires the ASIC OV426 to interface with a MCU.

Camera Module	Size [mm]	Pixel size [µm]	Resolution [pixel]	Power [mW]
NanEyeC	1.050 x 1.050 x 2.292	2.4	320 x 320	9.3
OVM6946	0.650 x 0.650 x 1.158	1.75	200 x 200	25
OVM6946	1.100 x 1.100 x 2.266	1.75	400 x 400	25

of the OV426 ASIC is not listed in the product brief. This makes the comparison between ams AG and OmniVision unfair, but still, the power consumption of the NanEyeC is substantially lower than the OmniVision camera modules.

2.1.3 MCU

In WCEs the MCU needs to be small, ultra-low power and have just enough memory and processing power for all the tasks. However, it needs a balance between size and power efficiency. Smaller transistor technology will significantly increase both dynamic and static power usage. 180 nm technology is commonly used for ultra-low power Central Processing Unit (CPU) cores, as this keeps the static/leakage power small compared to the dynamic power. When the transistor size becomes smaller, the leakage current will increase. For transistor technology smaller than 90 nm, the static power becomes larger than the dynamic, as the leakage current flows even in the inactive state. This limits the performance per footprint area of the chip [56, chap. 1].

The total power usage for CMOS chips is the sum of dynamic and static power. The dynamic power is given by

$$\text{Dynamic Power} = \alpha f C V = C V^2 f \approx C f^3, \tag{2.1}$$

where α is the activity factor, f the clock frequency, C the capacitive load and V the operating voltage [56, chap. 1]. It's clear that the power usage is very dependent on frequency and voltage.

Three different architectures are common for ultra-low power MCUs: 8-bit, 16-bit, and 32-bit. In [7] they compare a set of low power MCUs with 8, 16 and 32-bit architecture. The most important takeaway is that 32-bit MCUs are most efficient in terms of processing power per Joule. It is also worth noting that the energy is lower while the power is higher for 32-bit MCUs at higher frequencies. This makes the 32-bit MCUs faster and more energy-efficient compared to the 8- and 16-bit MCUs. Both [7] and [57] talk highly of the ARM Cortex-M series and sets it as a benchmark for the low power MCUs.

When comparing different CPU-cores that perform integer operations, it is common to use the Dhrystone benchmark. The benchmark is often referred to as

DMIPS/MHz - giving the number of clock cycles used for one Dhrystone loop [58]. This benchmark is given in the datasheets and is used to compare Microprocessor Units (MPUs). In [57] they compare three 32-bit, one 16-bit and one 8-bit MPUs. From the Dhrystone benchmark, they conclude that the two 32-bit MPUs ARM Cortex-M0 and Cortex-M3 are outperforming the competitors in execution speed and code size. The energy measurements show that they are the most energy-efficient, with the lowest power consumption in both active and sleep modes.

According to ARM, the Cortex-M family is made to be a cost-effective, ultra-low-power solution for embedded systems that require long battery life and moderate processing power. They have achieved this by using 180 nm ULL technology that keeps the leakage/static power low compared to smaller transistor technology [59].

ASIC is a more optimized solution than commercially available MCUs, but they are expensive to design and less flexible. However for very large production quantities they are cost effective [7]. FPGA are also an alternative. In [60] they present a complete WCE prototype using only an FPGA for memory, system control and image processing. The power consumption is comparable to ASIC implementations, but slightly worse. The benefits of using FPGA over ASIC is the flexibility and lower design costs. Compared to both FPGA and ASIC, MCUs typically have lower design costs given the use of libraries and Software Development Kit (SDK) provided by the manufacturer [61].

2.1.4 Battery

Powering the active components of the capsule requires an internal or external power source. The commercially available WCEs use batteries. The literature proposes 3D coil inductive powering to replace the batteries. However, it requires the patient to wear bulky and heavy equipment during the procedure, which is not ideal [62].

Review papers [63] and [5] from 2008 and 2011 respectively, states that all WCEs use mercury-free silver-oxide batteries, as they are the only batteries approved for clinical use. However, the batteries are not the most efficient batteries on the market in terms of power to weight ratio. WCEs typically uses two 3 V batteries at 55 mAh.

2.1.5 Transceiver

To enable live view and real time optimizations the WCE needs two-way communication with the equipment outside the body. A transceiver enables this as it works as both a receiver and a transmitter. Typical power usage of a transceiver in WCEs is 20-45 mW according to [8]. Transceivers can be divided into two categories: active and passive/semi-passive. The transmission data rate for a transceiver is typically calculated by

$$DR = \frac{FR \cdot R \cdot PD}{CR}, \tag{2.2}$$

where DR = data rate, FR = frame rate, R = resolution, PD = pixel depth and CR = compression rate [64].

Active Transceiver

The WCEs available on the market uses active transceivers for communication. Figure 2.6 shows a block diagram of a basic active Radio-Frequency (RF) transceiver. The Transmitter (TX) is shown in red, the Receiver (RX) in blue, and the black parts are shared between the receiver and transmitter. Both the transmitter and receiver use the same antenna and the same oscillator. The oscillator generates the carrier wave, which sets the signal's transfer frequency. The signal is sent to a modulator in the transmitter, which adds the signal to the carrier wave. The two main modulation techniques for RF-transmitters are Amplitude Modulation (AM) and Frequency Modulation (FM). The modulated carrier wave is amplified to increase its power, using a Power Amplifier (PA). The antenna then converts the signal to a RF-wave, which are transmitted. The power amplification and the antenna design decide how far the signal can be transmitted. Regulations and other practical aspects limit the transmission power and frequency. For the receiver, the antenna converts the captured RF-waves into a tiny alternating current. This current is amplified by a Low-Noise Amplifier (LNA) amplifier. The amplified signal typically consists of multiple frequencies due to noise from the environment. After amplification, the signal is passed through a band-pass filter, removing much of the unwanted noise. The mixer recognizes the carrier frequency, which is used in the subsequent filter to remove / attenuate the other frequencies [65].

The best-implemented transceiver solution found in the literature is; a transceiver from Zarlink Semiconductor Inc used in Given Imaging WCEs. The power consumption is 5.2 mW at a 2.7 Mbps data rate. Gao et al. implemented a single CMOS chip using ultra-wideband technology. At 10 Mbps data rate the transceiver used 74.5 mW [5]. In [66] they propose an ultra-low-power multi-mode baseband transceiver for capsule endoscopy and compare it with three other proposed solutions using the same technique. The transceivers are normalized for power comparison at a data rate of 10 Mbps using 0.9 V. From best to worst the power consumption are approximately 10.07 µW, 37.68 µW, 80.42 µW and 802.35 µW. M. I. Park et al. propose a FPGA based transceiver utilizing human body communication. At 6 Mbps data rate, the FPGA uses 6.34 mW to transmit image data - the transmitter alone uses 3.7 mW. The receiver uses 114 mW [67].

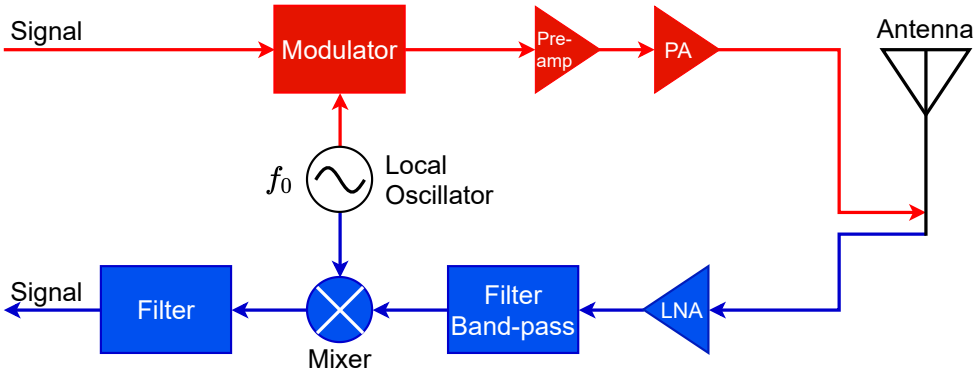


Figure 2.6: Block diagram of a basic active transceiver. The red lines and boxes illustrate the transmitter part, the blue are the receiver part, while the black oscillator and antenna are shared between the transmitter and receiver.

Passive/Semi-Passive Transceiver

A. Khalegi et al. propose a transceiver for medical implants using the backscattering technique. This is a semi-passive transceiver, where the power usage is reduced to only a micro-/nanowatt switch. The system consists of a reader with antennas placed on the body and the semi-passive transceiver in the WCE. The internal components in the WCE are an antenna and an active micro/nano switch - controlled by a serialized data stream. The reader sends an RF signal to the WCE. The switching modulates the reflected signal by altering the amplitude/phase of the incoming signal. The reader can then demodulate the backscattered signal [1, 8, 9]. This is explained in more detail in the next section.

2.2 Backscattering

Backscattering is a technique that enables very low-power wireless communication, were one of the nodes modulates the received RF signal and reflect it back to the source. A common use is Radio-Frequency Identification (RFID) tags. Figure 2.7 shows an illustration of a RFID backscatter system from the book *the RF in RFID* [68].

In RFID systems, they use monostatic antennas in the reader. This means that the transmitter and receiver share one antenna. A circulator is used to limit the coupling between the receiving and transmitting lines, which attenuates the transmitted signal in the receiving channel and the received signal in the transmitter channel. The coupling factor between the antennas limits how much power can be transmitted before the receiving channel is saturated. When the receiving channel is saturated, it is impossible to detect the received signal.

In the illustration, they have split the reader antenna in a receiver and transmitter antenna for clarity; in the actual monostatic antenna implementation, they share one antenna in the same manner as the transceiver in Figure 2.6. In the illustration, the reader transmits a carrier wave to the tag, called the CW signal.

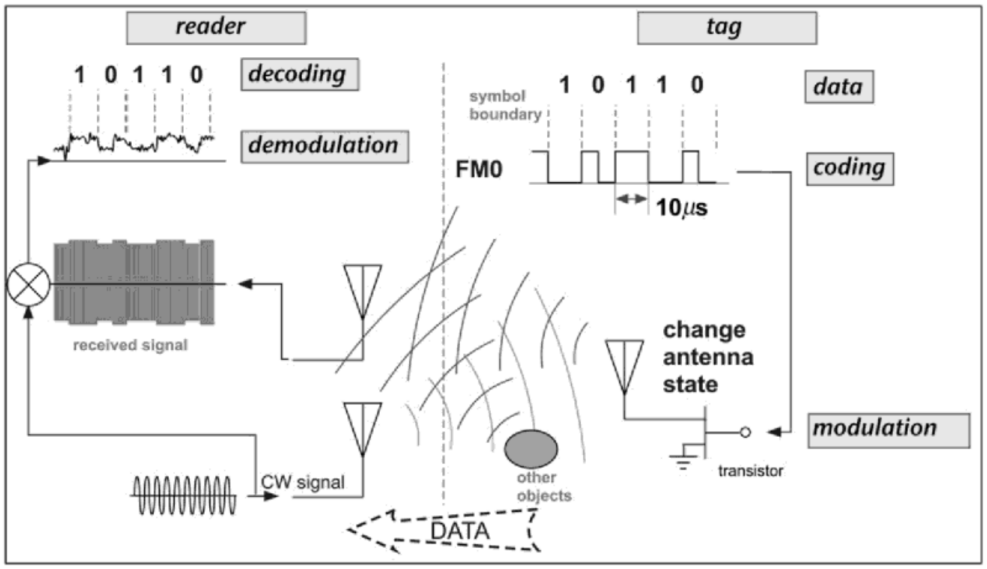


Figure 2.7: Illustration of a RFID system. The image is sourced from the book *The RF in RFID* [68].

The traits of the antenna are dependent on the connected load. The two end cases are the smallest load possible, open circuit, and the largest load possible, short circuit. These end cases are implemented by connecting the antenna to a transistor acting as a switch. Either the antenna is connected to the ground (open circuit) or the signal line (short circuit), as illustrated in Figure 2.7. When the antenna load is low impedance, the induced current in the antenna will be the same as in the transmitting antenna. By the principle of reciprocity in electromagnetic theory, this means that the antenna that receives a wave can transmit the same wave. This wave can be detected by the reader antenna, which sent the wave in the first place. The signal received by the reader is called the backscattered signal. The other case is an open circuit, where the transistor functions as a load with a large impedance. This results in no induced current in the antenna, which means no waves are transmitted from the tag antenna. This impedance manipulation is used to modulate the backscattered signal by using the 1's and 0's to turn on and off the transistor.

In Figure 2.7, the RFID code is `10110`. The RFID tags use some variation of Frequency-Shift Keying (FSK). The variation used in the illustration is subcarrier modulation called FM0. This is why the waveform does not directly correlate to the code's 1's and 0's. The modulated signal is transmitted from the tag antenna to the reader antenna. In addition to this signal, the antenna receives many other signals from the environment. The signal from the tag must be extracted before it can be demodulated and decoded. The common local oscillator with the frequency, f_0, is used to demodulate the received signal.

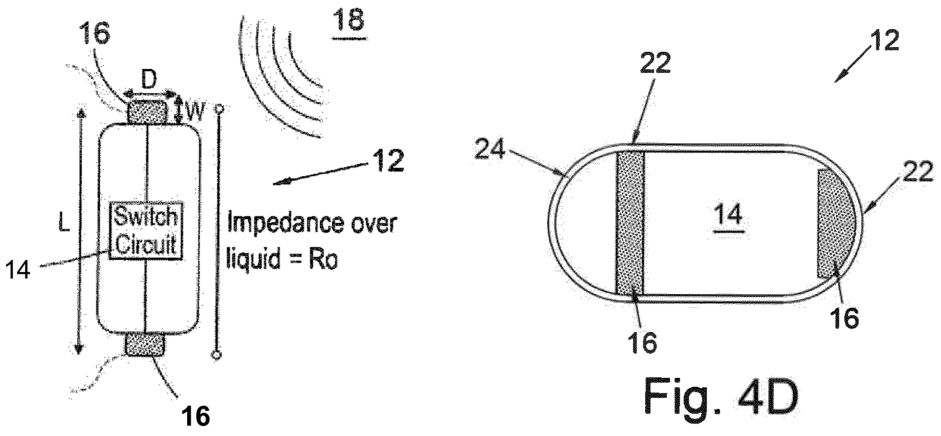


Figure 2.8: Illustration of a general medical implant (a) and WCE (b) prepared for backscatter communication. The images are sourced from the patent [1].

2.2.1 Backscattering for Wireless Video Capsule Endoscopy

A. Khaleghi and I. Balasingham have a patent called Medical Implant with Wireless Communication [1]. This patent covers their backscattering technique and the implementation for WCEs and other use cases. Figure 2.8a shows their illustration of a general backscatter implant and Figure 2.8b shows their illustration of a WCE for use in backscatter communication.

Figure 2.8a illustrates the medical implant device (12). The device consists of a data source and a non-self-resonant antenna used for backscattering communication. The antenna comprises two conductive patches (16) spaced apart at a set distance. The antenna is controlled by a switching circuit (14) that switches between a conductive pathway and coupling via body tissue (18), modifying the load's impedance connected to the antenna.

For the WCE application in Figure 2.8b the device (12) has a protective non-conductive outer layer (24) - the capsule. The two conductive patches (16) are expanded to copper foils. One of the patches is fitted in the rounded space at the end without the camera, and one circular copper strip, encasing the wall of the capsule right before the other rounded end of the capsule. The capsule creates a gap (22) between the body tissue and the patches; this gap operates as a capacitive element. The switching circuit connects to the common ground of the electronics in the capsule instead of the body tissue.

The backscatter system developed by A. Khalegi et al. is illustrated in the block diagram in Figure 2.9, consisting of the capsule and the backscatter reader. In addition, a short description of the different modules is provided; for more information, please refer to [1, 8, 9, 69, 70].

The block diagram shows that the backscatter reader is similar to the implementation of the traditional active transceiver in Figure 2.6, except for the input

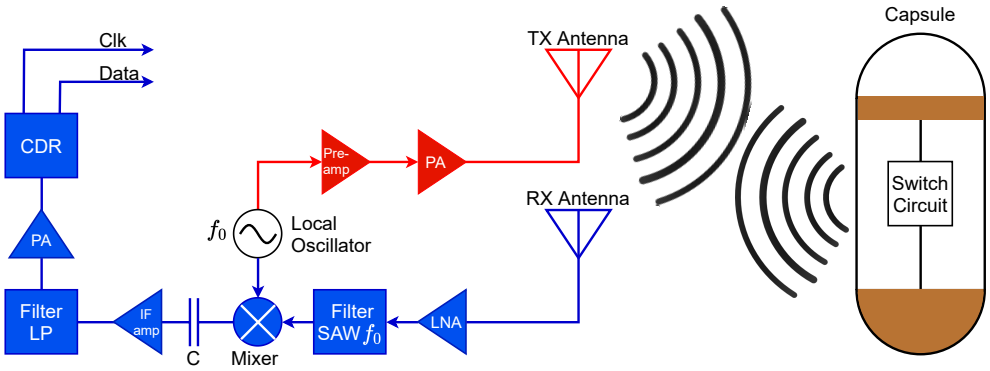


Figure 2.9: Block diagram of the backscatter system. The reader consist of a TX and RX part, shown in red and blue respectively. The local oscillator is shared between the TX and RX.

signal and modulator. However, the backscatter is implemented using bistatic antennas, meaning that it uses a separate antenna for RX and TX. This reduces the coupling, enabling higher transmit power and thus more extended range.

The transmitter part shown in red amplifies the carrier wave with frequency f_0, generated by the local oscillator, and transmits the wave to the capsule. The capsule modulates the signal, and the backscattered wave is received by the RX antenna. The signal first passes through a LNA, which amplifies the very low-power signal without considerably degrading the Signal-to-Noise Ratio (SNR) of the signal. Next, the amplified signal is passed through a Surface Acoustic Wave (SAW) filter with center frequency f_0. The SAW filter works as a band-pass filter with low insertion loss and good rejection.

The mixer is used to demodulate the received signal. The mixer has three ports. The local oscillator works as a gate turning it on and off. The signal on the right hand side is called RF or f_{RF} and the signal on the left hand side is called Intermediate Frequency (IF) or f_{IF}. This system is a zero-IF architecture. This means that the local oscillator frequency, f_{LO}, is the same as the carrier frequency: $f_{RF} = f_{LO}$. This gives a direct conversion with only one frequency conversion. The incoming RF-signal is sinusoidal with angular modulation. This signal is decomposed into two amplitude-modulated sinusoidal waves in the mixer. The two waves have a phase offset of a one-quarter phase. All the three waves have the same center frequency f_0. The two waves are known as In-Phase (I) and Quadrature (Q), and combined comprises the IF-signal also known as the I/Q-signal.

A drawback of direct conversion is that the output gets a Direct Current (DC)-offset. The subsequent capacitor removes the DC component. The following IF amplifier increases the power of the IF-signal. The Low-Pass (LP) filter lets the frequencies below the cutoff frequency of the LP filter pass, unattenuated. The signal is then amplified one last time before entering the Clock Data Recovery (CDR) module. This is a complex module that recovers the received data and the synchronized clock, with frequency f_0.

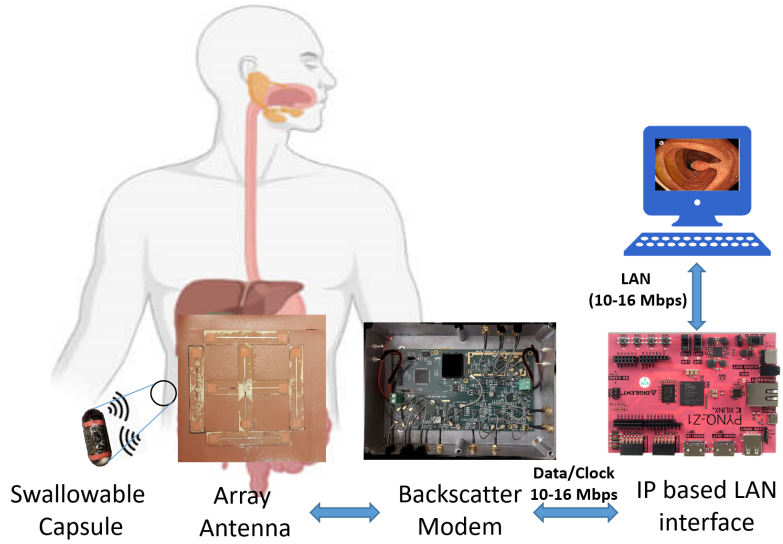


Figure 2.10: Illustration of the complete backscatter system described in [70]. The image is a modified version of Fig. 1 in [70].

Figure 2.10 shows the complete backscatter system described in [70]. The swallowable WCE is inside the GI-tract, where it receives the carrier wave from the antenna array and modulates it with the camera data. The antenna array placed on the abdomen receives the backscattered data and feeds it to the backscatter modem. The modem recovers the data and synchronized clock before sending it to the FPGA. Extraction of the data and image post-processing is performed by the FPGA before it sends the video stream to a PC over a Local Area Network (LAN) connection. Finally, the video stream is displayed on the PC. The system operates at 434 MHz, and the maximum transmission power is 250 mW. The backscatter system supports data rates from 8 to 16 Mbps.

2.2.2 Performance Review

In [8] they describe how backscatter is beneficial for biomedical applications with limitations to physical access to the implant and battery capacity. By removing the need for an active transmitter in the capsule, the power usage can be reduced by 20-45 mW. The design is less complex - saving space and production costs.

In [9, 69] they convey the use of backscattering in WCEs for a continuous data stream. The capsule is powered by a battery, where the wireless communication system using the backscattering technique only uses 3 µW for the active micro/nano-switch. By measurements, they show that 8 Mbps data transfer is feasible at 6 cm depth and 1 Mbps at 8 cm depth [9]. In an animal experiment, they demonstrated that a 1 Mbps data rate at 13 cm depth is feasible. The limiting factors of the backscatter systems data rates and range are the capsule antenna size, loss in the biological tissue, and safety regulations limiting the power of the transmitted RF-signal [69].

Chapter 3

Design Principals and Methodologies

This chapter briefly introduces some of the design principles and methods used in the implementation. For embedded programming; DMA, SPI and bit banging is presented. In this work, six PCBs will be designed. The PCB design tool Altium Designer is introduced with a brief overview of common manufacturing limitations that must be taken into account in PCB design. Finally, some properties of the various SMD components used in this design are introduced.

3.1 Direct Memory Access (DMA)

DMA is a method that allows a peripheral device to control the CPU memory bus directly. This relives the CPU from waiting on the peripherals memory accesses. The CPU are then free to perform other tasks while the peripheral uses the memory bus. This can substantially speed up a system with peripherals such as a camera that writes large data sets to the memory, potentially many times per second. A simplified functional description of DMA is illustrated in Figure 3.1. The order of execution is annotated with the numbers (1) to (6). (1): The peripheral device is ready to access the memory and sends a DMA Request (REQ) to the DMA controller. (2): The DMA controller requests control over the CPU buses. (3): The CPU have stopped using the CPU buses and sends a bus Acknowledge (ACK) signal, giving the DMA controller control over the CPU buses. (4): The DMA controller sends the DMA ACK signal to the peripheral device, giving the peripheral access to the memory. (5): The DMA controller puts the requested memory address on the memory bus, and the operation to read or write. (6): The peripheral device access the memory directly. For each transferred byte, the memory address is incremented by one. When the operation is complete, the DMA controller returns control over the CPU buses to the CPU [71, chap. 4].

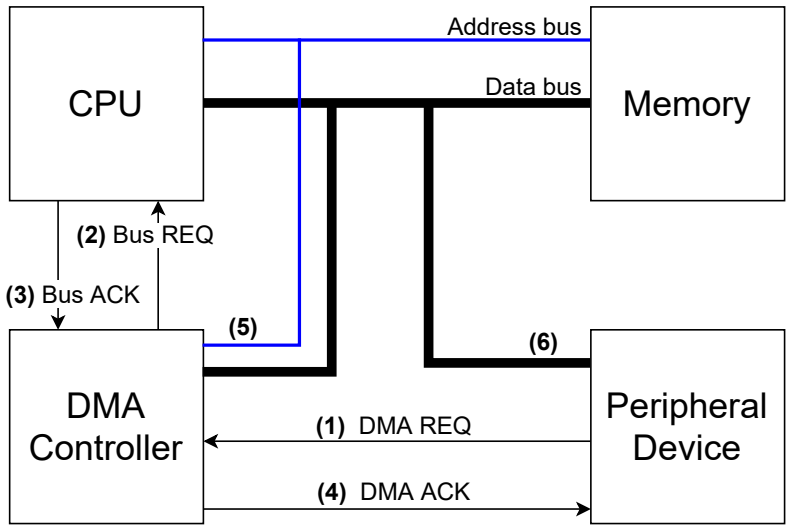


Figure 3.1: Simplified functional block diagram for DMA

3.2 Serial Peripheral Interface (SPI)

The SPI communication protocol are typically used for communication between a MCU and an external device. The big advantage of SPI over I2C and Universal Asynchronous Receiver-Transmitter (UART) is that any number of bits can be transmitted or received in a continuous stream. I2C and UART use packets with start and stop bits, while SPI use a clock for synchronization between the transmitter and receiver. This enables transfer speeds up to 10 Mbps, which is significantly faster than both I2C and UART. The SPI protocol requires one master and can have multiple slaves. Figure 3.2 shows a SPI block diagram. Master Output/Slave Input (MOSI) are used for transmission from master to slave and send the Most Significant Bit (MSB) first. Master Input/Slave Output (MISO) is used for transmission from slave to master and sends the Least Significant Bit (LSB) first. The Synchronization Clock (SCLK) are used for synchronization and sets the transfer speed - one bit is transferred each clock cycle. The final signal is used to activate the slave and choose which slave is active in a multiple slave configuration. The signal has many names, but the most common are Chip Select (CS) and Slave Select (SS). STM32 uses Negative Slave Select (NSS) to state that the signal is active low explicitly. Figure 3.2 also shows an example of a transfer of one byte from master to slave. The byte contains the data `10000111`, as the MSB is sent first. The MISO line are not included in the example since it is not used [72].

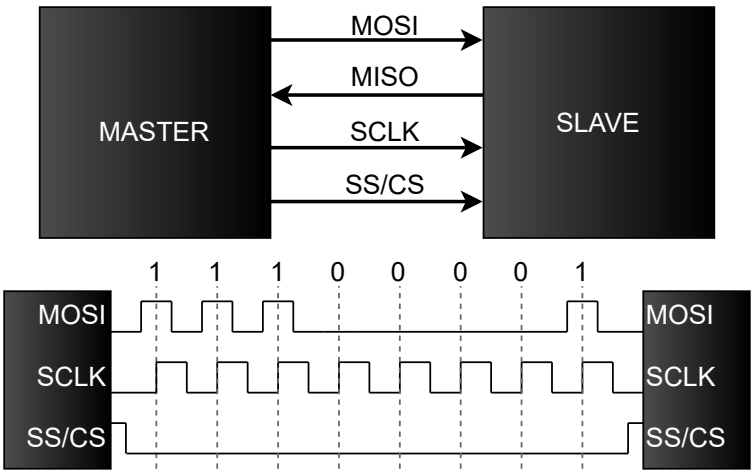


Figure 3.2: SPI block diagram and an example showing the transfer of one byte from master to slave

3.3 Bit Banging

Bit banging is the most basic form of serial communication, as it is implemented entirely in software and does not require any dedicated hardware. Bit banging is commonly used in embedded systems with low-performance requirements to save costs and space. The bit banging scheme can be used to emulate dedicated hardware like SPI, I2C and UART. Bit banging transmits data by using software to encode the data into a signal by toggling the state of the MCUs General-Purpose Input/Output (GPIO) pin. In order to receive data, the SPI pin is sampled at predetermined intervals determined by the transmission baud rate. As a result, bit banging is more complex to implement than existing hardware implementations of SPI, for example. All timing and synchronization are handled by the hardware with the SPI protocol. On the other hand, with a bit banging implementation of SPI, the developer must take care of all data and clock line synchronization and timing. This adds a software overhead which may lead to the system not meeting real-time deadlines. Bit banging has the advantage of portability, as the software implementation will be consistent across all systems, whereas MCU manufacturers may implement their hardware differently. While bit banging consumes several CPU cycles, a SPI solution utilizing DMA allows the CPU to perform other tasks while transferring serial data. In comparison to hardware implementations, the software overhead will limit the maximum baudrate achievable with bit banging [73].

3.4 PCB Design and Manufacturing

There are several PCB design tools available. We opted to use the industry-leading design software Altium Designer version 21.8.1, licensed by NTNU. Altium Designer offers a complete environment for PCB design. Altium Designer is a complex tool with a steep learning curve. It offers integrated support for version control, like git. PCB design starts by making a schematic of the circuit. To add a component to the schematic, a library entry must be made for the component, including the schematic, footprint, 3D model, and information. Altium Vault offers library entries for a large number of components. When the schematic is finished, the schematic is compiled to check for connections error or other violations. The software comes with a set of default rules that fit most purposes, but these can be altered to the users' preference. The PCB can then be generated by defining a stackup. The stackup sets the number of layers, electrical properties, and thickness. With a defined stackup the user can open the PCB editor. The PCB editor has three different modes: board planning, 2D, and 3D. In board planning mode, the user can define the PCB shape. In 2D mode, the user can import the compiled schematic, where the layout can be defined by dragging the components into place. Before connecting the components with traces, the rules must be set for the current project. Altium has a set of default rules suitable for larger PCB designs. There are many practical aspects and limitations when it comes to PCB manufacturing. Each PCB manufacturer has its own sets of specifications for the minimum limits of what they can produce. This list of limitations has to be implemented as rules in the PCB editor of Altium Designer. The user can draw traces between the components add vias and polygons with the correct rules defined. If a rule is violated, the software tries to help the user avoid it or gives an error. Not all violations are revealed in real-time. When the PCB design is finalized, it is compiled, and a rule check is run, which returns both warnings and errors as specified by the user. The 3D tool is especially nice to visualize the real-life PCB with all the components. Altium Designer also offers a multi-board project where several PCB projects can be merged into one system. The schematic illustrates all the electrical connections between the different PCBs. A 3D assembly mode lets the user connect the 3D models and analyze practical issues like collisions and misalignment [74]. The certification for rigid PCB designs is made by the IPC Association Connecting Electronic Industries®. The governing standard now is IPC-2221A, a 124 page long PDF discussing all aspects of rigid PDF design and manufacturing requirements. The guidelines in this standard will be followed when designing the PCBs. The document is available in [75]. There exist several online calculators and tools to calculate a large number of properties in PCB design, e.g., how traces and vias choices affect impedance, thermal properties, cross talk, and much more. The Saturn PCB Toolkit, popular in the industry, will be used for these calculations/simulations [76].

3.4.1 Via and Stackup

Figure 3.3 shows a typical stackup for a 4-layer PCB and the different via types that are possible to use. A 4-layer PCB means that stackup is comprised of four copper layers that can conduct current. The top and bottom of a PCB have two layers on top of the outer copper layers. The top layer, called *Overlay*, is the silkscreen, which is a non-inductive epoxy inc used to identify components, add logos, and other information. The silkscreen is typically white. The next layer, called *Solder Mask* is the layer that gives the color to the PCB, the most common and cheapest color is green. The solder mask is used to prevent copper oxidation and avoid soldering bridges. Openings in the solder mask are made wherever a solder is needed. For automatic assembly of PCBs solder mask is essential. Between each of the copper layers, there is a dielectric. The dielectric between copper layer 1-2 and 3-4 is called *Core* and the dielectric between layer 2-3 is called *Prepreg*. Both are made using the same materials. The most common is FR-4, a flame retardant woven glass-reinforced epoxy resin laminate. It is these layers that give the strength and stability to the PCB. The difference between core and prepreg is the quality. The core is often made by dedicated core manufacturers using a process that ensures maximum strength and a dielectric constant that is the same in the entire core. The prepreg is made as a part of the PCB manufacturing process, and thus the laminate gets partially dried during the process; hence it is less rigid than the core and has a dielectric constant that might vary. For most purposes, the use of prepreg is not a problem, as the core ensures a rigid material. The core method is possible to use for all layers but is very expensive and mostly just used in military applications [77]. The naming of the different layers are *Top Layer*, *Bottom Layer*. *GND* and *PWR*. It is a common practice to have dedicated power and ground planes to reduce Electromagnetic Interference (EMI) and to comply to Electromagnetic Compatibility (EMC) standards [78]. This will also increases the resistance to Electrostatic Discharge (ESD) [79].

Figure 3.3: Typical stack up for a 4-layer PCB, with all possible via types for a 4-layer PCB

The different via types are *Thru/Through Hole*, *Blind*, *Buried*, *µVia*, and *SkipV* - all are illustrated in Figure 3.3 in a 4-layer stackup. All vias can also be mirrored to "start" from the bottom of the PCB. Through-hole vias are the most common and cheapest to make. These can be drilled after the complete stackup of the PCB is finished. The other types must be considered during the manufacturing of the stackup and increases manufacturing time and cost.

The through-hole via illustrated in Figure 3.3 is a plated through-hole via. This is the most common type and enables the designer to connect to the required layers electrically. It is also possible to use non-plated through-hole vias, the cheapest and fastest option - mostly used for mounting holes in multi-layer designs. The other types of vias are commonly used in PCBs where the space is limited and to reduce inductance [79]. They are also essential in order to fan out components with several connection points underneath like MCUs and FPGAs with minimized packaged types.

Through-hole vias can be capped to make a pad on each side or to cover it entirely. This is known as capping of filling. Through-hole vias can also be tented, meaning it is covered with solder mask. Then the via does not have to be filled. On the other hand, the other via types have to be filled with either epoxy resin or copper. According to an interview with Mike Devine from ExceptionPCB [80], is filling especially important for buried vias because trapped air inside a PCB can cause damage and delamination or change the electrical properties of the PCB due to thermal changes.

3.4.2 Trace

The traces are lines of copper electrically connecting the components. The designer has to consider the traces' length, width, thickness, and placement. These considerations are application-driven, with a balance regarding cost, performance, and board size/density. The application aspects are design requirements like speed optimization, current, voltage, coupling mitigation, and noise.

The default trace size in Altium Designer, 0.254 mm / 10 mils, fit the purpose for most general applications without space limitations. A general rule is that traces narrower than 0.127 mm / 5 mils and spacing between traces less than 0.127 mm / 5 mils increases manufacturing costs, as it requires more expensive high-end tools for both manufacturing and inspection [81]. The standard copper pour thickness is 1 ounce (oz); this equals 0.0347 mm / 1.378 mils. Thinner or thicker copper pours typically increase manufacturing costs [82].

To calculate the required width of a trace, and how it will affect the signal, a set of equations based on the IPC-2221A standard is provided: Equations 3.1, 3.2, and 3.3 [81]. The IPC-2221A standard uses the following values for the constants, valid for the top and bottom layer: k = 0.048, b = 0.44, and c = 0.725. For the inner layers the requirements are lower, so the outer layer variables can be used for calculations for all layers, as the inner layers are more complicated with several dependencies [75]. In the equations T = temperature, R = resistance, ρ = resistivity of copper, and α = temperature coefficient of copper. According to [83], $\rho = 1.709776 \cdot 10^{-8}$ and $\alpha = 0.000850$ at 25°C.

$$Area = \left(\frac{Current}{k \cdot T_{rise}^{b}} \right)^{1/c} \tag{3.1}$$

$$Width = \frac{Area}{Thickness \cdot 1.378\,\text{mils/oz}} \tag{3.2}$$

$$R = \rho \frac{Length}{Thickness \cdot Width}(1 + \alpha(T - 25)) \tag{3.3}$$

A common practice for PCBs that might require rework and modifications is to use teardrops on the traces. This feature gives the trace a larger connection surface to pads, making it more resistant to heat. Especially in miniaturized designs where the traces are narrow, there is an increased risk to burn of traces when rework is done by hand soldering - teardrops counteract this problem. Figure 3.4 shows a screenshot from Altium Designer displaying the four different types of teardrops and their properties. *Via/TH Pad* is the type connecting the traces to pads and vias, *SMD Pad* is the connection of traces and pads for SMD components, which are the most important in terms of rework, *Tracks* is the transition between two different trace widths, and finally, *T-Junction* which removes the 90-degree corners.

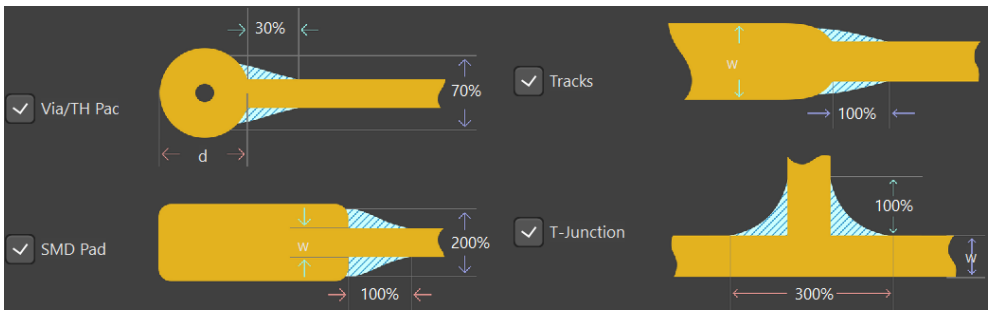


Figure 3.4: Screenshot from Altium Designer displaying the four different types of teardrops and its properties

3.4.3 Manufacturer Limitations

The most common limitations specified by PCB manufacturers are illustrated in Figure 3.5. The figure illustrates a cutout of a PCB with vias and traces. The figure shows an outer layer at the top and an inner layer at the bottom. The green color is the solder mask, the silver is the plating, the brown is copper, and the yellow is the insulation material. The traces on the top layer, shown in light green, illustrates that the copper is coated with the solder mask. In PCB manufacturing, they differ between the specifications for the outer layer and inner layer. The preceding letters "O" and "I" to the communal abbreviations Annular Ring (AR), Pad to Pad (PP), Trace to Pad (TP), Trace Width (TW), and Trace to Trace (TT) in Figure 3.5 refers to outer and inner layer respectively. In general, the outer layer has stricter tolerances compared to the inner layers.

The AR have several limiting aspects. When soldering a component or pin, an aspect ratio between the hole and Outer Annular Ring (OAR) must be respected

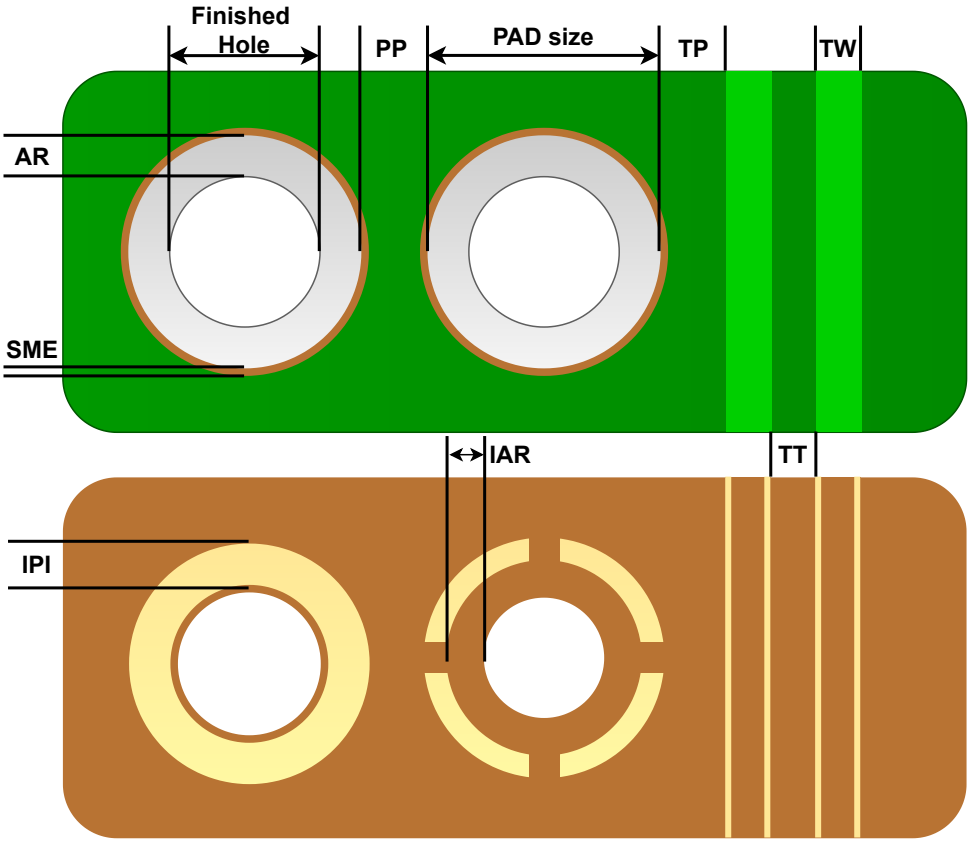


Figure 3.5: Cutout of a PCB with vias and traces, illustrating the most common limitations specified by PCB manufacturers

to enable an IPC certified solder. For vias, the AR must have a certain size to lead the required amount of current and to get a good coating of the inner walls. Here, an aspect ratio must be fulfilled between the height of the via and the diameter/finished hole size. This limits the smallest finished hole sizes for a via. If the relation is not fulfilled, the copper cannot flow through the via. Tolerances that must be included for the finished hole size are drill-bit size, drill-bit wear, and copper distribution. Another important aspect of the OAR is that there are always some inaccuracies when manufacturing a real-life product. Thus the manufacturers' tolerances limit the smallest size they can manufacture without error. This is also the reason for using Solder Mask Expansion (SME) as the hole in the solder mask might not be centered exactly at the same point as the hole and pad. The calculation to find the Inner Annular Ring (IAR) and OAR are the same, but the OAR typically have a higher tolerance than the IAR. The PP, TP, TW and TT are all limited by the manufacturing tools. With the standard manufacturing tools, they are typically limited to 0.127 mm / 5 mils. The most high-end factories use more expensive and advanced tools, using laser technology called High-Density Interconnect (HDI), which are much more accurate. This enables smaller minimum values with lower tolerances. However, this will significantly increase the cost of production as routines and requirements at the factories are much higher, with more expensive equipment. Sharp corners are another restraint imposed by manufacturers. If the angle between two objects is less than 90 degrees, it may result in material build-up or pour bonding. The solder mask, for example, can peel off in corners like this [80, 84, 85].

3.4.4 Thermal Properties

Thermal properties are an important consideration in PCB design. Active components generate heat that must dissipate. There are several techniques to optimize heat dissipation. The use of copper planes connected using thermal vias is the most effective way. The thermal vias are placed directly underneath or in close proximity to the component that generates heat. This will lead the heat down to copper layers which help dissipate the heat out from the PCB. Many components that generate much heat have large ground pads underneath, then in-pad thermal vias are an effective method to lead the heat down to the ground plane. It is beneficial to have several small thermal vias compared to a few large. Heat sinks are another option where heat is a large problem. However, this technique does require much space, and is not ideal for miniaturized designs [75, 86, 87].

Thermal simulations are often used to gauge the effectivity and requirement of the heat dissipation techniques. Altium Designer has a built-in thermal simulation tool, with many nice features like graphic visualization of the heat dissipation and problem areas placed upon the 3D-model of the PCB. The Saturn PCB Toolkit is also commonly used for thermal calculations [88, 89].

3.5 SMD Components

SMD components are components that are mounted on top of the PCB using Surface Mount Technology (SMT). This enables the use of pick and place machines and an automated assembly process. The use of SMD components reduce the size, cost, and manufacturing time. For small PCB design they are essential [90]. There is a vast number of different SMD components with varying specifications. When designing PCBs a large part of the design process is finding the SMD components best suited for the application. For this design, the most relevant are the capacitor, resistor, transistor, voltage regulators, and MCU.

3.5.1 Capacitors

The most commonly used SMD capacitors are made with a ceramic dielectric and are known as ceramic capacitors or Multilayer Ceramic Capacitors (MLCC). The ceramic dielectric used is divided into classes 1, 2, 3, and 4. Class 3 and 4 are rarely used today and are no longer standardized by the industry. Class 1 is the most expensive and stable, commonly used in military and medical equipment. Class 2 has a much higher permittivity than class 1; this results in lower stability and accuracy but a much better volumetric capacitance efficiency than class 1. However, class 2 has the drawbacks of a non-linear temperature coefficient and capacitance, to some degree, dependent on the applied voltage. The smaller size and much lower cost make the class 2 capacitors ideal as decoupling capacitors, as space and placement are more crucial than an accurate capacitor value. Another feature of ceramic SMD capacitors worth noting is the acoustic noise that they can create if an alternating electric field is applied. This acoustic noise can affect other parts of the circuit in sensitive equipment. If the frequency of the applied electric field is in the audible range, this can emit a humming/buzzing sound. This trait is less severe for class 1 dielectrics [91].

Due to the differences in the dielectrics of classes 1 and 2, they use different specification conventions. The conventions are called EIA codes, defining three properties of the dielectric. For class 1, the first character is the change of capacitance with temperature as ppm/°C. The second character is the multiplier of the first character, and the last character defines tolerance. In class 2, the two first characters define the temperature operating range, and the third character defines the tolerance.

The most popular class 1 capacitor is C0G, which has 0 drift with temperature changes and only a ±30 ppm/°C tolerance. For class 2, X5R and X7R are the most common, used as decoupling capacitors. Both feature the same tolerance of ±15 %, with the temperature ranges -55°C - 85°C and -55°C - 125 °C, respectively [92].

Decoupling Capacitors

Decoupling capacitors are essential to provide good power integrity, especially for low-power Integrated Circuits (ICs). The primary function of the decoupling ca-

pacitor is to work as a reservoir of extra charge. When logic gates in an IC switch, it can cause current transients. The capacitor ensures that these transients do not affect the supply voltage by either absorbing the excess charge or supplying extra charge by discharging itself. The low-power and high-speed ICs are especially sensitive to noise and disturbance. Power supplies are not perfect and might have noise and voltage fluctuations. A decoupling capacitor prevents voltage fluctuations from affecting the IC and provides a stable supply voltage. High-frequency noise is also removed, as the capacitor provides a low impedance path to the ground. For decoupling capacitors, low impedance ceramic capacitors are used. It is also beneficial to use the smallest package size possible to reduce loop sizes, reducing the impedance further.

There are several ways to calculate or estimate the decoupling capacitor size. Simulation tools like SPICE are often used. The simplest methods is load charging in the time domain,

$$C = \frac{\Delta I \cdot t_{rise}}{2V_{ic}}, \tag{3.4}$$

where ΔI is the change in current drawn by the IC, t_{rise} is the rise time of the transient in the IC and V_{ic} is the supply voltage to the IC [93].

3.5.2 Resistors

SMD resistors are divided into two main categories: thin-film and thick-film. Thick-film resistors are the most prolific SMD component used in electronics. This makes it very cheap and accessible. The significantly superior thin-film resistors are better in every way except cost and the smaller selection of resistor values. The resistive material used in thin-film resistors is 1000 times thinner than the material used in thick-film. It withstands the laser trimming process used to calibrate resistors without forming microcracks at the edges, significantly lowering the tolerances. Compared to thick-film, the benefits of thin-film are lower capacitance, lower parasitic inductance, lower noise, much lower tolerances, and much lower resistor temperature coefficient. Additionally, the thin-film resistor is less prone to aging and remains stable during its lifetime. At very high frequencies, thin-film resistors are essential for a stable design [94].

3.5.3 Transistors

Transistors have two main categories: Bipolar Junction Transistor (BJT) and Field Effect Transistor (FET), where the most common type of FET is Metal Oxide Semiconductor Field Effect Transistor (MOSFET). These two types have different properties based on the way they operate. The BJT is a current control device with low input impedance making it superior for high switching speeds and noise performance. It is also ideal for the high current drive due to high transconductance, which gives a high gain - with analog capabilities on amplification. The MOSFET

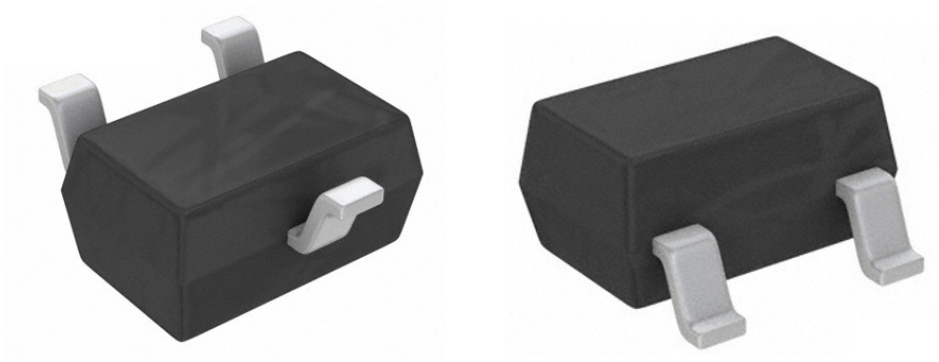


Figure 3.6: Example of a SMD transistor package, SOT323. The image is sourced from [98].

is vastly superior in packing density and high-yield integration of large and complex circuits and functions. Additionally it has less power dissipation compared to the BJT, but the noise margin is lower [95].

SMD transistors comes in various packages, but the most common are Small Outline Transistor (SOT), defined in IPC-7351B [96]. The naming convention is as follows:

$$\textbf{SOT} + Pitch + \textbf{P} + Lead\,Span\,\textbf{X}\,Height\text{-}PinQty,$$

where pitch, lead span, and height are measured as 10^{-2} mm. Figure 3.6 shows an example of SOT with pitch 1.40 mm, lead span 1.35 mm, height 1.00 mm, and 3-pins. This results in SOT140P135X100-3. However, this naming convention is considered inefficient, and thus the industry uses a common grouping to differ the transistor package sizes. This can lead to confusion as a SOT23 is just a SOT package with serial number 23 of SOTs in that size, and simultaneously there exist other non-regulated standards. The package in the figure is also called SOT323, SOT323-3, SC-70, SOT-363, where SOT323 is the most commonly used. Among the most common transistor packages, the smallest in descending order are SOT23, SOT323, and SOT523. Where the footprints sizes are 2.9 mm x 2.4 mm, 2.1 mm x 2.1 mm and 1.6 mm x 1.6 mm, respectively [96, 97].

3.5.4 Voltage Regulators

Voltage regulators are found in almost every electronic device. There are many different types of voltage regulators, but all are classified as either step-up or step-down. Step-up increases the output voltage compared to the input, while step-down decreases the voltage. Further, there are two main types of voltage regulators: linear and switching regulators. Linear regulators can only be used for step-down regulators. They have the advantages of being cheap, noise-free, simple, and thus can be made very small. However, compared to switching regulators, they have poor power efficiency. Switching regulators can be used for both step-down and step-up. However, they are much more expensive and complex and

add noise to the output. For step-down, the general rule is always to use a linear regulator unless the power dissipation is too high, which causes thermal issues.

It is imperative to consider power dissipation when using linear power regulators. If the power dissipation is too high, the linear regulator will burn and destroy itself. In systems running on batteries, it is also crucial to keep the power dissipation at a minimum to avoid excessive drainage of the batteries. The dissipated power, P_D, is calculated as

$$P_D = (V_I - V_O) \cdot I_I, \tag{3.5}$$

where V_I is the input voltage, V_O is the output voltage, and I_I is the input current, which is equal to the output current, I_O, plus the quiescent current, I_Q. The quiescent current is the required current to power the regulator when the load is zero and is negligible compared to the output current for most linear regulators.

When the dissipated power is calculated, the temperature increase of the regulator can be calculated by

$$Temperatureincrease = P_D \cdot R_{\theta JA}, \tag{3.6}$$

where the temperature increases are relative to the ambient temperature, and $R_{\theta JA}$ is the junction-to-ambient thermal resistance. The $R_{\theta JA}$ is given as a constant with unit °C/W. The temperature limit of components varies from the specific design and applications, but a general rule is never to exceed 125°C. It is also important to consider changes in the ambient temperature.

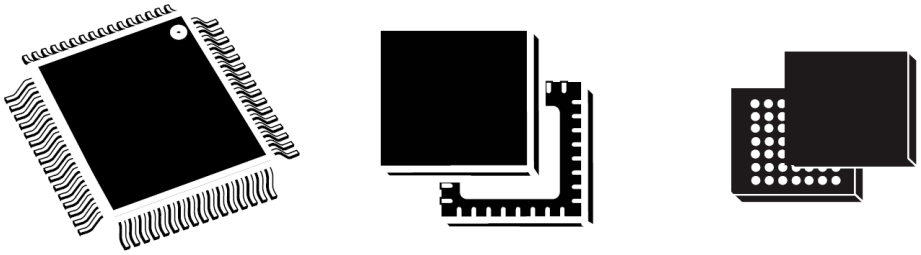
When the regulator operates in the linear region, it can be very power efficient if the differential voltage is small. The minimum differential voltage is called the dropout voltage. When the regulator operates in the dropout region, it can no longer regulate the voltage and is reduced to a resistor. A type of linear voltage regulator called Low Dropout Regulators (LDOs) has a very low dropout voltage, and thus the possibility of being very power efficient. The efficiency of the LDO is calculated by

$$Efficiency = \frac{I_O}{I_O + I_{GND}} \cdot \frac{V_O}{V_I} \cdot 100\%, \tag{3.7}$$

where I_{GND} is the difference between the input and output current, which includes the quiescent current and leakage currents. For space critical and low power designs a LDO is the best option to step-down the supply voltage [99–101].

The LDO comes in various packages. LDOs consists of transistors and are using the same package types discussed for transistors in the previous section. In addition to the SOT packages, LDOs are also available in packages without leads, called Dual Flat Non-leaded (DFN) and Quad Flat Non-leaded (QFN), depending on the pad placement. Figure 3.7 shows an example of DFN LDO package. The pad in the middle is a ground pad [102].

Figure 3.7: Example of a DFN LDO package. The image is sourced from [102].



(a) LQFP **(b)** QFNP **(c)** WLCSP

Figure 3.8: Illustration of the three most common package types for MCUs. The illustrated packages (a) and (b) have 32-pins and package (c) has 36 pins. The figures are sourced from [103].

3.5.5 MCU

There are a waste number of different MCUs available. This enables the user to pick an MCU optimized for the application. MCUs comes in various package types with varying pin number. The three most common package types are illustrated in Figure 3.8. (a) is a Low-profile Quad Flat Package (LQFP) with 32-pins, LQFP-32, (b) is a Quad Flat Non-leaded Package (QFNP) with 32-pins, QFNP-32, and (c) is a Wafer Level Chip Scale Packaging (WLCSP) with 36-pins, WLCSP-36. The LQFP package has gull-wing leads, which makes it suitable for prototypes and evaluation boards, where rework and modifications might be necessary. The drawback of this package is the larger size, compared to the others. The QFNP is smaller than the LQFP, but can only be soldered by hot air reflow or SMT reflow soldering. The large ground pad underneath is ideal for high-performance MCUs that generate much heat. The WLCSP is the smallest package and the most difficult to solder. The pads and solders are not visible for inspection, and much experience is required to feel if the solder is good when using hot air reflow soldering. At factories, they use X-ray imaging to check the solders. This package type has much of the same advantages as the QFNP when it comes to heat dissipation [103].

Chapter 4

Implementation

This section presents the design of the final capsule-size prototype that can interface with the backscatter system, including all the steps leading up to the final design. Figure 4.1 shows a block diagram of the system. The grey box to the left is the capsule, and the black box to the right is the backscatter system. The backscatter system is considered a black box in this work and will only be used in collaboration with the backscatter system developer team. The block diagram inside the capsule consists of a MCU which controls a set of LEDs and has two-way communication with a camera. The output from the camera is sent to the backscatter switch, where the camera data is transmitted wireless using the backscatter system. The backscatter switch and antenna have been designed and tested by the backscatter developer team. The choice of switch and antennas are based on their research. Power management must also be considered, as all of the active components, camera, MCU, LEDs, and backscatter switch, needs power. It was decided that the operating frequency for interfacing between the camera module, MCU, and backscatter switch is 12 MHz, as this is a frequency that can be supported by all three.

It is a large project with a limited time frame. This was not helped by the fact that we never had designed PCBs previously and had to learn PCB design and Altium Designer from scratch. In addition to this, the Covid-19 situation lead to longer delivery times and considerable pressure on the PCB manufacturers. At the same time, there was a massive shortage of components, especially active components like MCUs and LDOs. This shortage led to multiple redesigns because of components going out of stock when we were designing and checking if we could use that component. We learned the hard way that components had to be ordered on the fly.

Due to the time limitations and the assessment of the current situation in the PCB and component market, it was decided not to follow the typical way of developing a new product. The standard way of doing it would, for this case, be to first make a benchtop model with a MCU on a development board and a camera breakout board, develop software, and verify that the interfacing with the camera works correctly. Then the MCU could be chosen based on the data from

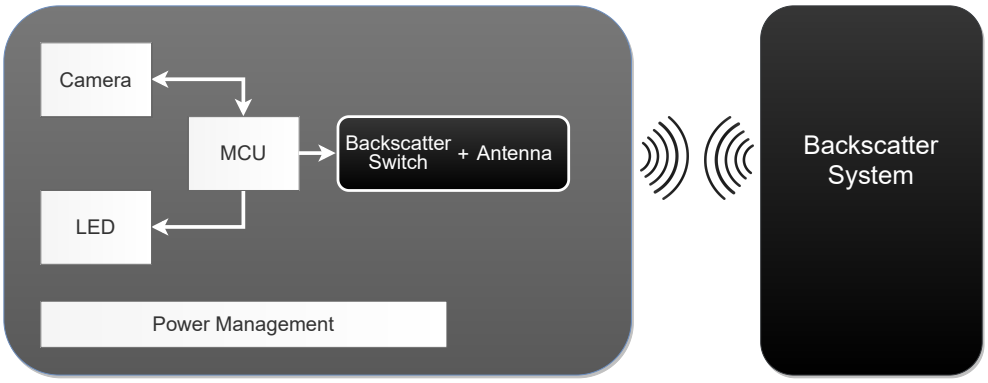


Figure 4.1: Block diagram of the system. The grey box contains the system that will be designed in this work. It contains a MCU, camera, LEDs, power management, and the backscatter switch and antennas that will be used to interface with the backscatter system. The power management includes all the active components. The backscatter system is considered a black box and will only be used as it is with the help of the developer team in this thesis work. The backscatter switch and antennas are also based on their work.

the benchtop model. A custom evaluation board containing all the parts of the capsule-size prototype could then be designed and manufactured. The software would then be tested, and all of the system's functionality verified. Based on the information gathered from this phase, the miniaturized capsule-size prototype could be designed and manufactured. This would then be tested, and typically a new revision would be required before a working prototype was obtained.

From our analysis of the viable timeline for this work, it was decided that the only way it would be possible to get to the finished capsule-size prototype in time was to skip the first step and move directly to the custom evaluation board. When the custom evaluation board was ordered, the 2-4 weeks waiting for it would be used to design the capsule-size prototype and prepare it for ordering. The order could be sent as soon as the custom evaluation board worked. Then when we received the custom evaluation board, the code would be developed and tested. The idea was sound, but with hardware design, there are always problems that need debugging and multiple revisions before it works properly. This turned out to be the case here as well. On the custom evaluation board, everything worked except for the camera. This forced us to do step one of the typical design methodology we skipped. A breakout board PCB for the camera was designed and manufactured, and a MCU on a development board was used to develop the code. Unfortunately, the code development did also come with some unexpected problems. The implication was that there was not enough time to order the capsule-size prototype we had designed. Ideally, we should also make a new custom evaluation board with the knowledge we gained and verify that everything works before moving on to the capsule-size prototype design. The reason why the camera on the custom evaluation board did not work is elaborated in Section 5.1 in the results.

Table 4.1: Names of the public repositories on GitHub containing all the software files made for this project

MCU	Description	GitHub Repository Name
L4+	12-bit SPI	WCE_NUCLEO_L4_NanEyeC_12_bit
L4+	8-bit SPI	WCE_NUCLEO_L4_NanEyeC
L4+	12-bit SPI w/DMA	WCE_NUCLEO_L4_NanEyeC_12_bit_DMA
L4+	Bit banging	WCE_NUCLEO_L4_nanosecond_delay
L4+	Bit banging w/interrupt	WCE_NUCLEO_L4_bit_bang_interrupt
L0	8-bit SPI (8 kB RAM)	WCE_Evaluation_Board_STM32L062K8T6
L0	8-bit SPI w/DMA	WCE_Evaluation_Board_DMA

Table 4.2: Names of the public repositories on GitHub containing all the PCB files made for this project

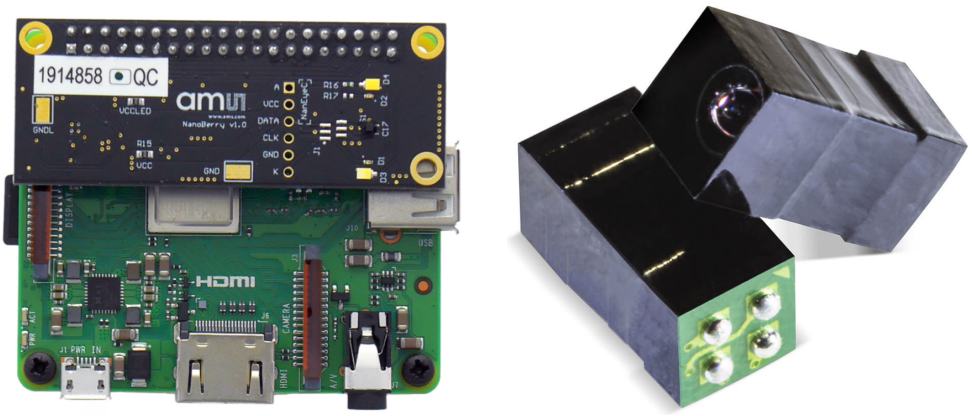
Description	GitHub Repository Name
Custom Evaluation board	EvaluationBoard_NanEyeC_BackScatter
NanEyeC breakout board	WCE_NanEyeC_board
PCB1: Camera and MCU	WCE_Camera_MCU_board
PCB2: Backscatter and Power	WCE_Power_Backscatter_board
PCB3: Battery holder	WCE_Battery_Bottom_board
Programming board	WCE_Programming_board
Multi-board Assembly	WCE_Master_Thesis
3D-models SolidWorks2021	WCE_thesis_3D-files

The first sections of this chapter argue for the choice of the camera module, development platform, and MCU. The chosen camera module is NanEyeC from ams. Next, for the development platform, STM32 is chosen, and the MCU STM32L062K8T6 is used in the custom evaluation board, the STM32L051T8Y7DTR in the capsule-size prototype and STM32L4R5ZI mounted on a NUCLEO-L4R5ZI development board, used for software development. The remaining sections are organized as follows: custom evaluation board, NanEyeC breakout board, capsule-size prototype, programming board, and finally, software development.

Table 4.1 and 4.2 list the software and PCB files that was made for this project. They are all available on GitHub. To access the files, go to

https://github.com/simenodegard/REPOSITORY_NAME

In Table 4.1 the MCU column refers to the MCU that have to be used with that project. L0 is the STM32L062K8T6 MCU and L4+ is the STM32L4R5ZI MCU. The complete Bill of Materials (BOM) for the custom evaluation board and capsule-size prototype are included in Appendix A and B, respectively. More detailed BOMs are available in the respective GitHub repositories.



(a) (b)

Figure 4.2: The NanoBerry evaluation board (a) and NanEyeC camera module (b). The images are sourced from [104] and [105] respectively.

4.1 Camera Module

From the review of cameras available on the market, the best option for our use was the NanEyeC camera module from ams AG [46]. This camera module was the only complete camera module with a digital output, sold in a package for surface mounting. The camera modules from OmniVision had analog outputs and required a custom ASIC to interface with a MCU. This chip was quite large, designed to be placed in the handle of a wired endoscope, and would not fit in our capsule. ams also offered an evaluation kit for the NanEyeC camera module, called NanoBerry. The evaluation board consists of Raspberry Pi 3A+ and a custom extension board with the camera module. We did not have access to the code, just some public reference schematics for the camera board. The Raspberry Pi 3A+ has a Broadcom BCM2837B0 processor, built on a ARM Cortex-A53 CPU with 64-bit architecture, running at 1.4 GHz [104]. The NanoBerry evaluation board and a close-up of the NanEyeC camera module are shown in Figure 4.2.

The NanEyeC miniature camera module has a footprint of only 1 µm × 1 µm and a 320x320 resolution. The camera is made with CMOS-technology and utilizes a high sensitive 2.4-micron rolling shutter pixel. The camera module supports a set of FPS ranging from 0 to 58 FPS. It has two different output modes: Single Ended Interface Mode (SEIM) and Low-Voltage Differential Signaling (LVDS). The LVDS mode is for use in wired endoscopy applications, requiring long signal driving lengths. This mode also requires a FPGA for interfacing with the camera module. The other method is simpler and transfers the output from the 10-bit ADC directly, without any encoding. However, the SEIM method does require a very short distance between the MCU and camera module.

The SEIM protocol implemented by ams is a half-duplex master SPI variation without CS. This means that the MCU do not have any direct control over when the

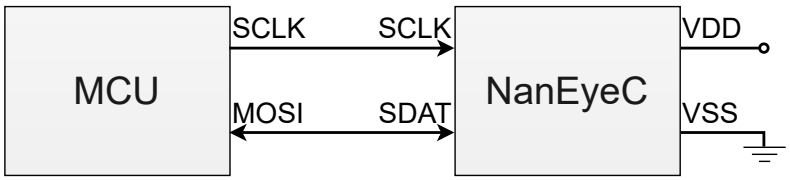


Figure 4.3: The half-duplex master SPI implementation for interfacing between a MCU and the camera module NanEyeC using the SEIM protocol

camera module transmits data on the MOSI line. To not interfere with a transfer from the camera module, the MCU must carefully take care of the synchronization in terms of when the MCU can send data. Figure 4.3 shows a half-duplex master SPI implementation and the four pads of the camera module: VDD, VSS (GND), SDAT and SCLK. VDD is the input voltage, and VSS is the ground connection. SDAT is short for SPI data and has a bidirectional connection to the MOSI port on the MCU. The final port SCLK is short for SPI clock and is an input to the camera module.

The internals of the camera module consists of a module that divides the input clock by 12. This internal clock is called the pixel clock, and for each clock cycle of the pixel clock, the value from one pixel is read. This analog reading is fed to the ADC, which converts it into a 10-bit digital representation. A serialized module adds padding on each side of the 10-bit pixel representation. The 12-bit package made up of a start bit (1-bit), data (10-bit), and a stop bit (1-bit) is called a Pixel Period (PP). The SEIM protocol has three different PPs: Pixel Data, Training Pattern, and End of Frame Pattern. The camera module reads one row at a time from the camera CMOS chip. The SEIM protocol adds 8 PPs of the training pattern before each row and 8 PPs of the end of frame pattern after each frame. Figure 4.4 shows the three different PPs used by the SEIM protocol.

To configure the camera module, it has two 16-bit registers that can be programmed by the user, called Config_0 and Config_1. A write to the registers is comprised of 4 PPs. Figure 4.5 shows two PPs, used to write to a register. The first four bits are the code used to recognize that a register write is starting. The next three bits are the register address. The address for Config_0 is `000` and `001` for Config_1. The subsequent 16-bits are the register content, and the last bit, called reset, is used to separate the two register writes. Detailed information about the registers can be found in the NanEyeC datasheet.

The camera module has an internal state machine, which simplifies synchronization between the MCU and the camera module. Figure 4.6 show the state machine and some information about each state, summarizing the datasheet. The blue color indicates that the camera module is in output mode, while the grey indicates input mode. The SEIM protocol can be used with both 8-bit and 12-bit SPI. The information in the boxes above and below the states are only valid for the 12-bit implementation.

Bit #	1	2	3	4	5	6	7	8	9	10	11	12
Function	Start	**Pixel Data** (10-bit)										Stop
Content	1	MSB									LSB	0

Bit #	1	2	3	4	5	6	7	8	9	10	11	12
Function	Start	**Training Pattern** (10-bit)										Stop
Content	0	1	0	1	0	1	0	1	0	1	0	1

Bit #	1	2	3	4	5	6	7	8	9	10	11	12
Function	Start	**End of Frame Pattern** (10-bit)										Stop
Content	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4.4: The three different PPs used by the SEIM protocol. From top to bottom: Pixel Data, Training Pattern, and End of Frame Pattern.

Bit #	1	2	3	4	5	6	7	8	9	···	22	23	24
Function	Update Code				Register address			Register Content (16-bit)					Reset
Content	1	0	0	1	**0**	**0**	**X**	MSB				LSB	0

Figure 4.5: The sequence used for a register write, consisting of two PPs for each of the two registers. The first four bits are the code used to recognize that a register write is starting. The next three bits are the register address. The address for Config_0 is `000` and `001` for Config_1. The subsequent 16-bits are the register content, and the last bit, called reset, is used to separate the two register writes.

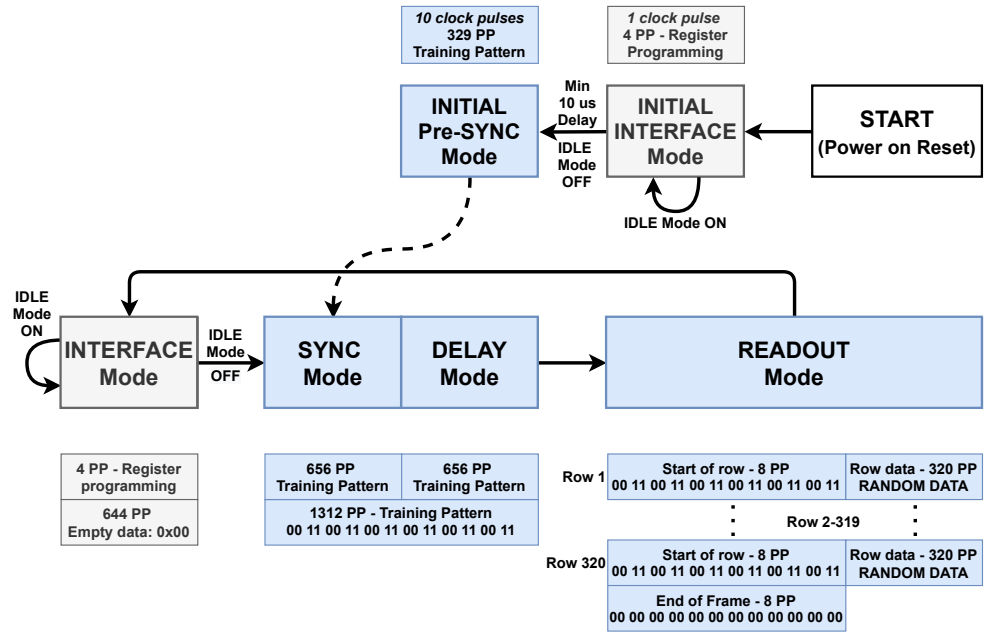


Figure 4.6: State machine for the NanEyeC camera module, with information about each state, summarizing the datasheet. The blue color indicates that the camera module is in output mode, while the grey indicates input mode. The state machine is valid for the 12-bit SPI implementation.

The state machine is comprised of two main parts: a startup sequence and the main while loop. The startup sequence consists of three states. When the camera module is powered on, it goes to the state START and immediately moves to the INITIAL INTERFACE state. In this state, it waits for a single clock pulse from the MCU, which activates the camera module. The MCU must then program the registers and set the IDLE bit low to disable idle mode. After the register programming, it moves to the next state, but it needs at least a 10 µs delay to configure the camera module set by the register programming. The next state is INITIAL Pre-SYNC, where 10 single clock pulses are generated by the MCU and 329 PPs are read and discarded to synchronize the camera module. Then the camera module moves to the main state machine, an infinite while loop. Note that the camera module goes directly to SYNC mode and not to the INTERFACE mode. This is because the camera has already been programmed for the first frame and has been synchronized to start the process of reading the first frame. The SYNC and DELAY states have been put together as they have the same output. Both outputs the training pattern. The SYNC mode consists of two rows (2x328 PPs). DELAY mode takes part in setting the exposure time of the pixels. The default is the minimum value - two rows, as indicated in the figure. The largest value is 496 rows. The main state is READOUT mode, where the image data is read and transmitted to the MCU. This consists of 320x328+8 PPs. After the READOUT state, the state machine goes back to the INTERFACE state, where the camera module is programmed again, before 644 PPs of empty data, just zeros, are transmitted. The empty data makes it easy to distinguish two frames visually.

Table 4.3: The number of SPI packages for each state in both the 8- and 12-bit implementations

Operation Mode	12-bit	8-bit
INTERFACE	648	972
SYNC	656	984
DELAY	656	984
READOUT	104960	157440
END of FRAME	8	12
Total	106928	160392

Table 4.4: SCLK frequency range and the corresponding FPS, specified in the datasheet

SCLK [MHz]			FPS
min	ideal	max	
11.07	12.3	13.53	9
14.13	15.7	17.27	12
22.23	24.7	27.17	19
27.99	31.1	34.21	24
44.19	49.1	54.01	38
56.34	62.6	68.86	49

Table 4.3 shows a summary of the number of SPI packages for each state and the total number of SPI packages. Both the 8- and 12-bit implementations are included. For the 12-bit implementation, the number of SPI packages is equal to the number of PPs. The number of SPI packages for the 8-bit implementation is equivalent to 1.5 times the 12-bit implementation, because it can only transfer $\frac{2}{3}$ of a PP in each SPI package. Table 4.4 list the ideal input SCLK frequencies that can be configured by the registers. The datasheet states that the clock should be within ±10 % of the ideal frequency. This range is shown in the min and max columns. The corresponding FPS are shown to the right. The datasheet also states that 0-58 FPS are achievable but does not clearly state how this can be done. From our understanding, it is most likely done by turning on the IDLE mode in the INTERFACE state and keeping the camera module here to delay the total execution time of a frame, and in this way, the whole range of FPS can be achieved. The camera module only consumes 3.2 mW in IDLE mode, while it consumes 9.7 mW in SEIM mode (IDLE mode off). Table 4.5 shows an example with a 12 MHz SCLK, where it compares the 8- and 12-bit implementations. The number of SPI packages is different, but the number of clock cycles and thus execution times are the same. At 12 MHz, the frame rate is 9.35 FPS.

Table 4.5: Comparison of the 8- and 12-bit implementations at 12 MHz

SPI bit size	# SPI packages	Clock Cycles	Time [ms]	FPS
12-bit	106928	1 283 136	106.928	9.35
8-bit	160392	1 283 136	106.928	9.35

4.2 Development Platform and MCU

For a project like this, with limited time and the fact that it is a prototype, a MCU with the extensive support of libraries and an excellent SDK is the best option. Furthermore, from the literature, the ARM Cortex-M series seemed to be the obvious choice, as it outperforms the competitors in terms of performance per watt (DMIPS/Watt).

In the search for a development platform using the ARM Cortex-M series MPU, STM32 from STMicroelectronics proved to be the best option. They have an extensive assortment of MCUs, optimized for different applications. They also make development boards for almost all MCUs, known as NUCLEO boards. The NUCLEO board simplifies flashing and debugging, as it includes a module that takes care of flashing and debugging. The module is known as ST-LINK/V2. The boards only require a single USB cable from the PC to the board. The SDKs includes a graphical user interface, CubeMX, to setup the MCUs internal functions and peripherals. CubeMX is integrated with the STM32CubeIDE (Integrated Development Environment); it generates all project files, including the generated setup code, and imports all the required libraries. The SDK also comes with tools for simple firmware updates and power analysis. The SDKs includes three levels of libraries STM32Cube Low-Level (LL), STM32Cube Hardware Abstraction Layer (HAL) and Common Microcontroller Software Interface Standard (CMSIS) defined by ARM. HAL is a set of simple functions that just requires inputs and outputs, which provide a fast and simple development process. However, the code will not be the most optimized, and the code size will be substantially bigger than implementations at lower levels. LL gives a high level of optimization, comparable to register level coding, but the development is more advanced and time-consuming than HAL. Previously STM provided a Standard Peripheral Library (SPL) based on CMSIS for low-level coding. SPL is still supported by some STM32 MCUs, but for the ultra-low powered MCUs, its replaced by the LL library. The CMSIS library can still be used, but the development time is more complex and time-consuming than LL. A combination of HAL and LL is also supported. [106, 107].

The ultra-low-power series STM32L have several subcategories, where L0, L1, and L4 are the options best suited for a WCE application [108]. The series provides roughly the same package sizes. All of them have 5x5 mm 32-pin QFNP and 7x7 mm 32-pin LQFP. They also come in WLCSP with 36 pins. The L0 and L1 WLCSP are 2.61x2.88 mm, and the L4 is 2.6x3.1 mm. The main differences between the L0, L1, and L4 are the memory sizes and maximum frequencies. The max flash memory sizes for the L0, L1, and L4 are 192 KB, 512 kB, and 1 MB, respectively.

The max Random-Access Memory (RAM) sizes are 20 kB, 80 kB, and 320 kB, respectively. The L4 has a max frequency of 80 MHz, while L0 and L1 are limited to 32 MHz. As the L4 has more memory and can utilize higher frequencies, one should think that it also consumes more power and is less power efficient than the L0 and L1. However, this is not true. The main reason for this is the release dates for the series. The L1-series is the oldest, released in 2010, the L0-series was released in 2014, and the L4-series is the newest, released in 2018. Thus the L4-series has a newer, more efficient architecture. From the Dhrystone benchmarks, the L1 clearly performs worst in all aspects, followed by the L0. The L4 had the best performance. The overall result reported for the L1, L0, and L4 series is 103, 161, and 217 - higher scores are better. Because of these results, the L1-series was omitted from further assessment [109].

When learning how the camera module worked and figuring out how to implement the interfacing between the camera, MCU, and backscatter switch, a few solutions were considered. The most straightforward solution was to use SPI, as described by the datasheet. However, when testing the ams NanoBerry evaluation board, the oscilloscope showed that the clock was continuous. With the SPI hardware implementation on STM32, this was not possible. The other idea was a solution mimicking the behavior of the NanoBerry evaluation board. This would use an external crystal oscillator to generate a continuous clock and use this to synchronize the MCU and just bit bang the programming sequence at the correct times. The MCU could then be in IDLE mode almost all the time, and the output from the camera would be routed directly to the backscatter switch. These two solutions have very different requirements to the RAM size. The second solution requires almost no memory at all. The first method can work with a smaller RAM size as well.

The SPI hardware implementation on STM32 MCUs are limited by a 16-bit counter. Because of this, the transfer of a frame has to be split into two operations, where each is 52 480 bits. This requires 64 kB RAM. The next RAM size is 40 kB, which would lead to further splitting of the frame readout operations. To limit the number of splits, it is beneficial to use 12-bit SPI. Only the L4 series supports 12-bit SPI, L0 is limited to 8-bit.

Based on these requirements, a MCU with the minimum specifications was chosen from the L0-series, and a MCU from the L4-series with 40 kB RAM, both supported the whole range of package sizes. Additionally a MCU with 64 kB was considered, however, this only comes in a QFNP 32-pin package. The L4 MCUs are STM32L412 (40 kB RAM) and STM32L432 (64 kB RAM) [110, 111]. The L0 MCU is STM32L051 (8 kB RAM) [112]. The differences in power efficiency between the L0 and L4 were verified by simulating STM32L412 and STM32L051 MCUs in the STM32CubeIDE power analysis tool. Table 4.6 gives a comparison of the most important factors, including the individual Dhrystone benchmarks for the two MCUs. The benchmarks are only valid in the temperature range 25 - 105 ℃. From the table, it is clear that the L4 MCU is the best choice for this design and can implement both ideas in the best way possible. The L4-series also

Table 4.6: Comparison of the STM32L412 and STM32L062 MCUs. The operation mode is simulated for the SPI peripheral with the core running at 32 MHz. The memory storage sizes are the largest available. The Dhrystone score is a performance measurement, where a higher score is better. The Dhrystone measure for power efficiency gives an estimate of the current consumption of the core at different frequencies.

Feature	STM32L412	STM32L051
Smallest package size	2.6x3.1 mm	2.6x2.8 mm
IDLE mode	3.11 mA	5.74 mA
Operation mode	4.25 mA	6.25 mA
Supported frequencies [MHz]	16, 24, 32, 48, 64, 72, 80	8, 16, 32
Support 12-bit SPI	Yes	No
Release date	2018	2014
Flash storage	64 kB	64 kB
RAM storage	40 kB	10 kB
Dhrystone score	1.25 DMIPS/MHz	0.95 DMIPS/MHz
Dhrystone power	100 µA/MHz	139 µA/MHz

supports Switched Mode Power Supply (SMPS), which is used to extend the power efficiency in run mode by providing the core with a voltage lower than 1.32 V from an external power DC/DC converter instead of the internal regulator [113].

The custom evaluation board was first designed with the STM32L412 MCU. When we were going to order it, the MCU was not in stock and had a very long lead time. The board was redesigned to an STM32L432 MCU, as this was in stock. However, when the design was finished, and we had verified that we could use the WLCSP or QFNP L4 MCU in the capsule-size prototype, the L4 MCUs small enough for the capsule-size prototype was out of stock. We then considered using an L0 for the capsule and an L4 for the evaluation board. However, when we were ready to order, all of the L4 MCU was out of stock. We then decided to start with a search of what MCU were in stock and order them before using them in the design. As it turned out the only WLCSP MCU in stock was STM32L051 (STM32L051T8Y7DTR). For the custom evaluation board, the only LQFP MCU in stock was STM32L062 (STM32L062K8T6), which closely resembles the STM32L051 but has some extra features that are useless for our design. The 5x5 mm QFNP STM32L412 MCU arrived in the last week of the thesis work, too late to have any use for this project. The use of SMPS, which can reduce the power consumption of the core up to 60 %, was not further investigated as we did not implement the design with an L4-series MCU. The impact on the interfacing with the camera was not clear from the documentation, but it is worth a closer look for future revisions of the capsule design [114].

For the development of code, the development board NUCLEO-L4R5ZI, with a 144-pin L4+ MCU with a maximum frequency of 120 MHz, was used. This is overpowered compared to the implemented MCUs but provides much better flexibility and options during the code development.

4.3 Custom Evaluation Board

The custom evaluation board contains everything that will be part of the capsule prototype. The custom evaluation board will only use components with gull-wing leads to simplify modifications and debugging. This opens for simple hand soldering for most of the components. All resistors and capacitors will use the 0603 package size by the same argument. The PCB is roughly the same size as a credit card, 75x50 mm. This size was chosen to keep it relatively compact while at the same time having enough space to follow the standard PCB design rules in Altium, which ensures that the manufacturing cost is kept at a minimum. A 4-layer design is used to comply with the STM32 recommendations.

The PCB is manufactured and assembled by Eurocircuits. They have a pricing system consisting of 10 different classes with sub-classes. These classes decide the pricing, and higher classes need more expensive manufacturing and inspection equipment. By choosing the minimum requirements within standard pooling, the class is 8C. Standard pooling means that the order is pooled with other customer orders. In PCB manufacturing industrial panels measuring 450x600 mm must be used even if just a single small PCB is ordered. By pooling, the use of the industrial panel is maximized, and multiple customers share the manufacturing costs. This PCB was class 8A, barely staying within standard pooling. The price for just manufacturing the two boards was 130 €. The same design, modified not to be part of standard pooling, increased the price to 912 €.

Figure 4.7 shows the schematic for the custom evaluation board. It contains all the blocks from the system block diagram in Figure 4.1 and some additional parts. The main use of a custom evaluation board is to verify the design and to simplify testing and debugging. All of the MCUs GPIO-pins are routed to PCB headers and PCB headers are added for easy access to important signals and the ground plane. The schematic shows the main blocks: camera, MCU, LEDs, backscatter switch and power management, which includes voltage regulators, batteries and power ports. This section will present them in this order. Figure 4.8 shows 2D and 3D views of the final custom evaluation board, in Altium Designer. The top 2D view is shown in Figure 4.8a and a flipped bottom view is shown in Figure 4.8b. The same views are shown in 3D in Figure 4.8c and 4.8d, respectively.

4.3.1 Stackup, Library, and Design Rules

The stackup for the custom evaluation board is the recommended stackup from STM32. An image of this stackup is shown in Figure 3.3 in Section 3.4.1. This is a 4-layer stackup, with two signal layers at the top (1) and bottom (4) and a ground (2) and power (3) plane inside. We made an Altium stackup file with all the specifications from STM32, which can be imported in new PCB designs in Altium Designer. This file is included in the folder for this project; see Table 4.2.

In order to design a PCB, each component must have a library entry comprised of a schematic, footprint, and 3D model. Altium has a Manufacturer library

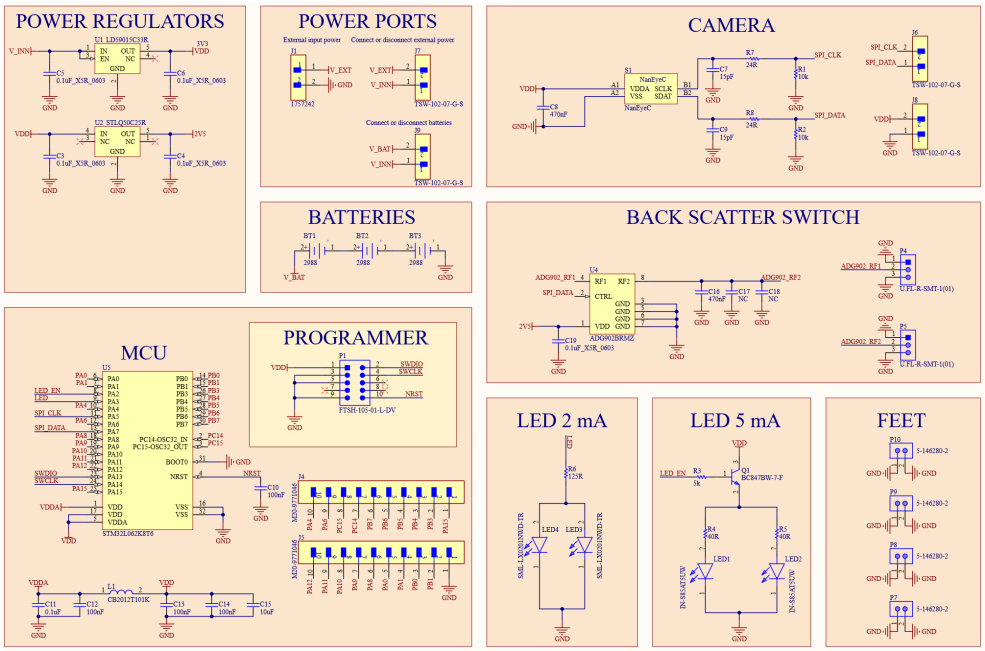
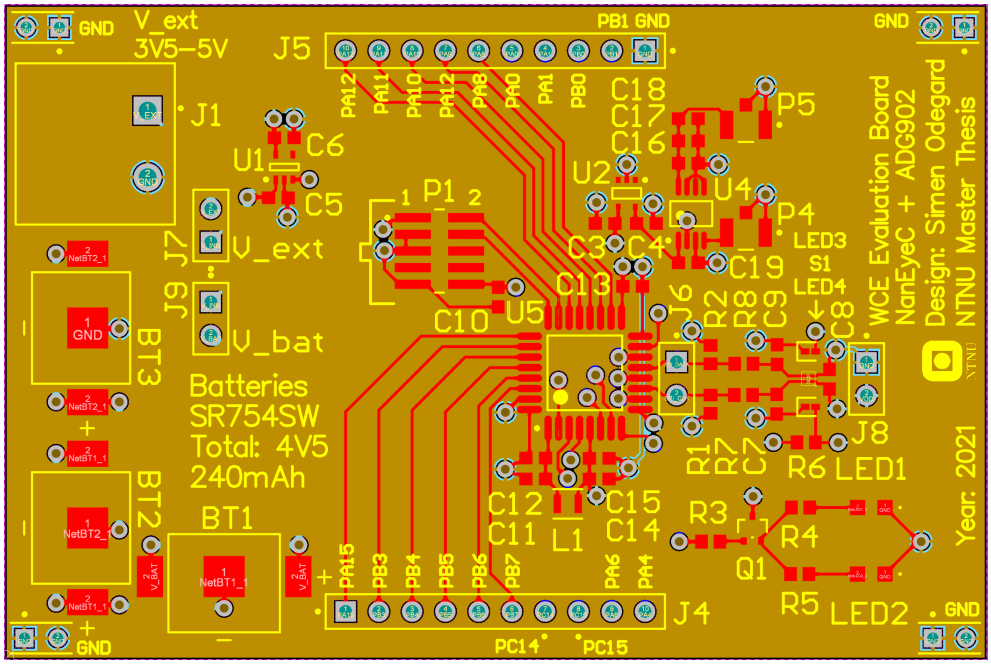


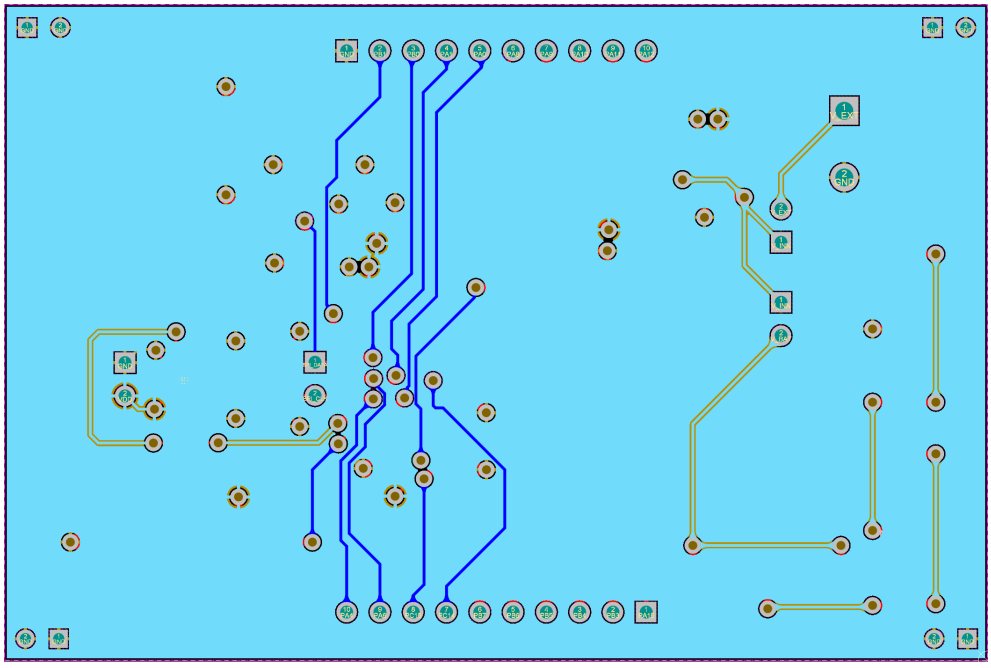
Figure 4.7: Complete Schematic of the custom evaluation board.

called Altium Content Vault, where the manufacturers can submit their library entry. If the entry is approved and verified by Altium, it is free to use for everyone. The drawback of using this library is that each manufacturer uses different guidelines and styles for their library entries. For the special parts like the MCU, NanEyeC camera module, and a few others, we made our own library entries. An integrated library called WCE_thesis was made, which contains the two linked files: WCE_thesis.PcbLib and WCE_thesis.SchLib. The schematic and pinout are defined in WCE_thesis.SchLib. The footprint and 3D model are defined in WCE_thesis.PcbLib. The pinout mapping is defined by adding a link to the footprint in WCE_thesis.SchLib. The 3D models are made using SolidWorks if they are not available online. When a new entry is added, the integrated library must be compiled - checking for errors by a set of predefined rules. The library can be used across multiple projects when it has been successfully compiled. The WCE_thesis integrated library is stored in the folder for this project.

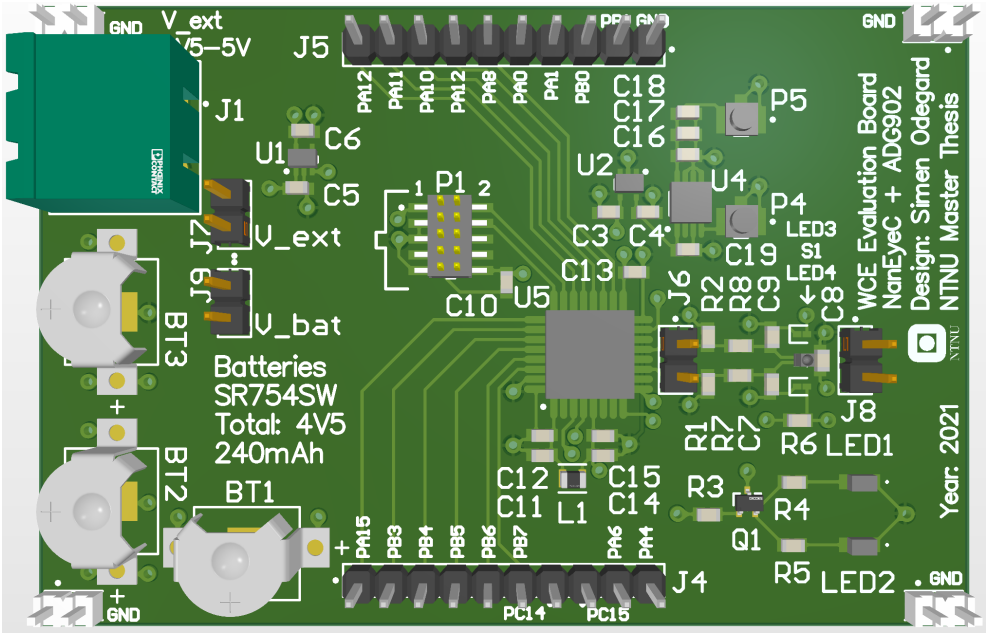
Because this PCB do not have a size limitation and uses "large" components with gull-wing leads, the standard PCB design rules were used. Only the camera module NanEyeC violated these rules. A separate room called NanEyeC was made and placed over the camera module. In this room, a separate set of design rules was defined for the small clearances in the NanEyeC footprint. Without this room, the board was valid for Eurocircuits class 5A. The highest class to still be in the standard pool was 8C. We made sure that the NanEyeC design rules did not violate the rules specified by class 8C.



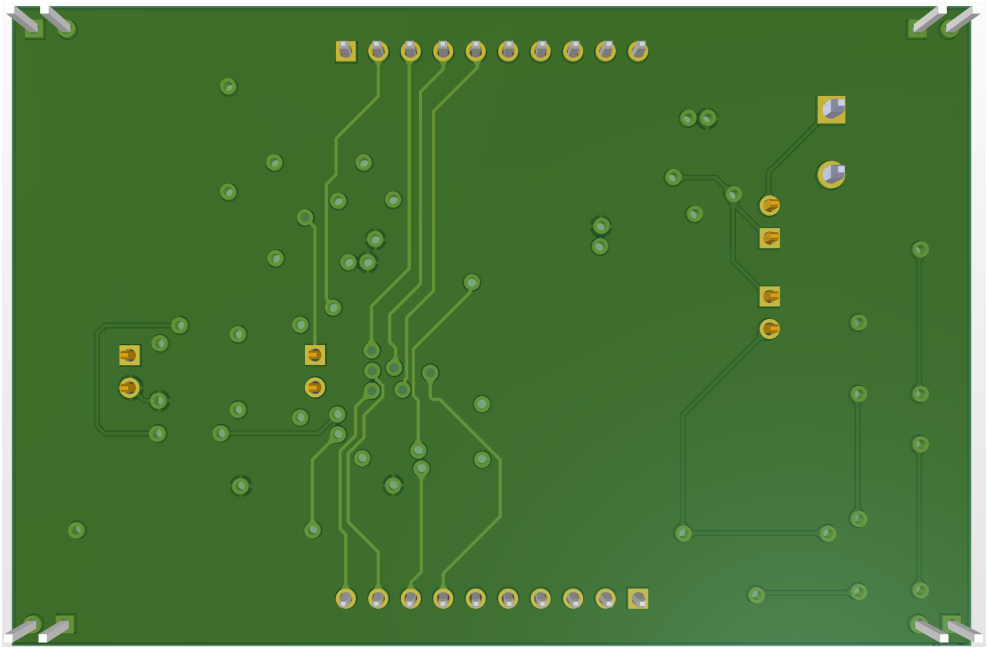
(a) Custom evaluation board 2D top view



(b) Custom evaluation board 2D flipped bottom view



(c) Custom evaluation board 3D top view



(d) Custom evaluation board 3D flipped bottom view

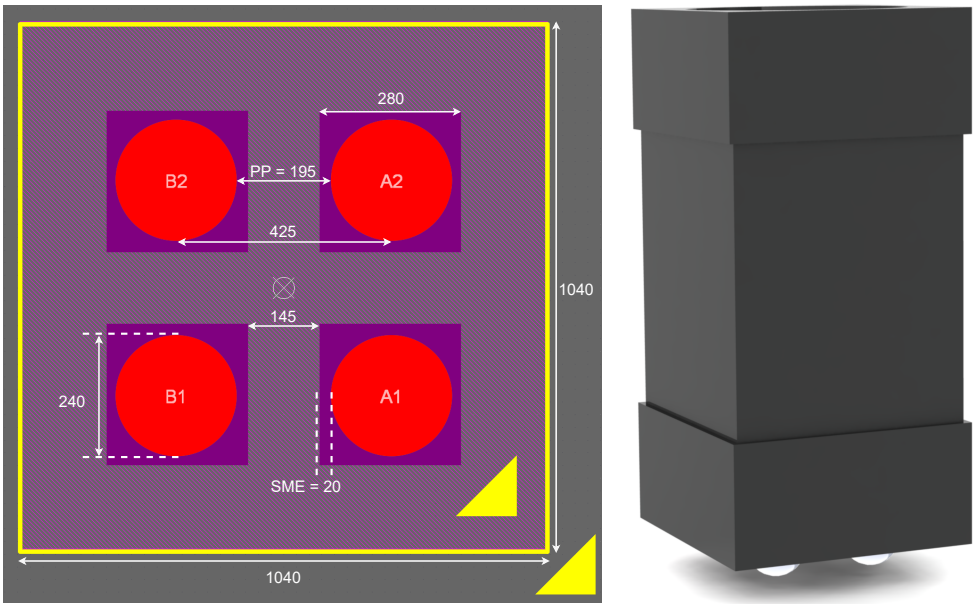
Figure 4.8: Images of the 2D and 3D view of the final custom evaluation board design. The images are screenshots from Altium Designer.

4.3.2 Camera Module - NanEyeC

The NanEyeC did not have any public library entry. The datasheet described the footprint with requirements and measurements. Based on this information, a footprint and schematic were made. The 3D model was designed in SolidWorks based on the mechanical drawing of the camera module provided in the datasheet. Figure 4.9 shows the footprint and 3D model for the library entry called NanEyeC. The measurements placed on top of the footprint in Figure 4.9a are all in the unit µm. There was only one concern to stay within class 8C in the NanEyeC room; the 20 µm SME, which violated this class. We contacted Eurocircuit about this issue, and they said they had a way to omit this, which was proven to work. We did not receive information about exactly what they did. The other measurement that is close to violating this class is the SME to SME clearance, which is limited to 125 µm. For this case, it was 145 µm.

The library schematic is the yellow box called NanEyeC with four pins, shown in Figure 4.10. This figure also shows the entire schematic for the camera module. The two parts TSW-102-07-G-S are 2.54 mm PCB headers used to give easy access to all four ports on the NanEyeC camera module. The NanEyeC datasheet states that the camera module only needs a 470 nF decoupling capacitor between VDD and VSS. However, the reference schematic provided for the camera module on the NanoBerry camera board includes two resistors and one capacitor for each of the data and clock lines. There was no documentation describing the functionality of these resistors and capacitors and why they were used. Because of this, we opted to include the six extra components. The camera module could then be tested both with and without the components. When examining the camera board in the NanoBerry kit under the microscope at a later stage, it was discovered that the components R1, C9, and R2 were not mounted.

When testing the custom evaluation board at the later stages, an expert in hardware and PCB design was consulted about these components. We were told that component configurations like this are commonly used on signal lines to reduce ringing. In this design, the traces are considered short. This means that the cause of ringing is parasitic inductance and capacitance. The ringing effect on the output is caused by sudden changes in the input of the signal line, which makes the parasitic components resonate at their characteristic frequency domain. Even at frequencies in the 0-100 MHz range, a fast edge on a couple of nanoseconds will be generated. Combined with the inductance and capacitance in the traces between the MCU and camera module, this may cause ringing. The resistors will reduce the Q value, and the capacitors will reduce the rise time to counteract the ringing. However, this causes a slower rising and falling edge of the signal and reduces the sampling window to some extent. The rule of thumb is not to implement this if the ringing effect is not observed. The reason why ams have included these on the camera board is most likely because they have observed this effect and have tested and optimized the values for exactly their trace lengths and other impacts from their PCB design. This means that the configuration should



(a) **(b)**

Figure 4.9: Footprint (a) and 3D-model (b) for the NanEyeC library entry. The 3D-model is made in SolidWorks based on the mechanical drawing in the NanEyeC datasheet. All measurements on the footprint in (a) are in the unit µm.

not be implemented before thoroughly testing the camera module without the six components. The reason why ams have removed three of the components is most likely because testing on the designed PCB have shown that they were not necessary and have been removed to avoid the drawbacks of using a configuration like this [115].

4.3.3 MCU

The MCU mounted on the custom evaluation board is STM32L062K8T6 32-LQFP. This MCU was available in Altium Content Vault. Figure 4.11 shows the schematic of the MCU. All the GPIO-pins were traced to the PCB headers, shown at the bottom right. The programmer connector is shown in the top right corner, and in the bottom left corner is the decoupling capacitors and inductor. The pinout is based on the default configurations in CubeMX, with some alterations based on a visual assessment in the PCB editor. In order to improve the EMC performance of the MCU, several considerations were made during routing. The signal lines were kept away from the power lines. The power lines were traced in the power or ground plane while ensuring that they did not cause ground loops in the planes. The planes directly underneath the MCU were kept as whole as possible. Additionally, floating pins were avoided by using internal pull-down for unused GPIO pins, handled in the software.

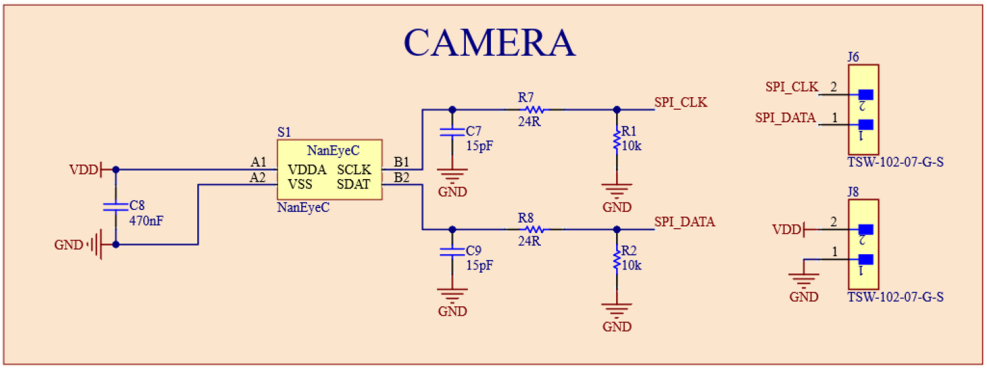


Figure 4.10: Schematic for the camera module NanEyeC

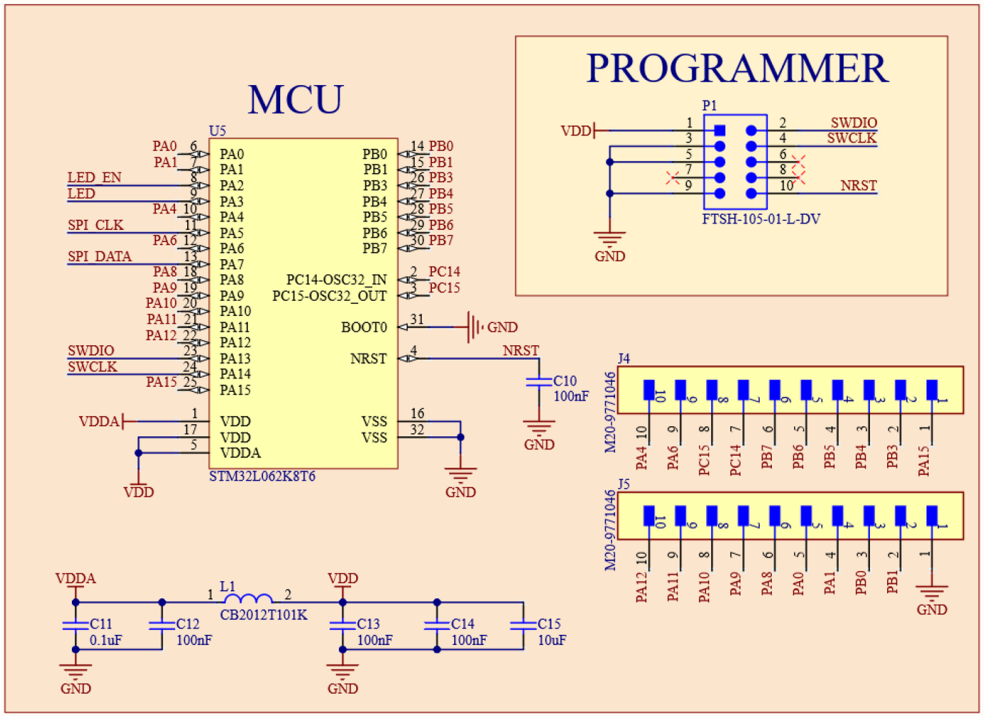


Figure 4.11: Schematic for the MCU, including the decoupling capacitors at the bottom, PCB header pins for each GPIO-pin and the JTAG programming connector.

Decoupling

The decoupling scheme is based on the Application note STM32L0 Hardware Development [116]. The MCU package has two VDD pins (1 and 17), one VDDA pin (5), and two VSS pins (16 and 32). VDD provides power for the internal power regulator and the I/Os. While VDDA provides power to the Phase-Locked Loop (PLL), ADC, comparators, reset blocks and communication subsystems (RCs). The application note states that the VDD pins on the MCU must be decoupled with a single ceramic capacitor for the hole MCU package. The minimum value is 4.7 µF, but commonly 10 µF is used. Additionally, it requires one 100 nF capacitor for each VDD pin. This MCU package has two VDD pins, and for the large capacitor, the typical value is chosen. This makes up the right part of the decoupling scheme shown in the figure. Because this design might require an accurate clock generated by the MCU, STM recommends connecting VDD and VDDA through a ferrite bead or inductor. This does require that the voltage difference VDD-VDDA is less than 300 mV. The application note does not mention anything about the requirements for the inductor. Thus we checked a NUCLEO board under the microscope, which uses a CB2012T101K inductor. This inductor has a ferrite core, with the following specifications: 100 µH, tolerance ±10 %, 130 mA and saturated at 60 mA [117]. In the application note, *Power Supply Filtering* [118], STM explains more about the filtering mechanism of the ferrite bead. It states that the ferrite must not be saturated by the DC current. For this design, 60 mA should be enough to avoid saturation. There was a minimal selection of inductors with ferrite cores at 100 µH, and CB2012T101K was the smallest one in stock. The VDDA pin must also be decoupled with a 1 µF and 100 nF ceramic capacitor. STM32 recommends using XR5 capacitors for decoupling.

For the reset line, NRST, STM recommends a pull-down capacitor to improve EMC and ESD performance and avoid parasitic resets. 100 nF gives the best protection, but for ultra-low power use cases, the security can be traded with power saving by reducing the capacitor size down to 10 nF. Because the WCE is a medical application, this trade-off was not made, and the most secure 100 nF value was chosen.

The decoupling capacitors are placed as close to the MCU as possible. It is also ensured that all VSS and VDD pairs are connected. There are multiple capacitors for each VDD and VSS pair for this implementation - the smallest value capacitor is placed closest to the MCU, as recommenced by STM. Figure 4.8a and 4.8c shows the placements of the decoupling capacitors and the inductor bead.

Programming

To program the MCU we opted to use Serial Wire Debug Interface (SWD), which is an alternative to JTAG, defined in ARM Debug Interface Specification v5. This is a two-wire protocol to access the ARM debugging interface. The protocols' physical layer is comprised of the two lines SWDIO: a bidirectional data line and SWCLK: a clock line driven by the host. Additionally, the protocol supports Serial Wire Out-

put (SWO) for more advanced debugging; this will not be used for this project. The protocol is enabled in CubeMX software. In order to use the protocol, additional hardware is required. An ST-LINK/V2 Programmer device will be used. The output of this device is a 20-pin JTAG connector. To use the SWD protocol for flashing and debugging the ST-LINK/V2 Programmer requires access to the NRST pin and the reference voltage and ground used by the MCU. This adds to a total of 5-pins. In order to avoid having to make a custom 5-pin cable adapter for the programmer, a 10-pin 1.27 mm pitch JTAG connector and cable is used. Realized by the adapter board ARM-JTAG-20-10 from Olimex. This is made for 10-pin JTAG but can be hacked to be used for SWD. The adapter board requires that pins 3, 5, and 9 are grounded. Pin 10 is NRST for both implementations; the same is true for VDD at pin 1. Pin 2 and 4 are programming lines and can be used for SWDIO and SWCLK, respectively. Pin 7, 6 and 8 must remain unconnected [119–121].

The MCU have a set of different boot modes. The STM bootloader has 16 different patterns, deciding the functionality of the bootloader. For the STM32L062-K8T6 MCU used in this design, the only boot pin accessible is BOOT0 at pin 31. The boot modes for this MCU have three options: boot from flash memory, boot from system memory, or boot from the embedded RAM. The bootloader itself is stored in the system memory and can use the internal SPI or UART lines to program either the flash or RAM. The BOOT1 signal is set in the programming software STM32CubeProgrammer or STM32CubeIDE. By default, the BOOT1 pin is set to 1. For this project, the RAM size is limited, and the startup speed is not crucial, so the flash memory is chosen to store the flashed program. To achieve this, BOOT0 must be low, and BOOT1 is *don't care*. As seen in the figure, BOOT0 is connected directly to the ground [103, 122].

Figure 4.12 shows a photograph of the custom evaluation board with the ST-LINK/V2 programmer connected via the ARM-JTAG-20-10 adapter board. It also shows the custom twisted power cable with banana plugs for ease of use.

4.3.4 LED

From the literature review of LEDs and the review of available LEDs in the market, in Section 2.1.1, two LEDs excelled in each class. In terms of uniform lightning and efficiency (luminosity/watt) the 0805 LED IN-S85AT5UW, was the best option [35]. Just in terms of low power consumption the 0201 LED, SML-LX0201NWD-TR, was the obvious choice [32]. A library entry for each LED was made in Altium and is included in the integrated library WCE_thesis. The 3D models were sourced from the manufacturers websites. Two LEDs of each type are included on the custom evaluation board for comparison purposes. Figure 4.13 shows the schematic for each of the LED implementations. The resistor values are calculated using a simple voltage divider. The 0201 LEDs consumes 2 mA each, while 0805 LEDs consumes 5 mA each. The typical forward voltage for the 0201 LED is 2.8 V, which results in

$$R = \frac{3.3\,\mathrm{V} - 2.8\,\mathrm{V}}{4\,\mathrm{mA}} = 125\,\Omega, \tag{4.1}$$

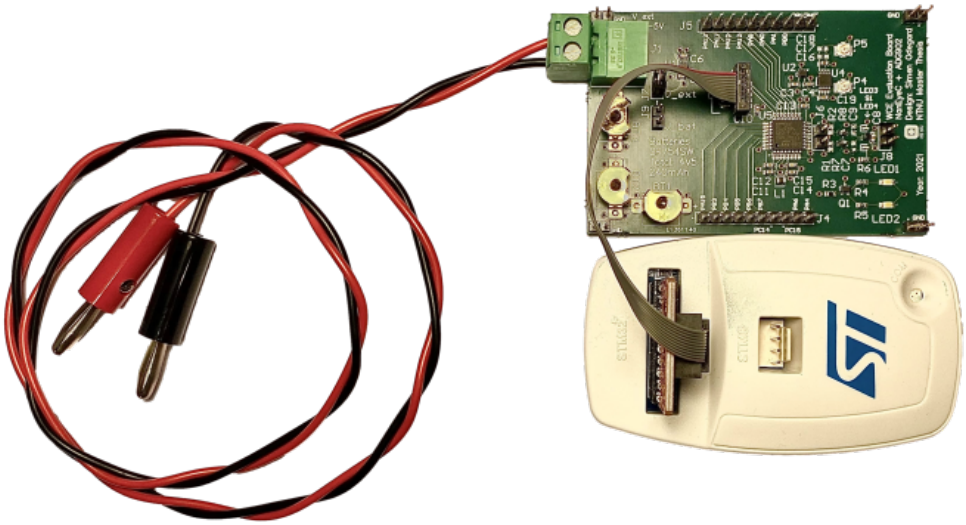


Figure 4.12: Photograph of the custom evaluation board with the ST-Link programming device and a twisted power cable with banana plugs

for the combined LED current. For the 0805 LED the typical forward voltage is 3.1 V, which results in

$$R = \frac{3.3\,\mathrm{V} - 3.1\,\mathrm{V}}{5\,\mathrm{mA}} = 40\,\Omega, \tag{4.2}$$

for each of the LEDs.

The maximum sink current for the GPIO-pins in a STM32L0 MCU is 25 mA [123]. The realistic use case in an actual WCE implementation is at least four LEDs. For the 0201 LEDs this is equivalent to 8 mA current drain, which is well within the limit. Four 0805 LEDs on the other hand, equals 20 mA, which does not leave enough margin to other peripherals like the SPI. Thus the 0805 LEDs has to be driven directly from the power source. To still be able to control the LEDs a transistor is used as a switch, controlled by a GPIO-pin. According to the review of transistors in Section 3.5.3 a BJT is the best option for this use. Based on a search of available BJT transistors, the transistor with the smallest package size and continuous collector current closest to 20 mA was chosen. This was the BC847BW-7-F, in a SOT323 package and rated to 100 mA continuous collector current [98].

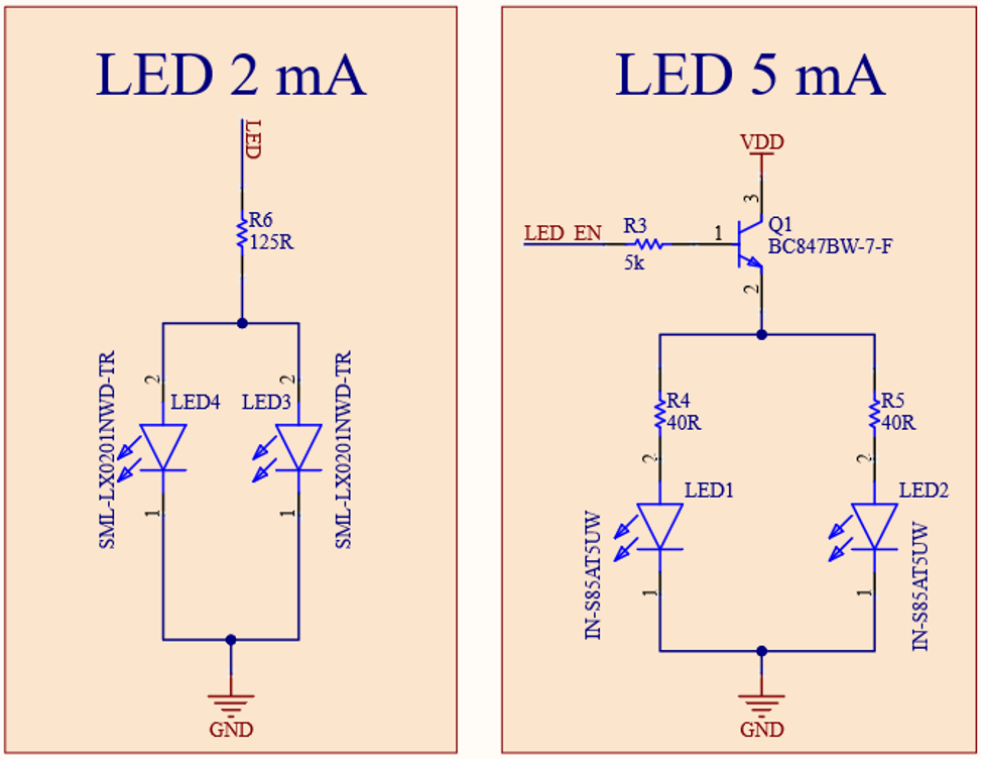


Figure 4.13: Schematic for the two different LED implementations

BJT Transistor calculation

All calculations are only valid in the saturation region. From the graph "Typical Collector-Emitter Saturation Voltage vs. Collector Current" in the datasheet, it states that the gain $H_{FE} = I_C/I_B = 20$ when the transistor is operating in the saturation area. The current consumed by each LED is 5 mA, resulting in the collector current $I_{C(sat)} = 10mA$. Hence the minimum current for the base is

$$I_{B_{min}} = \frac{I_{C(sat)}}{H_{FE}} = \frac{10mA}{20} = 0.5mA \tag{4.3}$$

From the datasheet $V_{CE(sat)} = 0.1V$ and $V_{BE(sat)} = 0.8V$. Kirchhoff's voltage law states that

$$V_{DD} - V_{R_{4/5}} - V_{LED1} - V_{CE(sat)} = 0 \tag{4.4}$$

Solving for V_{R_4}

$$V_{R_{4/5}} = 3.3V - 3V - 0.1V = 0 \Rightarrow V_{R_{4/5}} = 0.2V \tag{4.5}$$

Ohms law gives the resistor value for R_4 and R_5

$$R_4 = R_5 = \frac{V_{R_{4/5}}}{I_{LED}} = \frac{0.2V}{5mA} = 40\Omega \tag{4.6}$$

The resistor value for R_3 is also found using Kirchhoff's voltage law

$$v_{GPIO} - V_{R_3} - V_{BE(sat)} = 0 \tag{4.7}$$

$$V_{R_3} = 3.3V - 0.8V = 0 \Rightarrow V_{R_3} = 2.5V \tag{4.8}$$

$$R_3 = \frac{V_{R_3}}{I_{B_{min}}} = \frac{2.5V}{0.5\,\text{mA}} = 5\,\text{k}\Omega \tag{4.9}$$

BJT Power Dissipation

The power dissipation is given by

$$P_D = V_{CE} \cdot I_C + V_{BE} \cdot I_B \tag{4.10}$$

To drive the two 5 mA LEDs, the dissipated power is

$$P_D = 0.1\,\text{V} \cdot 10\,\text{mA} + 0.8\,\text{V} \cdot 0.5\,\text{mA} = 1.4\,\text{mW} \tag{4.11}$$

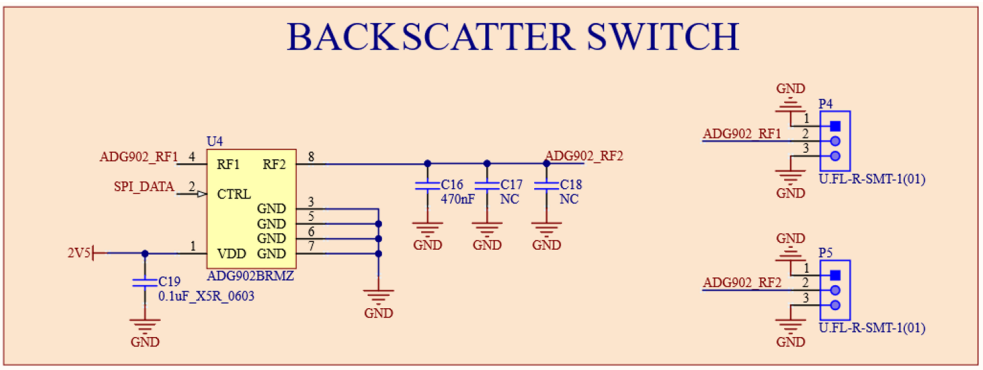


Figure 4.14: Schematic for the backscatter switch

4.3.5 Backscatter Switch

The backscatter developer team had tested two switches: ADG901 and ADG902 [124]. The ADG902 has the best performance but a larger footprint than ADG901. The ADG901 package measures 3x3 mm, while the ADG902 is 5.15x3.20 mm. The large size was a concern for the space in the capsule-size prototype. However, due to the better performance of the ADG902 switch, this was chosen for the custom evaluation board. Figure 4.14 shows the schematic for the backscatter switch. The input power is 2.5 V, and a 0.1 µF X5R decoupling capacitor is used, as specified in the datasheet. The input to the switch is the CTRL-port, which receives data directly from the camera module. The two outputs, RF1 and RF2, are connected to a copper film at each end of the capsule. However, a U.FL connector from Hirose is used for this evaluation board, a tiny male coaxial connector. On RF2, the backscatter developer team requested a 470 nF capacitor and two empty 0603 capacitor footprints that could be used for tuning during testing.

4.3.6 Power Management

Figure 4.15 shows the schematic for the power management, consisting of two power regulators, power ports, and batteries. The batteries in the schematic are battery holders. The power ports consist of a power-in plug, with an adapter plug for easy plugging and unplugging of the external power source. The two others are PCB headers, which cut the positive supply line from the batteries and the external power source. By shorting the two pins with a shunt, for example, SNT-100-BK-G, the user can change which power source to use. From the review of voltage regulators in Section 3.5.4 the obvious choice is a LDO.

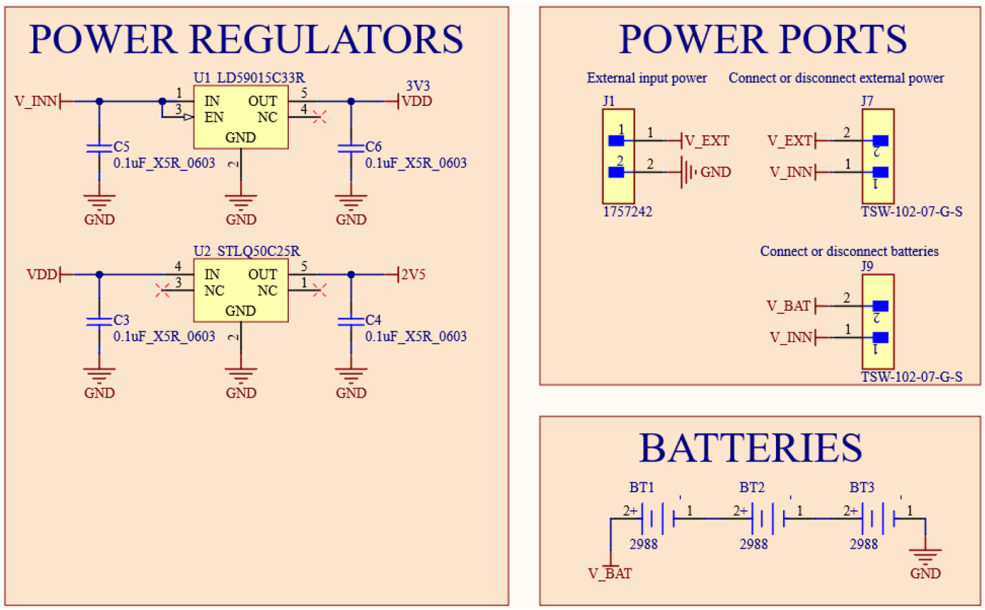


Figure 4.15: Schematic for the power management, including voltage regulators, batteries, and power ports

4.3.7 Batteries

According to the literature review, the common choice of batteries in WCEs are two 3 V batteries at 55 mAh. This design's required maximum VDD voltage is 3.3 V, making this battery configuration a poor option for power efficiency optimization. Because of considerable power dissipation from the LDO, due to the significant dropout voltage: 6.0 V - 3.3 V = 2.7 V. When the batteries used in this design were chosen, the capsule design was considered; see the argumentation in Section 4.5.1. It was decided to use three Renata SR754SW (SR48) batteries rated at 80 mAh capacity, and 1.55 V output voltage [125]. This results in a total voltage of 4.65 and 240 mAh capacity. The battery is a Zinc / Monovalent Silver Oxide, low drain battery with a 0 % mercury guarantee. The batteries weigh 1.09 g each and can operate in the -10 - 60 ℃temperature range. The datasheet provides a graph that discloses the voltage drop over the battery lifetime. During the first 92 %, the battery is relatively stable at 1.55 V; for the subsequent 7 %, the battery voltage decreases linearly from 1.32 V before it rapidly drops to 1.2 V for the last 1 %.

4.3.8 Voltage Regulator

Two LDOs are needed. One 3V3 LDO for the main VDD, used by the MCU, LEDs and NanEyeC camera module. The second LDO is 2V5 LDO to supply the backscatter switch. The input voltage from the three batteries is 4.65 V, which gives a dropout voltage of 1.35 V for the main 3V3 LDO. While the 2V5 LDO have VDD as the input voltage, which gives a dropout voltage of 1.3 V. The current going

through 2V5 LDO is only 0.1 µA. The current going through the 3V3 LDO is the total current used by the custom evaluation board, which comprises the MCU, camera module, LEDs and the 2V5 LDO. According to Table 4.6 the MCU consumes 6.25 mA. From the datasheet, the camera module consumes approximately 3 mA. With all four LEDs at full brightness, they consume 14 mA. This means that the output current rating of the 3V3 LDO must be at least 23.25 mA. To be safe, the lower limit was set to 100 mA. There was very few LDOs available in the market at this time. The criteria for the search were very low quiescent current, up to 5 V input voltage, at least 100 mA current rating, and the smallest package size available. STM has a series of LDOs fitting these criteria, which is designed for ultra-low power battery applications. The 3V3 LDO, LD59015C33R, in this series was chosen [126]. The package type was SOT323-5L with gull-wing leads, and the ratings are 3.5-5 V input voltage and 150 mA current. The low dropout voltage ensures that it is operations also during the last 8 % of the battery lifetime. The 2V5 LDO in this series was out of stock. For the 2V5 LDO the options were very slim, and one of the only ones in stock was STLQ50C25R, rated to 50 mA current in the same package types as the 3V3 LDO [127]. This LDO has poorer specs than the other series, but the effect of this LDO is negligible due to the small output current. Both LDOs have the same decoupling specifications; one 0.1 µF X5R capacitor on the input and output. The 3V3 LDO have an enable port. This is set to always-on by connecting it to the input voltage.

LDO Power Dissipation

The power dissipation for the 3V3 LDO is given by Equation 3.5:

$$P_D = (4.65 - 3.3)\text{V} \cdot 20\,\text{mA} \approx 31.4\,\text{mW}, \tag{4.12}$$

which results in a temperature increase of

$$T = 31.4\,\text{mW} \cdot 645.69\,^\circ\text{C/W} \approx 20.3\,^\circ\text{C} \tag{4.13}$$

For the 2V5 LDO the power dissipation is

$$P_D = (3.3 - 2.5)\text{V} \cdot 0.1\,\mu\text{A} \approx 80\,\text{nW}, \tag{4.14}$$

which results in a temperature increase of

$$T = 80\,\text{nW} \cdot 331.4\,^\circ\text{C/W} \approx 0\,^\circ\text{C} \tag{4.15}$$

The ambient temperature during operation of the custom evaluation board is around 22°C, resulting in approximately 40°C for the 3V3 LDO.

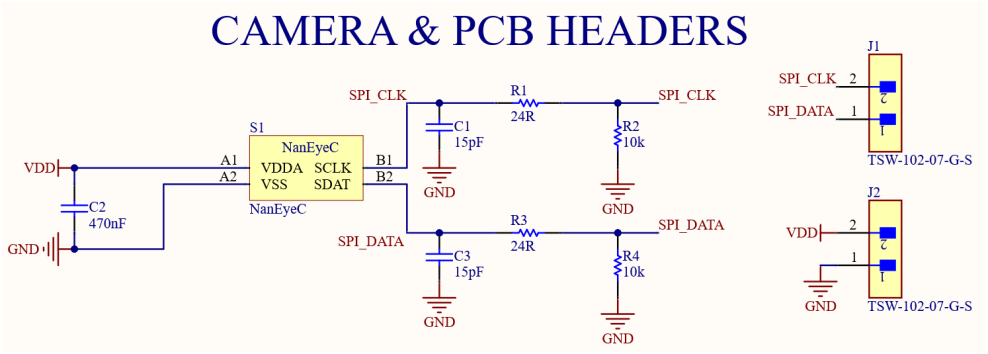


Figure 4.16: Schematics for the NanEyeC breakout board

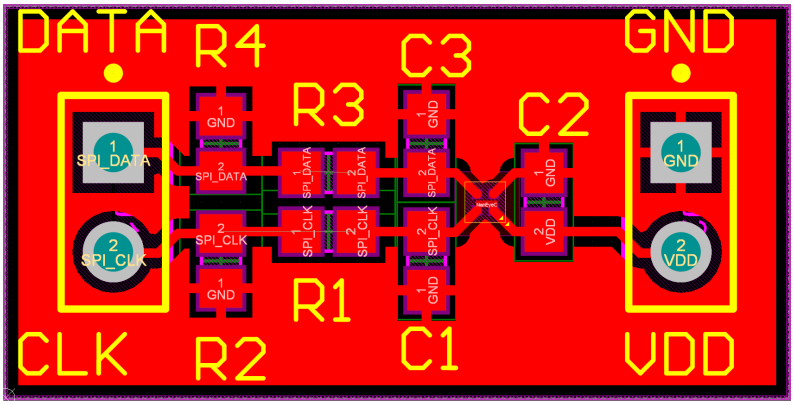
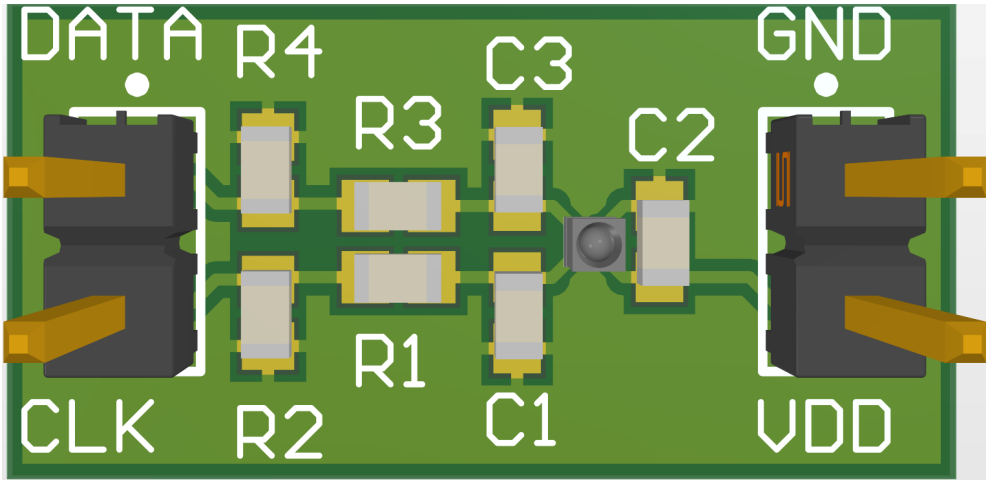


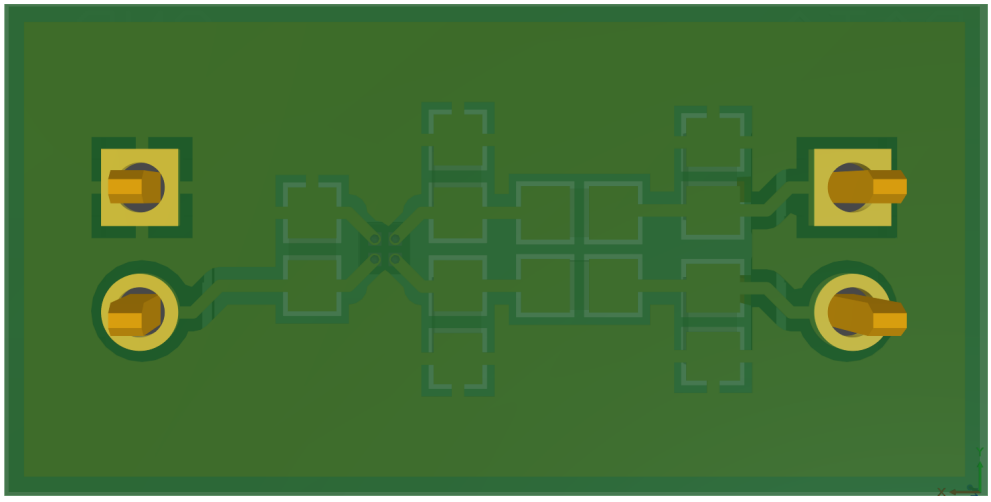
Figure 4.17: NanEyeC breakout board - 2D top view

4.4 NanEyeC Breakout Board

The NanEyeC breakout board is a simple one-layer PCB. This PCB was manufactured internally at NTNU. The manufacturing equipment at NTNU is not very advanced compared to PCB manufacturers like Eurocircuits. Because of this, a different set of PCB design rules was applied, specified by NTNU. Their technique only allowed for a single layer PCB with a copper plane, where they cut out traces and pads. The minimum requirement for the track width and all clearances was 0.254 mm. The schematic is shown in Figure 4.16, which is equivalent to the schematic for the camera on the custom evaluation board. The placement of the PCB headers exactly match the PCB headers for the camera module on the custom evaluation board. Figure 4.17 shows a 2D view from the top. Figure 4.18a and 4.18b shows 3D views from the top and flipped bottom, respectively. By soldering 2.54 mm female receptacles on the bottom of this board, it can be used as an extension board on the evaluation board. This requires removing the camera module and components from the custom evaluation board. Figure 4.19 shows this configuration, with the camera breakout board mounted on the custom evaluation board as an extension module.



(a) NanEyeC breakout board - 3D top view



(b) NanEyeC breakout board - 3D flipped bottom view

Figure 4.18: Images of the 3D view for the final NanEyeC breakout board design. The images are screenshots from Altium Designer.

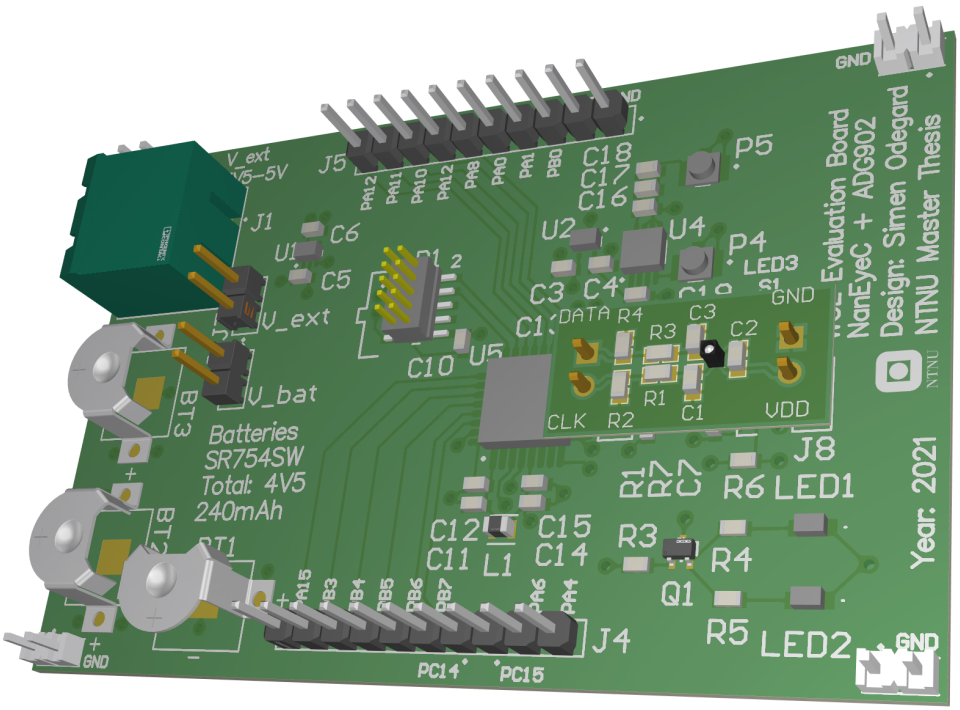


Figure 4.19: The NanEyeC breakout board mounted on top of the custom evaluation board as an extension module. Before this configuration can be used, the components between the two PCB headers on the custom evaluation board must be removed.

4.5 Capsule-Size Prototype

An expanded view of the final capsule design seen from both the top and bottom is shown in Figure 4.20. The assembled capsule prototype with batteries can be seen in Figure 4.36g in Section 4.5.6. The capsule prototype is comprised of three PCBs:

- PCB1: Camera and Microcontroller Unit
- PCB2: Backscatter Switch and Power Regulation
- PCB3: Battery Holder and Power Switch

Many different solutions were explored in the start phase of the capsule-size prototype design. A brief introduction to the design process of the final capsule prototype is given, with some general design choices, valid for all of the PCBs. Some of the solutions that were not implemented in the final design, which might have some value for future revisions, are briefly explained in Section 4.5.1. Subsequently, each of the PCBs is presented, followed by the mechanical and electrical assembly.

PCB1 was by far the most complicated PCB. Countless revisions have been made to fit everything and break as few EMC guidelines as possible. However, some had to be broken to limit the cost of the prototype. The STM32L051T8Y7DTR MCU, in a WLCSP-36 configuration with only 0.4 mm pitch, posed the greatest challenge. The recommended design principles for a 6x6-pin BGA requires a 6-layer PCB, and the use of blind, buried, and μVias. It was not possible to fan out the MCU without in-pad μVias, but the other types were omitted, and a 4-layer PCB was used. The consequence was that the ground and power planes had to be used to trace signals underneath the MCU and in some other places. This violates the EMC guidelines. This was discussed with PCB manufacturers, and they stated that for a prototype like this, it is okay and most likely will not cause any problems, but for an actual product, this would not have been approved. Adding an extra industrial panel would increase the cost of the order by 500 €. Since μVias had to be used, Eurocircuit did not have equipment advanced enough for this manufacturing process, which requires HDI manufacturing, see Section 3.4.3. Several PCB manufacturers was contacted and considered. The two finalists were AT&T and ExceptionPCB.

The design was discussed with both parties, and the PCB files were distributed for analysis. Both of them stated that use of 0.4 mm pitch BGA MCU is unnecessarily complicated for a prototype. It is hard to debug and increases the price significantly. However, this was the only option available on the market at the time. Both recommended to change the design to a flex-rigid design, using flexprint to interconnect the PCBs, instead of PCB headers. AT&T even proposed a design they had used in a similar project. Implementing a flexprint solution would also require a more advanced mechanical assembly, where the ideal case would be molding. This was not pursued further due to the limited time frame. The contact person at AT&T, Hofmann Volker, was the most experienced in PCB manufacturing for med-

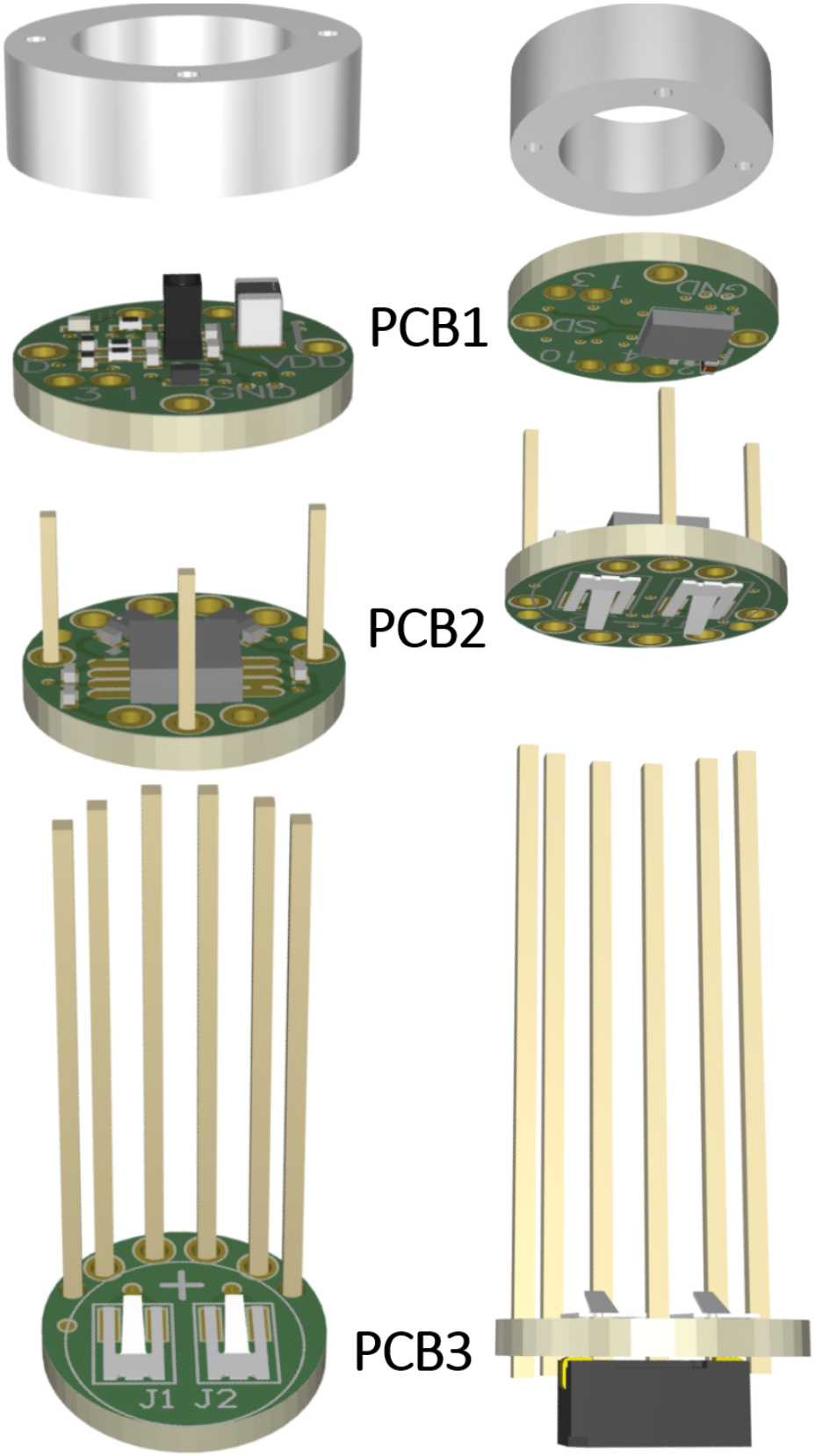


Figure 4.20: Expanded view of the final capsule design

ical use and even had experience with similar projects. However, ExceptionPCB was the only manufacturer that could promise delivery before the last applicable ordering date. After several discussions and revisions with the engineers at ExceptionPCB, the files were cleared for manufacturing. They just waited for us to verify our design before manufacturing the order. Unfortunately, there was not enough time to order it.

PCB2 was the PCB that imposed the limitation on the smallest size achievable. This was caused by the large backscatter switch ADG902, combined with the mechanical and electrical assembly scheme. This could have been significantly optimized by utilizing blind vias for the interconnecting PCB headers. However, it was not implemented to reduce manufacturing costs. PCB3 is the simplest PCB, with only a switch to turn on and off the battery supply. Additionally, a programming PCB was designed to enable simple and secure programming of the MCU. This PCB was supposed to be manufactured internally at NTNU. The design was approved, but it was not manufactured, as the capsule prototype was not ordered.

4.5.1 Initial Design Consideration

Several considerations had to be made when designing the capsule. Multiple designs have been made and checked in 3D. A few of the design choices that are not part of the final design are explained briefly here, as it may be of interest to implement in a future revision of the capsule. The main limiting factors for the design choices at this stage were the mechanical limitations: diameter, length, and how to assemble the PCBs both mechanically and electrically. This is a prototype, so the exact sizes are not crucial, but ideally, we wanted to stay within the typical WCE capsule size: 11 mm diameter and 26 mm length. We assumed an outer shell with a thickness of 1 mm; this leaves a diameter of 9 mm. However, the antennas and cables connecting the antennas to the PCB must also be considered. The antenna was measured to 0.5 mm and the cables 0.75 mm. The cables were not considered at this point. In order to minimize the impact of the cables, the idea was to make a notch in the PCB or use leftover space in the mechanical assembly. The antenna would add 1 mm total in diameter, so the PCB diameter goal was initially set to 8 mm.

Batteries and PCB Diameter

A critical choice for the design was the size of the batteries. The available battery sizes were limited. The best-suited batteries were 7.9 mm in diameter. In order to get the correct voltage and the largest battery capacity possible, it was decided to use three batteries. To visualize the capsule design and the impacts of the batteries, PCBs and components, the Computer-Aided Design and Drafting (CAD) software SolidWorks was used to do a visual analysis. A model consisting of three batteries was made, and simple box 3D models with the correct sizes of the main components. For the PCBs, discs with various diameters were made, all with thickness 1.55 mm, as this was the thickness of the recommended stackup

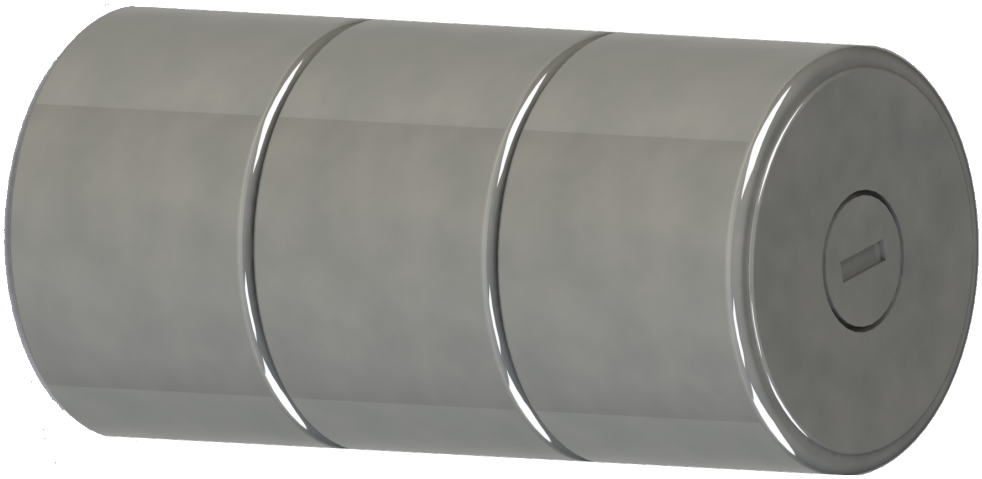


Figure 4.21: 3D-model of the battery pack, made in SolidWorks. The battery pack consist of three batteries. The height is 16.2 mm and the diameter is 7.9 mm.

from STM32. Figure 4.21 shows an image of the battery pack 3D-model. After trying several placements, it was clear that three PCBs was the maximum number of PCBs to stay within the length requirement. The length of the battery pack is 16.2 mm and added with three PCBs the total length is 20.85 mm. Additionally, the height of the MCU is 0.59 mm, and the height of the ADG902 switch is 1.1 mm, resulting in a minimum clearance of 1.69 mm. This totals to 22.54 mm, and with the NanEyeC camera module, the total is 24.84 mm. It was also clear that 8 mm diameter for the PCBs was not possible to achieve for a prototype, as the batteries were 7.9 mm in diameter. The diameter was thus increased to 8.7 mm. This was used quite far in the design, but it did become clear that it was not possible to make a mechanical assembly without making custom parts or using advanced mechanical techniques, which is not the focus of this work. Therefore, the final diameter ended at 9.5, where the mechanical assembly is possible with only a 0.01 mm margin.

PCB Layout

The next consideration was the electrical connection between the PCBs and ideal placements of the components. The camera and LEDs were obvious. For the placement of the MCU, the NanEyeC SEIM protocol dictates that the distance between the camera and MCU should be as short as possible. Because of this, it was decided to place the MCU on the bottom of the camera PCB. The next components that had to be placed were the backscatter switch and the two LDOs. The initial idea was to place the backscatter switch next to the MCU. However, there was not enough space for the mechanical assembly with this configuration. The backscatter switch was then placed on the second PCB. The LDOs was natural to place as close to the batteries as possible and was placed on the same PCB

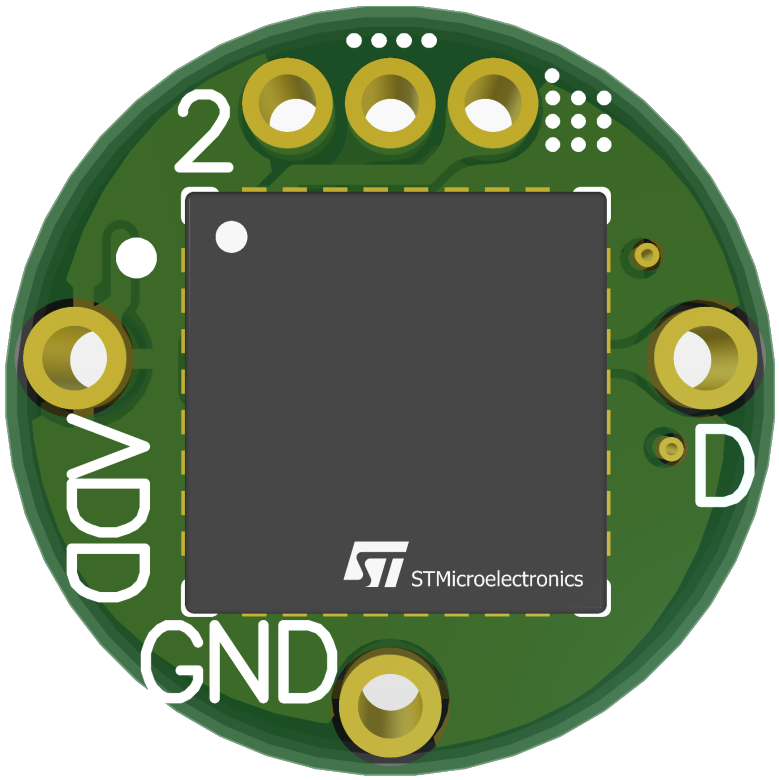


Figure 4.22: The STM32L432KBU6 MCU, we initially wanted to use, placed on the bottom of PCB2. The vias for VDD and GND, used for programming had to be removed to fit the MCU.

as the backscatter switch. Below the batteries, it had to be a PCB for mechanical support and an electrical connection. The final capsule-size prototype uses the STM32L051T8Y7DTR MCU with a footprint of only 2.61x2.88 mm. However, initially, we wanted to use an L4 MCU and avoid using a WLCSP. The L4 MCU we wanted to use was STM32L432KBU6. It has a footprint of 5x5 mm, with a diameter of approximately 7.1 mm. Figure 4.22 shows the STM32L432KBU6 MCU placed on the bottom of PCB2. The GND and VDD vias, used for programming, had to be removed to fit the MCU.

Programming and Electrical / Mechanical Assembly

The next challenge was to find a way to electrically and mechanically connect PCB1 and PCB2 and how to program the MCU. One of the most promising design ideas that were designed and analyzed was a connector pair with mechanical support between the two PCBs. Figure 4.23 shows this design. The idea was that the male connector mounted on PCB1, with the MCU, would serve three purposes: mechanical and electrical connection to PCB2, and to program the MCU. In order to program the MCU a programming extension PCB was designed, which has

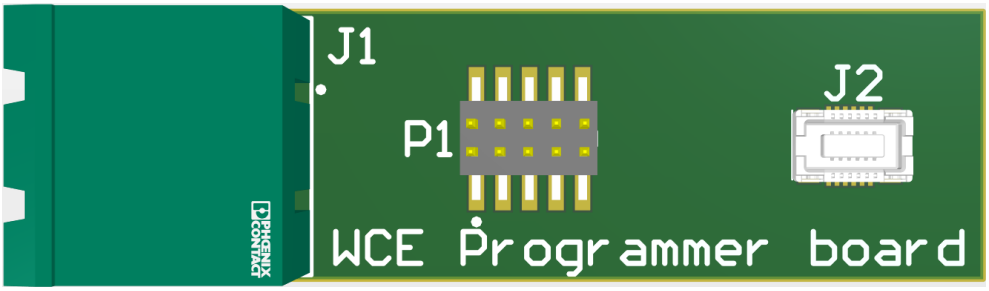
power-in, the 10-pin JTAG connector, and the female connector, which pairs with the male connector on PCB1. As seen in Figure 4.23c, the female connector is quite large and requires almost half of the PCB space. Because of this, the space left for the mechanical assembly of PCB2 and PCB3 is minimal. The four holes: two to the left and one above and below, are made for 1.27 mm PCB headers. A mock-up of this design was made, and it was clear that the four PCB headers were too weak. The pressure of the batteries would bend them. Molex manufactures the connectors; the part numbers are 513389974 and 5017459901 for the female and male connectors, respectively. We have made library entries for both in the WCE_thesis library. The connector pair used is 12-pin and is too large for this design. In the same series, Molex has 6- and 10-pin connector pairs, which could have been used in this design [128]. However, these connector pairs had 54 weeks lead time and were thus not considered.

Power Switch

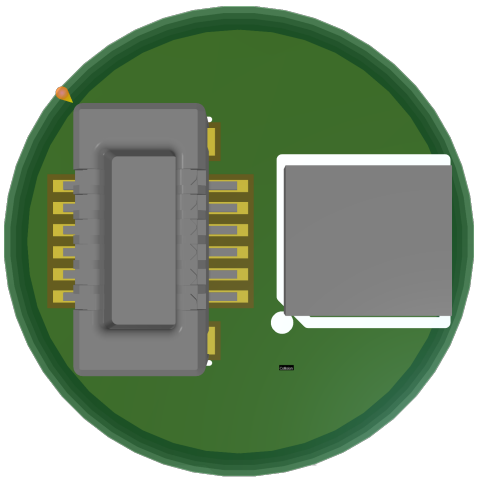
The next consideration was how to easily turn on and off the capsule during testing without removing the batteries. The first idea was to use a light sensor that controlled the 3.3 V LDO connected to the batteries. This had some drawbacks. It would require an extra signal from the MCU on PCB1 to the enable gate of the LDO on PCB2. This requires an additional PCB header between the two PCBs, and the space is already very limited. The other drawback was extra code and hardware that had to be implemented, and the time for the project was already very limited. Because of this, a simple solution was chosen, a switch at the bottom of the capsule. This makes the capsule a bit longer than necessary, but the most critical factor is simplifying the prototype's testing conditions.

LEDs

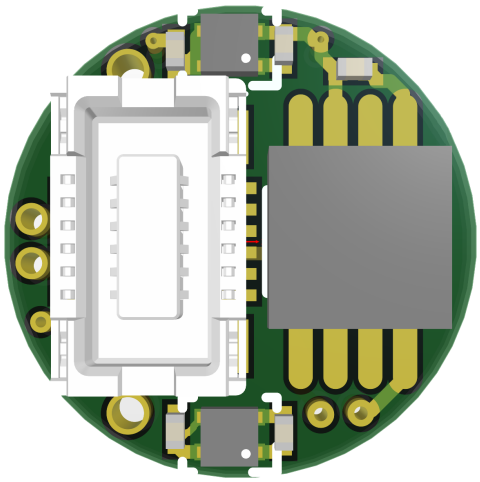
For the LEDs it was decided to only use two 0201 for the prototype, to keep the power consumed by the LED to a minimum. This would enable more testing on a set of batteries. During the design phase some ideas was considered related to the total uniformity of the LED lights. From the LED review in Section 2.1.1 we know that the 0805 LEDs was best for the WCE application in terms of uniform lightning. However, it was considered using eight 0201 LEDs spaced out, and see weather this increased number of LEDs, and the spacing between them would make a more uniform light source than four 0805 LEDs. Another consideration was to not get a shadow from the NanEyeC camera module. The 0201 LED have a 110° radiation angle. Figure 4.25 shows how to calculate the required distance between the camera module and the LEDs to avoid this. The length is found by trigonometric manipulation: $X = 2.1\,\text{mm} \cdot tan(55°) \approx 3.0\text{mm}$. The 0805 LED have a 120° radiation angle - the required distance for the 0805 LEDs is $X = 1.2\,\text{mm} \cdot tan(60°) \approx 2.0\text{mm}$. Figure 4.25 shows a comparison between eight 0201 and four 0805 LEDs. The illustrations are to scale, and the PCBs have diameter of 9.5 mm.



(a) PCB1



(b) PCB1



(c) PCB2

Figure 4.23: The design idea using connectors, serving the purpose of mechanical and electrical connection between PCB1 and PCB2 and at the same time can be used to program the MCU. The programmer board (a) has the same female connector as PCB2 (c) which both fits the male connector on PCB1 (b). The four holes next to the connector in (c) shows the initial idea for the mechanical assembly of PCB2 and PCB3. In (c) the box to right of the female connector is the backscatter switch and the two boxes above and below are LDOs. The height of the switch is 1.1 mm and the MCU is 0.59 mm. This requires a minimum clearance of 1.69 mm. The connectors in the 3D-models have a combined height of 2 mm.

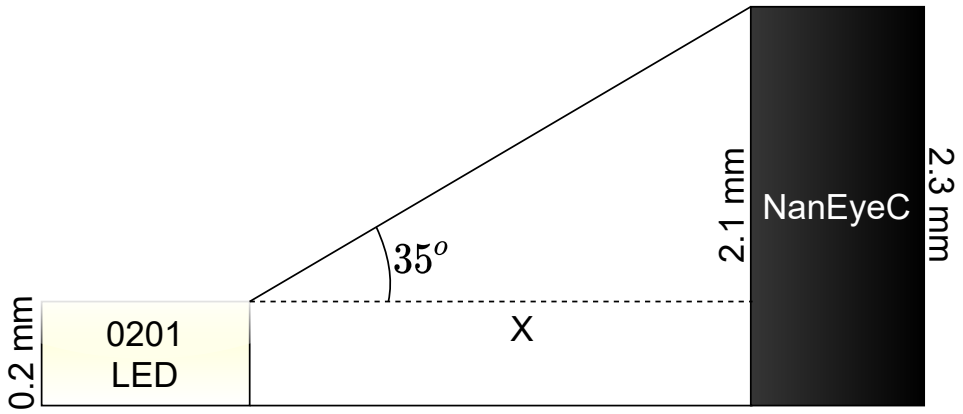
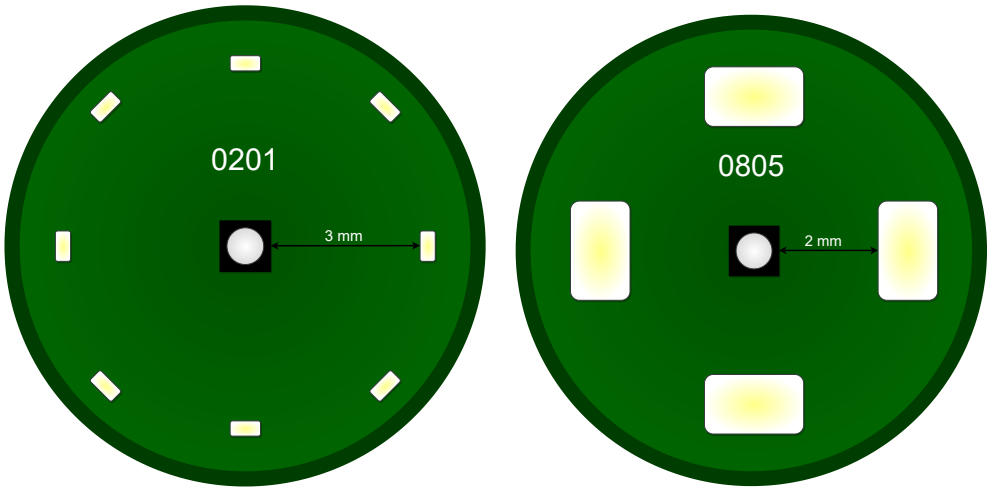


Figure 4.24: Illustration of how the minimum distance between the LEDs and the NanEyeC camera module are calculated to avoid a shadow from the camera module. This shows an example for 0201 LEDs, where X = 3 mm.



(a) **(b)**

Figure 4.25: Comparison of two LED configurations. (a) shows the placement of eight 0201 LEDs, 3 mm from the camera module. (b) shows the placement of four 0805 LEDs, 2 mm from the camera module. The illustrations are to scale, and the diameter of the PCBs are 9.5 mm.

Table 4.7: Summary of the most important PCB design rules, implemented in accordance with the ExceptionPCB Manufacturing Capabilities guidelines. The rules are used for all three PCBs.

PCB Design Rule	Min [mm]	Preferred [mm]	Max [mm]
Routing Width - ALL	0.11	0.254	0.254
Via Diameter	0.225	0.3	1.5
Via Hole Size	0.125	0.15	1
Solder Mask Expansion	0.05	0.102	0.102

4.5.2 General Design Guidelines and Design Rules

Some design guidelines are valid for all of the PCBs. Because all of the PCB are manufactured on the same industrial panel, the same PCB design rules can be used without increasing cost, except for μVias. The design rules for all three PCBs are set in accordance with the ExceptionPCB Manufacturing Capabilities guidelines [129]. From the discussions with the ExceptionPCB engineers, it was recommended to use a 10 % tolerance in the design, where possible. This tolerance is added in the PCB design rules and waived by the use of rooms, where it is necessary. This was only used in PCB1, close to the MCU. The clearance was set to 0.11 mm for all clearances, and the minimum 0.10 mm was used in the MCU room. This is also true for all clearances to the cutout line. The other important rules that were modified in accordance with the guidelines are summarized in Table 4.7. The minimum values are the absolute minimum, and the preferred values include at least 10 % tolerance. All PCBs utilize teardrops for the transitions between pads and traces, see Section 3.4.2.

Resistors and Capacitors

All resistors and capacitors used have the 0201 size, except the 10 µF capacitor for the MCU. The smallest package size for 10 µF capacitors is 0402. It was considered using components smaller than 0201, but as it is a prototype that might require modifications, smaller components than 0201 would not be possible to hand solder.

For the resistors, the PCB manufacturers recommended the medical-grade resistor series from Vishay and Yaego [130, 131]. These series features thin-film resistors with low tolerances that are 100 % thermal shock, electrical, and visually inspected.

The decoupling capacitors were specified to use X5R and X7R in class 2, rated at 0.1-10 % and 10V or higher. This was the case for the rest of the capacitors as well, except the backscatter capacitor. This capacitor was specified to be class 1, rated at 0.1-5 % and 10V or higher.

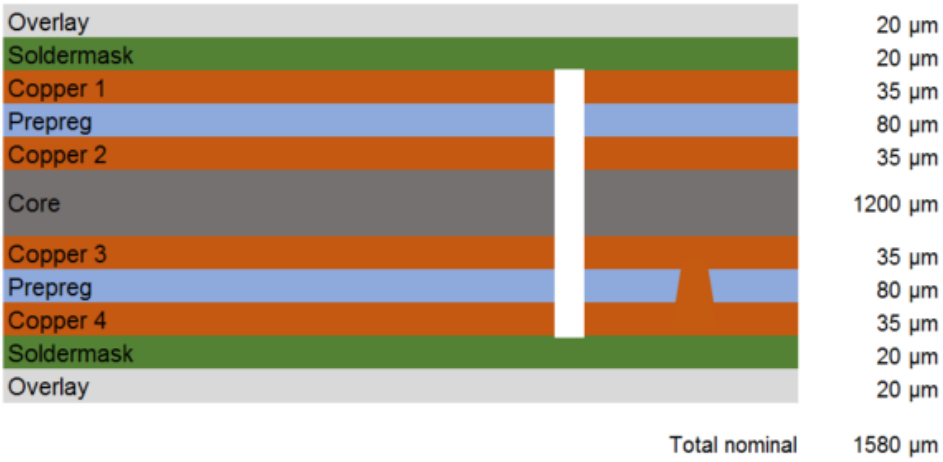


Figure 4.26: Stackup recommended by the PCB manufacturers. An example of a filled laser μVia from layer 4 to 3 is shown in the stackup. The image is sourced from an email exchange with AT&T.

Stackup

Because PCB1 requires the use of μVias, the PCB manufacturers stated that the stackup had to be modified. The recommended stackup is shown in Figure 4.26. The design files in Altium use the stackup used in the custom evaluation board recommended by STM. The PCB manufacturers stated that the stackup could be modified directly by them, and the changes would not affect the design in Altium. The reason why this stackup must be used is that the in-pad μVias requires copper filling. Thus, a given ratio between via height and diameter must be fulfilled to ensure proper filling, see Section 3.4.3. According to the PCB manufacturers, the aspect ratio must be at least 0.9 to enable copper filling of the smallest μVias used in this design. The simulation software SaturnPCB Design Toolkit was used to verify that the ratio was upheld and that parameters of the μVias allowed for sufficient current flow. The parameters from the stack up and Altium were inserted. The maximum current flow is 602 mA, and the thermal resistance is 36°C/W.

Surface Finish and Coating

For the surface finish, the PCB manufacturers recommended using Electroless Nickel immersion Gold (ENIG). In addition, for all PCB headers and connection points, it was recommended to use gold, and not tin, due to the increased noise and strength performance of the gold coating.

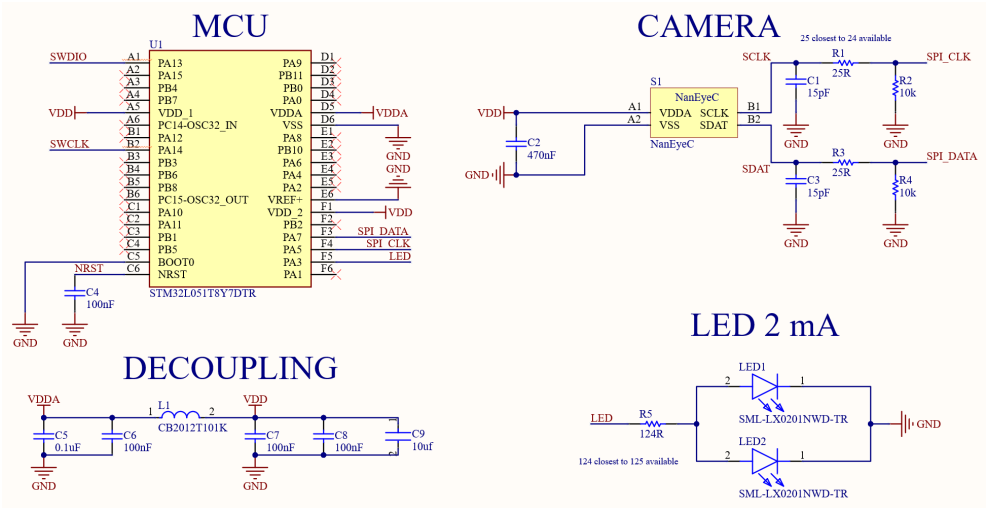


Figure 4.27: Schematic for PCB1, containing the NanEyeC camera module, MCU and LEDs

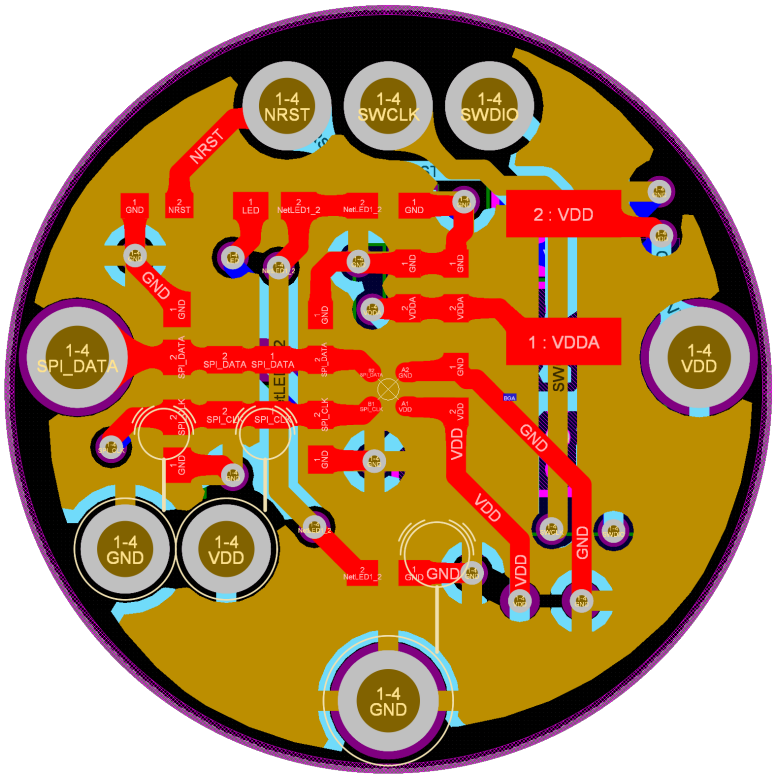
4.5.3 PCB1: Camera and Microcontroller Unit

PCB1 contains the camera module NanEyeC, the STM32L051T8Y7DTR 36-pin BGA MCU and two SML-LX0201NWD-TR 0201 LEDs. Figure 4.27 show the complete schematic for PCB1. The implementation of the camera module is consistent with the implementation for the custom evaluation board, described in Section 4.3.2. The camera is placed in the center of the circular PCB with a 9.5 mm diameter. The LED implementation is also equivalent to the 0201 LED implementation on the custom evaluation board, described in Section 4.3.4. The placement of the LEDs are 2.3 mm from the camera module on the upper and lower sides. The initial placement was 3 mm from the camera module, as discussed in Section 4.5.1. However, they were moved closer to not conflict with the mechanical assembly.

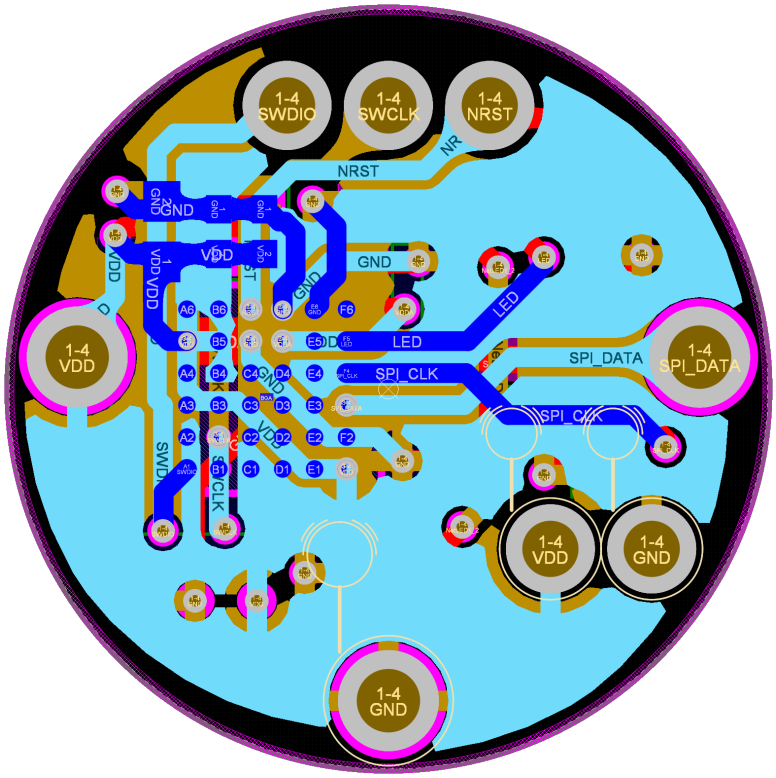
Figure 4.28 shows 2D and 3D views of PCB1 in Altium Designer. The top 2D view is shown in Figure 4.28a and a flipped bottom view is shown in Figure 4.28b. The same views are shown in 3D in Figure 4.28c and 4.28d, respectively.

MCU Footprint

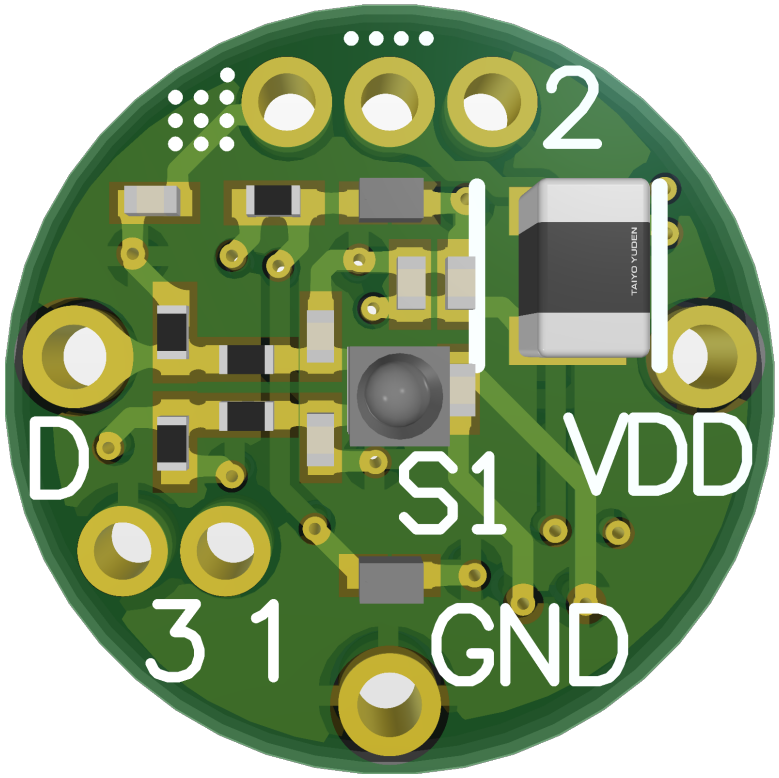
The library entry for this MCU had to be made manually in collaboration with the PCB manufacturer, so they were able to produce the design. This library entry is included in the Integrated library WCE_thesis. The pads in the footprint had to align with the μVias size in manufacturing. ExceptionPCB gave us the link to this YouTube video [80], where Mike Devine from ExceptionPCB explains how to use in-pad μVias properly. Additionally, he explains how to implement it in Altium, and the interviewer, Robert Feranec, gives some advice in terms of routing a BGA when using in-pad μVias. In accordance with the video and the capabilities document the μVias have the following specifications: Hole size =125 µm, AR = 50 µm, Pad



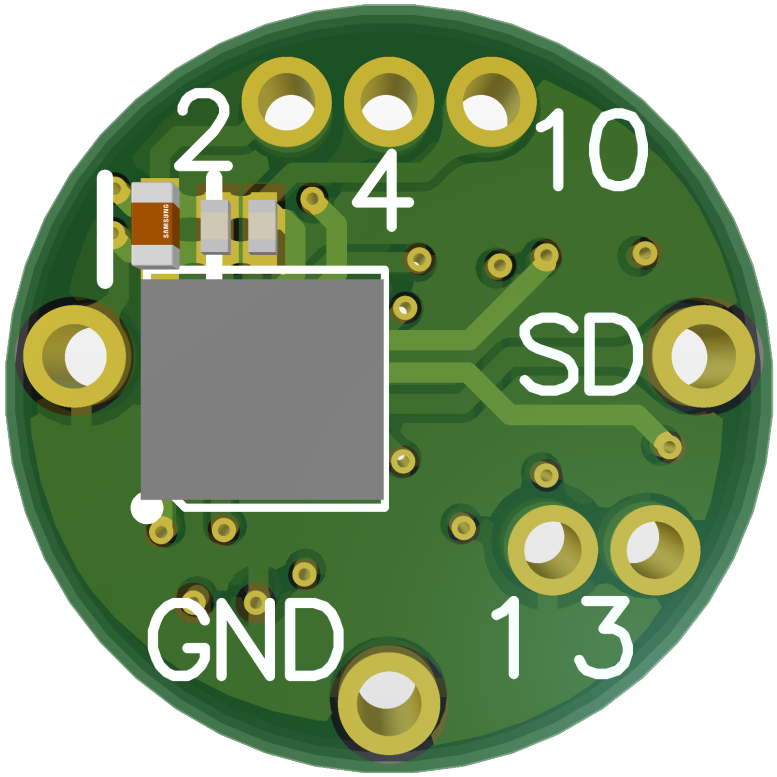
(a) PCB1 - 2D top view



(b) PCB1 - 2D flipped bottom view



(c) PCB1 - 3D top view



(d) PCB1 - 3D flipped bottom view

Figure 4.28: Images of the 2D and 3D view for PCB1. The images are screenshots from Altium Designer.

size = 225 µm, and SME = 0. The SME is zero because the footprint already have a SME = 50 µm. The pad size in the footprint is also 225 µm. The internal pad for the μVias is 325 µm to have tolerance for environmental changes and thermal conduction from other parts of the capsule prototype.

MCU Thermal Estimation

According to the datasheet for the STM32L051T8Y7DTR MCU, the maximum power dissipation is 338 mW, which would lead to a temperature increase of 85°C. This requires that the GPIO draws the maximum sink current of 25 mA and that the MCU operates at 32 MHz with many memory access and heavy continuous processing. For the two LEDs and SPI, the total sink current for the GPIOs is just 4.1 mA, and the core is running at 24 MHz with no heavy computational tasks. Thus it is safe to assume that the temperature increase will be well below 85°C. According to the STM thermal management guide [132], a rough estimate of the temperature increase is given by

$$T_{increase} = \text{Input current} \cdot \text{VDD} \cdot \Theta_{JA}, \tag{4.16}$$

where $\Theta_{JA} = 59°\text{C/W}$ for the STM32L051T8Y7DTR MCU. By simulating the current consumption for the MCU an estimate of the input current for this design is 13.2 mA, see Section 5.7. In addition to this the GPIOs are used to power the two LEDs. This results in

$$T_{increase} = 17.2\,\text{mA} \cdot 3.3\,\text{V} \cdot 59°\text{C/W} = 3.3°\text{C/W}, \tag{4.17}$$

which is negligible, and thus no thermal consideration are done for PCB1.

Programming and Interconnection with PCB2

For programming, the 5-pins from the SWD protocol are used. This is explained in detail for the custom evaluation board in Section 4.3.3. The numbering of the pins is included in the silkscreen print on both sides, in accordance with the numbering used on the custom evaluation board. This can seen in the images of the 3D models in Figure 4.28c and 4.28d. As discussed in Section 4.5.1, it is not enough space for a connector, and thus five plated through-hole vias are used. The pins that will used on the programming board, see Section 4.6, is 1.27 mm PCB headers. According to the datasheet of the 1.27 mm PCB headers, M52-040023VXX45 [133], the recommended hole size is 0.85 mm. The width of the header is 0.46 mm. From Pythagoras Theorem, the diameter is 0.65 mm. In order to have a little wiggle room, the hole size was set to 0.71 mm, with AR = 0.205 mm. The same PCB headers are used for the interconnect with PCB2. These holes are also slightly reduced due to space limitations and are 0.80 mm. The AR = 23.5, which keeps the relation between the pad and hole size, and is enough to get an IPC certified solder. The white rings around the two GND vias for the PCB header and the VDD via is called antennas and is a warning generated by Altium because it does not

know that it will be soldered and connected to the corresponding signal lines in the subsequent PCB2.

Decoupling

The decoupling scheme is more or less equivalent to the decoupling scheme described for the custom evaluation board in Section 4.3.3. The only differences are that this MCU have a V_{REF+}-pin and only one VSS-pin. According to the hardware implementation guide, V_{REF+} can be grounded, as we are not using the ADC or DAC. This will power down the external voltage reference and save power [116]. Because this MCU only have one VSS-pin, it has an unbalanced set of VDD and VSS pairs. As a result, extra precautions must be made to avoid ground loops. This was solved by placing the three VDD decoupling capacitors on the bottom as close to the two VDD pins as possible. From the smallest capacitor, a through-hole via goes to the inductor placed on the top, which is too large to be placed between PCB1 and PCB2. Next to the inductor on the top side is the two decoupling capacitors for VDDA, which are connected to another through-hole via, connected to the power plane, and the μVia connected to the VDDA pad. This decoupling placement and traces have been checked and verified by a PCB design expert.

Routing

The routing of this PCB was a small nightmare, with only four layers. In accordance with the EMC guidelines, the ground plane was used as little as possible, especially directly underneath the MCU. This caused the power plane to have many traces underneath the MCU. This is the reason for the orientation of the stackup; the μVias can only connect two layers at a time. Figure 4.28b clearly shows the impact this had on the light blue power plane. The plane is almost completely gone beneath the MCU. Several different trace thicknesses were explored. However, it did not make a positive impact before they were so narrow that they were close to the limitation of the minimum width supporting the current, based on Equation 3.1 and 3.2 in Section 3.4.2. In order to avoid any negative or unforeseen implications, it was decided to use traces larger than 0.127 mm. In the end, it was achieved only to use 0.254 mm trace widths. The GPIO outputs were also rearranged compared to the custom evaluation board in order to optimize the routing.

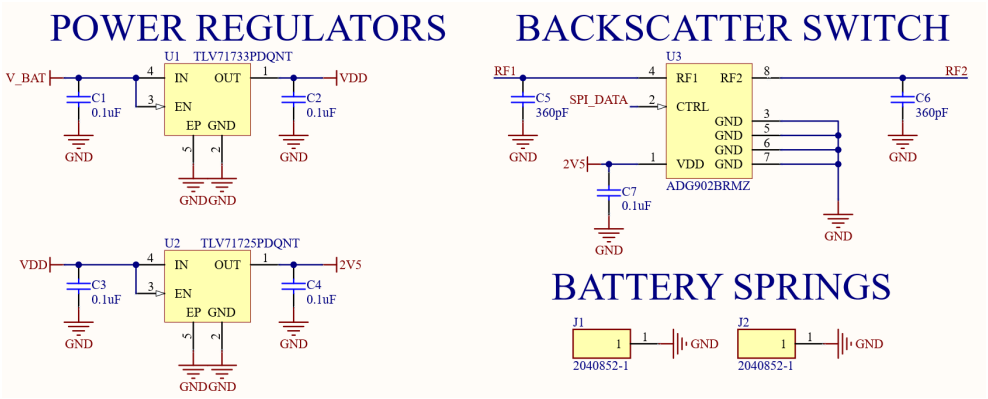


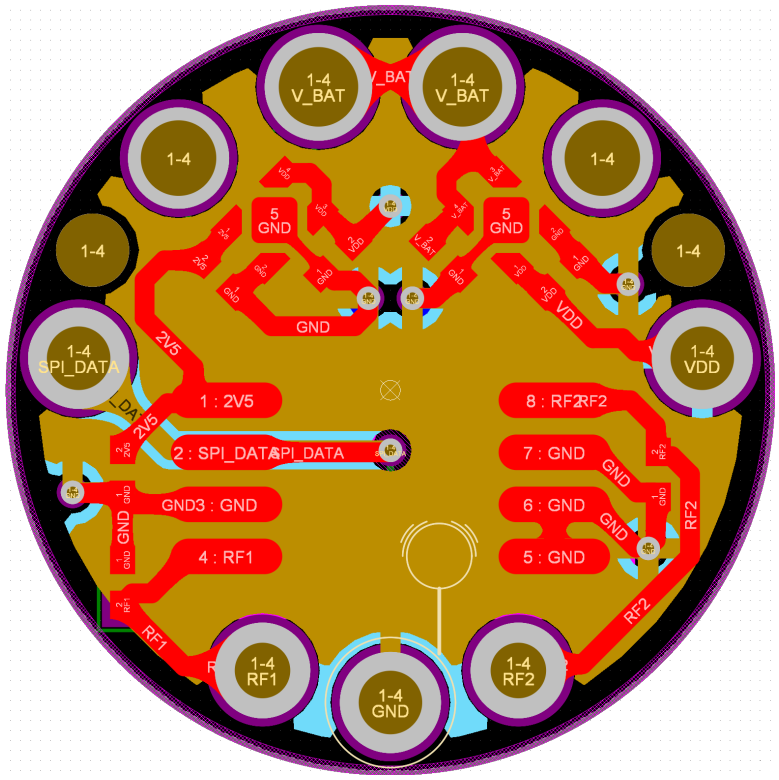
Figure 4.29: Schematic for PCB2, containing the two voltage regulators, the backscatter switch and battery springs

4.5.4 PCB2: Backscatter Switch and Power Regulation

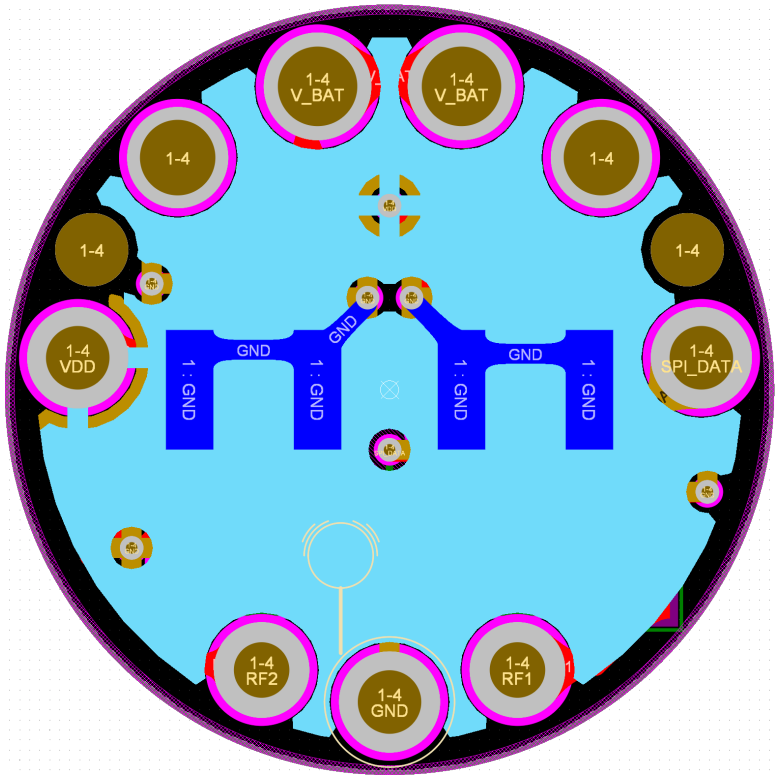
PCB2 contains the 2V5 and 3V3 LDOs and the backscatter switch. This PCB functions as the negative connecting pad for the battery pack. Thus all components have to be placed on the top side of the PCB. Figure 4.29 shows the complete schematic for PCB2. PCB2 acts as a connection link both electrically and mechanically between the three PCBs. The consequence of this is several plated through-hole vias along the rim of the PCB. These vias are pushed to the limit in terms of placement and AR size; they only have a 0.01 mm tolerance. These limited the possible placements of the components. The backscatter switch is quite large and was placed first. The 3V3 LDO had to be placed close to incoming battery voltage, and it was natural to place the 2V5 LDO input as close to the 3V3 LDO as possible. The next consideration was to minimize the length of the 2V5 trace. All of these factors lead to the combined placement. The placements were optimized to allow 0.254 mm width traces all over. On the bottom, the battery springs are placed in the center of the silkscreen ring, indicating the battery position. Figure 4.30 shows 2D and 3D views of PCB2 in Altium Designer. The top 2D view is shown in Figure 4.30a and a flipped bottom view is shown in Figure 4.30b. The same views are shown in 3D in Figure 4.30c and 4.30d, respectively.

Backscatter

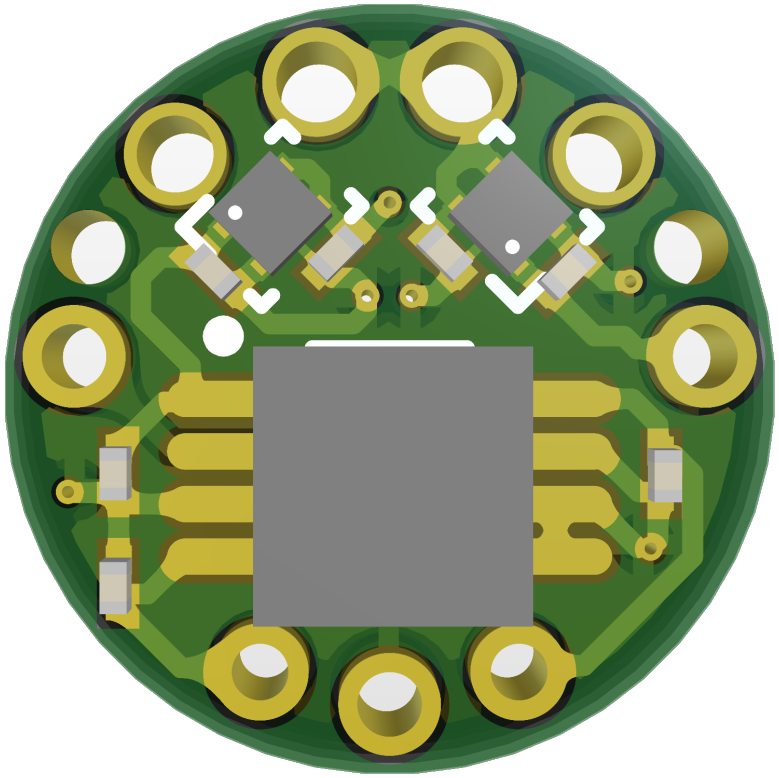
Due to the increased performance, it was decided to use the ADG902 backscatter switch to see if it was possible to implement in a capsule-size prototype regardless of the larger size. The schematic for the backscatter switch is different from the implementation for the custom evaluation board. This change is based on testing. For this implementation of the backscatter switch, both RF1 and RF2 have a bypass capacitor. We wanted to isolate the RF inputs of the switch from any incoming DC components. This is obtained by making the impedance 1 Ω. From the impedance



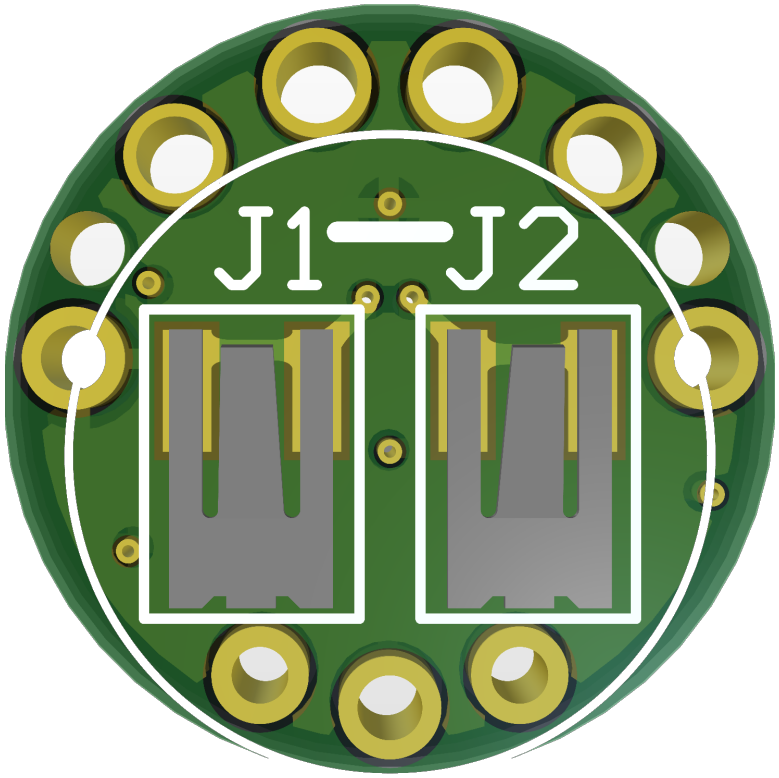
(a) PCB2 - 2D top view



(b) PCB2 - 2D flipped bottom view



(c) PCB2 - 3D top view



(d) PCB2 - 3D flipped bottom view

Figure 4.30: Images of the 2D and 3D view for PCB2. The images are screenshots from Altium Designer.

equation for capacitors, an equation to find the capacitor value can be derived

$$Z = \frac{1}{\omega C} = \frac{1}{2\pi f C} = 1\Omega \Rightarrow C = \frac{1}{2\pi f Z}, \tag{4.18}$$

where f is the frequency of the carrier wave used by the backscatter. The capacitor value is

$$C = \frac{1}{2\pi \cdot 434\,\text{MHz} \cdot 1\,\Omega} \approx 370\,\text{pF} \tag{4.19}$$

The closest capacitance for 0201 capacitors is 360 pF. This results in an impedance of 1.02 Ω, which was approved by the backscatter developer team. The two large vias beneath the leads of the backscatter switch are connected to RF1 and RF2. They are fitted for the cable that will connect the backscatter switch to the copper foils. The hole size is 0.7 mm, and the AR is 0.25 mm.

Voltage Regulators

For the voltage regulators, the space was very limited. The main consideration in the search for LDOs was to minimize the size. The smallest LDO series in stock was TLV717P from Texas Instruments [102]. The specification was optimal for our use and they came in a 1x1 mm DFN package. The current rating is 150 mA, which is more than enough for this implementation. The 3V3 and 2V5 voltage regulators from this series were used, TLV71733PDQNT and TLV71725PDQNT, respectively. LDOs are often the cause of thermal issues. In order to estimate the thermal impact of the LDOs, the same assumptions made for the LDOs on the custom evaluation board are used, see Section 4.3.8. The thermal junction resistance is 393°C/W. This results in a temperature increase of

$$T = 31.4\,\text{mW} \cdot 393\,^\circ\text{C/W} \approx 12.3\,^\circ\text{C} \tag{4.20}$$

for the 3V3 LDO which is almost negligible. The temperature increase for the 2V5 LDO is

$$T = 80\,\text{nW} \cdot 331.4\,^\circ\text{C/W} \approx 0\,^\circ\text{C}, \tag{4.21}$$

which is negligible.

The documentation for the TLV717P LDO was comprehensive and advanced. However, this raised some questions in terms of the decoupling layout. Texas Instruments was contacted about the issues. The customer support responded that they honestly did not understand the layout themselves. We were referred to an engineer who responded: "Yes, it is correct, follow the figure." The figure in question, with the copied figure text from the datasheet, is shown in Figure 4.31.

From forums and documentations for similar LDOs, it was opted only to use a single ground plane. A hardware and PCB expert was consulted about the issue, and he stated that it should not be any problem only using a single connected ground plane for our purpose. He did also note that for miniaturized designs using LDOs it must be taken special consideration in terms of heat dispersion and

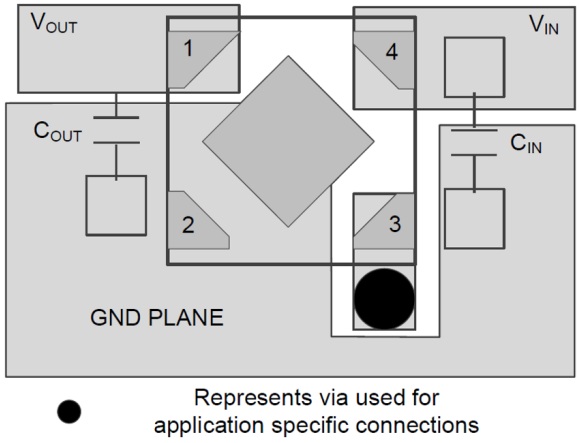
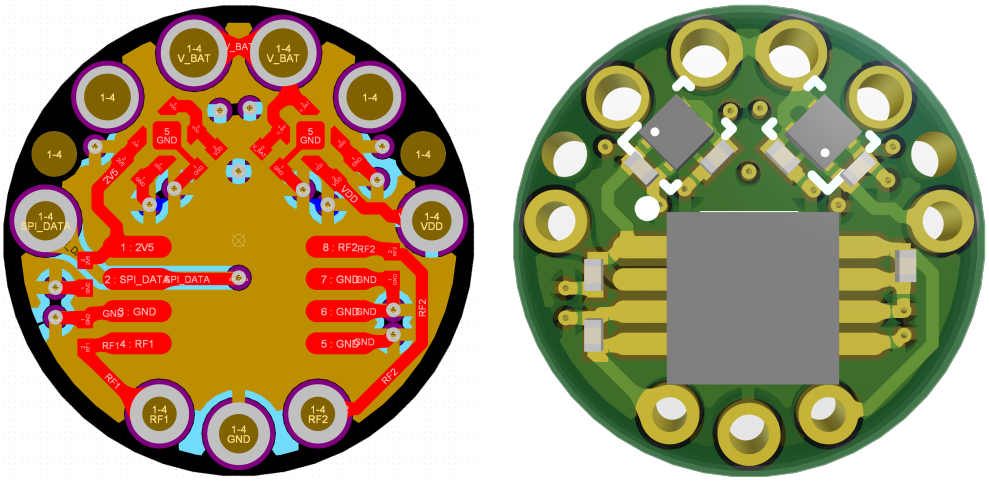


Figure 18. Recommended Layout Example

Input and output capacitors should be placed as close to the device pins as possible. To improve AC performance (such as PSRR, output noise, and transient response), TI recommends designing the board with separate ground planes for VIN and VOUT, with the ground plane connected only at the device GND pin. In addition, the output capacitor ground connection should be connected directly to the device GND pin. High ESR capacitors may degrade PSRR performance.

Figure 4.31: The recommended layout for the TLV717P LDOs, copied from the TLV717P datasheet [102]

inductance. In order to keep the inductance low, vias must not be placed in between the LDO and decoupling capacitor. However, several small vias on the other side of the capacitor are beneficial to decrease the inductance and improve heat dispersion. Leading the heat to the two copper layers inside the PCB enables effective heat dispersion. In encapsulated miniaturized designs, like the WCE even slight temperature increases might cause problems in the long-term operation of the device. Because of these inputs, it was explored using thermal vias to improve heat dispersion and lower the inductance. The via choice was based on simulations in SaturnPCB Design Toolkit, see Section 3.4.4. Figure 4.32 shows the 2D and 3D view of the implementation with six thermal vias placed on the outside of the decoupling capacitors. However, it was not possible to obtain without violating the SME to SME clearance. A solution to fit some of the thermal vias was to change the implemented thermal vias to utilize μVias, buried vias, and blind vias. However, this was not implemented to save manufacturing costs. We tried to simulate the thermal properties of the design using the Altium Designer Thermal simulation tool. This was not successful; due to lack of computational power, the PC just crashed after a few minutes of simulation.



(a) 2D top view **(b)** 3D top view

Figure 4.32: Images of the 2D and 3D view for thermal vias explored for PCB2. The images are screenshots from Altium Designer. This was not used in the final implementation due to space limitations

4.5.5 PCB3: Battery Holder and Power Switch

PCB3 is the simplest PCB only containing battery springs and a switch to turn on and off the battery supply. Figure 4.33 shows the complete schematic for PCB3. It was challenging finding a switch small enough with high enough current rating for this application. The selected switch is the DIP switch CHS-01TA with J-hook leads, rated at 6V 100 mA [134]. This was placed in the center of the PCB. Figure 4.34 shows 2D and 3D views of PCB3 in Altium Designer. The top 2D view is shown in Figure 4.34a and a flipped bottom view is shown in Figure 4.34b. The same views are shown in 3D in Figure 4.34c and 4.34d, respectively.

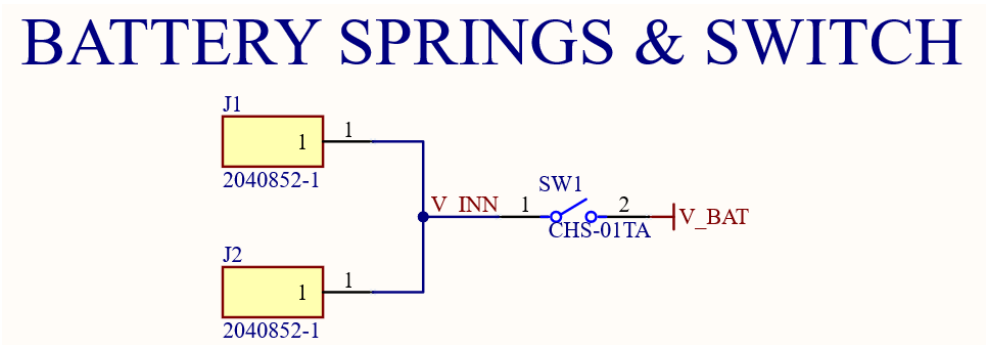
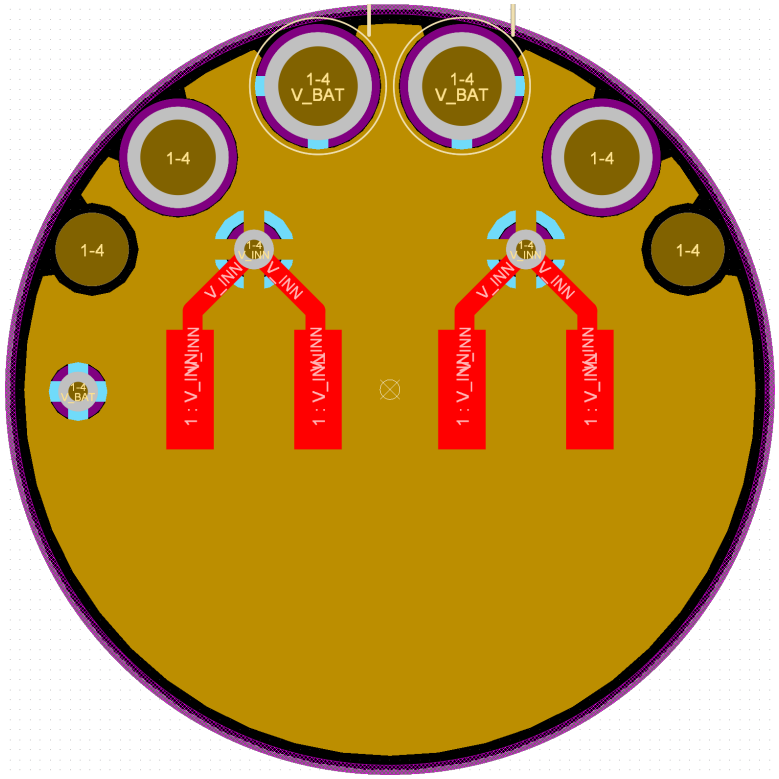
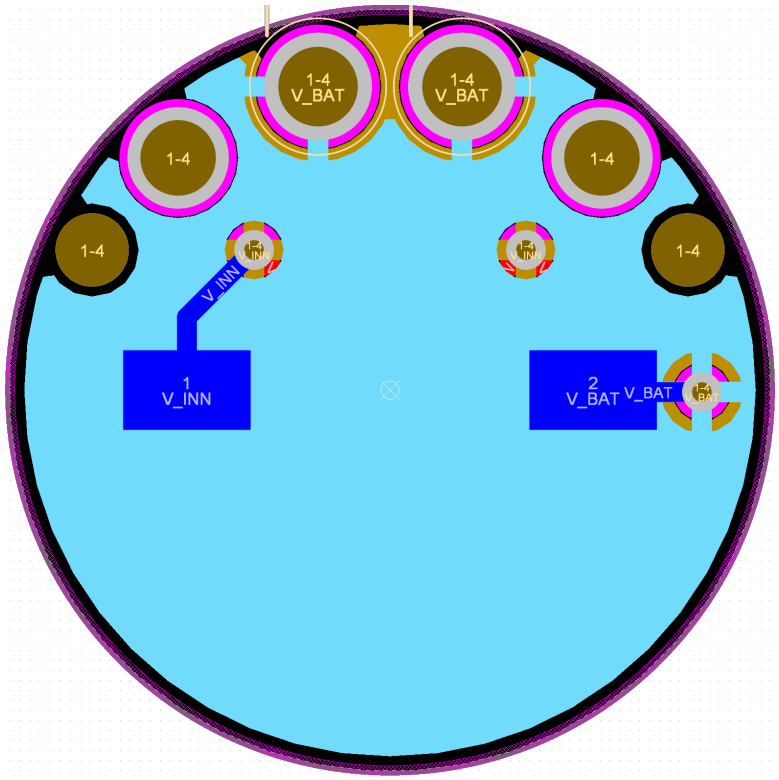


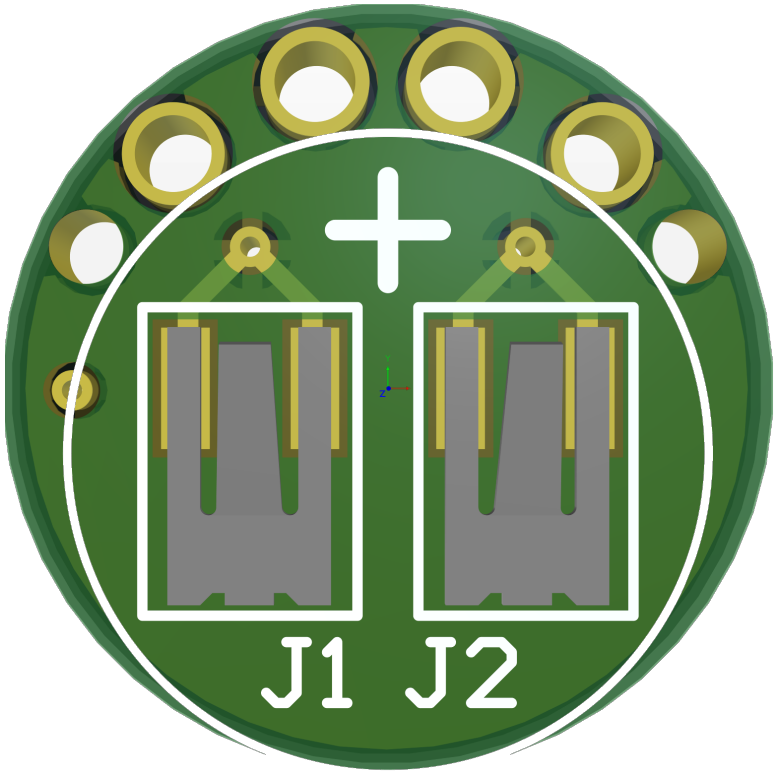
Figure 4.33: Schematic for PCB3, containing a power switch and battery springs for the positive end of the battery pack



(a) PCB3 - 2D top view



(b) PCB3 - 2D flipped bottom view



(c) PCB3 - 3D top view

(d) PCB3 - 3D flipped bottom view

Figure 4.34: Images of the 2D and 3D view for PCB3. The images are screenshots from Altium Designer.

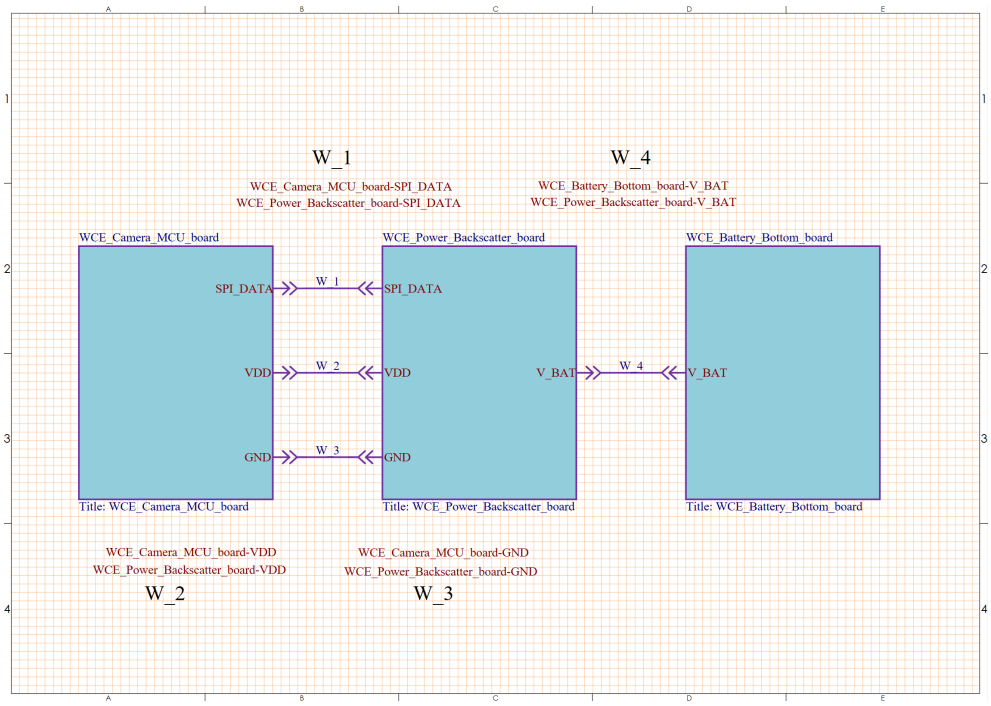


Figure 4.35: Schematic for the multi-board assembly in the Altium Designer Multi-board design tool

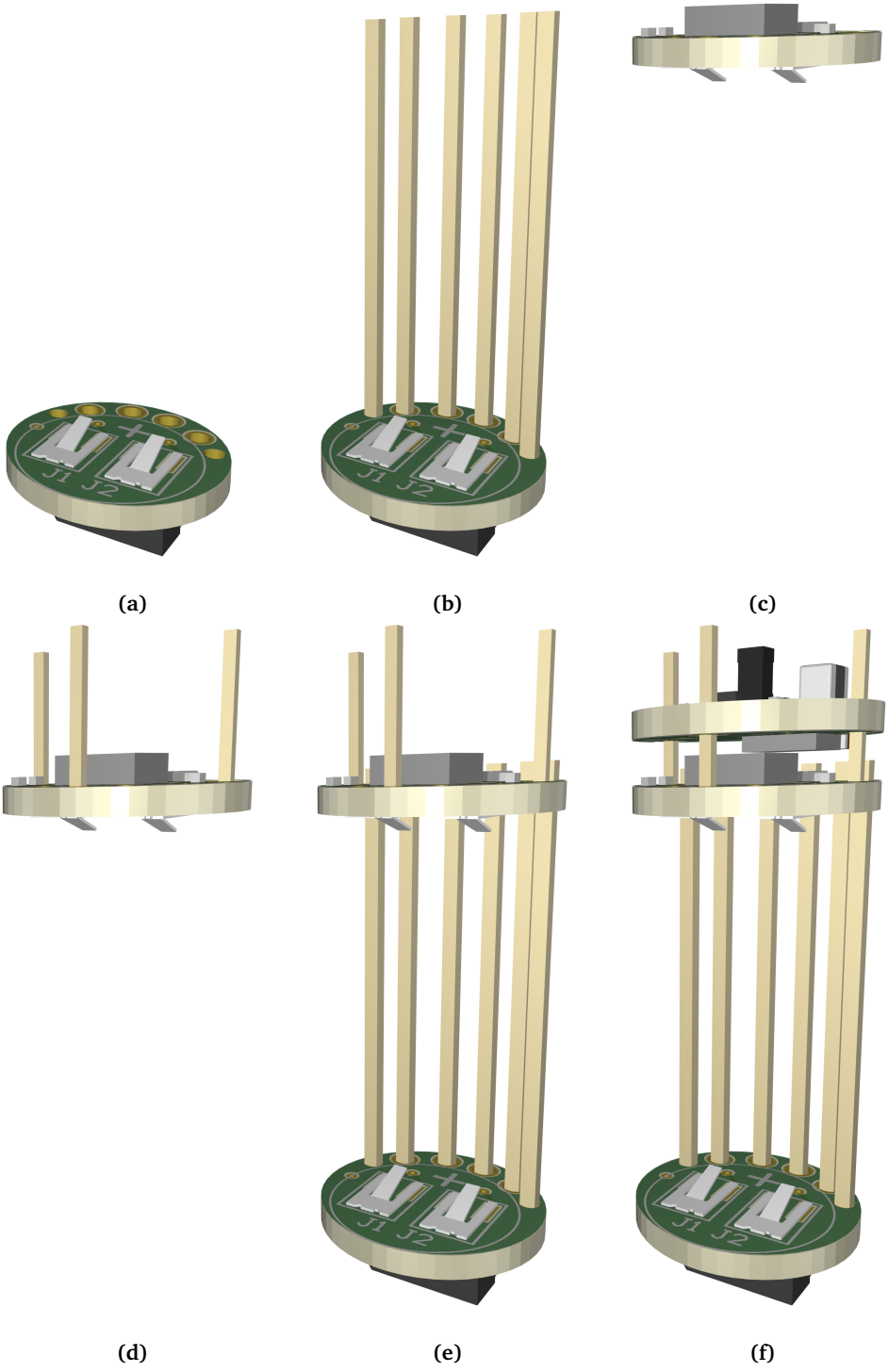
4.5.6 Mechanical Assembly

For the mechanical assembly, Altium Designer Multi-board design tool [135] and SolidWorks were used to visualize and analyze the assembly. Figure 4.35 shows the multi-board schematic, where all three PCBs are included. Each PCB have a box in the schematic with the connections between them. From left to right: PCB1 with the VDD, GND, and SPI_DATA connections to PCB2, connected to PCB3 with V_BAT.

Many different assembly methods have been tried out. The final version uses six 2.54 mm PCB headers [136] and three 1.27 mm PCB headers [133]. From the 3D analysis, it is estimated that the 2.54 mm PCB headers must be 20 mm to provide enough support for a strong solder. The 2.54 PCB header, TS-0769 929647-07-36 from 3M, is 23.11 mm long and must be cut to length. Because the camera module is very fragile and easily can break off, it should be strengthened by adding some epoxy resin to the base. Additionally, we opted to make a protection ring. This will be placed around the camera module to ensure that it will not be damaged. It was modeled to fit around the surrounding components. The protection ring is 3D-printed in transparent PLA. The ring is placed close to the LEDs, but as it is transparent and 3D-printed, it will cause high dispersion of the light, which helps mimic uniform lighting. To hold the protection ring in place, it has holes that perfectly align with the 1.27 mm PCB headers used to connect PCB1

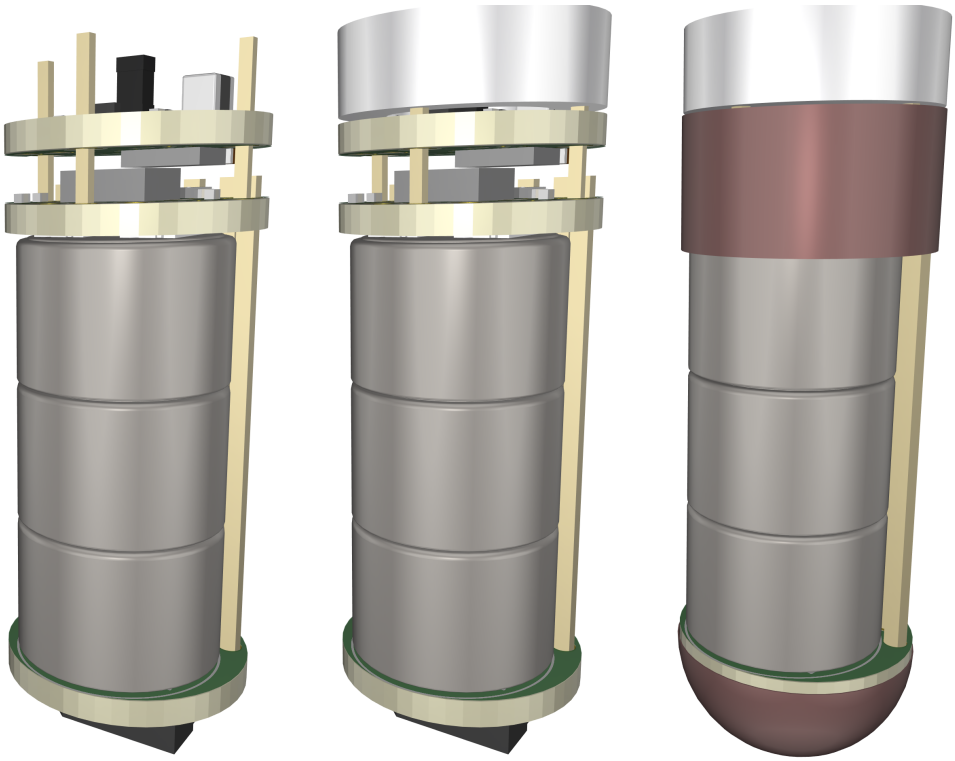
and PCB2. The 1.27 mm PCB headers, M52-040023V0545 from Harwin, are 8.8 mm, which fits perfectly. Figure 4.36 with 9 subfigures (a-i) shows the assembly process. Models of the PCB headers were made in SolidWorks, and the entire assembly was put together using the 3D-tool in Altium Designer Multi-board tool. The assembly process is described in the figure text. The total height of the assembled 3D model of the capsule prototype is 25.14 mm without the switch and 27.65 mm with the switch.

(Intentionally left blank to get the 9 figures side by side in a two-page view)



(a) **(b)** **(c)**

(d) **(e)** **(f)**



(g) (h) (i)

Figure 4.36: The assembly process of the capsule prototype, illustrated in 9 subfigures (a-i). (a) PCB3 is prepared. (b) the 20 mm long 2.54 mm PCB headers mounted on PCB3. The four PCB headers in the middle are soldered on. The two outer PCB headers must be glued on using epoxy because the holes are non-plated through-hole vias due to the limited space. (c) PCB2 is prepared. (d) the three 8.8 mm long 1.27 mm PCB headers are soldered on. Note that these must be placed halfway through the hole when soldered to avoid that they protrude and touch the battery pack. (e) PCB2 is soldered and glued on to the 2.54 mm PCB headers on PCB3. When making this solder, the batteries should be placed between the two PCBs to find the correct distancing. In order to maintain the pressure when soldering, Kapton tape can be used to hold them together. Kapton tape is ESD proof and heat resistant. (f) PCB1 is soldered onto PCB2. Before soldering, a flexible two-way ESD-proof tape should be placed between the MCU and backscatter switch to secure it for mechanical impacts. (g) the battery pack is placed between PCB2 and PCB3. It does not show on the 3D model, but the sides of the batteries should be covered by a round of Kapton tape. This will add mechanical support for the batteries and keep the batteries electrically insulated from the two 2.54 mm PCB headers in the middle, which carries the positive battery charge from PCB3 to the LDO on PCB2. (h) The transparent protective 3D-printed ring is placed on top of PCB1. (i) finally, the antennas are mounted. The width of the upper ring is 5 mm. The wires used to connect the backscatter switch to the antennas are not included in the 3D model.

Table 4.8: Summary of the optical dome manufacturers contacted

Company	Country	Medical grade
Shalom EO	Asia	YES
Sinoptix	Europe	YES
Artifex Engineering	Europe	YES
GH	Europe	YES
Shanghai Optics	USA	YES
Knight Optical (USA) LLC	USA	YES
Adamas Optics, LLC	USA	YES
Prolog Optics	Asia	NO
Lahat Technologies Ltd	Asia	NO
ECOPTIK LTD	Asia	NO
Sherlan Optics	USA	NO
Meller Optics	USA	NO
A. Optical Components Ltd.	USA	NO

Optical Dome and Capsule

The capsule and optical dome for a medical product like this have very strict requirements. For the prototype, the main concern was to make something waterproof and transparent. The first idea was to 3D-print or mold a capsule and order an optical dome.

13 manufacturers of optical domes was contacted. Table 4.8 summarizes the company names, their county of origin, and their response on whether they could manufacture optical domes for a WCE. They all requested the following information.

- Material specifications
- Filtering specifications
- Radiation angles for light dispersion
- Complete 3D-model
- Outer and inner diameters with tolerances
- Optical flatness
- Scratch-dig

This was not knowledge we had easy access to, and with our time perspective, this was not pursued further.

For the 3D printing of the capsule, a resin printer would be used, which gives a very accurate model. Resin printing also has the option of using flexible and waterproof materials. A few different capsules were designed in SolidWorks. The prototypes were printed with PLA on a Prusa i3 MK3S 3D printer. The ideal case is to mold the capsule. This is the manufacturing process used in the industry when making a capsule for the final product, but this is very expensive. Therefore, we consulted an expert on resin molding. He stated that it is possible to 3D-print a mold and fill it with epoxy resin. The mold will then be onetime use and have to

(a) (b)

Figure 4.37: The test tubes that would have been used to protect the capsule prototype from the liquids. The bottom of the test tube serves the purpose of the transparent optical dome. (a) shows the test tube and rubber cap from the manufacturer, and (b) shows the cut test tube and cut rubber cap that would have been used.

be pried off. Because we did not get a hold of a transparent optical dome, these capsule implementations were not pursued further.

The solution for the prototype was to use chemical test tubes. These are made with heat-strengthened glass and can be cut and molded with diamond tools. The test tubes used have 10 mm inner and 13 mm outer diameters. A Dremmel rotary tool with a diamond cutting disc was used to cut the length of the test tube to 35 mm, to have space for the entire capsule-size prototype, including the power switch. In order to increase the available space inside the test tube, 5 mm was cut off the tip of the rubber cap. Figure 4.37 shows the tube before and after cutting. It was tested and verified that it is waterproof when fully emerged in liquid. The tubes and rubber cap were ordered from Frederiksen Scientific, with the part numbers 012110 and 042510, respectively.

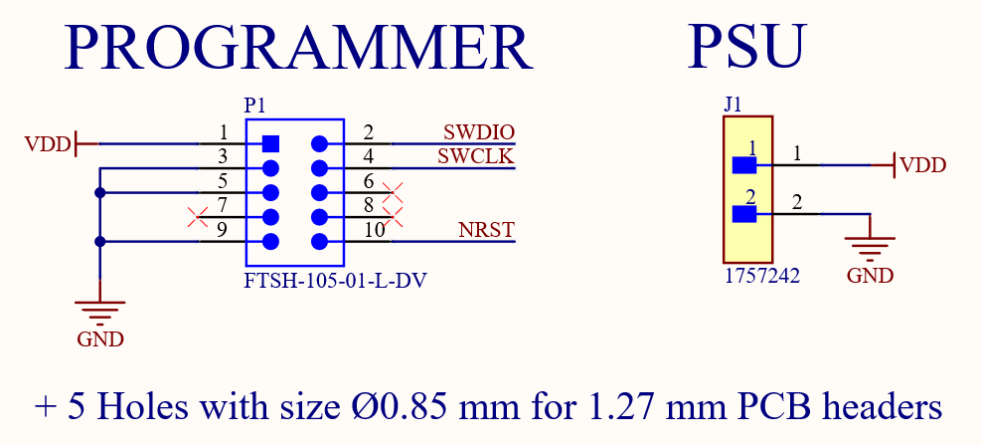


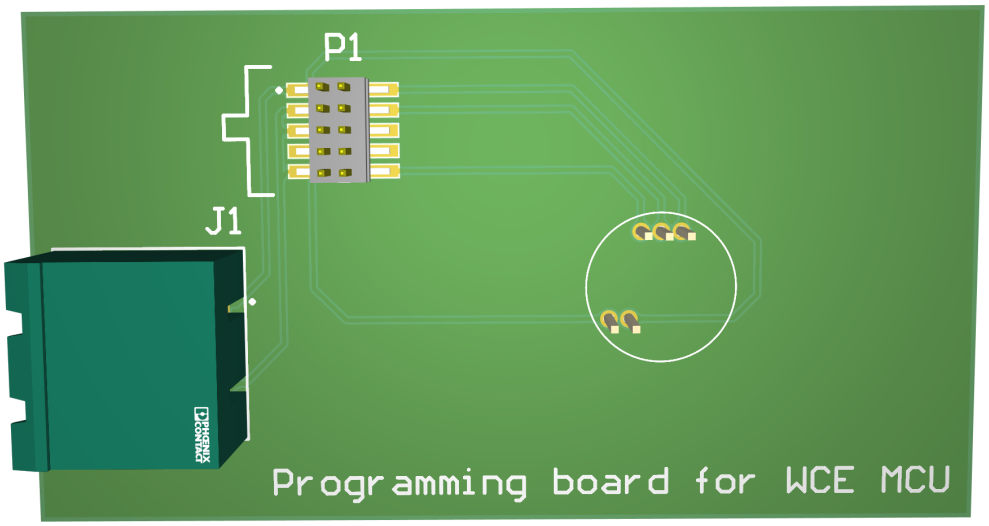
Figure 4.38: Schematic for the programming board

4.6 Programming PCB

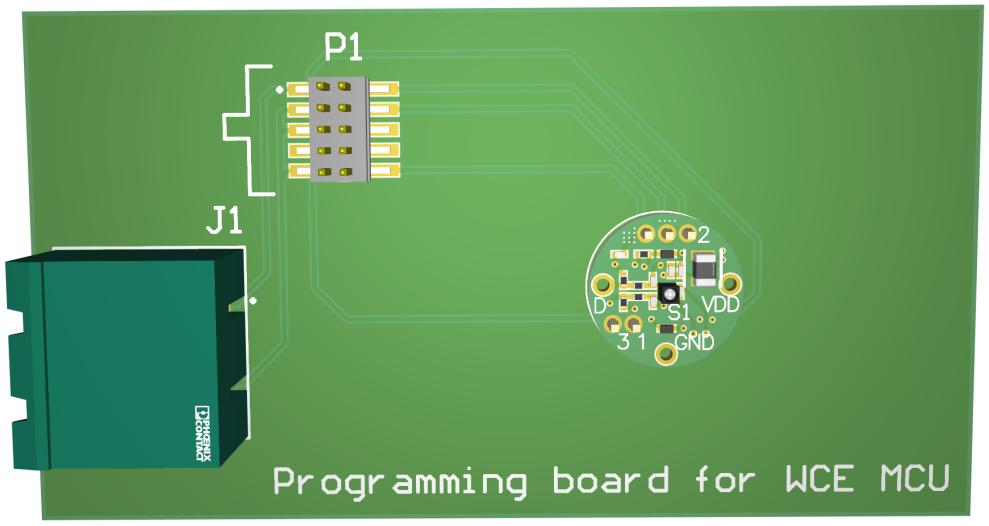
There were several different ideas on how to program the MCU. The main ideas that was considered was a pogo-jig, programming PCB with a connector, custom programming cable and the implemented solution; a programming PCB with PCB header pins. We discussed the design with three different hardware and PCB experts in the industry. All of them were clear that the pogo-jig was the best solution. For mass production, they claimed it to be essential. We started designing this, but it was deemed too time-consuming, and the time was allocated to other aspects of the work. We opted to use the least time-consuming design, which still is relatively robust and draws inspiration from pogo-jigs. Figure 4.38 shows the complete schematic of the programming PCB. This includes the 10-pin JTAG programming connector, described in Section 4.3.3, the same power port used on the custom evaluation board, and five holes for 1.27 mm PCB headers. The placement of the PCB headers corresponds with the vias for programming on PCB1. The pairs of two and three have 1.27 mm pitch so that the plastic stiffeners can be used. This will keep the tolerances to a minimum. Figure 4.39a shows a 3D model of the programming board. The plastic stiffeners that are attached to the PCB headers are not included in the 3D model. Figure 4.39b shows the programming PCB with PCB1 placed on the five PCB headers. Figure 4.40 gives a closer view of PCB1 placed on the PCB headers. PCB1 has a small clearance for the programming vias, making it a tight fit to ensure proper connection on all pins during programming.

4.6.1 Upgrade to Pogo-Jig

The idea is to replace the five PCB headers with spring-loaded PCB headers, with rounded heads for use with plated through-hole vias - which are the vias used for the programming connection on PCB1. The programming PCB is designed with extra space to allow for mounting of a clamping mechanism that will precisely



(a) Without PCB1



(b) With PCB1

Figure 4.39: The programming board without (a) and with (b) PCB1 placed on top of the five 1.27 mm PCB headers used for programming

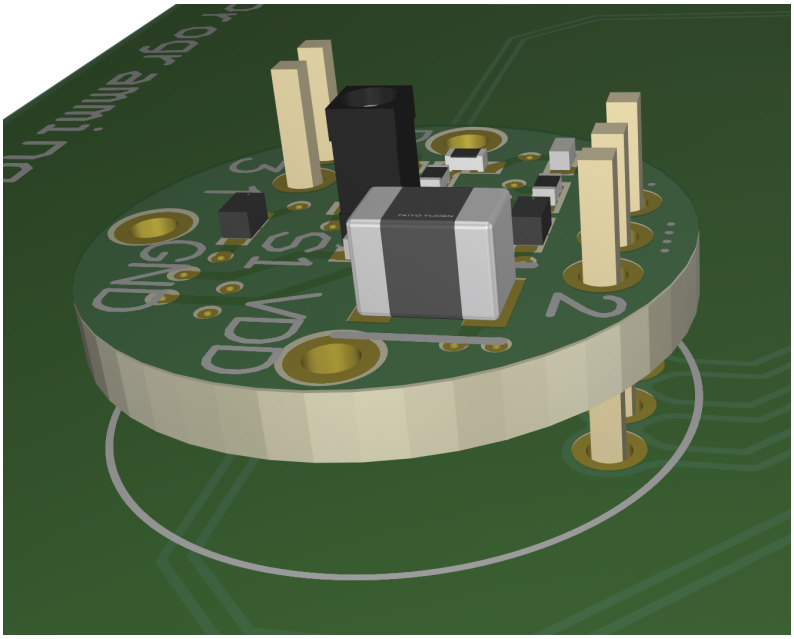


Figure 4.40: Close view of PCB1 placed on the five 1.27 mm PCB headers on the programming board

Figure 4.41: The pogo-pin that was considered using. Available at RS Components, RS Stock No.:261-5137, Out of stock until 22.02.2022. The image is sourced from [137].

press PCB1 down on the spring-loaded PCB headers guided by the three rigid PCB headers without damaging the camera. The three holes in PCB1 used to interconnect it with PCB2 can be used as guides to get the correct orientation of the pogo pins.

An alternative solution is to make a custom programming cable with spring-loaded PCB header pins. Where the PCB is placed in a holder, and then the programming cable is pressed on top. A problem with this solution is that it is easy to touch the camera module with the pins accidentally and, in the worst case, break it off or damage it.

4.7 Software Development

The software was developed on a NUCLEO-L4R5ZI development board connected to the NanEyeC breakout board with short wires. For the implementations that used small RAM size, the custom evaluation board connected to the camera breakout board was used for testing. In total, seven software implementations are made. Bit banging with and without interrupt, SPI with and without DMA for 12-bit and 8-bit, and SPI without DMA for 8 kB RAM, which can run on the custom evaluation board. The code is available on GitHub, see Table 4.1 in Section 4. These implementations can be divided into two main categories: SPI and bit banging. The main implementation techniques have a section each where the implementation of the techniques is presented, including the internal differences between the sub implementations. A brightness module for the LED is also developed and is presented at the end of this section.

The initial idea was to use 12 MHz SPI with DMA and 24 MHz core frequency. This implementation was chosen in accordance with both the STM32 documentation and the NanEyeC datasheet. The use of DMA would free up the CPU for other tasks like adjusting the LEDs. In addition, it would open up for future improvements, e.g., changing the camera configuration during operation to alter the frame rate and optimize the image. However, we could not get this implementation to work, so an alternative implementation using bit banging was pursued. This was a very time-consuming and manual process, with a lot of trial and error. It does not work properly yet and requires more work to enable proper synchronization between the clock and data, and the interrupt scheme must be verified. The current implementation only works properly for very short periods of time, at random times. During the last weeks of the thesis work, it was discovered that the DMA part of the SPI implementation was the reason why it did not work, see Section 5.2.1 in the Results. Based on this information the SPI implementation was rewritten to not using DMA at all. This implementation is stable and works as excepted but uses a core clock frequency of 96 MHz, compared to the 24 MHz clock used for the SPI with DMA implementation.

The datasheet for the NanEyeC was ambiguous when it came to the software implementation of the interfacing. It only states that the interfacing uses the SEIM protocol developed by ams, which is a version of half-duplex SPI master without CS. This means that the MCU do not have any control over when the camera sends data and must carefully follow the internal state machine in the NanEyeC camera module to stay in sync. The state machine is presented in Figure 4.6 in Section 4.1. This is the basis used for both implementations. Because of the ambiguity of the datasheet and missing information, the ams NanoBerry evaluation kit was used for extensive testing and verification of our assumptions. The NanoBerry board has pins for the data and clock lines, making it easy to connect to the oscilloscope. The observations on the oscilloscope provided valuable information for both implementations. We did not have access to the code running on the NanoBerry board, but from the testing, it is clear that it uses a continuous clock,

with no gaps in between each SPI package. Additionally, it proved that the startup sequence could use a very low frequency compared to the main state machine.

The NanoBerry board generates the first activation clock in the INITIAL INTERFACE state and the 10 synchronization clocks in the INITIAL Pre-SYNC state at approximately 450 Hz, while the programming sequence uses the frequency of SCLK. The same was true for the 329 PPs in the INITIAL Pre-SYNC state. The datasheet states that the activation clock and synchronization clocks must be bit banged. Further, they state that the transmission of SPI should use normal SPI, while the receiving of data from the camera should use SPI with DMA.

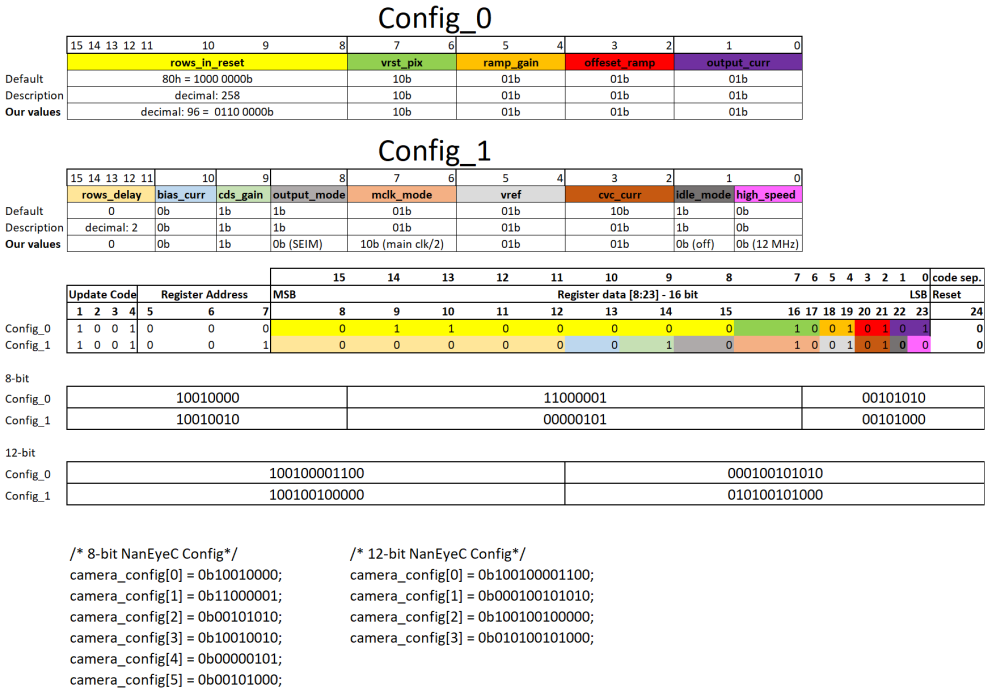
4.7.1 General Guidelines

Some general guidelines were followed for all implementations. To optimize the EMC performance all unused GPIO-pins was pulled low with internal resistors, controlled by software [116]. This was implemented using the CubeMX software. When configuring the clocks in the CubeMX clock three, the SYSCLK and HCLK were always set to the same value, and the APB prescalers set to one to fully utilize the core for SPI and bit banging.

4.7.2 Camera Configuration

The camera has two registers called Config_0 and Config_1, explained in more detail in Section 4.1. To ensure that the camera configuration sequence always was correct and easy to modify, an Excel sheet was made which automatically generates the code, shown in Figure 4.42. The two upper boxes, titled Config_0 and Config_1, show the two registers' different entries and their value. The row labeled *Default* is the default values from the datasheet, and the row labeled *Description* is the default register values from the description of the registers. It is not known why these values deviate. The row marked *Our values* are the register values used in this implementation.

When a value is changed in this row, the box below is automatically updated, using the Excel MID function. The 24-bit configuration sequence is then automatically divided into 8-bit and 12-bit representations, representing the changes made in the two upper boxes. The Excel CONCAT function is used for this purpose. Every time a change was made to the register values, the code sequences generated at the bottom were simply copied for the correct bit implementation. The default register values are used, with the exception of register entries: `rows_in_reset`, `output_mode`, `mclk_mode`, `cvc_curr` and `idle_mode`. `rows_in_reset` controls the exposure time; it was changed to comply with the values used by the NanoBerry evaluation board, where the image was checked in the Graphical User Interface (GUI). How to manipulate and calculate the exposure time is explained in detail on page 28 in the NanEyeC datasheet. `cvc_current` was changed to comply with the recommended value for our frequency. The default frequency is 25 MHz, while this implementation uses 12 MHz. The register entries controlling the frequency are `mclk_mode` and `high_speed`. These are set according to the reference table in



Config_0

	15 14 13 12 11	10 9 8	7 6	5 4	3 2	1 0
	rows_in_reset		vrst_pix	ramp_gain	offeset_ramp	output_curr
Default	80h = 1000 0000b		10b	01b	01b	01b
Description	decimal: 258		10b	01b	01b	01b
Our values	decimal: 96 = 0110 0000b		10b	01b	01b	01b

Config_1

	15 14 13 12 11	10	9	8	7 6	5 4	3 2	1 0	
	rows_delay	bias_curr	cds_gain	output_mode	mclk_mode	vref	cvc_curr	idle_mode	high_speed
Default	0	0b	1b	1b	01b	01b	10b	1b	0b
Description	decimal: 2	0b	1b	1b	01b	01b	01b	1b	0b
Our values	0	0b	1b	0b (SEIM)	10b (main clk/2)	01b	01b	0b (off)	0b (12 MHz)

	Update Code				Register Address			MSB																	LSB	code sep.
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		24
Config_0	1	0	0	1	0	0	0	0	1	1	0	0	0	0	0	1	0	0	1	0	1	0	1		0
Config_1	1	0	0	1	0	0	1	0	0	0	0	0	0	1	0	1	0	0	1	0	1	0	0		0

8-bit

Config_0	10010000	11000001	00101010
Config_1	10010010	00000101	00101000

12-bit

Config_0	100100001100	000100101010
Config_1	100100100000	010100101000

```
/* 8-bit NanEyeC Config*/
camera_config[0] = 0b10010000;
camera_config[1] = 0b11000001;
camera_config[2] = 0b00101010;
camera_config[3] = 0b10010010;
camera_config[4] = 0b00000101;
camera_config[5] = 0b00101000;
```

```
/* 12-bit NanEyeC Config*/
camera_config[0] = 0b100100001100;
camera_config[1] = 0b000100101010;
camera_config[2] = 0b100100100000;
camera_config[3] = 0b010100101000;
```

Figure 4.42: Screenshot of the Excel sheet made to automatically generate the correct coding sequences for the camera configuration

the datasheet to enable 12 MHz. Naturally `idle_mode` is set to 0, to disable IDLE mode. It is worth noting that the `rows_delay` register controls the length of the delay mode $\text{rows}_{\text{delay}} = 16 \cdot$ `rows_delay[4:0]`$+2$. The default value for this register is 0, which means that $\text{rows}_{\text{delay}} = 2$ rows. Thus, is this the reason why the register description says that the default value is 2. The impact of this register on the exposure time is also explained on page 28 in the datasheet.

4.7.3 SPI

The NanEyeC state machine in Figure 4.6 in Section 4.1 is used to implement the SPI implementation. Both the 8- and 12-bit implementations will be presented. It is mainly the 12-bit implementation that is used when explaining, as this coincides with the PP size. The differences for the 8-bit implementations are highlighted in red. The only difference between the 8- and 12-bit is the number of SPI packages for each state. Table 4.3 in Section 4.1, summarizes the number of SPI packages for each state in both implementations. The differences between the SPI implementations with and without DMA are small when only looking at the logic. According to the NanEyeC datasheet, only the receive part of SPI should use DMA. Thus the only visible changes in the main code are the change between `SPI_Receive()` and `SPI_Receive_DMA()`. In the pseudo-code `SPI_Receive()` is used for all lines indicating that the camera is sending data to the MCU. The pseudo-codes for the startup sequence and the while loop, presented in the NanEyeC state machine, are displayed in Listing 4.1 and 4.2. According to the datasheet, SPI with DMA can be fully utilized with a core frequency double of the SPI bandwidth. The core frequency was set to 24 MHz based on this information, as the SCLK is 12 MHz. For the standard SPI, without DMA, it only states that the minimum core frequency must be double of the SPI rate. Thus 24 MHz, was used for both implementation.

Startup

The first part is the startup sequence, shown as its own small state machine, leading to the main state machine in Figure 4.6 in Section 4.1. The startup sequence requires a combination of bit banging and SPI on the same GPIO-pin. The GPIO-pin can not be configured as both at the same time. Thus, an initializing function for bit banging functionality was created, called `Toggle_GPIO_pin_Init`. To initialize and deinitialize the SPI functionality, the HAL functions `MX_SPI1_Init` and `HAL_SPI_DeInit(&hspi1)` are used. The bit banging functionality did not have to be deinitialized as the SPI init function would overwrite it. The logic of the startup sequence is shown as pseudo-code in Listing 4.1. The lines special to the 8-bit implementation are highlighted in red. Figure 4.43 illustrates the waveform for the clock line during the startup sequence. The bit banging is done at 1 kHz, based on the observations done on the NanoBerry board, while the SPI executes at 12 MHz. The 12 clocks for the 8-bit implementation are marked in red. For the 12-bit implementation, these are included as an extra PP in the last block.

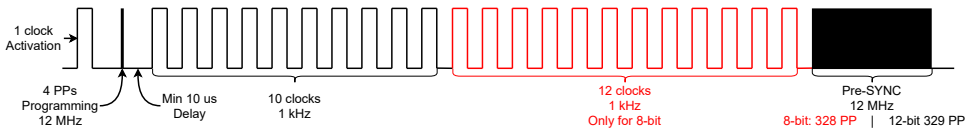


Figure 4.43: Illustration of the waveform for the clock line, during the startup sequence

Code listing 4.1: Pseudo-code for the NanEyeC startup sequence

```
//---- ACTIVATION CLOCK PULSE ----
Bit_Bang_Init()
Bit_Bang(1 clock)          //1 kHz

//---- CAMERA CONFIGURATION: 4 PPs ----
SPI_Init()
12-bit: SPI_Transmit(4)    //12 MHz
8-bit:  SPI_Transmit(6)    //12 MHz

//---- WAIT FOR IDLE STARTUP ----
Delay(1 ms)

//---- 10 SYNCHRONIZATION CLOCKS + 12 FOR 8-bit ----
SPI_DeInit()
Bit_Bang_Init()
Bit_Bang(10 clock)         //1 kHz
8-bit: Bit_Bang(12 clock) //1 kHz

//---- INITIAL PRE-SYNC MODE: 329 PPs FOR SYNCHRONIZATION ----
SPI_Init()
12-bit: SPI_Receive(329)
8-bit:  SPI_Receive(492)
```

Main State Machine

The pseudo-code for the main state machine, placed in the infinite while loop, is shown in Listing 4.2. The pseudo-code is valid for RAM $\geq$ 64 kB. The black code lines are the 12-bit implementation, and the red shows the 8-bit implementation. The effect of smaller RAM is that the READOUT mode must be divided into several operations. For the 8 kB RAM in the L0 MCUs the READOUT operations must be split into 32 operations for both the 8- and 12- bit implementations. The drawback of this is that the small delay between two operations increases the exposure time, so the beginning of every 10th row will be overexposed. According to the datasheet, this is possible to fix in the post-processing of the images. The reason why the READOUT has to be split into two operations is that the STM32 implementation of SPI is limited by a 16-bit counter: 0xFFFF = 65536-bit. This is also true for the STM32 DMA implementation. Another drawback to the STM32 SPI implementation is that there is a gap of 2 clock cycles between each SPI package and a significantly longer gap in between each SPI operation. Because of this, it

is important to keep the number of SPI operations to a minimum and, if possible, use the 12-bit implementation. This is the reason for merging the SYNC and DELAY states into one SPI operation.

Code listing 4.2: Pseudo-code for the NanEyeC main state machine, valid for RAM ≥ 64 kB. The 12- and 8-bit implementations are shown in black and red, respectively.

```
//---- CAMERA CONFIGURATION: 4 PPs ----
SPI_Transmit(4)
SPI_Transmit(6)

//---- INTERFACE MODE: 644 PPs ----
SPI_Transmit(644)
SPI_Transmit(966)

//---- SYNC and DELAY MODE: 1312 PPs -----
SPI_Receive(1312)
SPI_Receive(1968)

//---- READOUT MODE: 104960 PPs -----
SPI_Receive(52480) //1
SPI_Receive(52480) //2
SPI_Receive(52480) //1
SPI_Receive(52480) //2
SPI_Receive(52480) //3

//---- END OF FRAME -----
SPI_Receive(8)
SPI_Receive(12)
```

Implications of Using SPI

The consequences of the two clock cycle gaps between each SPI package are that the 8-bit package becomes 10-bit and the 12-bit package becomes 14-bit. The new execution times of a frame, accounting for this delay, are summarized in the third column of Table 4.9. The second column shows the execution times with a continuous clock used in the NanoBerry board. In addition to this, the delays between each SPI package must be accounted for. However, the length of this delay is not specified in the documentation. The number of SPI operations depends both on the RAM size and SPI implementation. The two last columns in the table list the number of delays with unknown lengths for each implementation. For 8 kB RAM the lowest hole integer that divides the SPI operations at the end of a row is 32 for both the 8- and 12-bit implementations. This results in an array size of 3280 and 4920 bytes for the 12- and 8-bit implementations, respectively.

The delay caused by the gap between each SPI package makes a noticeable impact, and lowers the frame rate from 9.35 FPS to 8.02 FPS and 7.48 FPS for the 12- and 8-bit implementations, respectively. In order to remove this gap, it was attempted to implement SPI using the LL-library and on register level. Neither of the implementations removed the two clock cycles. It appears to be a limitation of the STM32 SPI hardware implementation.

Table 4.9: Execution times of a frame with a continuous clock and the SPI implementations, accounting for the 2 clock cycle delay between each SPI package. The total number of SPI operations for each implementation are also included, which will cause a delay of unknown length. The SPI transfer speed is 12 Mbps.

SPI	Execution Time [ms]		# SPI Operation	
Implementation	Continuous	SPI	RAM = 8 kB	RAM≥ 64 kB
12-bit	106.928	124.749	36	6
8-bit	106.928	133.660	36	7

4.7.4 Bit Banging

This implementation is based on countless different versions and much trial and error. The version presented here is the only bit banging implementation that has achieved camera output and is thus the version we are pursuing. The implementation is based on many assumptions. ams was contacted for clarifications but did not provide any support at all. The idea is loosely based on the state machine in the datasheet for camera chip NaneEye (not NanEyeC) [46]. The state machine for the camera chip does not use the startup sequence and jumps directly into the main state machine. It is not clear from the datasheet how it obtains synchronization. However, we assume that it spams the programming sequence until it successfully aligns and receives data from the camera module, which triggers an interrupt, stops the sending of data, and starts a timer that keeps track of the synchronization and sends the programming sequence at the correct times.

The benefit of the bit banging method, compared to SPI is that it is not limited by the hardware implementation of SPI. Thus, a continuous clock can be used, and the implementation described in the NanEyeC datasheet used on the NanoBerry board can be replicated. However, the programmer has to take care of all synchronization between the clock and data for a pure bit banging implementation. This is much more complicated compared to using a pre-made custom hardware module, like SPI, which handles the synchronization. To generate the continuous clock, the Microcontroller Clock Output (MCO) pin on the MCU is configured to output 12 MHz. The MCO uses the internal PLLs to derive the clock. The accuracy of this process depends on several parameters like temperature, power supply decoupling scheme, and environmental noise. To get the correct clock voltage the MCO was configured to *very high speed* and *ldo boost*. On the data line, the only data that is transmitted from the MCU to the camera module is the 48-bit configuration sequence. This has to be generated by bit banging a GPIO-pin, i.e., toggle a GPIO-pin at the correct intervals.

Camera Configuration Sequence

This sequence must have the exact same frequency as the SCLK. In order to achieve this, all the internal processes and execution times of the commands must be considered. For instance, using the HAL-library to toggle a GPIO-pin comes with a

Table 4.10: Disassembly from toggling GPIO-pin PA5 as fast as possible, with one clock cycle delay

Function	Code	Disassembly
Set GPIO high	GPIOA->BSRR =(1<<5);	mov.w r3, mov.w r2, str r2, [r3, #24]
Set GPIO low	GPIOA->BSRR =(1<<21);	mov.w r3, movs r2, #32 str r2, [r3, #24]
1 clock cycle delay	__ASM volatile ("nop");	nop

Table 4.11: Number of cycles required for the assembly instructions

Instruction	# Cycles
mov.w	1
movs	1
str	2
nop	1

750 ns overhead. Ideally, bit banging should be implemented in assembly to have full control over every single process. However, this is very time-consuming and difficult to debug. Instead, the code can be written in C and analyzed using the disassembly viewer of the compiled code in the STM32CubeIDE debugging mode. From the disassembly, it was discovered that the fastest way to toggle a GPIO-pin is to directly write to the GPIO port Bit Set/Reset Register (BSRR). This implementation used GPIO-pin PA5. PA5 is set, by left shifting a '1' to position 5 in BSRR for GPIO port A: `GPIOA->BSRR =(1«5)`. To reset PA5, a '1' is left-shifted to position 5+16=21: `GPIOA->BSRR =(1«21)`. The shortest delay possible to implement is a one-clock cycle delay. This is achieved by issuing a No Operation (NOP) directly to the CPU. Inline assembly coding is used for this purpose: `__ASM volatile ("nop")`. The three lines of code are placed within a while loop to toggle a GPIO-pin with a delay of one clock cycle. Table 4.10 lists the assembly code generated by the compiler for these three lines of code [72].

This was implemented on the NUCLEO-L4R5ZI, which has an ARM Cortex-M4 core. The number of cycles required for each operation, according to the ARM Cortex-M4 Technical Reference Manual [138], is summarized in Table 4.11. According to the documentation, the infinite while uses 1+P number of clock cycles, where P depends on the pipeline. From testing, it uses two clock cycles. However, it is not certain that it always uses two clock cycles. P can be 1-3 clock cycles, depending on whether the processor speculates the address correctly and only require one cycle to fill the pipeline.

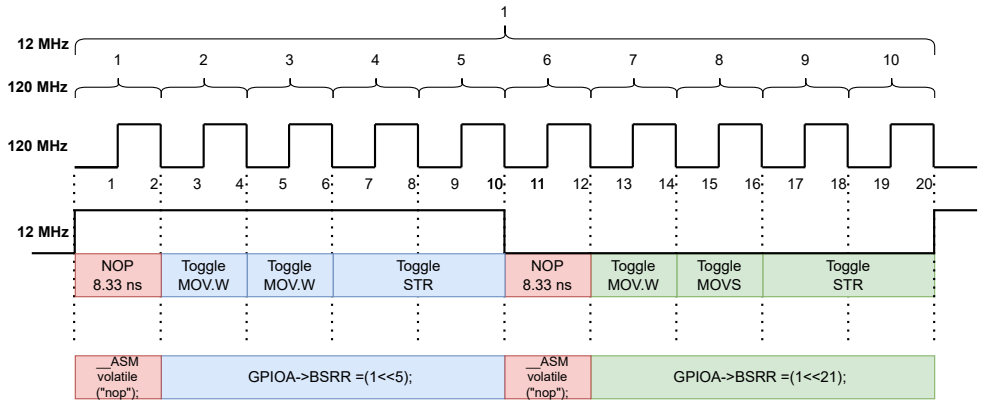


Figure 4.44: Waveform for the bit banging sequence at 120 MHz core frequency, toggling the GPIO-pin at 12 MHz. It also shows the execution time of the write to the GPIO BSRR for both setting and resetting of the register. It also shows the number of clock cycles required for each assembly instruction.

Table 4.10 and 4.11 shows that it requires 4 clock cycles to toggle the GPIO-pin. In order to make it simple, the core clock is set to ten times the SCLK frequency, which is 120 MHz. To toggle the GPIO-pin at 12 MHz, one NOP must be added to the BSRR write. Figure 4.44 shows a waveform of this implementation. The figure shows that one clock cycle at 12 MHz equals 10 clock cycles at 120 MHz. For the first half clock cycle at 12 MHz, the GPIO-pin is high. In order to reset the GPIO-pin and be in sync, a NOP is executed before the write to the BSRR. This ensures that it stays in sync, and after 5 clock cycles at 120 MHz, it is reset. The 120 MHz and 12 MHz periods are approximately 8.33 ns and 83.3 ns, respectively. Listing 4.3 shows the implemented code for the first 8-bits. The 48-bit sequence ends on a zero. In order to keep the data line high after the programming sequence, the line is pulled high with `GPIOA->BSRR =(1«5)`. This is done to comply with the observations of the NanoBerry data line.

Code listing 4.3: The implemented code for the first 8-bits of the 48-bit camera configuration sequence

```
/* [0] ---- 10010000 ---- */
GPIOA->BSRR =(1<<5);
__ASM volatile ("nop");
GPIOA->BSRR =(1<<21);
__ASM volatile ("nop");
GPIOA->BSRR =(1<<21);
__ASM volatile ("nop");
GPIOA->BSRR =(1<<5);
__ASM volatile ("nop");
GPIOA->BSRR =(1<<21);
__ASM volatile ("nop");
GPIOA->BSRR =(1<<21);
__ASM volatile ("nop");
GPIOA->BSRR =(1<<21);
__ASM volatile ("nop");
GPIOA->BSRR =(1<<21);
__ASM volatile ("nop");
```

When we tested this implementation for 8-bits, it did work. However, when all 48-bits of the programming sequence were included, it was totally out of sync with the 12 MHz clock. After much debugging and disassembly analysis, it appeared that the compiler had optimized the code and reduced the number of cycles required for writes to the BSRR. It was then discovered that the execution time was exactly 2.5 times too fast. We then divided 120 MHz by 2.5, which is equal to 48 MHz. When the core frequency was set to 48 MHz, it aligned perfectly with the 12 MHz clock. Exactly why this happened is not entirely clear. From forums, it seems like there are some settings and compiler flags that can be altered to avoid optimization. However, it is ideal to have the core frequency as low as possible to reduce the power consumption in our case.

During the programming and testing, a large number of combinations of the delays and writes to the BSRR was tried out. To program 48-bits in this manner was a very manual, time-consuming task prone to errors. In order to ensure that the code always was correct, an Excel sheet was made that generated the code output with the correct register configuration and the order and number of BSRR writes and delays. This is available in the GitHub repository and is linked to the Excel sheet in Figure 4.42.

Delay

In order to spam the programming sequence, it had to be decided how long the delay between each programming sequence should be. Countless different configurations have been tried out. The only one that resulted in output from the camera module is presented here. However, it is mainly based on trial and error and not theory due to lack of information.

From measurements on the NanoBerry board, the data line uses 126 cycles to stabilize before it transitions to the next state. Exactly why this happens is not known. 126 clock cycles is equal to 10.5 PPs. The total time left in INTERFACE mode is then 648-10.5 PPs = 637.5 PPs. It was used eight programming sequences - starting and ending with a programming sequence. By subtracting the last programming sequence the total number of PPs left for the 7 first is 637.5-4 PPs = 633.5 PPs. This leaves 633.5/7 PPs = 90.5 PPs for a programming sequence and the delay to the beginning of the next programming sequence. This results in a delay of 90.5-4 PPs = 86.5 PPs between each programming sequence. This delay was first implemented using the internal timers of the MCU, but it was not accurate enough. This was solved by using a while loop with NOP. The delay consisted of; the execution time of the main while loop, the timer while loop, the while loop counter, and the NOPs. This delay was fine-tuned using an oscilloscope, to 90.501 µs, which is almost exactly equivalent to 90.5 PPs at 12 MHz.

When the camera is in sync, it uses 106928 per frame. The INTERFACE mode is already accounted for. This leaves 106290.5 PPs, where the 10.5 PPs are included. This delay was implemented using a 106 ms delay with the `HAL_Delay()` function. The remaining 290.5 µs was implemented in the same manner as the delay for the

INTERFACE mode and fine-tuned on the oscilloscope. This is not a sustainable way to implement a synchronization mechanism, as some parameters are not entirely under our control. Thus the exact execution time can vary with a couple of clock cycles from every run, which most likely will cause the clock and data to become unsynchronized rather quickly.

Interrupt

The interrupt mechanism is implemented on a second GPIO-pin PB13. This is connected to PA5, and every time it detects data, it triggers the interrupt and checks a flag set in the main while loop, which dictates if the data came from the MCU. If this is not the case, the data came from the camera, and it will start the 106 ms delay followed by the fine-tuned 290.5 µs delay. The program is then halted for the frame's duration until the state machine arrives at the INTERFACE state. Listing 4.4 shows the pseudo-code for the implementation using interrupts and the delays. The interrupt implementation itself was tested, and it works. However, because of the unstable synchronization mechanism, it was not verified that it works with the camera data.

Code listing 4.4: Pseudo-code for the bit banging implementation, including the delays and interrupt

```
/*---- MAIN WHILE LOOP ----*/
WHILE (TRUE)
    FLAG_Data_Transmitting = TRUE
    Transmit_camera_configuration(48-bit)
    FLAG_Data_Transmitting = FALSE
    Delay(86.5 us)


/*---- INTERRUPT ----*/
IF (FLAG_Data_Transmitting = FALSE)
    Delay(106 ms)
    Delay(290.5 us)
```

4.7.5 Brightness Control Module for the LEDs

In order to control the brightness of the LED, a module using the Pulse-Width Modulation (PWM) technique is made. It takes the period and duty cycle as inputs. The period must be a whole integer of ms, making 1 kHz the lowest frequency possible. The duty cycle is given as a percentage. The duty cycle is a measure of the time the signal is high, given by the following relation

$$\text{Duty Cycle} = \frac{\text{ON time}}{\text{ON time} + \text{OFF time}} \tag{4.22}$$

Chapter 5

Results

The design of the capsule-size prototype was finished and ready to order from ExceptionPCB, but as the interfacing with the camera did not work properly, it was not ordered. The price of the capsule was 25 000 NOK, which is expensive, and we decided to verify the design and programming before ordering it to make sure that it would work. This section includes the verification and results from our custom evaluation board, and the NUCLEO board used to develop and test the code. Due to the lack of a functional capsule-size prototype, the backscatter interfacing was tested in an alternative way. These experiments were performed with Ilangko Balasingham and Ali Khaleghi's research team at NTNU. As a replacement for the capsule-size prototype, we made a dummy capsule with only the antennas. The camera data from the ams NanoBerry evaluation board was fed to the switching circuit, and the output from the switch was fed to the capsule antennas via wires. It was also tested by feeding the backscatter switch with a pre-recorded video stream from a PC, serialized by a MCU. We did not use our software implementations during this test because it was only during the last week of the thesis work we were able to get a stable output from the camera with our code. The tests had already been done at this point, and there was not enough time to repeat the test. However, the results from the tests should be almost identical. Finally, power measurements are performed on the custom evaluation board, which closely resembles the designed capsule-size prototype.

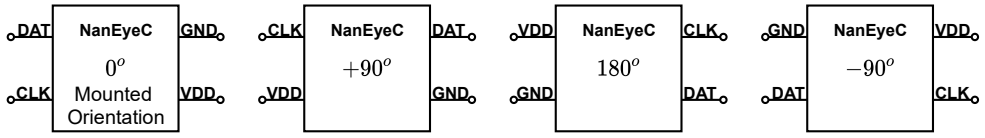


Figure 5.1: The four possible orientations of the NanEyeC camera module. From left to right the camera module is rotated 90° clockwise.

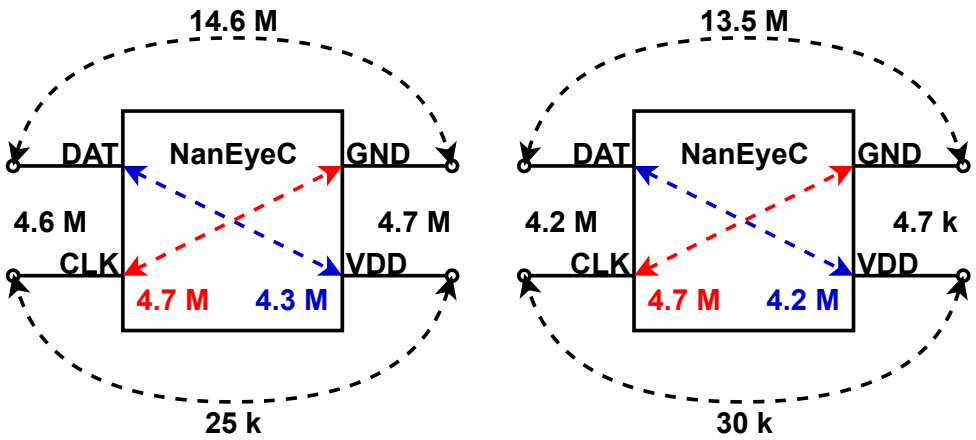
Table 5.1: The measured values for the four possible orientations of the NanEyeC camera module, shown in Figure 5.1. CC means short circuit and is triggered by the power supply when a short circuit is detected.

	V_{pp} [V]		Power Supply		
Rotation	DAT	CLK	V_{out} [V]	A_{out} [mA]	CC
0	2.24	0.77	0.82	13.00	NO
+90	2.13	0.65	0.67	MAX	NO
180	3.30	0.67	3.30	MAX	YES
-90	3.10	1.75	1.27	MAX	YES

5.1 Custom Evaluation Board

The camera on the custom evaluation board did not work and impacted the clock and data in strange ways. The peak to peak voltage of the clock was 3.3 V from the MCU, but when connected to the camera, it dropped to 0.8 V. The same occurred for the data line, which dropped from 3.3 V to 2.2 V. We suspected that the camera was mounted with the wrong rotation. This could be due to a mistake made by Eurocircuit during assembly, or it could be that the ordering data was incorrect due to an error in the NanEyeC library we created based on the NanEyeC datasheet. Another reason could be that Eurocircuit had damaged the camera module or made a short circuit between the pads. To locate the problem, the traces connecting the camera to the rest of the custom evaluation board were cut with a scalpel and wires soldered onto each of the four camera pads. This can been seen in Figure 5.27 in Section 5.4, where the four shortest wires are connected to the camera. Each pair of pins was checked for short circuits, but all of them had large resistance.

The four possible rotations are shown in Figure 5.1. The one to the left is how the camera module is mounted and connected on the evaluation board. The next three illustrate a 90° clockwise rotation compared to the previous one. The four different rotations were connected and measured systematically. Table 5.1 list the measured peak to peak voltage for the clock and data lines. All of the rotations maxed out with a 6 mA current limit, so it was increased to 20 mA for a very short time. Only the mounted orientation did not max out with this configuration, but the current was still way too high. The maximum current should be 3 mA. For the 180° and -90° rotations, the power supply stopped outputting current and indicated a short circuit.

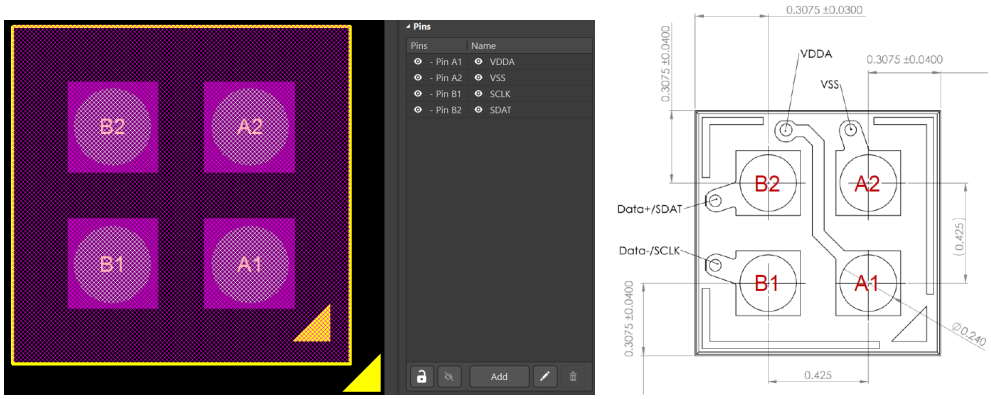


(a) Custom evaluation board **(b)** NanEyeC breakout board

Figure 5.2: Measurements of the resistance [Ω], between each pair of pads for the NanEyeC camera module mounted on the custom evaluation board (a) and the NanEyeC breakout board (b).

From the measurements and the fact that two rotations had short circuits, the most likely case was that the camera module was rotated +90° but had been broken by the use and testing of the custom evaluation board, where the camera module had drawn 13 mA to the wrong ports. When we received the NanEyeC breakout board PCB, with just the camera module on it, we measured the resistance between each port and compared it with the camera module mounted on the custom evaluation board. The measurements are shown in Figure 5.2, which shows that the resistances between the pairs of pads are almost identical. The differences are most likely because the camera module mounted on the custom evaluation board is damaged. Because the measurements verify that the camera modules are mounted the same way on three different PCBs, manufactured by two different PCB manufacturers, it is safe to assume that the source of the error is either the datasheet or the library entry we made for the NanEyeC camera module.

The schematic and footprint from our NanEyeC library were compared to the datasheet, to locate the source of the error. Figure 5.3 shows a side by side view of the two footprints. Figure 5.3a shows our footprint and Figure 5.3b shows the footprint from the datasheet.

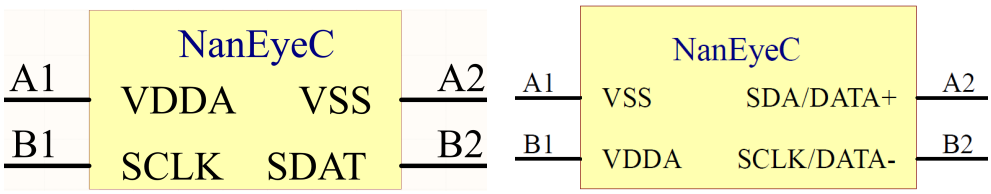


(a) Screenshot of Altium **(b)** NanEyeC datasheet

Figure 5.3: The footprints for NanEyeC. (a) shows a screenshot from altium, where A1 = VDDA, A2 = VSS, B1 = SCLK and B2 = SDAT, which is equivalent to the footprint in the NanEyeC datasheet (b).

At a later stage in the design process, we found some reference schematics for the camera board on the ams NanoBerry evaluation board. This schematic does not comply with our schematic. Figure 5.4 shows a comparison of our schematic and the schematic from the NanoBerry reference schematics. According to the Altium Designer documentation, the placement of the pins in a schematic is independent of the placement in the footprint because the only factor the Altium compiler considers during conversion of schematics to PCB layout is the pin mapping. Compared to the footprint, ams has changed the position of A1 and B2. For comparison's sake, our schematic has been altered to match this. Then it is clearly visible that they do not match. The reason for this is that the reference schematic from ams does not correspond to the footprint shown in Figure 5.3b. Then the pin mapping is not valid, and they must have used a different footprint when making the schematic. They have changed A1 from VDDA to VSS, A2 from VSS to SDAT, B1 from SCLK to VDDA, and B2 from SDAT to SLCK.

Figure 5.5 shows a comparison of the footprint in the datasheet and the footprint ams have used to generate the schematic used in their reference schematics. The color of the boxes belongs to the pin name to illustrate the rotation. The comparison shows that the footprint we have used must be rotated +90°, to comply with the footprint used by ams in their reference schematics. This is consistent with the measurements and observations we did. It does seem like it is a typo in the footprint presented in the ams NanEyeC datasheet.



(a) Our schematic **(b)** ams reference schematic

Figure 5.4: Comparison of our schematic (a), generated using a footprint equivalent to the footprint in the NanEyeC datasheet and the schematic from the ams reference schematics (b) for the NanoBerry evaluation board.

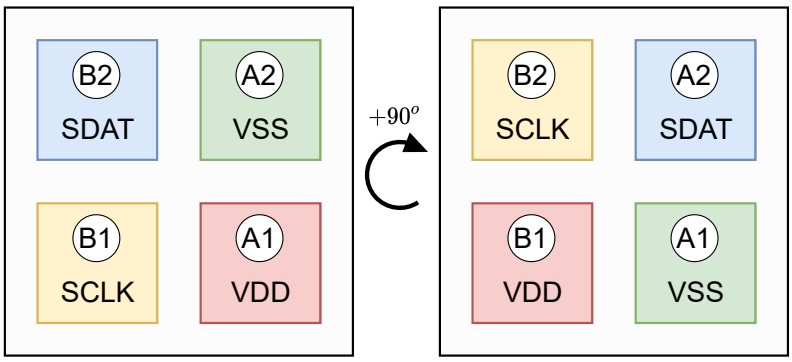


Figure 5.5: The footprint to the left is the footprint from the datasheet. The footprint to the right is the footprint used by ams when generating the NanEyeC schematic used in the ams reference schematics for the NanoBerry evaluation board. The figure shows that the footprint from the datasheet used in our design must be rotated +90° clockwise to comply with the footprint from the ams reference schematics.

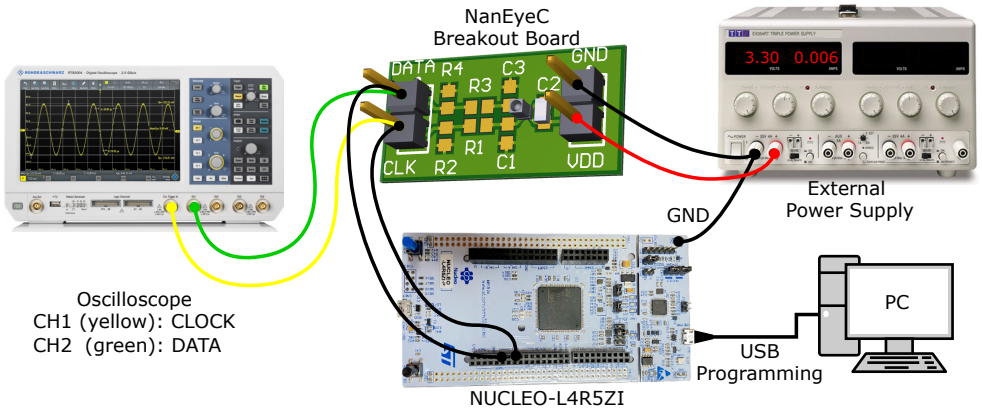


Figure 5.6: Test setup used to measure the execution times of a frame and verify the correct behavior of the camera module. The PC, programs the NUCLEO board, which feeds a clock and programming sequence to the camera and receives the data output from the camera. The camera module is powered directly by an external power supply with 6 mA current limit and 3.30 V output. The oscilloscope measures the clock and data lines using differential probes.

5.2 Capsule-Size Prototype

Figure 5.6 shows the test setup used to measure the execution time of a frame for different implementations and to verify the correct behavior of the camera module. The PC is connected to the NUCLEO-L4R5ZI with a USB cable used to power and program the board. The NanEyeC breakout board is powered directly by an external power supply with 6 mA current limit and 3.30 V output voltage. As discussed in the implementation, the breakout board is used without the four resistors and two capacitors, C1 and C3. Solder bridges replace R1 and R3 to short the pads. The camera module is connected directly to the NUCLEO board, which provides a clock and the programming sequence during INTERFACE mode. The data and clock lines are measured and recorded by the oscilloscope using differential probes. All five devices share a common ground. For all subsequent measurements with the oscilloscope, channel 1 in yellow is the clock and channel 2 in green is the data line.

5.2.1 SPI Method - Without DMA

The interfacing implementation closest to the described SEIM protocol in the theory was SPI with DMA. This was implemented for 12 Mbps transfer rate, using a 24 MHz core frequency - double of the SPI, which is the minimum requirement. We did not get any output from the camera for a very long time. In the last weeks of the thesis work, we discovered that the DMA did not work properly for the SPI implementation. In the startup sequence, the program changes between bit banging and SPI multiple times, and we believe that this might be the root of the

problem. When re-initializing SPI, DMA must also be re-initialized, but it seems like something goes wrong here. We're using the HAL library for both SPI and DMA. This library is complex with several layers, making it hard to follow all dependencies and actions. We have not been able to figure out exactly what goes wrong. However, by rewriting the code to SPI without DMA we got output from the camera. The problem now was that the output was unstable, and it usually just worked for around 20-30 seconds, with the occasional dropouts midway. The execution time of one frame was measured to approximately 308 ms, which is significantly longer than the expected 124.7 ms, where the two clock cycle gap between SPI packages is factored in.

During debugging, we discovered that increasing the core frequency from 24 to 96 MHz fixed the problem. The core frequency is then eight times the SPI bandwidth, and not two. In addition, the camera output was now stable and working as expected. We could not find anything about this in the STM32 documentation, but others reported the same issue on a couple of forums. The answer there was to implement SPI with DMA to fix the problem. They stated that for long transfers, the core is not fast enough to handle all the incoming data before new data arrives, and thus the CPU stalls the output SPI clock by expanding the gap between each SPI package. This does seem to be the case here as well. The extra delay also seems to cause problems with the synchronization with the camera state machine, requiring an occasional reset of the camera module during prolonged operation time.

Table 5.2 shows the measured execution time for 12 and 24 Mbps SPI without DMA with 24, 48, and 96 MHz core frequencies. The corresponding frame rates are also included to illustrate the effect of the execution times. It is interesting to see that the relation between 24 and 48 MHz for 12 Mbps SPI is almost half, which also is true for the relation between 48 and 96 MHz for 24 Mbps SPI. Both are a doubling of the core frequencies. The relation between 96 and 48 MHz for 12 Mbps SPI on the other hand, is only 20 %. It would have been interesting to check for even higher core frequencies, but the NUCLEO board is limited to 120 MHz. For the 24 Mbps SPI implementation, the register values in the programming sequence are changed accordingly.

Out of curiosity and to verify our understanding of how lower frame rates could be achieved, we tried a set of lower SPI frequencies. The NanEyeC datasheet has two contradictory statements. They state that frequency must be within ±10 % of the configured input frequency and that the camera module works for frequencies lower than the configured frequency, with a note; that it may cause strange artifacts to occur. In order to test this, the register values were set for the lowest possible frequency configuration, 12 MHz. We tried with 6, 8, and 10 Mbps SPI. For 6 Mbps, it worked for a short time, but then the external power supply indicated a short circuit, and the camera current went straight to 6 mA, which was the limit set on the power supply. For 8 and 10 Mbps, SPI both indicated a short circuit. When we tried it again the next day, it did work for a little while before the same thing happened again. From these tests, the first statement is verified;

Table 5.2: Measured execution time of one frame at the frequencies 24, 48 and 96 MHz for both 12 and 24 Mbps SPI without DMA. The corresponding frames rates are also included.

	12 Mbps SPI		24 Mbps SPI	
Core clock [MHz]	Time [ms]	FPS	Time [ms]	FPS
96	124.7	8.02	77	12.99
48	154	6.49	154	6.49
24	308	3.25		

the input frequency has to be within the range ±10 % of the configured input frequency, which for 12 MHz is 11.07-13.53 MHz. Thus our understanding of the IDLE mode manipulation is probably correct, where the lower ranges of FPS are achieved by turning on the IDLE mode in the INTERFACE state. The state machine is then halted until the desired execution time is obtained. This means that all frequencies can be used to generate lower FPS.

The SPI method is implemented for both 8- and 12-bit SPI. This gives a different compensation factor for the gap between each SPI package. The calculated execution time for the 12-bit implementation is

$$\frac{14\text{bit}}{12\text{Mbps}} \cdot 106928 \approx 124\,749.333\,\mu\text{s} \approx 124.7ms \tag{5.1}$$

The 8-bit implementation requires 150 % more SPI packages for each frame to compensate for only sending two thirds of a PP with each SPI package. This results in an execution time of

$$\frac{10\text{bit}}{12\text{Mbps}} \cdot 160392 \approx 133\,660\,\mu\text{s} \approx 133.6ms \tag{5.2}$$

Another factor to the total execution time is the RAM size, which dictates the maximum transfer size in a single operation. The L0 MCU mounted on the custom evaluation board has only 8 kB RAM. This forces the READOUT of a frame to be divided into 32 operations for both the 8- and 12-bit implementations. The L4 MCU we planned to use has 64 kB RAM and the L4+ has 512 kB. These are large enough for the frame size but are limited by the 16-bit counter used in the HAL SPI implementation. Thus the 12-bit implementation has to be done in two operations and the 8-bit in three operations. Table 5.3 lists the measured execution times for both small and large RAM size for the 8- and 12-bit implementations. The table shows that the RAM size have an almost negligible effect on the execution time, with only 0.2 ms extra for both implementation.

The measurements were performed on the stable working implementation, 12 Mbps SPI on the NUCLEO-L4R5ZI board running the L4+ core at 96 MHz. This includes accurate timing measurements and verification of the camera input and output data. Figure 5.7 shows a screenshot of the entire frame on the oscilloscope. The cursors measure the execution time of one frame to be approximately 124.8

Table 5.3: The measured execution times for both small and large RAM size for the 8- and 12-bit implementations. The small RAM is for the L0 MCU, while the large includes both the L4 and L4+ MCUs.

	12-bit SPI		8-bit SPI	
Ram Size [kB]	Time [ms]	FPS	Time [ms]	FPS
Large (≥64)	124.7	8.02	133.6	7.49
Small (8)	124.9	8.01	133.8	7.47

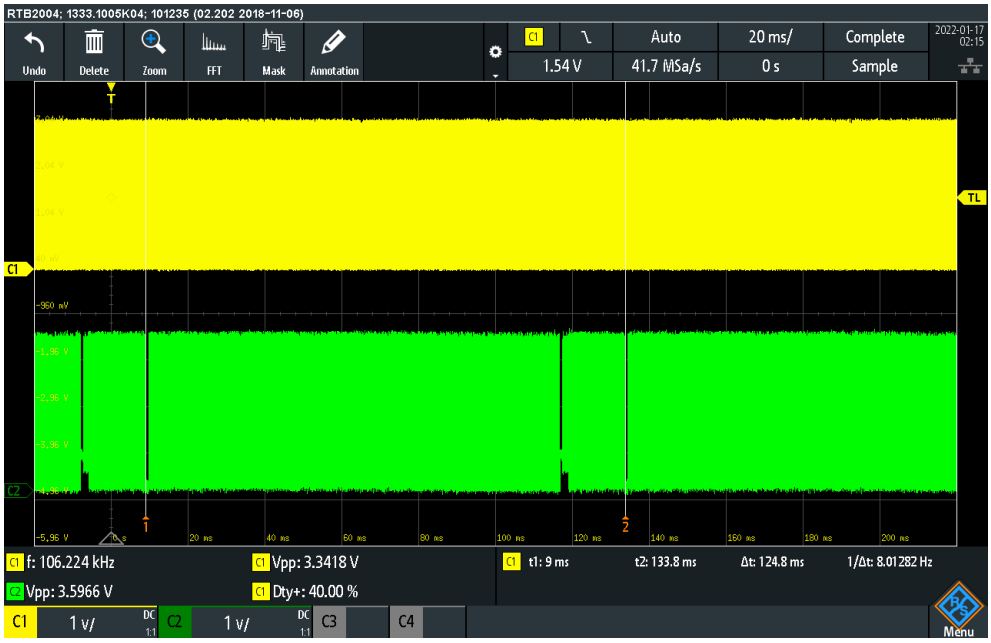


Figure 5.7: Recording of a frame for the 12 Mbps SPI implementation on the L4+ MCU running at 96 MHz. Channel 1 (C1) in yellow shows the SPI clock output from the MCU and channel 2 (C2) in green shows the data transferred between the camera and the MCU.

ms; this is inaccurate and just a visual estimate. According to the documentation, the peak to peak voltages is 3.30 V. The measurements show an average peak to peak voltage of 3.34 V for the clock and 3.60 V for the data. This deviation is primarily due to noise and a low sampling rate. The two gaps to the left of the cursors are always present for all combinations of frequencies and SPI bandwidths. It is not known why this occurs. The output from the ams evaluation board does not have this artifact. The reason might be that our camera has some damaged pixels since it always happens in the same spot.

Table 5.4 shows the accurate measurements done on the frame between the cursors in Figure 5.7. The measurements are done by zooming in on the oscilloscope to 80 ns per square resolution. This gives an accurate measurement. We measure the time at each point of interest and subtract the start point from the

Table 5.4: Accurate measurements done on the frame between the cursors in Figure 5.7. The execution time is measured for each camera state and the delays between each SPI operation. In addition to the measured time the calculated time and the difference between them are included. M-C, means Measured time - Calculated time. The last column shows the number of PP for each camera state.

Camera State and Delay	Measured Time [µs]	Calculated Time [µs]	Difference M-C [µs]	# PPs
INTERFACE: prog. seq.	4.65	4.667	-0.017	4
Delay	6.83			
INTERFACE: empty data	750.594	751.333	-0.739	644
Delay	7.194			
DELAY+SYNC	1 538.574	1 530.667	7.907	1 312
Delay	10.846			
READOUT - Part 1	61 186.217	61 226.667	-40.450	52 480
Delay	10.472			
READOUT - Part 2	61 189.229	61 226.667	-37.438	52 480
Delay	10.241			
END of FRAME	11.662	9.333	2.329	8
Delay	10.595			
Delay time	56.178	0	56.178	
Program time	124 680.926	124 749.333	-68.407	106 928
Total time	124 737.104	124 749.333	-12.229	

endpoint of each camera state or delay. These are listed in the first column of the table. These measurements were repeated for three different frames. The MCU and the camera was reset between each of the three recorded frames. They all coincided, with just a few hundred nanoseconds differences, which is expected with manual measurements like these. The table also lists the calculated times for each camera state. In theory, there is no delay between each SPI operation. However, this is not the case, as can be seen by the measured delays between each SPI operation. The total delay time is approximately 56 µs. The column with the difference between the measured and calculated execution times shows that the READOUT mode is faster in real life and makes the largest impact. When the delay times are considered, the total difference between the expected execution time and measured execution time is only 12.229 µs, where the measured execution time is the fastest.

In order to verify the behavior of the camera module during operation in the main state machine, five screenshots of different levels of detail are included in Figures 5.8-5.12. These are all from the recorded frame shown in Figure 5.7.

Figure 5.8 shows a closer view of the end of the previous frame and the beginning of the frame indicated by cursor 1 in Figure 5.7. The first part is the ending of the last READOUT from the previous frame. Then there is a delay before the END of FRAME sequence, where the camera outputs 8 PPs of only zeros. This looks a bit

strange on the picture because the data line, for some unknown reason, is pulled high at the beginning of END of FRAME and between each 12-bit SPI package. The same behavior is observed for the ams NanoBerry evaluation board. ams might have implemented it like this to distinguish the sequence from the previous and subsequent data. After the next delay, the programming sequence is transmitted from the MCU. Finally, there is another small delay before the second part of the INTERFACE state, called empty data, where 644 PPs of zeros are transmitted.

Figure 5.9 gives a closer view of the programming sequence consisting of 4 PPs and the beginning of empty data in the second part of the INTERFACE state. Right before the MCU starts the SPI operation for empty data, there is always a spike on the data line. This spike is also present for the ams NanoBerry evaluation board. The datasheet does not say anything about this spike. The spike is measured to six clock cycles, which is $\frac{1}{2}$PP.

Figure 5.10 shows a closer view of the READOUT mode, where 16 rows are visible. When the beginning of a new row sequence is sent, it is visible as a short line at the bottom of the green data on the oscilloscope. The cursors are placed on two of these lines, indicating one row. The time is 380 µs, which is correct, as one row is 328 PPs, which results in $328 \cdot \frac{14}{12\text{MHz}} = 382.7\,\mu\text{s}$.

Figure 5.11 zooms in further on 5.10, at the beginning of a row in the READOUT state. This sequence consists of eight PPs. The eight 12-bit SPI packages are placed in between the two cursors. The beginning of a row pattern is the training pattern, which is `001100110011001100110011`, the image confirms this.

Figure 5.12 zooms even further in and the cursors mark the gap between two 12-bit SPI packages. The measured time is 168 ns, which is equivalent to two clock cycles at 12 MHz. The two-clock cycle delay between each SPI package confirms the theory of the datasheet and basically makes each package 14-bit long.

Finally, the startup sequence for both the 8- and 12-bit implementations are verified in Figure 5.13. The channels have been mixed up for this image, so green is the clock, and yellow is the data. The 8- and 12-bit sequences start with one activation clock, register configuration, and 10 clocks after a delay. The last 12 clocks are only valid for the 8-bit implementation. The frequency of the clocks is 1 kHz. The narrow yellow line and the thicker green line below, after the first clock, is the programming sequence sent to the camera for the initial configuration. This is done with SPI at 12 Mbps; hence the execution time is much faster than the 1 kHz clock generation.

Figure 5.8: Closer view of the INTERFACE state. The bulk to the left is the end of the READOUT state from the last frame. The four gaps in the clock (yellow) are delays between SPI operations. The first SPI operation is the END of FRAME sequence, which transmits only zeros for eight PP. It looks alternating because the data line is pulled high between each SPI package. The next operation is the programming sequence to configure the camera registers. Then the next operation is the second part of the INTERFACE state, where only zeros are transmitted. The last part, barely visible, is the beginning of the first READOUT operation.

Figure 5.9: Closer view of the programming sequence and beginning of empty data. The spike that always occurs before empty data is also clearly visible.

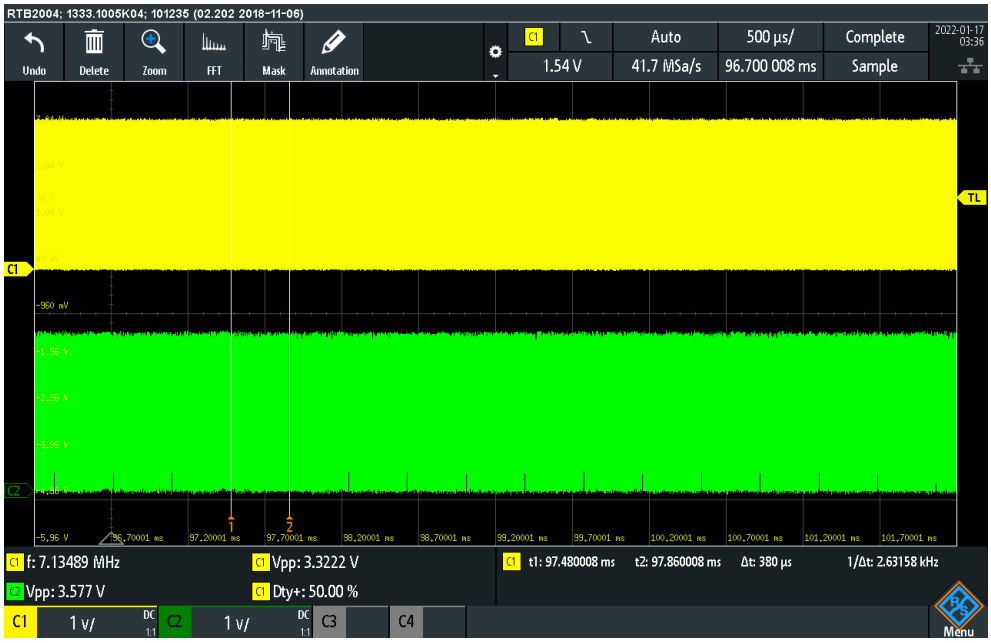


Figure 5.10: Closer view of the a READOUT operation. 16 rows are visible in the picture. Each line at the bottom of the data (green) indicates the start of a new row.

Figure 5.11: Closer view of the beginning of a row in READOUT mode. Shows that there are eight PPs of the training pattern.

Figure 5.12: Closer view of the gap between two SPI packages. The gap between the cursors are measured to 168 ns, which is equivalent to two clock cycles at 12 MHz.

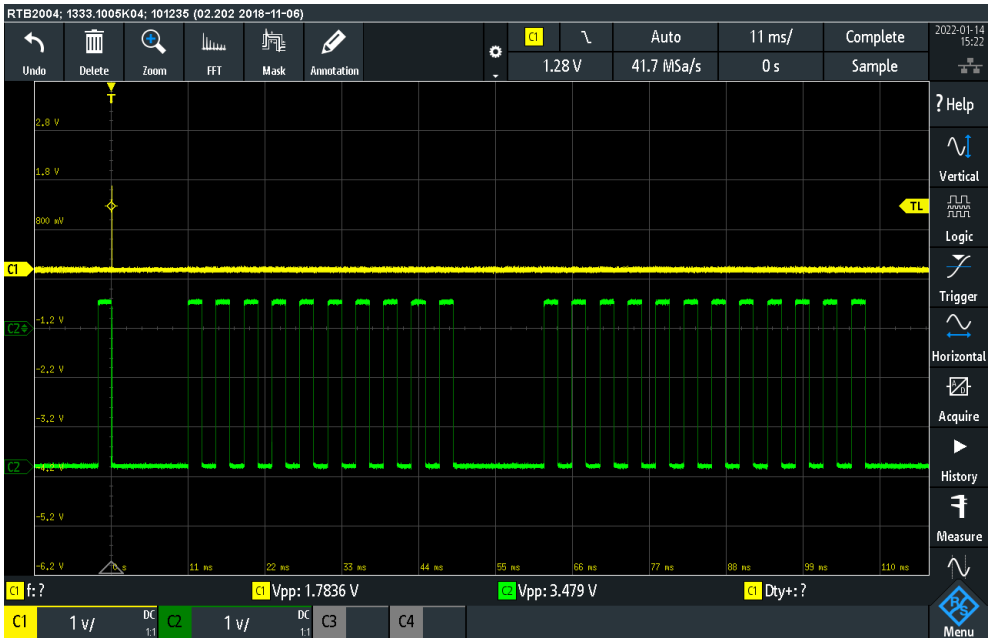


Figure 5.13: NB: the channels have been switched for this image! The clock is shown in green and the data in yellow. Verification of the beginning of the startup sequence. The frequency of the large peaks is 1 kHz. The first peak is the activation clock; then, the narrow yellow line shows the initial programming sequence sent to the camera at 12 Mbps. The subsequent 10 clocks after a delay are used for synchronization for both the 8- and 12-bit implementations. The last 12 clocks are only valid for the 8-bit implementation to compensate for the 12-bit misalignment, as the total number of PPs is odd in the startup sequence.

5.2.2 Alternative Implementation

The alternative implementation is using bit banging to generate the programming sequence from the MCU and the MCUs MCO to generate the 12 MHz clock. The core frequency of the MCU is 48 MHz. This implementation does not work properly. On some rare occasions, we are lucky with the synchronization, and it works for a very short period. It was very hard to test the implementation of the interrupt scheme, as it was very rare to get data from the camera module. The data we have been able to record appears to be corrupt. This is based on comparing the recorded data with the ams NanoBerry evaluation board camera output. The corruption is caused because the interrupt scheme was not used when the data was received. Meaning that the MCU is continuing to transmit the programming sequence every 90.5 µs at the same time as the camera module transmits the output data. The datasheet clearly states that this must be avoided. Figure 5.14 shows a recorded frame from this implementation. From the picture, it is clear that the data does not look correct. However, the different states seemingly have the correct number of PPs. The clock is continuous, so it is much harder to distinguish the different camera states. The INTERFACE state is still easy to distinguish from the rest, and the two cursors in the image are placed at the beginning and end of a frame. The total execution time of a frame is approximately 107 ms. The calculated execution time with a continuous clock at 12 MHz is 106.928 ms. This is also the value stated in the datasheet for SEIM operation at 12 MHz. The image shows that our idea is valid but needs more work to get synchronization in order and make sure that the interrupt scheme works as expected.

Following are six images to verify the behavior of the bit banging implementation. Figure 5.15 shows the 48-bit programming sequence generated using bit banging. The values for each bit in the 4 PPs are placed on top of the image. It is easy to see that all of the values are correct. The programming sequence on the data line perfectly aligns with falling edge of the 12 MHz clock. This is exactly the same as for the ams NanoBerry evaluation board. The total time is measured to approximately 4.001 µs. 48 clock cycles at 12 MHz are equal to $\frac{48}{12\,\text{MHz}} \cdot 4.000\,\mu\text{s}$, which indicates that the 12 MHz clock is very accurate.

Figure 5.16 shows a comparison of the clock generated by the ams NanoBerry evaluation board in green and our 12 MHz MCO clock in yellow. They are almost exactly identical. The oscilloscope measures the frequency to 12.006 MHz, which is quite accurate. The cursors also verify this. The duty cycle is measured to 50.48 %, which is also good, as we ideally want 50.0 %. Figure 5.17 verifies that the gap between the beginning of two subsequent programming sequences are 90.5 µs.

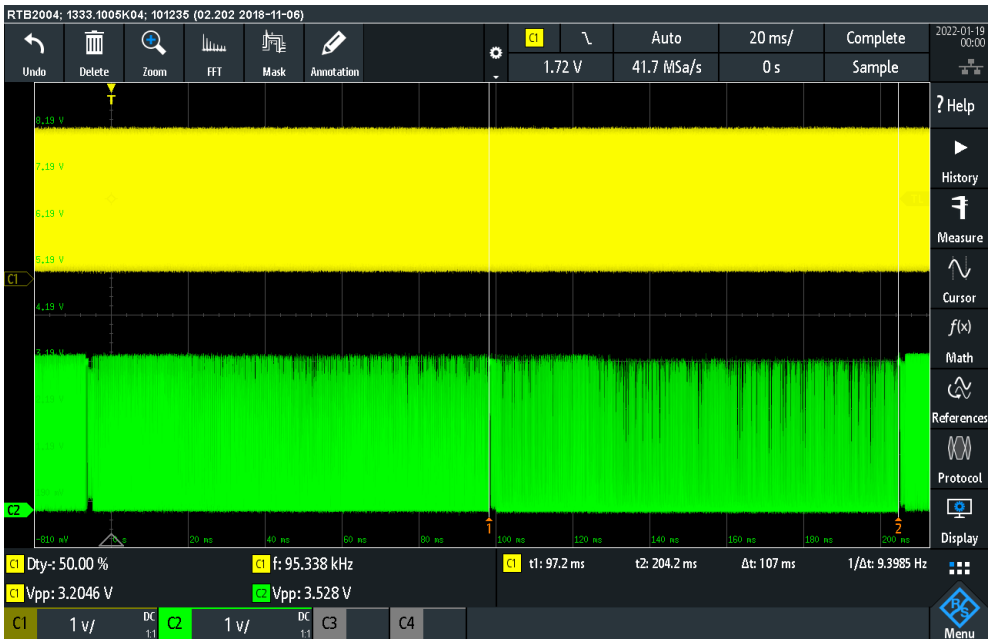


Figure 5.14: Recording of two frames for the alternative method using bit banging and a 12 MHz continuous clock. The data is corrupt due to data being transmitted from the MCU at the same time as it is receiving data from the camera. The image does prove that the idea is valid, and the measured execution time of a frame is 107 ms, which corresponds to the calculated execution time.

Figure 5.15: Verification of the 48-bit programming sequence generated using bit banging. The programming sequence in green is perfectly aligned with the falling edge of the 12 MHz clock. The boxes with white numbers in them placed on top of the image show that the sequence is correct with the register values for 12 MHz SEIM interfacing with the camera module. The total time is measured to approximately 4 µs, which exactly corresponds to 48 clock cycles at 12 MHz.

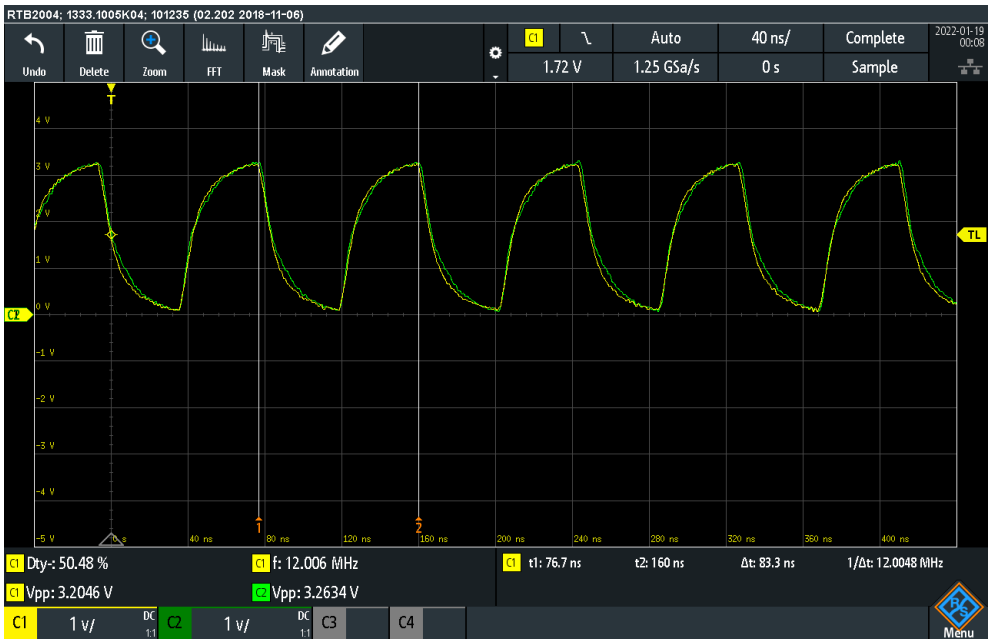


Figure 5.16: Comparison of the clock generated by the ams NanoBerry evaluation board in green and our 12 MHz MCO clock in yellow. They are almost exactly identical. The oscilloscope measures the frequency to 12.006 MHz, which is quite accurate. The cursors also verify this. The duty cycle is measured to 50.48 %, which is also good, as we ideally want 50.0 %.

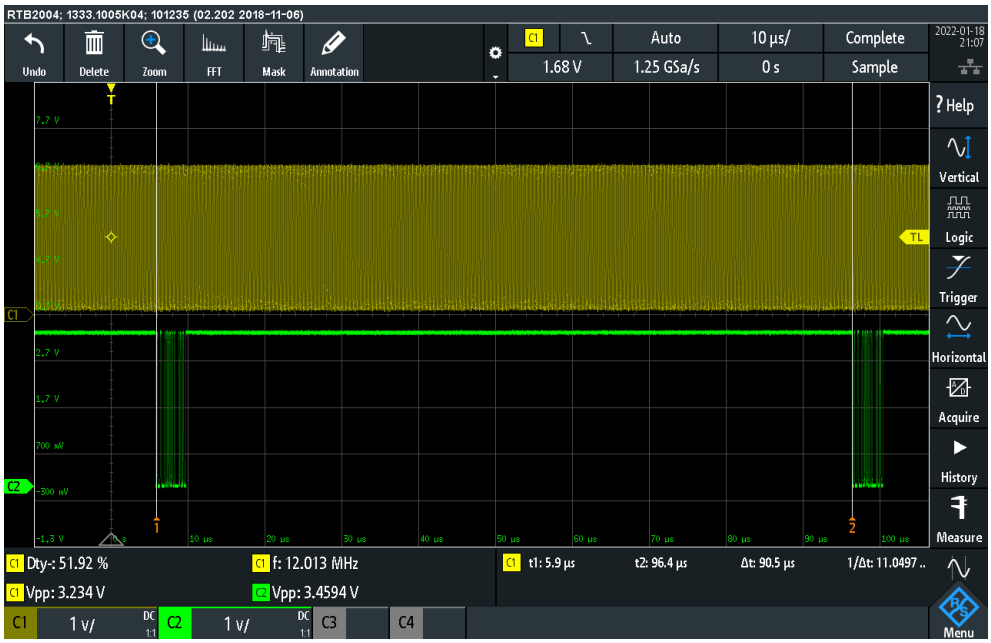


Figure 5.17: The cursors verify that the gap between the beginning of two subsequent programming sequences are 90.5 µs.

We believe that the main issue is the lack of synchronization, which is a common problem for bit banging implementations. Figure 5.18, 5.19 and 5.20 shows the three different alignments between the clock and data, that the program alternates between. From multiple measurements, there are some cases where we had perfect synchronization for a few programming sequences in a row. For this implementation to work, it most likely has to be perfectly in sync exactly at the right time during the INTERFACE state. The core clock of the MCU is 48 MHz, which has a period of approximately 20.8 ns. Figure 5.18 shows that the data is skewed one clock cycle to the left compared to the clock, Figure 5.19 shows perfect alignment and Figure 5.20 shows that the data is skewed one clock cycle to the right. After multiple runs where the programming sequence is sent every 90.5 µs, it is always one of these three alignments. We have not been able to find any pattern between them. We did try to reduce the delay between each programming sequence with one NOP; this only led to everything being shifted one clock cycle to the left, which does not help us.

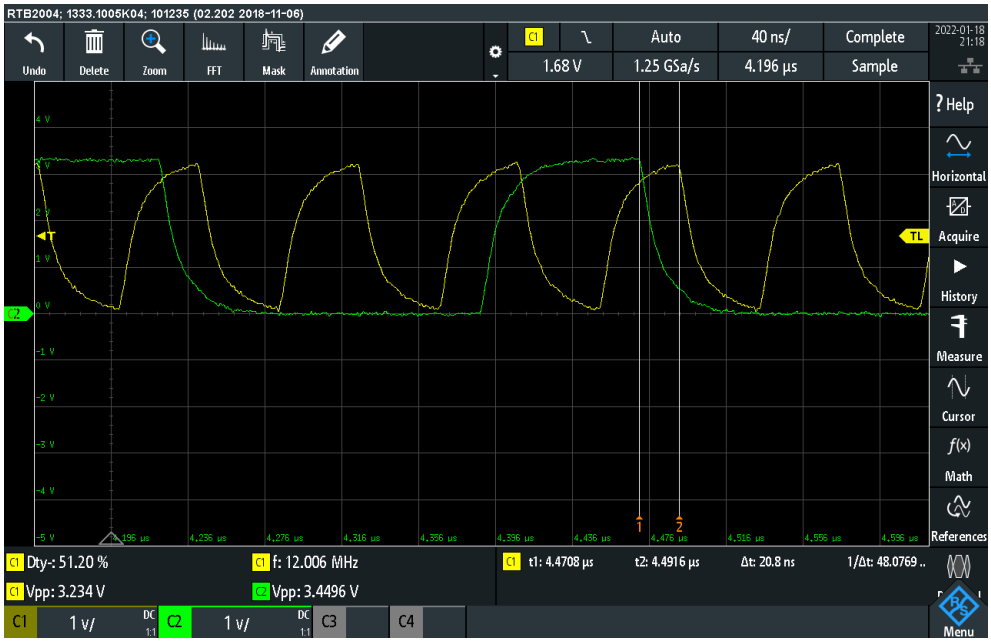


Figure 5.18: Data skewed 1 clock cycle to the left compared to the clock

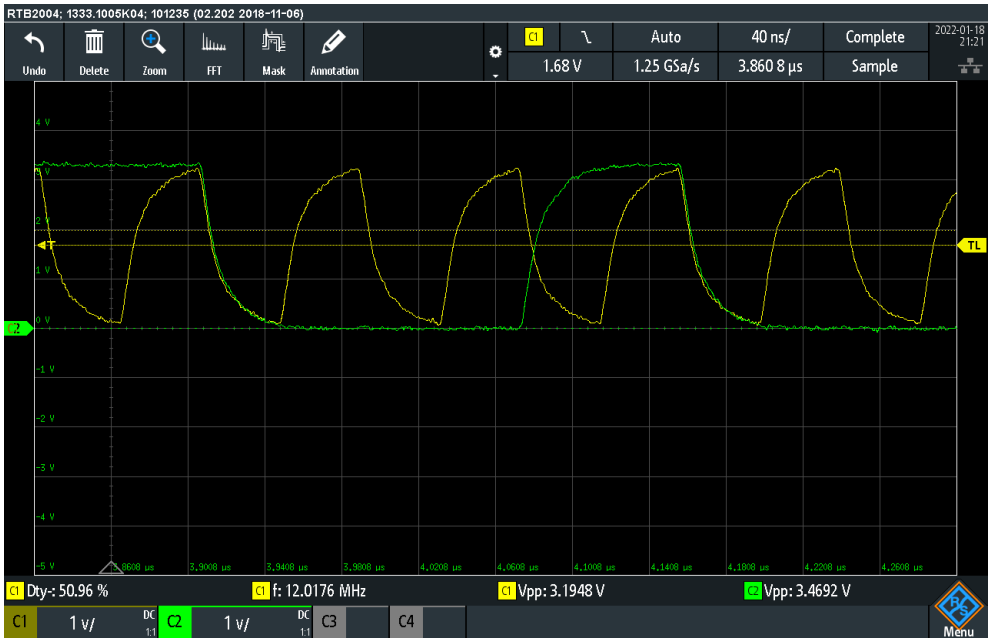


Figure 5.19: Data and clock perfectly aligned

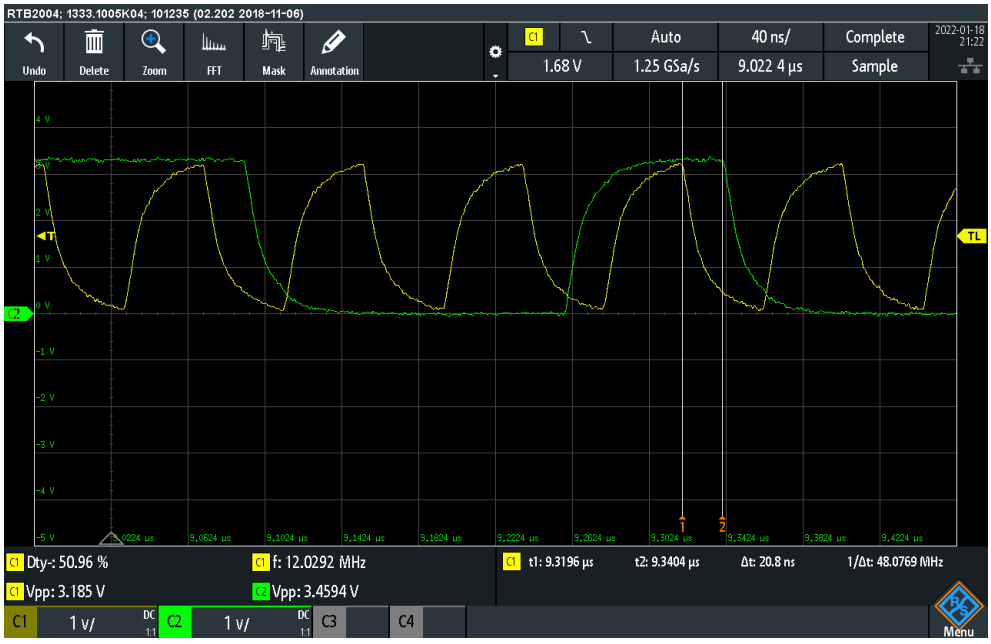


Figure 5.20: Data skewed 1 clock cycle to the right compered to the clock

Figure 5.21: Dummy capsule with antennas made of copper foil and one coaxial cable for external connection to the backscatter switch

5.3 Interfacing NanEyeC with the Backscatter System

Due to the lack of a functional capsule-size prototype, we made a dummy capsule with antennas made of copper foil, as described in Section 2.2.1. The capsule has one coaxial cable for an external connection to the backscatter switch. The coaxial cable shielding was connected to one pad and the core to the other antenna pad. Figure 5.21 shows the dummy capsule, which is ruffly 11 mm in diameter and 26 mm long.

The system was tested with two different sources, supplying image data to the backscatter switch. Figure 5.22a shows the test setup. The video source is fed to the backscatter switch, connected to the dummy capsule via the coaxial cable. The capsule is placed in a bucket containing a phantom, simulating the environment inside the human body. On the outside wall of the bucket, a RX and TX antenna are placed. The backscatter reader transmits the carrier wave to the capsule via the TX antenna, and the backscattered wave is received by the RX antenna. This antenna has an on-board LNA. The amplified signal is fed into the backscatter reader. The recovered clock and data from the reader are sent to an FPGA at a rate of 12 Mbps. The FPGA processes the received data and sends the video stream at a rate of 12 Mbps over a LAN connection to a PC, where the video stream is displayed. The system was tested with two different video sources: real-time data and a recording. The real-time data was the output data from the NanoBerry evaluation board, provided by ams. The recording was provided by a PC that sent the video stream to an STM32F746ZG MCU on an STM32 NUCLEO development board, which serialized and encoded the video stream. Figure 5.22b and 5.22c shows the two video sources, respectively.

Two different phantoms were used in testing: minced meat and a phantom liquid made by a combination of water, salt, and 1,2-Propanediol. The two phantoms in the bucket are shown in Figure 5.23a and 5.23b, respectively. The dummy capsule is placed in the center of the bucket, in the vertical center of the two an-

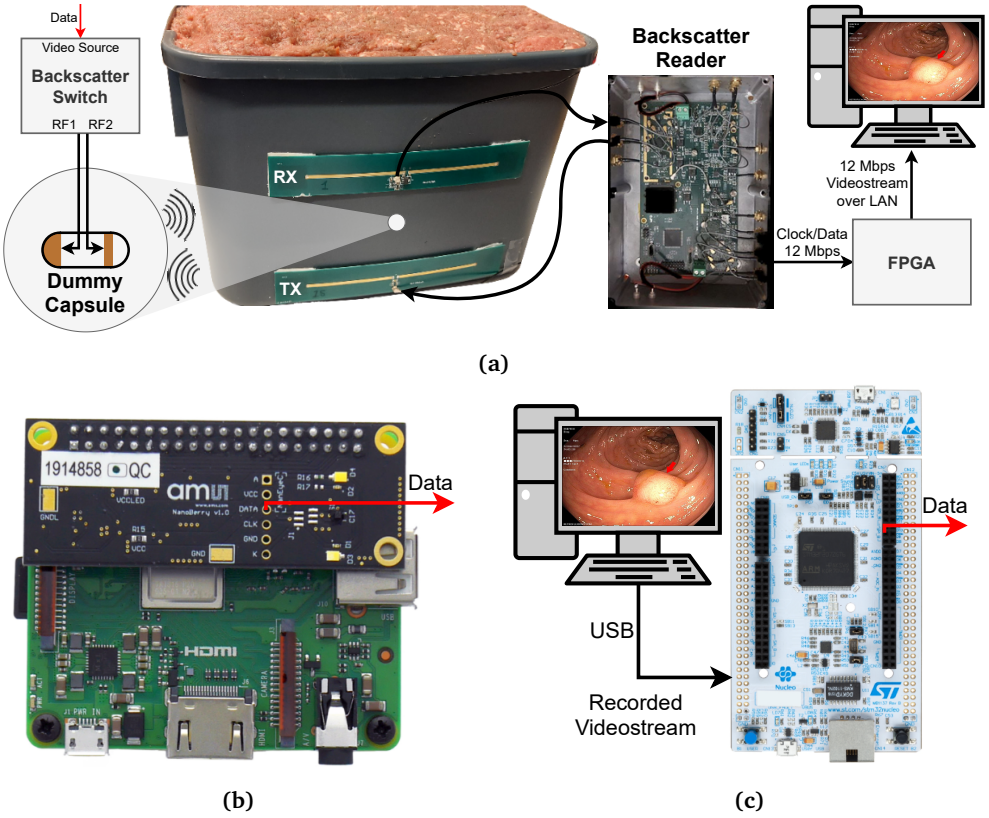


Figure 5.22: Block diagram of the test setup (a), including the two different video sources used for testing. (b) shows the ams NanoBerry evaluation board, providing real time image data. (c) shows the system feeding the recording to the backscatter switch. This includes a PC with the recorded video and a STM32 NUCLEO-F746ZG development board for serialization and encoding of the recorded image data.

Figure 5.23: The bucket with the phantom materials used for testing. (a) shows the bucket with minced meat and (b) shows the bucket with the liquid phantom. (c) shows that the capsule is places in the center of the bucket, approximately 9 cm from the antennas.

tennas - illustrated by the white circle in Figure 5.22a. Figure 5.23c shows that the capsule is placed approximately 9 cm from the antennas.

The spectrum of the backscattered video signal is measured with a Rohde & Schwarz FSH8 Spectrum Analyzer with a range of 100 kHz to 8 GHz. The measured spectrum is shown in Figure 5.24. The image shows a spectrum with a 20 MHz span, where the carrier wave frequency at 434 MHz is a clear peak in the middle at -13.5 dBm. From the spectrum, we can also see that there are no side lobes and that the noise is significantly lower than the backscattered signal.

For the pre-recorded video source, the backscattered video stream was displayed on the PC. The NanEyeC video stream was not displayed as the post-processing and decoding algorithm for the NanEyeC output had not yet been developed by the backscatter team. The recorded output from the backscatter reader was compared to the input for verification. To evaluate the quality of the backscat-

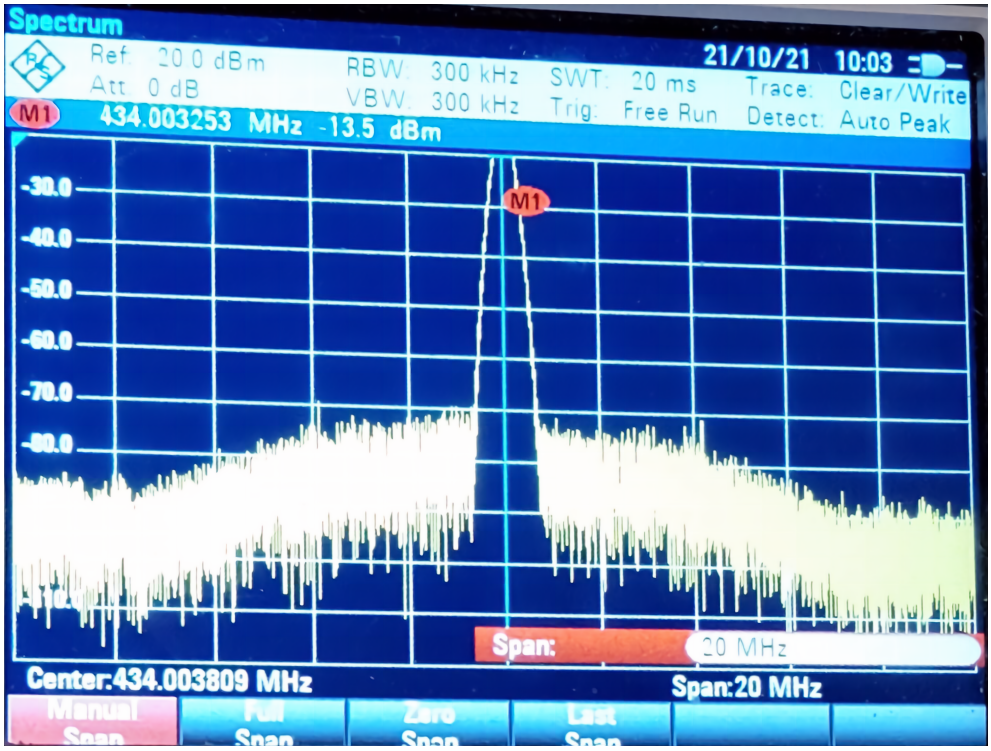


Figure 5.24: Spectrum of the backscattered video data

tered signal, we use the eye diagrams recorded by the oscilloscope. Eye diagrams are commonly used to evaluate the effects of dispersion, channel noise, and intersymbol interference in telecommunication. The eye-opening size indicates how easy it is to distinguish between 1's and 0's. Figure 5.25 shows the measurements done during the experiment with real-time data, using the NanoBerry evaluation board as the source. Channel 1 in yellow and channel 2 in blue show the eye diagrams for the recovered clock and data, respectively. The eyes are almost closed for both data and clock, indicating a poor signal quality prone to errors. Still, the backscatter reader is able to recover the clock, shown on channel 3 in pink.

Figure 5.26 shows the eye diagrams recorded during the experiment with recorded data. Figure 5.26a and 5.26b shows the eye diagrams for the recovered clock and data respectively. These signals have high quality, and the eyes are almost entirely open. The difference in the eye diagrams is likely due to the different encoding techniques used for the real-time and recorded data. The recorded data is encoded specifically to work well with the backscatter system.

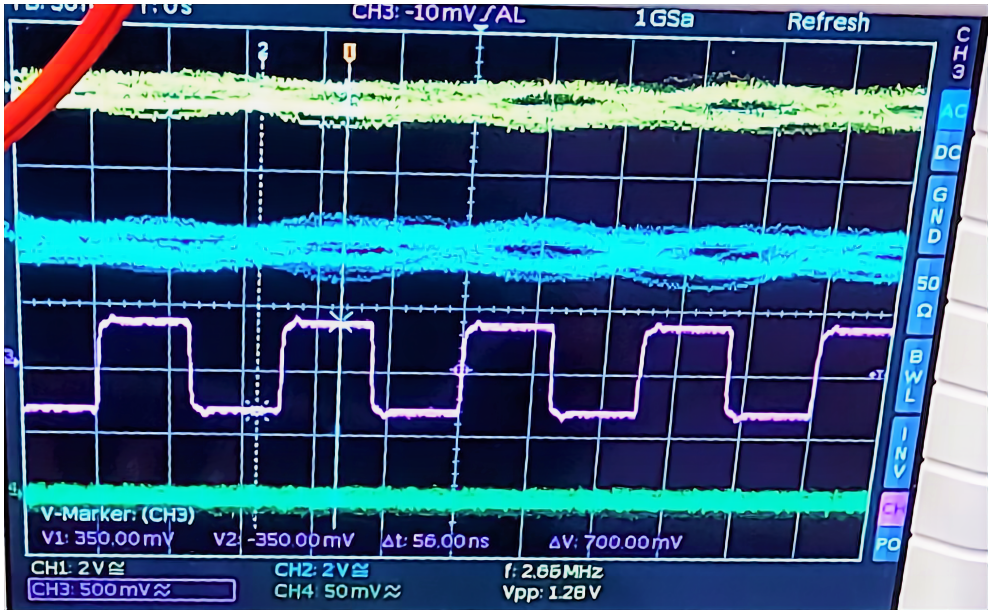
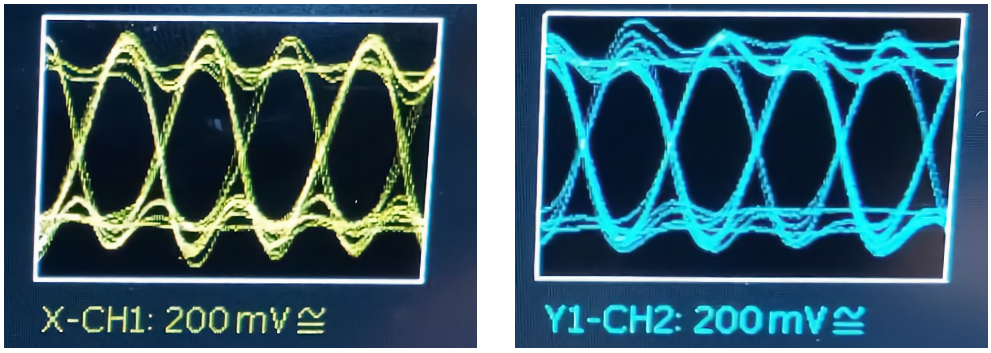


Figure 5.25: Oscilloscope measurements for the real time data. The eye diagram for the recovered clock and data are shown in yellow and blue, respectively. Channel 3 in pink shows the recovered clock.



(a) **(b)**

Figure 5.26: Eye diagrams for the recorded data. (a) and (b) shows the eye diagrams for the recovered clock and data, respectively.

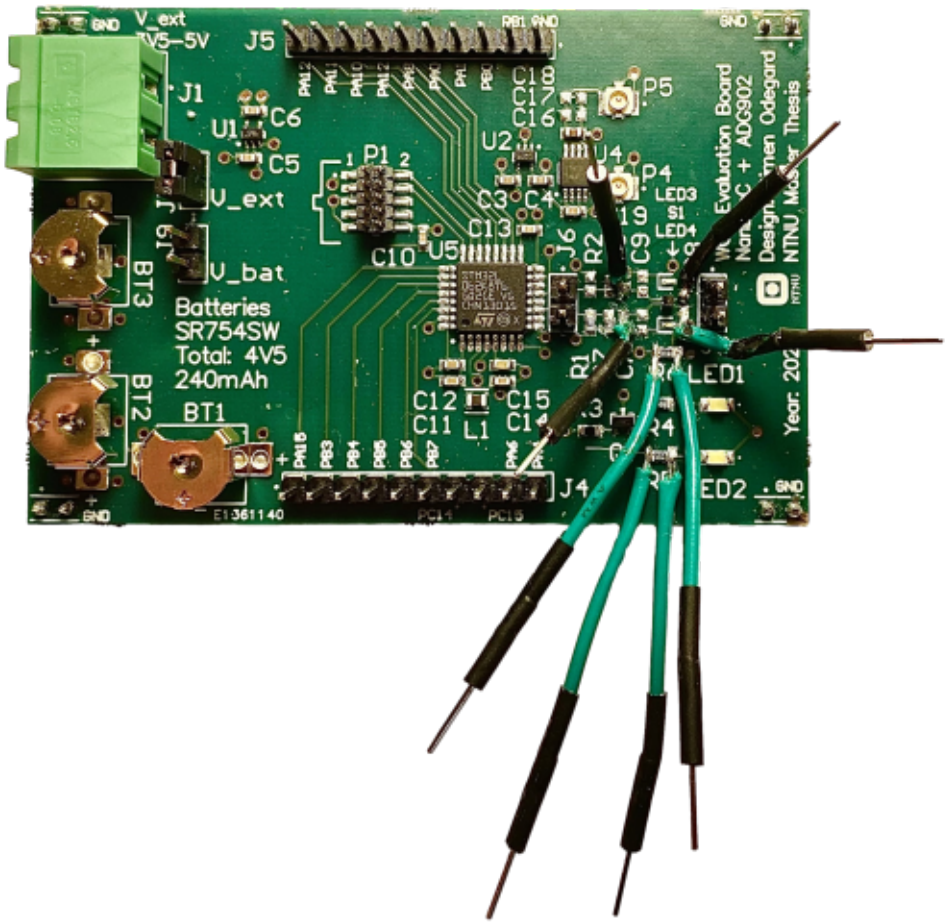


Figure 5.27: The modified custom evaluation board, used for power measurements

5.4 Power Consumption

The power consumption of the camera module, MCU, backscatter switch, LEDs and LDOs are measured and comprised to a table with an overview of the total power consumption. The power measurements are done on the custom evaluation board, which contains everything that will be part of the capsule-size prototype. The custom evaluation board was modified to get more accurate measurements. For the LEDs, wires were soldered onto the resistors to remove noise caused by poor connections during measurements. The camera module had already been isolated during the debugging phase. Figure 5.27 shows an image of the modified evaluation board.

The current consumption of the components is relatively low, which sets high requirements for the accuracy of the measuring equipment. This was a major challenge as we did not have access to equipment with this high level of accuracy. The

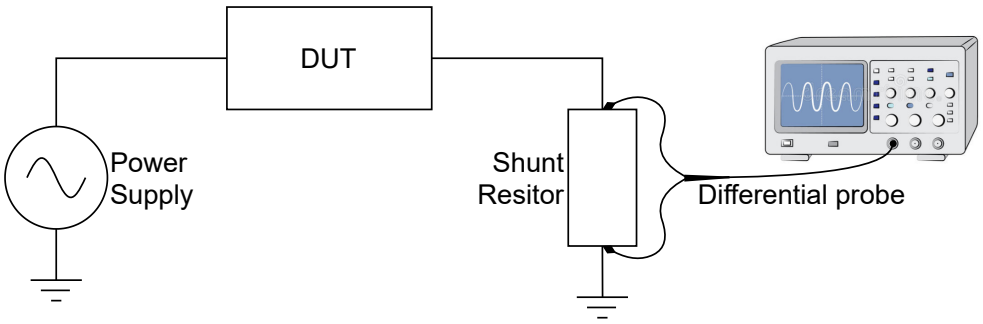


Figure 5.28: The measuring setup used to indirectly measure the current consumption of the DUT, using a shunt resistor and oscilloscope with differential probes.

most accurate equipment we had access to was a Rohde & Schwarz RTB2004 2.5 GSa/s Digital Oscilloscope, a DM-441B True RMS Digital Multimeter, and an Aim TTi Ex354RT Triple Power Supply 300 W, which has a digital display showing the current drawn by the connected device. Unfortunately, the oscilloscope did not have a current probe which is the recommended way of measuring small currents [139]. Thus we opted to measure the current indirectly by measuring the voltage drop across a shunt resistor.

Several factors come into play using the shunt resistor method. A large resistor results in a high voltage drop across the resistor, leading to more accurate measurements. However, this additional power drop must be taken into account, and the resistors' power dissipation increases with the square of the current. The extra resistor will also increase the inductive reactance. Another factor is that the differential oscilloscope has a capacitance connected in parallel with the shunt resistor, effectively making an RC filter. When all of these factors are taken into account, the resistor should be as small as possible. However, it must be large enough to remain within the accuracy of the oscilloscope. Typical values for shunt resistors are in the mΩ range [140]. An additional consideration is the placement of the shunt resistor in the circuit relative to the Device Under Test (DUT). By placing it as close to the ground as possible, the effect of common-mode signals is minimized. In order to limit the system noise, the oscilloscope sampling rate/bandwidth is limited, and x1 probe attenuation is used. Figure 5.28 illustrates the measuring setup for the shunt resistor method, where the shunt resistor is placed close to the ground after the DUT.

The multimeter DM-441B has only a two decimal accuracy, so a small mΩ could not be measured accurately. We opted to use a resistor in the 100-1000 mΩ range to get it as accurate as possible. The resistor was measured to $0.17\,\Omega = 170\,\mathrm{m}\Omega$. After measuring the voltage drop, the corresponding current is calculated by Ohm's law

$$I = \frac{V}{R_{shunt}} = \frac{V}{170\,\mathrm{m}\Omega}, \tag{5.3}$$

where the measured voltage is inserted for V.

Table 5.5: Current and power consumption of the camera module NanEyeC operation in IDLE and SEIM mode. The table includes the measured values and the values specified in the datasheet for comparison.

Mode	Current consumption [mA]		Power consumption [mW]	
	Datasheet	Measured	Datasheet	Measured
IDLE	0.97	0.90	3.2	2.97
SEIM	2.94	2.85	9.7	9.41

5.4.1 Camera Module - NanEyeC

The current consumption of the camera module NanEyeC is measured for the two operation modes IDLE and SEIM. Despite all the considerations made in the shunt resistor setup, the measured voltage jumped between a range of values. We took multiple measurements and calculated the average voltage level to compensate for this and approximated the current consumption as accurately as possible. The averages for IDLE and SEIM mode are approximately 153 µV and 485 µV, respectively. The supply voltage to the camera is 3.30 V. The corresponding current and power consumption are summarized in Table 5.5. The table also includes the power and current rating from the datasheet. In SEIM mode, the datasheet only states the power consumption for 30 MHz input frequency. This implementation operates at 12 MHz. It is unknown how the input frequency affects the camera module's power consumption, but it is reasonable to assume that lowering the frequency would lower the power consumption. Additionally, the camera is powered directly from the power supply - the digital display shows 1 mA for IDLE mode and changes between 2 and 3 mA for SEIM mode.

5.4.2 MCU

The measurements were performed on the STM32L062K8T6 MCU mounted on the custom evaluation board with 24 MHz core frequency. During measurements, all LEDs are disabled, the on-board camera module is disconnected, and the external camera PCB is used. The camera module is powered by an external power supply. Ideally, we should have measured the power consumption of the L4 MCU running at different frequencies as well. However, we did not receive this in time. In order to compensate for this, power simulations are performed in the built-in power analyzer in STM32CubeIDE. The power analyzer collects the user configurations from the project and lets the user perform a power analysis based on several parameters. During development, we used an STM32L4R5ZI MCU. It is not possible to measure the power used by this MCU, as it is embedded on a NUCLEO development board, which contains several other components that cannot be deactivated. This was included in the simulation for comparison purposes and to show the power consumption of the 96 MHz implementation.

In the simulations, the frequency was set to multiples of 12, which is the input frequency used for the camera module. We simulated the three different MCUs:

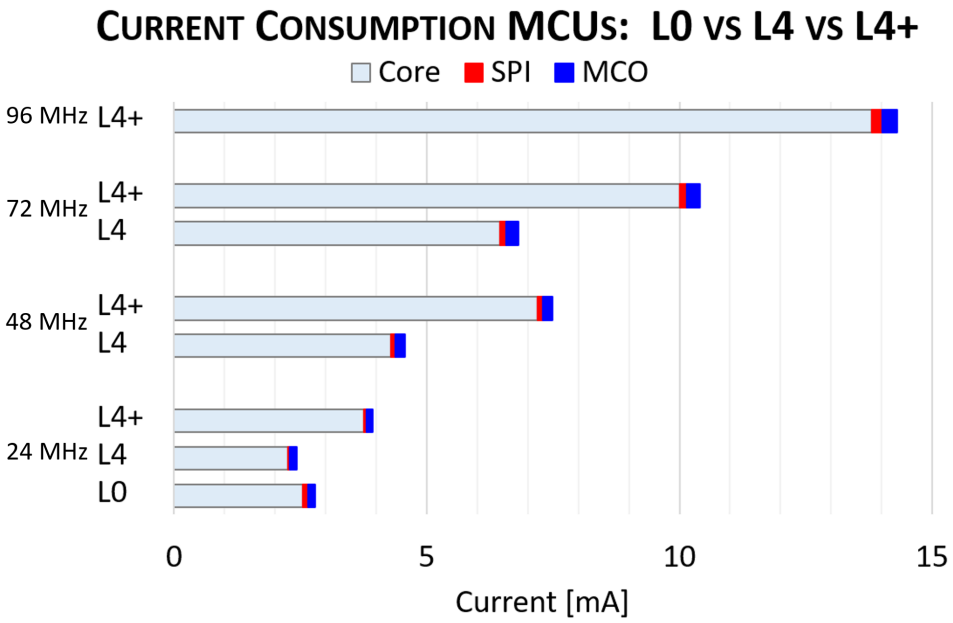


Figure 5.29: Simulation of the current consumption for STM32L062K8T6 (L0), STM32L412K8U (L4), and STM32L4R5ZI (L4+) MCUs. The current consumption of the two implemented solutions SPI (without DMA) and bit banging are shown in red and blue, respectively. The grey bar is the current consumption of the core.

STM32L062K8T6, STM32L412K8U, and STM32L4R5ZI. These belong to the L0, L4, and L4+ series, respectively. We simulated both SPI (without DMA) and bit banging. Where bit banging uses the MCO output to generate clock and toggling one GPIO pin to generate data. Figure 5.29 shows a summary of the results from the simulations for 24, 48, 72, and 96 MHz. The MCUs L0, L4, and L4+ have maximum frequencies of 120, 80, and 32 MHz, respectively. The bars consist of the current consumed by the core in grey, SPI in red and MCO in blue. It is worth noting that the total is the core + either SPI or MCO, and not both. With the final implementation in mind, the power consumption for SPI with DMA was also simulated for the L4 MCU. This showed that the DMA adds very little to the total consumption. At 24 MHz, the use of DMA only adds 0.1 mW to the total power consumption. The power consumption for each simulated case, presuming a 3.3 V power supply, is summarized in Table 5.6.

Table 5.7 shows the measured current consumption of the STM32L062K8T6 MCU, and the corresponding power consumption at 3.3 V supply voltage. The values are measured for three different operation modes: IDLE, core running, and SPI. IDLE is when the MCU is on but not running a program. Core running is just the core running an empty while loop. SPI is when the MCU is interfacing over SPI with the camera module and continuously receives or transmits data. The bit banging implementation requires a 48 MHz clock and could not run on the L0 MCU, which is limited to 32 MHz.

Table 5.6: The simulated power consumption of MCUs L0, L4, and L4+ for frequencies at multiples of 12. The power consumption of both methods are included: SPI (without DMA) and bit banging.

Core Frequency	SPI - Power [mW]			Bit banging - Power [mW]		
[MHz]	L0	L4	L4+	L0	L4	L4+
24	13.2	7.6	12.5	13.3	7.9	12.8
48		14.5	24.1		14.8	24.4
72		21.7	33.5		22.1	33.8
96			46.2			46.5

Table 5.7: Measured current consumption of the STM32L062K8T6 MCU with 24 MHz core frequency. The current is measures for the operation modes IDLE, Core running and SPI.

Operation Mode	Current [mA]	Power [mW]
IDLE	0.8	2.7
Core running	3.8	12.6
SPI	3.9	13.0

5.4.3 Backscatter Switch

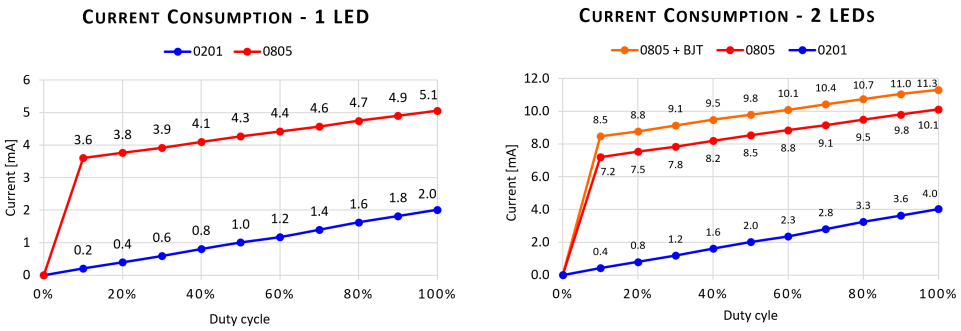
The current consumption of the backscatter was not possible to measure with the equipment we had access to; the measured voltage was below the oscilloscopes noise floor, so there was no change on the oscilloscope when the voltage was measured compared to not measuring anything. From the datasheet, the typical current is 0.1 µA. The supply voltage is 2.5 V. This results in typical power consumption of 0.25 µW or 250 nW.

When trying to do measurements, it still worked when the power supply was disconnected, and only the input signal was connected to the CTRL-port. It might get enough power to run the internals by just the input signal. However, the quality and stability of the switch performance were not tested in this condition.

5.4.4 LED

Two pairs of LEDs are implemented on the custom evaluation board. Two 0201 LEDs are driven directly by a GPIO-pin on the MCU. The other pair is two 0805 LEDs driven by a BJT transistor, where a GPIO-pin is used to control the output current of the transistor.

The measurements are done using the PWM module that controls the brightness of the LEDs by adjusting the duty cycle, see Section 4.7.5. Measurements are done at intervals of 10 from 0 to 100 % duty cycle. The frequency of the PWM is set to 20 kHz. Figure 5.30a shows the current consumption of one of each LED. The 0201 and 0805 LEDs are shown in blue and red, respectively. The figure shows that the 0805 has a larger threshold voltage. After reaching the threshold, both LEDs have a linear current consumption with respect to the duty cycle. From 10



(a) **(b)**

Figure 5.30: Current consumption of the 0201 and 0805 LEDs for different brightnesses. (a) shows the current consumption of a single LED of both types. The 0201 LED is shown in blue and the 0805 LED is shown in red. (b) shows the total current consumption for a pair of each of the LED types. The orange line includes power dissipation from the BJT transistor used to power and control the 0805 LEDs.

to 100 % the difference is 1.5 V for the 0805 LED and 1.8 V for the 0201 LED. A major difference between the two LEDs is that using 50 % duty cycle will save 50 % of the power consumption for the 0201 LED, while it will only save 16 % for the 0805 LED. To further prove this point, four 0805 LEDs is approximately equivalent to ten 0201 LEDs. At 3.3 V supply voltage, the power consumption is approximately 20 mW. Reducing the duty cycle to 50 % would lead to a 10 mW power saving for the 0201 LEDs, while the 0805 LEDs would only save 3.2 mW.

Figure 5.30b shows the total current consumption of two LEDs of each type including the drivers. The 0201 do not have additional hardware, but the 0805 uses a BJT transistor. This transistor has a power dissipation dependending on the output current. The total current consumed by the two 0805 LEDs and the BJT transistor is measured and is plotted in orange. The average gap between the red and orange graph is 1.25 mA. This is equivalent to approximately 4.13 mW power dissipation. The calculated power dissipation was 5 mW.

5.4.5 Total

In the composition of the total power consumption, we have considered the four cases that have been discussed: SPI without DMA running on the L4+ MCU at 96 MHz, Bit banging on the L4 MCU, with core frequency 48 MHz, SPI with DMA on the L0 MCU running at 24 MHz, and SPI with DMA on the L4 MCU running at 24 MHz. The first case is not viable to implement in a capsule design but is included for comparison purposes and the fact that it is the implementation that is working. The three other implementations are viable to implement in the capsule-size prototype. For the camera module NanEyeC the measured power during SEIM operation is used. For the MCUs the simulated values are used to give a fair com-

parison for the L0 MCU, as this was the only MCU where we could do power measurements. The power consumption of the backscatter switch is taken from the datasheet, and the LDO power dissipation is calculated using Equation 3.5.

A major factor to the total power consumption is the LEDs. Three different LED configurations are considered: two 0201 LEDs, four 0201 LEDs, and four 0805 LEDs - which is approximately equivalent to ten 0201 LEDs. Each of these LED configurations have a table each; Table 5.8, 5.9, and 5.10, respectively. Finally the best implementations from each LED configuration are extracted to a single table, which includes the impact of each part on the total power consumption, given as a percentage. These are presented in Table 5.11, 5.12 and 5.13, respectively. The best implementation is SPI with DMA running on the L4 MCU at 24 MHz core frequency for all configurations.

Table 5.8 gives the power consumption for the LED configuration with two 0201 LEDs. The table compares only the three solutions viable to be implemented in the designed capsule-size prototype, which have two 0201 LEDs. At the bottom of the table, the battery lifetime, LDO temperature increase, and the total LDO temperature is included. In order to find the total temperature of the LDO, the ambient temperature must be added. This will depend on the working environment. As the capsule will be working inside the body - it is assumed that the temperature on average is 37°C.

Table 5.9 and 5.10 includes all four implementations. These tables including Table 5.8 shows the great impact of the MCUs core frequencies, the LEDs and especially the LDO. The extreme impact of the LDO is better illustrated in Table 5.11, 5.12 and 5.13. It's worth noting that for all of the cases the LDO always consumes 57 % of the total power. This is due to the nature of the LDO power dissipation equation, where the power dissipation is proportional to the output current, as the voltage difference between the output and input voltage is constant. In our case the voltage difference is $3 \cdot 1.55\,\mathrm{V} - 3.3\,\mathrm{V} = 1.35\,\mathrm{V}$. The efficiency of the LDO is measured for the best case scenario in Table 5.11, which only have two LEDs. I_{GND} is measured to be approximately 40 µA for the LDO, which includes the quiescent current and leakage currents. The measurement was done on a stand alone TLV717P LDO with two adjustable resistors to simulate the correct current. The value is the average of multiple measurements. The efficiency of the LDO is then calculated using Equation 3.7. The efficiency of the LDO that will be used in the capsule-size prototype is approximately 71 %. From these tables, we also see that the backscatter switch and LDO has next to zero impact and is negligible compared to the rest.

Table 5.8: Power consumption of the most viable solutions to be implemented in the capsule-size prototype, with two 0201 LEDs. Including battery lifetime and LDO temperature.

	Power [mW]		
	Bit banging L4 (48 MHz)	SPI w/DMA L0 (24 MHz)	SPI w/DMA L4 (24 MHz)
Camera (SEIM)	9.4	9.4	9.4
MCU	24.4	13.2	7.7
Backscatter switch	0.00025	0.00025	0.00025
Backscatter LDO	0.0002	0.0002	0.0002
LED 2 (0201)	13.20	13.20	13.20
main LDO	63.5	48.3	40.8
Total	110.5	84.2	71.0
Battery lifetime [h]	2.17	2.85	3.38
LDO temp increase [°C]	32.0	26.0	23.0
LDO total temp [°C]	69.0	63.0	60.0

Table 5.9: Total power consumption and battery lifetime of the discussed implementations, with four 0201 LEDs.

	Power [mW]			
	SPI L4+ (96 MHz)	Bit banging L4 (48 MHz)	SPI w/DMA L0 (24 MHz)	SPI w/DMA L4 (24 MHz)
Camera (SEIM)	9.4	9.4	9.4	9.4
MCU	46.2	24.4	13.2	7.7
Backscatter switch	0.00025	0.00025	0.00025	0.00025
Backscatter LDO	0.0002	0.0002	0.0002	0.0002
LED 4 (0201)	26.40	26.40	26.40	26.40
main LDO	110.7	81.3	66.2	58.6
Total	192.7	141.5	115.2	102.0
Battery lifetime [h]	1.25	1.70	2.08	2.35

Table 5.10: Total power consumption and battery lifetime of the discussed implementations, with four 0805 / ten 0201 LEDs.

	Power [mW]			
	SPI L4+ (96 MHz)	Bit banging L4 (48 MHz)	SPI w/DMA L0 (24 MHz)	SPI w/DMA L4 (24 MHz)
Camera (SEIM)	9.4	9.4	9.4	9.4
MCU	46.2	24.4	13.2	7.7
Backscatter switch	0.00025	0.00025	0.00025	0.00025
Backscatter LDO	0.0002	0.0002	0.0002	0.0002
LED 4 (0805)	66.00	66.00	66.00	66.00
main LDO	164.2	134.7	119.6	112.1
Total	285.8	234.6	208.2	195.1
Battery lifetime [h]	0.84	1.02	1.15	1.23

Table 5.11: Power consumption of the best implementation with two 0201 LEDs, which is the design of the capsule-size prototype. This is SPI with DMA running on the L4 MCU at 24 MHz core frequency. The table includes the impact each of the parts have on the total power consumption, as a percentage.

	Power [mW]	% of Total
Camera (SEIM)	9.4	13%
MCU	7.7	11%
Backscatter switch	0.00025	0%
Backscatter LDO	0.0002	0%
LED 2 (0201)	13.2	19%
main LDO	40.8	57%
Total	71.0	100%

Table 5.12: Power consumption of the best implementation with four 0201 LEDs. This is SPI with DMA running on the L4 MCU at 24 MHz core frequency. The table includes the impact each of the parts have on the total power consumption, as a percentage.

	Power [mW]	% of Total
Camera (SEIM)	9.4	9%
MCU	7.7	7%
Backscatter switch	0.00025	0%
Backscatter LDO	0.0002	0%
LED 4 (0201)	26.4	26%
main LDO	58.6	57%
Total	102.0	100%

Table 5.13: Power consumption of the best implementation with four 0805 LEDs. This is SPI with DMA running on the L4 MCU at 24 MHz core frequency. The table includes the impact each of the parts have on the total power consumption, as a percentage.

	Power [mW]	% of Total
Camera (SEIM)	9.4	5%
MCU	7.7	4%
Backscatter switch	0.00025	0%
Backscatter LDO	0.0002	0%
LED 4 (0805) / 10 (0201)	66.0	34%
main LDO	112.1	57%
Total	195.1	100%

Chapter 6

Discussion

This chapter starts by discussing the two hardware implementations, the custom evaluation board and capsule-size prototype, including potential improvements that should be made based on the knowledge gained during the span of the project. Subsequently, the interfacing between the MCU and NanEyeC camera module is discussed, followed by a brief discussion of the interfacing with the backscatter system. Finally, the power consumption is discussed, focusing on the implementations viable to implement in the capsule-size prototype.

6.1 Custom Evaluation Board

Since the L4 MCU did not arrive until the very end, it was not implemented. Therefore, it would be beneficial to make a new custom evaluation board with the L4 MCU. Additionally, the NanEyeC camera module and the six components between the camera and MCU should be removed. The NanEyeC breakout board can then easily be mounted on the PCB headers without having cut the traces between the PCB headers on the board. However, to use it as an extension module, a new NanEyeC breakout board has to be made, with the correct rotation of the camera module. This requires that the footprint is updated to comply with the information laid out in Section 5.1. Additional improvements are to change the length of PCB headers used as feet; thus, they can properly function as feet and open up eight good connection points for ground, which are essential during debugging, primarily for ease of use for the differential probes.

6.2 Capsule-Size Prototype

The general feedback from all of the contacted PCB manufacturers and the PCB and hardware experts consulted was that a BGA with only 0.4 mm pitch should not be used in a prototype. Because this requires the use of μVias, more expensive stackups, more expensive assembly, are more prone to errors, and requires X-Ray imaging to verify the solders. If all guidelines are followed, it also requires a 6-

layer stackup compared to 4-layers. The reason why we opted to use this MCU was that there were no QFNP MCUs in the L0 and L4 series available on the market, with delivery times within the time frame of this project. The MCU we initially wanted to use was STM32L432KBU6. The placement of this MCU on the bottom of PCB2 is shown in Figure 4.22. The size of this MCU is 5x5 mm with a diameter of approximately 7.1 mm. Because of this large size, the vias for VDD and GND, used for programming, had to be removed. These can be implemented in another way but is a consideration that must be taken into account before manufacturing the programming PCB. With the L4 MCU in the QFNP, the capsule prototype could have been manufactured by Eurocircuit, which is much cheaper than ExceptionPCB. In the final design, a MCU in a WLCSP should be used to allow further minimization. The capsule-size prototype designed here will provide valuable information and experience toward the use of this type of MCU packages. The L4 MCU does also come in a 2.6x3.1 mm WLCSP-36, which should be used instead of the L0 MCU. To make an optimal design using this package type, blind and buried vias have to be utilized in addition to the in-pad μVias. This will increase the cost of the PCB. If it is decided to make this step at some point, it should be considered using blind vias to optimize the current mechanical assembly.

The second feedback from the PCB manufacturers was to change the design into a rigid-flex, where the rigid circular PCBs are interconnected with flexprint. This would isolate all signal and power lines between the PCBs and it would ensure a good connection and would not require assembly for testing. The connector idea was discussed with the PCB manufacturers, and they said that it is often more problems with connectors, and they are prone to user errors, which break the connectors. Because of this, they always recommend flex-rigid designs, where it is possible. Flex-rigid designs are much more expensive than manufacturing three single PCBs, and the flex rigid design sets higher requirements to the mechanical design of the capsule. For the end product, this should be pursued, but it is not worth it for this prototype. The mechanical design is simple and requires soldering and the use of epoxy resin, but from the 3D analysis and mock-ups, it should provide sufficient mechanical support. The optical dome and molding of the capsule is something that will require much more work and knowledge on the subjects. The glass test tube will be sufficient for this prototype as it is transparent, easy to open, and completely waterproof. The size of the cut test tube is 35 mm in length and 13 mm in diameter with a 10 mm internal diameter, having just enough space for the antennas and wires from the backscatter switch. The height of the assembled 3D model of the capsule-size prototype is 25.14 mm without the power switch and 27.65 mm with the power switch. The diameter is 9.5 mm.

The six components connected to the camera should be removed or added after testing the first capsule-size prototype. If the ringing effect is not observed, the six components are not needed, and the space can be used for other components, like more LEDs. We believe that the reason why ams have included these on the camera board is that they have observed the ringing effect and have tested and optimized the values for exactly their trace lengths and other impacts from their PCB

design. If this capsule-size prototype has a ringing effect, it should be tested and fitted with the resistor and capacitor sizes valid for that PCB design. However, the NanEyeC breakout board was used without these components without any problems, so most likely is, the effect negligible. Nevertheless, it is a consideration that should be accounted for in the final product.

6.3 Interfacing MCU and NanEyeC

The NanEyeC datasheet states that the SEIM protocol is a half-duplex SPI master variant without CS. The measurements on the ams NanoBerry Evaluation kit show that it uses a continuous clock. The NanoBerry board uses a Raspberry PI 3A+, which uses the powerful ARM Cortex-53A processor core running at 1.4 GHz. From the Raspberry PI datasheet, it seems that the SPI hardware implementation has gaps in between each SPI-package. However, with a core that fast and powerful, the SEIM protocol may have been implemented by ams as a custom software module with many of the SPI features combined with the special features of the SEIM protocol. The two implementations in this project used SPI with the known limitations. In contrast, the bit banging implementations lacked the proper synchronization due to the missing features of the SPI protocol. Both of our implementations are blocking due to the lack of DMA. In the final software implementation of the interfacing between the MCU and NanEyeC, that will be used in the actual product, DMA is essential both in terms of power efficiency and to have processing time to control the LEDs. Additionally, it is essential for future improvements, such as two-way communication and adding other sensors that require processor time.

A consideration that should be made for future version of the WCE capsule-size prototype is to replace the MCU with an FPGA. The FPGA can use IPs available for purchase, or a custom IP could be made that implements a SEIM module in hardware. This would optimize the power efficiency for the system, given that the core consumption is equal for the MCU and FPGA.

6.3.1 SPI

First of all, it is clear that the SPI hardware implementation on STM32MCUs are not ideal for interfacing with the NanEyeC camera module. The hardware module is limited by a minimum two-clock cycle delay between each SPI package. Accounting for these delays, the execution time of a frame is increased by approximately 14 and 20 % for the 12- and 8-bit implementations, respectively. This is a significant increase, resulting in a frame rate reduction from 9.35 to 8 and 7.5 FPS at 12 MHz for the 12- and 8-bit implementations, respectively. These extra clock cycles reduce both the performance and the power efficiency of the MCU. From the STM32 SPI documentation, it seems like these delays are required by the core, together with the SPI hardware implementation, to handle the data before receiving new data. This was made especially clear from the measurements of the frame

execution time at different core frequencies, summarized in Table 5.2. At 24, 48, and 96 MHz core frequencies for 12 Mbps SPI, the measured execution times are 308, 154, and 124.7 ms, respectively. The documentation stated that 24 MHz was enough for 12 MHz SPI, utilizing DMA. For this implementation without DMA, it is clear that the core alone is too slow to handle the long sequence of data and maintain the minimum two-clock cycles delay between each SPI package. The core frequency has to be eight times higher before it can work fast enough to achieve the minimum two-clock cycle delay. An increase to 96 MHz core frequency is not sustainable for an ultra-low-power WCE application. Additionally, this frequency is above the upper limits of the L0 and L4 MCUs, which caps out at 32 and 80 MHz, respectively. Thus, it is impossible to implement this in a capsule-size prototype because of the large L4+ footprint. The RAM size impact on the execution time, summarized in Table 5.3, shows that 8 kB RAM compared to $\geq$64 kB RAM extends the execution time with approximately 0.2 ms, making the impact nearly negligible. For the 12-bit implementations, this equals a reduction in frame rate from 8.02 to 8.01 FPS.

Much work was put into getting DMA to work, but it was not successful and required more time to debug. It proved to be complicated to implement DMA properly when we were forced to interchange between bit banging and SPI functionality of the same GPIO-pin during the startup sequence. In addition, the complexity of the HAL-library and all of the dependencies made the debugging process very difficult.

To remove the two-clock cycle gap between each SPI package, we implemented SPI using the HAL-library, LL-library and on register level. However, it did not make any difference, and it was clear that the limitation was in the STM32 SPI hardware implementation. It was suggested. that the gap could be removed by implementing a Ping Pong Buffer or some other First In, First Out (FIFO) buffering schemes on top of the STM32 SPI hardware implementation. However, this has not been pursued further due to lack of time. Most likely, this will require a higher core frequency, reducing the system's power efficiency.

6.3.2 Bit Banging

The pure bit banging implementation let us implement a software version of the SPI implementation. The main issue with this implementation is that it is blocking code and that all synchronization has to be taken care of by the programmer. This proved to be particularly difficult to achieve, and the current implementation only works for short time periods at random times. From observations on the oscilloscope, the clock and data line is only in sync for short periods. It jumps between three different alignments between the clock and data lines, where only one is correct - when the data changes align with the falling clock edge. This is most likely due to the implementation of the delays. The current µs delays are implemented using while loops, executing NOPs. The number of clock cycles required to execute the while loop depends on the pipeline and whether the processor can speculate

the addresses with the same accuracy for every run. An alternative method to implement the µs delays was suggested in the last week. The suggestion was to use *inline assembly wait*, which is described here [141]. This implementation does not use a while loop, and generates a very accurate delay, not affected by compiler flags, pipelining, and bus load. This has not been implemented due to a lack of time.

The bit banging implementation only works for very short periods of time in those rare cases where the synchronization is perfect. The implementation uses 48 MHz core frequency, meaning that it can be implemented on an L4 MCU, but not on the L0 MCUs. The execution time of the bit banging implementation is measured to approximately 107 ms, coinciding with the measurements on the NanoBerry board and the execution time indicated in the datasheet.

6.3.3 SPI (with DMA) vs. Bit Banging

For comparison of SPI and bit banging, it is only SPI with DMA that are of interest, as the standard SPI implementation cannot be used on an L0 or L4 MCU because of the high core frequency. The bit banging method can only be implemented on an L4 MCU; thus, this is the only one considered. The simulated power consumption of the L4 MCU for SPI with DMA and bit banging are summarized in Table 5.6. When comparing the power consumption at the same core frequency, it is clear that SPI uses less power than bit banging. This is expected due to the fact the SPI is implemented in hardware, whereas bit banging uses software to toggle a GPIO-pin. However, the difference is more significant when the minimum achievable core frequencies for 12 Mbps transfer speed are considered. SPI can execute at 24 MHz core frequency, while bit banging requires 48 MHz core frequency. This doubling in frequency is almost equivalent to a doubling in power consumption. The SPI implementation with DMA uses 7.6 mW at 24 MHz, while the bit banging implementation uses 14.8 mW. It should be noted that the frame rate for the bit banging implementation is 9.35, while it is only 8.02 for the SPI implementation. However, this increased performance does not make up for the increased power consumption, and the SPI implementation is definitely the most power-efficient solution. Based on the simulations for the power consumption of the MCUs, it is clear that the L4 MCU is the obvious choice. At 24 MHz core frequency, the L0 MCU uses 13.2 mW for the SPI implementation with DMA. The L4 has 43 % lower power consumption than the L0. Additionally, the L4 has more features and supports higher frequencies.

6.4 Interfacing NanEyeC with the Backscatter System

We did not have a stable software implementation when the testing of the interfacing of NanEyeC with the backscatter system was performed. Because of this, the ams NanoBerry evaluation board was used. However, the results should be roughly the same. The clock and data recovery might have been a bit affected by

the gaps in the SPI, but this is something that the backscatter developer team is working on.

6.5 Power Consumption

Power measurements were performed on all active components mounted on the custom evaluation board, and simulations were done for the L0, L4, and L4+ MCUs for SPI and bit banging. The NanEyeC camera module and the MCUs are discussed briefly before the combined total power consumption is discussed for the most viable implementations in the final version of the capsule-size prototype. The last part also includes a discussion of the LDOs and backscatter switch.

6.5.1 NanEyeC Camera Module

For the NanEyeC camera module, the measured power consumption is nearly the same as the values stated in the datasheet. The measured power consumption is 0.29 mW lower. This lower value might be correct because the datasheet value is only valid for 30 MHz input frequency, while our input frequency is 12 MHz. For this implementation, the camera always operates in SEIM mode.

6.5.2 MCU

For the MCU the power consumption of the STM32L062K8T6 mounted on the custom evaluation board was measured for the SPI implementation without DMA, with 24 MHz core frequency. This implementation has a frame rate of only 3.25 FPS due to the slow core. These measurements are not of interest for the final capsule-size implementation. Because of the lack of L4 MCUs and a functional SPI implementation using DMA, these power consumption was simulated in the STM32CubeIDE power analysis tool. The results from these simulations are summarized in Figure 5.29 and Table 5.6. We have already established that L4 is the MCU that is of most interest to implement in the final version. The simulations clearly verify this. The stable working SPI implementation without DMA is implemented on the L4+ MCU and cannot run on the L4 MCU due to the high frequency. The power consumption that is of interest for the final version is the SPI utilizing DMA with 24 MHz core frequency and the bit banging implementation using 48 MHz, core frequency. The simulations shown in the figure and table do not include the power consumption of the DMA module. At 24 MHz, this is approximately 0.1 mW, which is almost negligible. This results in 7.7 and 14.8 mW power consumption for the SPI with DMA and bit banging, respectively. This clearly shows that the SPI implementation is preferred in terms of power efficiency.

6.5.3 LED

A pair of 0201 and 0805 LEDs was implemented on the custom evaluation board for testing. The two graphs in Figure 5.30 shows that from a pure power consumption perspective, it is beneficial to power the LEDs directly from the MCU. Additionally, it shows that the lower threshold voltage for the 0201 LED enables a much lower power usage per LED. From the graphs, it is worth noting that a major difference between the two LEDs is the threshold voltage. At 50 % brightness the power consumption of the 0201 LED is reduced by 50 %, while it it is only reduced by 16 % for the 0805 LED.

Including the power dissipated by the BJT transistor one 0805 LED uses approximately 18.65 mA, while a 0201 LED only consumes 6.6 mA. However, it should be noted that the luminosity intensity is three times higher for the 0805 LED, compared the 0201. For a realistic implementation of a WCE, the number of 0805 LEDs should be at least four, in terms of total luminosity. Four 0805 LEDs results in 720 mcd. To achieve this with 0201 LEDs, 12 LEDs are required. The total power consumption at 720 mcd are 79.2 mW for the 0201 LEDs and 74.6 mW for the 0805 LEDs, making four 0805 the best choice in terms of efficiency (luminosity/watt).

The second consideration when it comes to the LED is the uniformity of the combined light. The literature review showed that the radiation field of the LEDs is essential. The best image quality is obtained with uniform lighting and is especially important for camera chips that use CMOS technology, which is the case for the NanEyeC camera module. As discussed in the initial design considerations for the capsule-size prototype, in Section 4.5.1, the 0805 LED have a 120° radiation field, compared to 110° for the 0201 LED. In addition to this, the 0805 LED has a taller package, lifting the light source higher, which opens for placements closer to the camera module without casting a shadow. For the WCEs on the market, the LED height is even more extended, to the degree that the light source is in level with the lens. This opens up for LED arrangements without having to care for the distance between LEDs and the camera module. However, the large number of 0201 LEDs might compensate for the lower radiation angle by spacing them out over a larger area. Figure 4.25a, shows an example of eight 0201 LEDs placed in a circle. This spacing might even make a more uniform light source than the four 0805 LEDs. Having a large number of LEDs does also open the possibility of completely disabling a few LEDs, saving the power used to uphold the threshold voltage with duty cycle dimming. Another benefit of using the 0201 LED is the fact that it can be powered directly from the MCU, saving additional hardware, cost, and space. It does, of course, depend on the number of 0201 LEDs used.

6.5.4 Total Power Consumption for the Most Viable Implementations

The total power consumption of the most viable implementation for the capsule-size prototype are presented in Table 5.8, 5.11, 5.12 and 5.13. The most surprising part is the huge impact the 3V3 LDO have on the total power consumption. The

power dissipation makes up 57 % of the total power consumption for each implementation. This factor depends only on the difference between the input voltage from the batteries and the VDD voltage level. The VDD voltage level is fixed at 3V3 because of the specifications of the active components. The largest impact on the power consumption would be to find a battery combination that gives an input voltage as close to 3.3 V + the dropout voltage as possible. For the LDO used in the capsule-size prototype, the dropout voltage is 215 mV.

The smallest impact on the total power consumption is the backscatter switch. Combined with the 2V5 LDO it only consumes 450 nW, which is negligible. According to the literature review, the typical power consumption of a transceiver in WCEs is 20-45 mW. This is a huge improvement! If the best and worst-case scenarios from the literature review had replaced the backscatter switch in this design, with two 0201 LEDs, they would make up 20 and 28 % of the total power consumption, respectively.

The second-largest impact on the total power consumption is the LEDs. For the SPI with DMA implementation the impact of the LEDs are 19, 26, and 34 % for two 0201, four 0201, and four 0805 LEDs, respectively. This is significant, and it is important to completely turn off the LEDs when they are not needed. This will cause an even larger impact than the percentages listed, as reducing the total current will also reduce the power dissipation from the LDO.

For the hardware implementations that were considered, the L4 variant has a total power consumption of 71.0 mW, which results in a total lifetime of 3 hours and 23 minutes, when the two 0201 LEDs are on at max brightness for the entire lifetime. The temperature increase of the 3V3 LDO is 23℃, which leads to a total temperature of 60℃ during operation. This is far below the 105℃, which all components can operate in, except the batteries that are rated to a maximum of 60℃. This temperature limit has to be considered.

For the current implementation of the capsule-size prototype, with the L0 MCU, the total power consumption is 84.2 mW. Comparing this to the same implementation with the L4 MCU shows how the LDO's power dissipation increases the impact of each component on the total power consumption. When summing the total power consumption of the L4 and L0 implementations without the LDO, the difference is only 5.5 mW. However, when the LDO is included, the difference is 12.2 mW, which is approximately 57 % higher.

The lifetime of the L0 implementation is 2 hours and 51 minutes, which is 32 minutes less than the L4 implementation. On the other hand, the bit banging implementation uses 110.5 mW, which is significantly higher. It only has a lifetime of 2 hours and 10 minutes, which is approximately 43 % less than the best implementation. It is worth noting that the bit banging implementation has a frame rate of 9.35, while the SPI implementation has 8 FPS

To reduce the power consumption when the capsule is traveling, in the current implementation, the MCU can go to sleep mode, where it only consumes 10 µW and the camera module can go to the IDLE mode, where it consumes 3 mW. However, the power consumption of the camera module is still significant,

so it should be considered implementing a digital switch controlled by the MCU that can cut the power to the camera module completely. The MCU would then disable the camera module before going to sleep and enable the camera when it is ready to start recording. The frame rate can also be lowered in areas of less interest. This would require the MCU to still operate with the same core frequency to keep track of synchronization, while the camera module would spend much of its time in IDLE mode. For 2 FPS the SPI implementation would reduce the power consumption of the camera from 9.4 mW to 4.8 mW, which is approximately a 50 % reduction.

The following scenarios are valid for the best implementation, SPI with DMA on the L4 MCU. When the MCU is in sleep mode, the camera in IDLE mode and the LEDs are turned off, the resulting power consumption is 7.5 mW, and the battery lifetime is 32 hours. If the camera were completely disabled, the total power consumption is only 10.5 µW, which is negligible. At 2 FPS and the LEDs only being one for the exposure time of the frames, the total power consumption would be 33 mW and the total battery lifetime of 7 hours and 15 minutes.

If the best implementation was used with a battery pack with 3.515 V output voltage, the power dissipation of the LDO is reduced from 40.8 mW to 6.5 mW. This would reduce the impact of the LDO from 57 to 18 %, which increases the battery lifetime to 6 hours and 32 minutes during maximum operations with the two 0201 LEDs on at maximum brightness.

Chapter 7

Conclusion

7.1 Summary

The main objective of this project was to make a capsule-size prototype of a WCE and interface it with the backscatter system developed by A. Khaleghi et al. [1]. The capsule-size prototype should implement the backscatter switch used to modulate the incoming carrier wave from the backscatter reader. The camera data modulates the carrier wave and transmits the data to the backscatter system. The backscatter reader and visualization of the images have been developed previously and were not a part of this work.

The first part of the project was to make a custom evaluation board containing all the components part of the capsule-size prototype. The complete camera module NanEyeC from ams was used. For the processing unit, it was decided to use a MCU. From the research, we wanted to use STM32L432, but this was not in stock and could not be delivered in time. Du to the extreme situation in the market, the only MCU available was an STM32L062 MCU for the custom evaluation board and STM32L051 for the capsule-size prototype. A pair of 0201 and 0805 LEDs was chosen for testing. For the backscatter switch, ADG902 was used. The custom evaluation board was designed and manufactured. Everything worked except for the camera module. A NanEyeC breakout board containing only the camera module was designed and manufactured. This board can be mounted as an extension module on the custom evaluation board. Testing on the NanEyeC breakout board led to discovering a typo in the reference footprint in the NanEyeC datasheet.

The software for interfacing the MCU with the camera module was developed on a NUCLEO-L4R5ZI evaluation board connected to the NanEyeC breakout board. The system data rate was set to 12 Mbps as this rate could be implemented for all parts of the system. The best software implementation in terms of power efficiency is SPI with DMA using a 24 MHz core frequency. However, this implementation does not work. The reason for this is that the startup sequence of the camera module requires the GPIO used for the data line to alternate between bit banging and SPI. It cannot be configured as both at the same time, and we believe that the initializing and deinitializing of SPI and DMA causes some problems. The

same SPI implementation, without DMA, works, but it requires a core frequency of 96 MHz to execute at maximum speed. A problem with the STM32 SPI hardware implementation proved to be that there is a minimum two-clock cycle delay between each SPI package. This delay extended the execution time of one frame to 124.7 ms, compared to 107 ms, without the delays. Therefore, an alternative implementation using pure bit banging was developed. This can use a continuous clock and have a 107 ms execution time per frame. However, this is not a stable working implementation and only works in rare cases for short periods due to a lack of proper synchronization.

The SPI implementation can achieve 8 FPS, while bit banging with a continuous clock achieves 9.35 FPS. The SPI implementation without DMA can be implemented on the L0 and L4 MCUs. Nevertheless, these have limits to the maximum frequencies. The achievable frame rates for 24 and 48 MHz core frequencies are 3.25 and 6.49, respectively.

From power measurement and simulations, it is clear that the solution that must be implemented is 12-bit SPI with DMA, running on an L4 MCU with 24 MHz core frequency. The total power consumption for this implementation is 71 mW, which results in 3 hours and 23 minutes battery lifetime when everything is running at max capacity. This includes two 0201 LEDs, using 2 mA each. The largest positive impact on the total power consumption is the implementation of backscattering for communication, saving 20-45 mW. The most significant impact on the total power consumption is the 3V3 LDO, responsible for 57 % of the total power consumption. It should be considered lowering the battery voltage in the final implementation to reduce the power dissipation of the LDO.

The capsule-size prototype is designed and assembled in 3D. Additionally, a programming PCB have been designed for simple and secure programming of the onboard MCU. The capsule-size prototype consists of three PCBs, and has a total length of 25.14 mm without the power switch and 27.65 mm with the power switch. The diameter is 9.5 mm. With thin walls on the capsule shell, the designed capsule-size prototype, without the power switch, will fit inside a capsule with outer measurements: 26 mm length and 11 mm diameter. The design is verified and approved for manufacturing by ExceptionPCB. Unfortunately, it was not ordered, because of lack of time. The design had to be verified from testing on the custom evaluation board, and the software had to be developed. It was only in the last week that we had a working software implementation.

Based on the knowledge gained during testing and conversations with PCB manufacturers, some changes should be made. The NanEyeC footprint has to be changed in accordance with the information in Section 5.1. Additionally the MCU should be replaced with a L4 MCU in a QFN package. Figure 4.22 shows an image of this. The change to the larger L4 MCU will significantly reduce the costs, as it can be ordered in standard pooling. The current implementation on the other hand uses a 36-pin WLCSP MCU with only 0.4 mm pitch. This requires the use of in-pad μVias, which requires a much more advanced and expensive manufacturing technique. An additional consideration that must be made is to change the MCU

with a FPGA and purchase or make a custom SEIM implementation in hardware. This could optimize the power efficiency of the system by having the same benefits of the SPI hardware implementation, with a continuous clock, that would result in 9.35 FPS compared to 8 FPS.

The capsule can be ordered as it is now, after updating the NanEyeC footprint, and it should work with the unstable SPI implementation without DMA at 24 MHz core frequency. This would result in 3 FPS, with 84.2 mW power consumption, resulting in a battery lifetime of 2 hours and 51 minutes. By changing the MCU to a L4 MCU, independent of WLCSP or QFNP, and implementing SPI with DMA the frame rate would be 8 FPS, with a power consumption of 71.0 mW, resulting in a battery lifetime of 3 hours and 23 minutes. Making this the obvious choice.

Due to the lack of a capsule-size prototype, the testing of the interfacing with the backscatter system was done using a dummy capsule with antennas connected to a backscatter switch, which got the camera data from an evaluation board for the NanEyeC camera module, provided by ams. Additionally, a pre-recorded video stream was used during testing. The clock and data were successfully recovered, and the video stream was displayed for the pre-recorded data. The NanEyeC video stream was not displayed as the post-processing and decoding algorithm for the NanEyeC output had not yet been developed by the backscatter team. The recorded output from the backscatter reader was compared to the input for verification. The ams evaluation board was also used for comparison when verifying the output of our working SPI implementation.

7.2 Concluding Remarks

A capsule-size prototype is designed, implementing the backscatter system as a transceiver. The transfer rate is set to 12 Mbps. The assembled capsule-size prototype, without the power switch, will exactly fit inside a capsule with outer measurements: 26 mm length and 11 mm diameter.

The best software implementation, 12-bit SPI with DMA, running on a L4 MCU with 24 MHz core frequency, enables 8 FPS. The total power consumption for this configuration is 71.0 mW, resulting in a lifetime of 3 hours and 23 minutes. By reducing the frame rate to 2 FPS and implementing an adaptive LED scheme, only flashing the LEDs during the exposure time, the lifetime is increases to 7 hours and 15 minutes. From the power analysis it is clear that the implementation of backscattering as the transceiver, reduces the total power consumption by 20-28 %. Giving this WCE implementation a huge advantage to the existing WCEs. Using a L4 compared to a L0 MCU reduces the total power consumption by 16 %. The largest impact on the total power consumption is actually the 3V3 LDO, responsible for 57 % of the total power consumption. Reducing the batteries output voltage, can reduce the impact of the LDO down to 18 %.

Chapter 8

Future Work

The first decision that has to be made is if the MCU should be replaced by a FPGA. The PCB designs can be reused for this purpose, with minor changes. Lattice Semiconductors have a series of ultra-small, ultra-low power FPGAs called iCE40 UltraPlus Family, which might be optimal for this implementation. It has better specs and more options than the L4 MCUs, but the power consumption is not stated in the datasheet [142]. Furthermore, this series only comes in WLCSP with 0.4 mm pitch, where the smallest is 30-pin 2.11 $\times$ 2.54 mm. This will lead to more expensive manufacturing of the capsule-size prototypes for the same reasons laid out for the current capsule-size prototype design.

Given that it is decided to continue to pursue an implementation using a MCU; the first step would be to make a new evaluation board with an L4 MCU, where the camera and the six components between the PCB headers are removed. A new version of the NanEyeC breakout board should be made, with the correct rotation of the camera module. Then it can be used as intended, as an extension module on the custom evaluation board. The NanEyeC breakout board can also be simplified by removing the four resistors and the two capacitors used by ams to remove the ringing effect. The ringing effect was not observed during testing without the components soldered on the breakout board. However, this has to be tested thoroughly at the later stages before it becomes a product. The component values then have to be optimized for the final PCB design of the capsule-size prototype.

The software can then be developed and tested directly on the evaluation board. The problem with the DMA in the SPI implementation has to fixed. Further improving the synchronization for the bit banging can be considered, or at least test the suggested delay implementation. However, based on the power measurements, this is not a viable solution in terms of power consumption due to the high core frequency.

For the capsule-size prototype, we have now received the STM32L432KBU MCU, which fits on the current design of the capsule-size prototype. Figure 4.22 shows this MCU placed on PCB1. This change will require rerouting of PCB1, and the decoupling scheme must be altered to fit the L4 MCU. Additionally, must the programming PCB be altered to fit this new design. The three pins at the top can be

kept in place, but the GND and VDD vias must be moved or removed. The simplest solution is to use the larger vias for the GND and VDD interconnection with PCB2. Implementing the pogo-jig for programming should also be considered, but it is not necessary for low volumes. The change to the large L4 MCU will reduce the manufacturing and assembly cost of the capsule-size prototype from 25 000 NOK to around 5000 NOK, based on estimations done in Eurocircuits PCB calculator. In later revisions of the capsule-size prototype, it might be necessary to use the WLCSP MCU from L4 for further minimization. For an optimal design, this will require 6-layers and the use of μVias, blind vias, and buried vias.

To further lower the power consumption, the main issue is the power dissipation from the 3V3 LDO. Finding a battery pack with a lower output voltage should be considered. The STM32 L4 MCUs have support for SMPS, which can be considered to further lower the power consumption of the MCU. The LED configuration has a huge impact on the total power consumption. The configuration of LED must be chosen carefully. A step in this direction is to test whether several 0201 or a few 0805 LEDs gives the most uniform lighting. It should also be considered implementing a digital switch controlled by the MCU that can cut the power to the camera module completely. This is crucial to reduce the power consumption to a minimum during traveling when no images are captured.

From our testing, it seemed like the backscatter switch ADG902 functioned without a power supply. This can be investigated further, as this would remove the need for the 2.5V output LDO, which is only used for this switch. This would lead to both savings in cost, space, and power. However, the power savings from this change is negligible.

The temperature limit of the batteries has to be considered. They are rated to a maximum of 60℃, and the temperature inside the capsule is estimated to be around 60℃. The manufacturer should be contacted for an evaluation of this issue.

Bibliography

[1] A. Khaleghi and I. Balasingham, "Medical implant with wireless communication," English, pat. WO2019141782A1, Jan. 2019. [Online]. Available: https://patentimages.storage.googleapis.com/eb/1a/0c/1304606462a023/WO2019141782A1.pdf.

[2] P. R. Slawinski, K. L. Obstein, and P. Valdastri, "Emerging issues and future developments in capsule endoscopy," *Techniques in Gastrointestinal Endoscopy*, vol. 17, no. 1, pp. 40–46, Jan. 2015. DOI: 10.1016/j.tgie.2015.02.006.

[3] J.-C. Saurin, N. Beneche, C. Chambon, and M. Pioche, "Challenges and future of wireless capsule endoscopy," *Clinical Endoscopy*, vol. 49, no. 1, pp. 26–29, Jan. 2016. DOI: 10.5946/ce.2016.49.1.26.

[4] R. Koprowski, "Overview of technical solutions and assessment of clinical usefulness of capsule endoscopy," *BioMedical Engineering OnLine*, vol. 14, no. 1, Dec. 2015. DOI: 10.1186/s12938-015-0108-3.

[5] G. Ciuti, A. Menciassi, and P. Dario, "Capsule endoscopy: From current achievements to open challenges," *IEEE Reviews in Biomedical Engineering*, vol. 4, pp. 59–72, 2011. DOI: 10.1109/rbme.2011.2171182.

[6] R. Shrestha, S. K. Mohammed, M. M. Hasan, X. Zhang, and K. A. Wahid, "Automated adaptive brightness in wireless capsule endoscopy using image segmentation and sigmoid function.," eng, *IEEE transactions on biomedical circuits and systems*, vol. 10, pp. 884–92, 4 Aug. 2016.

[7] I. Tsekoura, G. Rebel, P. Glosekotter, and M. Berekovic, "An evaluation of energy efficient microcontrollers," in *2014 9th International Symposium on Reconfigurable and Communication-Centric Systems-on-Chip (ReCoSoC)*, IEEE, May 2014. DOI: 10.1109/recosoc.2014.6861368.

[8] A. Khaleghi, A. Hasanvand, and I. Balasingham, "Radio frequency backscatter communication for high data rate deep implants," *IEEE Transactions on Microwave Theory and Techniques*, vol. 67, no. 3, pp. 1093–1106, Mar. 2019. DOI: 10.1109/tmtt.2018.2886844.

[9] R. Noormohammadi, A. Khaleghi, and I. Balasingham, "Battery-free wireless communication for video capsule endoscopy," in *2019 13th International Symposium on Medical Information and Communication Technology (ISMICT)*, IEEE, May 2019. DOI: 10.1109/ismict.2019.8743718.

[10] M. Carol DerSarkissian, *Digestive diseases and endoscopy*, Aug. 2021. [Online]. Available: https://www.webmd.com/digestive-disorders/digestive-diseases-endoscopy.

[11] iData Reaserch, *75 million endoscopies performed annually according to idata research procedure analysis*, Jan. 2018. [Online]. Available: https://idataresearch.com/75-million-endoscopies-performed-annually-according-to-idata-research-procedure-analysis/.

[12] H. Uysal and H. Daskaya, "Analysis of anesthesia administration in the endoscopy unit in terms of patient profile and complications: Retrospective study," *Medeniyet Medical Journal*, 2019. DOI: 10.5222/mmj.2019.52280.

[13] D. Medisin, *Beroligende bør gis ved behov*, Apr. 2011. [Online]. Available: https://www.dagensmedisin.no/artikler/2011/02/24/beroligende-bor-gis-ved-behov/.

[14] M. Michael W. Smith, *Colorectal cancer*, Nov. 2017. [Online]. Available: https://www.webmd.com/colorectal-cancer/guide/colorectal-polyps-cancer.

[15] A. C. Society, *What is colorectal cancer?* Jun. 2020. [Online]. Available: https://www.cancer.org/cancer/colon-rectal-cancer/about/what-is-colorectal-cancer.html.

[16] M. FREDERICK GANDOLFO, *What you should know about colon polyps*, Mar. 2018. [Online]. Available: https://www.kevinmd.com/blog/2018/03/know-colon-polyps.html.

[17] C. Clinic, *How long does it take colon polyps to turn cancerous?* Apr. 2018. [Online]. Available: https://health.clevelandclinic.org/how-quickly-do-colon-polyps-turn-cancerous/.

[18] Y. Xi and P. Xu, "Global colorectal cancer burden in 2020 and projections to 2040," *Translational Oncology*, vol. 14, no. 10, p. 101 174, Oct. 2021. DOI: 10.1016/j.tranon.2021.101174.

[19] W. H. Organization, *Colorectal cancer*, Dec. 2020. [Online]. Available: https://gco.iarc.fr/today/data/factsheets/cancers/10_8_9-Colorectum-fact-sheet.pdf.

[20] A. C. Society, *American cancer society guideline for colorectal cancer screening*, Nov. 2017. [Online]. Available: https://www.cancer.org/cancer/colon-rectal-cancer/detection-diagnosis-staging/acs-recommendations.html.

[21] N. C. Institute, *Cancer stat facts: Colorectal cancer*, Dec. 2019. [Online]. Available: https://seer.cancer.gov/statfacts/html/colorect.html.

[22] N. C. Institute, *Cancer stat facts: Small intestine cancer*, Dec. 2019. [Online]. Available: https://seer.cancer.gov/statfacts/html/smint.html.

[23] U. of Michigan - Michigan Medicine, *Polyps, tumors and malabsorptive diseases in the small bowel*, Jan. 2021. [Online]. Available: https://www.uofmhealth.org/conditions-treatments/digestive-and-liver-health/polyps-tumors-and-malabsorptive-diseases-small-bowel.

[24] S. Luke's, *Push enteroscopy*, Jan. 2021. [Online]. Available: https://www.saintlukeskc.org/health-library/push-enteroscopy.

[25] E. Britannica, *Examination of the human colon*, Dec. 2021. [Online]. Available: https://www.britannica.com/science/inflammatory-bowel-disease#/media/1/1085330/59955.

[26] S. H. Lakeland, *Capsule endoscopy*, Jan. 2021. [Online]. Available: https://www.spectrumhealthlakeland.org/lakeland-ear-nose-and-throat/ent-health-library/Content/3/40436/.

[27] B. A. P and J. A. H, "Analysis of current and future technologies of capsule endoscopy: A mini review," *Archives of Preventive Medicine*, pp. 031–034, Jul. 2020. DOI: 10.17352/apm.000016.

[28] Medtronic, *Pillcam™ ugi system*, Apr. 2021. [Online]. Available: https://www.medtronic.com/covidien/en-us/products/capsule-endoscopy/pillcam-ugi-system.html.

[29] Medtronic, *Pillcam™ sb 3 system*, Apr. 2021. [Online]. Available: https://www.medtronic.com/covidien/en-us/products/capsule-endoscopy/pillcam-sb-3-system.html.

[30] S. D. Anchal Khadse, "Pill camera," in *International Journal of Advanced Research in Computer and Communication Engineering*, 4, vol. 5, IJARCCE, Apr. 2016, pp. 329–338.

[31] Lumex, *Lumex low current smd/smt leds*, Jan. 2018. [Online]. Available: https://www.lumex.com/led-smd-low-current.html.

[32] Lumex, *Datasheet: Sml-lx0201nwd-tr*, May 2017. [Online]. Available: https://www.lumex.com/datasheet/SML-LX0201NWD-TR.html.

[33] Vishay Semiconductors, *Datasheet vlmw1500 white*, Mar. 2021. [Online]. Available: https://www.vishay.com/docs/82522/vlmw1500.pdf.

[34] Vishay Semiconductors, *Datasheet vlmw1300 white*, Mar. 2021. [Online]. Available: https://www.vishay.com/docs/82435/vlmw1300.pdf.

[35] Inolux, *Datasheet white led in-s85at series*, Feb. 2019. [Online]. Available: https://no.mouser.com/datasheet/2/180/IN_S85AT_Series_V1_1-2323391.pdf.

[36] LITE-ON Technology Corporation, *Datasheet lite-on chipled 1206 ltw-216ts5*, Jun. 2021. [Online]. Available: https://no.mouser.com/datasheet/2/239/lites11449_1-2271967.pdf.

[37] N. Blanc, "Ccd versus cmos – has ccd imaging come to an end?," Dec. 2003.

[38] P. Swain, "Wireless capsule endoscopy," *Gut*, vol. 52, no. 90004, pp. 48iv–50, Jun. 2003. DOI: 10.1136/gut.52.suppl_4.iv48.

[39] I. Cevik, X. Huang, H. Yu, M. Yan, and S. Ay, "An ultra-low power CMOS image sensor with on-chip energy harvesting and power management capability," *Sensors*, vol. 15, no. 3, pp. 5531–5554, Mar. 2015. DOI: 10.3390/s150305531.

[40] J. Gerber, A. Bergwerk, and D. Fleischer, "A capsule endoscopy guide for the practicing clinician: Technology and troubleshooting," *Gastrointestinal Endoscopy*, vol. 66, no. 6, pp. 1188–1195, Dec. 2007. DOI: 10.1016/j.gie.2007.06.003.

[41] A. A. Kishk, M. R. Basar, F. Malek, K. M. Juni, M. S. Idris, and M. I. M. Saleh, "Ingestible wireless capsule technology: A review of development and future indication," *International Journal of Antennas and Propagation*, vol. 2012, p. 807 165, 2012, ISSN: 1687-5869. DOI: 10.1155/2012/807165. [Online]. Available: https://doi.org/10.1155/2012/807165.

[42] D. R. Cave, D. E. Fleischer, J. A. Leighton, D. O. Faigel, R. I. Heigh, V. K. Sharma, C. J. Gostout, E. Rajan, K. Mergener, A. Foley, M. Lee, and K. Bhattacharya, "A multicenter randomized comparison of the endocapsule and the pillcam SB," *Gastrointestinal Endoscopy*, vol. 68, no. 3, pp. 487–494, Sep. 2008. DOI: 10.1016/j.gie.2007.12.037.

[43] HeraldKeeper, *Cmos image sensor (cis) for medical market outlook, share, size from (2021-2027)*, Dec. 2021. [Online]. Available: https://www.marketwatch.com/press-release/cmos-image-sensor-cis-for-medical-market-outlook-share-size-from-2021-2027-by--sony-omnivision-stmicroelectronics-2021-12-20?tesla=y.

[44] M. Intelligence, *Image sensors market - growth, trends, covid-19 impact, and forecasts (2021 - 2026)*, Jan. 2021. [Online]. Available: https://www.mordorintelligence.com/industry-reports/image-sensors-market.

[45] OmniVision, *Product brief: Ov6948 200x200*, Oct. 2020. [Online]. Available: https://www.ovt.com/download/sensorpdf/248/OmniVision_OV6948.pdf.

[46] AMS, *Datasheet naneye miniaturized image sensor*, Nov. 2021. [Online]. Available: https://ams.com/documents/20143/36005/NanEye_DS000501_8-00.pdf.

[47] Sony, *Product information: Imx568-aamj*, Dec. 2021. [Online]. Available: https://www.sony-semicon.co.jp/products/common/pdf/IMX530-AAMJ_AAQJ_Flyer.pdf.

[48] Canon, *Technical specification: 3u5mgxsbac*, Jan. 2021. [Online]. Available: https://canon-cmos-sensors.com/canon-3u5mgxs-5mp-cmos-sensor/.

[49] M. Intelligence, *Datasheet: Miniature cmos image sensor module*, Apr. 2015. [Online]. Available: https://www.fujikura.co.jp/eng/products/newbusiness/medicaldevices/01/2050259_13471.html.

[50] Himax, *Product brief: Hm11b1*, Jan. 2015. [Online]. Available: https://www.himax.com.tw/products/cmos-image-sensor/always-on-vision-sensors/hm11b1/.

[51] Samsung, *Product brief: Isocell 5e9*, Jan. 2021. [Online]. Available: https://www.samsung.com/semiconductor/minisite/isocell/mobile-image-sensors/isocell-slim-5e9/.

[52] M. Villien, *The disposable endoscopy show has begun and image sensors are under the spotlight - an interview with omnivision technologies, inc.* Jul. 2020. [Online]. Available: https://www.i-micronews.com/the-disposable-endoscopy-show-has-begun-and-image-sensors-are-under-the-spotlight-%E2%94%80-an-interview-with-omnivision-technologies-inc/?cn-reloaded=1.

[53] OmniVision, *Product brief: Ovm6948 200x200*, Oct. 2020. [Online]. Available: https://www.ovt.com/download/sensorpdf/553/OmniVision_OVM6948.pdf.

[54] AMS, *Ams cmos image sensors product lines*, Dec. 2021. [Online]. Available: https://ams.com/cmos-imaging-sensors.

[55] OmniVision, *Product brief: Ov426 companion asic*, Oct. 2019. [Online]. Available: https://www.ovt.com/download/sensorpdf/259/OmniVision_OV426.pdf.

[56] D. John L. Hennessy, *Computer Architecture A Quantitative Approach*, 6th ed., N. McFadden, Ed. Katey Birtcher, 2019, ISBN: 978-0-12-811905-1.

[57] U. Pesovic, Z. Jovanovic, S. Randjic, and D. Markovic, "Benchmarking performance and energy efficiency of microprocessors for wireless sensor network applications," in *2012 Proceedings of the 35th International Convention MIPRO*, 2012, pp. 743–747.

[58] A. R. Weiss, *Dhrystone benchmark*, Nov. 2002. [Online]. Available: https://www.johnloomis.org/NiosII/dhrystone/ECLDhrystoneWhitePaper.pdf.

[59] ARM, *Arm® 180nm ultra low power platform*, Online, Mar. 2009. [Online]. Available: https://developer.arm.com/documentation/pfl0307/a/.

[60] P. Turcza and M. Duplaga, "Low power FPGA-based image processing core for wireless capsule endoscopy," *Sensors and Actuators A: Physical*, vol. 172, no. 2, pp. 552–560, Dec. 2011. DOI: 10.1016/j.sna.2011.09.026.

[61] Y. Tanurhan, "Processors and fpgas quo vadis?" *Computer*, vol. 39, no. 11, pp. 108–110, 2006. DOI: 10.1109/MC.2006.395.

[62] R. Puers, R. Carta, and J. Thoné, "Wireless power and data transmission strategies for next-generation capsule endoscopes," *Journal of Micromechanics and Microengineering*, vol. 21, no. 5, p. 054 008, Apr. 2011. DOI: 10.1088/0960-1317/21/5/054008.

[63] P. Swain, "The future of wireless capsule endoscopy," *World Journal of Gastroenterology*, vol. 14, no. 26, p. 4142, 2008. DOI: 10.3748/wjg.14.4142.

[64] J. Thoné, J. Verlinden, and R. Puers, "An efficient hardware-optimized compression algorithm for wireless capsule endoscopy image transmission," *Procedia Engineering*, vol. 5, pp. 208–211, 2010. DOI: 10.1016/j.proeng.2010.09.084.

[65] W. Rhee, Ed., *Wireless Transceiver Circuits: System Perspectives and Design Aspects*. CRC PR INC, Feb. 2015, 580 pp., ISBN: 1482234351. [Online]. Available: https://www.ebook.de/de/product/22736582/wireless_transceiver_circuits_system_perspectives_and_design_aspects.html.

[66] J. H. Bae and M. H. Sunwoo, "10.07uw multi-mode baseband transceiver for encrypted capsule endoscopy," *Journal of Signal Processing Systems*, vol. 93, no. 1, pp. 25–32, May 2020. DOI: 10.1007/s11265-020-01552-z.

[67] M. Park, T. Kang, I. Lim, K.-I. Oh, S.-E. Kim, J.-J. Lee, and H.-I. Park, "Low-power, high data-rate digital capsule endoscopy using human body communication," *Applied Sciences*, vol. 8, no. 9, p. 1414, Aug. 2018. DOI: 10.3390/app8091414.

[68] D. Dobkin, *The RF in RFID : UHF RFID in practice*. Amsterdam: Elsevier/Newnes, 2013, ISBN: 9780123948304.

[69] A. Khaleghi and I. Balasingham, "Wireless capsule endoscopy using backscatter communication," in *2019 13th European Conference on Antennas and Propagation (EuCAP)*, 2019, pp. 1–5.

[70] A. Khaleghi, H. Qadir, P.-H. Lehne, and I. Balasingham, "Wireless capsule video endoscopy for population-based colon cancer screening using 5g network," Jan. 2022.

[71] M. Wolff, *Computers as components - Principles of embedded computing system design*, 4th ed., N. McFadden, Ed. Morgan Kaufman, 2017, ISBN: 978-0-12-805387-4.

[72] STMicroelectronics, *Reference manual stm32l4+ series advanced arm®-based 32-bit mcus*, Dec. 2020. [Online]. Available: https://www.st.com/resource/en/reference_manual/dm00310109-stm32l4-series-advanced-armbased-32bit-mcus-stmicroelectronics.pdf.

[73] E. Odunlade, *Introduction to bit banging: Spi communication via bit banging*, Aug. 2018. [Online]. Available: https://circuitdigest.com/article/introduction-to-bit-banging-spi-communication-in-arduino-via-bit-banging.

[74] Altium, *Altium designer documentation*, Sep. 2021. [Online]. Available: https://www.altium.com/documentation/altium-designer.

[75] I. Association, *Ipc-2221a generic standard on printed board design*, May 2003. [Online]. Available: http://www-eng.lbl.gov/~shuman/NEXT/CURRENT_DESIGN/TP/MATERIALS/IPC-2221A(L).pdf.

[76] S. PCB, *Saturn pcb design toolkit version 8.07*, Aug. 2021. [Online]. Available: https://saturnpcb.com/saturn-pcb-toolkit/.

[77] R. Shashikanth, *What is a prepreg in pcb manufacturing?* Jan. 2021. [Online]. Available: https://www.protoexpress.com/blog/prepreg-the-slice-of-cheese-in-your-pcb/.

[78] S. Tyagi, *7 tips and pcb design guidelines for emi and emc*, Aug. 2020. [Online]. Available: https://www.protoexpress.com/blog/7-pcb-design-tips-solve-emi-emc-issues/.

[79] T. Instruments, *Pcb design guidelines for reduced emi*, Nov. 2019. [Online]. Available: https://www.ti.com/lit/an/szza009/szza009.pdf?ts=1641544439777&ref_url=https%253A%252F%252Fwww.google.com%252F.

[80] M. Devine, *Using via in pad? what you need to know - guidelines, dimensions and more*, Mar. 2020. [Online]. Available: https://www.youtube.com/watch?v=FM3pRM0CxGw.

[81] S. F. Circuits, *The importance of pcb trace widths in pcb design*, Jan. 2021. [Online]. Available: https://www.sfcircuits.com/pcb-school/pcb-trace-widths.

[82] P. Universe, *Copper thickness faq*, Jan. 2021. [Online]. Available: https://www.pcbuniverse.com/pcbu-tech-tips.php?a=4.

[83] C. E. Schuster, M. G. Vangel, and H. A. Schafft, "Improved estimation of the resistivity of pure copper and electrical determination of thin copper film dimensions," *Microelectronics Reliability*, vol. 41, no. 2, pp. 239–252, Feb. 2001. DOI: 10.1016/s0026-2714(00)00227-4.

[84] ExceptionPCB, *Capabilities*, Jan. 2021. [Online]. Available: https://www.exceptionpcb.com/capabilities-%5C%e2%5C%80%5C%8b/.

[85] Eurocircuits, *Eurocircuits design guidelines*, Jul. 2014. [Online]. Available: https://www.eurocircuits.com/wp-content/uploads/ec2015/ecImage/blogs/Tolerances-on-PCB/ec-design-guidelines.pdf.

[86] T. I. S. of Power Supply Design, *Placement of thermal vias*, Nov. 2017. [Online]. Available: https://techweb.rohm.com/knowledge/dcdc/dcdc_pwm/dcdc_pwm03/3256.

[87] Altium, *Sizing vias in your next multilayer pcbs*, May 2020. [Online]. Available: https://resources.altium.com/p/sizing-vias-your-next-multilayer-pcbs.

[88] Z. Peterson, *Your complete guide to circuit board thermal analysis*, Mar. 2021. [Online]. Available: https://resources.altium.com/p/circuit-board-thermal-analysis.

[89] M. Harris, *Pdn analyzer simulation of 65w single ic led driver*, Jan. 2020. [Online]. Available: https://resources.altium.com/p/pdn-analyzer-simulation-of-65w-single-ic-led-driver.

[90] E. Quintero, *Smd vs smt vs tht: What technology works best for mass manufacturing?* Jan. 2020. [Online]. Available: https://titoma.com/blog/smt-smd-bga-tht-meaning.

[91] M. Laps, R. Grace, B. Sloka, J. Prymak, X. Xu, P. Pinceloup, A. Gurav, M. Randall, P. Lessner, and A. Tajuddin, "Capacitors for reduced microphonics and sound emission," *Assemblies and Materials Association*, Apr. 2007.

[92] E. Notes, *Ceramic capacitor*, Jan. 2021. [Online]. Available: https://www.electronics-notes.com/articles/electronic_components/capacitors/ceramic-dielectric-types-c0g-x7r-z5u-y5v.php.

[93] S. Tyagi, *What is the use of a decoupling capacitor?* Mar. 2021. [Online]. Available: https://www.protoexpress.com/blog/decoupling-capacitor-use/.

[94] I. Stackpole Electronics, *Thick film and thin film resistors – a comparison*, Jan. 2021. [Online]. Available: https://www.digikey.no/no/pdf/s/stackpole-electronics/thick-thin-film-resistors.

[95] R. Teja, *Classification and different types of transistors | bjt, fet, npn, pnp*, Apr. 2021. [Online]. Available: https://www.electronicshub.org/transistors-classification-and-types/.

[96] IPC, *Ipc-7351b naming convention for surface mount device 3d models and footprints*, Oct. 2021. [Online]. Available: https://www.pcb-3d.com/tutorials/ipc-7351b-naming-convention-for-surface-mount-device-3d-models-and-footprints/.

[97] Z.-A. E. Design and Manufacturing, *Datasheet bc846aw-bc848cw npn small signal transistor in sot323*, Sep. 2021. [Online]. Available: https://www.zaxis.net/smd-package-types-sizes/.

[98] Diodes, *Datasheet bc846aw-bc848cw npn small signal transistor in sot323*, Apr. 2017. [Online]. Available: https://www.diodes.com/assets/Datasheets/ds30250.pdf.

[99] J. Teel, *How to pick the right voltage regulator(s) for your design*, Sep. 2021. [Online]. Available: https://predictabledesigns.com/how-to-pick-the-right-voltage-regulators-for-your-design/.

[100] G. Morita, *Understand low-dropout regulator (ldo) concepts to achieve optimal designs*, Dec. 2014. [Online]. Available: https://www.analog.com/en/analog-dialogue/articles/understand-ldo-concepts.html.

[101] T. Instruments, *Understanding the terms and definitions of ldo voltage regulators*, Oct. 1999. [Online]. Available: https://www.ti.com/lit/an/slva079/slva079.pdf.

[102] T. Instruments, *Datasheet tlv71725pdqnt*, Apr. 2021. [Online]. Available: https://www.ti.com/lit/gpn/TLV717P.

[103] STMicroelectronics, *Datasheet stm32l062*, Aug. 2020. [Online]. Available: https://www.st.com/resource/en/datasheet/stm32l062k8.pdf.

[104] ams, *Nanoberry evaluation kit*, Dec. 2021. [Online]. Available: https://ams.com/en/nanoberry-eval-kit.

[105] ams, *Naneyec camera module*, Dec. 2021. [Online]. Available: https://ams.com/en/miniature-camera-modules.

[106] STMicroelectronics, *Stm32 development tools*, Mar. 2021. [Online]. Available: https://www.st.com/en/development-tools/stm32cubeide.html.

[107] STMicroelectronics, *Description of stm32l4/l4+ hal and low-layer drivers*, Feb. 2020. [Online]. Available: https://www.st.com/resource/en/user_manual/dm00173145-description-of-stm32l4l4-hal-and-lowlayer-drivers-stmicroelectronics.pdf.

[108] STMicroelectronics, *Stm32 ultra low power mcus*, Jun. 2021. [Online]. Available: https://www.st.com/en/microcontrollers-microprocessors/stm32-ultra-low-power-mcus.html.

[109] STMicroelectronics, *Stm32l series ultra-low-power 32-bit mcus releasing your creativity*, Aug. 2021. [Online]. Available: https://www.rutronik.com/fileadmin/Rutronik/Supplier/STMicroelectronics/Landing_Pages/STM32L496xx_Family/RS4386_brstm32ulp.pdf.

[110] STMicroelectronics, *Datasheet stm32l412*, May 2018. [Online]. Available: https://www.st.com/resource/en/datasheet/STM32L412.pdf.

[111] STMicroelectronics, *Datasheet stm32l432*, May 2018. [Online]. Available: https://www.st.com/resource/en/datasheet/stm32l432kc.pdf.

[112] STMicroelectronics, *Datasheet stm32l051*, May 2018. [Online]. Available: https://www.st.com/resource/en/datasheet/STM32L051.pdf.

[113] STMicroelectronics, *Design recommendations for stm32l4xxxx with external smps, for ultra-low-power applications with high performance*, Jul. 2018. [Online]. Available: https://www.st.com/resource/en/application_note/dm00350156-design-recommendations-for-stm32l4xxxx-with-external-smps-for-ultralowpower-applications-with-high-performance-stmicroelectronics.pdf.

[114] STMicroelectronics, *Design recommendations for stm32l4xxxx with external smps, for ultra-low-power applications with high performance*, Jul. 2018. [Online]. Available: https://www.st.com/resource/en/application_note/an4978-design-recommendations-for-stm32l4xxxx-with-external-smps-for-ultralowpower-applications-with-high-performance-stmicroelectronics.pdf.

[115] Altium, *What causes ringing on a pcb and how to solve it*, Dec. 2020. [Online]. Available: https://resources.altium.com/p/what-causes-ringing-on-a-pcb-and-how-to-solve-it.

[116] STMicroelectronics, *Datasheet stm32l432*, Jan. 2015. [Online]. Available: https://www.st.com/resource/en/application_note/dm00112257-getting-started-with-stm32l0xx-hardware-development-stmicroelectronics.pdf.

[117] T. Yuden, *Datasheet lsqna201212t101k*, Sep. 2021. [Online]. Available: https://ds.yuden.co.jp/TYCOMPAS/ut/detail?pn=LSQNA201212T101K&u=M.

[118] STMicroelectronics, *Layout recommendations for the design of boards - power supply filtering*, Aug. 2020. [Online]. Available: https://www.st.com/resource/en/application_note/an5240-layout-recommendations-for-the-design-of-boards-with-the-st25r39161720-devices-stmicroelectronics.pdf.

[119] STMicroelectronics, *St-link/v2 in-circuit debugger/programmer for stm8 and stm32*, Oct. 2018. [Online]. Available: https://www.st.com/resource/en/user_manual/dm00026748-st-link-v2-in-circuit-debugger-programmer-for-stm8-and-stm32-stmicroelectronics.pdf.

[120] Olimex, *Arm-jtag 20-10 adapter*, Sep. 2021. [Online]. Available: https://www.olimex.com/Products/ARM/JTAG/ARM-JTAG-20-10/resources/ARM-JTAG-20-10-schematic.pdf.

[121] A. Keil, *Arm target connectors - serial wire mode interface*, Sep. 2021. [Online]. Available: https://www.keil.com/support/man/docs/ulinkME/ulinkme_hw_connectors.htm.

[122] STMicroelectronics, *Stm32 microcontroller system memory boot mode*, Oct. 2021. [Online]. Available: https://www.st.com/resource/en/application_note/cd00167594-stm32-microcontroller-system-memory-boot-mode-stmicroelectronics.pdf.

[123] STMicroelectronics, *Stm32 microcontroller gpio configuration for hardware settings and low-power consumption*, Aug. 2021. [Online]. Available: https://www.st.com/resource/en/application_note/an4899-stm32-microcontroller-gpio-configuration-for-hardware-settings-and-lowpower-consumption-stmicroelectronics.pdf.

[124] A. Devices, *Datasheet adg901/adg902*, Jan. 2017. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/adg901_902.pdf.

[125] Renata, *Datasheet renata sr754sw-sr48*, Oct. 2014. [Online]. Available: https://www.mega-piles.com/im/RENATA-SR754SW-SR48-309-1-55V-80mAh_1234.pdf.

[126] STMicroelectronics, *Datasheet ld59015c33r*, Jul. 2017. [Online]. Available: https://www.st.com/resource/en/datasheet/ld59015.pdf.

[127] STMicroelectronics, *Datasheet stlq50c25r*, Jul. 2019. [Online]. Available: https://www.st.com/resource/en/datasheet/stlq50.pdf.

[128] Molex, *Slimstack board-to-board/board-to-fpc connectors*, Dec. 2021. [Online]. Available: https://www.molex.com/molex/products/family/slimstack_fine_pitch_smt_board_to_board_connectors.

[129] ExceptionPCB, *Excpetionpcb manufacturing capabilities*, Nov. 2021. [Online]. Available: http://5zb.f4e.myftpupload.com/wp-content/uploads/2019/08/Generic_Exception_capabilities.xls.

[130] Vishay, *Vishay dale resistors - medical 101*, Dec. 2021. [Online]. Available: https://www.vishay.com/docs/49875/_medical_101_dale_thin_film_vmn_pl0482.pdf.

[131] Yaego, *Yageo at automotive grade thin film chip resistors*, Aug. 2021. [Online]. Available: https://no.mouser.com/new/yageo/yageo-at-series-resistors/.

[132] STMmicroelectronics, *Thermal management guidelines for stm32 applications*, Apr. 2019. [Online]. Available: https://www.st.com/resource/en/application_note/an5036-thermal-management-guidelines-for-stm32-applications-stmicroelectronics.pdf.

[133] Harwin, *Datasheet m52-040023vxx45*, Jan. 2014. [Online]. Available: https://cdn.harwin.com/pdfs/M52-040023VXX45.pdf.

[134] N. Copa, *Datasheet chs-01ta*, Nov. 2021. [Online]. Available: https://www.nidec-copal-electronics.com/e/catalog/switch/chs.pdf.

[135] D. Haboud, *Multi-board design basics in altium designer*, Aug. 2018. [Online]. Available: https://resources.altium.com/webinars/multi-board-design-basics-in-altium-designer-webinar.

[136] 3M, *Datasheet ts-0769*, Nov. 2008. [Online]. Available: https://www.mouser.com/datasheet/2/1/ts0769-35914.pdf.

[137] RS, *Rs pro test pin 1.5a*, Oct. 2021. [Online]. Available: https://no.rs-online.com/web/p/test-pins/2615137.

[138] ARM, *Cortex-m4 technical reference manual r0p0*, Nov. 2021. [Online]. Available: https://developer.arm.com/documentation/ddi0439/b/CHDDIGAC.

[139] A. Vinci, *How to measure current with an oscilloscope*, Apr. 2021. [Online]. Available: https://www.tek.com/en/blog/how-can-an-oscilloscope-measure-current.

[140] T. Instruments, *How to choose an appropriate shunt resistor*, Jan. 2021. [Online]. Available: https://training.ti.com/sites/default/files/docs/2.02_-_how_to_choose_a_shunt_resistor.pdf.

[141] S. E. Bug, *Cycles per instruction in delay loop on arm, second answer*, Sep. 2015. [Online]. Available: https://stackoverflow.com/questions/32719767/cycles-per-instruction-in-delay-loop-on-arm.

[142] L. Semiconductors, *Ice40 ultraplus family data sheet*, Dec. 2020. [Online]. Available: https://www.farnell.com/datasheets/3215488.pdf.

Appendix A

BOM: Custom Evaluation Board

Table A.1: The complete BOM for the custom evaluation board. The most important information is included. This BOM was verified and all components sourced by Eurocircuit. The complete BOM with more detailed information is available in the GitHub repositories described in Section 4.

Designator	Comment	Suggested	Qty
BT1, BT2, BT3	Battery holders	2988	3
C3-6, C19	0.1uF_X5R_0603	06033D104KAT2A	5
C7, C9	15pF	06035A150JAT2A	2
C8, C16	470nF	CGA3E1X7R1C474K080AC	2
C10, C12-14	100nF	06033D104KAT2A	4
C11	0.1uF	06033D104KAT2A	1
C15	10uF	06034D106KAT2A	1
C17, C18	NC		2
J1	Terminal block female	1757242	1
J4, J5	PCB header 2.54 mm (1x10)	M20-9771046	2
J6, J7, J8, J9	PCB header 2.54 mm (1x2)	TSW-102-07-G-S	4
L1	100uH	CB2012T101K	1
LED1, LED2	0805 LED	IN-S85AT5UW	2
LED3, LED4	0201 LED	SML-LX0201NWD-TR	2
P1	10-pin JTAG connector	FTSH-105-01-L-DV	1
P4, P5	Coaxial connectors - Male	U.FL-R-SMT-1(01)	2
P7-10	PCB header 2.54 mm (1x2)	5-146280-2	4
Q1	BJT transistor	BC847BW-7-F	1
R1, R2	10k	TNPU060310K0AWEN00	2
R3	5k	PCAN0603E501BST5	1
R4, R5	40R	RT0603DRE0740R2L	2
R6	125R	RT0603FRE07124RL	1
R7, R8	24R	804-8710	2
S1	Camera Module	NanEyeC	1
U1	3V3 LDO	LD59015C33R	1
U2	2V5 LDO	STLQ50C25R	1
U4	Backscatter switch	ADG902BRMZ	1
U5	MCU	STM32L062K8T6	1
NA	Programming device	ST-LINK/V2 Programmer	1
NA	JTAG adapter board	ARM-JTAG-20-10	1
NA	Banana plug red	108-1702-101	1
NA	Banana plug black	108-1703-101	1
NA	Shorts	SNT-100-BK-G	5
NA	Male power plug	1757019	1

Appendix B

BOM: Capsule-Size Prototype

Table B.1: The complete BOM for the whole capsule-size prototype, including the programming PCB. The most important information is included. This BOM was verified and all components sourced by ExceptionPCB. The complete BOM with more detailed information is available in the GitHub repositories describe in Section 4.

Designator	Comment	Suggested Part	Qty
C1, C3	15pF	CL03C150JA3NNNC	2
C2	470nF	EMK063BBJ474KPLF	1
C4, C6, C7, C8	100nF	CL03A104KP3NNNC	4
C5	0.1uF	CL03A104KP3NNNC	6
C9	10uF	CL05A106MP8NUB8	1
C5, C6	360pF	TMK063CG361JT-F	2
L1	100 uH	CB2012T101K	1
LED1, LED2	0201 LED	SML-LX0201NWD-TR	2
R1, R3	25R	TNPW020125R0BEED	2
R2, R4	10k	RT0201FRE0710KL	2
R5	124R	CPF0201B124RE1	1
S1	Camera module	NanEyeC	1
U1	MCU	STM32L051T8Y7DTR	1
J1, J2	Battery Spring	2040852-1	4
SW1	DIP Switch	CHS-01TA	1
U1	3V3 LDO	TLV71733PDQNT	1
U2	2V5 LDO	TLV71725PDQNT	1
U3	Backscatter switch	ADG902BRMZ	1
NA	PCB header 1.27 mm	TS-0769-929647-07-36Q	6
NA	PCB header 2.54 mm	M52-040023V0545	8
programming	Terminal block female	1757242	1
programming	10-pin JTAG connector	FTSH-105-01-L-DV	1

Simen Håbet Ødegård

Wireless Video Capsule Endoscopy Prototype Utilizing a Backscatter Transceiver for High Data Rate Transmission

NTNU
Norwegian University of
Science and Technology