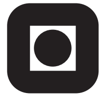


NTNU – Trondheim
Norwegian University of
Science and Technology

Image Processing Module for Autonomous Vehicles

Advanced PCB design, fabrication, and assembly.

Mohammad Ali Koteich

Master of Science in Engineering Cybernetics
Submission date: July 2012
Supervisor: Amund Skavhaug, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Image Processing Module for Autonmous Vehicles

Advanced PCB design, fabrication, and assembly.

MOHAMMAD ALI KOTEICH

NORWEGIAN UNIVERSITY OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF ENGINEERING CYBERNETICS

July 11, 2012

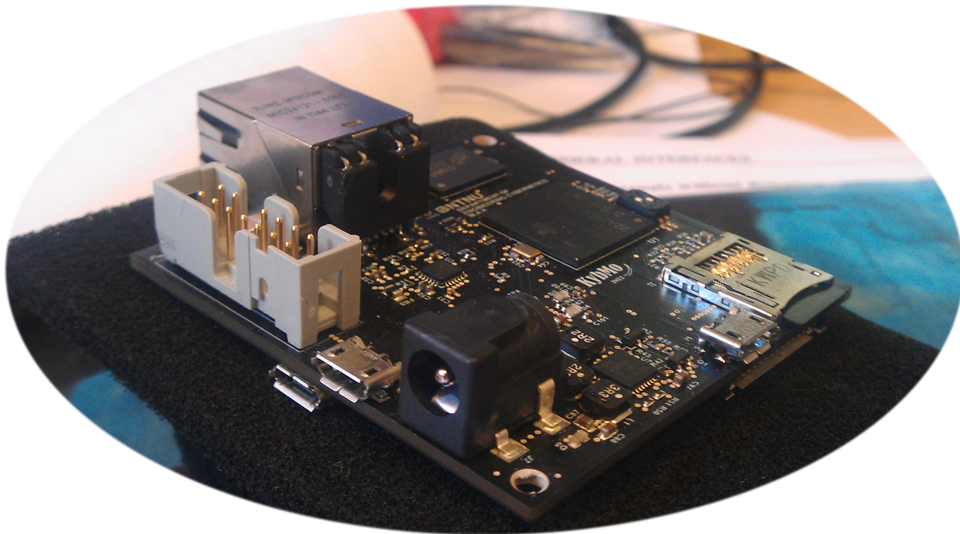


Figure 1: The KybMo board

“Our efforts up till this moment have but turned over a pebble or shell here and there on the beach, with only a forlorn hope that under one of them was the gem we were seeking.“

- **Isaac Newton*[1643- 1727]***

Project description

The project has several objectives unified under one application goal. The application goal is to have a low power system that is capable of performing real time image processing, and which supports connectivity to high speed peripherals. The objectives of the project are:

- Test the waters for how advanced microelectronic systems the facilities and equipment at NTNU can produce.
- Set precedence at The Department of Engineering Cybernetics for high-speed peripheral bus design.
- Develop competence within the field of advanced printed circuit board (PCB) design, which includes working with computer aided design (CAD) tools, multi-layer architectures, small discrete components (1x0.5 mm resitors and capacitors), and advanced chip packages, such a Ball Grid Array (BGA) and Quad Flat No-Lead (QFN) packages.
- Present necessary background theory in a fashion that can be useful for future students.
- Present necessary know-how information in a fashion that can be useful for future student.

Supervisor: Associate professor Amund Skavhaug, Department of Engineering Cybernetics, NTNU.

Summary

The aim of this project is to design a small profile, low-power, high performance printed circuit board (PCB) with support for high speed peripherals, while relying on the equipment and facilities of NTNU as much as possible. The intended usage of this module is in image processing applications for autonomous vehicles. The accomplishment of this goal encompasses gaining experience within proficient usage of computer aided design (CAD) tools, implementation of high speed design theory, fabrication of a multilayer PCB, and assembly of advanced components with small footprints and high pin densities, such as ball grid arrays (BGA) and quad flat no leads (QFN) packages. In addition, validation of functionality and testing is also a part of the process.

There are several reasons for why such a project is important, and the benefits of carrying out such a project can be leveraged in several areas. First of all, it is desirable to get an overview of the production limitations at the NTNU campus. Where does the limit go for how advanced designs we can produce? What is not possible to do at the NTNU campus? How far can we push our capabilities? Then there is the desire of setting precedence within advanced PCB design and production, in order to build competence at the University that the students and researchers can benefit from in the future. All for the purpose of increasing the quality of their education. Furthermore, low power image processing capabilities with support for high speed peripherals are highly desirable in autonomous vehicles and increase the potential for future research.

In order to achieve the goals set for this project, one needs to gain an understanding of what the various components require in terms of design and assembly, and an overview needs to be worked out in order to see how much of these requirements can be met at campus. The process started at finding the correct CAD tool. An attempt was made at using the OrCAD package which The Department of Cybernetics had licenses for. It was discovered that the OrCAD licenses we were in possession of were not suited for the requirements of high-speed bus design. Therefore, an Altium Designer license was borrowed from The Department of Electronics and Telecommunication (IET). Extensive research was done to acquire knowledge about high-speed bus design, which resulted in successful DDR2, Ethernet, and USB bus implementations. The finished design was given the name *KybMo*. The workshop at The Department of Engineering cybernetics is in possession of a milling machine for PCB fabri-

cation. However, this machine can only accommodate fabrication of two layer boards, and not 8 layers, as the design required. Thus the fabrication of the board had to be outsourced to an external manufacturer. The department of Engineering Cybernetics recently acquired a re-flow oven that can accommodate assembly of surface mount components such as BGAs and QFNs, but the oven had never been used before, so no one possessed any experience on how to operate it. Gaining this experience was therefore important and the oven was used with successful results. Hand soldering of extremely small discrete components on the bottom side of the board was carried out, as well as successful testing of alternative methods such as using a heating plate and a heat gun with extremely low air pressure in order to avoid blowing the components away. Software packages including stand alone examples, and a Linux system configured for reference designs based on the same System on Chip (SoC) utilized by the KybMo, the OMAP-L138, were modified to fit the configuration of the KybMo and was used for validation of functionality.

The project resulted in successful fabrication and assembly of a small, low-power, high performance 8 layer PCB, with over two hundred components. All of the high speed buses were validated and the system was successfully able to boot Linux. This proves that it is definitely possible to design and assemble systems of industrial complexity at the NTNU campus. However, the cost was more than expected. In order for student projects such as this one to be economically supported by the Department, an effort has to be made to push the limits of the design so that less than eight PCB layers are required. Otherwise, an attempt at acquiring external sponsorship is instrumental.

Sammendrag

Målet med dette prosjektet er å designe en liten profils, lavt strømforbruk, høy ytelses kretskort (PCB) med støtte for høy hastighets periferiutstyr, ved å benytte seg av utstyret og fasilitetene på NTNU så mye som mulig. Den tiltenkte bruk av denne modulen er i bildebehandlingsprogrammer for autonome kjøretøy. Oppnåelse av dette målet omfatter å få erfaring innen dyktig bruk av dataassistert konstruksjon (DAK) verktøy, implementering av høyhastighets design teori, fabrikasjon av en flerlags PCB og montering av avanserte komponenter med små fotavtrykk og høye pin tettheter, som for eksempel ball grid matriser (BGA) og Quad Flat No lead (QFN) pakker. I tillegg er validering av funksjonalitet, og testing også en del av prosessen.

Det er flere grunner til at et slikt prosjekt er viktig, og fordelene ved å gjennomføre et slikt prosjekt kan utnyttes på flere områder. Først av alt er det ønskelig å få en oversikt over produksjons-begrensningene på NTNUs campus. Hvor går grensen for hvor avansert design vi kan produsere? Hva er ikke mulig å gjøre ved NTNU campus? Hvor langt kan vi presse våre evner? Deretter er det ønske om å sette presedens innen avansert PCB design og produksjon, for å bygge kompetanse på universitetet som studenter og forskere kan dra nytte av i fremtiden. Alt i den hensikt å øke kvaliteten på utdanningen. Energieffektive systemer med bildebehandlings evner og støtte for høyhastighets enheter er svært ønskelig i autonome kjøretøy og øker potensialet for framtidig forskning.

For å oppnå målene for dette prosjektet, trenger man å få en forståelse av hva de ulike komponentene krever i form av design og montering, og en oversikt må utarbeides for å se hvor mye av disse kravene kan oppfylles på campus. Prosessen starter ved å finne riktig CAD-verktøy. Et forsøk ble gjort på å bruke ORCAD pakken som Institutt for Teknisk kybernetikk hadde lisenser for. Det ble oppdaget at de ORCAD lisensene vi var i besittelse av ikke var egnet for kravene til høyhastighets buss design. Derfor ble en ALTIUM Designer lisens lånt fra Institutt for elektronikk og telekommunikasjon (IET). Omfattende forskning ble gjort for å tilegne seg kunnskap om høyhastighets buss design, noe som resulterte i vellykket DDR2, Ethernet og USB-buss implementeringer. Det ferdige designet fikk navnet textit KybMo. Workshopen ved Institutt for teknisk kybernetikk er i besittelse av en fresemaskin for PCB fabrikasjon. Men denne maskinen kan bare imøtekomme fabrikasjon av to lags kretskort, og ikke 8 lag, som dette designet krever. Dermed måtte fabrikasjonen settes

ut til en ekstern produsent. Institutt for teknisk kybernetikk kjøpte nylig en re-flow ovn som kan imøtekomme montering av overflatemonterte komponenter som BGAs og QFNs, men ovnen hadde aldri blitt brukt før, så ingen hadde noen erfaring på hvordan man skal bruke den. Å tilegne seg denne kompetansen var derfor viktig, og ovnen ble brukt med gode resultater. Hånd lodding av ekstremt små diskrete komponenter på undersiden av brettet ble gjennomført, samt vellykket testing av alternative metoder som å bruke en varmeplate og en varmepistol med ekstremt lavt lufttrykk for å unngå å blåse komponentene unna. Programvarepakker inkludert frittstående eksempler, og et Linux system som er konfigurert for et referanse design basert på samme system på Chip (SoC) som benyttes av KybMo, OMAP-L138, ble modifisert til å passe konfigurasjonen av KybMo og ble brukt til validering av funksjonalitet.

Prosjektet resulterte i vellykket fabrikasjon og montering av et lite, lavt strømforbruk, høy ytelse 8 lags PCB, med over to hundre komponenter. Alle høyhastighets bussene ble validert, og systemet var vellykket i stand til å boote Linux. Dette beviser at det er definitivt mulig å designe og sette sammen systemer av industriell kompleksitet ved NTNU campus. Imidlertid kostet det mer enn forventet. For at studentprosjekter som dette skal bli økonomisk støttet av instituttet, må en innsatts gjøres for å presse design grensene, slik at mindre enn åtte PCB lag er påkrevd. Ellers er et forsøk på å skaffe ekstern sponsing avgjørende.

Preface & Acknowledgment

I had never touched a soldering iron in my life prior to this project. One can only imagine how gratified I feel sitting here, 20 weeks later, holding a small form factor advanced circuit board that can be compared, and even compete in terms of performance, with professional products on the market today. Well, 20 weeks might not be the accurate number, as the research process started in a pre-study during the fall semester. Still, it was apparent from the get-go that producing a functional system was a long shot. Thorough research and scrutiny of every aspect was therefore of the utmost importance. Not because I wanted to succeed, but because I wanted to make sure I had something to blame my failure on if anything went wrong!

Joking aside, the first one I would like to thank is God, for opening the right doors in front of me, and for being such a reliable source of support whenever things looked grim. I would also like to thank my friends and family who stood by me and encouraged me to succeed. A special thanks goes out to my supervisor, Amund Skavhaug, who directed me towards this goal, and allowed me to embark on such a risky journey.

My conscience would not allow me to leave out the names of everyone else who has been instrumental for the success of this project. Fortunately, there are a number of knowledgeable and skilled people at NTNU who were willing to lend me their ears, and contribute with help and support. I am forever grateful for all your help! Alphabetically sorted, my sincere appreciation and gratitude goes to:

The Department of Engineering Cybernetics, Institutt for Teknisk Kybernetikk:

Jo Arve Alfredsen
Terje Haugen
John Olav Horrigmo
Rune Mellingseter
Karsten Rennæs
Per Inge Snildal

Department of Electronics and Telecommunications, Institutt for Elektronikk og Telekommunikasjon:

Ingulf Helland
Terje Mathiesen
Andreas Østensen (Special thanks to you for mentoring me on some of the practical challenges)

Mohammad Ali Koteich
July 2012, NTNU Trondheim

Contents

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Background . . . 1
1.3 Scope . . . 2
1.4 The Solution . . . 3
1.5 Specifications . . . 4
1.6 Existing Solutions . . . 5
1.7 Project Webpage . . . 6
1.8 Attachments . . . 6

2 Report structure **7**
2.1 Development & Computer Aided Design tools . . . 7
2.2 DDR2 Memory Interface . . . 7
2.3 Power & Reset . . . 7
2.4 Peripheral Interfaces . . . 8
2.5 Fabrication . . . 8
2.6 Assembly . . . 8
2.7 Board Validation, Debugging, and Setup . . . 8
2.8 Discussion & thoughts for the future . . . 8
2.9 Conclusion . . . 8
2.10 Appendices . . . 8

3 Development & Computer Aided Design tools **10**
3.1 Introduction . . . 10
3.2 The Basics . . . 10
3.2.1 Schematics . . . 10
3.2.2 Layout . . . 11
3.3 OrCAD . . . 12
3.4 Altium Designer . . . 12
3.5 Discussion & Personal Experience . . . 13

4 DDR2 Memory Interface **15**
4.1 Initial assessments . . . 15
4.2 The Schematics . . . 16
4.3 The Layout . . . 16
4.3.1 Simulation . . . 17
4.3.2 Signal timing constraints . . . 18
Net Length Matching- OrCAD . . . 19
Net Length Matching- AD(Altium Designer) 10 . . . 22
4.3.3 Signal integrity & Impedance Control . . . 26
Impedance discontinuities . . . 26
Termination . . . 26
4.3.4 Impedance discontinuities, difference between theory and practice . 27
4.3.5 Crosstalk . . . 29
4.3.6 Placement . . . 31
4.3.7 BGA breakout . . . 36
4.3.8 Layer Stack-Up . . . 38
Important considerations . . . 39
Layer Configuration . . . 40
4.3.9 Bypass and decoupling capacitors . . . 41

5 Power &Reset **42**
5.1 Power supply & distribution . . . 42
5.1.1 Required Voltage Levels . . . 42
5.1.2 Architecture . . . 43
5.1.3 Power & Ground planes . . . 44
5.2 Reset . . . 46

6 Peripheral Interfaces **47**
6.1 USB . . . 47
6.1.1 USB Interface . . . 48
6.1.2 USB data signal routing . . . 48
6.1.3 Differential pair routing of the USB2.0 interface in Altium . . . 49
6.1.4 Connector . . . 52
6.2 Ethernet . . . 53
6.2.1 Connector & Magnetics . . . 54
6.2.2 Strapping resistors . . . 55
6.2.3 Topology & Placement . . . 56
Deviation from the topological rules . . . 59
6.2.4 Routing Guidelines for the TX and RX Differential Pairs . . . 61
6.2.5 Filtering Analog Vcc . . . 62
6.3 microSD . . . 62
6.4 JTAG . . . 64

6.5 UART . . . 66
6.6 I/O Expansion Header . . . 68
6.7 Chassis Ground/Electro Static Discharge (ESD) Ring . . . 71

7 Fabrication **74**
7.1 Introduction . . . 74
7.2 PCB manufacturing . . . 74
7.2.1 Limitations of the manufacturer . . . 75
7.3 Ordering . . . 77
7.3.1 Finishes . . . 77
7.3.2 Price . . . 79
7.4 Post Ordering Communication . . . 79
7.4.1 Impedance Control . . . 79
7.4.2 VIA tenting . . . 82

8 Assembly **83**
8.1 Moisture Sensitivity . . . 83
8.2 Solderpaste Stencil . . . 85
8.2.1 Stencil Printing . . . 86
8.2.2 The "stencil" layer . . . 87
8.2.3 Special Paste mask considerations for QFN packages . . . 88
8.2.4 Stencil thickness . . . 91
8.2.5 Ordering the Stencil . . . 92
8.2.6 Home-brewed Stencil Printer . . . 94
8.2.7 Alignment of the PCB and the Stencil . . . 95
8.2.8 Printing . . . 96
8.3 Component Placement . . . 97
8.4 Reflow Soldering . . . 98
8.4.1 Reflow Profile . . . 99
8.4.2 Reflow Process . . . 100
Problems under reflow . . . 101
8.5 Hand Soldering . . . 102
8.5.1 Using a heat gun and a heating plate . . . 106
8.5.2 Advanced Package Rework (APR) machine . . . 108

9 Board Validation, Debugging, and Setup **110**
9.1 Introduction . . . 110
9.2 Validation Overview . . . 110
9.3 Booting . . . 111
9.4 Validating UART . . . 112
9.5 Validating JTAG . . . 114
9.6 Validating MicroSD, Booting U-Boot & Linux . . . 117

9.6.1 U-Boot 119
9.6.2 Configuring the microSD card 122
9.6.3 Modifying the Linux Kernel 123
9.6.4 Modifying for UART0 124
9.7 USB 125
9.7.1 USB 1.1 OHCI: 125
9.7.2 USB 2.0: 125
9.8 Ethernet 127
9.9 Memory 129
9.10 Power consumption 130
9.11 Summary of lessons learned 131
9.12 Resources 132
9.12.1 Linux SDK 132
9.12.2 Uboot 132
9.12.3 Linux Kernel 133
9.12.4 SD Card 133
9.12.5 Practical TI Wiki Pages/Literature 133
9.12.6 Software used 134
9.13 Configuration & Set-Up 134
9.13.1 Power 134
9.13.2 Serial Communication 135
9.13.3 Booting 136
9.13.4 USB 2.0: 136

10 Discussion and Thoughts for the future 137

A Advanced Multilayer & High-Speed Printed Circuit Board Design 141
A.1 Introducion 141
A.1.1 High Frequency & High Speed, what's the difference? 142
A.1.2 High Speed/High Frequency Peripheral buses 142
A.2 Multilayer Printed Circuit Boards 143
A.2.1 Why multilayer? 144
Radiation 144
Ground & Power planes? 144
Signal Escape 145
A.2.2 Layer stack-up 145
4-Layers 145
6-Layers 146
A.2.3 Recommended Reading 146
A.3 Integrated Circuit Packages 147
A.3.1 Ball Grid Array(BGA) 147
BGA Assembly Process 148

Assembly Challenges 148
Detecting Assembly Faults in BGA packages 149
A.3.2 Recommended reading 150
A.4 Routing 151
A.4.1 Flight time delay and skew 151
A.4.2 Signal integrity and impedance matching 151
A.4.3 Crosstalk 152
A.4.4 Power supply bypassing 153
A.4.5 EMI (Electro Magnetic Interference) 154
A.4.6 Recommended Reading 155
A.5 Manufacturing Capabilities 156
A.5.1 Multilayer PCBs 156
A.5.2 BGA Assembly 157
A.5.3 Tools 157
A.5.4 Recommended Reading 158
A.6 Literature & Resources 159
A.6.1 Recommended Reading 159
High Speed PCB Design 159
Signal Integrity and Controlled Impedance 159
Multilayer Stack-Up & EMI 159
Ball Grid Array Package & Soldering 159
PCB manufacturing details & Materials 160
Boundary Scan Chain 160
Software Tools & Tutorials 160
A.6.2 Reference Designs 161
A.7 References 162

B Terminology **165**

List of Figures

1 The KybMo board 1

1.1 KybMo Block diagram 3
1.2 KybMo Layout 4
1.3 From left to right: Beagleboard, Beaglebone, KybMo 5

3.1 Schematic Drawing (KybMo power siwtching) 11
3.2 Footprints and connections (KybMo, Ethernet) 12

4.1 Compatible DDR2/mDDR Devices [T.I.L-138, 2009, p. 134] 15
4.2 DDR2 net class definitions, [T.I.L-138, 2009, p. 128] 17
4.3 Skew Length Mismatch rules [T.I.L-138, 2009] 19
4.4 Skew Length Mismatch, various net classes [T.I.L-138, 2009] 20
4.5 Etch length report, OrCAD PCB Editor 20
4.6 Attempts at track length matching, OrCAD PCB Editor. 21
4.7 Net classes in the project. AD 10 23
4.8 Design rules, matched net lengths. AD 10 24
4.9 Route nets directly, different colors indicate signals on different layers. AD 10 24
4.10 Interactive Length Tuning, the bar is red indicating that the particular track is too short and needs serpentine patterns. AD 10 25
4.11 Pattern properties. AD 10 25
4.12 Series termination, www.electronicdesign.com 26
4.13 90 degree turns in serpentine pattern 28
4.14 Recommended spacing parameters, [T.I.L-138, 2009] 29
4.15 Signal spacing, AD10, KybMo 30
4.16 Placement recommendations, [T.I.L-138, 2009] 31
4.17 Long side of chip on horizontal axis. Several attempts were made at optimally routing the design 32
4.18 Long side of chip on horizontal axis. ADDR_CTRL class routed. Longest track: 1891 mils 32
4.19 The Beaglebone, SoC and Memory chips. 33
4.20 Vertical orientation. 34

4.21 Entire PCB Area 35
4.22 Center to center spacing was decreased with 415 mils in the final attempt. . 35
4.23 OMAP-L138, Ball Grid Array configuration 36
4.24 "Rats nest" lines indicating pin to pin connections 37
4.25 Complete BGA breakout of both devices for the DDR2 interface. 38
4.26 6 layer minimum stack up recommendation, [T.I.L-138, 2009] 39
4.27 8 layer stack-up, [Ott, 2000] 40
4.28 HS bypass capacitor recommendations, [T.I.L-138, 2009] 41

5.1 KybMo, TPS650250 Powersupply schematics 43
5.2 Zener Diode for over-voltage protection 44
5.3 KybMo, plane and layer partitioning 45
5.4 KybMo, reset generation schematics 46

6.1 The USB Interface, wikipedia.org/Universal_Serial_Bus 48
6.2 Altium differential pair routing tool 49
6.3 Altium Differential Pair Editor 50
6.4 Y1: 24MHz Crystal, Blue lines: USB2.0 Differential pair 51
6.5 Circumventing the crystal in the KybMo design. 52
6.6 LAN8720AI PHY. 53
6.7 KybMo, Ethernet schematics. 54
6.8 RJ45 with integrated magnetics, WE-LAN 7498011121 55
6.9 Mode selection straps (Mode[2:0] = 011. 100Base-TX, full duplex, auto negotiation disabled 56
6.10 Mode selection straps (REGOFF = 0, regulator enabled. nINTSEL = 1, no sourcing of clock. 56
6.11 Ethernet Interface Layout. Green square denotes the absence of ground and power planes under the connector, and sections of the differential traces. . . 58
6.12 KybMo PHY Distance from RJ45 connector and board edge (Altium Designer) 60
6.13 Beaglebone PHY Distance from RJ45 connector and board edge (Allegro Free Physical Viewer) 60
6.14 TX and RX pairs 61
6.15 KybMo, ferrite bead for Vcc filtering 62
6.16 microSD schematics 63
6.17 KybMo microSD Connector: Molex 5027740891 63
6.18 TI 14 pin header pin-out overview 64
6.19 KybMo, JTAG interface schematics 65
6.20 TI 14 Pin Header,3M MHB14K-ND 65
6.21 Spectrum Digital XDS100v2 emulator 65
6.22 DE9 connector, female. www.solentcables.co.uk 66
6.23 KybMo, USB to UART FT232 schematics 67
6.24 KybMo (bottom side), FT232RQ 67

6.25 KybMo (bottom side), JAE electronics FI-WE31P-HFE with plastic mate FI-W31S 68
6.26 I/O expansion header block diagram 68
6.27 I/O expansion header schematics 69
6.28 KybMo, signal layers had to be filled with a polygon around the board edge, while power and ground planes had to be split 72
6.29 The contours of the ESD ring can be seen on the KybMo PCB 73

7.1 Standard PCB capabilities, www.pcbcart.com 75
7.2 Special PCB capabilities, www.pcbcart.com 76
7.3 KybMo PCB order specification, www.pcbcart.com 77
7.4 KybMo PCB cost, www.pcbcart.com 79
7.5 Impedance control information sent to manufacturer 80
7.6 Calculated Impedance values from manufacturer 80
7.7 Modified parameters sent to manufacturer 81
7.8 The KybMo PCBs that were delivered. 82

8.1 The acid-room(syrerom) at The Department of Engineering Cybernetics, where the reflow-oven is located. 83
8.2 NTNU, baking oven at the Electronics and Prototype (ELPRO) Lab 84
8.3 KybMo OMAP-L138 IC, humidity indicator card 85
8.4 Stainless steel PCB stencil, www.pentalogix.com 86
8.5 LPKF ProtoPrint S, stencil printer 86
8.6 KybMo, Paste Layers in Altium Designer 87
8.7 KybMo, Paste Layers in Altium Designer 88
8.8 TI TPS650250 IC used on the KybMo 89
8.9 High-centering due to too much solder on the thermal pad 89
8.10 KybMo, TPS650250 footprint, partitioned thermal pad (the four gray squares) 90
8.11 TPS650250 via tenting in Altium Designer by creating holes over the vias in the Top Solder layer (purple) 91
8.12 KybMo, TPS650250 pads after solder paste printing. The black spots on the thermal pad shows how the vias are tented with solder mask 91
8.13 Creating a steel stencil with the LPKF milling machine at the ITK workshop 93
8.14 The attempt failed 93
8.15 KybMo, Polymex Polyethylene Terephthalate stencil 94
8.16 Attempt at creating a cardboard stencil printer frame 94
8.17 The final stencil printer design 95
8.18 Alignment of PCB and Stencil 95
8.19 Solder paste printing by use of rubber squeegee 96
8.20 First attempt, messy results 96
8.21 Placement of components 97
8.22 The workstation in the acid-room at the ITK Department 98

8.23 LPKF ProtoFlow E, ITK Department NTNU 99
8.24 Misaligned chamber causes drawer to grind against chamber walls. 101
8.25 KybMo after the reflow process. 102
8.26 Workstation with camera, ITK Workshop 103
8.27 Microscope, IET ELPRO Lab 103
8.28 Induction based soldering iron 104
8.29 Hand soldered 0402 capacitors 105
8.30 Fixing short circuits 105
8.31 heating plate 106
8.32 Heat gun settings 107
8.33 Results 107
8.34 Fixing the USB to UART connector 108
8.35 IET ELPRO lab, Metcal APR-5000 108

9.1 Texas Instrument's AISgen tool 112
9.2 UART Boot switch settings 112
9.3 UART Boot Host Utility 113
9.4 Hello World UART application 114
9.5 JTAG Boot Switch Settings 115
9.6 JTAG Code Stepping 116
9.7 MMC/SD Boot Switch Settings 117
9.8 U-Boot Test Environment, external FT232 board borrowed from co-student, Karsten Rennæs 120
9.9 Memory Register Spread Sheet 121
9.10 Kybmo DDR2 Register configuration with HawkBoard values. 122
9.11 U-Boot Console 122
9.12 Kernel files tagged with the KybMo string. 123
9.13 TI's Arago distribution running on KybMo. 124
9.14 Linux console output upon detecting a USB flash drive on USB 1.1 port. . . 125
9.15 Forcing 5V sourcing by wire patch. The point closest to the IC is connected to the 5V plane. The other point is connected to the VBUS signal on the USB 2.0 port on the other side of the board. 126
9.16 Linux console output upon detecting a USB flash drive on USB 2.0 port. . . 126
9.17 Browsing and writing to the USB flash drive connected to the USB 2.0 port. 127
9.18 Kernel discovering the Ethernet PHY. 128
9.19 Pinging www.ntnu.no 129
9.20 mtest command in U-Boot. 129
9.21 "free" command displaying memory info. 130
9.22 **1:** wire connecting the WDO signal from U8 to 3.3V **2:** Plastic tape isolating the WDO pin from the pad. 135
9.23 Tera Term Serial Port configuration. 136

A.1 Multilayer PCB (www.kursus.com) 143
A.2 Vias(wikipedia) 143
A.3 4-Layer stack-ups 145
A.4 Common & acceptable stack-up 146
A.5 IC Packages(www.sparkfun.com) 147
A.6 BGA package(www.sparkfun.com) 147
A.7 PCB signal traces (www.locxtronic.com) 151
A.8 Schematic of a typical bypass capacitor configuration (From the Logic-PD SOM M1 reference design files) 153
A.9 PCBcart capability chart (www.pcbcart.com) 156
A.10 Metcal APR-5000 157

B.1 Annular ring = X - Y, wiki.altium.com 165

List of Tables

1.1 KybMo Specifications 4

8.1 Moisture Sensitivity Levels, defined in IPC/JEDEC J-STD-20 85
8.2 Stainlees steel stencil prices, www.pcbcart.com 92

List of Abbreviations

- **AD:** Altium Designer
- **ARM:** Advanced RISC Machine
- **BGA:** Ball Grid Array
- **CAD:** Computer Aided Design
- **CPU:** Central Processing Unit
- **CS:** Chip Select
- **CV:** Computer Vision
- **DDR:** Double Data Rate
- **DSP:** Digital Signal Processor
- **e2e:** Engineer to Engineer
- **eCAP:** Enhanced Capture
- **ELPRO:** Elektronikk og Prototype Lab
- **ENIG:** Electroless Nickel Immersion Gold
- **EPWM:** Enhanced Pulse Width Modulation
- **GND:** Ground
- **GP:** General Purpose
- **HASL:** Hot Air Solder Leveling
- **HS:** High-Speed
- **I2C:** Inter-Integrated Circuit
- **IET:** Institutt for Elektronikk og Telekommunikasjon
- **I/O:** Input/Output

- **IC:** Integrated Circuit
- **IPC:** Industry association for Printed circuit board and electronics manufacturing service Companies
- **ITK:** Institutt for Teknisk Kybernetikk
- **JTAG:** Joint Test Action Group
- **LED:** Light Emitting Diode
- **MMC:** Multi Media Card
- **MSL:** Moisture Sensitivity Level
- **NTNU:** Norges teknisk-naturvitenskapelige universitet
- **OHCI:** Open Host Controller Interface
- **OSP:** Organic Solderability Preservative
- **PC:** Personal Computer
- **PCB:** Printed Circuit Board
- **PLL:** Phase Locked Loop
- **PSC:** Power Sleep Controller
- **PWM:** Pulse Width Modulation
- **QFN:** Quaf Flat No-lead
- **RAM:** Random Access Memory
- **RISC:** Reduced Instruction Set Computing
- **SD:** Secure Digital
- **SDK:** System Development Kit
- **SDRAM:** Synchronous Dynamic Random Access Memory
- **SMSC:** Smart Mixed Signal Connectivity
- **SoC:** System on Chip
- **SPI:** Serial Peripheral Interface
- **TI:** Texas Instruments
- **UART:** Universal Asynchronous Receiver/Transmitter
- **USB:** Universal Serial Bus

- **WE:** Würth Electronics

Chapter 1

Introduction

1.1 Motivation

Visual capabilities is a useful feature in autonomous vehicles. The applications can range from visual feedback for vehicle control to pattern recognition and visual analysis. Such functionality depends on hardware with sufficient computational power and support for a certain combination of high speed peripheral interfaces. At the same time, the hardware needs to adhere to the restrictions introduced by the nature of a small autonomous vehicle, such as low power consumption, small physical profile, and preferably, low cost. It is of interest to know how capable we are of producing advanced and demanding circuitry to meet the needs of such advanced applications at the IME faculty at NTNU.

1.2 Background

In the fall of 2011, a pre-study was conducted by me and a fellow student, Karsten Rennæs, with the aim of suggesting an overall technical UAV solution. An important part of the overal solution was the system's image processing capabilities. A thorough assessment was made in regards to the available off-the-shelf solutions and a conclusion was reached to design an in-house module, based on the OMAP-L138 System on Chip (SoC) in order to best meet the requirements listed below:

Application requirements:

- Low power
- Small size
- Digital Signal Processor

- Computationally powerful
- Rich support for various high-speed peripherals
- Easy to utilize the DSP. Support for the popular open source computer vision library OpenCV
- As affordable as possible

Academic requirements:

- Gain experience and knowledge within high-speed peripheral bus design, and advanced PCB production.
- Get an overview of the practical limits and capabilities of in-house designing and assembly at NTNU.
- Document the process so the department and future students can benefit from the lessons learned, and use the knowledge gained thoughout the project in organizing future projects and courses.

1.3 Scope

The scope of the project is to focus on presenting the know-how needed to produce an advanced professional circuit board tailored to meet the needs of a low-energy image processing application, and to document the development process every step of the way. This includes topics such as high speed peripheral interface design, routing challenges on dense circuit boards populated by fine pitch components, power supply design, fabrication and assembly challenges, and validation.

1.4 The Solution

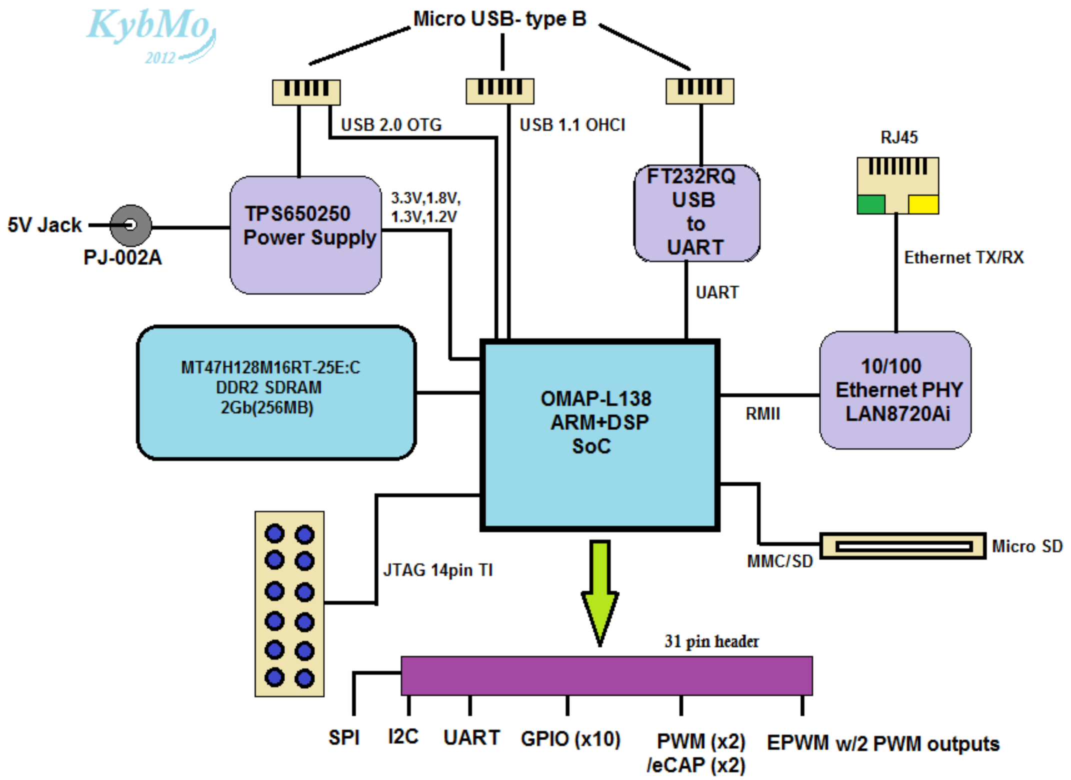


Figure 1.1: KybMo Block diagram

1.5 Specifications

Processing unit	OMAP-L138 SoC, ARM + DSP @ max 456MHz
Memory	2Gb (256MB) DDR2 SDRAM, Micron MT47H128M16
Peripheral interfaces	Ethernet 10/100, USB2.0/1.1, MMC/SD, UART to USB, SPI, I2C, UART, PWM/EPWM/eCAP, Up 30 general purpose I/O ports
Size	5.97x6.11 cm
Power Consumption	Max 1.9W (380mA) observed during validation

Table 1.1: KybMo Specifications

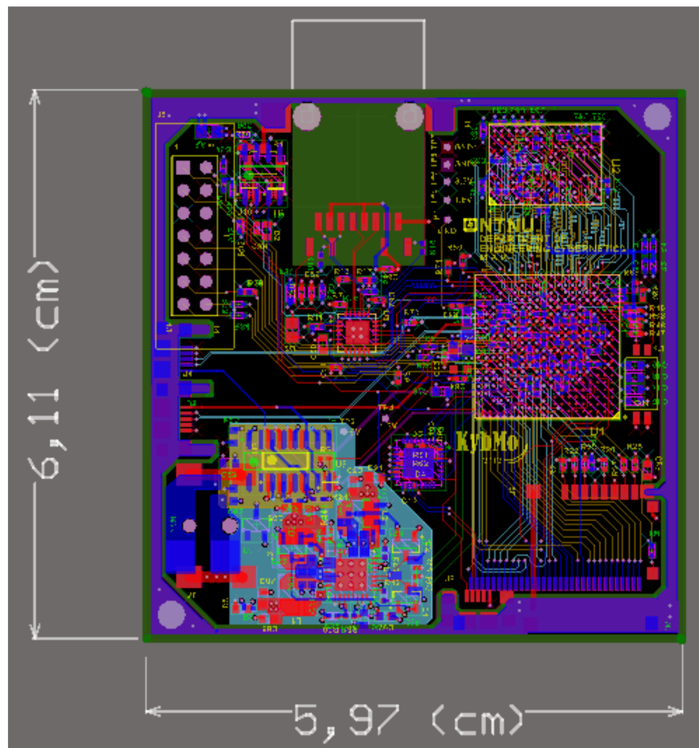


Figure 1.2: KybMo Layout

1.6 Existing Solutions

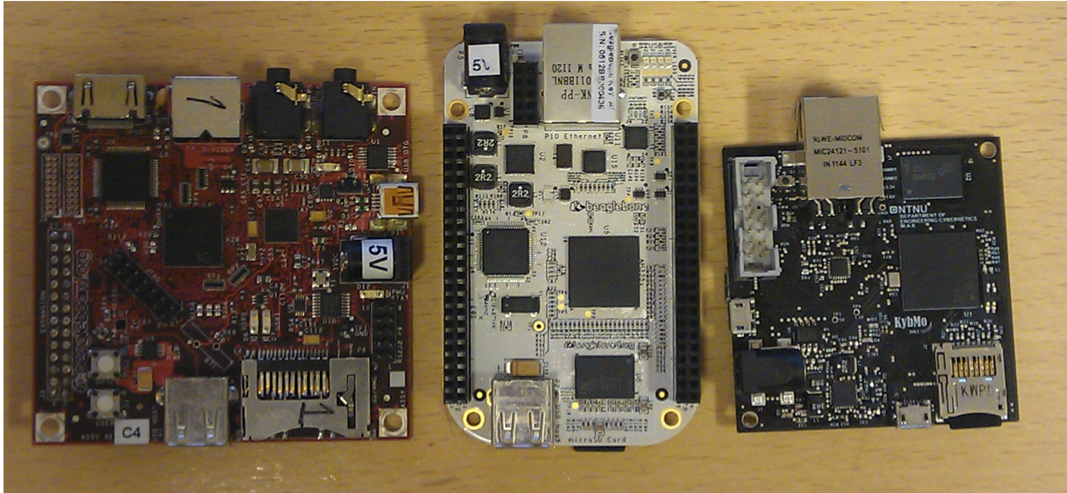


Figure 1.3: From left to right: Beagleboard, Beaglebone, KybMo

There exist no solution that combines the traits of the KybMo and that is customized for the project's intended application purposes. The closest comparisons can be made to popular high-performance, general purpose, embedded development platforms. These are: Beagleboard -XM, LCDK/Hawkboard, Beaglebone, Raspberry Pi.

The Raspberry Pi supports strong multimedia capabilities, with a 700MHz ARM, and a dedicated GPU. The price is around 25-35 USD at the moment. The formfactor is 8.5x5.4 cm. This is the cheapest and device of the lot. However, the fact that there is no dedicated DSP means that one would have to implement computer vision applications on the GPU for best performance. This entails using OpenGL for the implementation, and while that can be an interesting project, the fact is that it deprives the developer from the possibility of utlizing the OpenCV library which is dedicated for computer vision applications. This is not the case with TI's ARM+DSP architectures, where there are tools for rapid DSP development. There are even tools that easily allow the developer to convert C code to the DSP platform[1].

Both the LCDK and the Hawkboard use the same SoC as the KybMo, but they are much larger in size(9x10 cm), and support a variety of unnecessary peripherals (in terms of our application goals) such as Mic in, Line out, VGA out, and LCD out.

As for the Beagleboard -XM and the Beaglebone, the Beaglebone is only equipped with an ARM, so there is no DSP to utilize. The Beagleboard -XM is based on the DM3730 1GHz

[1] `http://processors.wiki.ti.com/index.php/C6EZ_Tools`

ARM+ 800MHz DSP, which is a more powerful system than the OMAP-L138. However, it's also larger with it's 7.8x7.6 cm, and not as power effecient. Furthermore, the DSP on the Beaglebone is a fixed-point DSP, while the KybMo has a fixed-and floating point DSP.

1.7 Project Webpage

Information about the project, and relevant files can be found on:

http://stud.itk.ntnu.no/kybmo

1.8 Attachments

All relevant schematic, design, and system files can be found on the CD, and will not be included at the end of the paper as attachments due to their sizes. The same applies to all the software that was tested.

The CD also includes demonstration videos showcasing the functionality of the system, as well as some aspects of the assembly process. The videos can be found in the assembly and validation folders. The videos titled "0 Full DEMO" demonstrates most of the major capabilities and are recommended to watch.

Chapter 2

Report structure

The report is written in a somewhat tutorial-oriented fashion in order to preserve the experiences and lessons learned in the best way possible, so that future students can use that to their benefit. Discussions are also carried out throughout the project whenever they're relevant instead of saving everything to the end. A compendium that was written in the fall of 2011, titled "Advanced multilayer & high-speed printed circuit board design" is included as an appendix. It is recommended to read through the compendium in order to get a basic understanding of some of the relevant topics, as some parts of the report might assume that the reader knows the basics of what is being discussed.

2.1 Development & Computer Aided Design tools

This chapter discusses the computer aided design tools that were used for creating the schematics and board layout.

2.2 DDR2 Memory Interface

This chapter presents the DDR2 interface, and describes how it was designed and implemented in the layout. All relevant practical and theoretical theory is also discussed.

2.3 Power & Reset

A chapter that discusses and presents the power supply solution used on the system.

2.4 Peripheral Interfaces

This chapter discusses the design of all the peripheral interfaces that the system supports, such as Ethernet, USB, USB to UART, SPI, PWM, etc..

2.5 Fabrication

This chapter discusses the printed circuit board (PCB) ordering and fabrication process of the project.

2.6 Assembly

This chapter discusses the assembly of the components and the methods and techniques used, along with relevant background theory.

2.7 Board Validation, Debugging, and Setup

This section describes the validation process, show cases the results that were produced, and describes the system configuration and setup. Also includes a brief discussion on the lessons learned and how these should be into consideration in the future.

2.8 Discussion & thoughts for the future

This section includes a final discussion with suggestions for a future roadmap.

2.9 Conclusion

This section concludes the project.

2.10 Appendices

- A compendium on high-speed design: Introduces the most important aspects of advanced PCB and high speed peripheral bus design.

- Terminology: This is a short chapter that introduces some of the terminology and basic concepts that may be encountered later on in the report.

Chapter 3

Development & Computer Aided Design tools

3.1 Introduction

A designer's capabilities depend heavily on what kind of software is being utilized. Design software are usually called "CAD" tools, which stands for Computer Aided Design. Two industrial and well known PCB CAD tools were utilized during the course of this project. OrCAD 16.2 and Altium Designer 10. Both these tools are too expensive for hobby purposes, and are primarily aimed at the professional market.

3.2 The Basics

There are two main elements in circuit board design: Schematics, and Layout.

3.2.1 Schematics

Schematics are the files produced in the first part of the design process, where the connections between all the components are systematically mapped. A schematic abstracts the circuit and is concerned with representing the connections and the setup of all the components in a systematic and graphical manner. It does not attempt to make a mechanical or topological description of the circuit. This means that no mechanical sizes, such as component and electrical track dimensions are graphically described, and no information is given concerning the placement and orientation of the components. However, each component in

the circuit is represented by a so called "Symbol" in the schematics. This symbol is usually linked to mechanical information, called a "footprint", describing the dimensions of the component. When the schematics are finished, something called a "netlist" is generated. This netlist is loaded into the Layout tool, and is used to link the correct pins of each footprint together so that the designer knows how to physically route the design.

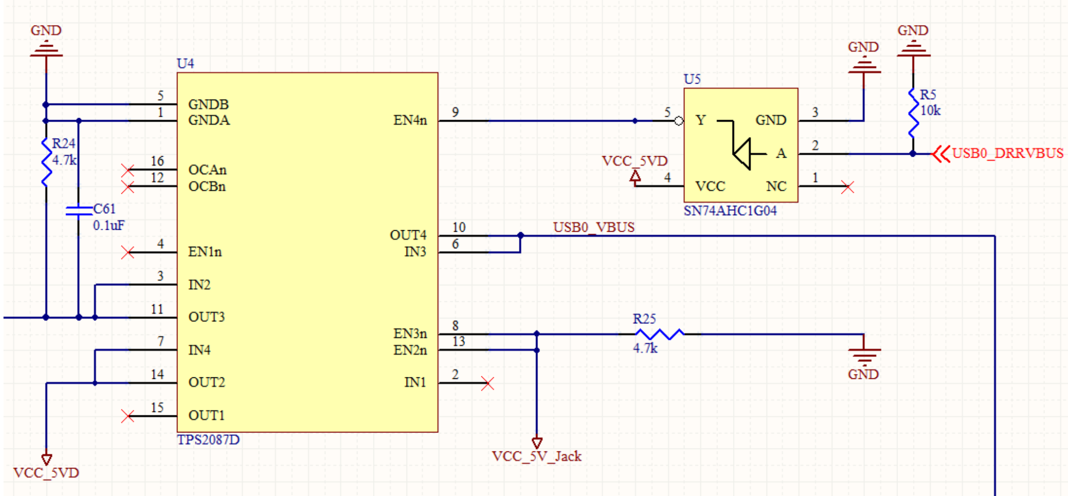


Figure 3.1: Schematic Drawing (KybMo power siwtching)

3.2.2 Layout

The layout tool is the environment where the connections are physically drawn. The components are represented by footprints. A footprint is the geometric layout of the copper pads that the pins of a component are supposed to get soldered to.

The layout tool controls all mechanical and topological aspects of the circuit, such as track width, length, board dimensions, component dimensions, topological placement, holes, and solder mask. Professional layout tools also have a support framework for design rules and constraints that can be set up to ensure that the routing stays within the allowed restrictions. Examples of such rules are: minimum and maximum track width, component spacing, hole clearances, and high-speed design rules.

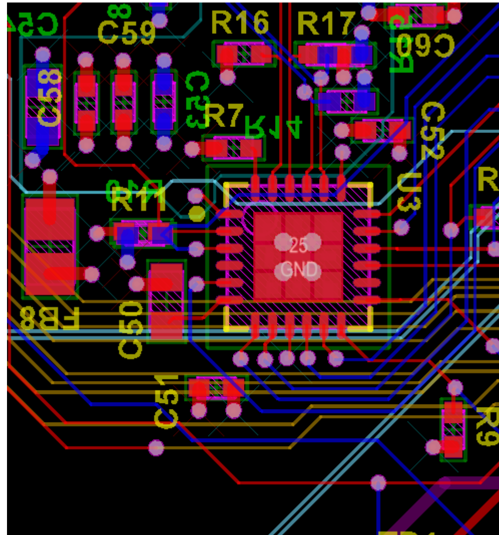


Figure 3.2: Footprints and connections (KybMo, Ethernet)

3.3 OrCAD

The Department of Engineering Cybernetics owns OrCAD 16.2 licences, which made it the first CAD tool of choice. Produced by Cadence, one of the most popular companies in the business of circuit board design, OrCAD is a package of separate tools, where each one is specifically tailored for a certain step in the design process.

The two most essential tools are:

- OrCAD Capture - Schematics
- OrCAD PCB Editor - Layout

PCB Editor is only an "entry level" version of Cadence's full fledged layout tool, Allegro. This means that certain important features are not available, such as exporting high-speed design rules from the schematics to the PCB Editor. An Allegro license must be purchased in order to gain full functionality. Allegro is widely used in the industry, and is often one of the few tools companies will provide component libraries for.

3.4 Altium Designer

The Department of Electronics and Telecommunication(IET, institutt for elektronikk of telekommunikasjon) owns 8 floating licenses for Altium Designer 10. Unlike OrCAD,

Altium integrates all the different parts of the design process in one single package. The program also has an import wizard that makes it easy to migrate from other widely used tools, such as Cadence's OrCAD and Allegro. Full functionality is granted out of the box and the user is instantly introduced to a set of well written and intuitive tutorials.

3.5 Discussion & Personal Experience

Seeing as how The Department of Engineering Cybernetics(ITK, Institutt for Teknisk Kybernetikk) owned OrCAD licenses that hadn't been used a lot in the past, it was of interest to find out how much could be achieved with the tool and what it could be used for. In addition, it was the only PCB design tool the department owned a license for. Naturally, that lead to OrCAD becoming the default development tool.

OrCAD was the first PCB design tool I had ever used, and learning how to use it was a complicated process due to the difficulty of properly trying to understand the available documentation. After a while, it became apparent that OrCAD's capabilities were too limited for this project. First of all, the license that the department owned was for version 16.2 from 2006, a six year old version. Furthermore, it was discovered that critical functionality was not available for the license we had. This functionality was the support of high-speed bus design rules, which was needed to route the DDR2 traces in accordance to the restraints and rules specified in the datasheet of the OMAP-L138. After a lot of trial and error attempting to route the DDR2 traces without high-speed design tools, no progress was being made, and a conclusion to leave OrCAD was reached. It was an unfortunate decision that had to be made in order to secure the progress of the project. The reason it was unfortunate was because it meant that a lot of time was wasted learning a tool that eventually proved to be insufficient. It also meant that some time had to be used on learning an entirely new tool from scratch. However, it wasn't all a waste of time, first of all we could finally conclude that the license owned by the department is insufficient for designing advanced circuitry with high-speed peripheral buses. Secondly, the basic concepts that were learned from using OrCAD was helpful and eased the learning process of the new tool.

I had heard rumors that there was a tool other than OrCAD that could be used for the purposes of the project. That tool was Altium Designer 10. I talked to students that had some experience with it and asked about the high-speed capabilities. No one had worked with high-speed design but they were aware of the existing capabilities. I conducted some research on the internet and quickly confirmed their statements. The next step was to find someone on the NTNU campus who had a license for Altium. The first place I looked was the Department of Computer and Information Science (IDI, institutt for datateknikk of informasjonsvitenskap). The contact person from IDI informed me that they had Altium licenses but was unsure of which version they had. Awaiting more details, I visited the Elpro-lab at the department of electronics and telecommunications, where I learned that they had Altium

10 licenses and that I could borrow a license from them.

Migrating to Altium was simplified by the import wizard, which allowed me to import the schematics I had made for the DDR2 interface in OrCAD. After spending a little over a week on installing and familiarizing my self with Altium, the high-speed design rules were configured and the routing process started from scratch. Fortunately, the rate of progression increased and results were finally being produced.

However, this didn't mean that OrCAD was abandoned all together. As mentioned previously, Cadence's tools are among the few tools that companies provide libraries and reference designs for. Two products were used as PCB layout reference designs, namely the Hawkboard, and the Beaglebone. OrCAD's PCB Editor was used to inspect the Hawkboard design whenever there was a physical routing issue I needed to get a clearer idea of. Studying other designs proved very helpful in gaining a better understanding on how to practically implement certain things, such as power islands under headers, and the placement of decoupling capacitors. The Beaglebone design was also used for this purpose, however, OrCAD v16.2 was too old to read the Beaglebone design files. Fortunately, Cadence provides a free viewer utility called "Allegro Free Physical Viewer" which allows the user to view any design file produced with the latest version of Allegro.

Chapter 4

DDR2 Memory Interface

The memory interface was the first part of the solution that was implemented. It was also the most time consuming part of the project. Since the memory interface is physically the most intricate and complex high-speed interface to implement, it played an integral part in setting the physical requirements for the rest of the board, such as layer stack-up and board space. This chapter discusses the assessments that were made, and describes the technical challenges involved in the process of implementing the DDR2 memory interface.

4.1 Initial assessments

The memory that was chosen for the system is a 256MB DDR2 SDRAM module from Micron Electronics: **MT47H128M16RT**. The system clock frequency of the DDR2 unit is intended to be set at 150-156MHz, as it's the maximum clock rate the SoC specifies for the memory. The IC runs at 1.8V input power. This chip comes in a BGA package with 84 balls. The price of one such chip is 28 USD from Digikey.

It should be noted that designing the system with a SO-DIMM socket to support off-the-shelf laptop memory components was considered. However, the OMAP-L138 datasheet specifies a maximum device bit width of 16 bits for the memory devices:

NO.	PARAMETER	MIN	MAX	UNIT
1	JEDEC DDR2/mDDR Device Speed Grade(1)	DDR2/mDDR-400		
2	JEDEC DDR2/mDDR Device Bit Width	x8	x16	Bits
3	JEDEC DDR2/mDDR Device Count(2)	1	2	Devices

(1) Higher DDR2/mDDR speed grades are supported due to inherent JEDEC DDR2/mDDR backwards compatibility.
(2) Supported configurations are one 16-bit DDR2/mDDR memory or two 8-bit DDR2/mDDR memories

Figure 4.1: Compatible DDR2/mDDR Devices [T.I.L-138, 2009, p. 134]

Unfortunately, SO-DIMM memory components only come in 64 bit width configurations, and are thus rendered incompatible with this system. This is also the reason behind settling for a capacity of 256MB, even though the SoC supports up to 512MB. No 512MB DDR2 component with a 16 bit device width was found available at the time of research. However, it was deemed that a capacity of 256MB should be enough based on other high performance OMAP-L138 systems, such as the HawkBoard and PD-Logic's OMAP-L138 SOM-M1, who are both designed with 128MB RAM. Furthermore, the system is intended to be a specific purpose system with a limited set of applications to handle. Support for extensive multitasking which requires a substantial amount of RAM is not a goal. This provides the confidence to say that 256MB of RAM for the system should be sufficient.

4.2 The Schematics

The Schematics of the DDR2 interface were created in OrCAD during the fall semester of 2011, and can be found on the CD.

4.3 The Layout

The DDR2 memory interface layout is one of the most important and crucial parts of the solution. It is also the most difficult part to implement. The challenges that had to be tackled were:

- **Simulation**
- **BGA breakout**
- **Signal timing constraints**
- **Signal integrity & Crosstalk**
- **Topology and component placement**

The signals in the OMAP-L138's DDR2 interface are divided into several groups, so called net classes. There are three clock net classes and four signal net classes.

CLOCK NET CLASS	Soc PIN NAMES
CK	DDR_CLKP / $\overline{\text{DDR_CLKN}}$
DQS0	DDR_DQS[0]
DQS1	DDR_DQS[1]

SIGNAL NET CLASS	ASSOCIATED CLOCK NET CLASS	Soc PIN NAMES
ADDR_CTRL	CK	DDR_BA[2:0], DDR_A[13:0], $\overline{\text{DDR_CS}}$, $\overline{\text{DDR_CAS}}$, $\overline{\text{DDR_RAS}}$, $\overline{\text{DDR_WE}}$, DDR_CKE
D0	DQS0	DDR_D[7:0], DDR_DQM0
D1	DQS1	DDR_D[15:8], DDR_DQM1
DQGATE	CK, DQS0, DQS1	DDR_DQGATE0, DDR_DQGATE1

Figure 4.2: DDR2 net class definitions, [T.I.L-138, 2009, p. 128]

Each net class has a set of restrictions and design rules that should be followed during the routing process in order to ensure proper functionality. The purpose of these design rules is to alleviate the need of simulating the design as well as to provide a safer and more reliable method to ensure a fault free DDR bus. Texas Instrument's published an application report that explains the background behind these rules, titled "Understanding TI's PCB Routing Rule-Based DDR Timing, SPRAAV0A [Shust and Cobb, 2008]". A lot of the information in the upcoming subsections is from that application report, in case the reader is interested in a reference.

Texas Instruments ensures that no simulation is needed if the rules are followed precisely, while at the same time specifying that the approach of just following these rules blindly does not enable a completely inexperienced person to design a high speed PCB [Shust and Cobb, 2008, p. 4]. This means the designer must understand the purpose behind each decision, even if that decision is based on a recommendation from the manufacturer.

4.3.1 Simulation

Classical high-speed design flow consists of a lot steps and is highly centered around simulations to ensure successful results. The main concern is to ensure that a high speed signal can travel from a component to another in a given design within a certain time frame, without being distorted or corrupted on the way, and without causing distortion to other signals as well. However, simulation has it's perils and is not a trivial task. Errors from inaccurate models and a variety of other sources are common, a substantial amount of resources are required, and the tools are expensive, rendering many smaller companies unable to perform simulations.

No simulations were conducted in this project, because not only is the process dependent on expensive tools, accurate models, and knowledge of how to configure the simulation environment, but extensive knowledge is also required to interpret the results from the simulations and apply them in the real world. In fact, it would've been impossible to design a DDR interface for this project without the design rules provided by TI to eliminate the need

of simulation. Based on this, it is recommended that future projects that intend to work on similar designs choose a microprocessor that specifies routing rules for it's DDR interface, as opposed to a microprocessor that leaves the task of finding the correct parameters up to the PCB designer.

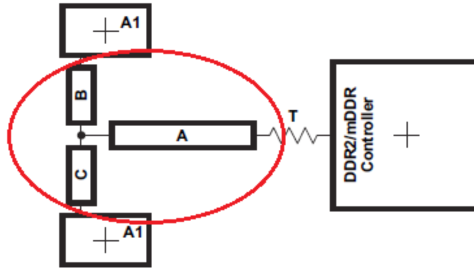
4.3.2 Signal timing constraints

Timing is a fundamental issue in high-speed design, especially in DDR interface design. The DDR interface is source synchronous; the clock for any data transfer is sourced from the device that is providing the data. A reason for why source synchronous clocking is useful is that it has been observed that circuits within a given semiconductor device experience the same process-voltage-temperature (PVT) variation[Wiki:Source-synchronous]. The result is that the data and the clock signals out of a semiconductor device experience the same propagation delay, as opposed to the scenario where a global clock generated by another device is utilized and the propagation delay is different, making it hard to ensure proper clocking of high-speed data bits. However, source synchronous clocking results in a horse race between the clock and each of the data bits due to signals being sourced from the same device. This imposes strict timing requirements on the design, which in practice translates to track-length requirements and design rules that each net class in the interface has to follow. In other words, in order to control the timing of when the signals arrive, one needs to control the track lengths and ensure that they're all matched throughout the interface.

Terminology commonly used in regards to this issue include terms such as "flight time delay" and "skew", referring to the delay of signals from one device to the another, and the skew, i.e the difference, between the flight time/track length of different signals. A problem that was encountered in the early stages of the project was interpreting the phrase "Skew Length Mismatch" found in the datasheet of the OMAP-L138.

DDR2/mDDR CK and ADDR_CTRL Routing

Figure 5-23 shows the topology of the routing for the CK and ADDR_CTRL net classes. The route is a balanced *T* as it is intended that the length of segments B and C be equal. In addition, the length of A should be maximized.



CK and ADDR_CTRL Routing and Topology

CK and ADDR_CTRL Routing Specification

NO.	PARAMETER	MIN	TYP	MAX	UNIT
1	Center to Center CK-CKN Spacing(1)			2w(2)	
2	CK A to B/A to C Skew Length Mismatch(3)			25	Mils
3	CK B to C Skew Length Mismatch			25	Mils

Figure 4.3: Skew Length Mismatch rules [T.I.L-138, 2009]

Having no prior experience with timing constraints, it was hard to interpret the phrase in the context it was presented in the datasheet, and no one at the NTNU Campus could provide an explanation for what the phrase was referring to in this exact case. The intuitive interpretation is that the phrase refers to the difference in track length between each of the segments, but it was unclear how to interpret the details around the "A to B/A to C" part. Furthermore, the CK net consists of two signals, so what was the maximum allowed skew between the two signals? After spending some time trying to figure this out independently, I finally decided to send a question to TI by posting on their e2e (Engineer to Engineer) forums. It turned out that the intuitive interpretation was correct, and that I could ignore everything regarding the balanced T in figure 4.3, because I only had one memory device, as opposed to the dual memory device configuration illustrated in the figure. Furthermore, I was told that the maximum allowed skew between the CK signals was 25 mils[1] as well.

Net Length Matching- OrCAD

As mentioned previously, controlling the timing practically translates to controlling the physical track length of each net: the longer the track, the more time it'll take for the signal to travel from one point to the other.

The following rules were specified in the datasheet for the different net classes.

[1] 1 mil = 0.001 inch(1/1000th of an inch). Common unit in PCB design.

PARAMETER	MIN	TYP	MAX	UNIT
ADDR_CTRL to CK Skew Length Mismatch			100	Mils
ADDR_CTRL to ADDR_CTRL Skew Length Mismatch			100	Mils
D to DQS Skew Length Mismatch (4)			100	Mils
D to D Skew Length Mismatch (4)			100	Mils
DQGATE Length F		CKB0B (1)		
DQGATE Skew (3)			100	Mils

(1) CKB0B1 is the sum of the length of the CK net plus the average length of the DQS0 and DQS1 nets.
(3) Skew from CKB0B1

Figure 4.4: Skew Length Mismatch, various net classes [T.I.L-138, 2009]

It was at this part of the project where using OrCAD became problematic. First of all there was no easy way to define net classes in the PCB Editor, only in OrCAD Capture (the schematics program), but since ITK department didn't have an Allegro license, this information could not be transfered from Capture to PCB Editor. Furthermore, the license owned by the ITK department didn't support the feature of an interactive length bar that shows the length of the track you're routing in real time. An etch length report like the one shown in figure 4.5 has to be generated each time you want to know the length of a given track, making it extremely difficult to tune track lengths.

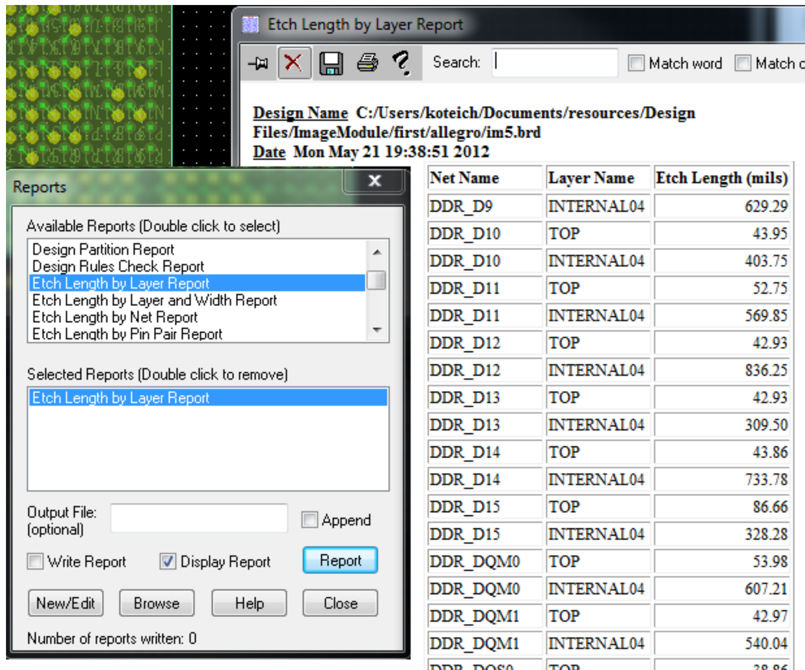


Figure 4.5: Etch length report, OrCAD PCB Editor

Having no choice at the time other than to continue using OrCAD, several attempts were made to manually equalize the track lengths. A common way to increase the length of a given track in order to match the length of another is to create a serpentine pattern as illustrated in figure 4.6.

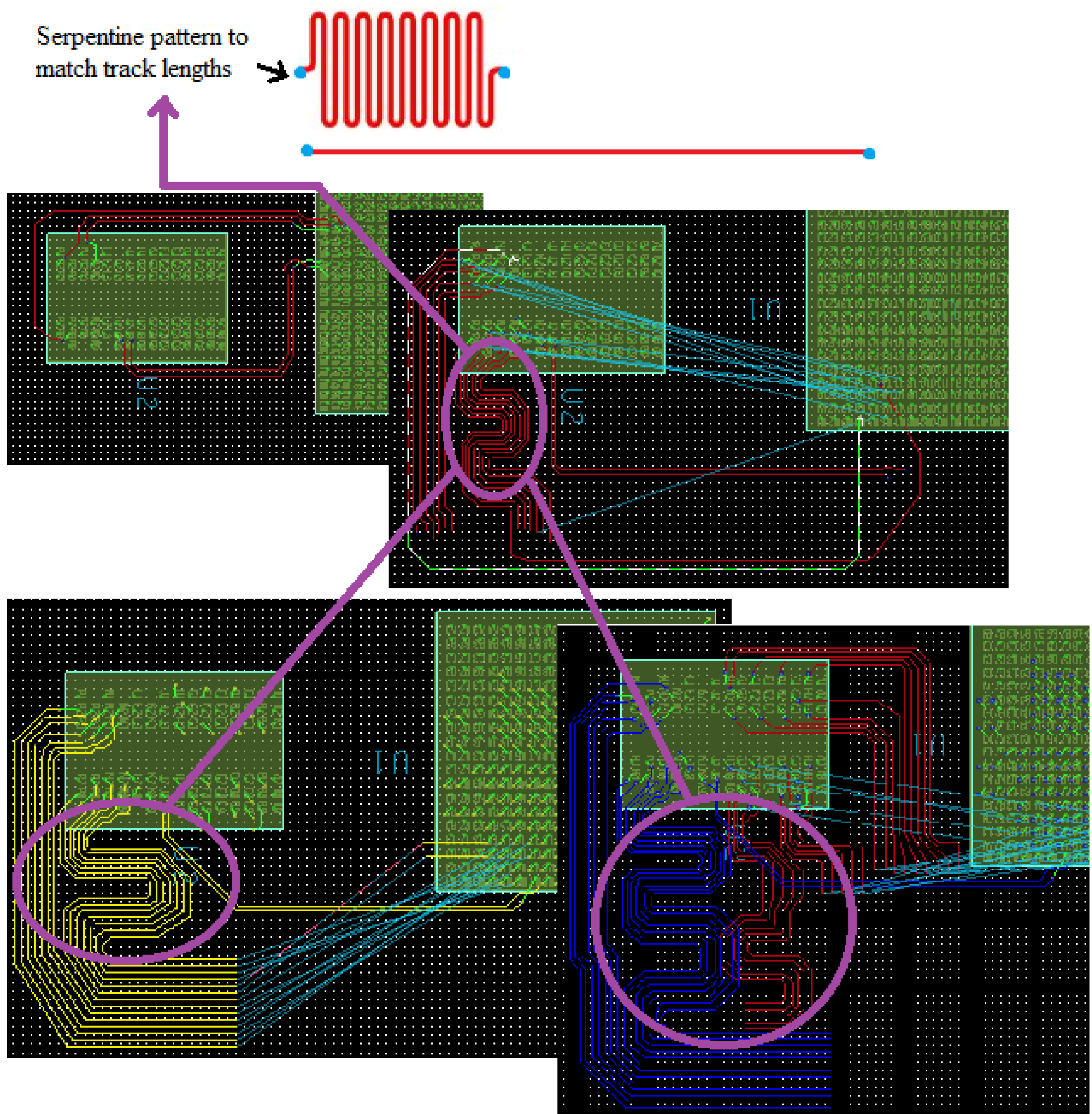


Figure 4.6: Attempts at track length matching, OrCAD PCB Editor.

This was an extremely tedious and time consuming task. After a lot of trial and error it

became obvious that it was nearly impossible to route the DDR interface in this manner within a feasible time frame. Unfortunately, a lot of valuable time was spent on trying to make things possible in OrCAD. However, this experience is part of fulfilling the goal of having an overview of our in-house capabilities. So even if it was a waste of time in terms of the application goals, it was still valuable experience for the department. It is now possible to confidently say that the OrCAD license owned by the department of engineering cybernetics is not suited for high-speed PCB design.

Net Length Matching- AD(Altium Designer) 10

Length matching in Altium Designer is done in a systematic and interactive way. The following steps need to be followed:

- **Create net classes:** This is the first step of the process, and even if the signals are only divided into seven net classes in the data sheet, it was necessary to group some of these classes together and create custom net classes in Altium in order to follow the rules easier. For example, since the maximum acceptable skew between the CK and the ADDR_CTRL classes is 100 mils, the same as the maximum skew between the signals within the ADDR_CTRL class, these two classes were merged together into a custom net class (while of course being aware of the 25 mil skew requirement between the differential pairs within the CK class. Signals can be a part of several net classes at once). The Skew in a net class is given by comparing the longest and the shortest net in the class.

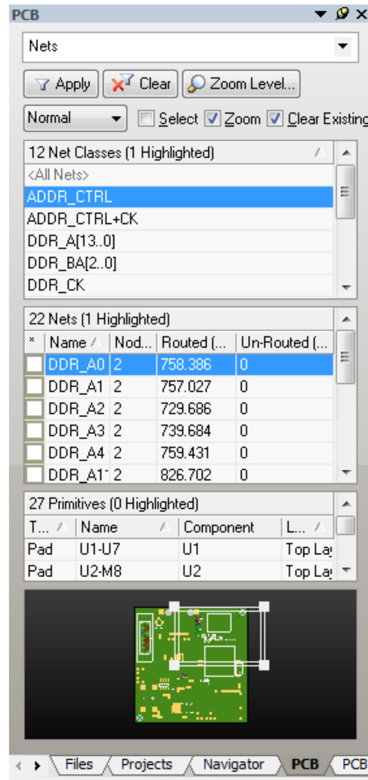


Figure 4.7: Net classes in the project. AD 10

- **Create design rules:** The next step is to use the "PCB rules and constraints editor" to create rules where the maximum acceptable skew between signals in a given net class is defined.

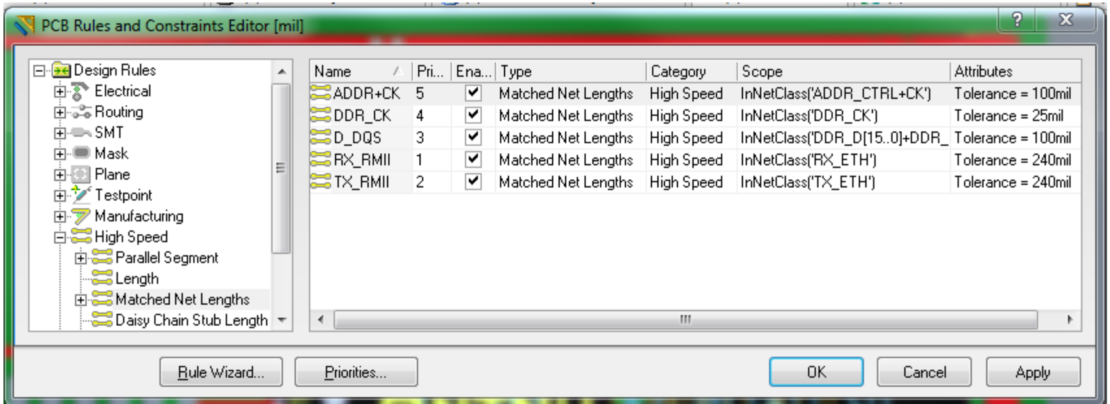


Figure 4.8: Design rules, matched net lengths. AD 10

- **Route the signals** without giving too much thought to length matching. But consideration should still be made in regards to spacing between signals in order to facilitate for the serpentine patterns that will be added at a later stage.

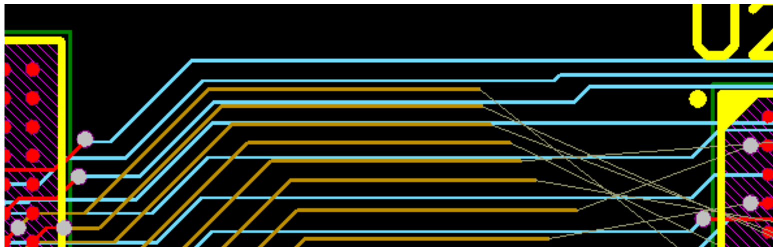


Figure 4.9: Route nets directly, different colors indicate signals on different layers. AD 10

- **Use the "Interactive Length Tuning" tool** on each signal to create serpentine patterns in order to match the longest track in the net class. The tool displays a bar that indicates how far a given net is from being within the tolerance specified by the rules earlier. Serpentine patterns are created as the mouse is dragged along the track until the bar indicates an acceptable level.

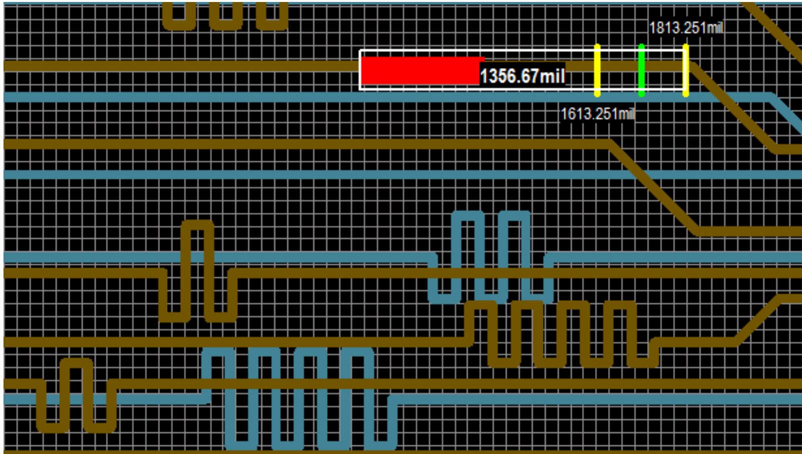


Figure 4.10: Interactive Length Tuning, the bar is red indicating that the particular track is too short and needs serpentine patterns. AD 10

- Modify the amplitude and the gap of the serpentine pattern to best fit the spacing needs of various places in the design.

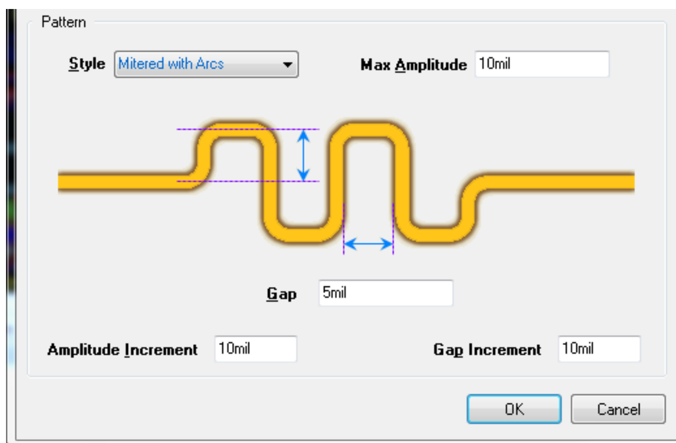


Figure 4.11: Pattern properties. AD 10

4.3.3 Signal integrity & Impedance Control

Some of the background theory concerning this topic is also discussed in the compendium I wrote last semester titled "Advanced Multilayer &High-Speed Printed Circuit Board Design", and can be found in appendix A.

Basically, to ensure the integrity of a signal one needs, among other things, to control the impedance of the track. Simply put, impedance is the "total resistance" of a circuit with resistive, capacitive, and inductive properties lumped together.[Robjohns, 2003]. TI recommends a single ended[2] impedance between 50 and 70 ohms for the electrical tracks in the DDR2 interface [T.I.L-138, 2009, p. 130]. The impedance of a given track is determined by several factors, such as the width of the track, the height above the reference layer (the dielectric thickness), and the dielectric material being used in the PCB. Setting an impedance level for a group of tracks is often as simple as letting the manufacturer know about the requirements in order to tune the thickness of the dielectric material to match the desired impedance level.

Impedance discontinuities

When a signal travels on a path that has several impedance levels, an undesired electrical phenomenon called reflection occurs(some of the signal energy gets reflected back to the source)[Chen, 2001].

Reflection causes problems such as overshoot and ringing(unwanted oscillation of the signal, due to forward echoes caused by reflected signals that get reflected an additional time on their way back to the source, striking the receiver at different intervals) at the the receiving end, rendering it unable to properly interpret data.

Termination

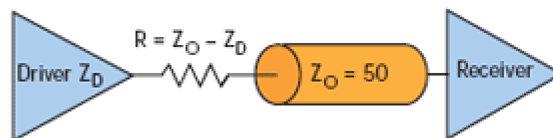


Figure 4.12: Series termination, www.electronicdesign.com

[2]Single ended impedance refers to using the circuit ground as a reference for the impedance value, as opposed to differential impedance, which specifies the relative impedance between two electrical tracks.

A common strategy employed to reduce signal reflections in DDR interfaces is to "terminate" the electrical track by placing a resistor in series with the track near the source (also called the driver). The value of the resistor added with the output impedance of the driver should match the value of the track's impedance, this is referred to as impedance matching. The signal will travel over the resistor, causing the voltage to drop, and upon reaching the receiver, a reflection will occur due to the impedance mismatch between the input pin and the transmission line, boosting the voltage up to it's original value (the boost is equal to the original drop), enabling the receiver to interpet the full signal. This reflection will also propagate back to the series resistor, but the voltage will drop to zero over the resistor(since the voltage level of the reflection was equal to the drop initially caused by the resistor), preventing anymore current to enter the line, thus eliminating forward echos.[3]

However, no termination techniques were utilized in the design. This is because TI specifies that no termination is required since the drive strength of the OMAP-L138's DDR interface can be configured, and recommends configuring the drive strength to operate at 60%. Impedance matching is carried out by configuring the drive strength directly instead of using a resistor for this purpose, freeing up board-space and decreasing the overal component count of the system.

4.3.4 Impedance discontinuities, difference between theory and practice

Some confusion aroused while researching the topic of impedance discontinuities and trying to implement the recommendations found in the literature. Various sources[4] recommend avoiding sharp 90 degree turns in the routing, because the track will become a little thicker in the turn, causing an impedance mismatch at that exact spot. The impedance mismatch results in unwanted reflection and impacts signal integrity. It is important to note that the recommendation of avoiding sharp turns is never made for DDR signals, only for USB2.0 and Ethernet signals. This was confusing because they're all high speed signals where signal integrity and controlled impedance are important. I also noticed that the serpentine patterns created by Altium more or less consist of sharp turns.

[3]references: [Johnson and Graham, 1993], [ARM, 2010], [Klos, 2004], [Altera.AN75, 2001], [Wiki:Signal_Reflection], [Wiki:Characteristic_impedance]

[4][TI.SPRAAR7, 2007], [SMSC.AN186, 2008], [Panasonic, 2011b]

Figure 4.13: 90 degree turns in serpentine pattern

It was important to get to the bottom of this to ensure that things were being done properly in the design. After some research, I posted a question on TI's e2e forums and asked them why they recommend avoiding 90 degree turns for USB2.0 [TI.SPRAAR7, 2007] but mention nothing about avoiding sharp turns for the DDR2 interface. A TI employee answered with the following:

> "*Even at USB2 speeds this is not really necessary. That recommendation will not really have an effect on signal integrity. See this article for some analysis:*
>
> *http://www.sigcon.com/Pubs/edn/bigbadbend.htm*
>
> *Jeff*"[J.C-TI, 2012]

The article I was referred to, titled "Who's afraid of the big bad bend?"[Johnson, 2000], is written by Dr. Howard Johnson, one of the leading experts within the field of high-speed design, co-author of the well renounced book "High Speed Digital Design, a handbook of black magic, 1993", inventor of the PhoneMail (first voicemail system), and chief technical editor of Ethernet standards.[5]

Dr. Johnson concludes, based on experimental analysis, that the reflection from a right-angle bend is too small to worry about in most digital high speed design cases. Furthermore, the notion that electrons are moving too fast to be able to turn properly is rejected, and radiation issues in a corner is proven to be a misconception. However, Dr.Johnson mentions that microwave designers should worry about right-angle bends, since microwave designs can be 1000 times more sensitive to this issue than digital designs. He also states that right-angle bends might become a problem in the future, since 10Gbps serial connections are just beginning to show signs of being affected, and that the accumulation of a lot of corners through serpentine routing might cause a little extra delay, which is acceptable since that is the purpose of the serpentine patterns in this design.

[5] http://www.sigcon.com/hj.htm

4.3.5 Crosstalk

Crosstalk is an electrical phenomenon where an undesired effect on a signal is caused by a signal on another circuit or track. This can happen when the spacing between the electrical tracks is too small and when there is no high-quality signal return path. The signal return path can either go through the ground or the power plane, and it is important that signals don't cross over discontinuities such as splits and separated islands in these planes. Discontinuities in the return path will cause the return current to flow in larger loops, which increases radiation from the board as well as increases the crosstalk between adjacent traces [Ott, 2000]. Due to this important requirement, no splits were created in the power and ground planes surrounding the DDR interface in the KybMo design.

As for the spacing between the tracks, TI recommends the following spacing parameters for the different classes:

PARAMETER	MIN	TYP	MAX
Center to center CK to other DDR2/mDDR trace spacing[1]	4w[2]		
Center to center ADDR_CTRL to other DDR2/mDDR trace spacing[1]	4w[2]		
Center to center ADDR_CTRL to other ADDR_CTRL trace spacing[1]	3w [2]		
Center to center DQS to other DDR2/mDDR trace spacing[1]	4w[2]		
Center to center D to other DDR2/mDDR trace spacing[1]	4w[2]		
Center to Center D to other D trace spacing[1]	3w[2]		
Center to center DQGATE to any other trace spacing	4w[2]		

(1) Center to center spacing is allowed to fall to minimum (w) for up to 500 mils of routed length to accommodate BGA escape and routing congestion.
(2) w = PCB trace width

Figure 4.14: Recommended spacing parameters, [T.I.L-138, 2009]

These spacing recommendations were followed in order to avoid crosstalk. Some foresight was required while spreading out the tracks, to accommodate for the amplitude of the serpentine patterns that were gonna be added afterwards. The minimum (w) referred to in note 1 in figure 4.15 is 4 mils. It was necessary to reduce the spacing distance to this minimum under the BGA packages due to the high density of the pins. The signals were then spread out according to the recommendations after clearing the dense area.

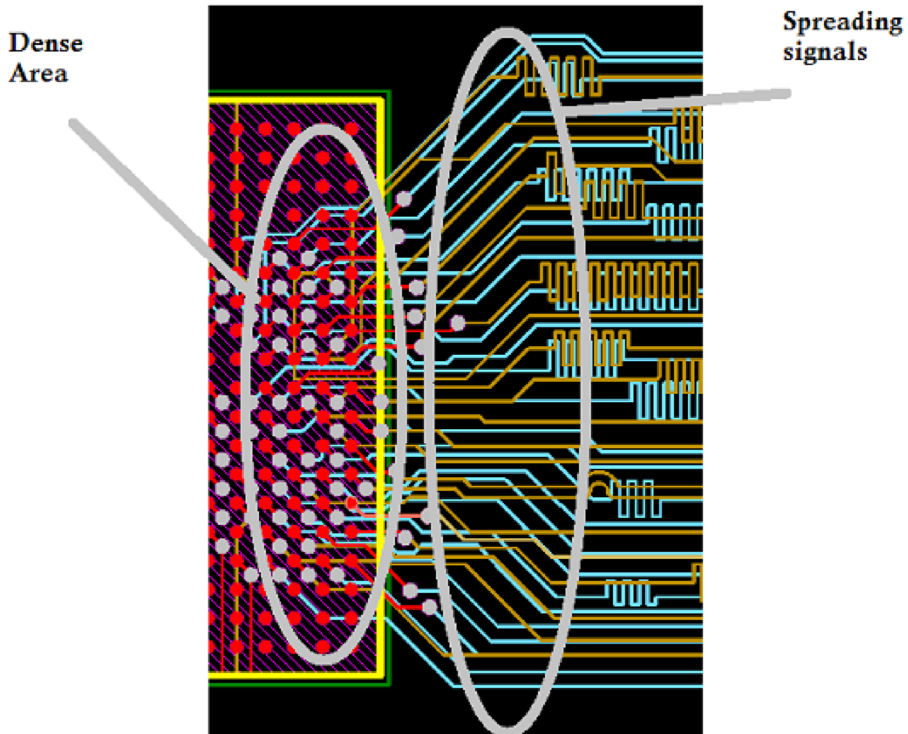
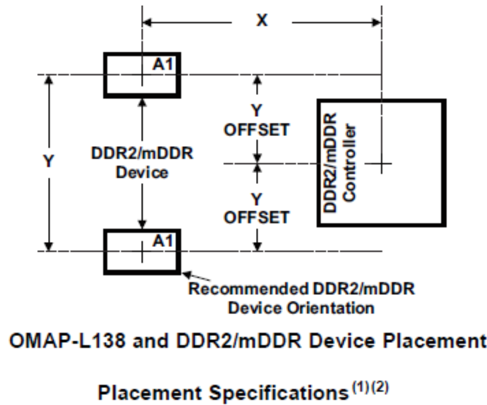


Figure 4.15: Signal spacing, AD10, KybMo

4.3.6 Placement

The placement of the memory chip relative to the SoC was practically one of the first things that had to be done after creating the footprint. This seemed like a trivial task to begin with, but proved to be a challenging one. TI provides the following recommendations for component placement in the OMAP-L138 datasheet:



NO.	PARAMETER	MIN	MAX	UNIT
1	X		1750	Mils
2	Y		1280	Mils
3	Y Offset		(3)650	Mils
4	Clearance from non-DDR2/mDDR signal to DDR2/mDDR Keepout Region (4)	4		w (5)

(1) See drawing for dimension definitions.
(2) Measurements from center of device to center of DDR2/mDDR device.
(3) For single memory systems it is recommended that Y Offset be as small as possible.
(4) Non-DDR2/mDDR signals allowed within DDR2/mDDR keepout region provided they are separated from DDR2/mDDR routing layers by a ground plane.
(5) w = PCB trace width as defined in

Figure 4.16: Placement recommendations, [T.I.L-138, 2009]

As we see in figure 4.16, it seems to be suggested that the memory device should lie with the long side on the horizontal axis. Interpreting it in such a way, and without giving too much thought about other options in terms of orientation, I spent a lot of time trying to route the design in a successfull way but to no avail. The main problem was that too many tracks crossed each other's paths, making it hard to route the signals to their destination without using an excessive amount of vias[6], and without bringing the signals out of the inner layers which are designed to protect them from EMI.

[6]Vias cause undesired signal reflections due to the impedance mismatch between the track and the via, see section 4.3.4

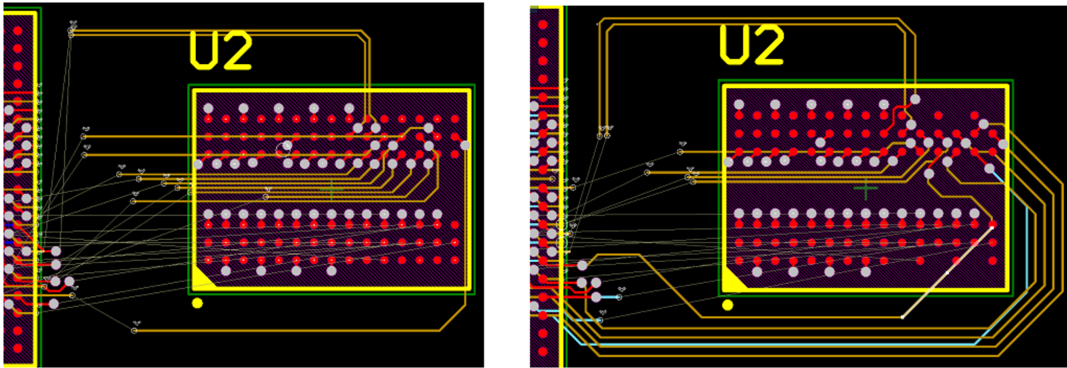


Figure 4.17: Long side of chip on horizontal axis. Several attempts were made at optimally routing the design

Another problematic issue was the fact that this orientation caused the difference between the shortest and the longest track to become large. Even though it is possible to match the net lengths at a later stage, it is desirable that all the tracks be as short as practically possible, to put less strain on the timing requirements by accommodating for lesser flight time for the signals. This orientation imposed restrictions on how short the longest track could be.

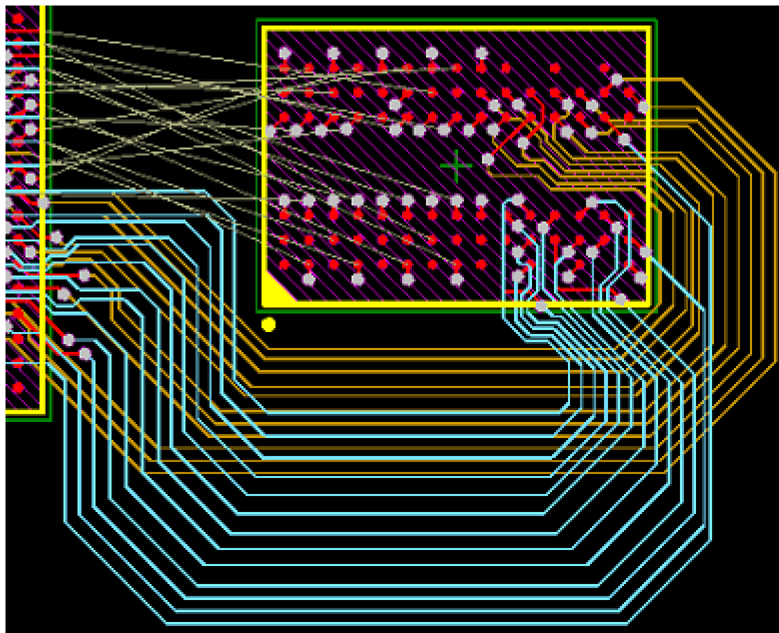


Figure 4.18: Long side of chip on horizontal axis. ADDR_CTRL class routed. Longest track: 1891 mils

In figure 4.18 we see that more space is required to be able to bring out the remaining signals, and even if the space was available, the criss-crossing of the remaining signals would've been extremely difficult to circumvent.

I soon realized that this was a hopeless endeavor and that I must be missing something. It should be noted that a reference design, called Hawkboard, was used throughout the project for practical tips and solutions on how to do things. The problem is that the Hawkboard, a general purpose platform based on the OMAP-L138 SoC, was taken out of production due several issues, amongst them a malfunction in the DDR2 memory interface. The Hawkboard's memory chip was also aligned with the longside on the horizontal axis. TI's official explenation for the malfunction is that the designers did not follow TI's routing recommendations. Thus, replicating the routing architecture used in the Hawkboard design was not wise.

While trying to figure out a smart way to route the signals from the memory chip to the SoC, I noticed something important in another TI certified general purpose board, the BeagleBone, which I was in posession of. The memory chip was aligned with it's long side on the vertical axes!

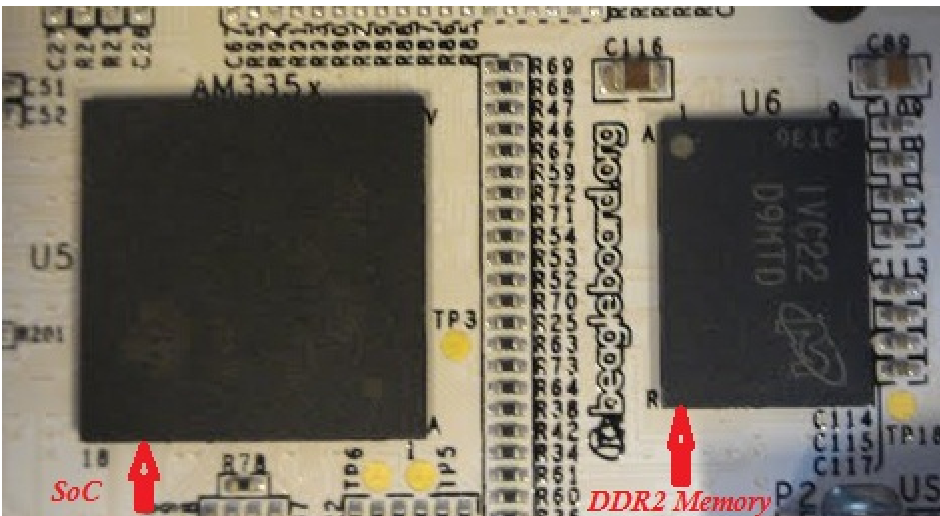


Figure 4.19: The Beaglebone, SoC and Memory chips.

I realized that placing the memory chip in that orientation would alleviate a lot of problems. However, since the Beaglebone uses a different TI SoC, the AM3359 MPU, I wasn't sure if using the same orientation for the memory chip would cause problems for the KybMo. Studying the AM3359 datasheet[T.I, 2012], the same exact drawing shown in figure 4.16 was found, confirming that there is no requirement for horizontal orientation of the chip, and that the drawing in figure 4.16, when referring to recommended orientation, is only

concerned with showing where the A1 pin should be and how far away the memory device is allowed to be located from the processor. This was not an easy interpretation to make without prior experience with such designs.

And sure enough, using a vertical orientation, the track length, routing space, and criss-crossing issues were easier to deal with. The longest track length was now mainly dictated by the distance between the SoC and the memory chip, not by the orientation.

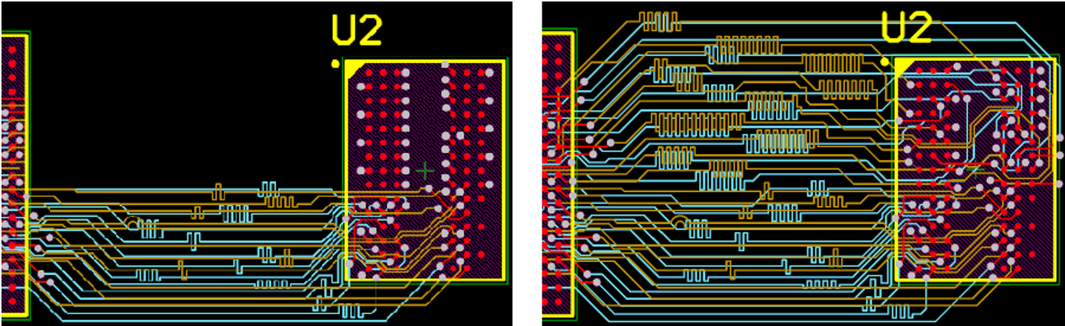


Figure 4.20: Vertical orientation.

Even with the center to center spacing(1212 mil) between the SoC and the memory chip in figure 4.20 being greater than the center to center spacing(845 mil) in figure 4.18, the same track that was 1891 mils long in the ADDR_CTRL net class got reduced to 1203 mils by using the vertical orientation.

But the story doesn't end there! The 1212 mil center to center spacing was initially chosen to ensure that there would be enough space for the serpentine patterns. Too small of a space could force the serpentine patterns on adjacent lines to be stacked directly on top of eachother, forcing the signals to be spread out over a wider area in order to comply with the spacing rules between the tracks discussed in section 4.3.5. However, as new components got added to the PCB, it became clear that the DDR interface was occupying too much valuable PCB space.

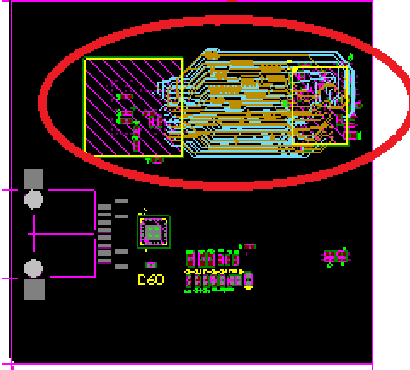


Figure 4.21: Entire PCB Area

This could become a problem as more components get added to the design. A lot of components were still missing from the design at this stage, so it was important to make sure that nothing occupied more space than what was absolutely necessary.

A new attempt was made at routing the interface, with the intention of decreasing the center to center spacing between the SoC and the memory chip as much as possible, and a reduction of 415 mils was achieved. This reduced the longest net length from 1744 mils to 999 mils.

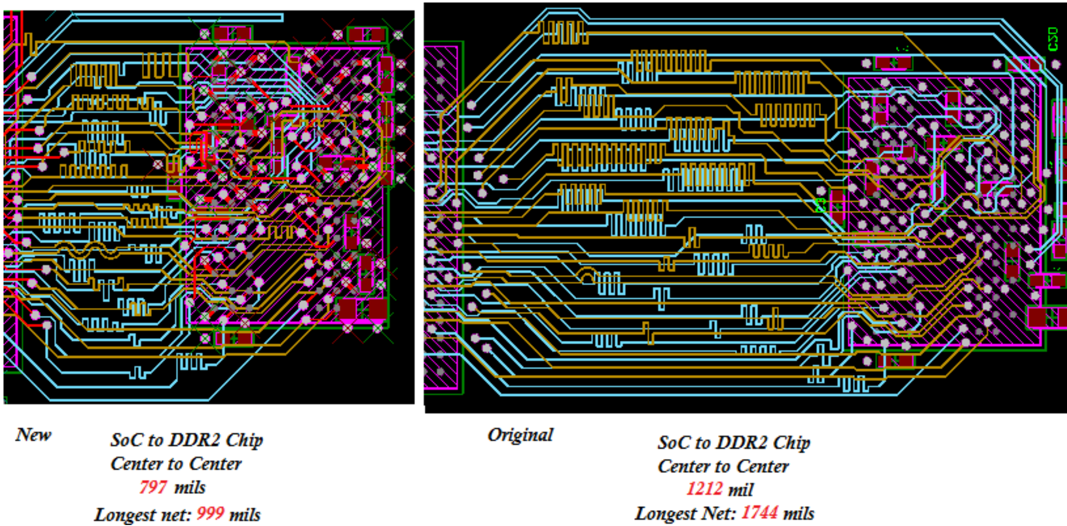


Figure 4.22: Center to center spacing was decreased with 415 mils in the final attempt.

4.3.7 BGA breakout

"BGA breakout" refers to the process of routing signals out from the BGA balls to the border of the component. The reason this is a challenging task is due to the nature of the BGA package: Balls(sometimes referred to as "pins") are arranged in a filled quadratic shape, as opposed to arrangements found in other packages, where the pins are aligned on the same horizontal or vertical line, ensuring that all signals have a clear escape route out to the PCB environment.

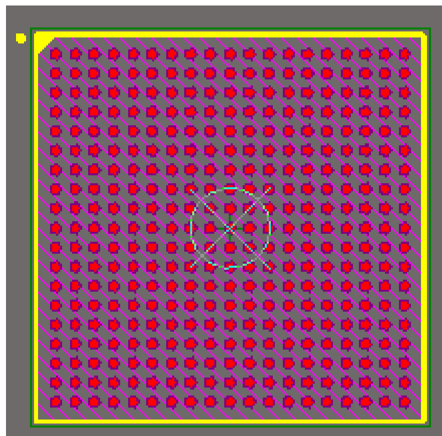


Figure 4.23: OMAP-L138, Ball Grid Array configuration

It is not possible to route the signals out from the component by only utilizing one layer. In order to route out the inner rows, several layers are required.

Both the memory chip and the OMAP-L138 come in BGA packages. The Memory chip has 84 balls while the OMAP-L138 has 361 balls. While BGA devices with less than 500 balls are considered to be "low-pin count"[Pfeil, 2007], and relatively easy to breakout compared to FPGAs with more than 1700 balls, the story is different when it comes to the DDR2 interface in this project.

First of all, instead of one BGA device, there are two, and both need to be connected together. Secondly, The tracks need to use as few vias as possible to circumvent eachother due to signal integrity purposes. Thirdly, in order to keep the cost of the board at a minimum, only through-hole vias are used to take a signal from one layer to another. A Through hole via occupies space on all of the layers since it's basically a hole that pierces from one side of the board to the other, thus the name "through-hole". This means that signals on all layers need to move around these vias, and can't be routed across them. Finally, most of the high-speed signals in the design are restricted to the two inner layers of the board, except for the stretch where they have to be connected to their respective BGA balls on the top-layer.

This produces a situation which is hard or even impossible for auto routers to solve(the

problem is akin to an NP-hard or NP-complete problem, and similar BGA breakout problems have been described as such [7]. The Altium auto router[8] was not able to route the design due to the complexity, so it had to be done manually by hand.

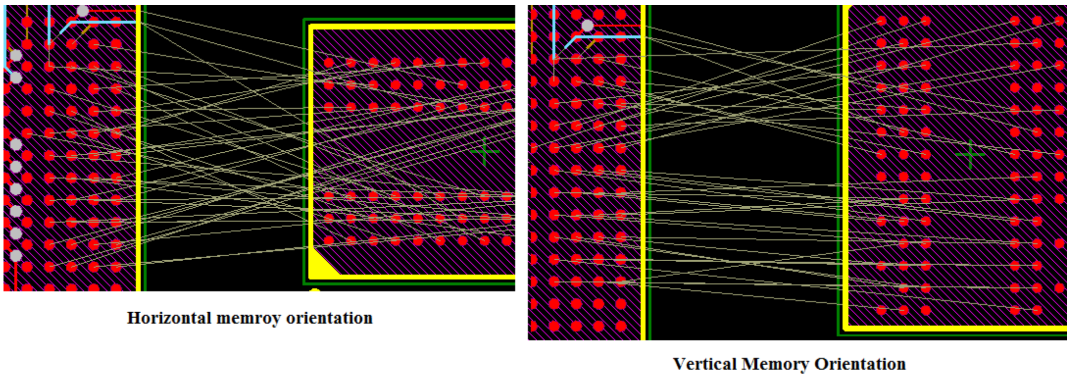


Figure 4.24: "Rats nest" lines indicating pin to pin connections

Figure 4.24 shows how important proper orientation is for the complexity of the routing. The process that was devised for breaking out the BGAs here can be divided into three main steps:

- Break out the densest and least flexible device first, in this case the OMAP-L138. Decide which layer individual tracks should be on to accommodate the breakout.
- Break out the memory chip in accordance to the net topology dictated by the previous step. Circumvent crossings in order to accommodate for a direct connection between tracks on both breakout sides.
- Modify the layer or placement of individual breakout tracks from the first step if it is found that it is not possible to circumvent a certain crossing during the second step.

[7] [Herhberger and Suril, 2006], [Yu et al., 1996]

[8] In fact, no auto-routing was used in the design for any of the peripherals. This was to ensure complete control of the entire process.

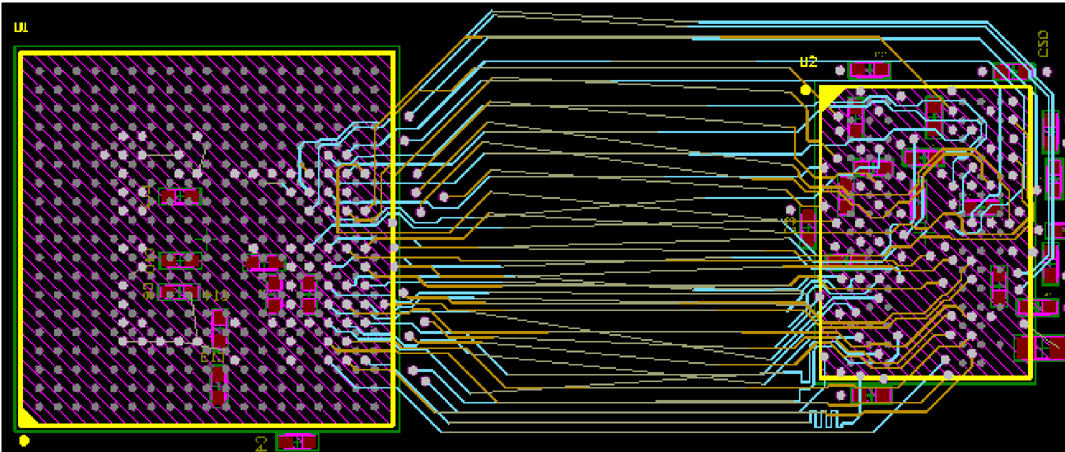


Figure 4.25: Complete BGA breakout of both devices for the DDR2 interface.

4.3.8 Layer Stack-Up

Layer stack-up refers to the layer configuration of the PCB, i.e the number of layers, and the order they're stacked in. Conventional student and hobby projects usually stick to designs with two-layer boards (top and bottom). Two-layer designs can be manufactured at the NTNU campus, eliminating dependency on an external manufacturer, and enabling easier rework and correction of mistakes. However, a two-layer board for this project was not an option. The DDR2 memory interface plays an important role in defining the layer structure of the PCB.

There are several reasons for why this design needs to have more than two layers:

- **Breakout for BGA packages:** The dense configuration of the balls require more than two layers in order to be routed successfully out of the devices.
- **Requirements for controlled impedance:** Some traces in the DDR-interface are required to have a controlled impedance of a certain value. In order to control the impedance, dedicated ground layers adjacent to the signal layers are required.
- **Frequency of high-speed signals and EMI considerations:** High frequency signals should be shielded from the outside world between solid planes, otherwise EMC problems can occur if the PCB is in an unshielded enclosure[Ott, 2000].
- **Board dimension requirements:** More layers enable smaller board dimensions.

TI recommends a minimum of six layers for this design:

Device Minimum PCB Stack Up

LAYER	TYPE	DESCRIPTION
1	Signal	Top Routing Mostly Horizontal
2	Plane	Ground
3	Plane	Power
4	Signal	Internal Routing
5	Plane	Ground
6	Signal	Bottom Routing Mostly Vertical

Figure 4.26: 6 layer minimum stack up recommendation, [T.I.L-138, 2009]

Important considerations

An important factor is the configuration of the reference planes, i.e the power and ground layers. They're called "planes" because they're layers that consist purely of one or more large copper areas, as opposed to point to point tracks. Power and ground planes can be used to shield emissions from high frequency signals. They also decrease power and ground impedance, compared to point to point power and ground rails, which in turn results in less power and ground noise [Ott, 2000].

TI's recommended minimum stack-up was tried to begin with, but was left behind for an 8 layer configuration instead. The main problem with the minimum recommendation is that it only offered one shielded layer (layer 4 in figure 4.26). Routing the entire DDR2 interface on one layer was too challenging. This caused the ammount of layers to be increased from 6 to 8 (PCB layers are commonly increased in increments of two. An odd layer count has a higher fabrication cost).

However, the price of increasing the layer count from 6 to 8 was not negligible. There was a difference of 19 USD per card at the same quantity (5 pieces), and a difference of 140 USD in the tooling cost. It should be noted that these exact numbers were not available from the start. Due to a lot of other price decisive fabrication variables still being unknown at the time, it was not easy to calculate an exact price estimation, but a price difference was of course anticipated.

Furthermore, using a 6 layer design and sticking to only one shielded layer for the DDR interface would mean that more PCB-space would've been needed, increasing the board dimensions. More vias would have to be utilized in the BGA breakout, because instead of being able to bring the signal directly out to the shielded layer, one would need to bring the signal from the top layer, where the component pad is, to the bottom layer to circumvent crossings, and then back up to the shielded layer. The same process would have to be carried out at both sides of the interface, drastically increasing potential signal reflec-

tion and impedance mismatch problems. This would possibly require a more experienced high-speed designer in order to ensure proper functionality. Routing the interface would've been even more time consuming, and designing the power distribution would've been a more challenging task due to one less power layer being available compared to the 8 layer configuration. Everything comes at a price, and a difference of 240 USD (approximately 1350 NOK) would ensure a safer, smaller, and less time consuming design. It is worth mentioning that the Hawkboard used a 6 layer configuration, but in order to circumvent the challenge of only one shielded layer, the designers removed a ground layer from the design and replaced it with a signal layer. That would've been a solution for this project if the Hawkboard didn't end up having problems with it's DDR interface, thus, repeating their actions was not wise. The only OMAP-L138 known to be functional, and available reference design at the time of research, Logic PD's SOM-M1, used 10 layers for the entire design, granted that more of the OMAP-L138's functionality is utilized too, but the DDR interface utilized two inner layers [LogicPD, 2010].

Layer Configuration

Upon deciding on using 8 layers, a configuration for the layer sequencing had to be found, sine TI only specified a configuration for 6 layer boards.

[Ott, 2000] describes several stack-up configurations for 4, 6, 8 and 10 layer boards. The following configuration was described as a basic stack-up for eight-layer boards with excellent EMC performance[Ott, 2000, p. 8], and was chosen for this project:

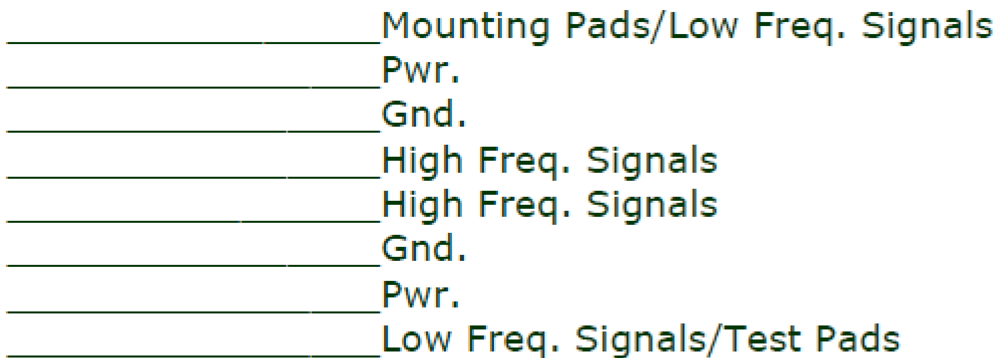


Figure 4.27: 8 layer stack-up, [Ott, 2000]

4.3.9 Bypass and decoupling capacitors

> *"Precise power supply bypassing is essential for a properly functioning high-speed PCB. It is fundamental to control the power supply high-frequency impedance parameter which means controlling power supply inductance. Power supply high-frequency impedance is beaten down by utilizing many physically small capacitors connected between the power and ground planes. Using many capacitors, rather than one large one, results in their parasitic inductances being placed in parallel, thereby, reduced." [Shust and Cobb, 2008]*

The OMAP-L138 datasheet specifies the bypass capacitor recommendations:

High-Speed Bypass Capacitors

NO.	PARAMETER	MIN	MAX	UNIT
1	HS Bypass Capacitor Package Size [1]		0402	10 Mils
2	Distance from HS bypass capacitor to device being bypassed		250	Mils
3	Number of connection vias for each HS bypass capacitor	2 [2]		Vias
4	Trace length from bypass capacitor contact to connection via	1	30	Mils
5	Number of connection vias for each DDR2/mDDR device power or ground balls	1		Vias
6	Trace length from DDR2/mDDR device power ball to connection via		35	Mils
7	DDR_DVDD18 Supply HS Bypass Capacitor Count [3]	10		Devices
8	DDR_DVDD18 Supply HS Bypass Capacitor Total Capacitance	0.6		µF
9	DDR#1 HS Bypass Capacitor Count [3]	8		Devices
10	DDR#1 HS Bypass Capacitor Total Capacitance	0.4		µF

(1) LxW, 10 mil units, i.e., a 0402 is a 40x20 mil surface mount capacitor
(2) An additional HS bypass capacitor can share the connection vias only if it is mounted on the opposite side of the board.
(3) These devices should be placed as close as possible to the device being bypassed.
(4) Only used on dual-memory systems.

Figure 4.28: HS bypass capacitor recommendations, [T.I.L-138, 2009]

These recommendations were followed and the result was 16 highspeed 0.1uF capacitors in prallel, and one bulk bypass 22uF capacitor. The convention is, as far as practically possible, to have one capacitor pr power ball on the device. Because the memory device had 16 powerballs, and enough space under it, it was possible to use 16 bypass capacitors. The capacitors were placed directly under the device on the bottom side of the PCB, to stay as close as possible to the power balls, which is an important aspect of high-speed bypassing. If the capacitors are too far away they will not ensure proper bypassing[Shust and Cobb, 2008].

Chapter 5

Power &Reset

5.1 Power supply & distribution

The power supply is the umbilical cord of the system and several reference designs were studied in order to choose the best solution for the KybMo.

5.1.1 Required Voltage Levels

First of all, it was important to get an overview of how many different voltage levels the system required.

The OMAP-L138 SoC alone requires 3.3V, 1.8V, 1.3V, and 1.2V. Most of the remaining ICs in the system run on 3.3V, and the USB interface requires 5V. This means a total of 5 different voltage levels.

One reference design, the evaluation kit from Logic-PD, utilized several low drop out regulators to generate all the desired voltage levels from a 5V input. While the LCDK and the Hawkboard used a dedicated power supply IC that converts a 5V input to several voltage level outputs. This is also done by utilizing LDO regulators, but having this integrated into one 5x5 mm package saves more space on the board. Thus, the LCDK/Hawkboard power supply solution was chosen for the KybMo.

Some modifications had to be made, such as utilizing 1.3V for the processor core instead of 1.2V, in order for the CPU and DSP clocks to be able to reach 456 MHz.

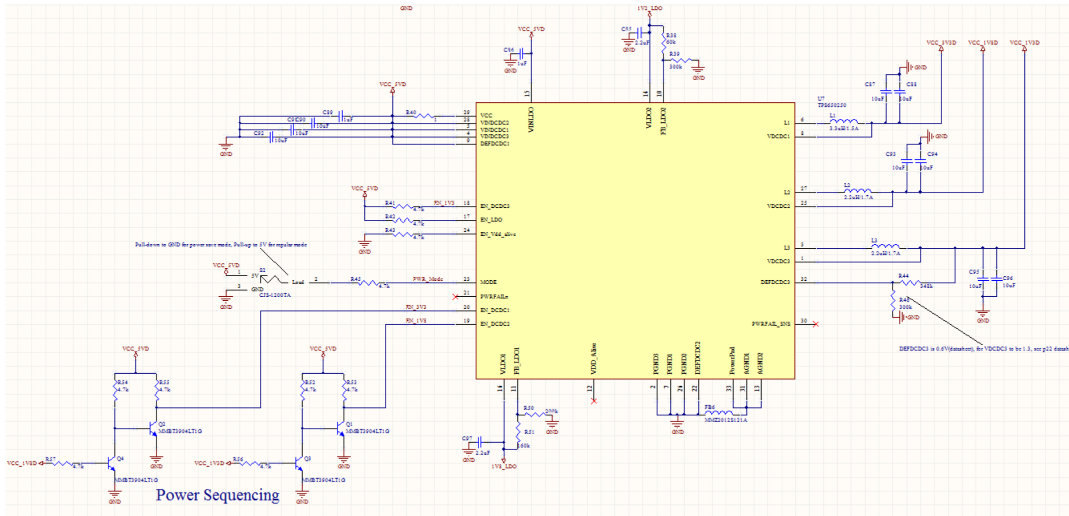


Figure 5.1: KybMo, TPS650250 Powersupply schematics

5.1.2 Architecture

The power supply system consist of the following ICs:

- **TI TPS650250:** Power supply IC, converts 5V input to desired voltage levels and out puts these on separate power rails.
- **TI TPS3705:** Monitors power levels and generates a reset signal out to the circuitry which is held until all levels are stable.
- **TI TPS2068 & TPS2087:** USB and Power-Jack power switching and current protection.

In addition, there's a 5.1V Zener diode at the 5V Jack output which is intended to protect the circuit from voltages higher than 5V. if a voltage higher than 5V is passed, the zener diode will open and regulate the voltage level to stay at 5V.

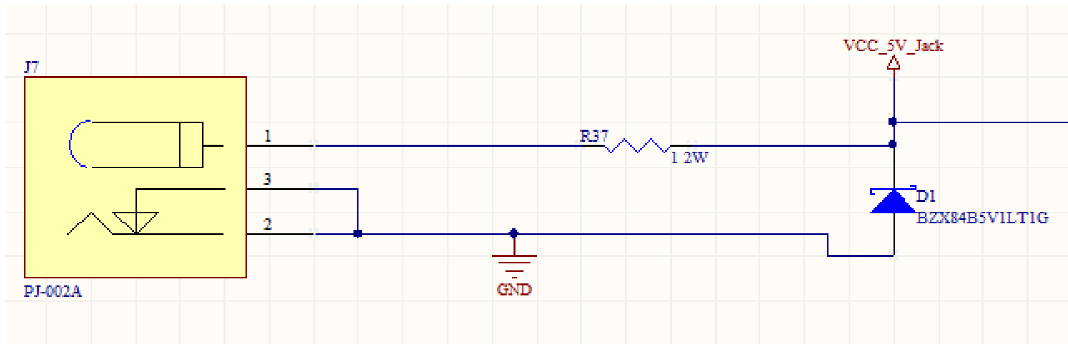


Figure 5.2: Zener Diode for over-voltage protection

5.1.3 Power & Ground planes

There are two power planes and 2 ground planes on the KybMo. Even though there are only 5 required voltage levels, two important peripheral modules in the OMAP-L138 (DDR2 and the USB2.0) need 1.8V, so 1.8V was sourced on two separate rails in order prevent over-load on one rail. Furthermore, an analog 3.3V island had to be made for the Ethernet PHY.

This means that the 2 power planes had to be partitioned to accommodate seven power rails(8 including the unregulated 5V signal). Partitioning the planes properly, while making sure not to create splits under any high speed nets was a challenging task, and it was soon discovered that more of the remaining layers had to be utilized. This included a small part of the Ground-2 layer, and parts of the highspeed-1 and highspeed-2 layers

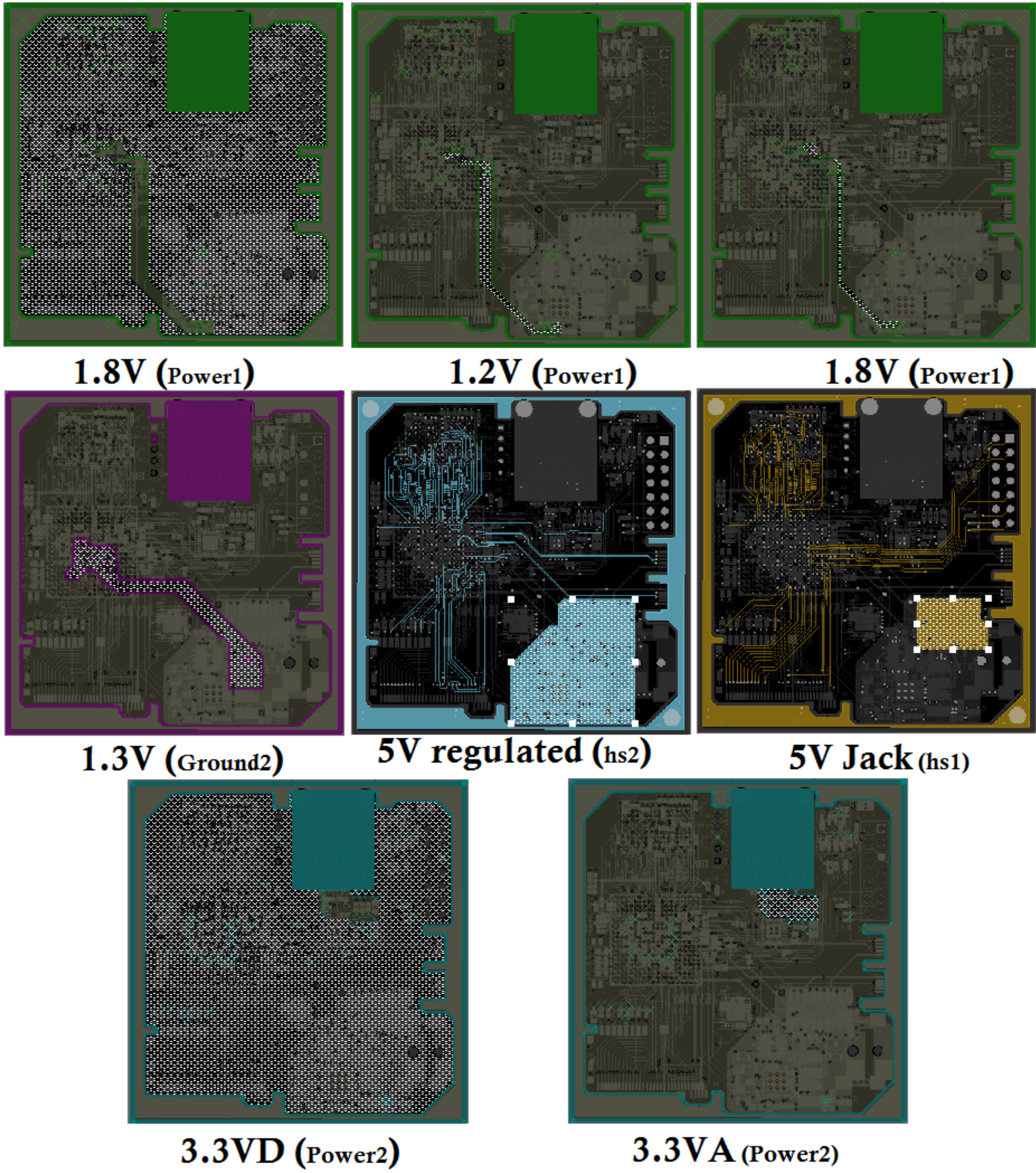


Figure 5.3: KybMo, plane and layer partitioning

5.2 Reset

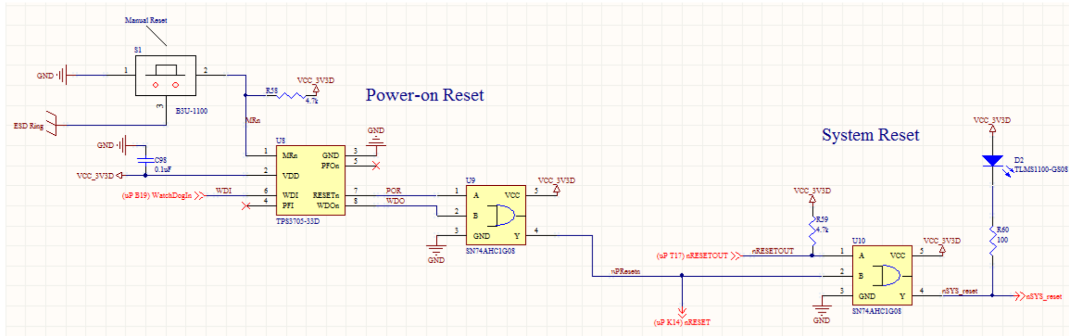


Figure 5.4: KybMo, reset generation schematics

A careless mistake was made in the design process which was of some importance. The watchdog functionality of the TPS3705 reset generator was utilized and connected to system reset. The watchdog timer is set to 1.6 in the TPS3705, and if no transition is registered on the WDI pin within that time-frame, a reset signal is generated. The aspect that wasn't taken into consideration during the design phase is that the system needs more than 1.6 seconds to wake up from reset and service the watch-dog before it triggers again. This functionality will therefore keep the entire system in indefinite reset. A hardware patch was needed to bypass this issue. The patch is described in subsection 9.13.1, and can be seen on Figure 9.22.

This was a careless and unnecessary step, as it was later discovered that the OMAP-L138 has an internal watchdog timer that can be configured from software.

Chapter 6

Peripheral Interfaces

This chapter discusses the peripheral interfaces of the KybMo and gives an overview of the design guidelines that had to be followed for the high speed interfaces.

6.1 USB

The guidelines that were followed on the KybMo for the USB2.0 interface are specified in TI's application note [TI.SPRAAR7, 2007], titled *"USB 2.0 Board Design and Layout Guidelines"*

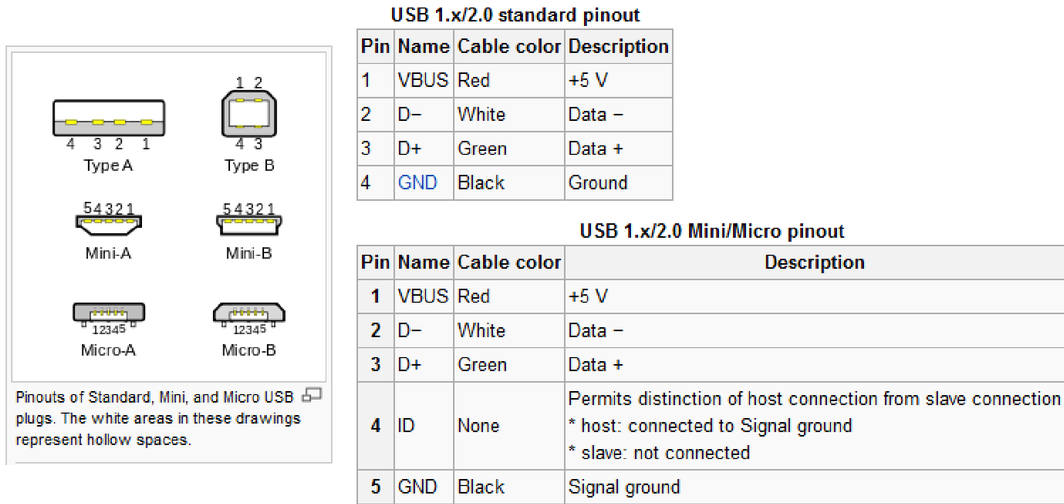
The USB 2.0 PHY[1] is integrated into the OMAP-L138 SoC, and is not a seperate device. Therefore, some of the sections regarding PHY placement and routing were not applicable in the KybMo design. The relevant areas for this design were:

- USB data signal routing.
- Connector shielding & electro static discharge (ESD)

Connector shielding and ESD is discussed in section 6.7.

[1]PHY is an abbreviation for the physical layer of the Open Systems Interconnection(OSI) model(wikipedia.org/PHY_(chip)). In practice this referrs to the device/chip that is responsible for providing the interface between the processor and the physical cable being used.

6.1.1 USB Interface



USB 1.x/2.0 standard pinout

Pin	Name	Cable color	Description
1	VBUS	Red	+5 V
2	D−	White	Data −
3	D+	Green	Data +
4	GND	Black	Ground

USB 1.x/2.0 Mini/Micro pinout

Pin	Name	Cable color	Description
1	VBUS	Red	+5 V
2	D−	White	Data −
3	D+	Green	Data +
4	ID	None	Permits distinction of host connection from slave connection * host: connected to Signal ground * slave: not connected
5	GND	Black	Signal ground

Figure 6.1: The USB Interface, wikipedia.org/Universal_Serial_Bus

6.1.2 USB data signal routing

The guidelines that were followed in the KybMo design are:

- Route the high-speed USB signals on a layer adjacent to the ground plane, whenever possible.
- Route the high-speed USB signals using a minimum of vias and corners. This reduces signal reflections and impedance changes.
- When it becomes necessary to turn 90°, use two 45° turns or an arc instead of making a single 90° turn. This reduces reflections on the signal traces by minimizing impedance discontinuities[2].
- Do not route USB traces under or near crystals, oscillators, clock signal generators, switching regulators, mounting holes, magnetic devices or IC's that use or duplicate clock signals.
- Avoid stubs(branches from the signal) on the high-speed USB signals because they

[2]The topic of right-angle turns is discussed in closer detail in subsection 4.3.4. Even though it is concluded that 90 degree turns are not harmfull, there is no need to make a right-angle turn instead of two 45 degree turns unless it is somehow dictated by the topology or routing space. The KybMo uses 45 degree turns for the USB interface because there was never a need for 90 degree corners.

cause signal reflections. If a stub is unavoidable, then the stub should be less than 200 mils.

- Route all high-speed USB signal traces over continuous planes (VCC or GND), with no interruptions. Avoid crossing over anti-etch, commonly found with plane splits.
- D+/D- traces should always be matched lengths and must be no more than 4 inches in length(the shorter the better); otherwise, it may affect the symmetry required by the high speed differential scheme of the D+ and D- signals, and the energy of the signals may decrease.
- Route D+/D- traces close together for noise rejection on differential signals, parallel to each other and **within two mils** in length of each other (start the measurement at the chip package boundary, not at the balls or pins).
- A high-speed USB connection is made through a shielded, twisted pair cable with a differential characteristic impedance of 90 Ohm ±15%. In layout, the impedance of D+ and D- should each be 45 Ohm ± 10%.
- D+/D- traces should not have any extra components to maintain signal integrity. For example, traces cannot be routed to two USB connectors.

The USB 1.1 interface was also routed in accordance to the same rules, even though the requirements aren't as strict as they are for USB2.0.

6.1.3 Differential pair routing of the USB2.0 interface in Altium

After defining a differential pair in the schematics, it becomes possible to use the interactive differential pair routing tool.

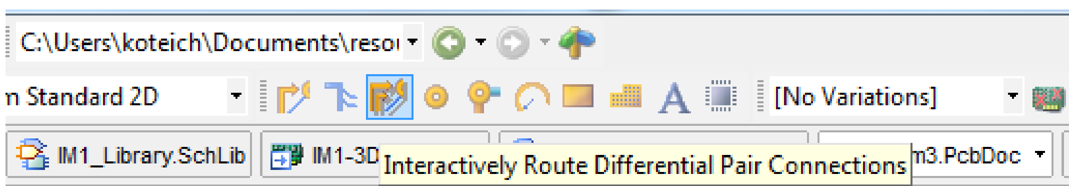


Figure 6.2: Altium differential pair routing tool

This tool basically allows you to route both traces in the differential pair along each other at the same time. The spacing used between the traces can be defined in the design rules. If no rule is defined, the tool uses the minimum spacing allowed between traces, which is defined as 4 mils in this case.

The length of the differential pair can be monitored in the differential pairs editor.

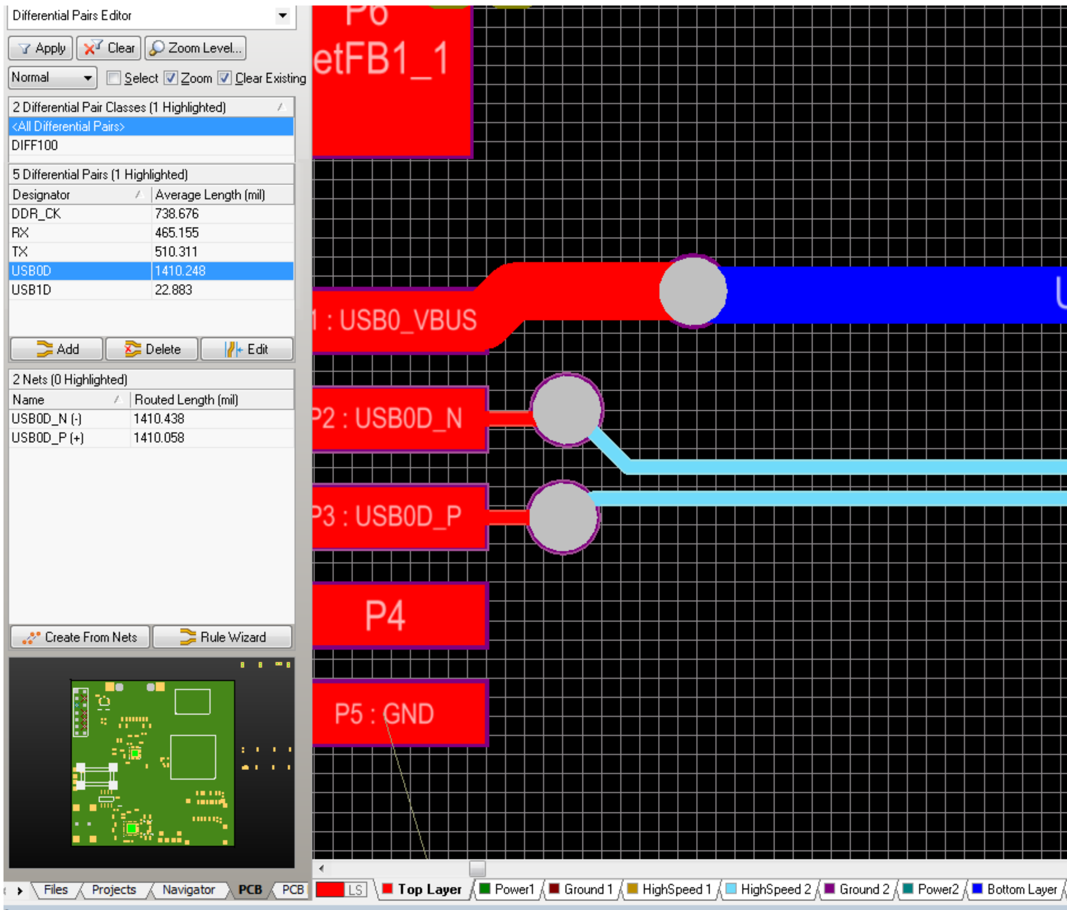


Figure 6.3: Altium Differential Pair Editor

Figure 6.3 shows that the USB0 trace lengths were precisely matched, measured from the chip boundary to the connector. (the connections from the chip boundary into their respective balls were temporarily deleted in order to obtain a measurement starting from the chip boundary).

The USB traces were not routed over any splits in the ground plane. However, keeping the signals away from the crystal was a bit challenging.

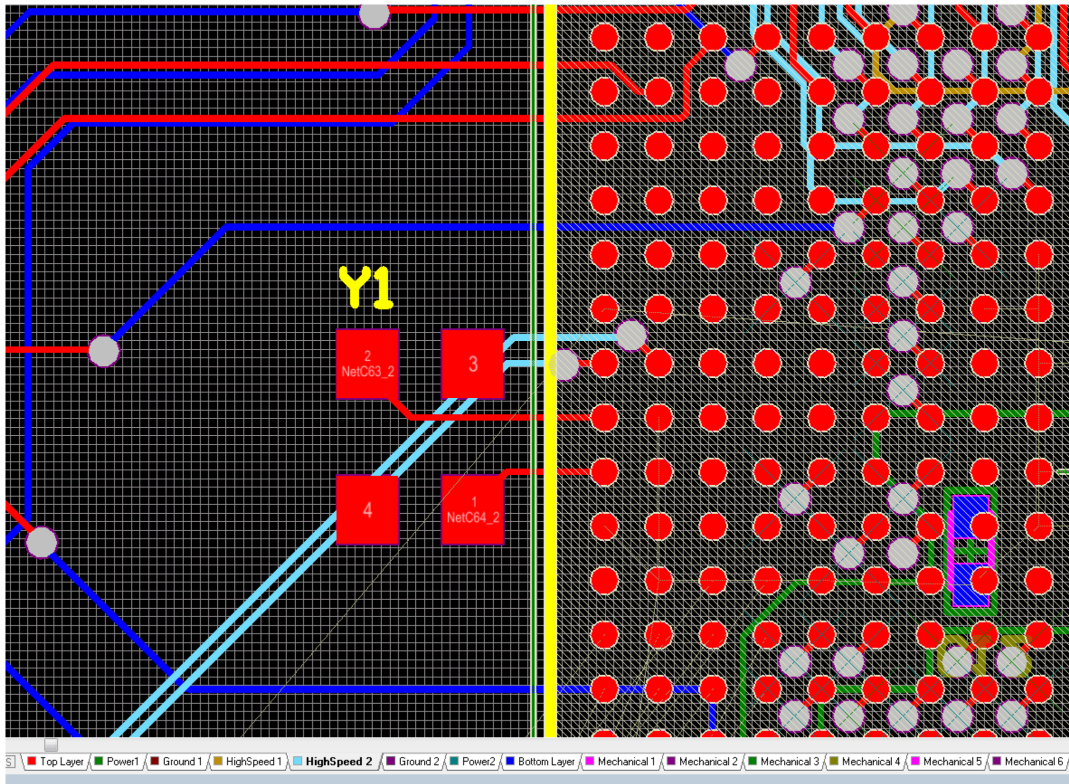


Figure 6.4: Y1: 24MHz Crystal, Blue lines: USB2.0 Differential pair

Routing the signals under the crystal would result in the shortest track length for the USB2.0 interface, however, this is not allowed due to the differential pair being sensitive to noise. The signal switching during high-speed operation is around 400 mV+- 10%, which means that any differential noise picked up can affect the received signal. The solution was to move the crystal down as far as possible, and circumvent it while keeping the USB trace length as short as possible.

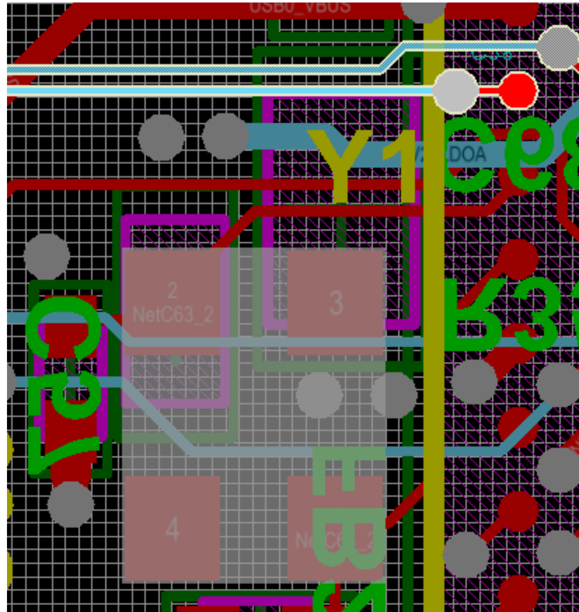


Figure 6.5: Circumventing the crystal in the KybMo design.

There was some uncertainty in regards to what the minimum clearance requirement between the signals and the crystal was. An inquiry was made at TI's forums, but no specific answer was provided. The maximum distance that was possible to achieve without affecting the USB trace length too much was 57mils. Examining the HawkBoard design files showed that the distance between the USB lines and the crystal was around 10-20 mils, which is an indication that 57 mils should be OK.

The topic of how the impedance control was carried out for all of the peripherals is described in subsection 7.4.1.

6.1.4 Connector

The KybMo uses micro-usb surface mount technology (SMT) connectors for both USB peripherals. SMT connectors were chosen over connectors that utilize mounting holes in the PCB. While utilizing mounting holes would provide connector mounts that are more solid, the disadvantage is that the space beneath the connectors on the other side of the board can't be used for routing and component placement.

the micro USB connector was chosen due to it's small form factor.

6.2 Ethernet

The Ethernet PHY used on the KybMo is the SMSC LAN8720AI-CP(24 pin 4x4mm, QFN-package[3]). The device utilizes the Reduced Media Independent Interface (RMII) to communicate with the processor. This is a deviation from most of the reference designs, such as the Logic-PD SOM-M1 and the Hawkboard, who utilize the Media Independent Interface (MII) interface. The MII interface use 16 signals, while the RMII interface uses 10. Thus, using a PHY that supports RMII saves routing space and implies smaller PHY size.

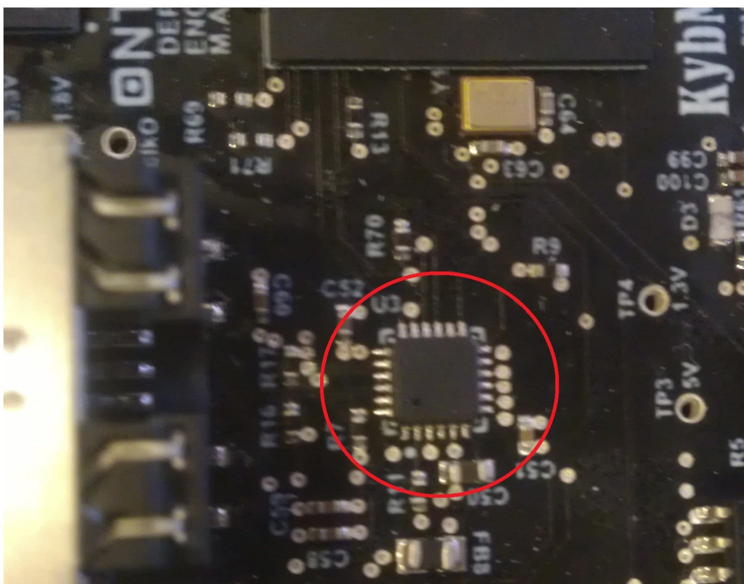


Figure 6.6: LAN8720AI PHY.

[3]Read more on QFN packages in subsection 8.2.3

Figure 6.7: KybMo, Ethernet schematics.

The schematics for the Ethernet interface can be viewed in closer detail on the CD.

6.2.1 Connector & Magnetics

Traditionally, a magnetics chip is used between the PHY and the connector. However, connectors with integrated magnetics are becoming more and more popular. An RJ45 connector with integrated magnetics was chosen to be used on the KybMo to simplify the design process and board layout.

The functions of Ethernet magnetics [Ceetus, 2006, TI.SNLA107, 2008]:

- Electrical isolation per IEEE 802.3 specification.
- Transfer of Ethernet signals without distortion.
- Electro Magnetic Interference (EMI) suppression.

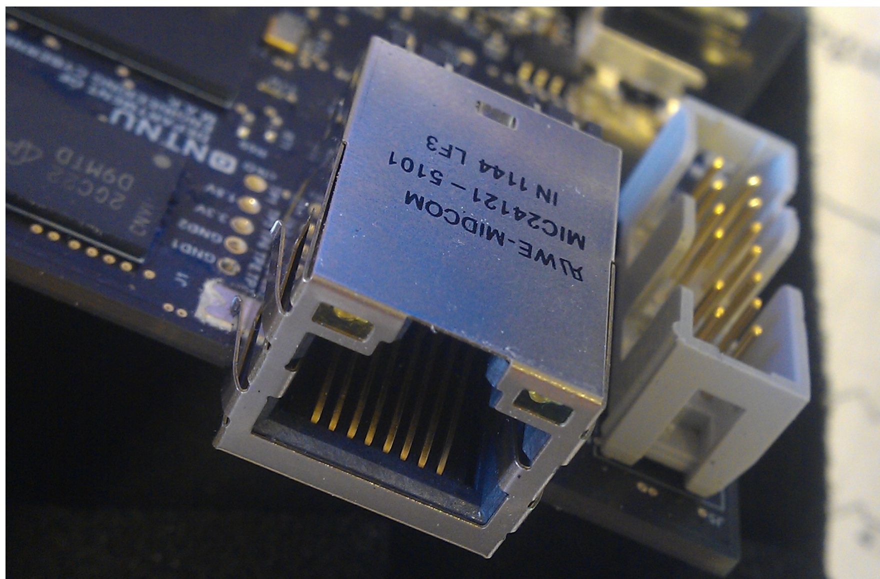


Figure 6.8: RJ45 with integrated magnetics, WE-LAN 7498011121

6.2.2 Strapping resistors

Strapping resistors are resistors that are put on certain signals to either pull the signals up or down. The chip latches the values on these signals at boot up and configures itself in accordance to how the strapping resistors have been set up. The configuration includes speed settings (base 10, base 100, or auto negotiation), internal voltage regulation or not, and sourcing of clock or not. The PHY on the KybMo is configured in accordance to the datasheet for:

- 100Base-Tx
- Full duplex
- auto negotiation disabled
- no sourcing of 50Mhz clock, the SoC sources the clock.
- Internal regulator enabled to provide 1.2V for the internal circuitry from a 3.3V source.

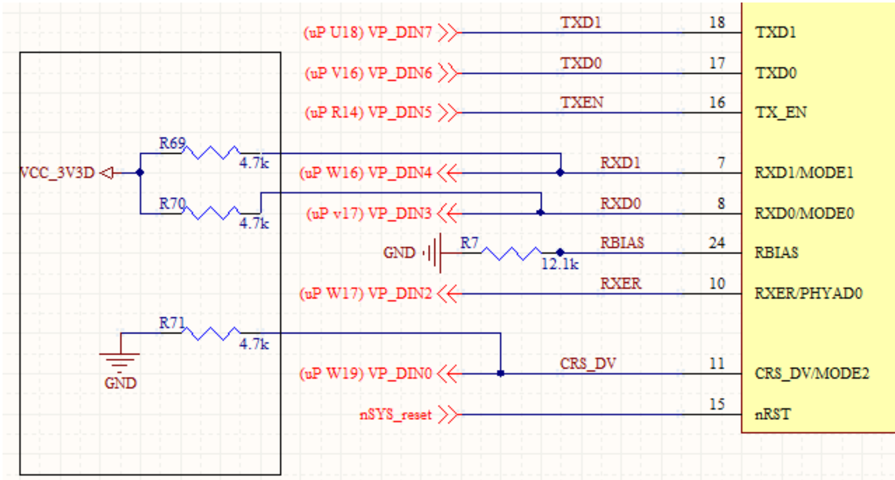


Figure 6.9: Mode selection straps (Mode[2:0] = 011. 100Base-TX, full duplex, auto negotiation disabled

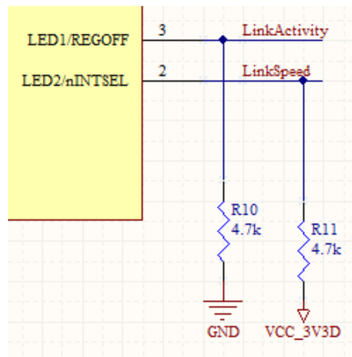


Figure 6.10: Mode selection straps (REGOFF = 0, regulator enabled. nINTSEL = 1, no sourcing of clock.

6.2.3 Topology & Placement

The guidelines for the Ethernet interface are mainly specified in [SMSC-AN18.6, 2008]:

- The distance between the PHY and the RJ45 connector needs to be 25mm (1000 mils) or greater. This rule is considered good design practice for EMI considerations. The intention is to isolate the PHY from the magnetics.
- The crystal oscillator and its resistors and capacitors must be placed within 12mm (500 mils) of the PHY.

- The power supply capacitors must be placed within 12mm (500 mils) of the PHY power supply pins.
- Keep the PHY device and the differential transmit pairs at least 25mm (1000 mils) from the edge of the PCB, up to the connector. The differential pair should be routed to the back of the connector, away from the board edge.
- The 49.9 ohm pull-up resistors on the differential lines, TXP/TXN and RXP/RXN, must be placed within 10mm (400 mils) of the PHY device. This ensures the transmit path is identical between the TX and RX.
- The RMII signals associated with each port (TX or RX) should be independently matched in length to within 6 mm (240 mils)
- The strapping resistors need to be located within 20mm (800 mils) of the Ethernet PHY to ensure the voltage into the pin at boot-up is at the correct level.
- Under no circumstances should a ground plane exist under the magnetics, the RJ45 connector, or in between the magnetics and RJ45 connector.
- Try to route as much of the TX and RX differential pairs with no planes under them.

The rules listed above resulted in the topology shown in Figure 6.11, where some deviations from the rules had to be made due to practical limitations imposed by the geometry of the board.

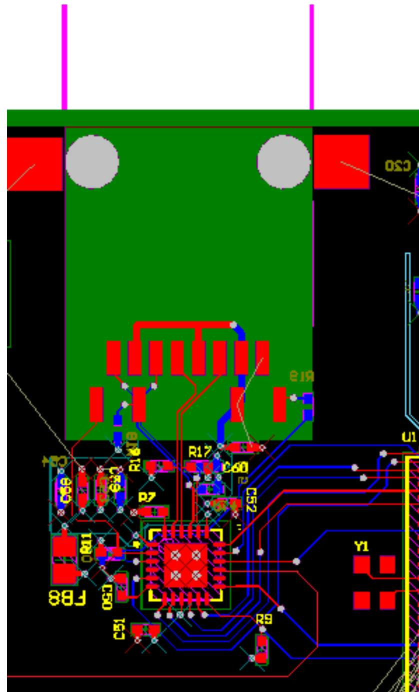


Figure 6.11: Ethernet Interface Layout. Green square denotes the absence of ground and power planes under the connector, and sections of the differential traces.

Deviation from the topological rules

If the distance guidelines between the PHY and the connector, and between the PHY and the board edges were to be followed, a larger board size would be required. Furthermore, Several reference designs were found that use the MII variant of the same chip, where these distancing guidelines were not followed(Beaglebone & Hawkboard), which means that the system can still function even though if some of these guidelines aren't followed to the letter. The distance between the PHY border and the RJ45 connector on the KybMo is approximately **477 mils**, almost half of the recommended distance, but still more than the Beaglebone's approximate of 147 mils. The distance between the PHY border and the closest board edge is approximately **812 mils**, approximately 200 mils less than the recommended distance, but still more than the Beaglebone's approximate of 740 mils. Figure 6.14 and Figure 6.13 show the measured distances on the KybMo and the Beaglebone.

In addition, the guidelines in [SMSC-AN18.6, 2008] only mention that no ground plane should exist under the connector. However, the general idea is that no plane regardless if it's a power or ground plane should exist, in order to avoid coupling of noise with the magnetics and planar interference of noise on the differential pairs. This can be confirmed by several sources[4], and by the lack of power planes under the connectors in the reference designs. There is therefore no ground or power plane that extend under the RJ45 connector on the KybMo.

[4][TI.SNLA107, 2008], [Micrel.AN-111, 2007], [Micrel.AN-139, 2007], [Panasonic, 2011a]

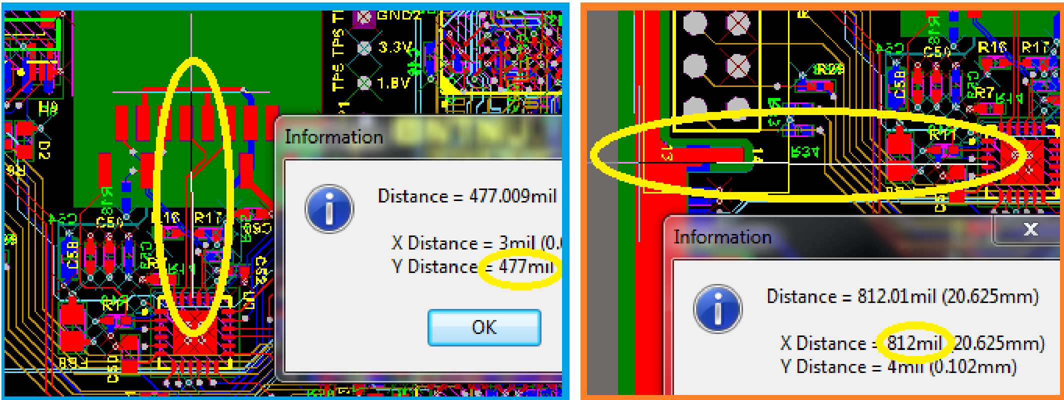


Figure 6.12: KybMo PHY Distance from RJ45 connector and board edge (Altium Designer)

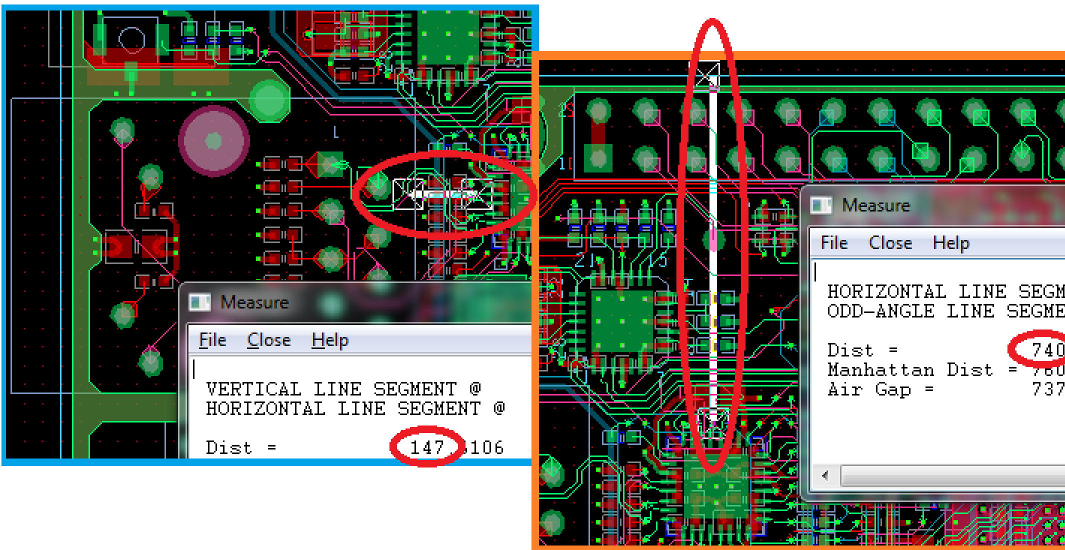


Figure 6.13: Beaglebone PHY Distance from RJ45 connector and board edge (Allegro Free Physical Viewer)

6.2.4 Routing Guidelines for the TX and RX Differential Pairs

- The differential pairs need to be designed with 100 ohm differential impedance[5], with minimal distance between the positive and negative nets within the differential pair.
- Differential pair nets must maintain symmetry. TXP and TXN must be of approximately equal length and symmetric in regards to shape, length, and via count. For example, if TXP goes through a via at 8mm, then TXN should also go through a via at 8mm. The same applies to the RXP and RXN nets.
- The TX/RX traces must be isolated from nearby circuitry and signals. Maintain a distance of parts to lines that is greater than or equal to 5 times the distance of the spacing between the traces. Do not route the TX/RX traces under parts. Do not cross TX/RX lines with other PCB traces unless they are on the opposite side of the ground plane.
- To avoid crosstalk, the TX and RX differential pairs should be routed on different layers.

The TX/RX pairs were routed according to the guidelines with no deviations.

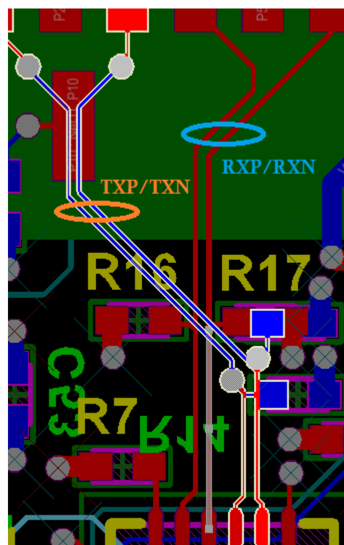


Figure 6.14: TX and RX pairs

[5] Impedance control is discussed in closer detail in subsection 7.4.1

6.2.5 Filtering Analog Vcc

The LAN8720 has analog power supply pins and therefore require Vcc filtering in order to stop high frequency noise from affecting the rest of the board's circuitry. The KybMo dedicates an isolated 3.3V island in the power plane to the LAN8720. This island is connected to the main 3.3V plane through a ferrite bead. Basically, ferrite beads are used to filter high frequency noise by increasing their impedance as a function of the frequency[6].

The main ferrite bead used for filtering on the KybMo is the MMZ2012S121A from TDK. This bead has 120 ohm impedance at 100MHz, and a maximum current rating of 0.8A. 120 - 150 ohm impedance beads seem to be a common choice for this chip (Beaglebone, Hawkboard, LCDK). The Hawkboard and the LCDK both use the MMZ2012S121A, and it was therefore chosen for the KybMo.

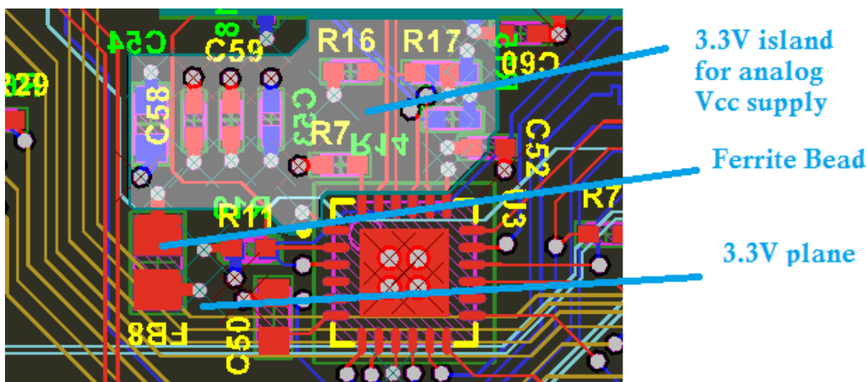


Figure 6.15: KybMo, ferrite bead for Vcc filtering

6.3 microSD

The KybMo utilizes a 4-bit SD interface with a programmable clock (0-52 MHz). The implementation of the interface is pretty straight forward, and no official routing guideline exist. However, there is no reason for why high-speed design practices such as good spacing, minimal track distance, minimal via count, and approximately matched net-lengths shouldn't be followed.

The schematic checklist for OMAP-L138[7] recommends weak pull-up resistors[8] on all the

[6] [TI.SCAA048, 2001], [Altera.AN583, 2009], [Johnson.Ferrite_Beads, 2000]

[7] http://processors.wiki.ti.com/index.php/OMAP-L13x_/_C674x_/_AM1x_Schematic_Review_Checklist

[8] Weak pull-ups: A "weak" pull up resistor is a resistor with a high value, usually in the range of 4-50K ohms. It "weakly" pulls the voltage up to it's own voltage level, meaning that it takes little current to over-ride it.

signals. Studying several reference designs shows that a pull-up resistor value of 10k is common, thus making it the preferred value for the SD interface pull-ups on the KybMo.

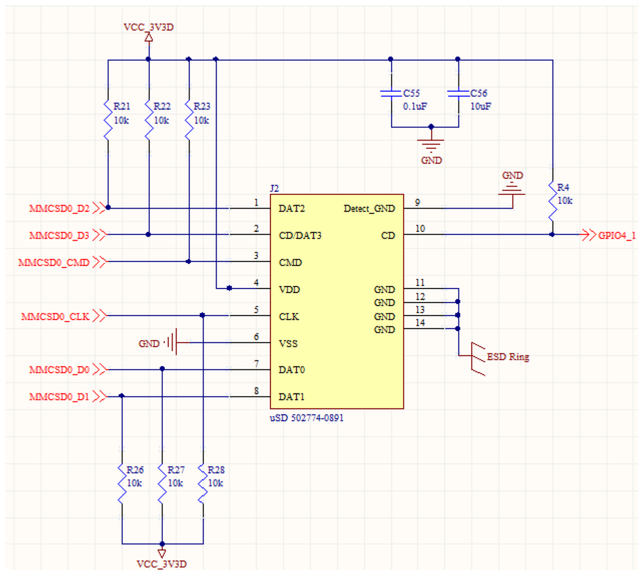


Figure 6.16: microSD schematics

A Card-Detect signal is routed from the connector to the GP4[1] pin on the OMAP-L138. The value of this pin is used to indicate whether a card is present or not, and can be monitored in software (1 = Card is present, 0 = No card is present).

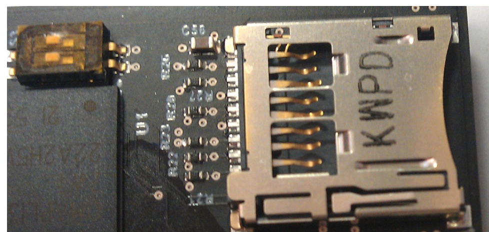


Figure 6.17: KybMo microSD Connector: Molex 5027740891

6.4 JTAG

If the distance between the component and the JTAG connector is more than 3 inches, or if there are several devices in the scan chain, a buffered configuration has to be used. This is the case for the Hawkboard. The KybMo has only one device in the scan chain and the distance from the connector to the processor is less than 3 inches. Thus, the unbuffered configuration described in [TI.SPRU655H, 2012] in conjunction with the XDS target connection guide [TI.Wiki.Target.Connection.Guide, 2012] was used.

The JTAG interface is designed to be used with a TI 14 pin header utilized by the XDS 100v2 Emulator.

Pin	TI 14-pin
1	TMS
2	TRST
3	TDI
4	TDIS (GND)
5	TVRef
6	KEY
7	TDO
8	GND
9	RTCK
10	GND
11	TCK
12	GND
13	EMU0
14	EMU1

Figure 6.18: TI 14 pin header pin-out overview

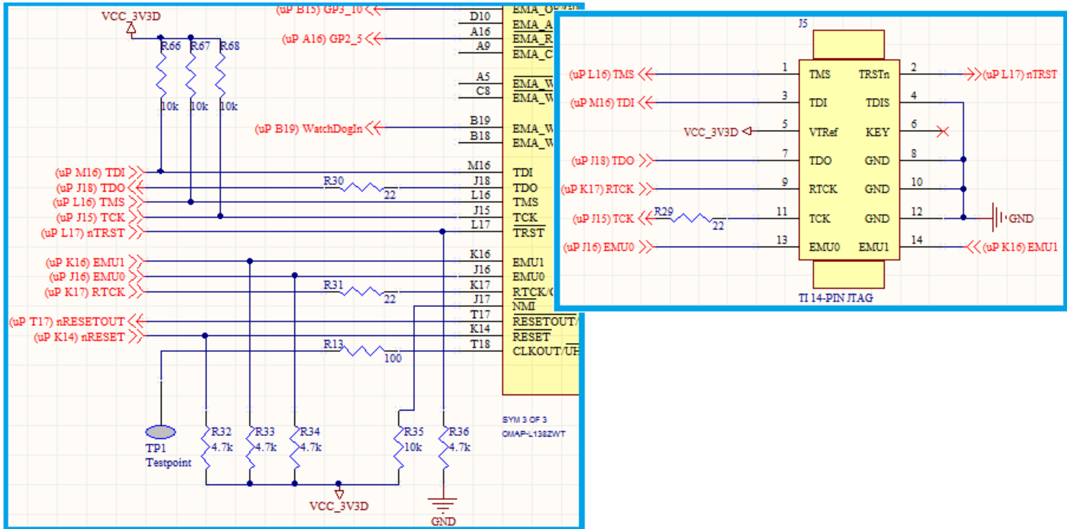


Figure 6.19: KybMo, JTAG interface schematics

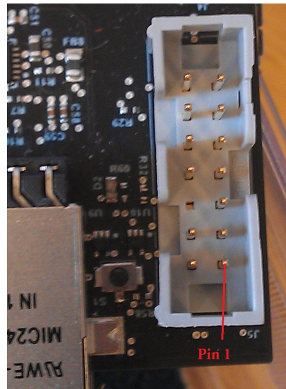


Figure 6.20: TI 14 Pin Header,3M MHB14K-ND

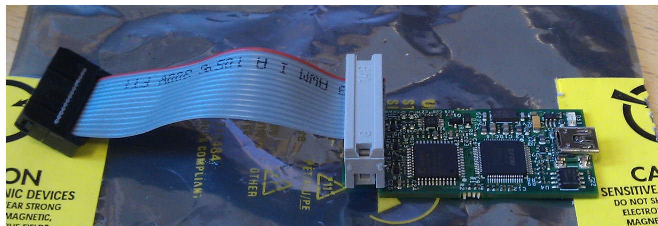


Figure 6.21: Spectrum Digital XDS100v2 emulator

6.5 UART

The Universal Asynchronous Receiver/Transmitter (UART) interface on the KybMo utilizes the UART0 port on the OMAP-L138 SoC. In all the reference designs for the OMAP-L138, the UART is passed through the classical RS-232 interface for communication with a PC. The most common connector used for RS-232 communication in embedded designs is the DE9 connector.

Figure 6.22: DE9 connector, female. www.solentcables.co.uk

While this is a common connector found in many designs, it is bulky and requires a fair share of space on a printed circuit board. Furthermore, some laptops have started to exclude DE9 connectors from their designs. An alternative solution is to pass the UART communication through a USB interface instead. The FTDI FT232 IC is a popular device used in such applications. Using a USB interface instead of RS-232 reduces the space required on the circuit board, in addition to ensuring compatibility with all types of PCs and laptops, as USB ports are always supported.

The KybMo uses the QFN version (5x5mm) of the FTDI FT232 USB to UART IC (FT232RQ) hooked up to a micro USB port in order to use as little space as possible. The IC is located on the bottom side of the board,and is connected to two LEDs on the top side of the board, where the orange LED indicates reception of data, and the green LED indicates transmission of data.

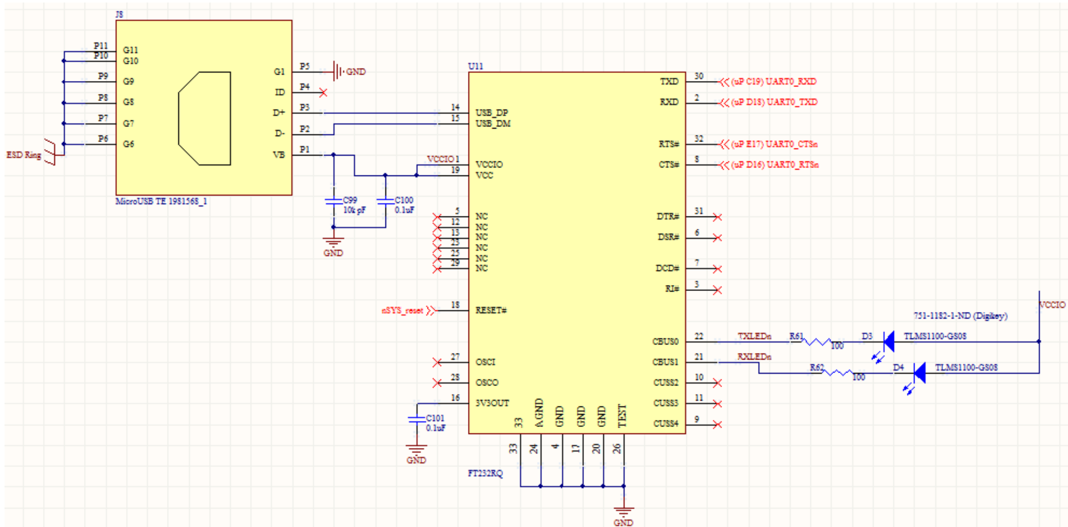


Figure 6.23: KybMo, USB to UART FT232 schematics

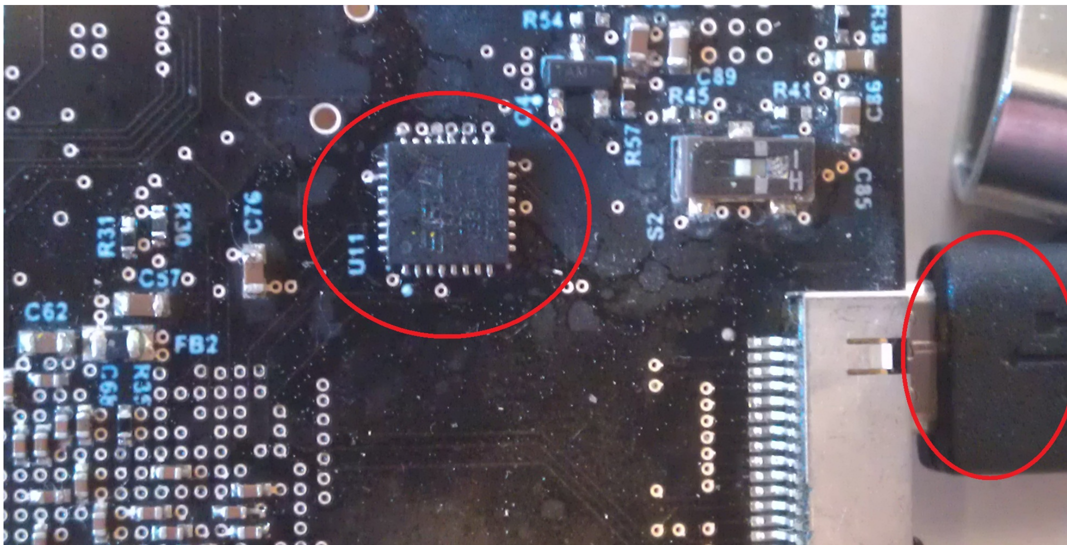


Figure 6.24: KybMo (bottom side), FT232RQ

6.6 I/O Expansion Header

The I/O expansion header on the KybMo is located on the bottom side of the board, and is a 31 pin header from JAE electronics.

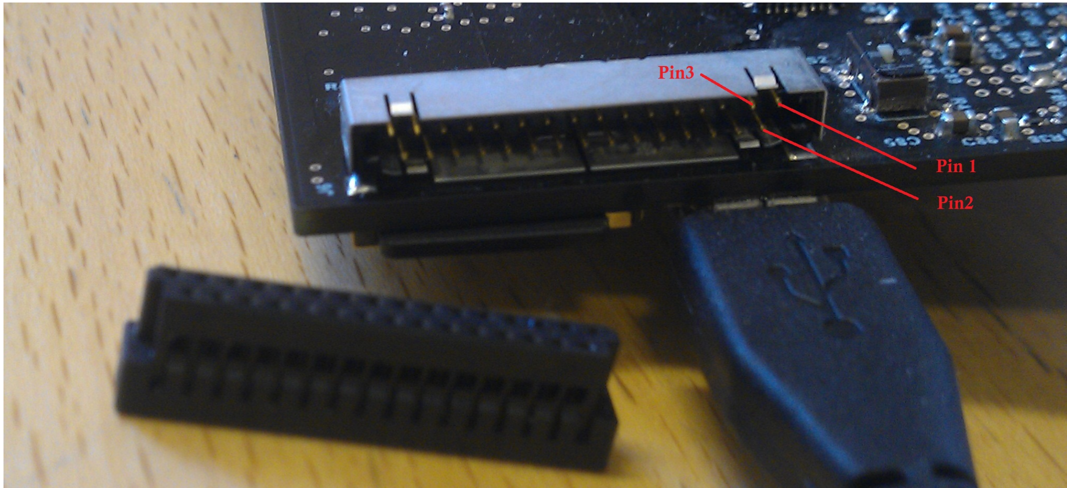


Figure 6.25: KybMo (bottom side), JAE electronics FI-WE31P-HFE with plastic mate FI-W31S

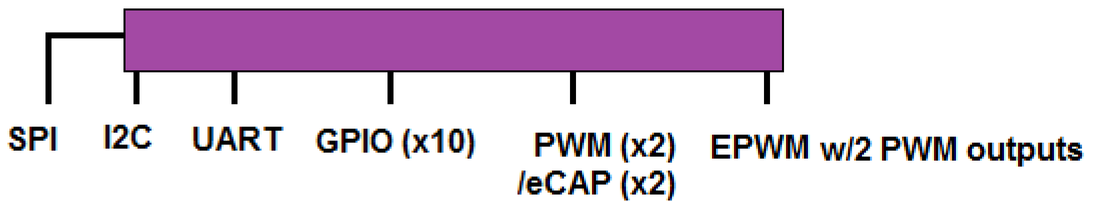


Figure 6.26: I/O expansion header block diagram

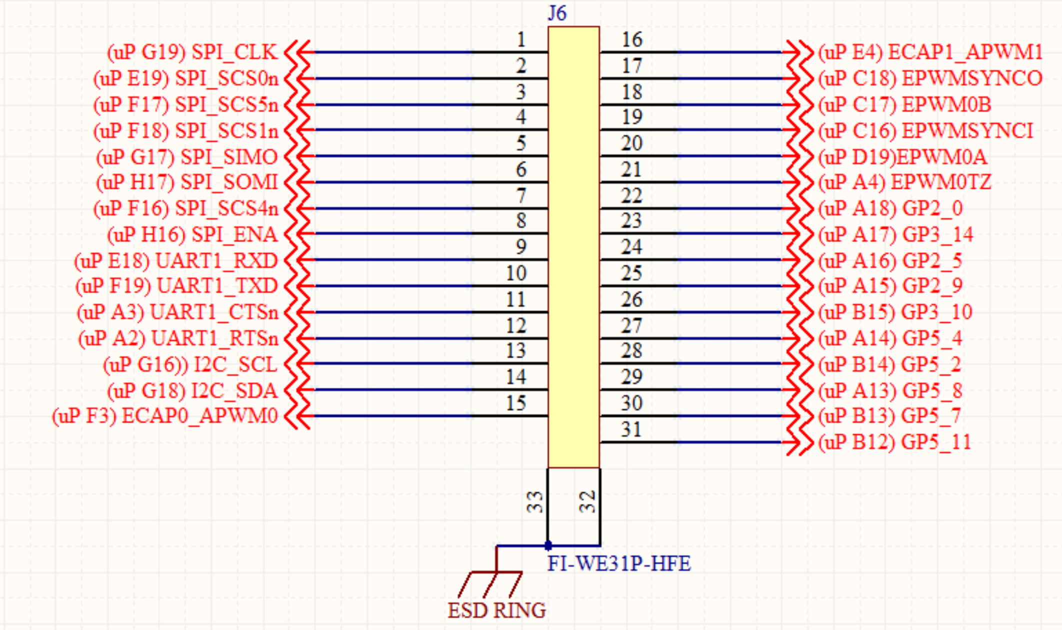


Figure 6.27: I/O expansion header schematics

It is important to note that the block diagram configuration shown in Figure 6.26 only represents a suggestion to one specific configuration of a number of possible configurations. Most of the signals on the expansion header are multiplexed with several peripheral functions, which allows for different configurations based on the application needs.

Multiplexed functions

- Pins 1, 5, 6,8: SPI1 channel or General purpose I/Os
- Pins 2 and 4: SPI1 CS/EPWM1/General Purpose I/Os
- Pins 3 and 7: SPI1 CS/UART2/I2C1/General Purpose I/Os
- Pins 9 and 10: UART1/SPI1 CS/General Purpose I/Os
- Pins 11 and 12: UART1 RTS-CTS/General Purpose I/Os
- Pins 13 and 14: I2C0/SPI1 CS/General purpose I/Os

- Pins 15 and 16: eCAP_APWM1-2/General Purpose I/Os
- Pins 17 to 19: EPWM0/SPI0/Generap Purpose I/Os
- Pin 20: EPWM0B/SPI0_ENA
- Pin 21: EPWM0TZ/eCAP_APWM/General Purpose I/O

This means it's possible to configure the expansion header with up to 7 PWM(3APWM + 4 EPWM) outputs, two separate SPI channels, two separate I2C channels, 2 separate UART channels or up to 30 general purpose I/O signals.

The suggestion of 10 general purpose signals in Figure 6.26 is to provide the possibility of having a fixed 8 bit bus, with one R/W bit, and one Address/Data bit.

"EPWM" abbreviates "Enhanced Pulse Width Modulation", which is a TI specific solution that provides high resolution PWMs with independent resources in order to minimize CPU overhead. Each EPWM instance has two PWM outputs associated with it: EPWMxA, EPWMxB. These two outputs can be configured as:

- Two independent PWM outputs with single-edge operation
- Two independent PWM outputs with dual-edge symmetric operation
- One independent PWM output with dual-edge asymmetric operation

"eCAP" abbreviates "enhanced capture", and is a dedicated input peripheral for use cases such as:

- Sample rate measurement of audio inputs
- Speed measurements of rotating machinery (for example, toothed sprockets sensed cia Hall sensors)
- Elapsed times measurements between position sensor pulses
- Period and duty cycle measurements of pulse train signals
- Decoding current or voltage amplitude derived from duty cycle encoded current/voltage sensors.

It is recommended to use the eCAP module instead of other peripherals for applications that require precise timing measurements due to the fact that the eCAP module has dedicated resources specifically suited for these purposes.

"APWM" abbreviates "Auxiliary Pulse With Modulator", and can be used to generate PWM signals. The APWM modules support working in conjunction with each other for multichannel PWM generation with Synchronization or with Phase control.

Refer to [TI.SPRUH77, 2011] for more information about these modules.

6.7 Chassis Ground/Electro Static Discharge (ESD) Ring

Two important factors need to be taken into consideration in regards to the area surrounding the connectors and the board edges. These are electro magnetic interference (EMI), in terms of emitted radiation from the circuitry to the outside world, and electro static discharge (ESD), in terms of harmful discharge from the outside world into the circuitry.

First of all, a cable connector in a system can be a source for radiated noise. Any noise that originates from inside the system can couple through the connector to the chassis and to the cable [TI.SNLA107, 2008]. This is an issue that needs to be addressed in order to reduce emissions that might affect other circuitry. The second issue, is that a rapid current flow into the circuitry can occur if the connectors come in contact with an object that has a different electrostatic potential (this can sometimes be the human body for example). This is referred to as electro static discharge, and is observed as a high-voltage transient with fast rise time and fast decay time. Small levels of energy delivered quickly can easily cause the silicon in integrated circuits to melt and deform due to the material's inability to dissipate heat fast enough. The device might become fatally damaged or weakened [Micrel.AN-139, 2007].

The solution to both these problems is to connect the connectors in the system to a ground area, referred to as chassis ground. Chassis ground should be separated from the signal ground with a gap, but connected to the signal ground through a discrete component that equalizes the potential levels between both ground fields and provides a controlled discharge path. Various methods are described in the literature for what kind of discrete component to use and how it should be connected. [TI.SNLA107, 2008] suggests starting with a large 0 ohm resistor, and replacing it with either a ferrite bead or a capacitor if deemed necessary during testing. [Micrel.AN-111, 2007] & [Micrel.AN-139, 2007] recommends using a 1000pF/2KV capacitor. Inspecting the reference designs also show various ways of implementation both in regards to the layout of the chassis ground and to the usage of discrete components.

This caused some confusion in regards to how to deciding on an optimal implementation. An inquiry was made at TI's e2e forums in regards to their recommendations. The response was that it's a lot more complicated than just one right answer, but the good news were that it's hard to choose a method that will actually make the board not work[9].

[9] http://e2e.ti.com/support/dsp/omap_applications_processors/f/42/t/184990.aspx

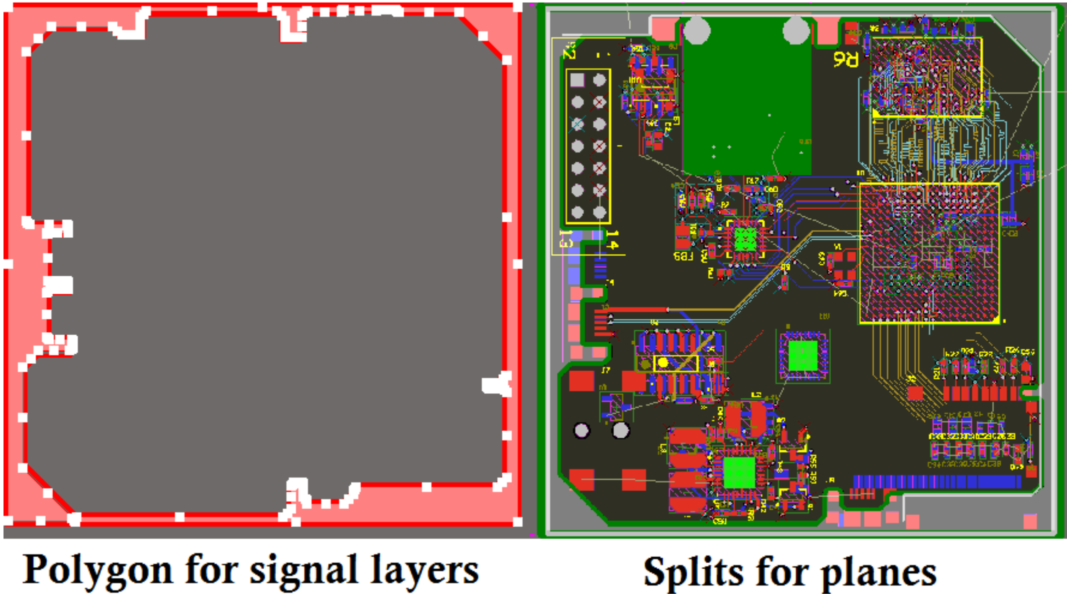


Figure 6.28: KybMo, signal layers had to be filled with a polygon around the board edge, while power and ground planes had to be split

In the end, the implementation of the Beaglebone was used: A planar ring surrounding the board edge, separated from the signal ground with a gap, and only connected through a current sensing resistor that increases it's resistance as a function of the current. This seemed to be the tidiest solution, Figure 6.28 shows how this was implemented on the KybMo. The Hawkboard uses an isolated ground island under the Ethernet chassis, where the island is connected to signal ground through a ferrite bead. The other connectors on the board all have their own ferrite beads that connects them to signal ground. This seemed like an excessive use of components. The LCDK use 0 ohm resistors between it's connectors and signal ground, and SOM-M1 from Logic PD uses a direct connection to ground with no separation. Thus, the Beaglebone's implementation seemed as the most reliable (seperating chassis ground from signal ground according to the litterature) and the most convenient as it only uses one discrete component; the current sensing resistor.

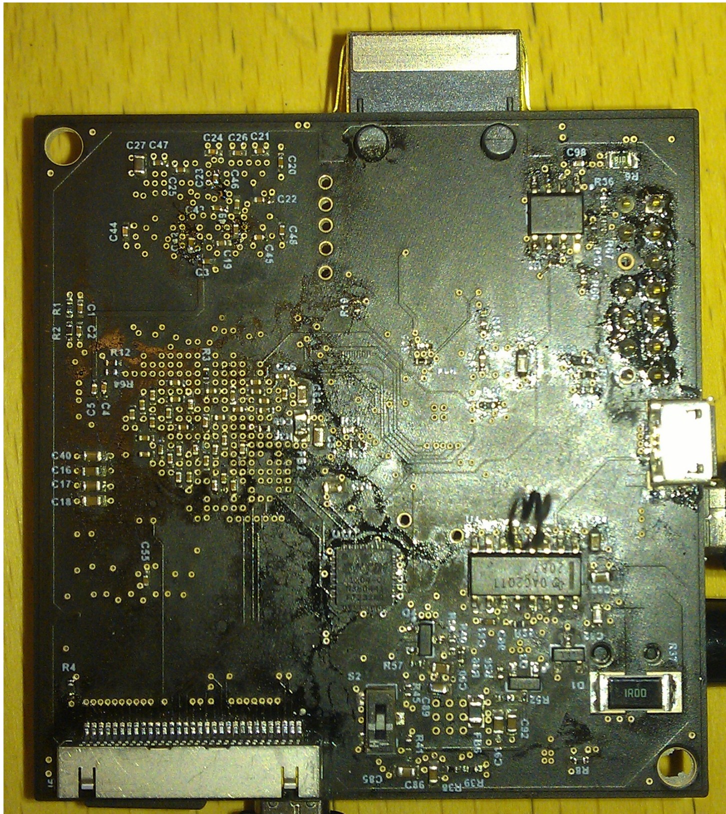


Figure 6.29: The contours of the ESD ring can be seen on the KybMo PCB

It is important to mention that the chassis ground/esd ring should exist on all the layers, whether they are signal layers or power and ground planes. The ring on the different layers should be connected through several vias and mounting holes, as seen in Figure 6.29.

Chapter 7

Fabrication

7.1 Introduction

This chapter, and the next one, discusses the practical challenges that were encountered in terms of fabrication of the PCB, and the assembly of the PCB and the components. These were demanding processes and an important part of the project. The knowledge and experiences gained through this stage of the project can be used to the benefit of The Department of Engineering Cybernetics and future student projects.

7.2 PCB manufacturing

PCB manufacturing is limited to two layer boards at NTNU. An external manufacturer is required for the production of PCBs with more than two layers. A simple Google search will yield a list of manufacturers. This project used a Chinese manufacturer, www.pcbcart.org, based on the recommendations and prior experiences of several co-students. It was also the only site with the required limitations and capabilities (discussed in subsection 7.2.1) that offered an on-line price suggestion for more than 4 layers, without having to resort to direct communication with a salesperson.

7.2.1 Limitations of the manufacturer

Different PCB manufacturers use different machinery and methods, resulting in different capabilities and limitations. It is important to be aware of these capabilities and limitations before starting the PCB layout design process. The most important aspects in terms of the manufacturer's limitations that can affect the layout are:

- Minimum track width
- Minimum spacing between tracks.
- Minimum hole diameter.
- Minimum annular ring. (See Figure B.1)
- Maximum number of layers.

PCBCart's Capabilities

Features	Capability
Layers	1-10 layers
Material	FR4
Copper Thickness	1/2 to 2oz (18um-70um)
Board Thickness	.016-.126" (0.4mm-3.2mm)
Surface finish	HASL,Ni/Au,OSP
Soldermask	LPI, different colors
Board Dimension	600x700mm
Min Hole Diameter	8mil (0.2mm)
Min line width	8mil (0.2mm)
Min line spacing	8mil (0.2mm)
Min SMT pitch	16mil (0.4mm)
Min. Annular Ring	.008" (0.2mm)
Aspect Ratio	5:1
Surface/hole plating	ave. 25um min. 20um
Tolerance:	
Hole Tolerance (PTH)	.002" (0.05mm)
Hole Tolerance (NPTH)	.003" (0.075mm)
SM Tolerance (LPI)	.003" (0.075mm)
Dimension	.004" (0.1mm)
Electrically test	10V-250V, flying probe or testing fixture

Figure 7.1: Standard PCB capabilities, www.pcbcart.com

Features	Capability
Material	Flex, Aluminum Base, Rogers
Copper Thickness	3 to 6oz (105um-210um)
Min line width	4mil (0.1mm)
Min line spacing	4mil (0.1mm)

Figure 7.2: Special PCB capabilities, www.pcbcart.com

A designer is sometimes forced to impose restrictions on the design in order to meet the manufacturer's capabilities. Other times one has no choice than to find a manufacturer that can meet the requirements set by the physical nature of certain components. PCBCart's standard capabilities did not satisfy the requirements of this design. A successful BGA-breakout for the OMAP-L138 requires a minimum track width and line spacing of 4 mils in a lot of areas. The manufacturer's standard capabilities could only go as low as 8 mils for track width and spacing. However, their special capabilities could meet the project's requirements. The only thing required from the customer is to send an email with the design files after placing the order, specifying the need to use the special PCB capabilities in the email.

7.3 Ordering

Board Specification

Material	FR4	Layers	8
Material Details	Standard Tg 140C	Reorder Status *	--
Board Size (width) *	59.7 mm	Board Size (height) *	61.1 mm
Quantity	5 pcs	Different Design in Panel *	--
Route Process *	--	Thickness (Finished Board)	1.6 mm
Layer Stack	specified in PCB file	Layer Stack Details	Top Signal, PWR1, GND1, Signal, Signal, GND2, PWR2, Bottom Signal
Impedance Control	Yes as +-10%	Surface Finish	Immersion Silver - RoHS
Outer Layer Copper Weight (Finished)	35 um	Inner Layer Copper Weight	35 um
Min. Tracing/Spacing	0.10 mm	Min. Annular Ring	0.10 mm
Smallest Holes	0.20 mm	Holes Numbers	Over 600
Buried/Blind Vias	No	Times of Buried/Blind Via	--
Surface Mount	2 sides	Soldermask	Both Sides
Peelable Soldermask	None	Soldermask Color	Black
Matt Color (only add to Green or Black)	Black Matt	Silkscreen Legend	2 sides
Silkscreen Legend Color	White	Gold Fingers	No
Gold Fingers Chamfer	None	Slots in Board	No Slot in Board
Testing	Yes	UL Marking *	No
Date Code Marking *	No	Lead Time	in 12 days

Figure 7.3: KybMo PCB order specification, www.pcbcart.com

Figure 7.3 shows the details of the order that was placed for the KybMo PCB. The manufacturer's default values were chosen for the thickness of the board and the copper weight. No buried or blind vias[1] were used in the design, because they are more expensive and complicated to manufacture. Impedance control was required for the DDR2, USB2.0, USB1, and Ethernet interfaces. "Gold fingers" are the gold plated connection pads usually found on the periphery of PCBs that are designed to be inserted into connection slots, such as conventional PC RAM cards, and PCI cards. No such pads exist for this purpose on the KybMo. PCBCart produce PCBs that comply with the UL verification standard, and also offers UL marking on the PCBs. The UL label is usually printed on a product to show that the product has been tested and evaluated to meet UL requirements, which include product safety, environment, and life & health verification standards[2]. It was not important to have this label on the KybMo PCBs as they are not intended for the commercial market.

7.3.1 Finishes

Professionally manufactured PCBs have their copper connection pads protected by a chemical finish. Otherwise, the oxidation of the copper can be destructive to the PCB. Surface

[1] See Appendix A for a more detailed explanation of blind/burried vias.
[2] www.ul.com

finishes are used over the copper pads to provide protection from oxidation and optimal electric connectivity. Information about the different types of finishes is abundant on the Internet. Below is a short description of the surface finishes that were available at pcbcart:

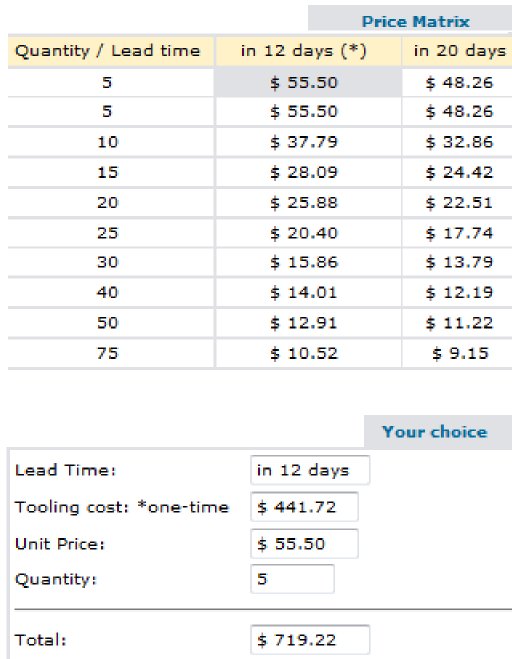
Information from www.multicircuits.com and www.trianglecircuits.com:

- **Electroless Nickel Immersion Gold (ENIG):** One of the best finishes available. However hard to manufacture, not cheap. Quality issues may occur if the manufacturer does not control the process properly.
- **Immersion Silver:** Good wettability[3], acceptable cost. Flat surface. Thickness range from 5 - 12 mils (the thinner the better). Good shelf life. Increasing in popularity. Must be packed using sulphur-free paper.
- **Immersion tin:** Almost same qualities as immersion silver, but thicker (20-40 mils), and has poorer shelf life.
- **Organic Solderability Preservative (OSP):** Inexpensive, easy to apply. Limited number of heat cycles, degrade with high temperatures, poor wettability.
- **Pb-free Hot Air Solder Leveling(HASL):** Inexpensive and excellent for lower technology PCBs. Not suited for multilayer boards and fine pitches. uneven surface.

Immersion Silver quickly emerged as the best option for the KybMo. While ENIG is the superior finish, the risk of quality issues exist if the manufacturer isn't careful enough, and it's a more expensive process. Immersion silver is a popular, well tested, easier to produce, and cheaper alternative.

[3]Wettability in this case refers to how well the solder paste distributes it self on the surface area.

7.3.2 Price



Price Matrix		
Quantity / Lead time	in 12 days (*)	in 20 days
5	$ 55.50	$ 48.26
5	$ 55.50	$ 48.26
10	$ 37.79	$ 32.86
15	$ 28.09	$ 24.42
20	$ 25.88	$ 22.51
25	$ 20.40	$ 17.74
30	$ 15.86	$ 13.79
40	$ 14.01	$ 12.19
50	$ 12.91	$ 11.22
75	$ 10.52	$ 9.15

Your choice	
Lead Time:	in 12 days
Tooling cost: *one-time	$ 441.72
Unit Price:	$ 55.50
Quantity:	5
Total:	$ 719.22

Figure 7.4: KybMo PCB cost, www.pcbcart.com

Refer to chapter 10 for a discussion on the price of the KybMo.

7.4 Post Ordering Communication

After the order had been placed, communication was established with the manufacturer in order to exchange the remaining required information.

7.4.1 Impedance Control

The KybMo design requires impedance control on several of it's high-speed nets. Figure 7.5 was sent to the manufacturer in order to describe the desired qualities of these nets.

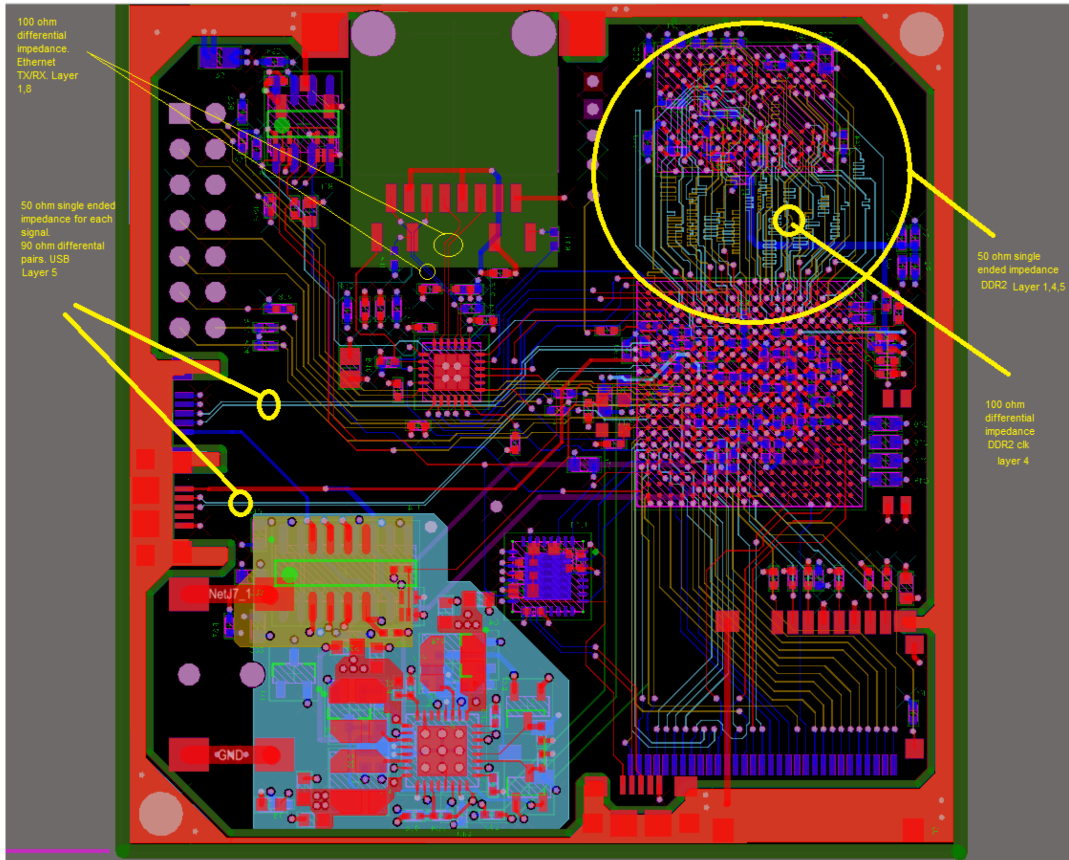


Figure 7.5: Impedance control information sent to manufacturer

The manufacturer then performed the necessary calculations and returned with the following results:

Layer	Tracing/Spacing mil	Calculated ohm	H1(mil)	Er1	H2(mil)	Er2
L1	4 mil	58	3.72	3.85		
L1	4/6.2 mil	98	3.72	3.85		
L4	4 mil	60.5	15.99	3.75	6.14	3.65
L4	4/6 mil	100.1	15.99	3.75	6.14	3.65
L5	4 mil	60.5	15.99	3.75	6.14	3.65
L5	4/6 mil	100.1	15.99	3.75	6.14	3.65
L5	4/4 mil	90	15.99	3.75	6.14	3.65
L8	4/4 milto 3.2/4.8 mil	99.4	3.72	3.85		

Figure 7.6: Calculated Impedance values from manufacturer

Figure 7.6 Shows that the calculation of possible impedance values deviate from the desired values. The nets that were desired to be 50 ohm in Figure 7.5 had a tolerance of +/-10% specified in their data-sheets, meaning that the impedance on the net can vary within +/-10% as the signal travels through it. In this case, the impedance would vary from 58 to 60.5 as the signals move from one layer to another. This is within the +/-10% tolerance range. Furthermore, the data-sheets for these nets accept controlled impedances in the range of 50-75 ohm. 50 ohm was specified as the desired value, but even if that could not be accommodated, 58/60 ohm is still within the allowed range. The red row in Figure 7.6 indicates that the manufacturer had to make changes in the trace width and spacing in order to reach the desired impedance value.

However, some changes had to be done in the original design files as well in order to meet the calculated values. For example, the USB 1.1 lines on Layer 5 had 4/6 tracing/spacing in the design. This resulted in a calculated impedance of 100.1. But the desired impedance of these USB lines was 90 ohm. The calculations that were made in Figure 7.6 shows that the USB2.0 lines on layer 5, who had 4/4 tracing/spacing was able to reach 90 ohm differential impedance. Thus, the USB 1.1 lines were changed from 4/6 to 4/4. The same principle was followed for some of the other nets as well. Figure 7.8 shows the figure that was sent to the manufacturer after the changes were made.

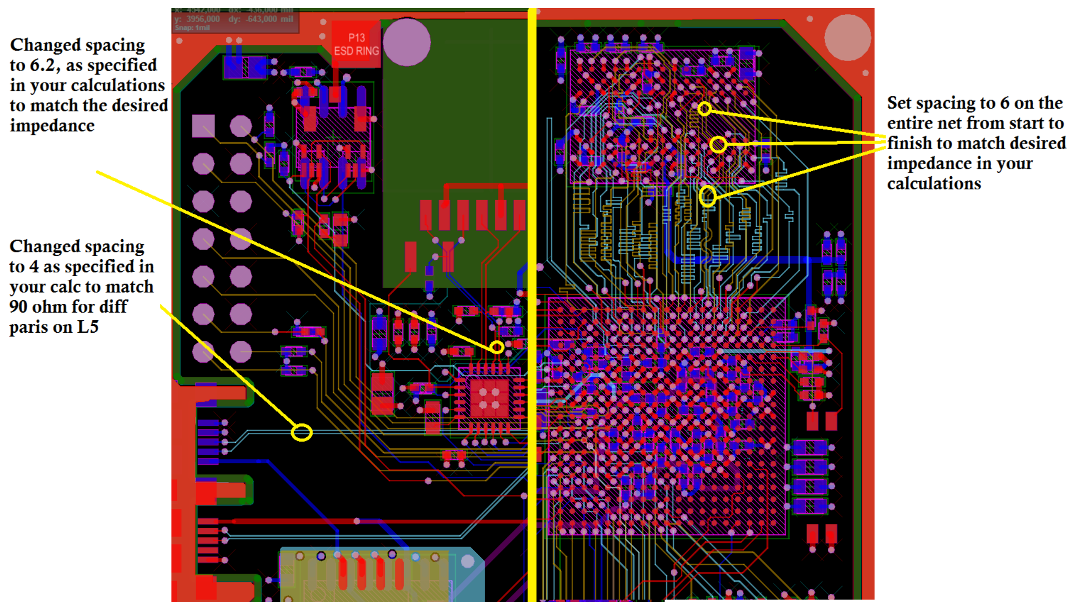


Figure 7.7: Modified parameters sent to manufacturer

7.4.2 VIA tenting

Via tenting(covering the via with solder mask), which is discussed later in subsection 8.2.3, should be used to avoid short circuits in dense areas and to prevent solder paste from slipping into the via holes under QFN packages. Surprisingly, the contact person from the manufacturer didn't quite understand this concept (tenting vias under QFN packages) and was asking if a mistake was made in the design. The reason behind the tenting had to be explained before the contact person was willing to comply with the desire to tent those vias.

It should also be noted that there was a language barrier which slowed down the communication process. But this is something that has to be expected when communicating across country borders.

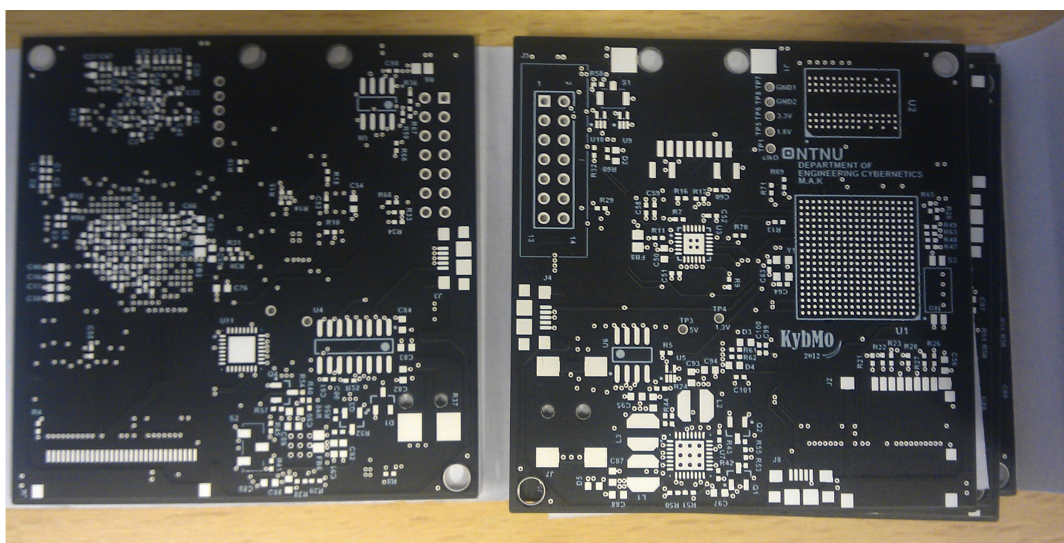


Figure 7.8: The KybMo PCBs that were delivered.

Chapter 8

Assembly

Figure 8.1: The acid-room(syrerom) at The Department of Engineering Cybernetics, where the reflow-oven is located.

8.1 Moisture Sensitivity

The first thing to take into consideration is the issue of trapped moisture inside the components. The problem that can occur when moisture is trapped inside a component, is that during the "reflow cycle" of the assembly stage, where components are exposed to temperatures over 200 C for soldering, the entrapped moisture will become pressurized steam that can cause small cracks and deformations in the integrated circuits. This is known as

"popcorn cracking".

Figure 8.2: NTNU, baking oven at the Electronics and Prototype (ELPRO) Lab

In order to remove trapped moisture, special ovens are used to "bake" the components at a controlled temperature (typically 60 C, but can vary in accordance to device specifications). The ELPRO lab at the IET (Institutt for Elektronikk og Telekommunikasjon) department is in possesion of such an oven. However, it was discovered that some safety requirements in the operating environment has yet to be satisfied for legal operation of the oven, thus rendering it unusable in the time-frame of the project.

But there was no reason for panic. All moisture sensitive components are classified under a specific level of sensitivity, referred to as the Moisture Sensitivity Level (MSL) of the component. The MSL specifies the amount of time a component, starting from when it was brought out of the moisture barrier confinement it was shipped in, can be exposed to the outside world, before it has to be baked.

Level	Floor life (30 C/ 60% humidity)
MSL 1	unlimited
MSL 2	1 year
MSL 2a	4 weeks
MSL 3	168 hours
MSL 4	72 hours
MSL 5	48 hours
MSL 5a	24 hours
MSL 6	Time indicated on the label of packaging

Table 8.1: Moisture Sensitivity Levels, defined in IPC/JEDEC J-STD-20

Most of the moisture sensitive components in the KybMo are level 3, and come with humidity indicators that show whether or not the components should be baked on reception (in case the moisture barrier bags leak or get punctured). No baking was necessary for the KybMo components.

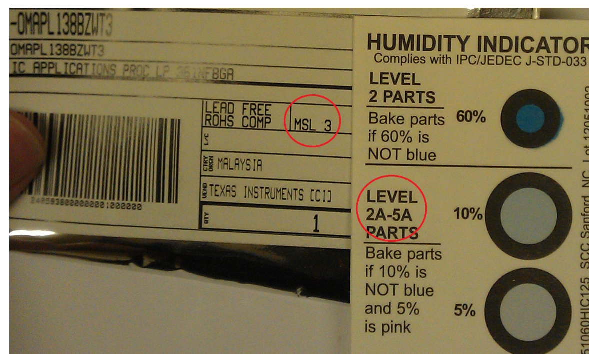


Figure 8.3: KybMo OMAP-L138 IC, humidity indicator card

8.2 Solderpaste Stencil

Advanced high density PCB designs can have several hundred connection pads. Placing solder paste on each individual pad can be cost ineffective. A widely adopted method in the industrial world, targeted at solving this challenge is solder paste stencil printing. This section will provide information on how this method is used in the prototyping industry, followed by how it was adopted for the KybMo.

8.2.1 Stencil Printing

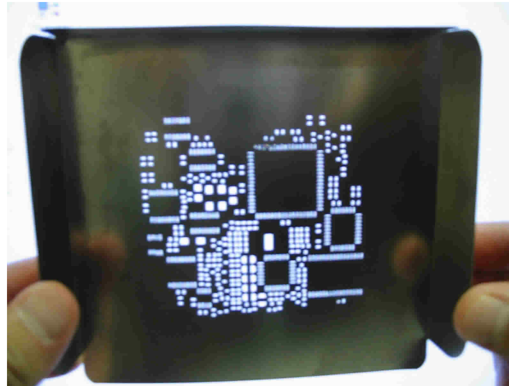


Figure 8.4: Stainless steel PCB stencil, www.pentalogix.com

Solder paste screen printing basically consists of creating a stencil with openings at the location of the connection pads. The stencil is then attached to a stencil printer. The PCB is placed under the stencil and aligned properly, after which a squeegee is used to wipe solder paste across the stencil. This provides a fast and clean way of distributing solder paste to all of the connection pads in one stroke.

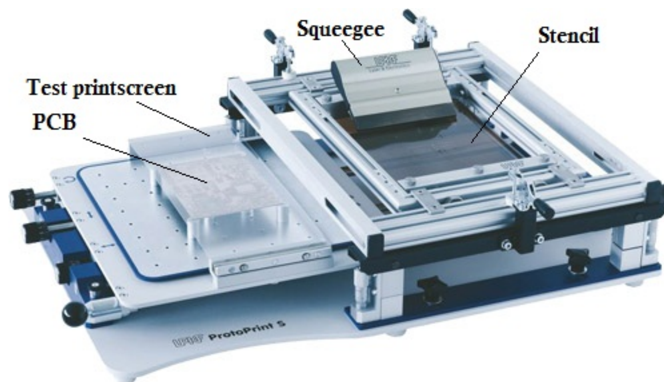


Figure 8.5: LPKF ProtoPrint S, stencil printer

Different stencil types are common. The most solid and reliable stencils are stainless steel, laser cut stencils. They're also the most expensive. The cheapest alternative is a plastic polymer stencil. Stencils can come in frames or without frames, all depending on what the stencil printer requires to secure the stencil. The most common professional choice is to use framed stencils due to the extra support a frame provides.

Unfortunately, neither of the ITK and IET departments were in possesion of a stencil printer, so a home-brewed solution had to be made. More about this in subsection 8.2.6.

8.2.2 The "stencil" layer

In order to create the stencil, a digital file describing the stencil design needs to be provided. In Altium Designer, the stencil design can be found on the "Paste" layers,

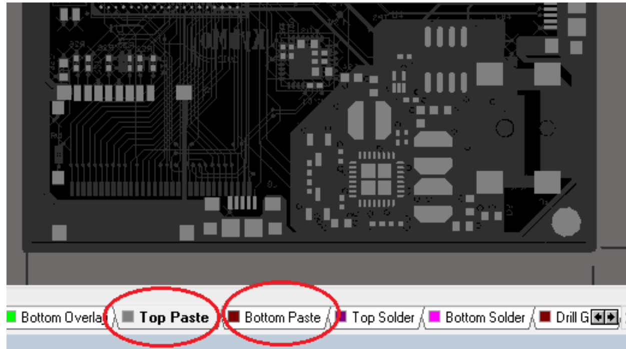


Figure 8.6: KybMo, Paste Layers in Altium Designer

A pad in altium is automatically created with so called "paste mask" information. The paste mask of a pad resides in a paste layer. The paste mask properties of a pad can be edited by double clicking on the pad and navigating to the appropriate paste layer.

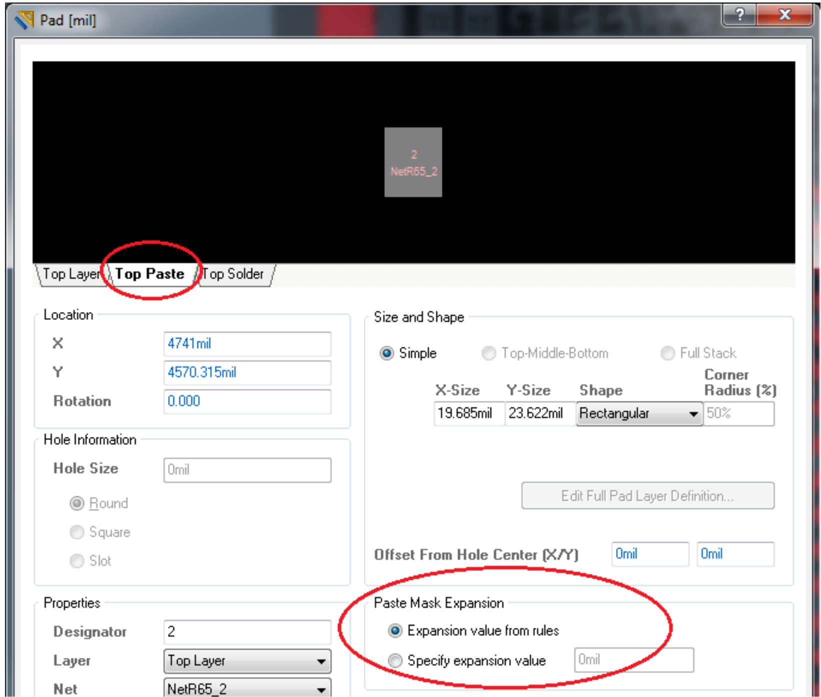


Figure 8.7: KybMo, Paste Layers in Altium Designer

For high-density designs, it could be wise to configure the paste mask to be slightly smaller than the pad size. This will help prevent solder bridges from occurring between the pads. However for the KybMo, the size of the paste masks was kept equal to the size of the pads, as it was unsure how precise the printing process would be. Therefore, avoiding the risk of winding up with insufficient amount of solder paste on the pads was prioritized.

These layers can be exported to Gerber format like the rest of the design in order to be sent to the stencil fabricator. Several options were evaluated in terms of the type of stencil to use for the KybMo.

8.2.3 Special Paste mask considerations for QFN packages

QFN packages usually come with a thermal pad in the center of the component.

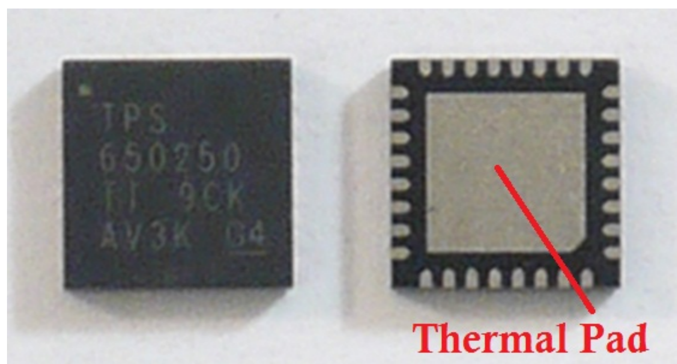


Figure 8.8: TI TPS650250 IC used on the KybMo

Altium Designer automatically creates a paste mask that fills the entire area of the thermal pad. This is not desirable, because the nature of this pad being larger than the rest of the signal pin pads can create a situation where the solder paste deposit in the center will be taller than the depostis on the small signal pin pads. This can cause the device to high-center without getting the opportunity to achieve contact with the signal pads. [ScreamingCircuits.12-11-06, 2006]

Figure 8.9: High-centering due to too much solder on the thermal pad

The solution proposed by the literature[1] is to partition the paste mask. It is also important to ensure that 70-80% of the thermal pad gets covered[SMSC.AN1815, 2010].

[1] [design, 2002], [ScreamingCircuits.12-11-06, 2006], [CirrusLogic.AN315REV1, 2007]

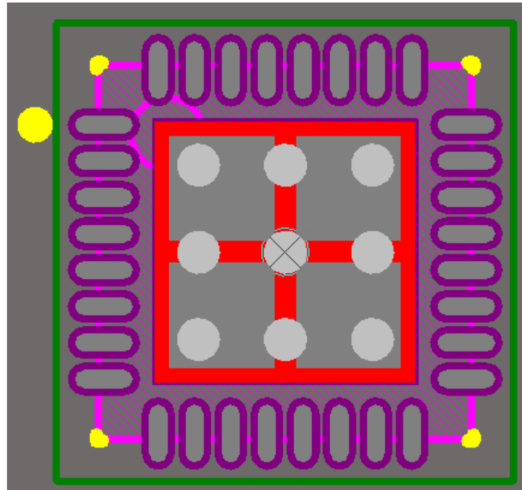


Figure 8.10: KybMo, TPS650250 footprint, partitioned thermal pad (the four gray squares)

Notice that there are vias on the thermal pads (the circles). This is referred to as a via array, and is used to connect the thermal pad to signal ground for grounding and heat dissipation. There is no strict rule for how many vias there should be under a thermal pad, but an even distribution across the pad should be used unless a specific number is specified by the manufacturer. It is important to remember that these vias should be **tented** (filled with soldermask), otherwise solderpaste might escape through the vias and on to the other side of the board. Tenting can be done in Altium Designer by creating holes in the solder mask layer (called Top or Bottom Solder) of the footprint where the vias are located (Figure 8.11), or by telling the PCB fabricator which vias one would like to be tent. However, it is wise to do both (edit in Atlium, and inform the PCB fabricator) in order to avoid problems.

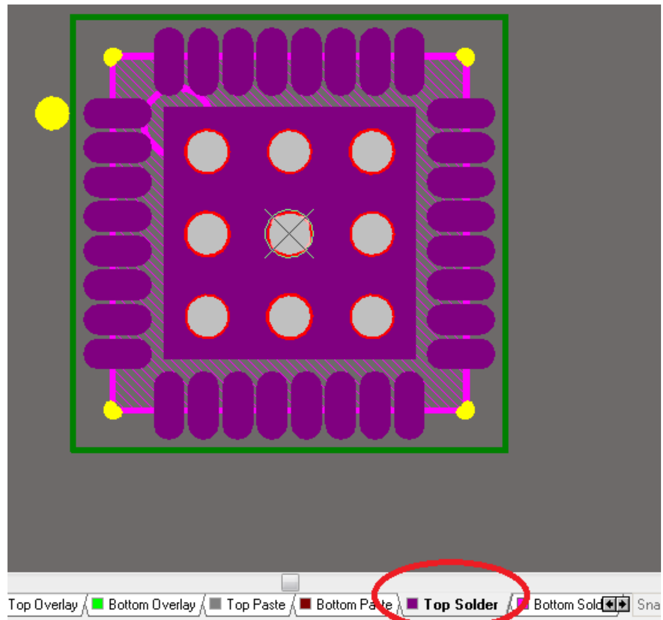


Figure 8.11: TPS650250 via tenting in Altium Designer by creating holes over the vias in the Top Solder layer (purple)

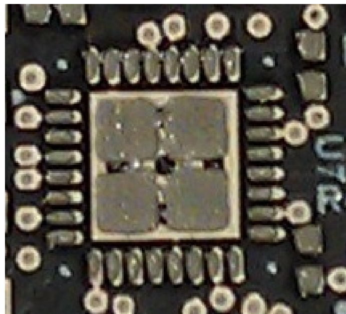


Figure 8.12: KybMo, TPS650250 pads after solder paste printing. The black spots on the thermal pad shows how the vias are tented with solder mask

8.2.4 Stencil thickness

The stencil thickness decides how tall the solder paste walls will be on the pads. Too thick a stencil can result in too much solder paste on the pads, which can cause bridging in high density designs. Unfortunately, stencil thickness is not an exact science, and the research done yielded no direct answer. An interesting point to note about this was an article I found

on ipcoutlook where two experienced consultants from a companny called ITM Consulting discusses the problem.

> "*...From the qualitative side, I know if I use this stencil too thin, I won't get enough paste and I won't have good fillets. Likewise, on a small fine pitch part, if I use a stencil that's too thick, I will get too much paste...It comes back to what do you need in terms of the volume of solder paste. And to be quite frank, is there a source that can supply that answer?...I've talked to a number of people in the industry and when we finally get around to hemming and hawing and talking about it, nobody can provide a direct answer..." - Jim Hall, ITM Consulting.*[2]

Therefore, experiences of designers had to be relied upon, instead of academic reports, in terms of what works and what doesn't. Several forum threads were examined and a 5 mil stencil thickness (0.13 mm) seemed to work well for fine pitch BGA designs and for designs with 0402 (the size of the KybMo's smallest resistors and capacitors, 1x0.5 mm) components.

8.2.5 Ordering the Stencil

Several options were considered in terms of stencil type, such as stainless steel, chemically treated/laser cut, or polyester stencils. Framed or unframed. The prices were:

Dimensions cm	Thickness mm	Technique	Framed	Price
26 x 25	0.13	Laser cut	Yes	170 USD
16 x 16	013	Laser cut	No	118 USD
16 x 16	013	Chemical etch	No	100 USD

Table 8.2: Stainlees steel stencil prices, www.pcbcart.com

These prices were regarded as being expensive for the project, especially since there was no stencil printer at NTNU, which means that successful results weren't guaranteed.

An attempt was made at using the milling machine at the ITK workshop to create a steel stencil.

[2] http://www.ipcoutlook.org/mart/50456D.shtml

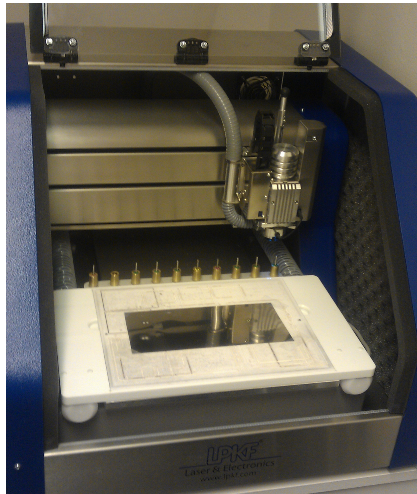


Figure 8.13: Creating a steel stencil with the LPKF milling machine at the ITK workshop

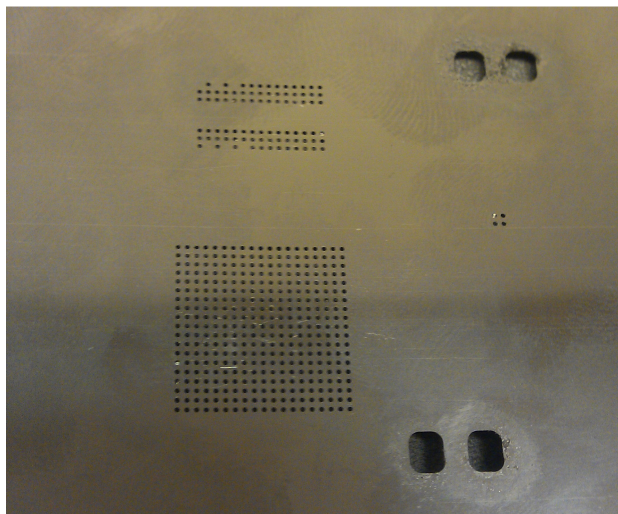


Figure 8.14: The attempt failed

The attempt failed as the software of the milling machine wasn't interpreting the stencil files correctly. The cut out seen on Figure 8.14 is the only part of the design the software was able to interpret. Videos, of the attempts, titled "Metal Stencil" 1 and 2, can be found on the CD.

The only alternative left was to order a polyester stencil. The price was 16.5 GBP from www.smtstencil.co.uk. The most suitable thickness they had to offer was 0.1 mm.

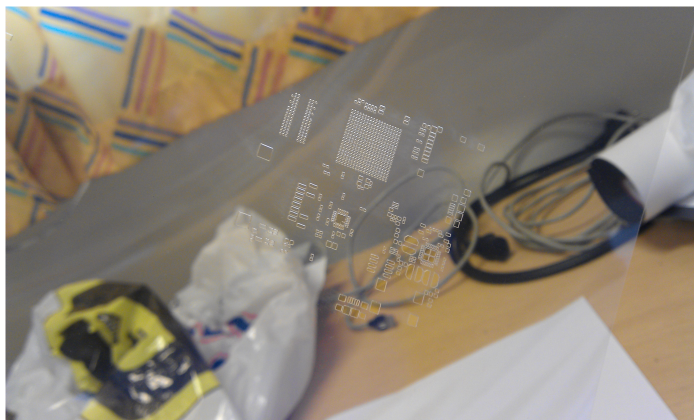


Figure 8.15: KybMo, Polymex Polyethylene Terephthalate stencil

8.2.6 Home-brewed Stencil Printer

The next step was to make something that could be used as a stencil printer. A base for holding the card in a fixed position was created at the ITK Mechanical Workshop, and a cardboard box was cut out to be used as the main body.

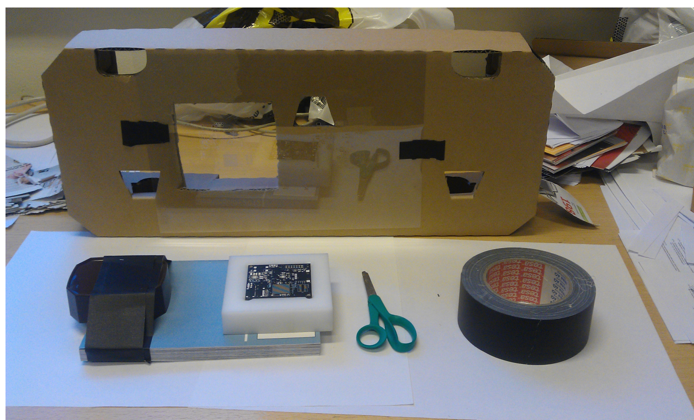


Figure 8.16: Attempt at creating a cardboard stencil printer frame

This setup didn't work, as the cardbox proved to have an uneven surface, and wasn't rigid enough to keep the stencil completely still.

Therefore, a special frame was created for the base instead, which gave a better foundation for holding the stencil in a fixed position.

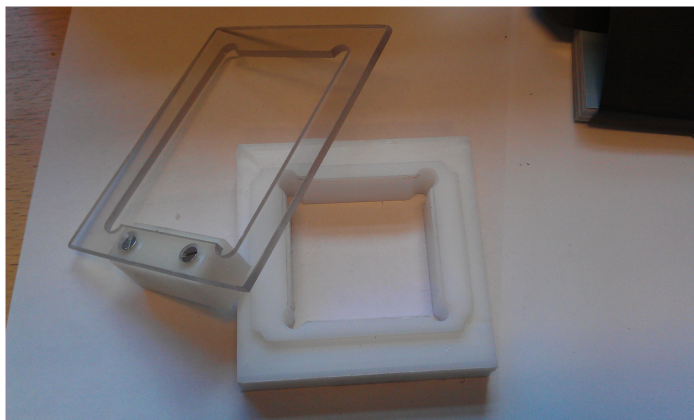


Figure 8.17: The final stencil printer design

8.2.7 Alignment of the PCB and the Stencil

The alignment was an excruciating process. The main challenge is that such high density designs require down to 0.1 0.2 mm precision, or else the solder won't be printed properly on the tiny BGA 0.45 0.55 mm pads. Fixing misplaced solder is not a trivial task with over 440 BGA pads in the design.

The problem with polyester stencils, is that they're not made of a rigid material that keeps them 100% flat, they bend easily from small forces in their surroundings, which creates an un-even surface that makes it extremely hard to operate with the required precision level.

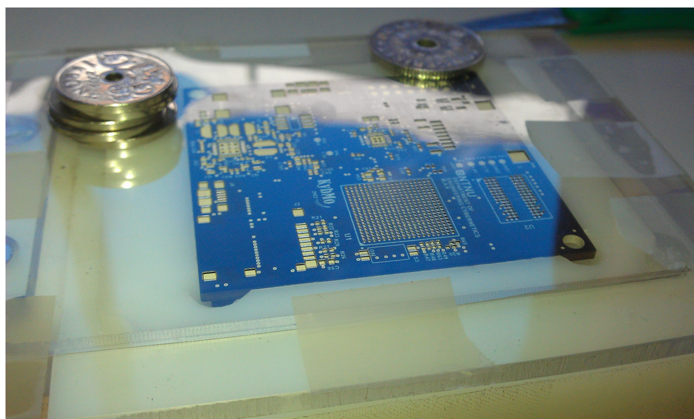


Figure 8.18: Alignment of PCB and Stencil

8.2.8 Printing

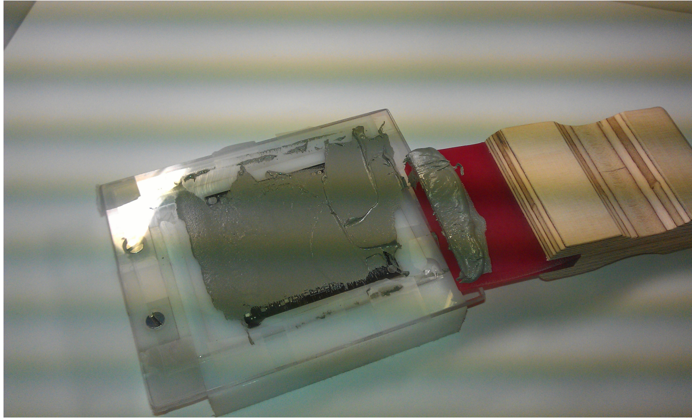


Figure 8.19: Solder paste printing by use of rubber squeegee

Although the printing process sounds like something that should be easy, the elasticity of the polyester stencil presented challenges in this part of the process as well. The movement of the squeegee and the friction of the solder paste, made the stencil follow the squeegee as it was lifted up after swiping. Upon loosing contact with the squeegee the stencil would fall back again, creating a mess of the printed solder paste (note that this is a matter of very small movements in the vertical direction).

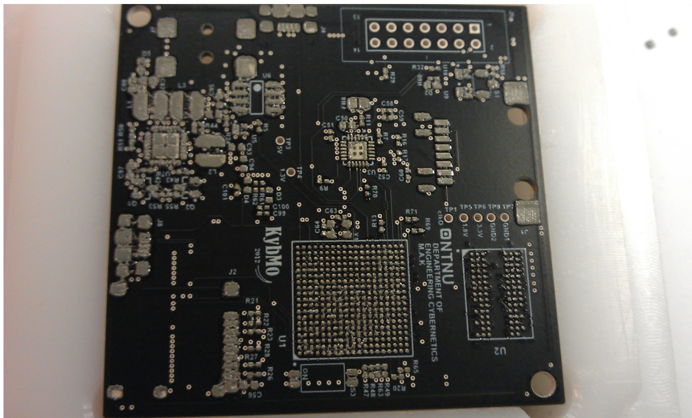


Figure 8.20: First attempt, messy results

It took a couple of attempts before the right swiping technique was mastered and satisfactory results were achieved. The stencil had to be cleaned with alcohol each time on both sides, which also meant that it had to be re-aligned each time as well. The most important things to

remember when swiping is that the speed needs to be constant from start to finish, and being fast and determined in the motion of swiping is better than being slow. Upon removing the squeegee from the stencil at the end, caution should be taken in order to avoid the stencil being pulled up with the squeegee (swiping the squeegee all the way out of the board's surface area is better than lifting it up vertically at the end).

8.3 Component Placement

Satisfactory printing results were finally achieved, and the placement of components could finally start.

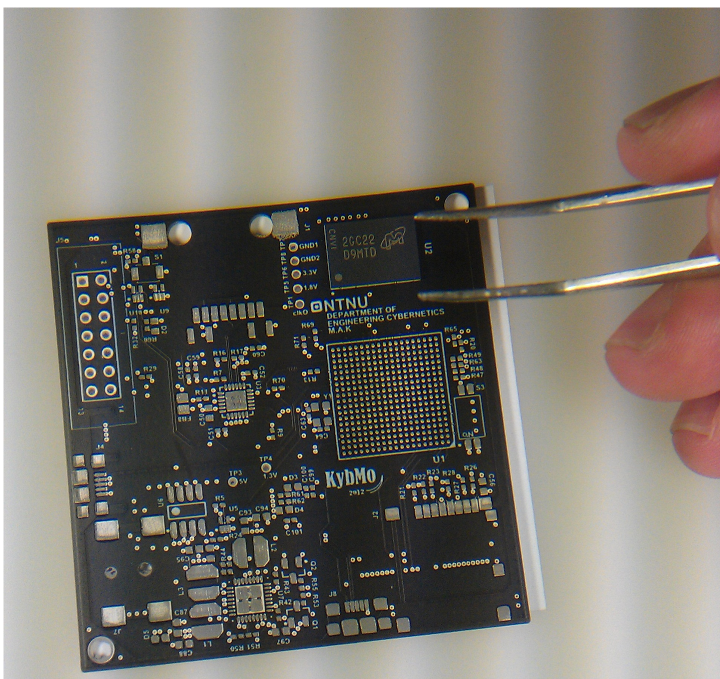


Figure 8.21: Placement of components

One of the most frequent questions that was asked in regards to this part of the process, was about the placement of the BGA chips. This is usually done by automated machinery in the industry in order to avoid misplacement. On the KybMo, the BGA placement was done by hand while using a magnifying glass. Having the chip borders outlined in the solder mask was the main lifeline for determining proper placement.

This was also the case for the rest of the components. Industrially, it is common to use "pick and place" machines. The most basic version of such machinery consist of a hand held

pen-like suction device that enables the operator to pick up components by placing it on their surface, and dropping them vertically by turning the suction off when they're properly aligned over their pads. More advanced and automated version also exist where a robotic arm takes care of the placement after being programmed properly.

The IET department has purchased such an automated pick and place machine, and it might become available for students sometime during the fall semester of 2012.

For ease of placement, a component placement guide was created and used under the assembly process. This guide can be found on the CD.

Figure 8.22: The workstation in the acid-room at the ITK Department

8.4 Reflow Soldering

After finishing the placement of components on the top side, the board was inserted into the LPKF ProtoFlow E oven for reflow soldering. The oven uses heat convection under controlled temperatures to melt the solder paste.

Figure 8.23: LPKF ProtoFlow E, ITK Department NTNU

8.4.1 Reflow Profile

Before starting the reflow process, a reflow profile has to be determined. The reflow profile consist of four stages, also referred to as "zones":

- Pre-heat: A temperature zone used to pre-heat the assembly.
- Thermal soak: A temperature zone for removal of solder paste volatiles and activation of the fluxes within the solder paste.
- Reflow: The temperature zone when the solder paste turns into liquid state tin. This is where the peak temperature is reached.
- Cool down

Each zone is defined by a temperature level, and a time duration. The correct parameters can be found by studying the properties of the solder paste that is used, and by checking the maximum temperature that the components in the assembly can withstand, and for how long they can withstand it.

Studying the datasheets, and using the Beagleboard's reflow profile as a reference[TI.SPRAAV2, 2008], a reflow profile for the KybMo was worked out:

- **Pre-Heat:** 150 C, 100 s
- **Reflow:** 245 C, 10 s
- **Cool down:** 80 s

Notice that there is no "thermal soak" parameter. This is because there'll be a natural soak zone in the period starting from when the temperature rises from 150 C and up to the reflow zone. Thus, the ProtoFlow E oven only defines these three zones for smaller boards. Larger boards may have a separate thermal soak zone to even out the temperature of the components across the board.

The maximum temperature specified in the reflow profile above, 245 C for a period of 10 seconds, is specified as the maximum tolerance level of the Ethernet RJ45 connector. Thus, the reflow process can't exceed these limits even if there are other components in the assembly who can tolerate up to 260 C for 20 seconds.

8.4.2 Reflow Process

One thing that is important to understand is that the most common reflow ovens actually contain several chambers, i.e several physical zones that the assembly passes through. This results in precise temperature control, where the board is passed from one chamber with a specific temperature to another chamber with another temperature as quick as possible (the ramp-up rate defined in the data-sheet of the component defines the speed for how fast this can be). However, the ProtoFlow E is a **single-chamber** reflow oven that simulates several zones.

This means that the transition from one "zone" to another is not as controlled. The "Reflow" zone starts immediately after the "Pre-Heat" zone, even if the temperature is still 150C. In order to stay at 245C for 10 seconds, several timing parameters had to be tested while the oven was running with an empty chamber. After a couple of runs, the following parameters for the ProtoFlow E were found to conform with the reflow profile listed in subsection 8.4.1:

- **Profile name used in the oven:** LF small (Lead Free, small board)
- **Pre-heat temp:** 155 C (Because the chamber loses heat upon opening the drawer to inset the assembly)
- **Pre-heat time:** 120 s (same reason as above)
- **Reflow temp:** 245 C
- **Reflow time:** 118 s (because it takes about 108s to reach 245C)
- **Reflow power:** 100%
- **Cool down:** 80 s

Editing these parameters on the oven is not a difficult task, and the method of how to do it is described in the ProtoFlow E user manual.

A video of the reflow process can be found on the CD.

Problems under reflow

The reflow process successfully finished. However, there was one issue that could've been destructive. The chamber of the ProtoFlow E at NTNU seems to be mis-aligned with the drawer. This causes the drawer to grind against the side of the chamber, creating more friction the further in the drawer goes, until reaching a point where it slams into its closed state right before the end.

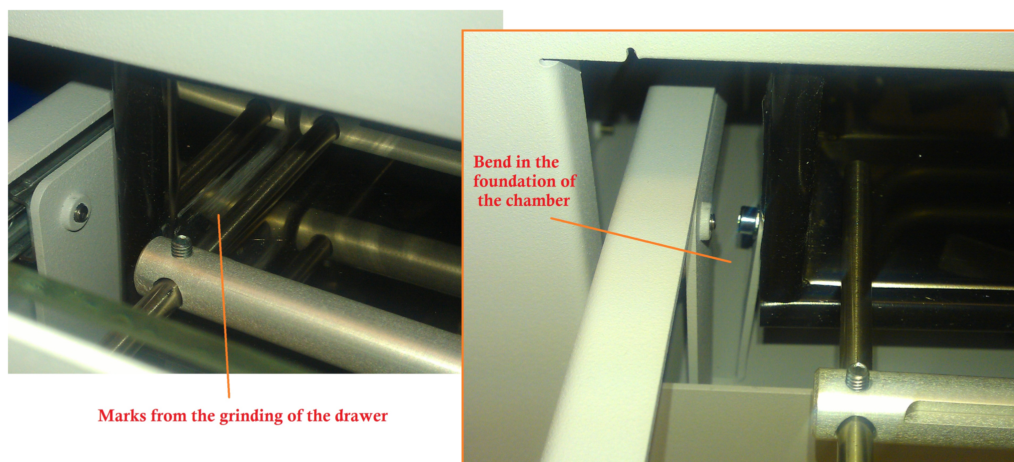


Figure 8.24: Misaligned chamber causes drawer to grind against chamber walls.

Fortunately, the only thing that was affected by this in the assembly was the USB connector used for the USB to UART interface.

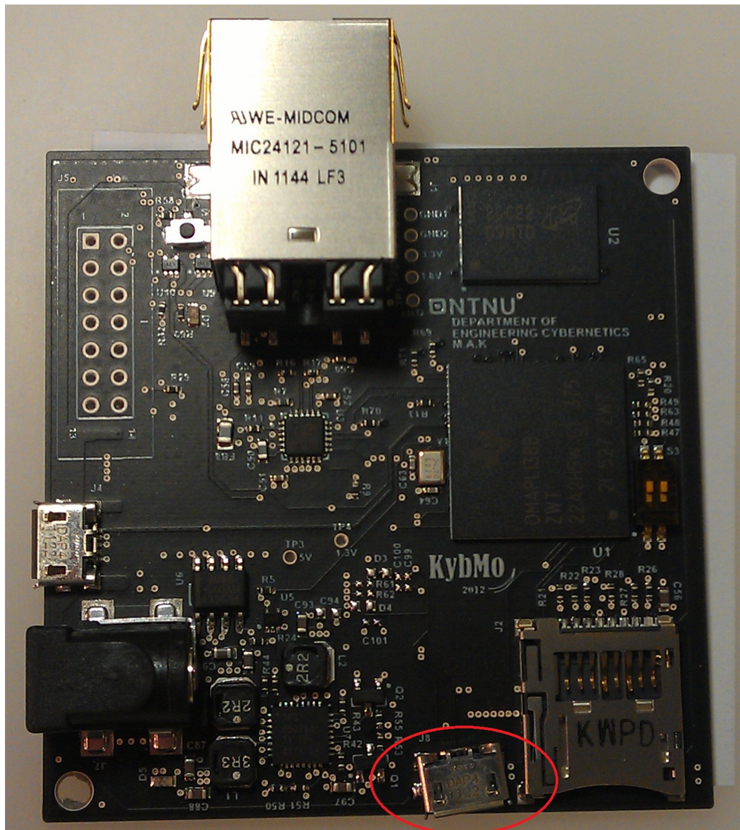


Figure 8.25: KybMo after the reflow process.

8.5 Hand Soldering

The bottom side of the board had to be soldered by hand, as placing the board in the reflow oven for re-flow soldering would cause the components on the top side of the board to fall down during re-flow.

workstations at both the ITK workshop and the IET ELPRO lab were used during this part of the project.

Figure 8.26: Workstation with camera, ITK Workshop

The ITK workshop has a camera that can magnify up to 30x while displaying the image on a computer screen to the right. The IET ELPRO lab on the other hand has a microscope that can magnify up to 30x. The microscope workstation at the ELPRO lab was found to be better suited such fine pitch soldering, because staring into a microscope with your entire body aligned in one direction gives better stability than soldering with your body aligned in one direction, while your head is turned in a different direction. Furthermore, the ELPRO lab uses induction based soldering irons with bended tips, which enables a higher level of precision than the soldering irons at the ITK workshop.

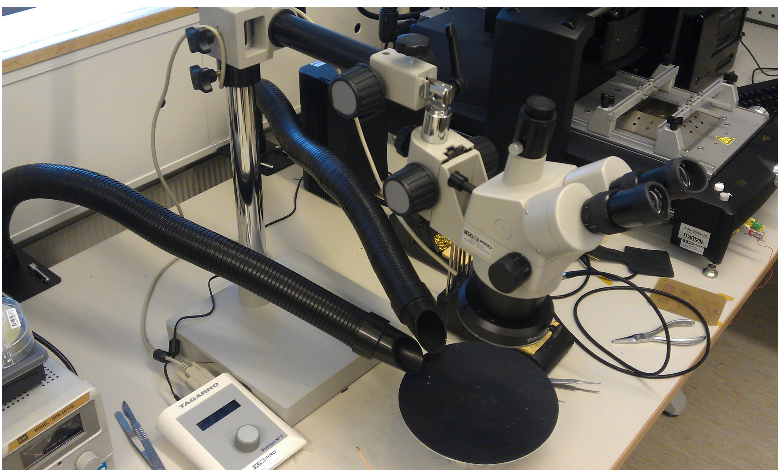


Figure 8.27: Microscope, IET ELPRO Lab

Figure 8.28: Induction based soldering iron

The smallest components in the assembly was capacitors and resistors in the 0402 package (1x0.5 mm). It was challenging to hand solder such small components while keeping a steady hands and ensuring clean results.

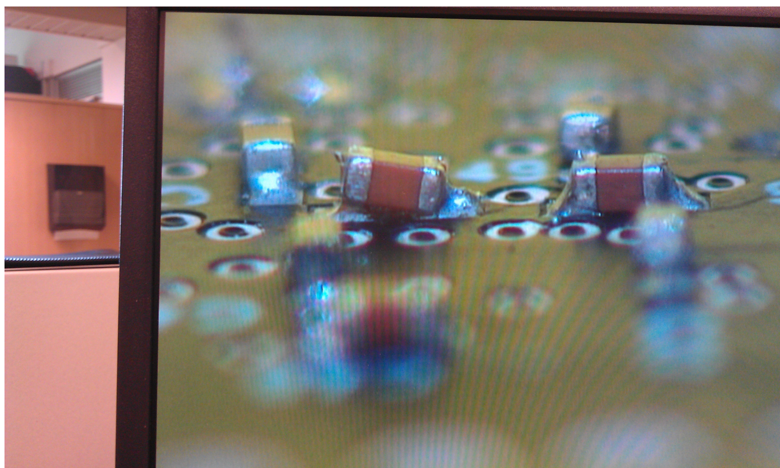


Figure 8.29: Hand soldered 0402 capacitors

Situations such as the one shown in Figure 8.29, where the component is not soldered in a flat position, can easily occur. This happens when the components slightly slide up on the tweezer due to magnetic forces. Short circuits can easily occur when working with such fine pitches, and it's not always easy to tell where the short is. A lesson that was learned the hard way during the course of the project was to test for short circuits between power and ground planes after each component was soldered, otherwise, if done at the end, locating the source of the short can be an extremely difficult task.

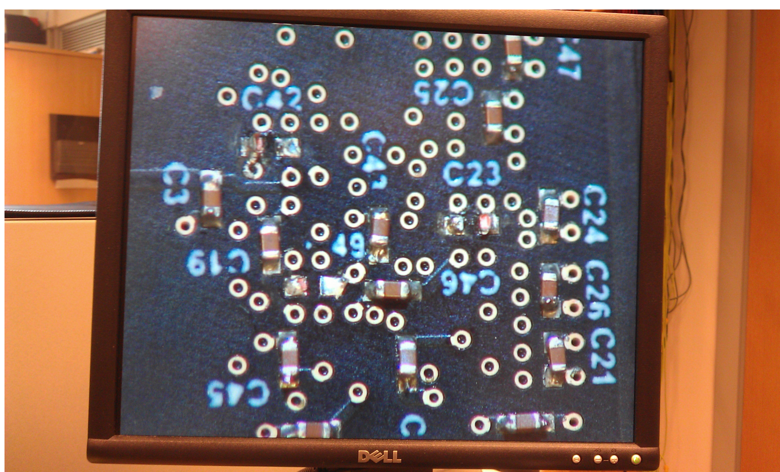


Figure 8.30: Fixing short circuits

8.5.1 Using a heat gun and a heating plate

Soldering discrete components one by one takes a lot of time, so it was desirable to test a method that could solder more than one component at a time. The method tested here was aimed at soldering the decoupling capacitors under the OMAP-L138 chip by using a heating plate set to 150 C on one side, and a heat gun directed at the capacitors on the other side. Lead based[3] solder paste was placed on the pads, one by one, by using a solder paste dispenser, followed by placing the capacitors on their respective pads.

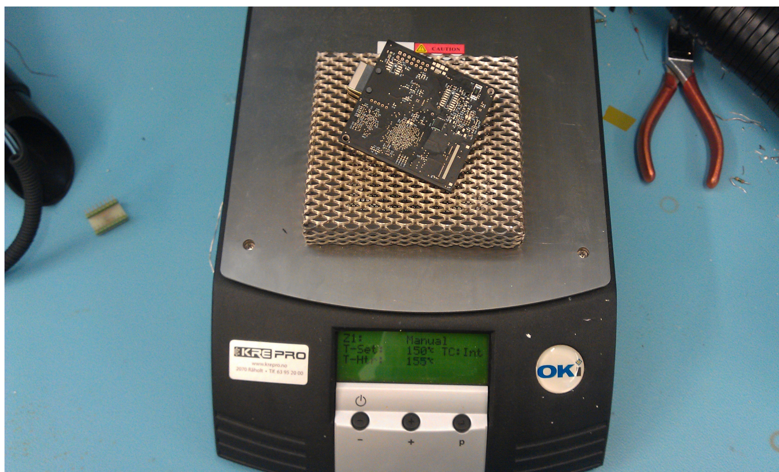


Figure 8.31: heating plate

The board was then placed on a heating plate, and a heat gun configured with low air force was directed at the capacitors. Low air force is important in order to avoid blowing the capacitors away.

[3] Pb-free soler paste was not used in this case, because it was desirable to be able to reach the re-flow point of the solder paste at a lower temperature. Lead based solder paste is legal for prototyping purposes, but not for mass production.

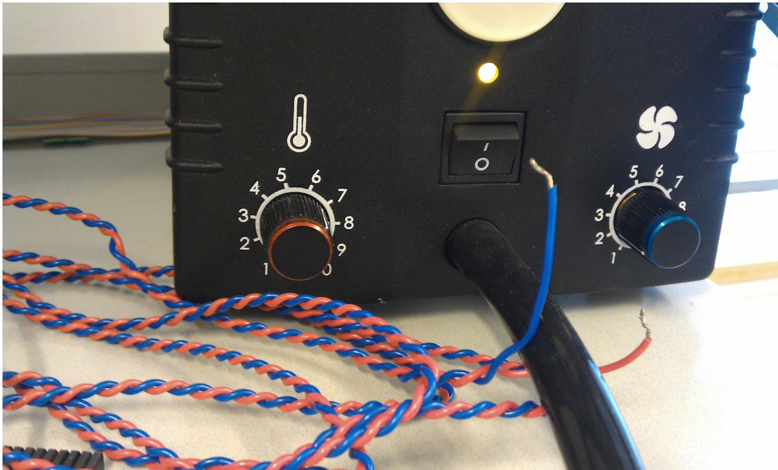


Figure 8.32: Heat gun settings

This produced cleaner results compared to the capacitors soldered by hand. One component got misplaced by the air force, but individually soldering it afterwards was not a problem. A video of the process can be found on the CD under the "Assembly" folder.

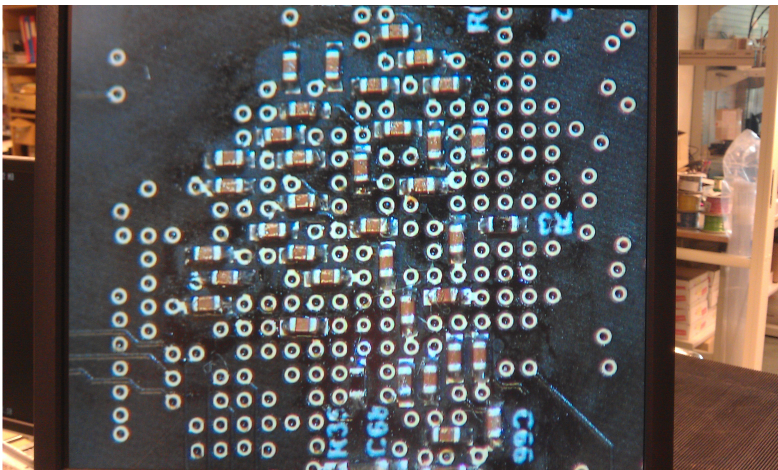


Figure 8.33: Results

Fixing the USB connector that got misplaced during the reflow process was achieved with same technique, but this time more air force could be used as the rest of the assembly was already secured to the board.

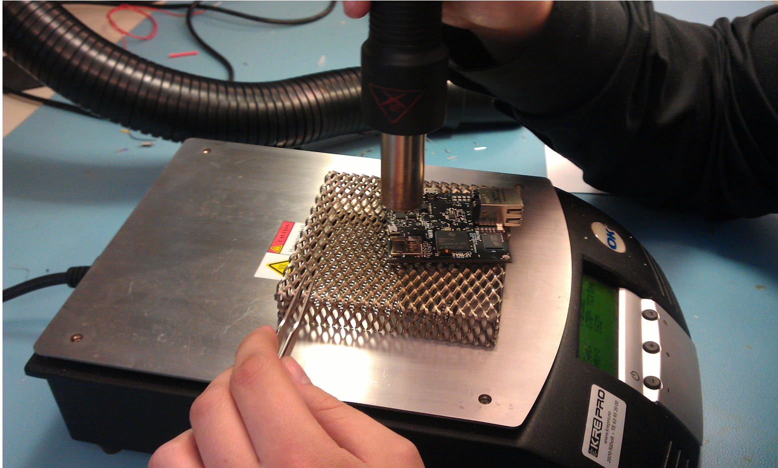


Figure 8.34: Fixing the USB to UART connector

8.5.2 Advanced Package Rework (APR) machine

The FT232 USB to UART IC is located on the bottom layer of the PCB. The IC comes in a QFN package and it was desirable to test the capabilities of the Metcal APR-5000 machine in possession of the ELPRO lab. The APR machine is usually used for the assembly or rework of BGA and QFN packages, and offers precise temperature control with the possibility of setting up accurate reflow profiles.

Two videos of the rework process that was used, titled "Advanced Package Rework", can be found on the CD.

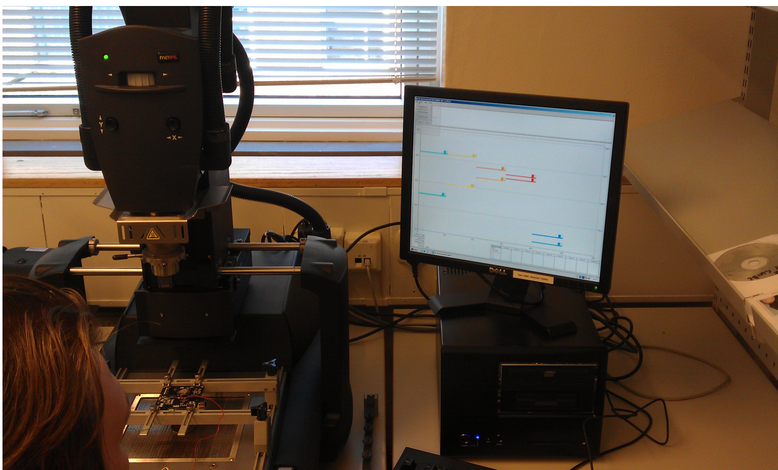


Figure 8.35: IET ELPRO lab, Metcal APR-5000

Several attempts were made with the APR machine but all failed. The process required more heat from the heating plate located under the board. The fact that there were components on the other side of the board restricted the temperature level of the APR's heating plate. The heat dissipated quickly on the board and the desired temperature levels were not reached.

The chip had to be soldered by hand, but an important lesson was learned: Most of the ICs should be placed on one side of the board, especially QFN and BGA packages. Having packages that are difficult to solder by hand on both sides of the board can critically complicate the assembly process.

Note: It was mentioned earlier that the design could not be placed in the oven a second time for reflowing of the opposite side, due to the fact that the components facing the ground would fall off when their solder connections melt. However, an idea could be to first use the oven with Pb-free solder paste on one side, which has the highest reflow temperature. And on the other side of the board, one could use Pb-based solder paste, which has a lower reflow temperature than Pb-free, and set the reflow profile of the oven to peak right below the reflow temperature of the Pb-free solder paste. The concept would be similar to what was displayed in subsection 8.5.1, with the combination of a heating plate, heat gun, and Pb-based solder paste.

Chapter 9

Board Validation, Debugging, and Setup

9.1 Introduction

Although software development falls out of the scope of the project, it is still a priority to see how much of the board's functionality can be validated. Challenges related to boot loader theory and microprocessor software development were tackled. Lessons learned during this stage of the project should be taken into consideration in the initial design phase of future projects.

9.2 Validation Overview

Validating the KybMo is a tricky and time consuming task. Even though software development wasn't a part of the project, the process of validating the board requires software to be written, thus straining the time budget to the limit. The software used in this stage was all based on code originally written for the OMAP-L138 Evaluation Module board from Logic-PD. While a set of existing code might seem like something that'll make the validation process a walk in the park, the fact that it was tailor made for a system with a completely different configuration than the KybMo complicated things more than expected.

The following list was compiled of functionality that should be validated given the limited amount of time:

- Booting - To ensure that there is some form of "life" in the SoC.
- UART - Communication through a PC terminal.

- JTAG - Important for debugging software and reading register values.
- MicroSD - To ensure support for non-volatile storage[1].
- USB 1.1/2.0 - Ensures that we have USB peripheral support.
- Ethernet - Ensures that we have network support.
- DDR2 Memory

Everything was successfully validated, although there were some minor issues with the USB peripherals. A demo video demonstrating the full validated functionality of the KybMo can be found on the CD. The following sections describe the validation process in more detail.

Note: Usually, for designs such as the KybMo, the natural thing to start with is a JTAG Boundary scan, to ensure that all the pins in the design, and especially under the BGAs, have proper connectivity. However, the University does not own any boundary scan software. An attempt at obtaining free trial software was made, but to no avail. The companies would not write me back.

9.3 Booting

The OMAP-L138 supports an array of different boot modes, three of which are accessible on the KybMo through the boot-switch configuration:

- Boot from UART
- Boot from JTAG
- Boot from microSD

All three boot modes were successfully validated. [Coombs, 2011], titled "Using the OMAP-L132/138 Bootloader" describes how the bootloader image located in the SoC's Read Only Memory (ROM) works. Gaining a basic understanding of that functionality is important in the debugging process. The details behind the inner workings of the ROM bootloader will only be discussed in the following sections when it's relevant.

The boot process involves compiling an application, then loading it with TI's AISgen tool. This tool lets the user set the initial configuration of the board, such as clock speeds, DDR registers and peripheral register configuration. A binary file is then exported, and it is this file that the OMAP's internal boot loader can interpret.

[1] Non-volatile storage refers to storage technology that can store data without the need of power to be present.

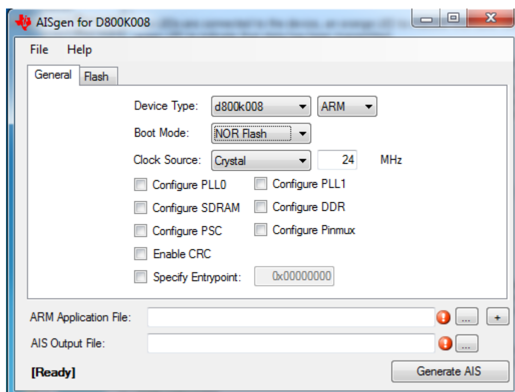


Figure 9.1: Texas Instrument's AISgen tool

9.4 Validating UART

Booting from UART is done by setting boot switch 2 towards the ON mark (towards the micro SD card reader), and boot switch 1 in the opposite direction.

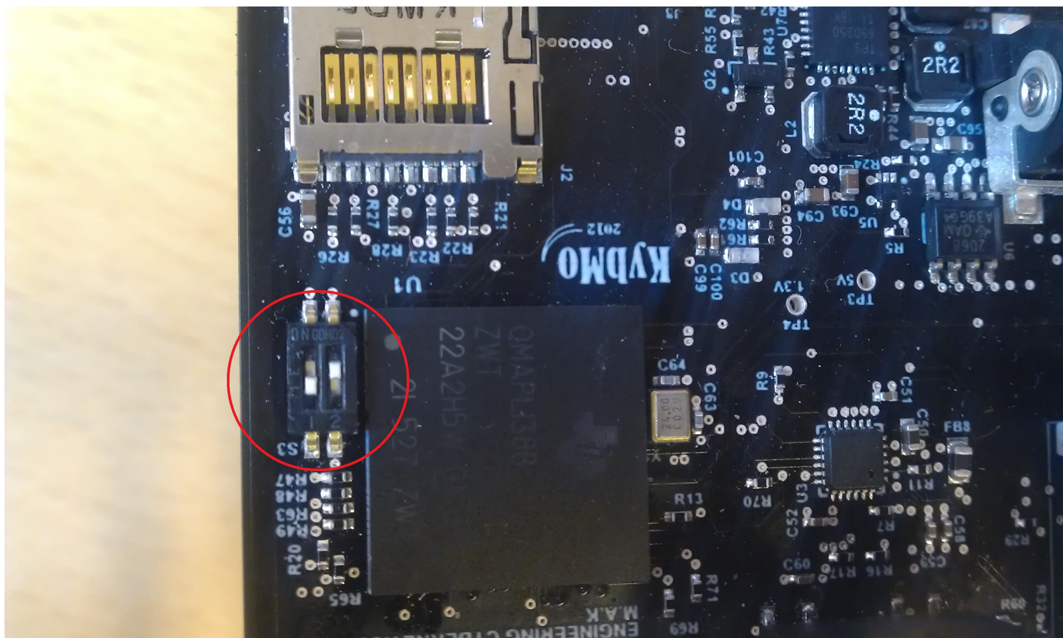


Figure 9.2: UART Boot switch settings

Booting the board from UART validates that both the microprocessor and the UART interface is functional. After setting the boot switches for UART boot mode and powering on the board, TI's UART Boot Host utility is used to load the binary file generated by AISgen into the internal RAM of the SoC.

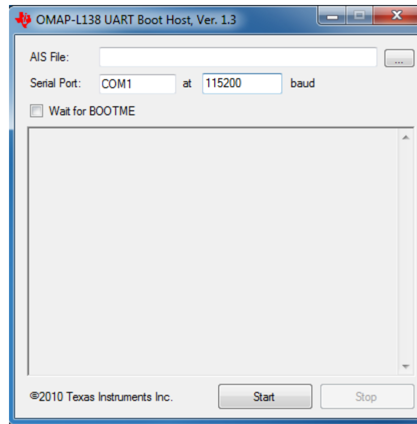


Figure 9.3: UART Boot Host Utility

The UART interface is made accessible by the PC through the FT232 UART to USB device. Drivers for this device[2] have to be installed on the PC in order to set up a virtual COM port and enable communication through a terminal.

Two LEDs are connected to the device, an orange LED to indicate that data has been received, and a green LED to indicate that data has been transmitted.

To boot the board through UART and test if the ARM core could run an application, a HelloWorld UART application was compiled from the OMAPL138 StarterWare package and sent in to the AISgen tool, and then to the UART Boot Host. Initially, the board would boot, but no serial communication would be possible afterwards due to the application utilizing the UART2 port instead of the UART0 port which the KybMo uses for serial communication. The Code was hacked to support UART0 and the following result was observed:

[2] http://www.ftdichip.com/Drivers/VCP.htm

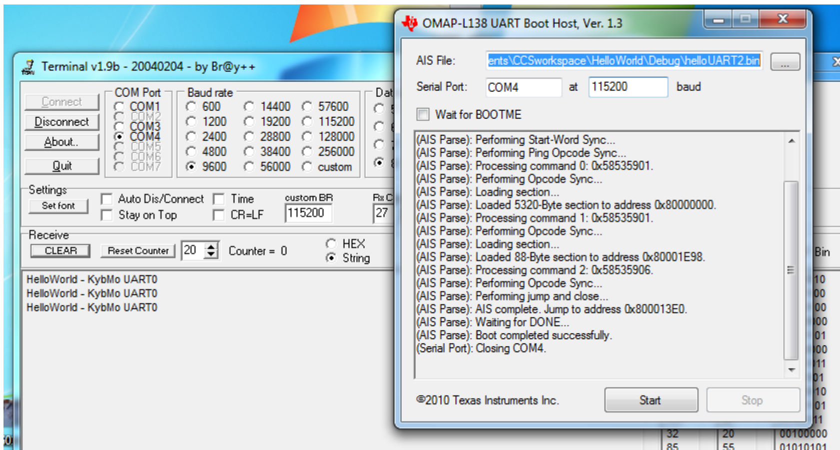


Figure 9.4: Hello World UART application

A video of the UART boot process can be found on the CD.

9.5 Validating JTAG

The JTAG emulator device used here is the XDS100v2 from Spectrum Digital, specifically designed for TI systems with a 14 pin TI connector. The default Integrated Development Environment(IDE) for TI microprocessors is Code Composer Studio (CCS). The only way to obtain a free license for CCS is to own a XDS100v2 emulator.

In order for the JTAG interface to be able to load code onto the SoC, the boot switch pins need to be set to JTAG Boot Mode: Both switches should be swicthed away from the micro SD connector.

Figure 9.5: JTAG Boot Switch Settings

Unlike other boot modes, JTAG booting does not include the use of the AISgen tool for initial configuration. Instead, it utilizes a .GEL file (General Extension Language). GEL syntax is similar to C, so it's easy to read for C developers. The modified Gel file can be found on the CD under the validation folder. A link is provided to download the original Gel file in subsection 9.12.5.

JTAG validation was executed by loading an application from the OMAPL138 StarterWare package, connecting to the ARM Core and stepping through code while monitoring register values and variables.

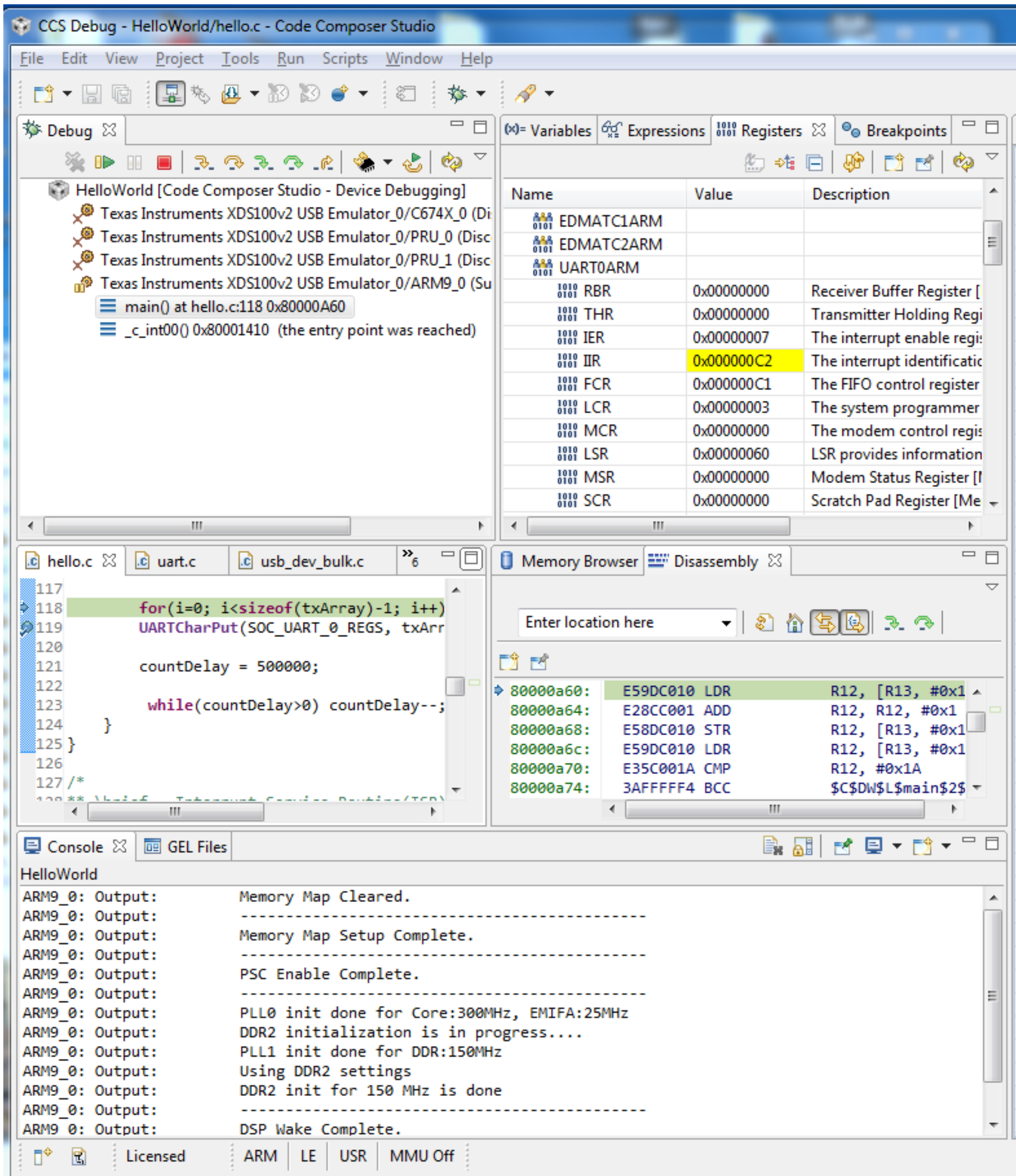


Figure 9.6: JTAG Code Stepping

A video of the JTAG validation process can be found on the CD.

9.6 Validating MicroSD, Booting U-Boot & Linux

The board can be configured for MMC/SD boot by setting boot switch 1 towards the direction of the SD card reader, and setting switch 2 in the opposite direction.

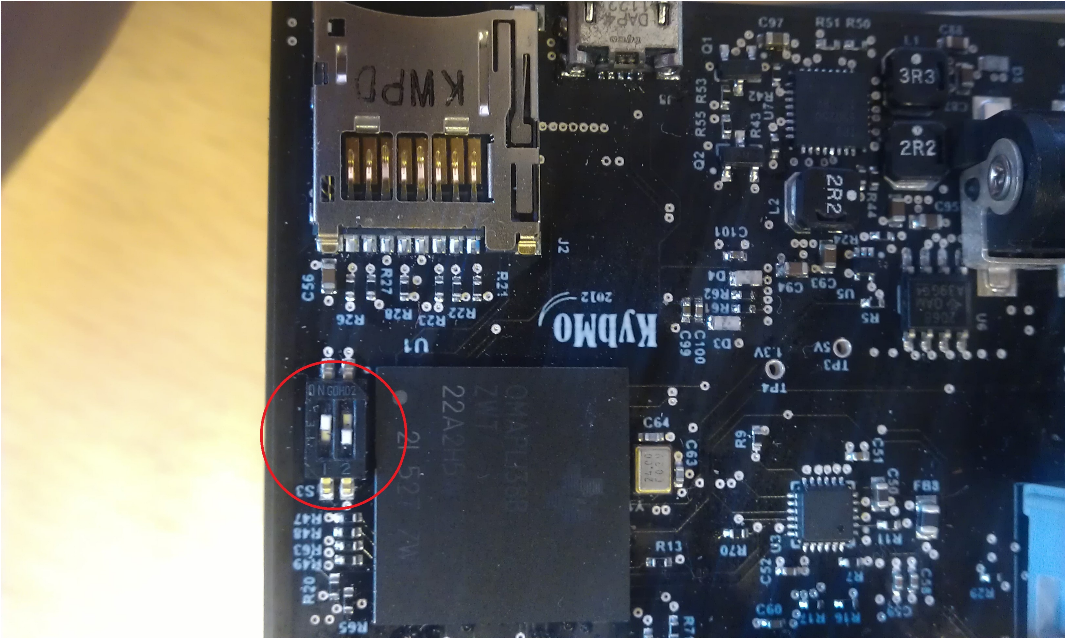


Figure 9.7: MMC/SD Boot Switch Settings

In the MMC/SD boot mode, more precisely referred to as MMC/SD0 boot mode, the internal ROM boot loader starts at address 0 within the first region of the SD card(which should be left unpartitioned) and searches for a so called magic word (0x41504954) to detect the AIS image generated by the AISgen tool. Booting fails if the magic word isn't found within the first 2MB of the memory card.

Hence it is important to start the first partition with an offset in order to leave unpartitioned space at the beginning of the SD card for the AIS image to reside in. The AIS image is in this case synonymous with what is classically referred to as the Master Boot Record (MBR) of a storage device. The internal ROM boot loader executes a series of commands found within the AIS binary image. Basically, the function of these commands is to load an application program contained within the rest of the AIS image into the RAM. Control is then handed over to this application program, which for example can be an Operating System boot loader, such as U-Boot or LILO (Linux-Loader).

Validating the MMC/SD0 boot mode had to be carried out in two steps:

- **Creating a functional application in order to verify success:** Creating an application and testing it through a validated boot-mode, such as UART. The Application chosen was U-Boot.
- **Proper configuration of the SD card:** Partitioning the SD card properly and flashing the application in accordance to the procedure discussed in the previous paragraph.

The following sections describe the process of bringing up U-Boot, preparing the SD card, and loading Linux.

9.6.1 U-Boot

Although booting any kind of simple application would be enough to validate the MMC/SD0 boot mode, getting a system up and running with console capabilities would be of great help in further validation of the remaining peripherals. U-Boot is a console based Linux boot loader used with a number of similar platforms (BeagleBoard, BeagleBone, HawkBoard, LCDK), and it was therefore the application of choice to use for validating the MMC/SD0 Boot mode.

This part of the validation was an extensive process of trial and error. The first priority was to ensure that a functional version of u-boot for KybMo was available. The Linux SDK for OMAP-L138 EVM includes U-Boot source code written for the EVM. An overview of the relevant files used and their respective paths is given in section 9.12

Porting U-Boot for the KybMo requires the following:

- Configuring the DDR2 registers properly in AISgen.
- Enabling UART0 support instead of UART2 for serial communication.
- Enabling MMC/SD0 configuration instead of flash (The EVM loads U-Boot from flash memory).
- Switching off the MII interface and using the RMII interface for Ethernet instead. (The Ethernet PHY on the KybMo uses RMII. Further more, MII blocks UART0 due to pin-multiplexing)

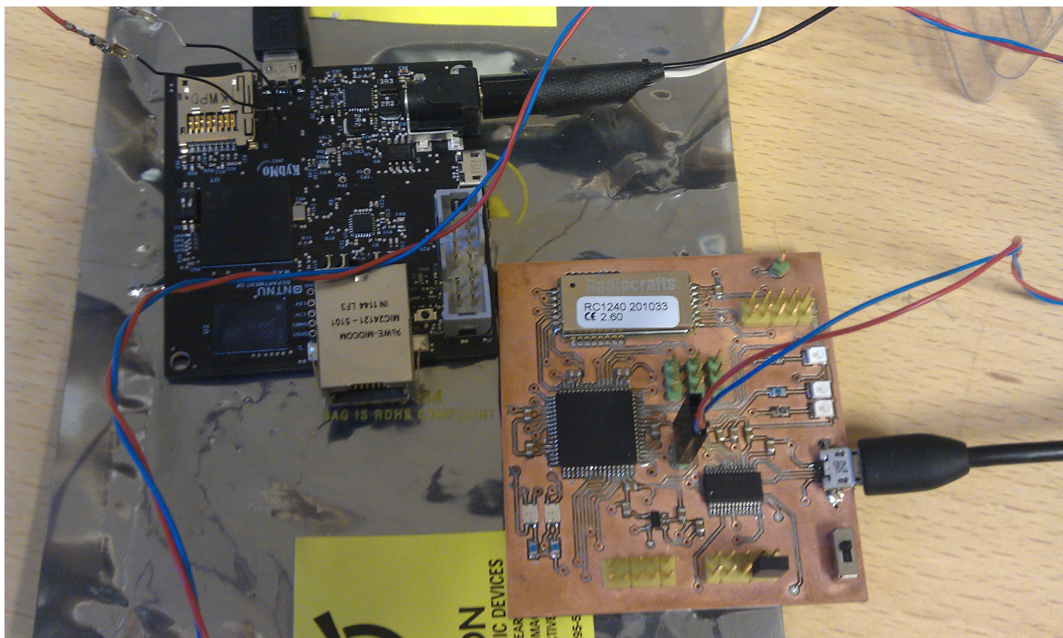


Figure 9.8: U-Boot Test Environment, external FT232 board borrowed from co-student, Karsten Rennæs

Figure 9.8 shows the setup of the test environment. In order to eliminate as many sources of errors as possible, wires were hooked up to the UART2 traces and interfaced to an external board with a separate FT232 chip. Before attempting to port it with UART0 functionality, the unaltered UART2 console functionality had to be verified. The board was set in UART boot mode, and U-Boot was loaded through the UART Boot Host utility. But before any of that could be done, U-Boot had to be passed into the AISgen utility where the DDR2 registers had to be programmed according to the timing specifications of the memory device's datasheet[Micron, 2006].

The spreadsheet[3] in Figure 9.13 was used to convert the timing requirements and calculate the register values.

[3]Download from: `http://processors.wiki.ti.com/index.php/Programming_mDDR/DDR2_EMIF_on_OMAP-L1x/C674x`

OMAP-L1x/C674x/AM1x mDDR/DDR2 Memory Controller Register Setting Calculator

	Parameter	Description	Data Manual Value	Register Field Name	Field Value (Dec)	Field Value (Hex)	Register	Register Value
	fDDR/mDDR_CLK (MHz)	Memory clock frequency	150					
	tck (ns)	Memory clock period	6,67					
	MemType	Memory type: mDDR vs DDR2	DDR2					
SDCR Configuration							**SDCR**	**0x0013C633**
	Data Bus Size	Valid values are 16 or 32	16	DDR2TERM1	0	0		
	CAS Latency	Valid values are 2-5	3	IBANK_POS	0	0		
	Number of Banks	Valid values are 1, 2, 4, and 8	8	MSDRAMEN	0	0		
	Page Size	Valid values are 256, 512, 1024, and 2048	2048	DDRDRIVE1	0	0		
				BOOTUNLOCK	0	0		
				DDR2DDQS	0	0		
				DDR2TERM0	0	0		
				DDR2EN	1	1		
				DDRDLL_DIS	0	0		
				DDRDRIVE0	0	0		
				DDREN	1	1		
				SDRAMEN	1	1		
				TIMING_UNLOCK	1	1		
				NM	1	1		
				CL	3	3		
				IBANK	3	3		
				PAGESIZE	3	3		
SDRCR Configuration							**SDRCR**	**0x00000492**
	tREFI (us)	Average Periodic Refresh Interval	7,8	LPMODEN	0	0		
				MCLKSTOP_EN	0	0		
				SR_PD	0	0		
				RR	1170	492		
SDTIMR1 Configuration							**SDTIMR1**	**0x3A923249**
	tRFC (ns)	Refresh cycle time	195	T_RFC	29	1D		
	tRP (ns)	Precharge command to refresh or activate command	18	T_RP	2	2		
	tRCD (ns)	Activate command to read/write command	18	T_RCD	2	2		
	tWR (ns)	Write recovery time	15	T_WR	2	2		
	tRAS (ns)	Active to precharge command	42	T_RAS	6	6		
	tRC (ns)	Activate to Activate command in the same bank	66	T_RC	9	9		
	tRRD (ns)	Activate to Activate command in a different bank	12	T_RRD	1	1		
	tWTR (ns)	Write to read command delay	13	T_WTR	1	1		
SDTIMR2 Configuration							**SDTIMR2**	**0x3E1EC722**
	tRAS(MAX) (us)	Active to precharge command	70	T_RASMAX	7	7		
	tXP (tCK cycles)	Exit power down to a non-read command	4	T_XP	3	3		
	tXSNR (ns)	Exit self refresh to a non-read command (for mDDR use tXSR)	205	T_ODT	0	0		
	tXSRD (tCK cycles)	Exit self refresh to a read command (for mDDR use tXSR)	200	T_XSNR	30	1E		
	tRTP (ns)	Read to precharge command delay (only required for DDR2, for mDDR set this to 1)	13,333	T_XSRD	199	C7		
	tCKE (tCK cycles)	CKE minimum pulse width	3	T_RTP	1	1		
				T_CKE	2	2		
DRPYC1R Configuration						0	**DRPHYC1I**	**0x00000084**
				EXT_STRBEN	1	1		
				RL	4	4		
				PWRDNEN	0	0		

Figure 9.9: Memory Register Spread Sheet

Even though the register values were calculated specifically according to the data obtained from the datasheet, they didn't seem to work. After some searching, the values used for the HawkBoard, which uses the 128MB version of the same chip, were found and tried. These settings worked for the KybMo, and the U-Boot console showed up in the terminal window.

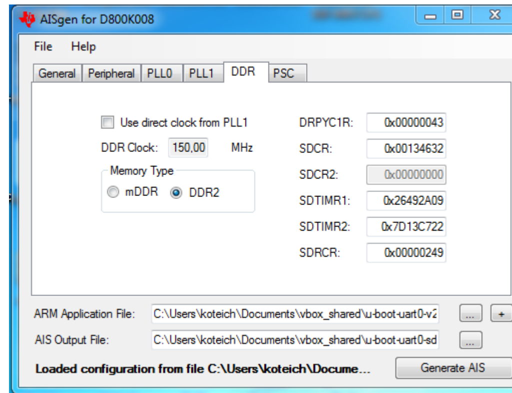


Figure 9.10: Kybmo DDR2 Register configuration with HawkBoard values.

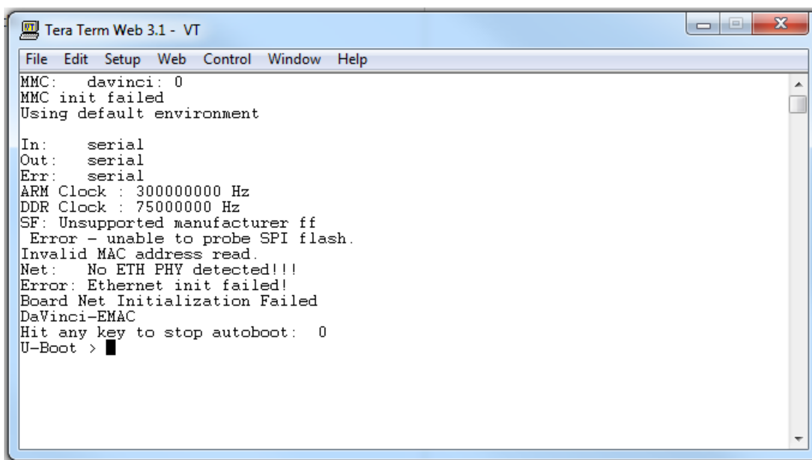


Figure 9.11: U-Boot Console

9.6.2 Configuring the microSD card

The next step was to configure an SD card properly and place U-boot on it. The OMAP L138 EVM SDK provides a script that creates an SD boot card, and the instructions found on a TI wiki page[4] were followed. The Wiki's instructions are to use the script found in the SDK, followed by using the linux command: *dd if=$aisfile of=$sdcard seek=10* in order to overwrite the MBR of the card. "seek =10" is intended to avoid overwriting the boot table. This method did not work at all. Upon inspecting the SD boot card script, it was discovered that the script flashes a precompiled U-Boot image to the SD card by using a flashing utility

[4] http://processors.wiki.ti.com/index.php/OMAP-L138_Preparing_SD_Card_for_Boot

that comes with U-Boot called uflash. The script was modified to flash the image generated by the AISgen utility instead. This produced positive results.

9.6.3 Modifying the Linux Kernel

The SD card script also creates two partitions, one with a linux Kernel image on it, and one with a linux file-system. Like U-Boot, the Linux Kernel needs to be modified in order to be ported to the KybMo. The modifications that need to be made are:

- Enable UART0 instead of UART2.
- Enable RMII instead of MII.
- Modify SD logic to load the file system.
- Configure the USB peripheral settings.

A list of the relevant files that were modified is given in section 9.12. Editing and compiling the linux kernel found in the SDK was a challenging process due to several factors. First of all, the relevant files are scattered around a maze of folders, and whenever a method is being used, there is no naming convention that indicates which file the method is implemented in.

A solution to this is to use the find command in the following way to search for a specific string(All files that were modified are tagged with the "KybMo" string:

sudo find -type f | xargs grep -l "KybMo"

This will search for the string KybMo in the current directory and within all the folders, and a list of files containing the string will be returned. Searching the linux directory with the above commands yields:

```
[linux-devkit]:~/ti-dvsdk_omapl138-evm_04_03_00_06/psp/linux-2.6.37-psp03.21.00.04.sdk> sudo find -type f | xargs grep -l "KybMo"
[sudo] password for user:
./drivers/net/davinci_emac.c
./arch/arm/mach-davinci/include/mach/uncompress.h
./arch/arm/mach-davinci/da850.c
./arch/arm/mach-davinci/board-da850-evm.c
./arch/arm/boot/zImage
./arch/arm/boot/compressed/misc.c
./arch/arm/boot/compressed/misc.o
./arch/arm/boot/compressed/vmlinux
./arch/arm/boot/uImage
```

Figure 9.12: Kernel files tagged with the KybMo string.

The string can be replaced with a function name and all files using or implementing the function will be listed. In some cases, the list may be large due to a function being implemented in several files for different systems. Most of the files referring to "davinci", "da850" or "da8xx" are used for the OMAP-L138 SoC.

Another reason that made modifying the Kernel a challenge was the host machine. The host machine being used in the project was a Windows 7 machine running Linux from a virtual machine, this affects the performance of the linux system and makes the compilation time of the kernel slow. It took about 30 minutes to compile the kernel each time. It is therefore recommended to use a dedicated Linux machine for future projects that need to work on modifying the Linux kernel.

The Linux kernel was eventually modified and booted successfully on the KybMo.

```
File  Edit  Setup  Web  Control  Window  Help
x-gnueabi/ti-linuxutils-1_2_26_02_05-r57d/linuxutils_2_26_02_05/packages/ti/sdo/
linuxutils/cmem/src/module/cmemk.c
CMEM Range Overlaps Kernel Physical - allowing overlap
CMEM phys_start (0xc2000000) overlaps kernel (0xc0000000 -> 0xd0000000)
CMEMK Error: Failed to request_mem_region(0xc2000000, 18874368)
FATAL: Error inserting cmemk (/lib/modules/2.6.37/kernel/drivers/dsp/cmemk.ko):
Bad address
DSPLINK Module (1.65.01.05_eng) created on Date: Dec 23 2011 Time: 12:07:46
Calibrating touchscreen (first time only)ts_open: No such file or directory
.
Starting Matrix GUI application.

Arago Project http://arago-project.org arago ttyS0

Arago 2011.06 arago ttyS0

arago login: root
root@arago:~#
```

Figure 9.13: TI's Arago distribution running on KybMo.

9.6.4 Modifying for UART0

In order to get a console running on the intended UART channel, four separate steps needs to be carried out. While modifying the code to work with UART0 instead of UART2 might sound like a trivial task, it took a while to figure out what to change and where the relevant files were located.

The four steps eventually discovered through trial and error are:

- Change the channel from ttys2 to ttys0 in the SD card script.
- Change the register being used in U-Boot from UART2_REGS to UART0_REGS, disable MII, pass ttys0 in the bootargs variable. (These steps ensure that the U-Boot console appears on UART0)
- Do the same for the Linux Kernel as was done in U-Boot. (This makes the kernel print out messages to UART0 while booting)
- After the SD card has been created, navigate to the linux file system partition on the SD card titled ROOTFS, and edit the "inittab" file found in the etc/ directory to use ttyS0 instead of ttyS2. (This brings up the Linux console on UART0)

9.7 USB

Configuring the Kernel for USB support was done in accordance to the information found on the wiki page listed in subsection 9.12.5, titled "Building 03.22 PSP release Components for OMAP-L138". Unfortunately, the results were not as expected for USB 1.1:

9.7.1 USB 1.1 OHCI:

The USB 1 port on the OMAP-L138 complies with the Open Host Controller Interface specification, referred to as USB 1.1, which basically means that the board can act as a host without any specific driver set-up on the device being connected.

The following output was produced when a usb flash drive was connected to the USB 1.1 port on the KybMo:

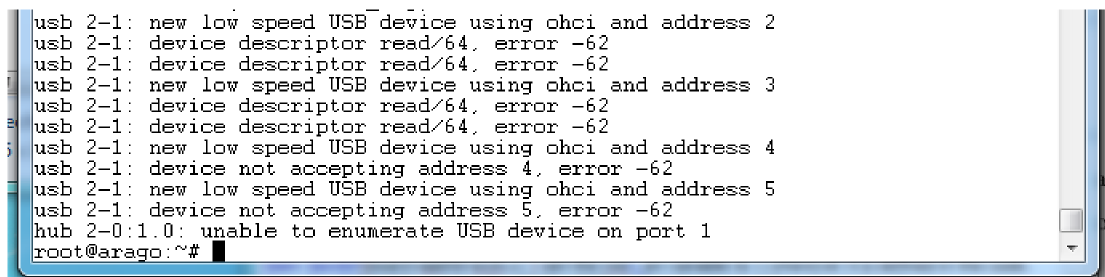


Figure 9.14: Linux console output upon detecting a USB flash drive on USB 1.1 port.

This means that the USB 1.1 port is recognizing the device. Thus, the error is most likely a configuration issue in the Kernel, as there is no other interface between the USB connector and the OMAP-L138, the USB PHY is integrated into the SoC.

9.7.2 USB 2.0:

Connecting the USB flash drive to the USB 2.0 port produced no output at first. Upon closer inspection, it was discovered that the USB0_DRRVBUS signal which is responsible for turning on 5V sourcing to the USB 2.0 port was inactive. This can be due to either a software fault and wrong port configuration in the Kernel, or that the USB0_DRRVBUS pin (or "ball") is not soldered properly under the BGA package.

Providing 5V to the USB port through a regular wire solved the issue and Linux responded that it had discovered the device.

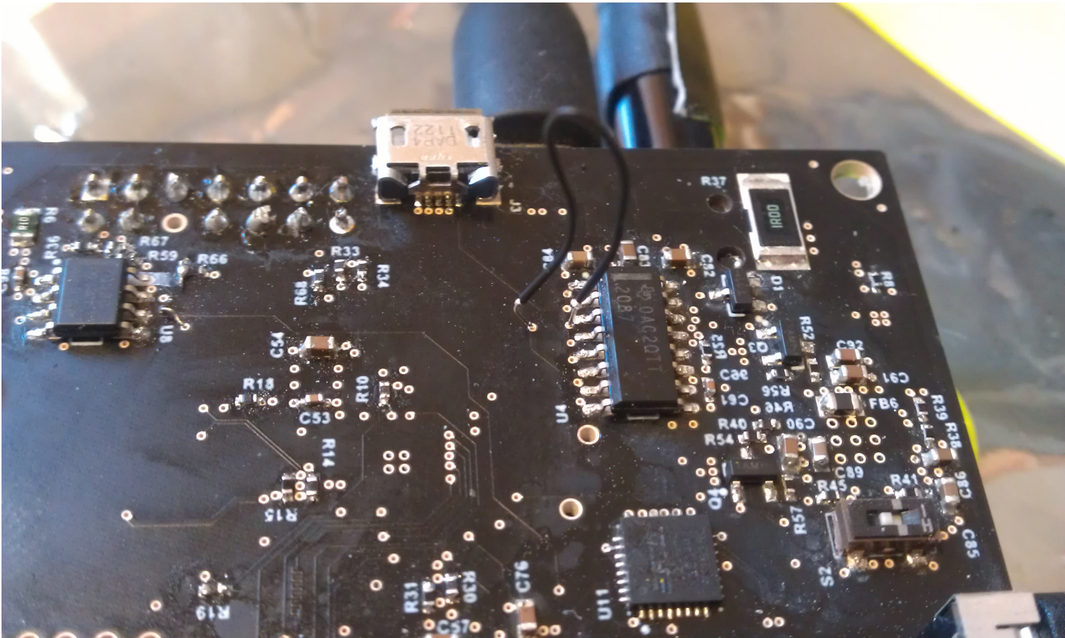


Figure 9.15: Forcing 5V sourcing by wire patch. The point closest to the IC is connected to the 5V plane. The other point is connected to the VBUS signal on the USB 2.0 port on the other side of the board.

File Edit Setup Web Control Window Help

```
root@arago:~# usb 1-1: new high speed USB device using musb-hdrc and address 2
scsi0 : usb-storage 1-1:1.0
scsi 0:0:0:0: Direct-Access                    Flash Drive      8.07 PQ: 0 ANSI: 2
sd 0:0:0:0: [sda] 1978368 512-byte logical blocks: (1.01 GB/966 MiB)
sd 0:0:0:0: Attached scsi generic sg0 type 0
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Assuming drive cache: write through
sd 0:0:0:0: [sda] Assuming drive cache: write through
 sda: sda1
sd 0:0:0:0: [sda] Assuming drive cache: write through
sd 0:0:0:0: [sda] Attached SCSI removable disk
VFS: Can't find a Minix filesystem V1 | V2 | V3 on device sda.
FAT: invalid media value (0x00)
VFS: Can't find a valid FAT filesystem on dev sda.
EXT3-fs (sda): error: can't find ext3 filesystem on dev sda.
EXT2-fs (sda): error: can't find an ext2 filesystem on dev sda.
cramfs: wrong magic
FAT: invalid media value (0x00)
VFS: Can't find a valid FAT filesystem on dev sda.
ISOFS: Unable to identify CD-ROM format.
```

Figure 9.16: Linux console output upon detecting a USB flash drive on USB 2.0 port.

There seems to be some configuration issues for configuring the stick properly, as seen by the output. However, simply pressing the enter key after the initialization and mounting the device by creating a mounting point through: *mkdir -p /mnt/usb*, followed by mount */dev/sda1 /mnt/usb* successfully mounts the device and makes accessing it and viewing the content possible.

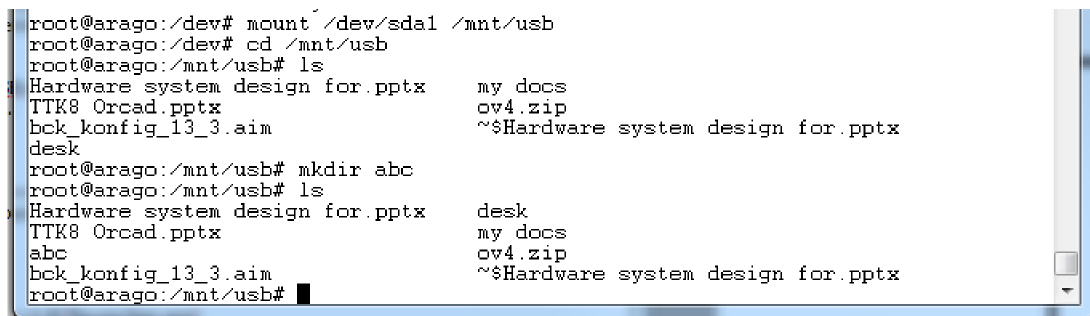


Figure 9.17: Browsing and writing to the USB flash drive connected to the USB 2.0 port.

9.8 Ethernet

The Ethernet PHY on the EVM, which the SDK is customized for, uses the Media Independent Interface(MII) to communicate with the OMAP-L138. The Ethernet PHY on the KybMo uses the Reduced Media Independent Interface (RMII). Therefore, configuring U-Boot and the Linux kernel for the Ethernet interface to work properly entailed disabling the MII interface and enabling the RMII interface instead.

The modifications include:

U-Boot:

- **da850evm.h:** Define RMII macros and undefine MII macros.
- **da850evm.c:** Comment out MII pinmux configuration, replace with RMII pinmux configuration according to the data sheet.
- **da850evm.c:** Comment out RMII selection logic for the EVM, which is directed at EVM specific selection hardware. The KybMo always uses the RMII interface, no selection logic is needed.

Linux Kernel:

- **board-da850-evm.c:** Comment out the contents of the da850_evm_mii_pins[] array.
- **board-da850-evm.c:** set the rmii_en variable to 1 wherever it is defined in the code.

- **da850.c:** Set the 50 MHz RMII clock to be sourced from the SoC in the mux configuration according to the data sheet.
- **davinci_emac.c:** Comment out 100Mbps conditional test, KybMo always uses 100Mbps.
- **davinci_emac.c:** Set the rmii_en variable to 1 wherever it's defined.

The modifications above resulted in successful detection of the PHY:

```
net eth0: attached PHY driver [SMSC LAN8710/LAN8720] (mii_bus:phy_addr=0:00, id=
7c0f1)
ADDRCONF(NETDEV_UP): eth0: link is not ready
net eth0: DaVinci EMAC: ioctl not supported
eth0      no wireless extensions.

udhcpc (v1.13.2) started
Sending discover...
PHY: 0:00 - Link is Up - 100/Full
ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
Sending discover...
Sending select for 129.241.154.99...
Lease of 129.241.154.99 obtained, lease time 691200
adding dns 129.241.0.200
adding dns 129.241.0.201
done.
Setting up IP spoofing protection: rp_filter.
```

Figure 9.18: Kernel discovering the Ethernet PHY.

To test if there was a functional link established to the Internet, a simple ping test was carried out with several websites.

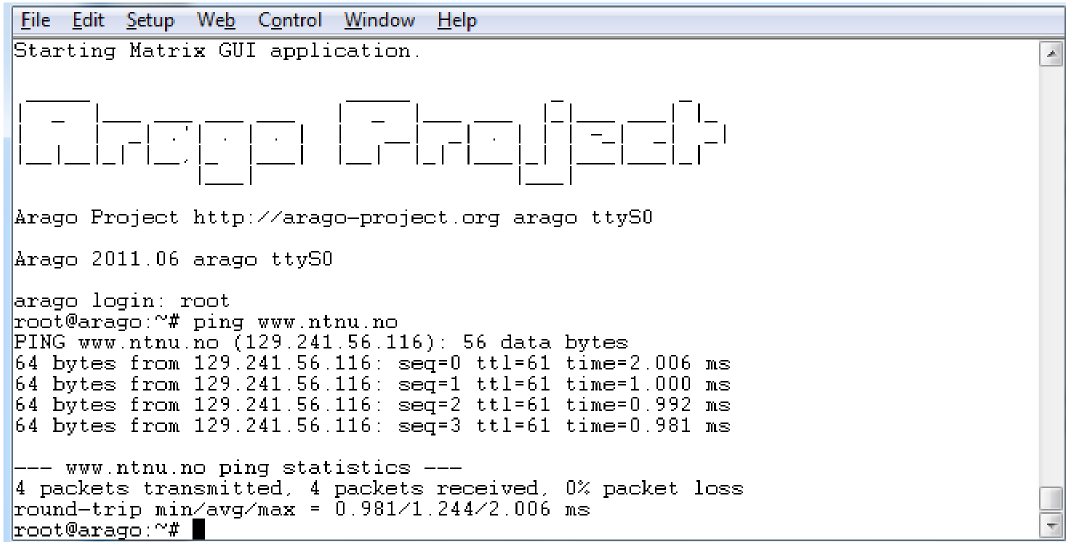


Figure 9.19: Pinging www.ntnu.no

Note: Under the process of configuring the Ethernet PHY, a point was reached where linux were recognizing the PHY without being able to establish a link to the internet. After some debugging, it was discovered that the solder joints of the RJ45 connector were weak/non-existent. This was easily fixed by applying more solder paste and heating up the pins with a soldering iron to establish stronger solder joints.

9.9 Memory

U-Boot Provides a simple test utility for testing the RAM. The command is "mtest" and the parameters it takes is a range of random memory values. (The base address of the DDR interface is 0xC0000000, and mtest can test adresses from 0xC2000000). The test passes if it quickly repeats writing and reading information.

The KybMo passed the test, a simple video of running mtest can be found on the CD.

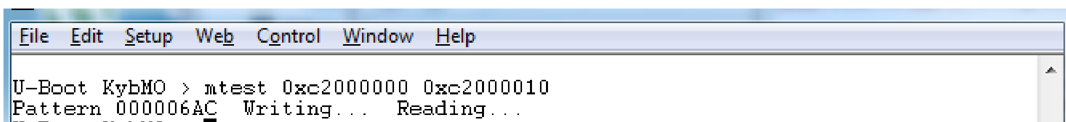


Figure 9.20: mtest command in U-Boot.

The "free" command in linux shows that the operating system recognizes the capacity of the memory chip.

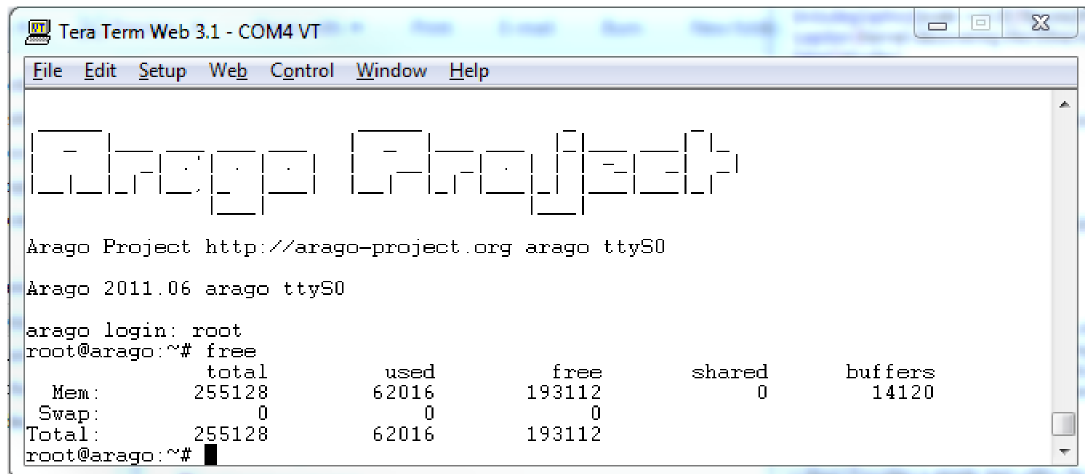


Figure 9.21: "free" command displaying memory info.

The very fact that Linux is booting and running validates that the RAM is functional. The incorrect DDR settings discussed in subsection 9.6.1 prevented the board from booting all together. Indicating that the RAM has to be functional in order to boot in the manner done here.

9.10 Power consumption

Power consumption is application dependent and it's hard to calculate an exact value at this exact moment. The demo video shows a peak of around 380mA while the Ethernet is actively pinging and with the USB stick powered on. That equates to 1.9W (5V * 380mA).

Software that performs read and write operations while downloading something from the internet and running several threads could be written to see how much the power consumption is under heavy load. Unfortunately, the project was already pushing the time limit, so an extensive stress test was not performed.

9.11 Summary of lessons learned

If rapid development is a priority, then an effort should be made to avoid deviation from known reference designs. Especially if the software available is tailor made for the reference designs. The KybMo differs from the reference designs in that it utilizes the RMII interface for the Ethernet PHY instead of the MII. It also utilizes UART0 as the main communication channel instead of UART2. Furthermore, the KybMo has no flash memory chip and relies only on the MMC/SD card interface for non-volatile storage. The reference designs all utilize flash memory and the software for booting U-Boot and Linux is configured for booting from those flash devices.

The KybMo deviated from the reference designs in order to become as small and as power efficient as possible, without any redundant or non-desired functionality. RMII uses less connections than MII and it therefore needs less routing space. RMII PHYs are naturally also physically smaller than their MII counterparts. Using UART0 as the main channel enables the possibility of leveraging all the UARTs on the board, as well as other functions that these remaining UART channels are multiplexed with. UART1 and UART2 are multiplexed with SPI chip select signals and are both routed out to the expansion header. This gives the KybMo the flexiblity of choosing whether to utilize more UARTs, or to have more SPI chip select signals, all dependent on the application. Flash memory was not included in the design to keep the board-size at a minimum. MMC/SD support was preferred for non volatile storage based on the practical flexibility and advantages of removable media.

These choices affect the software development by demanding that more time be spent on customizing code during the initial stages. However, customizing the basic underlaying layers of software is a one time job, and the long term advantages of a smaller board and less power consumption outweighs the cons.

9.12 Resources

This section provides an overview of the System Development Kit, and lists the relevant files that were edited in the validation process. All the files that were modified are tagged with the string: "KybMo" in the places were modification was necessary. Old code was not deleted, just commented out. Important wiki pages with instructions on how to perform certain actions are also listed.

9.12.1 Linux SDK

- Linux SDK for OMAP-L138 / TMS320C6748 EVM

The Linux system development kit for the OMAP-L138 is a kit that includes several development tools, software packages, and scripts that can be used with the OMAP-L138. Contents:

- **Platform Support Package:** Linux Kernel 2.6.37, Boot loader (u-boot).
- **Multimedia Package:** Codecs, DSP optimized Codecs, Encoders/Decoders.
- **DSP Package:** C6Run (tool to easily run C code on the DSP), C6Accel, DSPLink Inter Processor Communication, DSP/BIOS Real Time Operating System, C6000 code generation tool chain.

The SDK can be found on the CD. Both the original compressed package, and an uncompressed version with edited files are provided. The SDK can also be downloaded from: `http://www.ti.com/tool/linuxsdk-omapl138?DCMP`

Note: TI also provides a Windows Embedded CE/Compact SDK. The Windows CE SDK is advertised as a Board Support Package (BSP) that provides a clear separation between CPU-dependent and board-dependent features. That is an interesting quality, since the Linux SDK is custom made made for a specific board, with no clear separation between CPU-dependent and board-dependent features, which makes modifying the code for a different board other than the EVM a challenging task. Unfortunately, the Windows SDK was discovered at a late stage and there was not enough time to evaluate it. It can be downloaded for free from:

`http://www.adeneo-embedded.com/Products/Board-Support-Packages/Texas-Instruments`

9.12.2 Uboot

- **da850evm.h** `(SDKpath)/psp/u-boot...sdk/include/configs`
- **da850evm.c** `(SDKpath)/psp/u-boot...sdk/board/davinci/da8xxevm`

9.12.3 Linux Kernel

- **davinci_emac.c** [Ethernet related code] `(SDKpath)/psp/linux-2.6...sdk/drivers/net`
- **uncompress.h** [Kernel decompression] `(SDKpath)/psp/linux-2.6...sdk/arch/arm/mach-davinci/include/mach`
- **da850.c** `(SDKpath)/psp/linux-2.6...sdk/arch/arm/mach-davinci`
- **boardda850evm.c** `(SDKpath)/psp/linux-2.6...sdk/arch/arm/mach-davinci`
- **misc.c** `(SDKpath)/psp/linux-2.6...sdk/arch/arm/boot/compressed`

9.12.4 SD Card

- **mksdboot.sh** [Creates bootable SD card] `(SDKpath)/bin`

9.12.5 Practical TI Wiki Pages/Literature

There is a lot of information on the TI wiki pages, and it's easy to get lost in the maze of articles out there. It took a lot of time to get a proper overview of the relevant information. Below is a list of the most useful pages that were used in the project. It is important to be aware of the fact that these pages are often written by TI community members, and sometimes the information might not be 100% correct. Nonetheless, the articles are still valuable resources and provide good pointers.

- **TI Embedded Processors Wiki:** `http://processors.wiki.ti.com/index.php/Main_Page`
- **OMAP-L138 Software Developers Guide:** `http://software-dl.ti.com/dsps/dsps_public_sw/sdo_sb/targetcontent/dvsdk/DVSDK_4_00/latest/exports/OMAPL138_Software_Developers_Guide.pdf`
- **Building Software Components for OMAP-L1/AM1x:** `http://processors.wiki.ti.com/index.php/GSG:_Building_Software_Components_for_OMAP-L1`
- **Building 03.22 PSP release Components for OMAP-L138:** `http://processors.wiki.ti.com/index.php/Building_PSP_Components_for_OMAP-L138_on_v3.x_Kerenl`
- **Building The OMAP-L1 SDK:** `http://processors.wiki.ti.com/index.php/Building_The_OMAP-L1_SDK`

- **OMAP-L138 Software Design Guide:** `http://processors.wiki.ti.com/index.php/OMAP-L138_Software_Design_Guide`
- **OMAP-L1x Debug Gel Files:** `http://processors.wiki.ti.com/index.php/OMAP-L1x_Debug_Gel_Files`
- **Programming mDDR/DDR2 EMIF on OMAP-L1x/C674x:** `http://processors.wiki.ti.com/index.php/Programming_mDDR/DDR2_EMIF_on_OMAP-L1x/C674x`

9.12.6 Software used

- AISgen
- UART Boot Host
- Code Composer Studio 5
- Linux SDK for OMAP-L138 Evaluation Module (EVM) from Logic-PD.
- OMAPL138 StarterWare
- Virtual Box
- Ubuntu 10.04
- Tera Term hyper terminal
- Br@ay++ terminal

9.13 Configuration & Set-Up

9.13.1 Power

The board can be powered up through the USB 2.0 port connected to a host machine, or through a 5V power supply.

Important: Make sure the watch dog-out (WDO) signal coming from U8 at the bottom of the board and going into the logical AND gate U9 on the top side is pulled high to 3.3V. Also, the WDO pin of U8 should be isolated from the connection pad. The proper configuration is displayed in Figure 9.22:

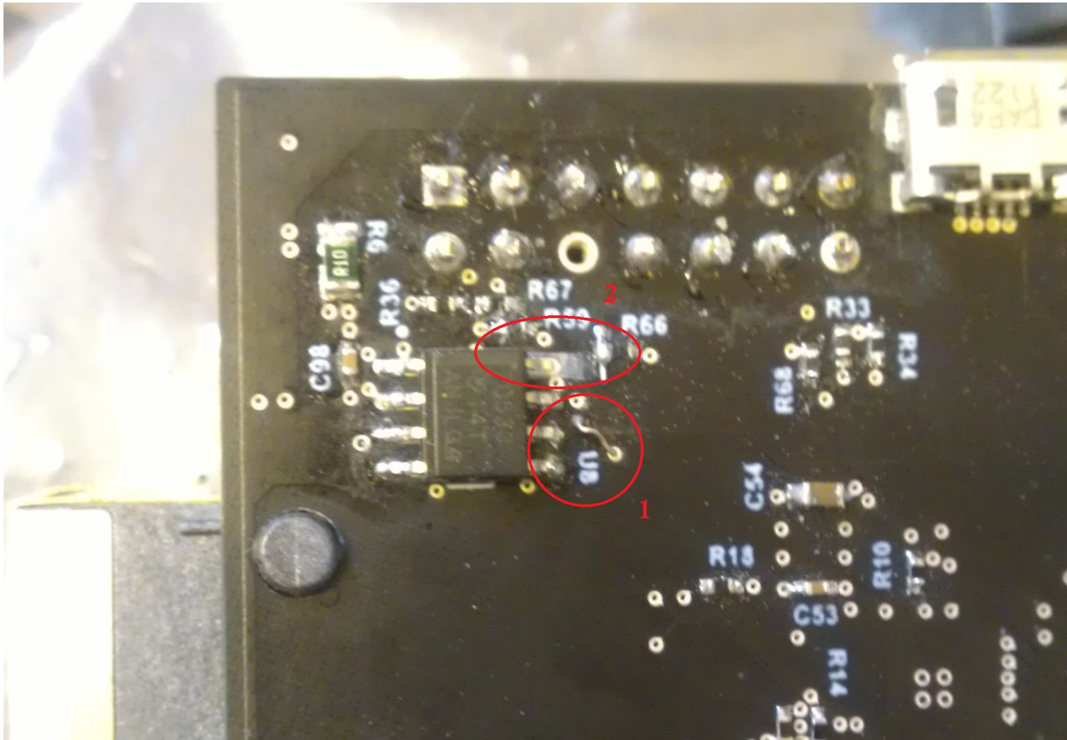


Figure 9.22: **1:** wire connecting the WDO signal from U8 to 3.3V **2:** Plastic tape isolating the WDO pin from the pad.

9.13.2 Serial Communication

Install the FTDI virtual com port drivers on the development PC in order to be able to commuicate with the USB to UART port on the KybMo. The drivers can be found on: `http://www.ftdichip.com/Drivers/VCP.htm`

Use Tera Term hyper terminal on windows with the following settings to communicate with the U-boot and Linux consoles:

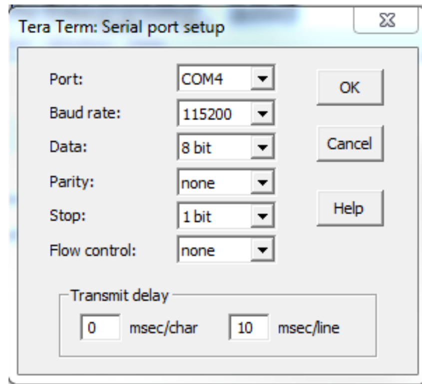


Figure 9.23: Tera Term Serial Port configuration.

9.13.3 Booting

It is not possible to stay connected to the virtual COM port while the KybMo is turned off. Therefore, one must connect after the board has been powered. U-Boot waits for 3 seconds before booting Linux. The boot process can be stopped by pressing the enter key within the first 3 seconds.

9.13.4 USB 2.0:

Until the issue of why the USB0_DRRVBUS is inactive gets solved, be sure to connect the VBUS signal of the USB2.0 connector as shown in Figure 9.15 to use the KybMo as a USB host for a slave device. However, **do not** use this configuration when connecting the USB2.0 port to a PC.

Chapter 10

Discussion and Thoughts for the future

Several aspects of the project have been discussed throughout the report. However, certain topics remain untouched. One important issue in regards to the feasibility of future similar project is economy. Over 7000 NOK was spent on components and PCB production. Components for three copies of the board was purchased, and 5 PCBs were manufactured.

Crunching the numbers yields (all in NOK):

4314 (PCB) / 5 = 862 ,-
2664 (Components) / 3 = 888 ,-
420 (JTAG) + 151 (Polyester Stencil)= 571 ,-

862 + 88 + 571 = 1521 NOK pr card.

Which doesn't seem like too bad of a price, except for the fact that ordering one single PCB would not cost 862,-. It would cost 2976 NOK, since the tooling cost of the PCB manufacturer(2646 ,-) for this design remains the same no matter how many PCBs are ordered. That would take the price of the system up to 4435 NOK, more than ten times the price of a Raspberry Pi for instance. But if we assume that an off the shelf commercial product has exactly what the application requires, and can be bought for a low price, then that still doesn't do anything for the student in terms of acquisition of knowledge and competence development within the field of advanced circuit board design.

Furthermore designs such as the KybMo are doomed to fail in terms of cost if they're pit up against their commercial counterparts, who are mass-produced and manufactured with no middle men. Just the price of the components ordered from the Digi-Key corporation for

the KybMo was 888 NOK. The price of the Micron DDR2 chip used on the KybMo alone is equivalent to the price of a Raspberry Pi board. However, it should be noted that the Raspberry Pi is a charity project, so it might not be a realistic bench-mark in a price discussion. Still, if we examine the closest relatives of the KybMo, the BeagleBone, HawkBoard, and the Beagleboard platforms, that are marketed in the range of 120-200 USD, the KybMo still can't come out on top. Keep in mind that most of the components on the KybMo can be found on these sister platforms as well. Not only that, but the KybMo uses less components than these designs, and the component price alone is the same as the marketed price for these systems. Regardless, the knowledge and competence a student comes out with after designing such a system is worth a lot. A good solution could be to make an effort of setting up a sponsorship arrangement with a couple of companies who might be interested in being involved. There have been years where the EuroBot project has gotten up to 100 000 NOK, so sponsorship arrangements for such projects is not that far-fetched of an idea.

But of course, reducing the amount of layers in the PCB design would drastically reduce the tooling cost and the PCB price. There is definitely room for optimization. Choosing an SoC with integrated memory, or one that offers PoP (Package on Package) support, where the memory chip can be placed directly on top of the SoC (used on the Beagleboard), would make it possible to reduce the amount of layers down to 6 or even possibly 4. Or even being a bit brave and reducing the amount of power and ground layers could very well result in a successful design. A way of doing this while staying as safe as possible, could for example be by creating power and ground islands on the top and bottom layers, right above and below the DDR interface. This could ensure proper shielding and alleviate the need of extra power and ground planes. The Ethernet interface could be replaced with a wifi module, which would reduce the size of the board and open the doors for mobile communication.

If creating such designs from scratch is not a desirable process to repeat, there is still the possibility of assembling 2 more KybMo systems with the components that are left, and using them in embedded designs. DSP development is an interesting field, and an attempt at writing an image processing application on the KybMo system should definitely be made. It would also be interesting to create expansion cards for the KybMo utilizing the expansion headers, in order to increase the potential for further functionality.

Conclusion

High-speed design theory was an unfamiliar topic, and no one at The Department of Engineering Cybernertics had any hands on practical experience with high-speed design prior to this project. The topic was thoroughly studied, as careless mistakes, especially in the DDR2 interface, could compromise the functionality of the board. The knowledge acquired was presented in detail for the benefit of future students. It was important to find, and learn how to use a CAD tool that could implement the required high-speed design rules. It was discovered that the ITK department had limited capabilities in terms of the software lisence they were in possession of for OrCAD. An Altium Designer lisence was borrowed from the IET department, and the most important information on how to use the relevant main functions of Altium Designer was presented.

The KybMo was designed from scratch, all schematic symbols (except the OMAP-L138's, which were provided by TI) and footprints for the devices were created by using the specifications provided in the data sheets. All the routing was done manually from point to point.

A high density eight layer design was sent to fabrication, which was also an important process to document properly. Practical knowledge had to be gained in terms of how to assemble advanced, fine pitch SMT components. The assembly of 206 components in total, including BGAs and QFNs, was sucessfully carried out and documented.

Software, including the Linux kernel, was modified in order to work with the KybMo and was used to validate the functionality of the most important peripherals.

The Department of Engineering Cybernetics now has a valuable reference design, that can be used as a source of theoretical and practical information on various independent subjects. This is precious information that previously couldn't be found structured in one place, under a unified purpose. The KybMo can also serve as a basis for future application development on autonomous vehicles.

The project resulted in a functional system that combines the highly desirable traits of low power consumption, high-end computational performance through a dedicated DSP, support for high speed peripherals and popular open source operating systems, such as Linux, support for software packages that allow rapid DSP development, and a form factor of approximately 6x6 cm. A combination that can't be found in any of the available off the shelf commercial solutions. Which makes it possible to conclude that the design and assembly of high end complex microelectronic designs can be carried out at the NTNU campus, as long as all the relevant aspects are well understood and thoroughly carried out.

The road to low-power, rapid DSP application development has been paved. Future work can focus on evaluating the performance boost a low power DSP provides, and the implementation of computationally intensive image processing can start. Creating embedded designs that use the KybMo's high speed and I/O peripherals, utilizing it as a control module, is also possible. The design can be further improved by reducing both the amount of layers, and the total form factor by switching to an SoC with integrated DDR2 memory while utilizing most of the KybMo's existing peripheral and power supply design architecture.

Appendix A

Advanced Multilayer & High-Speed Printed Circuit Board Design

A.1 Introducion

The aim of this chapter is to demystify the process of designing advanced multilayer Printed Circuit Boards (PCBs) that implement high speed peripheral buses. Researching this subject and acquiring the required knowledge within all the different parts of the process is a challenging task. This chapter, which is intended to be used as a stand-alone compendium on the topic, attempts to make research easier by gathering most of the essentials in one place. The basic concepts one should know about will be presented, and relevant literature will be provided to gain a better understanding of the various topics. Also, for those who are interested in NTNUs manufacturing capabilities, an overview of what can and can't be done locally at campus is included.

Each sub-chapter ends with a "recommended reading" section, where relevant resources and references are listed at the end of the chapter. In section A.6, all references and recommended literature are sorted by topic. References can be found on the CD indexed with the same reference numbers used throughout the compendium. Recommended literature can be found on the CD indexed with the letters specified in the compendium. Section A.7 lists all the references with a brief description on each one in order to provide the reader with better insight.

A.1.1 High Frequency & High Speed, what's the difference?

This might be a natural first thought to a lot of people. Generally we can say that there are two basic types of circuits that fall under the heading of "High Frequency" [1, Hartley 2002 p.1]:

- RF/Analog (aka-RF/Microwave)
- High Speed Digital

The article will focus on digital circuits, but most of the theory can be applied to RF/Analog circuits as well. As mentioned above, a high speed digital circuit is regarded as a high frequency circuit.

A.1.2 High Speed/High Frequency Peripheral buses

It is useful to have an idea of what's actually being referring to when we're talking about High Speed peripheral buses. Some of the most common high speed peripheral buses are:

- DDRX RAM: 100/1600 - 266/17066 [MHz/MBs] (IO Bus clock up to 1Ghz)
- USB2/3: up to 480 Mb/s for USB2, 5Gb/s for USB3
- Ethernet: up to 100 Mb/s
- Gigabit Ethernet: up to 1Gb/s

A.2 Multilayer Printed Circuit Boards

Figure A.1: Multilayer PCB (www.kursus.com)

Generally, the term "multilayer PCBs(Printed Circuit Boards)" refers to boards that use more than two layers. That is, signals are routed on more than two physical layers of copper. Typically, two-layer boards consist of two copper surfaces separated by what is called a dielectric material (often FR-4). Signal traces are etched on the copper, and components such as integrated circuits and resistors/capacitors, etc. are connected to these signal traces by soldering the component's pins to the traces. (More specifically, the pins are soldered to one of the ends of a trace, where the copper is a bit wider than the rest of the trace. This wider part is called a pad, and is especially formed to be wider than the trace to accommodate soldering).

A multilayer board is based on the same concept, but in addition to the top and bottom layers, it also has internal layers. Each layer is separated by a dielectric material. Components are only soldered on the top and bottom layers, while signals are routed between the surface and internal layers by using what is referred to as "VIAs". A via is a hole that connects a trace on one layer to another trace on a different layer. Vias are also used on two-layer boards to connect the top and bottom layers. Multilayer boards might use micro-vias,

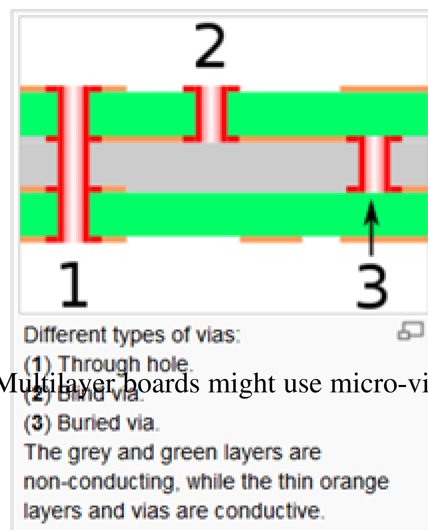


Figure A.2: Vias(wikipedia)

these are vias that are either blind or buried. A blind via is a via that starts on a surface layer and stops on an internal layer. A buried via is a via that only connects internal layers together, without going through any of the surface layers. A "‘regular"’ via that goes from one end of the board to another is typically referred to as a "through hole" via.

A.2.1 Why multilayer?

Multilayer boards provide the following advantages:

- Better EMC performace
- Reduced power and ground impedance
- More space to route signals

Radiation

When we’re dealing with high frequency peripheral bus design, we are required to keep in mind certain electrical phenomena that were negligible or nonexistent at lower frequencies. One of these phenomena is related to the EMC performance of the board: radiation from high speed signal traces. It is desired to shield these traces somehow. This is typically done by using a multilayer design and routing the high-speed signals on a layer that is buried between a ground and a power plane.
Even though two-layer boards have been used successfully in unshielded enclosures at 20 to 25MHz, these cases are the exception rather than the rule. It is recommended to consider multilayer boards once you go above about ten or fifteen MHz [2, Ott p.2].

Ground & Power planes?

As mentioned above, high-speed signals can be shielded by burying the layer they are routed on between a ground and a power plane. But what are ground and power planes? Typically, in the breadboard world and single/two-layer boards, power is provided to a component by routing a physical trace all the way from the power supply to the component. But there is another way to provide the power: dedicate an entire conductive layer to the power by sending the power signal from the power source directly to that layer, and whenever you need power for a component, you just bring the power up from the power layer. This eliminates the need of a signal trace between the component and the power supply, freeing up space that can be used to route other signals.

Furthermore, it is important that the components are able to draw current without causing voltage drops. This means that the power supply impedance must be as low as possible.

Using an entire layer for power will result in low power supply impedance. Likewise, using an entire layer for ground is also advantageous. A ground plane will decrease the ground impedance (and therefore the ground noise), in addition to ensuring that the ground reference level, or potential, is the same all over the board.

Signal Escape

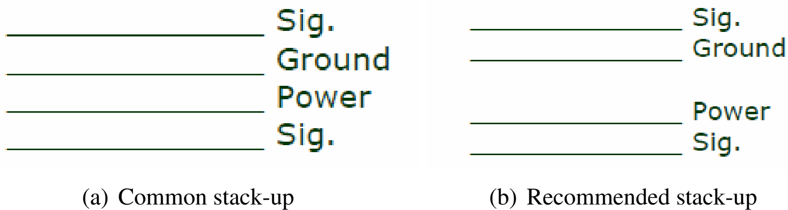
Often, working with systems that support high speed peripheral buses can entail working with components that have a large amount of pins, or components that come in Ball Grid Array (BGA) packages(see 3.1). Such components might require more than one or two layers to successfully route out all the signals.

A.2.2 Layer stack-up

After deciding to go through with a multilayer design, it is important to arrange the layers in such a way that will work well with high-speed signals.

The article titled "PCB-Stack Up" written by Henry Ott[2] discusses the most common configurations for layer arrangement on 4, 6, 8, and 10 layer boards, with the advantages and disadvantages of each configuration. Ott recommends the follow configurations for 4 and 6 layer boards:

4-Layers



(a) Common stack-up (b) Recommended stack-up

Figure A.3: 4-Layer stack-ups

The recommended configuration specifies that the spacing between the signal and planes should be less than 10mils, and the spacing between the planes should be larger than 40mils. PS: a "mil" is not the same as "mm". A "mil" is 1/1000th of an inch, or a "milli inch". This unit is extremely common in the PCB world.

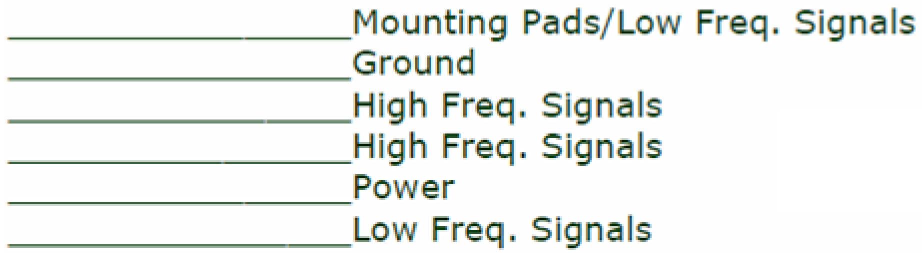


Figure A.4: Common & acceptable stack-up

6-Layers

A.2.3 Recommended Reading

- Reference 2, Henry Ott: PCB Stack-Ups

A.3 Integrated Circuit Packages

By package, we mean the physical format of the IC. Below is a chart displaying the most common packages.

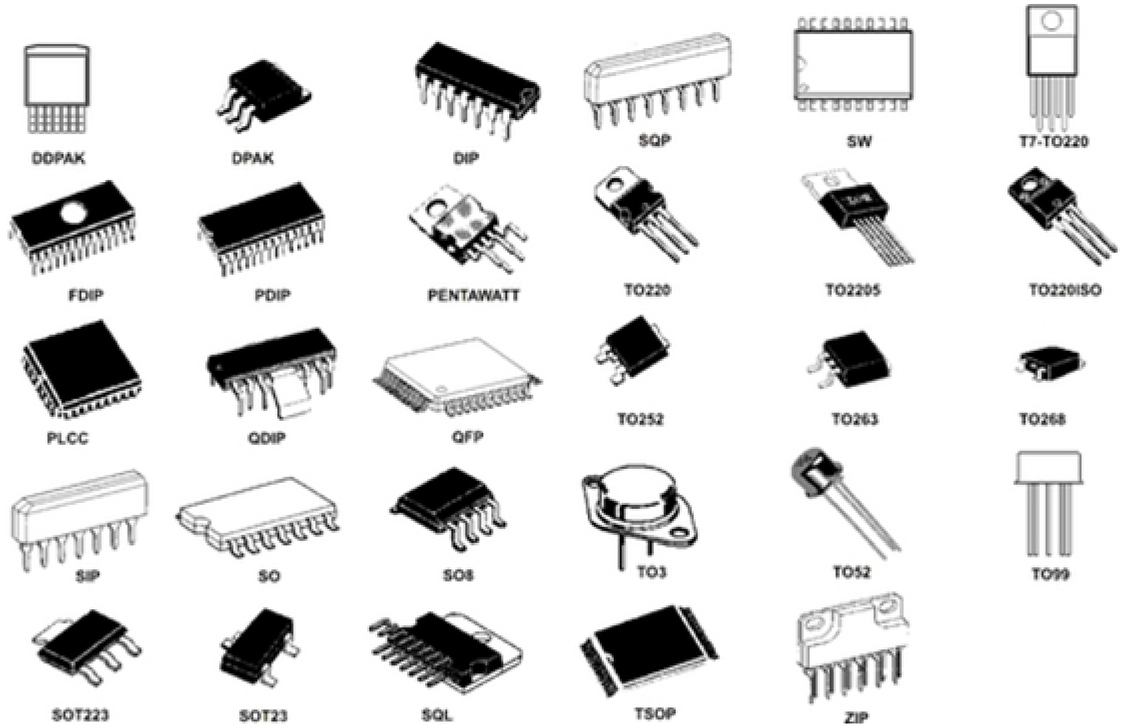


Figure A.5: IC Packages(www.sparkfun.com)

A.3.1 Ball Grid Array(BGA)

BGA packages differ from other packages by utilizing small balls instead of conventional pins.

The balls are located under the IC, which makes the soldering process complicated. This package is however widely used for advanced high speed ICs because [3 & 4]:

- It solves the problem of producing a miniature package for an IC with many hundreds of pins

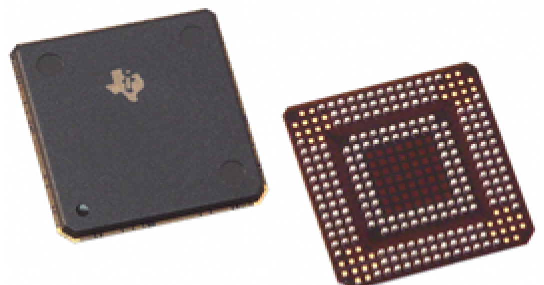


Figure A.6: BGA package(www.sparkfun.com)

- Lower Thermal Resistance between the package and the PCB allows heat generated by the IC to flow more easily to the PCB, preventing the chip from overheating.
- Low-inductance leads as a result of the short distance between the package and the PCB. Inductance is a property that causes unwanted distortion of signals in high-speed electronic circuits

BGA Assembly Process

Packages with conventional pins can be assembled through soldering by hand with a solder iron. This is not possible on a BGA due to the physical arrangement of the balls (under the IC) and the fine pitch between each ball. The good news is that the balls usually come with solder already applied to them by the manufacturer, relieving the person doing the assembly from placing solder on each pad and ensuring that no bridging between the different balls with solder occurs. The following steps outline the assembly process:

1. **Fluxing**: Either solder paste(cream), paste flux, or liquid flux (i.e., spraying, dispensing, or foaming) must be applied to the pads on the PCB. This is necessary to reduce oxides formed on the pads and the solder balls [5, Motorola p.5]. Furthermore, the tackiness of the paste/flux hold the components temporarily in place until soldering occurs [6, Indium p.1]
2. **Device Placement**: Both manual placement by hand, or automatic Pick and Place machines can be used. Due to the fact that the BGA is self-centering in the reflow process, a device can be placed up to 50% off-pad and still be expected to align itself [5, Motorola, p.6]. The self-centering feature is a result of the surface tension of the molten solder.
3. **Reflow**: "Reflow Soldering" refers to the process of melting the solder by applying controlled heat to the entire assembly, or to specific areas of the board. Several methods and different equipment can be used to achieve this: Passing the assembly through a reflow oven, using an infrared lamp, or a convection rework machine. Care must be taken and temperatures must be controlled, due to the fact that excessive heat can damage the IC.

Assembly Challenges

Several problems might arise during the assembly [7]:

1. The heat applied might have a temperature that is not sufficient enough to solder all the balls. In this case, some balls are soldered while others are not.
2. Too high of a temperature might cause shortening of the balls, due to the solder melting more than what is desired.
3. In the scenario of silver filled pads: There is a via within the BGA pad and the hole is filled with silver. If the via is not properly filled with silver, it can cause joint integrity issues. The via can drop down, leaving a pad with a hole in between. The solder then flows through this hole to the other side of the board creating an "open ball": Either a marginal or no connection is made.
4. A void: no connection to a particular solder joint region on a BGA ball. Acceptable joints are specified in industry standards set by the*Industry association for printed circuit board and electronics manufacturing service companies* (IPC). IPC Class I specifies a joint with a gap of up to 40% of the ball size as acceptable. IPC Class II: a void up to 25%, and IPC Class III: a 12-15% void is acceptable.

Detecting Assembly Faults in BGA packages

- **X-Ray**: Most professional boards with BGA packages are inspected by either 2D or 3D X-Ray imaging. X-ray makes it is possible to detect any of the problems mentioned in the previous section.
- **Boundary Scan**: Another, less expensive way, is to perform electrical testing using a method called "Boundary Scan".

  Boundary Scan uses IEEE-1149.1 JTAG, which provides a means to test interconnects and clusters of logic without using physical test probes. It is important to note that the IC has to support boundary scan. Once the circuit is set into a test mode, the normal function of the pins/balls will be overridden, and the Boundary scan will run test procedures through the JTAG interface to test the connectivity of all the pins/ball. Typically, the JTAG system will allow the import of a BSDL(Boundary Scan Description Language) file that is specific to the component being tested. This file should be provided by the manufacturer and describes the contents of the boundary scan for the component. The JTAG testing system uses the BSDL file to automatically generate test applications.

  It is common to have a board with several components that one wants to test with a boundary scan. This is typically done by setting up a boundary-scan chain. Resources O-Q descrive how a chain of devices should be connected, how to route the signals

out of the JTAG Interface Connector, and what considerations and engineer should take into account.

A.3.2 Recommended reading

- **References 3-5.** See reference table.
- **Reference 6**, Soldering 101- A Basic Overview.
- **Reference 7**, BGA Assembly.
- **O.** Considerations- Boundary Scan Chain: `http://www.corelis.com/education/Tips_DFT_Considerations_Boundary_Scan_Chain.htm`
- **P.** Design Engineers Role in Boundary-Scan Test: `http://www.corelis.com/education/Design_Considerations_for_the_Engineer.htm`
- **Q.** Wiki: `http://en.wikipedia.org/wiki/Boundary_scan`

A.4 Routing

Routing is the process of laying out physical traces in a PCB design. Certain factors have to be taken into account when routing high-speed signals. These factors are, according to[8, Shust&Cobb 2008, p.3]:

- Flight time delay and skew
- Signal integrity and impedance matching
- Crosstalk
- Power supply bypassing
- EMI (Electro Magnetic Interference)

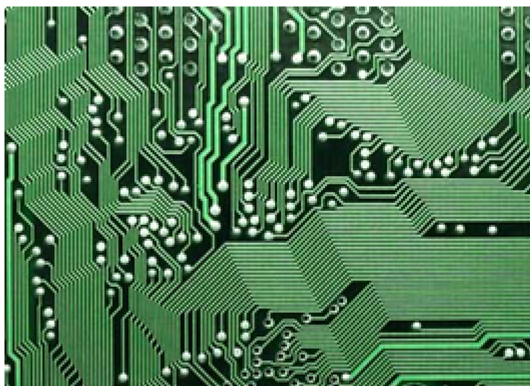


Figure A.7: PCB signal traces (www.locxtronic.com)

A.4.1 Flight time delay and skew

Certain signals (usually signals within the same clock domain) need to arrive at the same time. The physical length of the traces is the main factor in determining when signals arrive. Keeping the so called "skew" (difference in trace length) between the traces to a minimum is desired. The Datasheet of the microprocessor used should specify the maximum skew allowed within certain clock domains for high speed signals such as DDR signals.

A.4.2 Signal integrity and impedance matching

Signal integrity refers to controlling overshoot, ring back, and transition edges. The concept of "impedance matching" revolves around making the signal "see" as little change as possible on its way from one point to another. If a signal moves from a trace with certain impedance, onto another trace with different impedance, then it is said that there is an impedance mismatch, and a phenomena called Reflection occurs [9, Chen p.9]. This can cause unwanted distortion and can affect the functionality of a component (by producing overshoot/undershoot, ringing, etc..).

Effective signal integrity control on high-speed designs requires:

1. That the impedance of the PCB traces be controlled. Trace impedance is governed by the trace width as well as the thickness and dielectric constant of the PCB insulating material (usually FR-4). **Fortunately for the PCB designer, this aspect can often be left to the PCB fabrication contractor by specifying the desired single**

ended impedance for the various PCB traces. The Datasheet usually specifies the required impedance on specific lines. The term "Single Ended" means the signals use a common ground potential as reference, as opposed to "differential" signals, which are compared to each other.

2. Choosing an appropriate termination scheme. Some form of reflection will always be present due to minor production and design inconsistencies. Reflection is minimized by terminating the signal properly. A simple and common termination scheme is to use a resistor as the terminator, and hook it up in series with the signal at the end of the trace. This is referred to as series termination. Some components may require specific termination schemes to be used, other than series termination, while some may not need termination at all. Read the Datasheet!

3. Avoiding stubs and sharp turns in the traces. Route the signals with the minimum amount of vias, and use two 45 degree turns instead of a 90 degree turn, to avoid reflection issues and impedance discontinuities [10, TI 2007, p.2]. However, avoiding sharp 90 degree turns is only a general recommendation and should not be taken too seriously. Dr Howard Johnson's article titled "Who's afraid of the big bad bend, May 2000" discusses how 90 degree turns have very little impact on signal integrity. To read the article: http://www.sigcon.com/Pubs/edn/bigbadbend.htm

A.4.3 Crosstalk

Crosstalk is when a signal on one trace creates an undesired effect on another trace. Control crosstalk by:

1. Controlling the current return path. Each signal routing layer should have an adjacent full ground plane to provide the shortest return current path.

2. Spacing between signals. Spread the signals out beyond the PCB minimum spacing capabilities. Just because a manufacturer can support as low as 4 mils minimum spacing doesn't automatically mean you should space all your signals at 4 mils.

Minimum spacing between critical lines (or *nets* as they are called) might be specified in the datasheet of the component.

A.4.4 Power supply bypassing

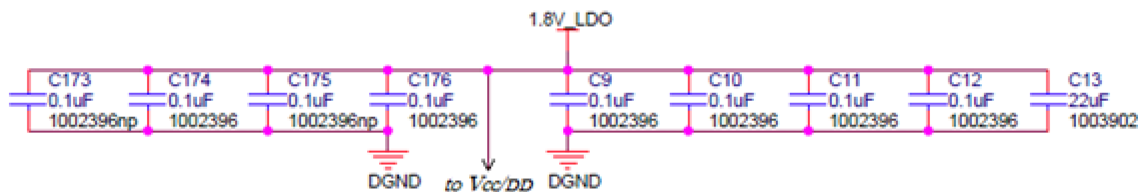


Figure A.8: Schematic of a typical bypass capacitor configuration (From the Logic-PD SOM M1 reference design files)

Power supply high-frequency impedance is beaten down by utilizing many physically small capacitors connected between the power and ground planes. Using many capacitors, rather than one large one, results in their parasitic inductances being placed in parallel, thereby, reduced. The parasitic inductance of a capacitor is dependent upon its size. These capacitors are referred to as "bypass capacitors" or "decoupling capacitors". The bypass capacitors need to be placed very close to the device they are bypassing.

To optimize the effectiveness of decoupling capacitors, surface-mounted capacitors mounted on the bottom-side of the PCB keeps the parasitic effects to a minimum. Placing capacitors directly underneath the BGA package will improve the high frequency response of very small value capacitors. [11, Lattice 2011, p.8] Using surface mounted capacitors, the layout should not be allowed to reduce their effectiveness by connecting them through long, skinny traces leading to the power and ground. Use large or multiple smaller vias, and use short and fat traces to capacitors where possible. [11, Lattice 2011, p.18]

The datasheet for a specific component might include how many capacitors are required and give guidelines for placement.

A.4.5 EMI (Electro Magnetic Interference)

Following are the steps that can be followed to control the EMI of the system.

1. Termination: Even though certain components might not require the use of a termination scheme to achieve acceptable signal integrity, it doesn't mean that omitting termination from the design is good from an EMI perspective.

   Termination that reduces ringing also reduces EMI, but the terminations comes at a cost of increased bill of material[1] count and PCB space. Leaving optional terminations off a design may result in an EMI certification failure. A middle of the road approach is to design with terminators and populate them with zero Ohm resistors. A prototype is then manufactured and checked for EMI compliance and only those terminations that are required are added back. It is easy to remove the zero Ohm terminators in a board spin[2]. It is much more difficult to add terminators if no space were originally reserved for them.

2. Minimizing the number of signal vias reduces EMI by reducing inductance at high frequencies. [10, TI 2007, p.2 & 3]

3. Route all high-speed signal traces over continuous planes (Power or Ground), with NO interruptions in the planes. Avoid crossing over anti-etch, commonly found with plane splits. [10, TI 2007, p.2] [2, Ott 2002, p.16]

4. If you absolutely must route the signal across the split plane, place a few small stitching capacitors across the split, one on either side of the trace. This will provide high-frequency continuity across the split while maintaining dc isolation between the isolated sections of the split plane. The capacitors should be located within 0.1" of the trace and have a value of 0.001 to 0.01 uF according to the frequency of the signal. [2, Ott 2002, p.16]

[1] Bill of material, abbreviated "BOM", is a list of all the components/materials used in a design

[2] A "spin" can be regarded as an update/change in the initial design.

A.4.6 Recommended Reading

The resources below indexed with letters can be found on the CD with the same indexation.

- **A.** References 8, 10, and 11. See reference table.
- **B.** Altera High Speed (Application note on high speed board design)
- **C.** PCB Design Guidelines (check list of factors that should be kept in mind)
- **D.** High Performance Multilayer PCBs Design and Manufacturability. (General PCB design fundamentals with the focus on high performance materials instead of FR4)
- **E.** Bypass Capacitor Selection for High-Speed Designs (Technical note from Micron presenting electrical theory that can be applied for bypass capacitor selection)
- **F.** Reference 9, Chen-Signal Integrity
- **G.** Controlling Controlled Impedance Boards (Article that presents ways of how to achieve controlled impedance through field solving software)

A.5 Manufacturing Capabilities

Unfortunately, there are certain limitations as to what can be produced locally at the NTNU campus. This section will give a brief overview of what is possible and not possible.

A.5.1 Multilayer PCBs

It is only possible to produce up to two-layer PCBs at the NTNU Engineering Cybernetics workshop. If more layers are to be used, it is necessary to find an external manufacturer. There are various PCB manufacturers to choose from on the internet, these are often located in other countries but shipping is usually not a problem. An affordable manufacturer that has been used by students at the university and which often comes recommended is: `www.pcbcart.com` Do not assume that all manufacturers can produce everything you want them to. Most manufacturers have certain mechanical limitations, and it is important to get an overview of their capabilities before making a choice. In order to avoid creating unrealistic designs, getting a rough overview of what is regularly possible is recommended before starting the design stage. Below is a figure displaying pcbcart's capabilities to serve as an example for typical limitations a PCB manufacturer might have.

Features	Capability
Layers	1-10 layers
Material	FR4
Copper Thickness	1/2 to 2oz (18um-70um)
Board Thickness	.016-.126" (0.4mm-3.2mm)
Surface finish	HASL,Ni/Au,OSP
Soldermask	LPI, different colors
Board Dimension	600x700mm
Min Hole Diameter	8mil (0.2mm)
Min line width	8mil (0.2mm)
Min line spacing	8mil (0.2mm)
Min SMT pitch	16mil (0.4mm)
Min. Annular Ring	.008" (0.2mm)
Aspect Ratio	5:1
Surface/hole plating	ave. 25um min. 20um
Tolerance:	
Hole Tolerance (PTH)	.002" (0.05mm)
Hole Tolerance (NPTH)	.003" (0.075mm)
SM Tolerance (LPI)	.003" (0.075mm)
Dimension	.004" (0.1mm)
Electrically test	10V-250V, flying probe or testing fixture

Figure A.9: PCBcart capability chart (www.pcbcart.com)

A.5.2 BGA Assembly

Ball Grid Array Package assembly requires more advanced equipment than what can usually be used on habby-level oriented designs.

However, the Electronics & Prototyping Lab (Elpro-lab) at NTNU's Institute of Electronics and Telecomunications, located on the 2nd floor in the C-block (room C251), has recently acquired an Advanced Package Rework machine, APR5000 from Metcal. This machine can be used to assemble BGA packages by locally heating up the component with convective heat, causing the solder on the BGA balls to melt, as described earlier in the "BGA Assembly Process" section. It can also be used to remove an already soldered BGA IC, thus the name "rework".

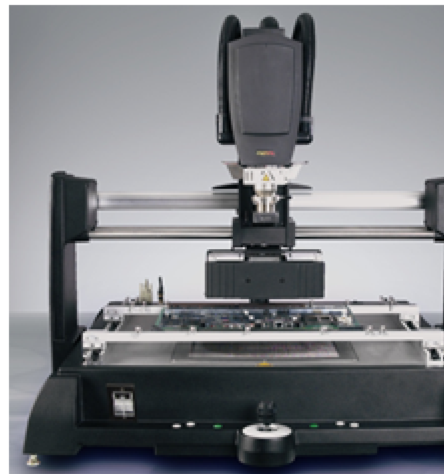


Figure A.10: Metcal APR-5000

A.5.3 Tools

It is important to note that usual, widely used hobby-level oriented tools, like Eagle CAD, can't be used to design advanced multilayer high speed systems. Support for more than 4 layers is usually not present. Defining high speed constraints such as differential pair, and timing & delay constraints, is not possible. In addition, working with BGA packages that use a large amount of pins can be very difficult. The only option is to use high-end, professional software packages. NTNU currently has licenses for two such software systems:

- Cadence OrCAD 16.2
- Altium Designer

A.5.4 Recommended Reading

The resources below indexed with letters can be found on the CD with the same indexation.

- **R.** Complete PCB Design Using OrCAD Capture and PCB Editor(Book accessible through an NTNU internet connection): `http://www.sciencedirect.com/science/book/9780750689717`
- **S.** A Basic Introduction to Cadence OrCAD PCB Designer.
- **T.** High-Speed Digital Board Design with Altium Designer, Istvan Nagy, Blue Chip Technology
- **U.** Jones, PCB Design Tutorial (practical PCB design fundementals)

A.6 Literature & Resources

A.6.1 Recommended Reading

Recommended literature is arranged by topic in this section and can be found on the CD in the folder titled "Recommended Reading", indexed by letters as specified bellow.

High Speed PCB Design

- **A.** References 8, 10, and 11. See reference table.
- **B.** Altera, High Speed Board Designs (Application note on high speed board design)
- **C.** PCB Design Guidelines (check list of factors that should be kept in mind)
- **D.** High Performance Multilayer PCBs Design and Manufacturability. (General PCB design fundamentals with the focus on high performance materials instead of FR4)
- **E.** Bypass Capacitor Selection for High-Speed Designs (Technical note from Micron presenting electrical theory that can be applied for bypass capacitor selection)

Signal Integrity and Controlled Impedance

- **F.** Reference 9, Chen-Signal Integrity
- **G.** Controlling Controlled Impedance Boards (Article that presents ways of how to achieve controlled impedance through field solving software)

Multilayer Stack-Up & EMI

- **H.** Reference 2, Ott- PCB Stack-Ups.
- **I.** Intel EMI Design Guidelines for USB Components

Ball Grid Array Package & Soldering

- **J.** References 3-5. See reference table.
- **K.** Reference 6, Soldering 101- A Basic Overview.
- **L.** Reference 7, BGA Assembly

PCB manufacturing details & Materials

- **M.** Reference 1, Hartley 2002- Base materials for High Speed, High Frequency PCBs
- **N.** How to build a PCB (overview presentation of the manufacturing process)

Boundary Scan Chain

- **O.** Considerations- Boundary Scan Chain: `http://www.corelis.com/education/Tips_DFT_Considerations_Boundary_Scan_Chain.htm`
- **P.** Design Engineers Role in Boundary -Scan Test: `http://www.corelis.com/education/Design_Considerations_for_the_Engineer.htm`
- **Q.** Wiki: `http://en.wikipedia.org/wiki/Boundary_scan`

Software Tools & Tutorials

- **R.** Complete PCB Design Using OrCAD Capture and PCB Editor(Available only through an NTNU internet connection): `http://www.sciencedirect.com/science/book/9780750689717`
- **S.** A Basic Introduction to Cadence OrCAD PCB Designer.
- **T.** High-Speed Digital Board Design with Altium Designer, Istvan Nagy, Blue Chip Technology
- **U.** Jones, PCB Design Tutorial (practical PCB design fundementals)

A.6.2 Reference Designs

These are board and schematic design files included on the CD in the folder titled "Reference Designs", and serve as good practical examples of actual high speed designs. The .DSN and .BRD files were generated in OrCAD Capture (dsn) and Allegro PCB Designer (brd), Allegro is part of the OrCAD package, and is referred to as "OrCAD PCB Editor" in the package.

- **BeagleBoard** (OMAP-3) `http://beagleboard.org/`
- **HawkBoard** (OMAP-L138) `http://www.hawkboard.org/`
- **LeopardBoard** (DM355) `http://designsomething.org/leopardboard/`
- **Logic PD OMAP-L138 SOM-M1** (OMAP-L138) `http://www.logicpd.com/products/system-on-modules/omap-l138-som-m1/`
- **PandaBoard** (OMAP-4) `http://pandaboard.org`

A.7 References

Reference #	Title & Description	Author
1	**Base Materials for High Speed, High Frequency PC Boards** Article that discusses the choice of different laminate materials for high speed designs. (Alternatives for FR4)	Rick Hartley, 2002
2	**PCB Stack-Up** Article that discusses Different layer stack up configurations, and Power/Ground plane considerations & practices that should be taken into account in regards to EMC and High Speed designs.	Henry Ott
3	**Ball Grid Array** http://www.siliconfareast.com/bga.htm Internet Article about the BGA package; it's advantages and disadvantages.	Siliconfareast.com
4	**Ball Grid Array (Wiki)** http://en.wikipedia.org/wiki/Ball_grid_array Wikipedia article about the BGA package	
5	**Plastic Ball Grid Array(PBGA)** [AN1231] Application Note about the BGA package. The P denotes the plastic substrate material to which the array is attached. A lot of the technical theory within applies to BGA packages in general.	Motorola
6	**Soldering 101- A Basic Overview** [97773] An Application Notethat gives a basic overview for those new to the world of soldering.	Indium Corporation

7	**BGA Assembly** http://www.bga.net/bga-assembly.aspx An internet article that describes the challenges and problems typically faced with BGA Assembly	Bga.net
8	**Understanding TI's Routing Rule-Based DDR Timing Specification** [SPRAAV0A] Application Report describing High speed design issues and recommended practices. Aimed to explain the foundation of the DDR Routing rules & specifications found in TI's datasheets.	Mike Shust & Jefferey Cobb, Texas Instruments, 2008
9	**Signal Integrity** A compendium chapter regarding Signal Integrity and related theory. In addition to Modeling and Simulation examples in IBIS.	Raymond Y. Chen, Sigrity, Inc., Santa Clara, California. E-mail: chen@sigrity.com
10	**USB 2.0 Board Design and Layout Guidelines** [SPRAAR7] An Application Report that discusses schematic and design guidelines when designing a universal serial bus system.	Texas Instruments, 2007
11	**High Speed PCB Design Considerations** [TN1033] A Technical Note describing high speed PCB design considerations, with a focus on physical PCB interconnections and backplane design. A lot of the theory applies to general high speed PCB design, and is not only specific to backplane designs.	Lattice Semiconductor Corporation, 2011

Appendix B

Terminology

- **Annular Ring**:

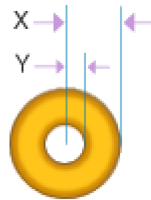


Figure B.1: Annular ring = X - Y, wiki.altium.com

- **Discrete component:** An elementary electronic device constructed as a single unit(pcmag.com), and not in an integreated circuit. Such as a stad alone resistor, capacitor, transistor or diode.
- **Floating pins:** A pin on a component that is not connected to anything is said to "float". One can never be sure of what the logic level is on a floating pin, it can be 0, and it can be 1, all depending on the environment around and in the component at that given moment in time.
- **Footprint:** The physical representation of a component's connection pads in a CAD program.
- **Mil(s):** 1 mil = 1/1000th of an inch (0.001"). Not to be confused with mm (millimeter).
- **Net:** The physicall connection between two pins is referred to as a net.

- **Package (IC):** An IC package is the physical chip format of an integrated circuit. See **??** for examples.
- **PHY:** PHY is an abbreviation for the physical layer of the Open Systems Interconnection(OSI) model(wikipedia.org/PHY_(chip)). In practice this referrs to the device/chip that is responsible for providing the interface between the processor and the physical cable being used.
- **Reference Layer/spot**: Signal ground. The layer or spot in the design any given voltage level is relative to, providing a return path to the current for a given signal. The reference is usually not connected to anything, which means it should have a voltage level of 0V.
- **Via:** Electrically conductive hole linking tracks on several layers together.

Bibliography

Altera.AN583. Designing power isolation filters, 2009.

Altera.AN75. High speed board designs, 2001.

ARM. Dui0517b system design reference, 2010.

Ceetus. Function of ethernet magnetics, 2006.

Raymond Y. Chen. Signal integrity, 2001. URL chen@sigrity.com.

CirrusLogic.AN315REV1. Thermal considerations for qfn, 2007.

Joseph Coombs. Using the omap-l132/l138 bootloader sprab41d, 2011.

TI.SLUA271A QFN PCB design. Qfn layout guidelines, 2002.

John Herhberger and Subash Suril. Efficient breakout routing in pcbs, 2006. URL http://www.springerlink.com/content/u418xu70174p6207/abstract/.

J.C-TI. Ddr2 corners in traces, 2012. URL http://e2e.ti.com/support/dsp/omap_applications_processors/f/42/t/169223.aspx.

Dr. Howard Johnson. Who's afraid of the big bad bend? edn magazine, 2000. URL http://www.sigcon.com/Pubs/edn/bigbadbend.htm.

Dr. Howard Johnson and Dr Martin Graham. High speed digital design, a handbook of black magic, 1993. URL http://www.sigcon.com/books/bookHSDD.htm.

Johnson.Ferrite_Beads. Ferrite beads, 2000. URL http://www.sigcon.com/Pubs/edn/ferritebeads.htm.

Hans Klos. Series termination, 2004.

LogicPD. Som-m1 layout, 2010.

Micrel.AN-111. General pcb design guidelines, 2007.

Micrel.AN-139. Pcb design considerations, 2007.

Micron. Ddr2- mt47h512m4,256m8,128m16, 2006.

Dr. Henry Ott. Pcb stack-ups, 2000. URL http://www.hottconsultants.com/techtips/pcb-stack-up-6.

Panasonic. Ethernet pcb layout guide rev.3, 2011a.

Panasonic. Ethernet pcb layout guide, 2011b.

Charles Pfeil. Bga breakout challenges, 2007. URL http://www.mentor.com/products/pcb-system-design/resources/bga-breakouts-routing/upload/bga-breakout-challenges.pdf.

Hugh Robjohns. Understadning impedance, 2003. URL http://www.soundonsound.com/sos/jan03/articles/impedanceworkshop.asp.

ScreamingCircuits.12-11-06. Qfn layout guidelines, 2006.

Mike Shust and Jeffrey Cobb. Understanding ti's routing rule-based ddr timing specification spraav0a, 2008.

SMSC-AN18.6. Ethernet phy layout guidelines, 2008.

SMSC.AN1815. Pcb guidelines for qfn and dqfn packages, 2010.

SMSC.AN186. Layout guideline lan8720, 2008.

T.I. Am335x arm cortex-a8 mpu, 2012. URL http://www.ti.com/lit/ds/symlink/am3359.pdf.

T.I.L-138. Omap-l138 datasheet, 2009. URL http://www.ti.com/lit/ds/symlink/omap-l138.pdf.

TI.SCAA048. Filtering techniques, 2001.

TI.SNLA107. Reducing radiated emissions in ethernet10/100, 2008.

TI.SPRAAR7. Usb2.0 board design and layout guidelines, 2007.

TI.SPRAAV2. Pcb assembly guidelines for pop, 2008.

TI.SPRU655H. Emulation and trace headers, 2012.

TI.SPRUH77. C6-integra technical reference manual, 2011.

TI.Wiki.Target.Connection.Guide. Xds target connection guide, 2012.

Wiki:Characteristic_impedance. Characteristic impedance. URL http://en.wikipedia.org/wiki/Characteristic_impedance.

Wiki:Signal_Reflection. Signal reflection. URL http://en.wikipedia.org/wiki/Signal_reflection.

Wiki:Source-synchronous. Source-synchronous. URL http://en.wikipedia.org/wiki/Source-synchronous.

Man-Fai Yu, Joel Darnauer, and Wayne Wei-Ming Dai. Interchangeable pin routing with application to package layout, 1996. URL http://delivery.acm.org/10.1145/250000/244954/p668-yu.pdf.