Hardware summary

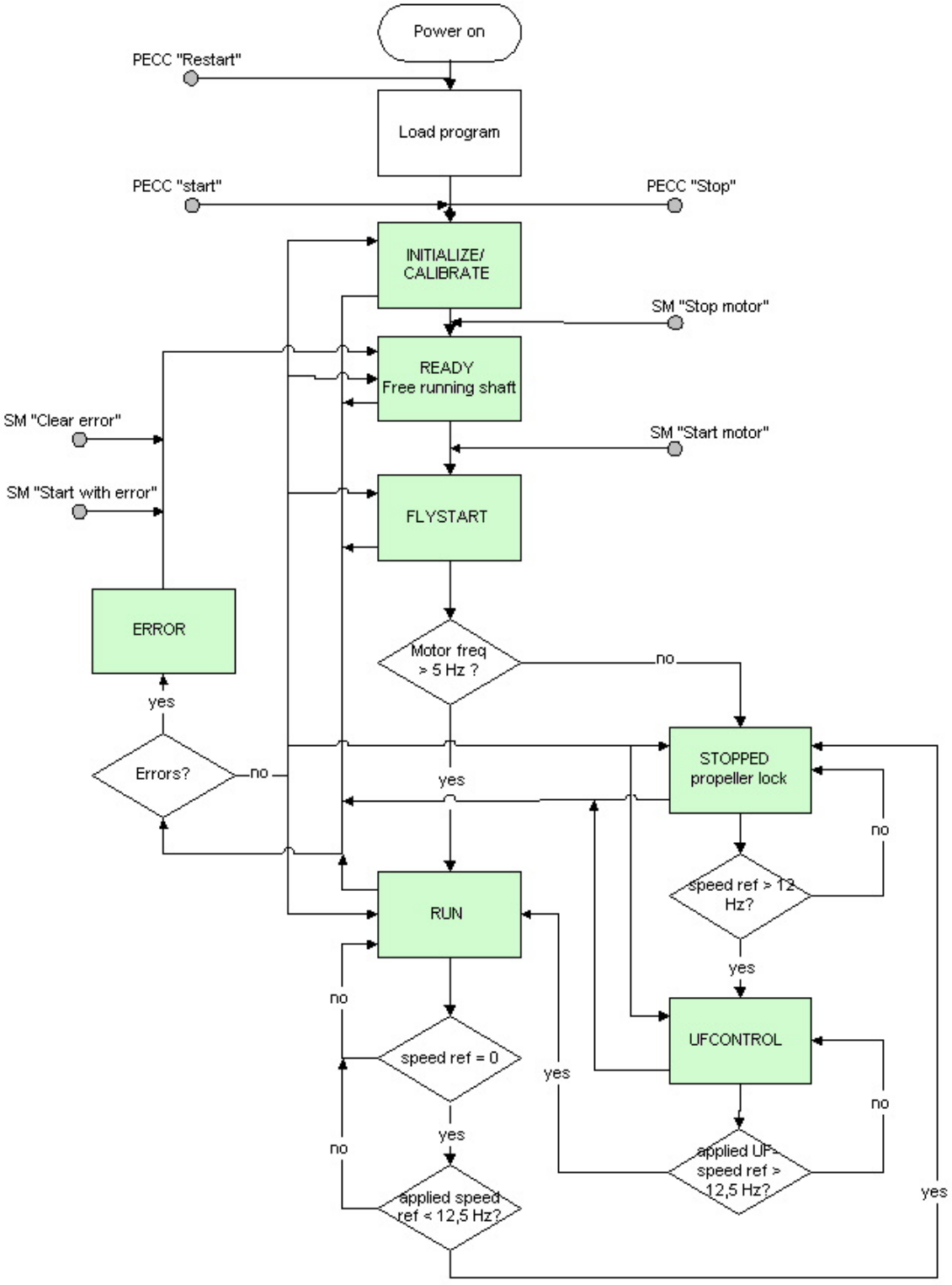
DSP settings

CPU clock	**150 MHz**
Internal flash	256 kB
Internal RAM	36 kB
External Flash	1 MB
External RAM	1 MB
Serial port A	Enabled, 115200 baud
Serial port B	Disabled
CAN	Disabled
Event manager A	Enabled, 75 MHz
Event manager B	Enabled, 75 MHz

State machine and power up sequence.

After a reset or power up of the DSP, certain functions are called. These functions initialize the DSP, verifies the program that has been programmed in to flash, and moves time critical code from flash in to faster internal or external RAM. If the flash memory has been erased, and not reprogrammed (contains only 0xFF), a “program not loaded” error will be given.
After a start command is sent to the DSP, the motor control software is started (if program exists). Before it is possible to run the motor, parameters are initialized from the external EEPROM, and sensors are being calibrated. After the calibration is finished (~0.14 s), the motor software is in the “Ready” state. To start the motor, it is necessary to run the virtual function “Start motor”, after this function has been run, the motor software will try to perform a flystart. If the flystart fails, the motor state will be set to “UFControl”(unless the speed reference is 0, then it will be set to “Stopped”), and the applied speed will follow the reference. When the applied motor frequency is higher than 12 Hz, the motor state is set to “Run”. The motor is now in normal speed regulated operation using flux estimators. The motor will stay in this state until the speed reference is set to zero, or the virtual function “Stop motor” is run. If any errors occur, the motor software will be put in the “Error state”.
If the error disappears, the respective error flag will be cleared after one second. To get the state machine out of “Error state”, one must run the virtual function 3, “Clear error”. If the error endures, one should always try to find the cause, and correct the error before bringing the state machine out of “Error state”. However, it is possible to run virtual function 4, “Start with error”, to mask out the respective error flags, and forcing the state machine back to ready. *SmartMotor cannot take responsibility for the electronics if this function is used, and the error hereby causes damage to equipment.*

In figure 1, the state machine is described graphically. Note that the check for applied UF-speed ref is just and internal variable, and will not be reflected in the applied speed ref read from the controller. Also note that the state machine will not go by Ufcontrol from RUN to STOPPED. This is because the Ufcontrol have no knowledge of the rotor position.


Power on
PECC "Restart"
Load program
PECC "start"
PECC "Stop"
INITIALIZE/
CALIBRATE
SM "Stop motor"
READY
Free running shaft
SM "Clear error"
SM "Start motor"
SM "Start with error"
FLYSTART
ERROR
Motor freq
> 5 Hz ?
no
yes
Errors?
no
yes
STOPPED
propeller lock
no
speed ref > 12
Hz?
yes
RUN
UFCONTROL
no
speed ref = 0
yes
no
yes
applied UF
speed ref >
12,5 Hz?
no
yes
applied speed
ref < 12,5 Hz?

Virtual control functions

The following virtual functions are available in the SW:

Address	Name	Description
0	Start motor	The motor state machine will go from ready to flystart-stopped-ufcontrol-run
1	Stop motor	The motor state machine will go to Ready (except if it is in Error state)
2	Save system log	The system log is programmed in to flash; this function can only be run after the program has been stopped.
3	Clear error	The motor state machine will go from error to ready, provided all error flags are cleared.
4	Start with error	The active error flags will be masked out, and the state machine brought to ready. This is only to be used in an emergency. USE WITH CAUTION!
5	ResetSW	Running this function will cause the software to reset as if the power was reset. Requires the communication to function.

Virtual control variables

All control variables are accessed through the serial communication interface.

Virtual address	Access (**R**ead / **W**rite)	Variable
Motor Control		
0	R/W	Speed reference
1	R	Applied speed reference
2	R	Estimated speed
3	R	Temperature PCB
4	R	Temperature Heat sink
5	R	48 V DC voltage
6	R	15 V DC voltage
7	R	Motor torque
8	R	DC current
9	R	DC Power

Speed reference (Read/Write)

Size: 32 bit, signed
Format: Q.20

This variable is the speed reference in pu Q.20 format.

Range: <-q3.0, q3.0>
Error range: <-∞, -3.5> and <3.5, ∞>

Table 1: Speed reference encoding/decoding

Q.12	Decimal view	RPM
-2.0	-2097152	-353
-1,5	-1572864	-264.75
-1.0	-1048576	-176.5
-0.5	-524288	-88.25
-0.1	-104858	-17.6
0.1	104858	17.6
0.5	524288	88.25
1.0	1048576	176.5
1.5	1572864	264.75
2.0	2097152	353

Applied speed reference (Read)

Size: 32 bit, signed
Format: Q.20
Range: [-q1.45, -q0.24] and [q0.24, q1.45]

This variable is the output from the speed reference ramp function, and the input to the speed regulator. Value's are in the same format as the "Speed reference".

Estimated speed (Read)

Size: 32 bit, signed
Format: Q.20
Range: [-2^12, (2^12-1)]

This variable is the speed estimate from the sensorless speed estimator. The format used is the same as for the Speed reference.

Temperature PCB (Read)

Size: 32 bit, signed
Format: Q.20
Range: [-2^12, (2^12-1)]

This variable is the temperature on the electronics board in Q.20 format, i.e. q30.0=30.0 degree Celsius.

Appendix I - The Q. format

The Q.x format is a standard method for representing floating point numbers in fixed point DSPs and micro controllers.

16 bit numbers:

Q.x means that the lower x bits are used to represent the decimal part of the number, and the remaining (16-x) bits is used to represent the integer part, and the sign bit.

Example:

$$1.45 \text{ in Q.12} = 1.45 \cdot 2^{12} = \underline{\underline{5939}}$$

$$1.45 \text{ in Q.8} = 1.45 \cdot 2^{8} = \underline{\underline{371}}$$

32 bit numbers:

Q.x means that the lower x bits are used to represent the decimal part of the number, and the remaining (32-x) bits is used to represent the integer part, and the sign bit.

Example:

$1.45 \text{ in Q.20} = 1.45 \cdot 2^{20} = \underline{\underline{1520435}}$

$1.45 \text{ in Q.24} = 1.45 \cdot 2^{24} = \underline{\underline{24326963}}$

Notation:
The notation q1.45 in the respective Q.x format means 1.45*2^x. That is: q1.45 in Q.20 format means 1.45*2^20.