

YROTATE-IT-RX220

UM-YROTATE-IT-RX220
Rev.1.00
Jan 15, 2014

Low Cost Motor Control Kit based on RX220

Introduction

The Renesas Motor Control Kit, YROTATE-IT-RX220, is based on the RX220 device from the powerful 32-bit RX microcontroller family.

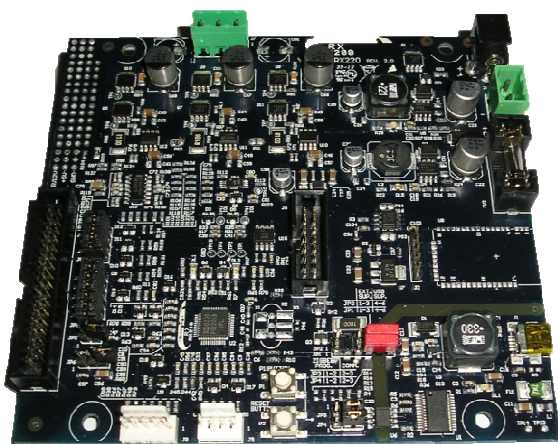
The kit enables engineers to easily test and evaluate the performance of the RX220 in a laboratory environment when driving any 3-phase Permanent Magnet Synchronous Motor (e.g. AC Brushless Motor) using an advanced sensorless Field Oriented Control algorithm. Typical applications for this type of solution are compressors, air conditioning, fans, air extractors, pumps and industrial drives.

The phase current measurement is done via three shunts which offers a low cost solution, avoiding the need for an expensive current sensor. A single shunt current reading method is also available.

The powerful user-friendly PC Graphical User Interface (GUI) gives real time access to key motor performance parameters and provides a unique motor auto-tuning facility.

The hardware is designed for easy access to key system test points and for the ability to hook up to an RX220 debugger. Although the board is normally powered directly from the USB port of a Host PC, connectors are provided to utilise external power supplies where required.

The YROTATE-IT-RX220 is an ideal tool to check out all the key performance parameters of your selected motor, before embarking on a final end application system design.



Target Device: RX220

Contents

1. Hardware overview	3
2. Connectors description	4
3. Power supply selection	5
4. LEDs function description	6
5. Test points for debugging	7
6. Internal power stage brief description	9
7. Interface with an external power stage.....	10
8. Single shunt current reading.....	14
9. Current reading timing in three shunts and single shunt configurations	15
10. Control MCU overview.....	16
11. Permanent magnets brushless motor model	17
12. Sensorless Field Oriented Control algorithm	23
13. Software description.....	24
14. Start-up procedure.....	28
15. Reference system transformations in details	30
16. Rotor position estimation	31
17. Internal representation of physical quantities.....	33
18. EEPROM parameters list	37
19. PC Graphical User Interface	38
20. Motor Auto-calibration using the PC GUI	40
21. List of motors tuned automatically using the PC GUI	50

1. Hardware overview

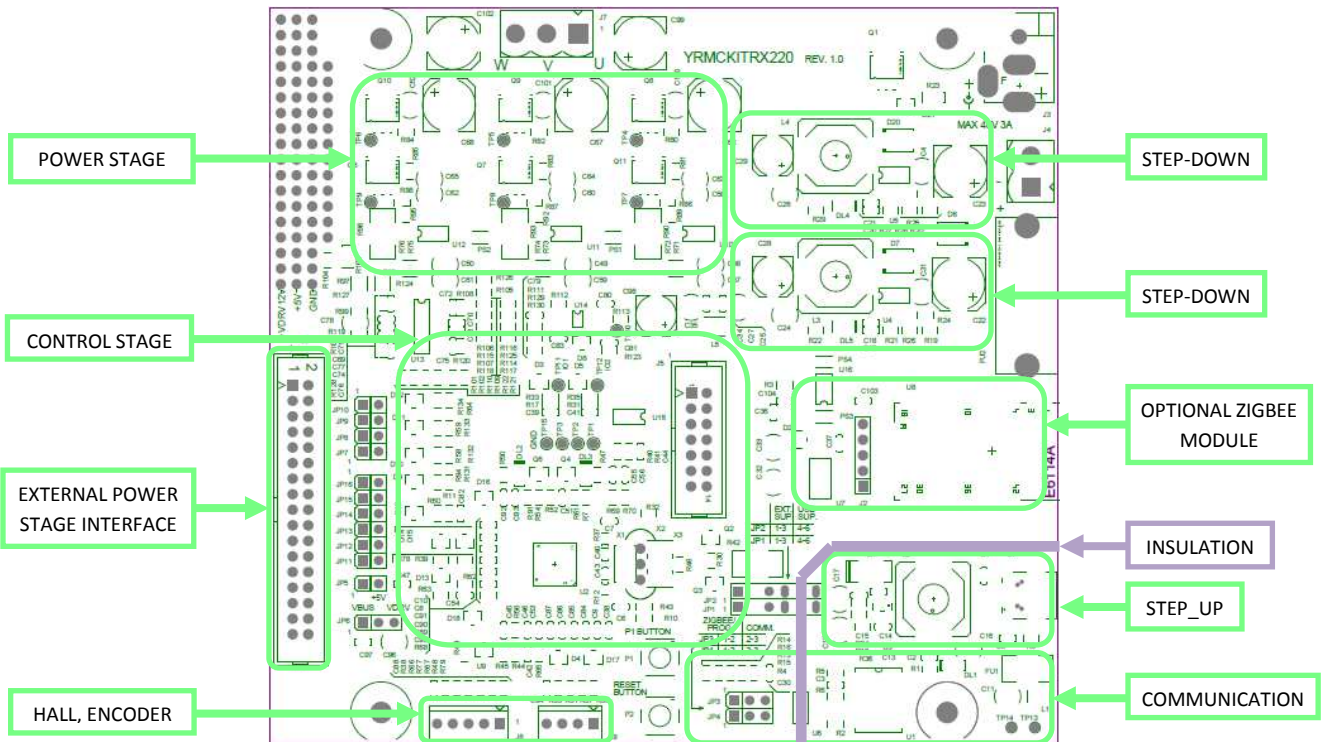
The inverter kit YROTATE-IT-RX220 is a single board inverter, based on the RX series microcontroller RX220 and includes a low-voltage MOSFETs power stage and a communication stage.

To obtain the maximum flexibility, the reference kit includes:

- A complete 3-phase inverter on-board with a low voltage motor, so it becomes easy to test the powerful sensorless algorithm running on the RX220 microcontroller.
- An insulated USB communication with the PC.
- Connectors for hall sensors and encoder connections (both encoder and hall sensors are not managed in the sensorless software but they can be supported under request).
- Compatibility with the existing Motor Control Reference Platform external power stage.
- USB supply possibility to avoid external power supplies (where galvanic insulation is lost).

To achieve these aims, three different DC-DC converters are used:

1. A step-up DC-DC converter to increase the voltage from the USB standard (5V) up to 13.5V_{DC}
2. A step-down converter from the DC bus (which can reach up to 48V_{DC} in case of external supply) to the power switches drive voltage (12V).
3. A step-down converter from the DC bus to the CPU supply voltage (5V).

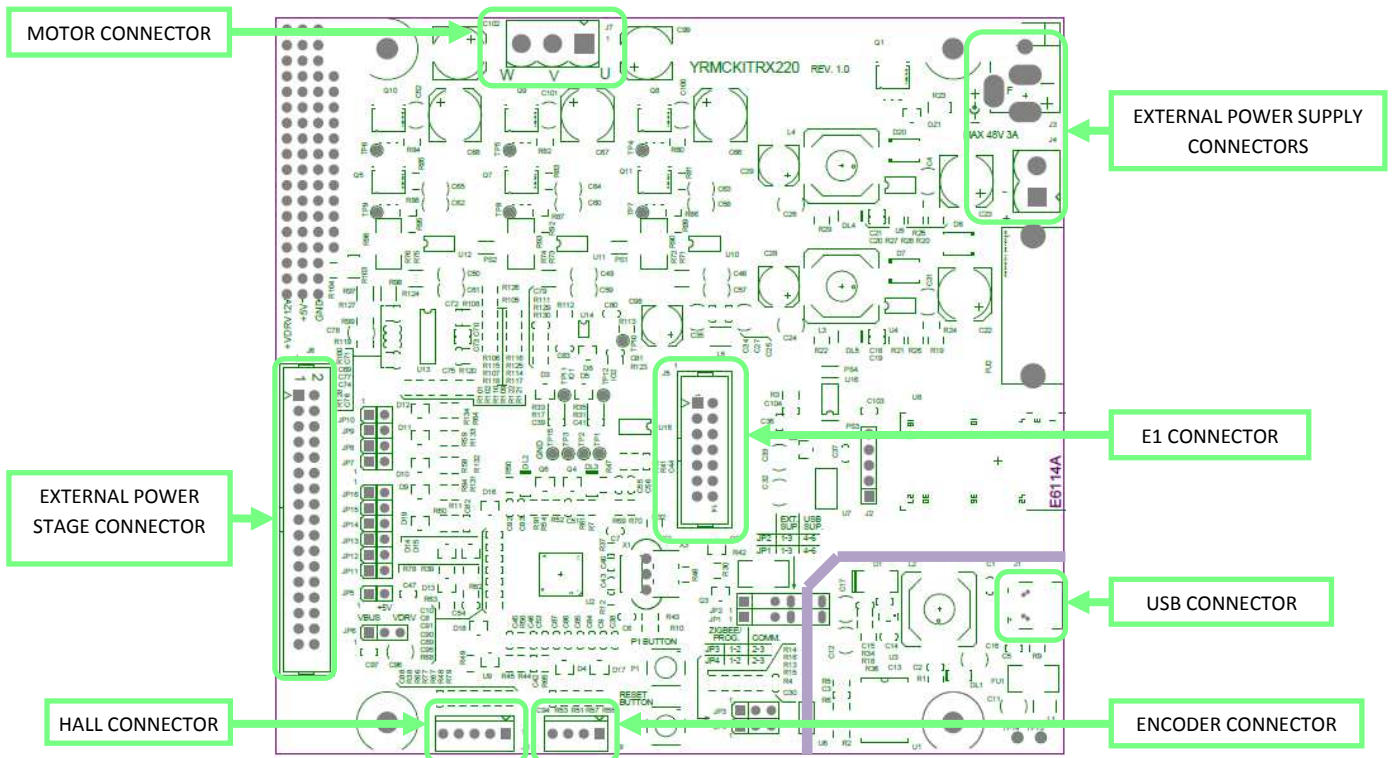


2. Connectors description

As in the following figure, you can find the position and the description of the connectors present on the board. Please refer to the board schematics for the full description of the connectors.

The E1 connector is used for the programming and the debugging of the software running on the RX220.

The external power stage connector is compatible with the power stages, designed for Renesas inverter kits, which are able to drive 230V_{AC} motor up to 1.5KW.



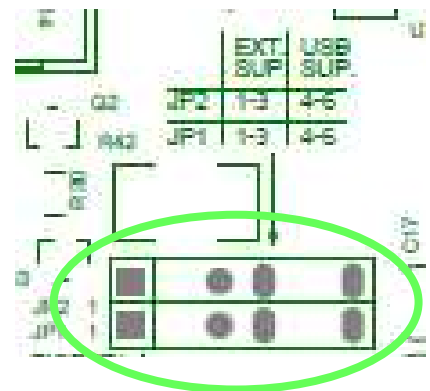
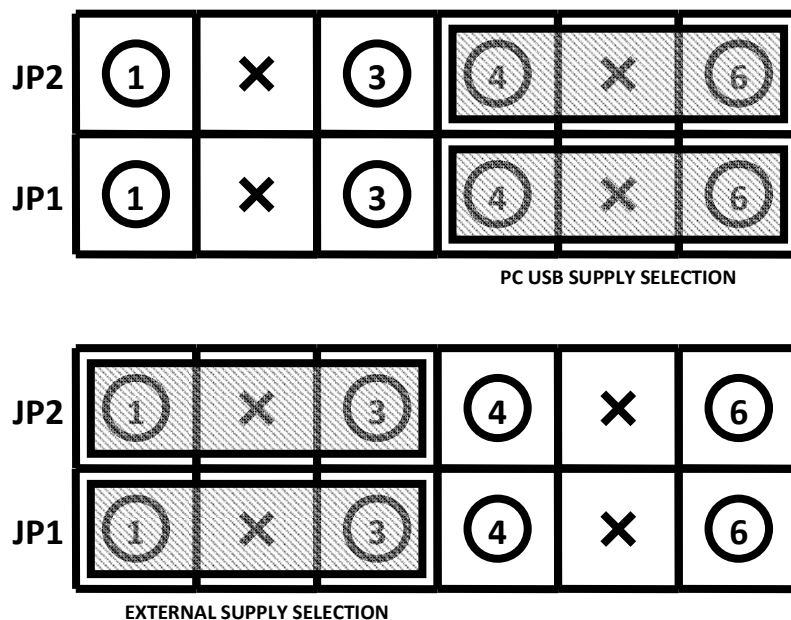
3. Power supply selection

As stated before, there are two ways to supply power to the board.

1. The first possibility is to use directly the PC USB supply. In this case the current you can give to the motor is limited by the USB current capabilities.
2. The second possibility is to use an external voltage DC source to supply the board.

The recommended voltage value is between 12V_{DC} and 48V_{DC}. In this case the communication stage is insulated from the inverter.

The selection between the two possibilities is made through two jumpers: **JP1** and **JP2**. Please find below the description:



- 1) The first jumper configuration connects the USB ground to the inverter ground and the output of the step-up converter to the inverter DC link.

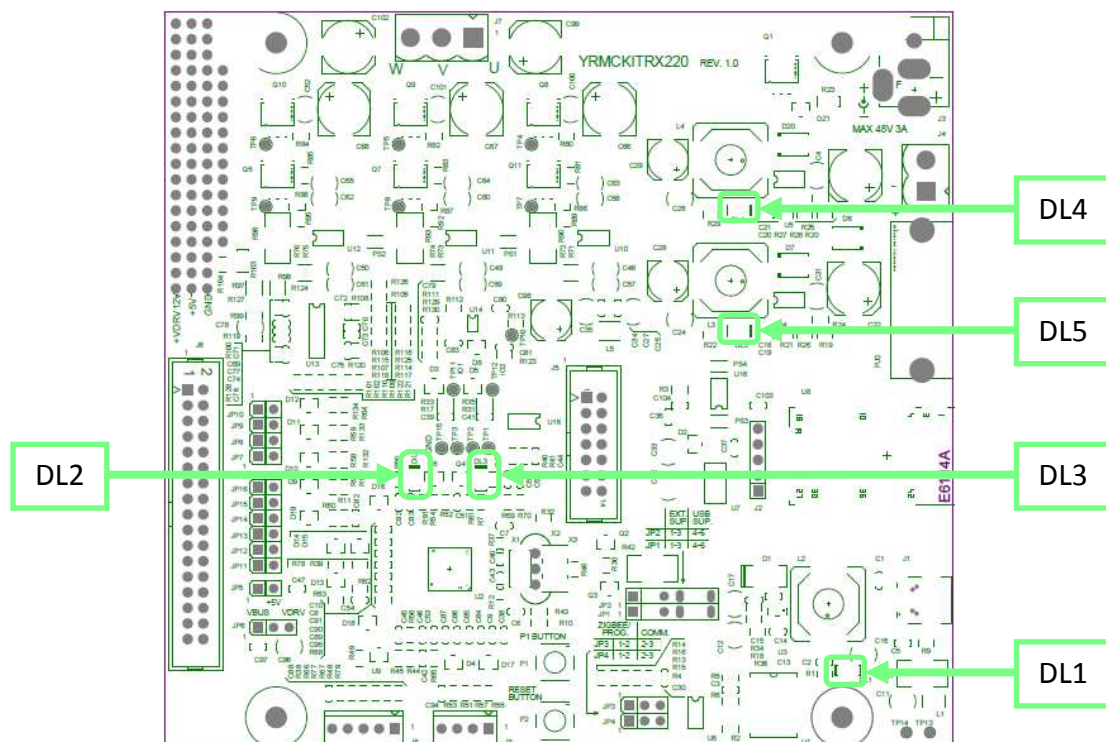
Please notice that in this case there is no galvanic insulation between the device connected to the USB and the board.

- 2) The second jumper configuration connects the external power supply ground to the inverter ground and the external + V_{DC} to the inverter DC link.

4. LEDs function description

Some of the LEDs available on the board are directly connected to the hardware and allow the user to understand the status of the board. Please refer to the LED map for the following indications:

- DL4 is connected to the output of the 12V step-down DC-DC converter and indicates the presence of the switches drive supply;
- DL5 is connected to the output of the 5V step-down DC-DC converter and indicates the presence of the logic power supply;



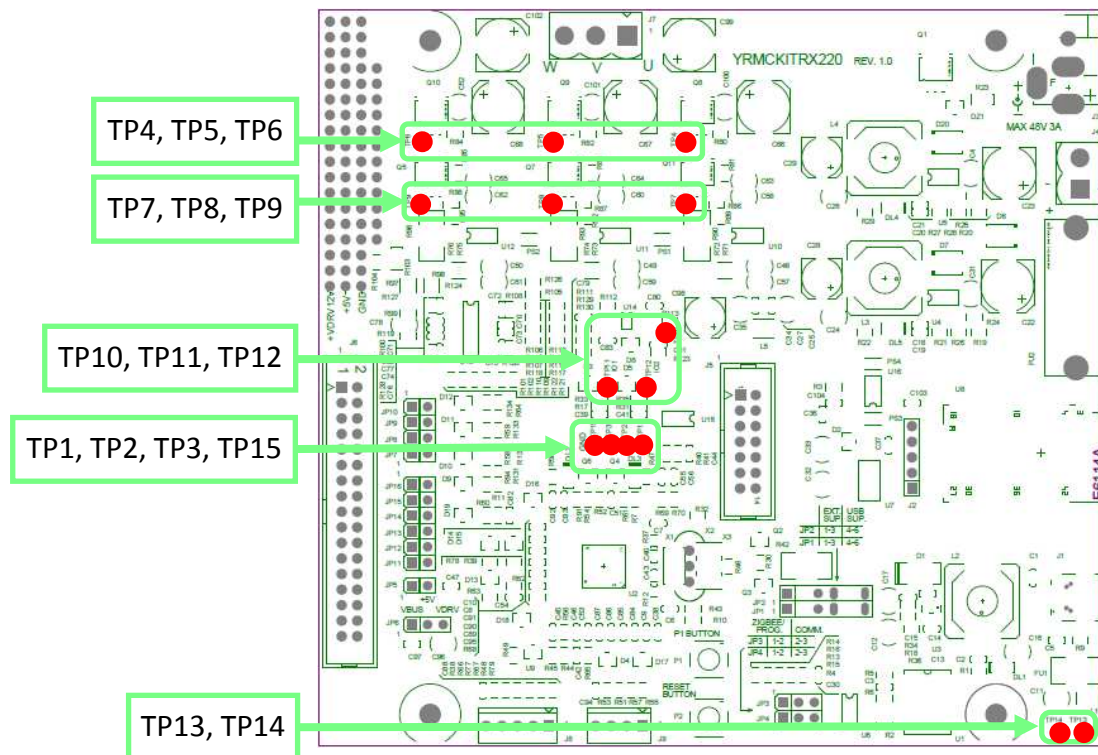
Other LEDs in the board are driven via software, in particular:

- DL1 is the USB communication indicator and blinks when there are data exchanges between the PC and the board.
- DL2 is free for the user (in the default software it is on when the main interrupt is active).
- DL3 is blinking slowly if the control section MCU (RX220) is running normally. In case of hardware or software alarms, the LED DL3 is blinking quickly.

5. Test points for debugging

Several specific test points are available on the board to visualize with the oscilloscope the behavior of some internal analog signals. Furthermore, it is possible to visualize internal variables as analog waveforms using filtered PWM outputs.

Finally, it is very useful during the tuning process for adapting the software to a new motor to use the test points.



Please find below the description of the test points:

- **TP13, TP14:** are connected to the two USB communication signals, for debug purposes. Please refer to the board schematics for more details.
- **TP4, TP5, and TP6:** they are connected to the three output of the inverter (sources of the higher switches).
- **TP7, TP8, and TP9:** they are connected to the sources of the lower switches of the inverter.
- **TP10** is connected to the hardware over-current signal from the power stage.
- **TP11, TP12:** they are connected to two digital ports of the microcontroller (TP11 → Port3Bit5, TP12 → Port1Bit7) and they can be used as digital outputs or digital inputs.
- **TP15** is connected to the board ground.

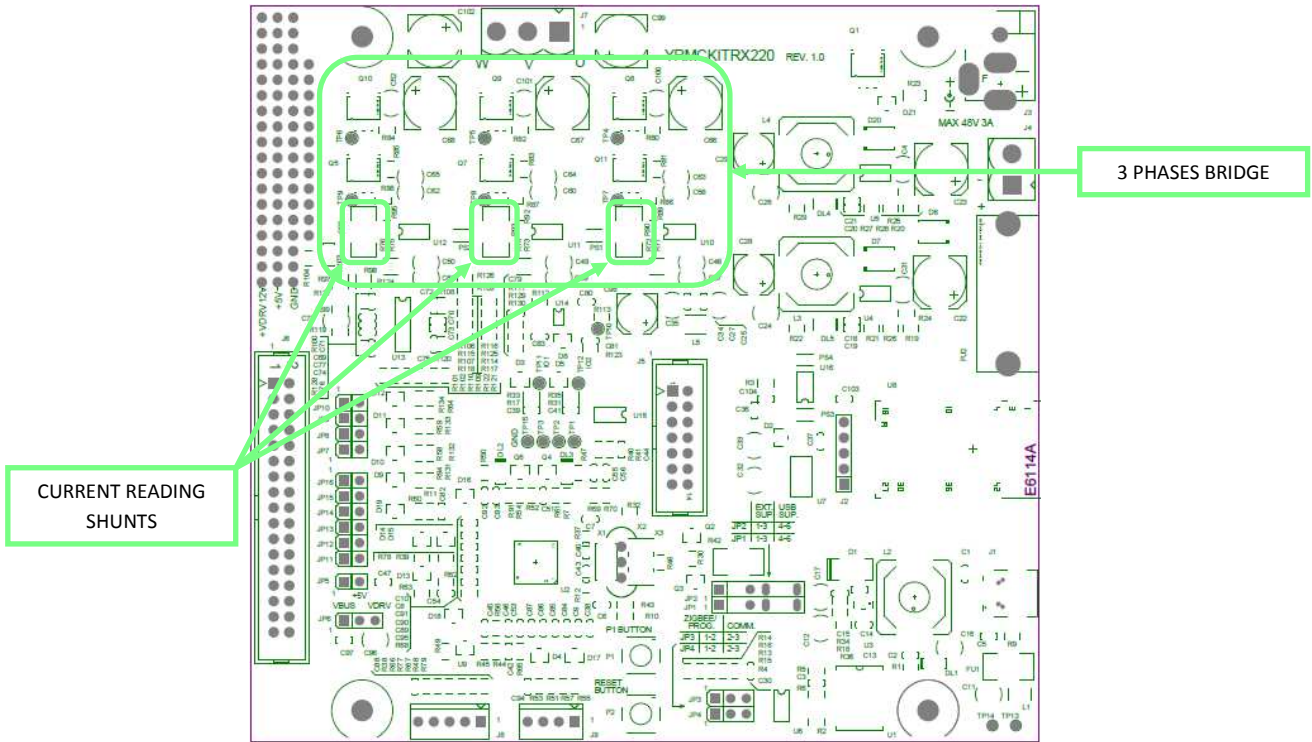
- **TP1, TP2, TP3** are three filtered PWM outputs which can be used to visualize the behavior of internal variables.
 - In the standard software TP1 is the output for the system phase, e.g. estimated rotor position
 - TP2 is used in the current PI tuning facility to show the normalized response of the system to a current step request.
 - TP3 is not used in the standard software.

6. Internal power stage brief description

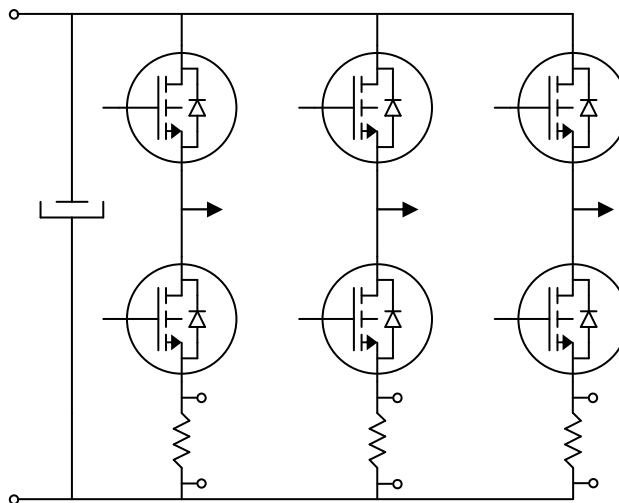
The power stage is a complete 3-phase bridge composed with discrete low voltage power MOSFETs.

The MOSFETs are the Renesas **RJK0654DPB** n-channel power MOSFETs. Please refer to the data-sheet for the switches characteristics and to the board schematics for the details on the driving circuit.

The maximum current is 30A, and the maximum voltage is 60V



The inverter has the classical schema with the three shunts on the lower arms:



7. Interface with an external power stage

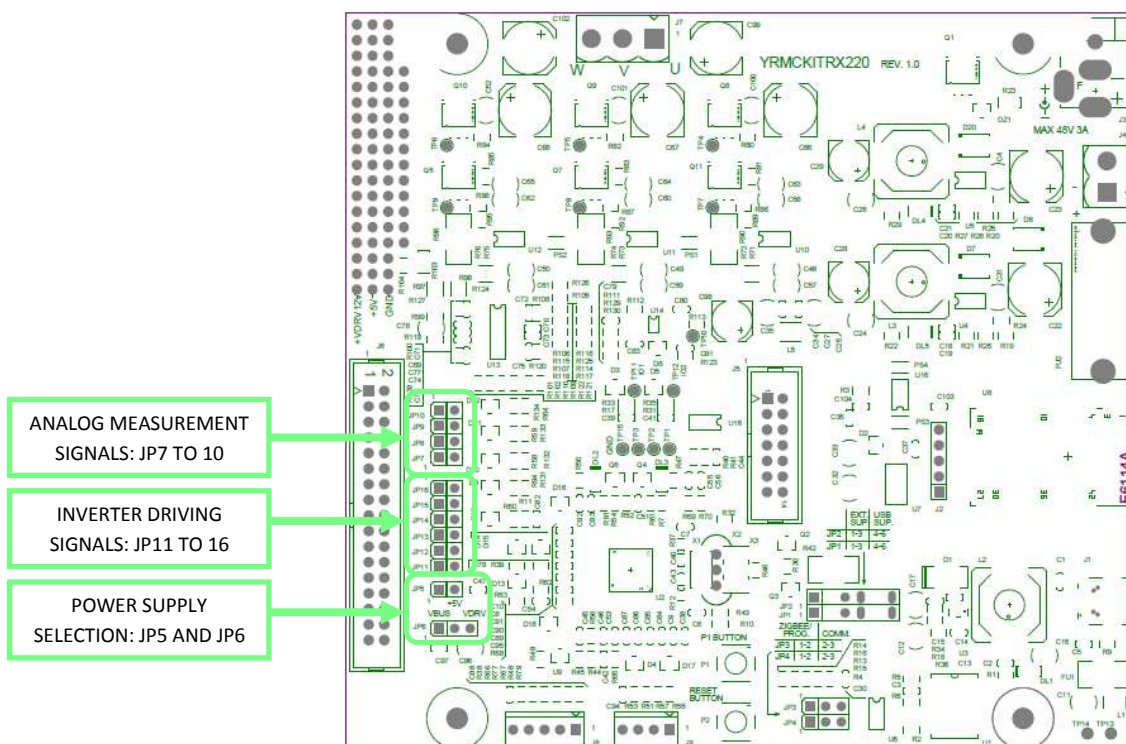
Since internal power stage allows only the management of low voltage motors, an interface with an external power stage has been developed.

The selection between the internal power stage and the external power stages is ensured by jumpers. It is a safe way to ensure that the right voltage and current signals are active.

When the external power stage is connected to the kit, it is by default the **active** one. So the microcontroller pins are directly connected to the external power stage connector.

In this case the internal power stage should be disconnected, and this must be done by disconnecting the appropriate jumpers.

Please find below the jumpers description.



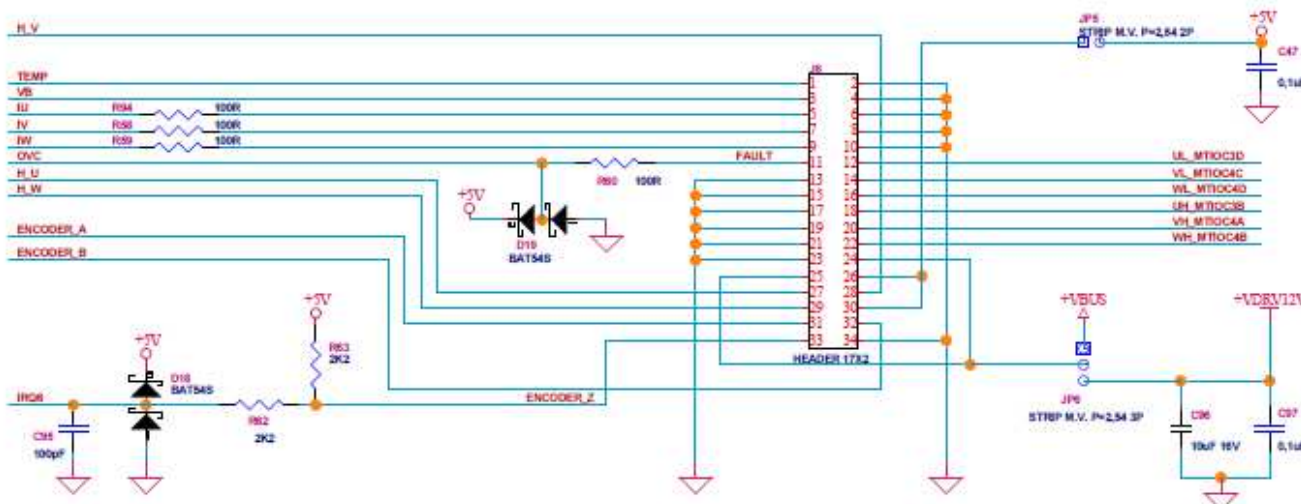
- **JP7:** if closed, then the internal power stage DC Bus voltage is connected to the opportune A/D converter pin.
- **JP8, JP9, and JP10:** if closed, then the internal power stage currents measurements (U, V, W) are connected to the opportune A/D converter pins.
- **JP11 to JP16:** if closed then the inverter driving signals are connected to the internal power stage drivers.
- **JP5:** if closed, the external power stage 5V is connected with the internal one.
- **JP6:** it allows the following selection:

- if pins 1 and 2 are shorted, then the external power stage low voltage supply (between 15V to 24V) is connected to the internal DC Bus Voltage; in this case both the step-down converters of the board will work;
- If pins 2 and 3 are shorted, then the external power stage low voltage supply (between 15V to 24V) is connected to the internal driving circuit supply; in this case only the step-down converter which generates the internal logic supply will work.

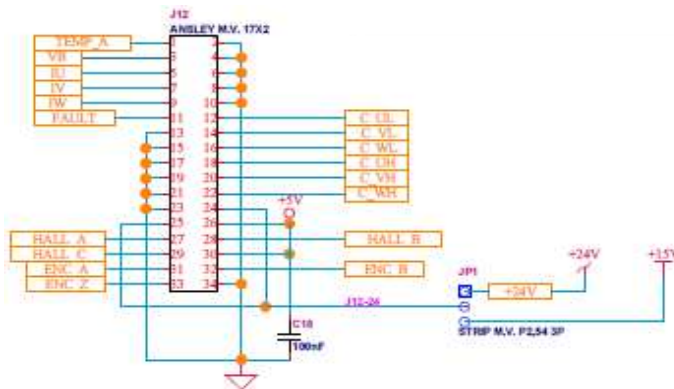
Please be careful to take into account the following precautions:

1. Please avoid to connect both the external power stage connector (J6) and to close the jumpers JP5 to 16: this would produce short circuits between signals coming from different sources.
2. In JP6, chose the configuration with the pins 1 and 2 shorted, while pin 3 is open, in order to avoid supplying some internal circuits with a too high voltage.

Please find below the drawing of the interface connector.



For a comparison, find below the drawing of the corresponding connector in the MCRP07 external power stage (E6108A).

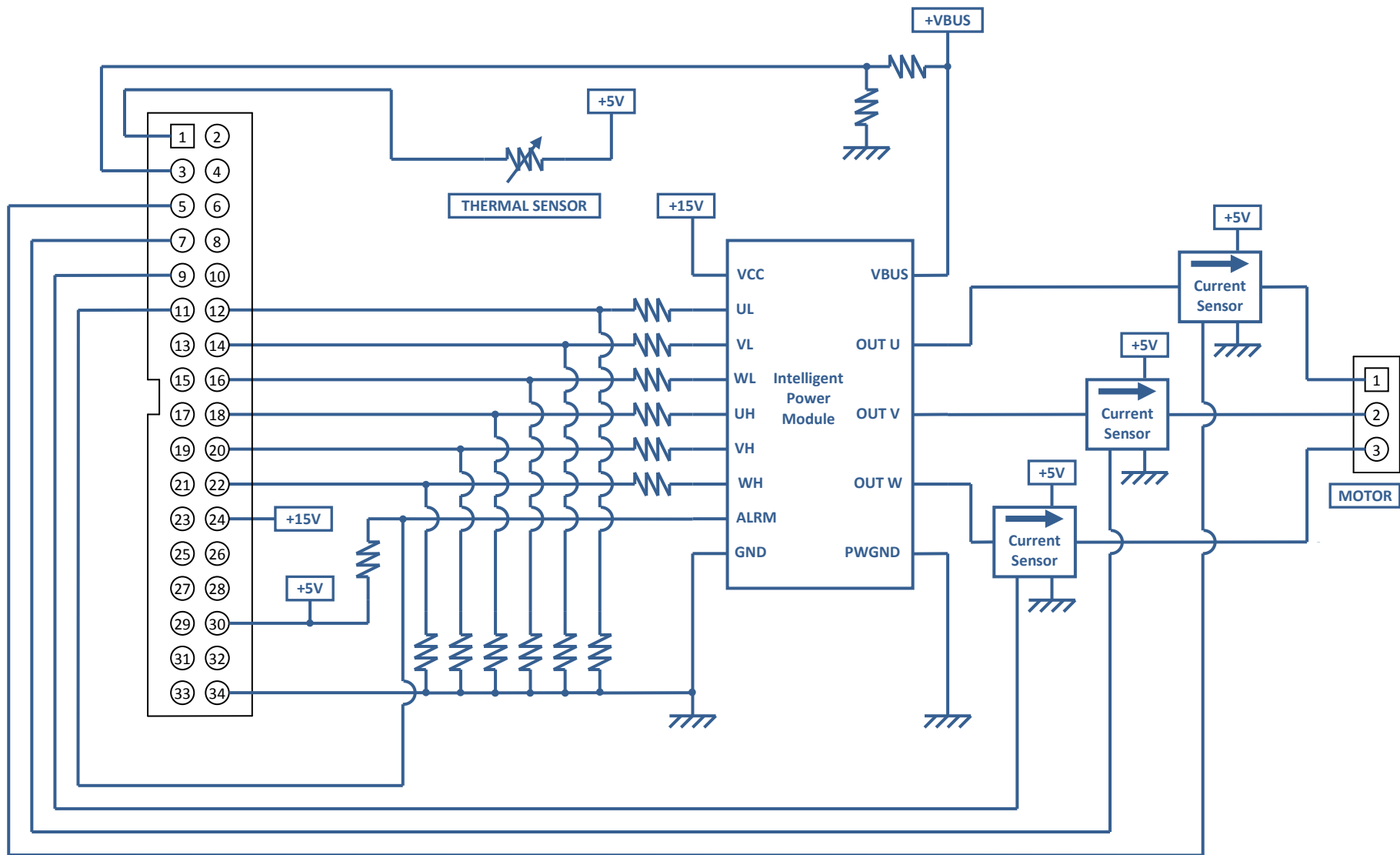


If using a different external power stage, please keep present the following notes:

- a) The PWM drive signals are directly connected to the microcontroller output pins, and there is no pull-up or pull-down resistor connected, so the polarization has to be done in the power stage. In case of alarm, the microcontroller output pins can be placed in high impedance state, so the external polarization is necessary. These output commands are logic level signals, with limited current output capability, so an external driver is probably required. A further line is connected to the microcontroller: it is the external alarm signal, connected to the POE input pin; this pin is polarized with a 10K pull-up toward the logic supply.
- b) The analog measurement signals from power stage, in particular the current readings and the DC link voltage reading are clamped (with diodes from logic GND and to logic V_{cc}) and weakly filtered, then directly connected to the A/D converter input pins of the microcontroller, so the external power stage has to take care of the gain and the offset of these signals.
- c) The ground connection is always active, and it represents the reference for all the interface signals.

In the next figure a simple example regarding how the power board connections have to be arranged is presented. In this schema it is supposed that the power board has its own supply for the power module (+15V); this supply is also used to supply the microcontroller (thanks to jumper JP5 in microcontroller board); the 5V supply for current sensors and for the signal polarization is derived from the microcontroller board (through JP5).

Please refer to the complete schematics for further details.



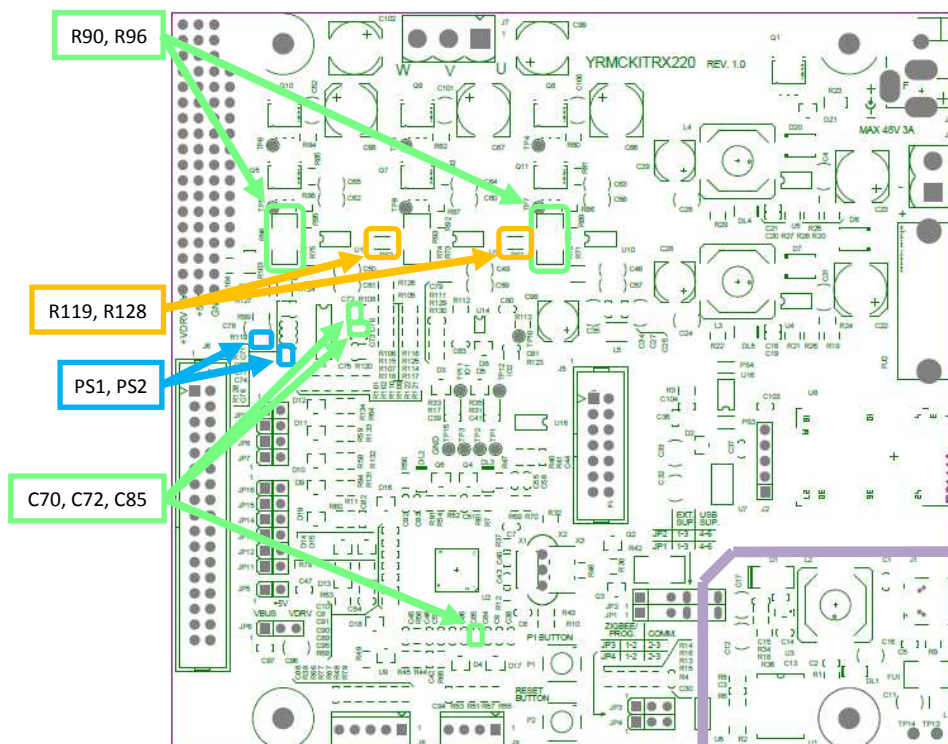
8. Single shunt current reading

While the normal configuration of the board and the standard software are based on three shunts current reading, we also offer the possibility to configure the board for single shunt current reading. Some hardware modifications are required, and a different software version has to be loaded.

The required hardware modifications are the following (please refer to the board schematics):

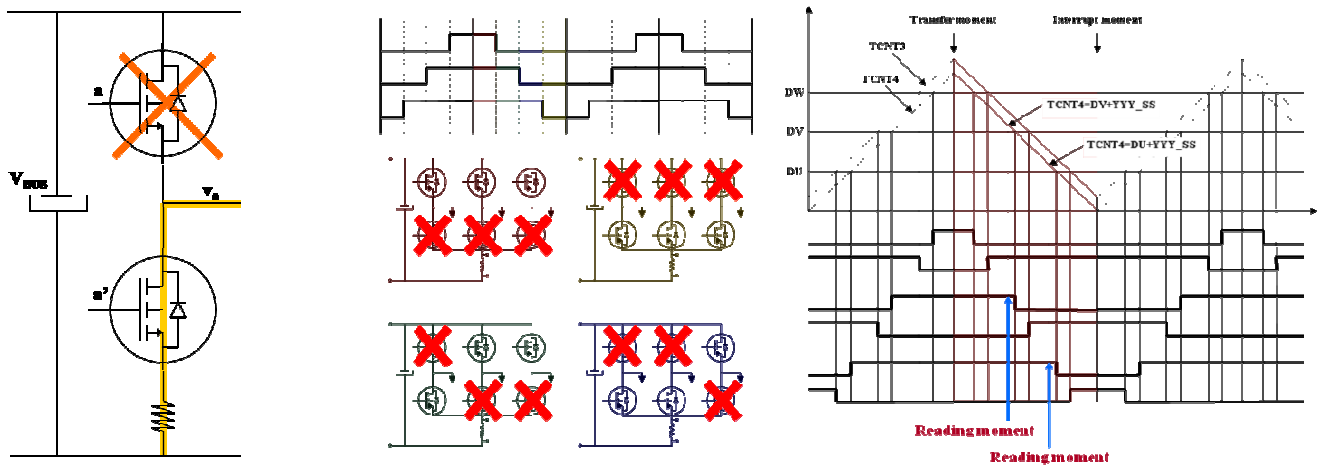
- Remove from the board R90 and R96 (they are the shunts related to the phases U and W).
- Close the soldering points PS1 and PS2 (those soldering points put the three inverter harms in common, below the lower switches and above the shunts).
- Change the value of R119 and R128 from 3K3 to 1K (this will adapt the gain of the over-current detection circuit).
- Remove the capacitors C70, C72, C85 (it speeds up the current reading circuit).

The components involved in the modifications are indicated in the figure below.



9. Current reading timing in three shunts and single shunt configurations

The figures below show the different situations related to the two configurations. The first figure is related to three shunts current reading, the other are related to the single shunt current reading.



Three shunts

In the three shunts configuration the current in one shunt is equal to the corresponding phase current when the corresponding lower switch is ON.

The most suitable moment to read the current in this configuration is at the trough of the PWM.

Single shunt

In the single shunt configuration, only when one or two of the lower switches are ON the current through the shunt is related with the phase current.

When only one of the lower switches is ON, the current in the shunt is equal to the current of the corresponding inverter phase.

When two of the lower switches are ON, the current in the shunt is equal to the sum of the currents of the corresponding phases that is it is minus the current of the third phase.

Important Note:

Software projects for each method are provided for both the High-performance Embedded Workshop (HEW) and e2studio development environments.

The appropriate project for each are listed below.




Three shunts current for HEW: [MCRP09_RX220_3_shunts_verXXX](#)
 Single shunt current for HEW: [MCRP09_RX220_1_shunt_verXXX](#)

Three shunts current for e2studio: [YROTATE-IT-RX220_3_shunts_verXXX](#)
 Single shunt current for e2studio: [YROTATE-IT-RX220_1_shunt_verXXX](#)

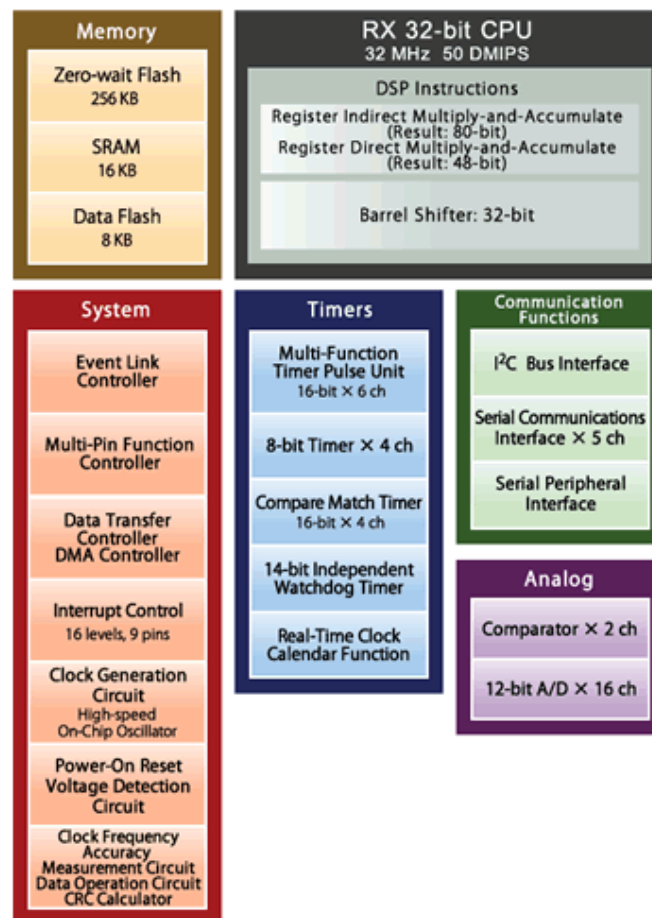
10. Control MCU overview

The RX220 Group is a low power entry-level 32-bit microcontroller. The RX220 Group of products is the slimmed-down version of the higher model RX210 Group, and has very high compatibility in terms of pin arrangement and on the software level.

The main specifications of the RX220 microcontrollers are as follows.

Item	Description
CPU	Max. 32 MHz (1.56 DMIPS/MHz, max. approx. 50 DMIPS)
Operating voltage	1.62 V to 5.5 V
Memory	ROM: 32 to 256 KB, RAM: 4 to 16 KB, data flash: 8 KB
 Safety functions	Clock frequency accuracy measurement circuit (CAC), data operation circuit (DOC), independent watchdog timer (IWDtA), ADC self-diagnostic function/disconnection-detection function, CRC calculator (CRC)
Peripheral functions	 Event link controller (ELC),  multifunction pin controller (MPC), three-phase motor control timer (MTU2), 12-bit A/D converter (S12ADb)
Pin count/packages	LQFP 48-pin to 100-pin packages, small LGA, wide pin pitch LQFP (planned)

Please find below the block diagram of the RX220 microcontroller:



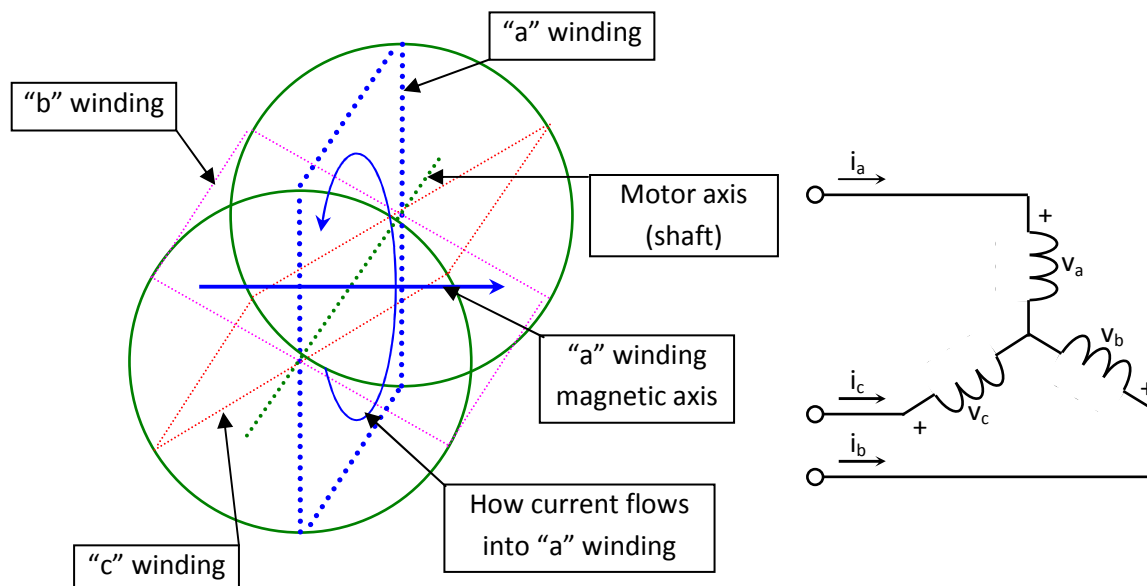
*The maximum specifications for the group are shown.

11. Permanent magnets brushless motor model

The synchronous permanent magnets motor (sinusoidal brushless motor) is widely used in the industry. More and more home appliance makers are now using such brushless motor, mainly because of the intrinsic motor efficiency.

The permanent magnet motor is made with few components:

1. A *stator* formed by stacking sheared metal plates where internally the copper wiring is wound, constructing the stator winding
2. A *rotor* in which permanent magnets are fixed
3. Two covers with ball bearings that keep together the stator and the rotor; the rotor is free to rotate inside the stator



The working principle is quite simple: if we supply the motor with a three-phase system of sinusoidal voltages, at constant frequency, in the stator windings flow sinusoidal currents, which create a rotating magnetic field.

The permanent magnets in the rotor tend to stay aligned with the rotating field, so the rotor rotates at synchronous speed.

The main challenge in driving this type of motor is to know the rotor position in real-time, so mainly implementation are using a position sensor or a speed sensor.

In our implementation, the system is using either one or three shunts to detect the rotor position in real-time.

Let's analyse the motor from a mathematic point of view.

If we apply three voltages $v_a(t)$, $v_b(t)$, $v_c(t)$ to the stator windings, the relations between phase voltages and currents are:

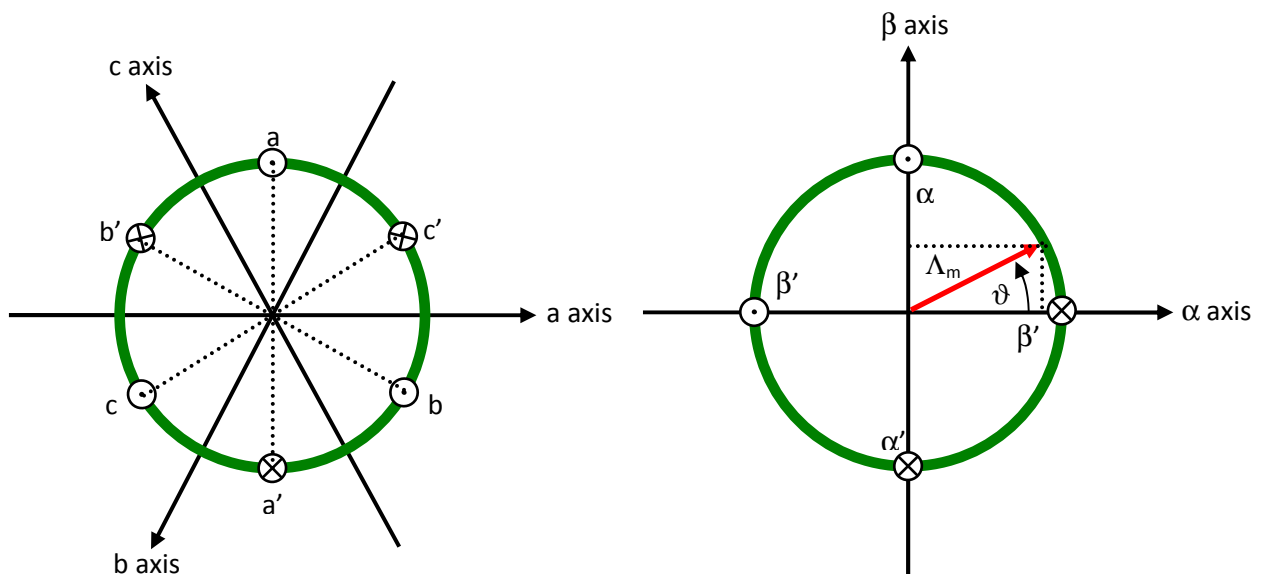
$$v_a = R_S i_a + \frac{d\lambda_a}{dt}$$

$$v_b = R_S i_b + \frac{d\lambda_b}{dt}$$

$$v_c = R_S i_c + \frac{d\lambda_c}{dt}$$

- λ_i is the magnetic flux linkage with the i -th stator winding
- R_S is the stator phase resistance (the resistance of one of the stator windings)

The magnetic flux linkages λ_i are composed by two items, one due to the stator currents, one to the permanent magnets.



Real axes (a, b, c) and equivalent ones (α , β); a fixed amplitude vector can be completely determined by its position respect the (α , β) system (angle ϑ)

The permanent magnet creates a magnetic field that is constant in amplitude and fixed in position in respect to the rotor. This magnetic field can be represented by vector Λ_m whose position in respect to the stator is determined by the angle ϑ between the vector direction and the stator reference frame.

The contribution of the permanent magnets in the flux linkages depends on the relative position of the rotor and the stator represented by the mechanical-electric angle ϑ .

It is, in every axis, the projection of the constant flux vector Λ_m in the direction of the axis:

$$\lambda_a = Li_a + \Lambda_m \cos(\vartheta)$$

$$\lambda_b = Li_b + \Lambda_m \cos(\vartheta - 2\pi/3)$$

$$\lambda_c = Li_c + \Lambda_m \cos(\vartheta - 4\pi/3)$$

Supposing that the rotor is rotating at constant speed ω (that is: $\vartheta(t) = \omega t$) the flux linkages derivatives can be calculated, and we obtain:

$$v_a = R_S i_a + L \frac{di_a}{dt} - \omega \Lambda_m \sin(\vartheta)$$

$$v_b = R_S i_b + L \frac{di_b}{dt} - \omega \Lambda_m \sin(\vartheta - 2\pi/3)$$

$$v_c = R_S i_c + L \frac{di_c}{dt} - \omega \Lambda_m \sin(\vartheta - 4\pi/3)$$

A “three phases system” may be represented by an equivalent “two phases system”. So the by using specific transformations, our three equations system is equivalent to a two equations system. It is basically a mathematical representation in a new reference coordinates system.

In the two phases (α, β) fixed system the above equations become:

$$v_\alpha = R_S i_\alpha + \frac{d\lambda_\alpha}{dt}$$

$$v_\beta = R_S i_\beta + \frac{d\lambda_\beta}{dt}$$

For the magnetic field equations, we got:

$$\lambda_\alpha = Li_\alpha + \lambda_{cm} = Li_\alpha + \Lambda_m \cos(\vartheta)$$

$$\lambda_\beta = Li_\beta + \lambda_{\beta m} = Li_\beta + \Lambda_m \sin(\vartheta)$$

After performing the derivation:

$$\frac{d\lambda_\alpha}{dt} = L \frac{di_\alpha}{dt} - \omega \Lambda_m \sin(\vartheta) = L \frac{di_\alpha}{dt} - \omega \lambda_{\beta m}$$

$$\frac{d\lambda_\beta}{dt} = L \frac{di_\beta}{dt} + \omega \Lambda_m \cos(\vartheta) = L \frac{di_\beta}{dt} + \omega \lambda_{cm}$$

Finally, we obtain for the voltages in (α, β) system:

$$v_{\alpha} = R_S i_{\alpha} + L \frac{di_{\alpha}}{dt} - \omega \lambda_{\beta m}$$

$$v_{\beta} = R_S i_{\beta} + L \frac{di_{\beta}}{dt} + \omega \lambda_{\alpha m}$$

A second reference frame is used to represent the equations as the frame is turning at the rotor speed. So the “d” axis is chosen in the direction of the magnetic vector Λ_m , and with the “q” axis orthogonal to the “d” axis. The new reference system is (d, q).

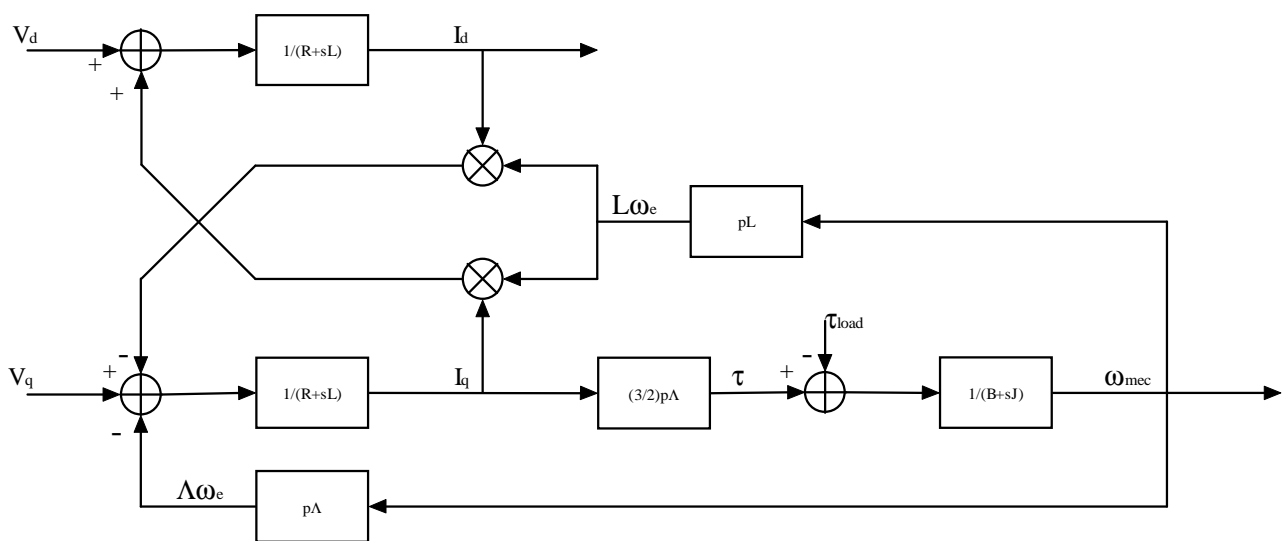
The reference frame transformations from the (α, β) system to the (d, q) system depends on the instantaneous position angle ϑ

So we obtain two inter-dependant equations in the (d, q) system:

$$v_d = R_S i_d + L \frac{di_d}{dt} - \omega L i_q$$

$$v_q = R_S i_q + L \frac{di_q}{dt} + \omega L i_d + \omega \Lambda_m$$

These two equations represent the mathematical motor model.



A control algorithm which wants to produce determined currents in the (d, q) system must impose voltages given from the formulas above.

This is ensured by closed loop PI control on both axis “d” & “q” (Proportional Integral).

Since there is a mutual influence between the two axes, decoupling terms can be used.

In the block scheme the mechanic part is included, where “p” is the number of pole pairs, while “B” represents friction, “J” the inertia, “ τ_{load} ” the load torque and “ τ ” the motor torque.

$$\tau = \frac{3}{2} \times p \times \Lambda$$

The angular speed ω is represented in the scheme as ω_e to distinguish the electrical speed from the mechanical one.

Let's now consider the equations we have seen in (α, β) system:

$$v_\alpha = R_s i_\alpha + \frac{d\lambda_\alpha}{dt}$$

$$v_\beta = R_s i_\beta + \frac{d\lambda_\beta}{dt}$$

These equations show that magnetic flux can be obtained from applied voltages and measured currents simply by integration:

$$\lambda_\alpha = \lambda_{\alpha 0} + \int_0^t (v_\alpha - R_s i_\alpha) dt$$

$$\lambda_\beta = \lambda_{\beta 0} + \int_0^t (v_\beta - R_s i_\beta) dt$$

Furthermore:

$$\Lambda_m \cos(\vartheta) = \lambda_\alpha - Li_\alpha$$

$$\Lambda_m \sin(\vartheta) = \lambda_\beta - Li_\beta$$

If the synchronous inductance L is small, the current terms can be neglected, if not they have to be considered. In general:

$$x = \Lambda_m \cos(\vartheta) = \lambda_\alpha - Li_\alpha = \lambda_{\alpha 0} + \int_0^t (v_\alpha - R_s i_\alpha) dt - Li_\alpha$$

$$y = \Lambda_m \sin(\vartheta) = \lambda_\beta - Li_\beta = \lambda_{\beta 0} + \int_0^t (v_\beta - R_s i_\beta) dt - Li_\beta$$

So in the (α, β) system phase we obtain from the flux components:

$$\vartheta = \arctan\left(\frac{x}{y}\right)$$

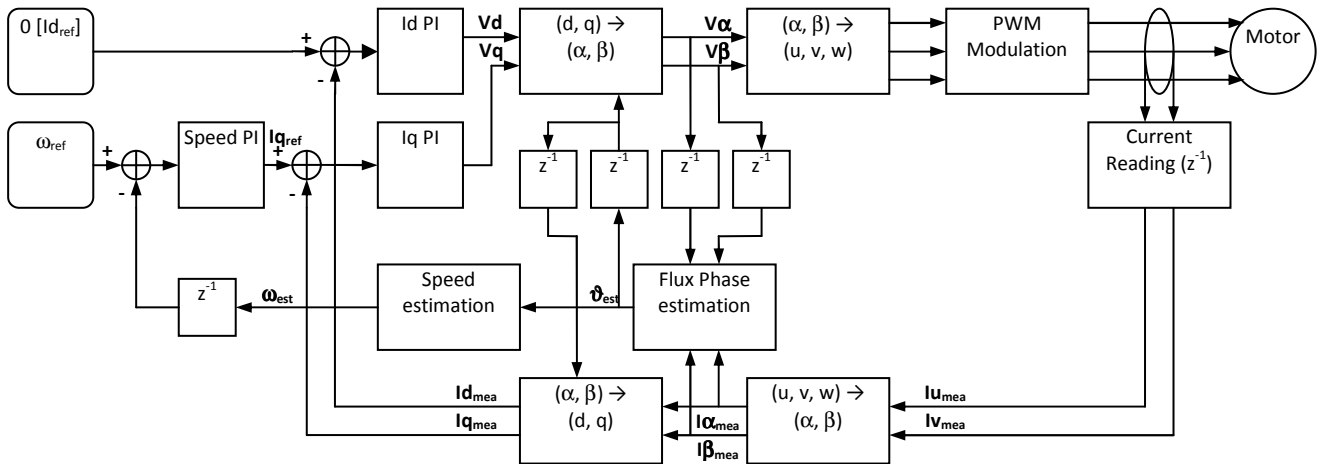
The system speed ω can be obtained as the derivative of the angle ϑ .

$$\omega = \frac{d}{dt} \vartheta(t)$$

Based on this, a sensorless control algorithm was developed to give the imposed phase voltages, to measure phase currents, to estimate the angular position ϑ and finally the system speed.

12. Sensorless Field Oriented Control algorithm

Please, find below the sensorless vector control algorithm block diagram.

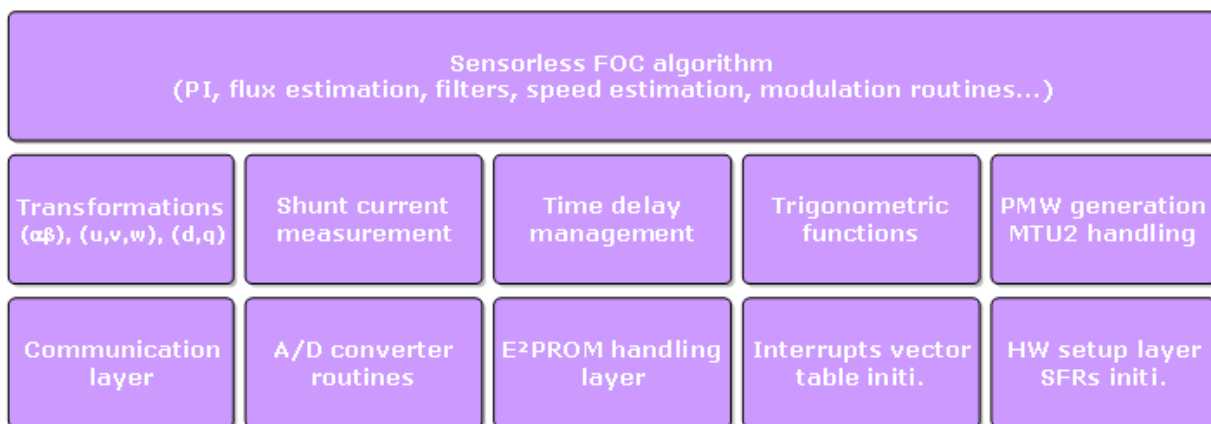


The main difference between the three shunts configuration and the single shunt one is in the "Current Reading" block, the rest of the algorithm remains the same in principle, even if the blocks order has been adjusted.

13. Software description

On YROTATE-IT-RX220 the software previously described is working on RX220 clocked at 32MHz. Using the interrupt skipping function it is possible to regulate separately the PWM frequency and the sampling frequency, allowing the user to add additional tasks to the motor control.

SOFTWARE BLOCKS



The total software uses the following resources (in both three shunts case and single shunt case, the first is smaller):

- FLASH usage: below 24Kbytes
- RAM usage: below 3Kbytes

Please Note that these data include also the communication interface and the kit board management.

The default program is based on the 8KHz sampling frequency, i.e. 125 μ s for the sampling period.

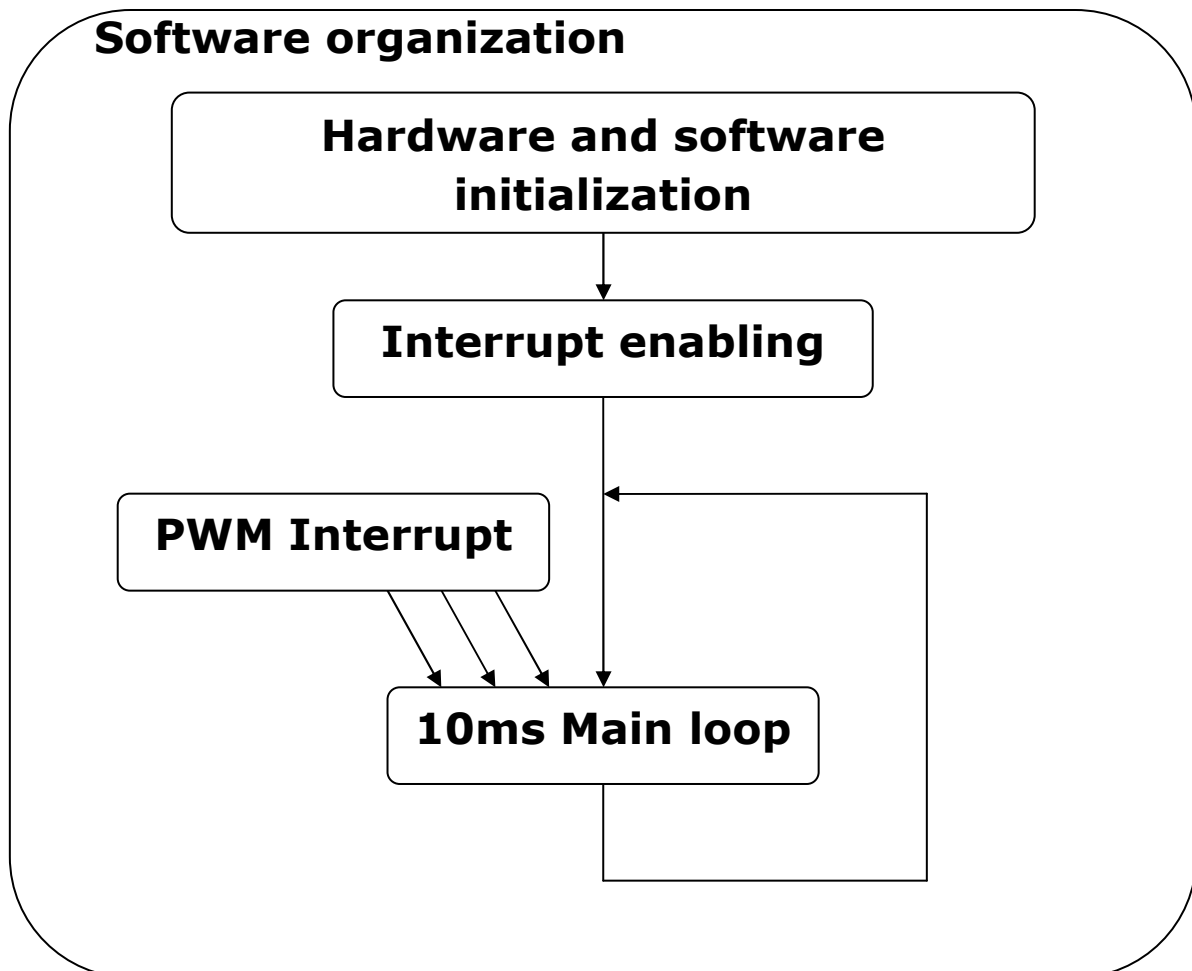
The PWM frequency is by default 16KHz and can be modified in the header file "customize.h".

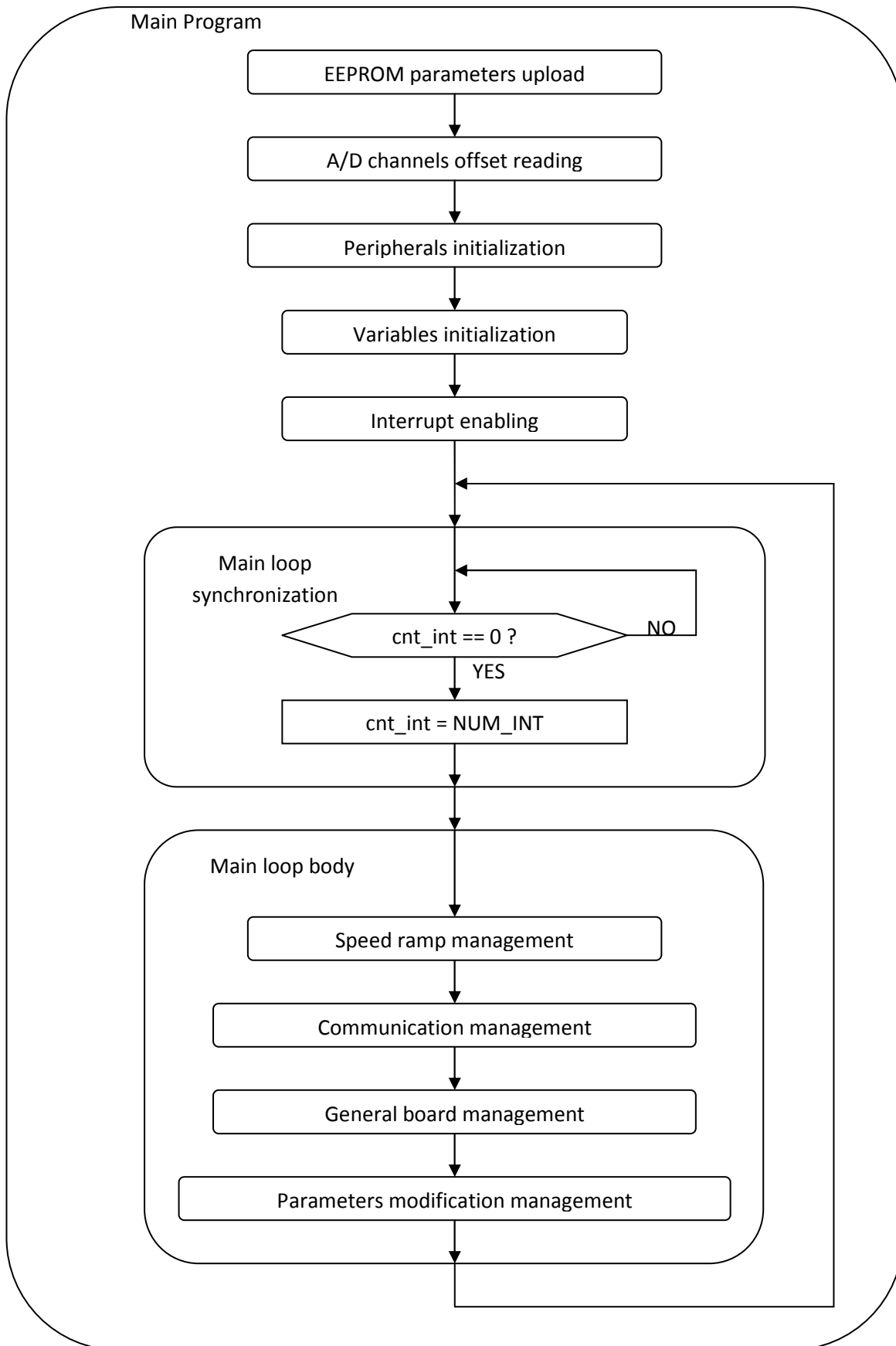
The execution time of the motor control interrupt is:

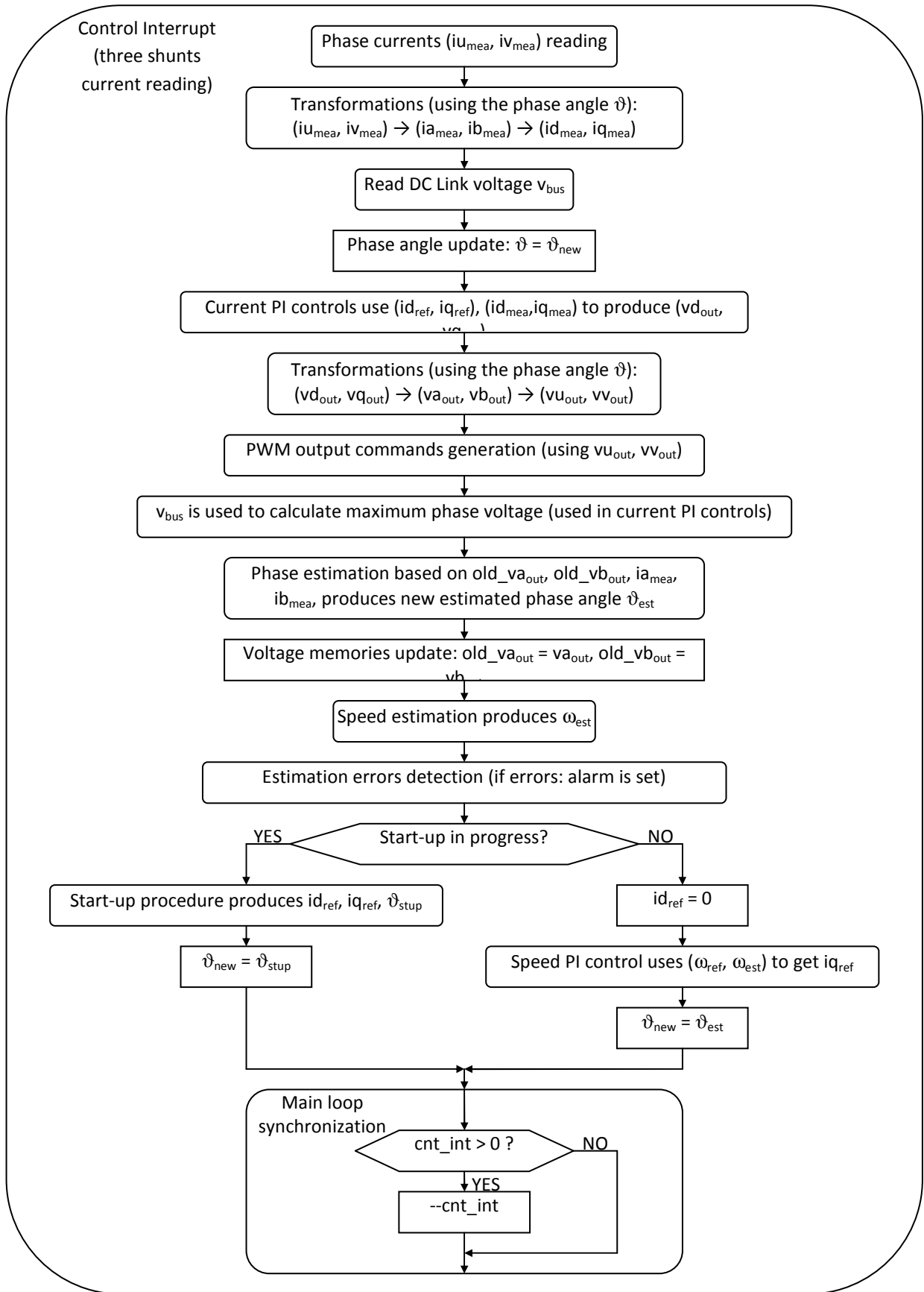
- For the three shunts software: <58 μ s
- For the single shunt software: <56 μ s

The following flowcharts show the software implementation of the motor control part of the software.

Please find below the flowchart for the main loop:





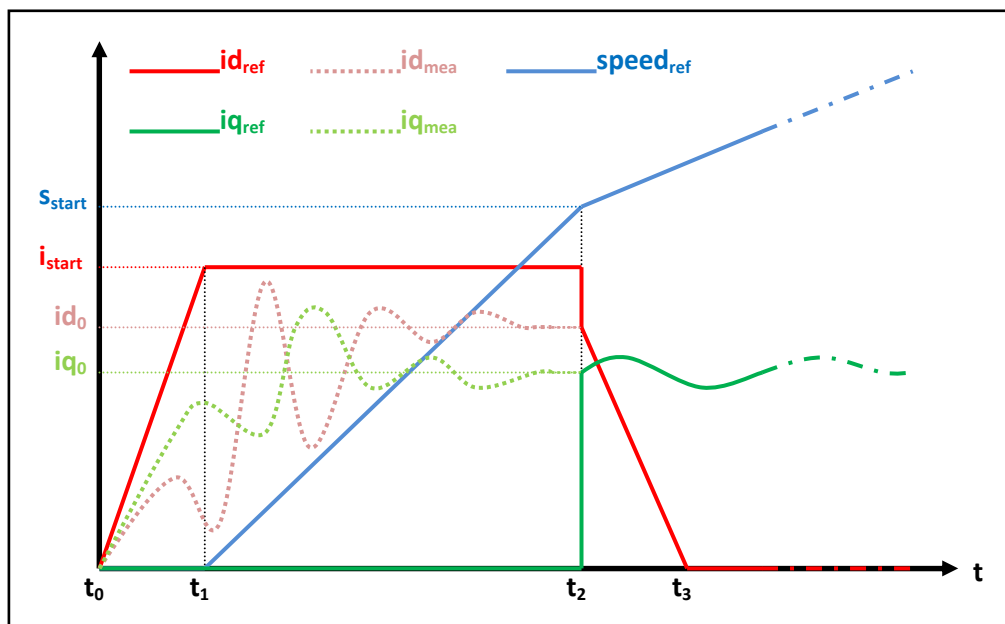


14. Start-up procedure

When the motor is in stand-still, the phase of the permanent magnet flux vector cannot be detected with the used algorithm. So an appropriate start-up procedure has to be applied.

The idea is to move the motor in feed-forward (with higher current than that required to win the load), till a speed at which the estimation algorithm can work. Then the system can be aligned to the estimated phase, and the current can be reduced to the strictly necessary quantity.

The following graph illustrates the strategy used (the suffix “*ref*” stands for *reference*, the suffix “*mea*” stands for *measured*).



Referring to the graph, the start-up procedure (in case of three shunts current reading) is described below.

- At the beginning t_0 , the system phase is unknown. No current is imposed to the motor; the system phase is arbitrarily decided to be $\vartheta_a=0$. All the references: i_{d_ref} , i_{q_ref} and $speed_ref$ are set to zero.
- From the moment t_0 , while the i_{q_ref} and the $speed_ref$ are maintained to zero, i_{d_ref} is increased with a ramp till the value i_{start} is reached at the moment t_1 .

The references are referred to an arbitrary (d_a, q_a) system based on the arbitrary phase ϑ_a . From this moment, the phase estimation algorithm begins to be performed, and the estimated phase ϑ_{est} is used to calculate the components of the measured current, referred to the (d, q) system based on the estimated phase, i_{d_mea} and i_{q_mea} . The components of the current referred to the arbitrary (d_a, q_a) system are controlled to follow the references by the current PI controllers. On the other hand, since the phase ϑ_{est} is still not correctly estimated, i_{d_mea} and i_{q_mea} have no physical meaning. Even if they are not shown in the graph, the applied voltages are subjected to the same treatment (v_{d_mea} and v_{q_mea} are calculated in the algorithm).

- c) At $t = t_1$, while $i_{q_{ref}}$ is maintained to zero and $i_{d_{ref}}$ is maintained to its value i_{start} , $speed_{ref}$ is increased with a ramp till the value s_{start} is reached at the $t = t_2$. The system phase $\hat{\vartheta}_a(t)$ is obtained simply by integration of $speed_{ref}$; in the meanwhile, the phase estimation algorithm begins to align with the real system phase. Furthermore $i_{d_{mea}}$ and $i_{q_{mea}}$ begin to be similar to the real flux and torque components of the current. The real components are supposed to be i_{d_0} and i_{q_0} (those values are obtained applying a low-pass filter to $i_{d_{mea}}$ and $i_{q_{mea}}$).

The interval $(t_2 - t_1)$ is the start-up time, and it is supposed to be large enough to allow the estimation algorithm to reach the complete alignment with the real phase of the system.

- d) At $t = t_2$, the phase estimation process is supposed to be aligned. At this point a reference system change is performed: from the arbitrary (d_a, q_a) reference to the (d, q) reference based on the estimated phase $\hat{\vartheta}_{est}$.

The current references are changed to the values i_{d_0} and i_{q_0} , and all the PI controllers are initialized with these new values. The speed PI integral memory is initialized with the value i_{q_0} , while the current PI integral memories are initialized with the analogous voltage values v_{d_0} and v_{q_0} , obtained from $v_{d_{mea}}$ and $v_{q_{mea}}$.

- e) After $t > t_2$, the normal control is performed, based on the estimated phase $\hat{\vartheta}_{est}$; the speed reference is increased with the classical ramp; the i_d current reference is decreased with a ramp, till it reaches the value zero at the moment t_3 ; then it is maintained to zero; the i_q current reference is obtained as output of the speed PI controller.

15. Reference system transformations in details

Find below the detailed equations used for the coordinates transformations in the embedded software for the RX220 microcontroller.

$$g_{\alpha} = \frac{2}{3}(g_u - \frac{1}{2}g_v - \frac{1}{2}g_w) = g_a$$

$$g_{\beta} = \frac{2}{3}(\frac{\sqrt{3}}{2}g_v - \frac{\sqrt{3}}{2}g_w) = \frac{1}{\sqrt{3}}(g_v - g_w) = \frac{1}{\sqrt{3}}(g_u + 2g_v)$$

(u, v, w) → (α, β)

$$g_u = g_{\alpha}$$

$$g_v = -\frac{1}{2}g_{\alpha} + \frac{\sqrt{3}}{2}g_{\beta} = (-g_{\alpha} + \sqrt{3}g_{\beta})/2$$

$$g_w = -\frac{1}{2}g_{\alpha} - \frac{\sqrt{3}}{2}g_{\beta} = (-g_{\alpha} - \sqrt{3}g_{\beta})/2$$

(α, β) → (u, v, w)

$$g_d = g_{\alpha} \cos(\vartheta) + g_{\beta} \sin(\vartheta)$$

$$g_q = -g_{\alpha} \sin(\vartheta) + g_{\beta} \cos(\vartheta)$$

(α, β) → (d, q)

$$g_{\alpha} = g_d \cos(\vartheta) - g_q \sin(\vartheta)$$

$$g_{\beta} = g_d \sin(\vartheta) + g_q \cos(\vartheta)$$

(d, q) → (α, β)

$$\left\{ \begin{array}{l} v_u = V \cos(\omega t + \varphi_0) \\ v_v = V \cos(\omega t + \varphi_0 - 2\pi/3) \\ v_w = V \cos(\omega t + \varphi_0 - 4\pi/3) \end{array} \right\} \leftrightarrow \left\{ \begin{array}{l} v_{\alpha} = V \cos(\omega t + \varphi_0) \\ v_{\beta} = V \sin(\omega t + \varphi_0) \end{array} \right\} \leftrightarrow \left\{ \begin{array}{l} v_d = V \cos(\varphi_0) \\ v_q = V \sin(\varphi_0) \end{array} \right\}$$

16. Rotor position estimation

The rotor position estimation method which has been chosen is the direct integration of the back EMF. Please find below the fundamental equations:

$$x = \Lambda_m \cos(\vartheta) = \lambda_{\alpha} - Li_{\alpha} = \lambda_{\alpha 0} + \int_0^t (v_{\alpha} - R_S i_{\alpha}) dt - Li_{\alpha}$$

$$y = \Lambda_m \sin(\vartheta) = \lambda_{\beta} - Li_{\beta} = \lambda_{\beta 0} + \int_0^t (v_{\beta} - R_S i_{\beta}) dt - Li_{\beta}$$

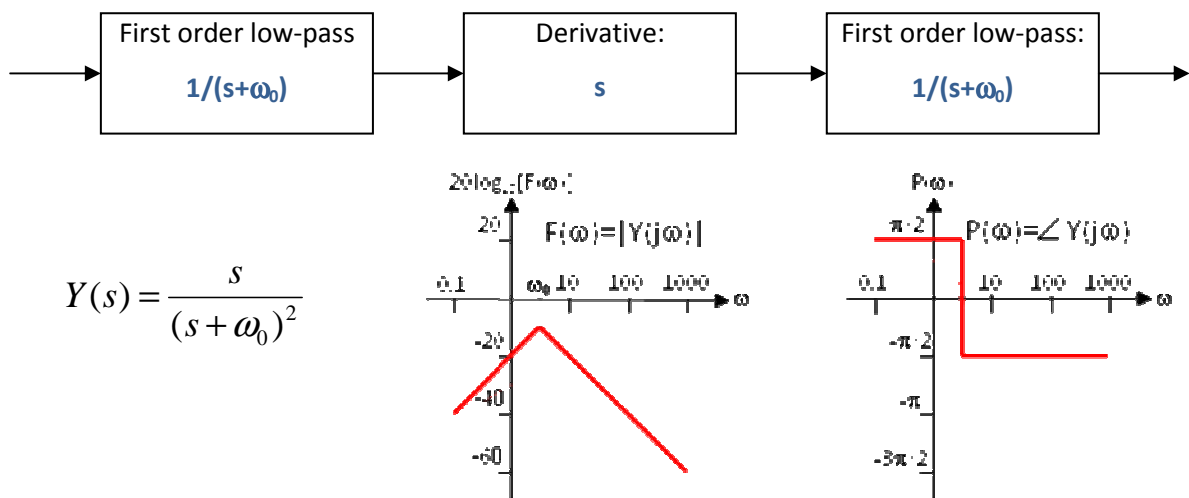
$$\vartheta = \arctan(x/y)$$

$$\omega = \frac{d}{dt} \vartheta(t)$$

The challenges in this approach are the calculation of the integrals which is well known as a problematic issue in a numeric context, and the choice of the initial conditions, which are not known in general. There are two possibilities to overcome these difficulties:

1. To use a so-called “approximated integration”, which means that instead of using an integral (1/s), a special transfer function is chosen, which is very similar to the integral in certain conditions.
2. To correct the result of the integration with a sort of feedback signal, obtained combining the estimated phase with the real flux amplitude, known as a parameter of the system.

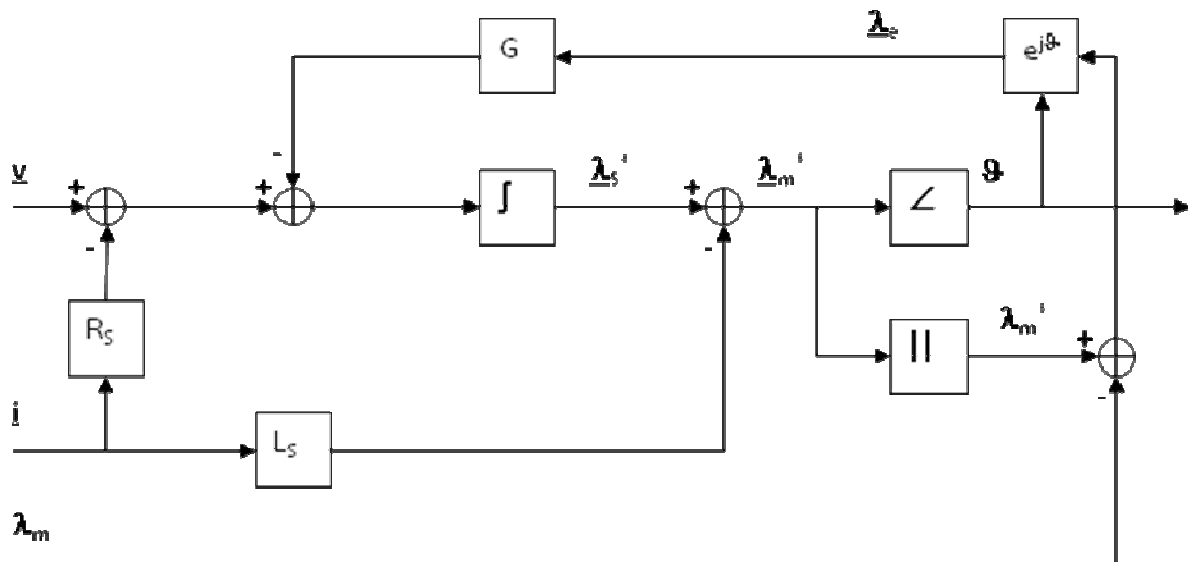
In the first case, we choose an integral approximation function which has a limited memory of the errors and with a zero DC gain. The goal is to reject any low frequency component, preventing the result to diverge, and automatically forgetting the errors (noise, etc.). This is obtained by combining a low-pass filter with a high-pass filter, as in the following scheme:



It is evident the relationship between $Y(s)$ and the integral $I(s)=1/s$ for $s=j\omega$, when $\omega \gg \omega_0$.

In the second case, to prevent the integral to diverge, and the errors related to wrong initial conditions are rejected, by the correcting action of the feedback.

The block scheme of the exact BEMF integration method for flux position estimation is the following:



The inputs of the system are the imposed voltage vector V and the measured current vector I . The motor phase resistance R_s , the synchronous inductance L_s and the permanent magnet flux amplitude λ_m are known as parameters and motor dependant.

The integral operation is corrected with a signal obtained modulating accordingly with the estimated phase the error between the estimated flux amplitude and the amplitude of the permanent magnets flux.

The gain of this correction is indicated with G . It is this feedback which avoids the integral divergence due to the errors or offsets. The higher G is, the higher is the relationship between the estimated amplitude and the theoretical one, but the larger can be the induced phase error.

The choice of G is a trade-off, in order to guarantee that the integral remains close to its theoretical value, but free enough to estimate the correct system phase.

In the standard embedded software the first strategy is selected. The choice to test the second one is left to the user thanks to the setting of the macros in the source code. Such modifications required a compilation of the embedded software.

17. Internal representation of physical quantities

The idea which lies under the internal representation of physical variables is to maximize the resolution, keeping as simple as possible the calculations and keeping reasonably low the memory occupation. So whenever it had been possible, the physical variables have been represented under a “per unit” criteria.

Please find below the description of the representation for each physical quantity.

Angles

The interval $[0, 2\pi]$ is represented with the interval $[0, 65536)$, with the resolution of $2\pi/65536$ rad.

$$\text{Angle}[\text{internal_angle_unit}] = KA * \text{Angle}[\text{rad}]$$

$$KA = 32768 / \pi (= 10430.37835)$$

Note that in this way the angle can be considered unsigned in the range $[0, 65536)$, or signed in the range $[-32768, 32768)$, with identical results. In every case the representation requires a 16bit word.

Trigonometric functions

$\sin(a)$, $\cos(a)$ are normalized to the value $NORMVAL = 16384$.

$$\text{Internal_sin}(a[\text{internal_angle_unit}]) = NORMVAL * \sin(a[\text{rad}]), \text{NORMVAL} = 16384$$

$$-NORMVAL \leq \text{Internal_sin}() \leq NORMVAL \text{ (the same for Internal_cos())}$$

Time

The time is expressed as a multiple of the sampling period T_s .

$$\text{Time}[\text{internal_time_unit}] = KT * \text{Time}[\text{sec}]$$

$$KT = F_s \text{ (} F_s = \text{ sampling_frequency} = 1 / T_s \text{)}$$

Angular velocity

The angular velocity is expressed as a function of angles and time, in order to obtain it as the subtraction of two angles in two sampling moments; for resolution reasons, an amplification is needed, and we choose this amplification equal to $NORMVAL=16384$.

$$\begin{aligned} \text{Omega}[\text{internal_angular_velocity_unit1}] &= KO1 * \text{Omega}[\text{rad / sec}] = \\ &= (KO1 * KT / KA) * \text{Angle}[\text{internal_angle_unit}] / \text{Time}[\text{internal_time_unit}] \end{aligned}$$

Since we want:

$$\begin{aligned} \text{Omega}[\text{internal_angular_velocity_unit1}] &= \\ &= NORMVAL * \text{Angle}[\text{internal_angle_unit}] / \text{Time}[\text{internal_time_unit}] \end{aligned}$$

$$(\rightarrow \text{Omega}[\text{internal_angular_velocity_unit1}] = \text{NORMVAL} * (\text{Angle}(n) - \text{Angle}(n - k)) / k)$$

We obtain:

$$\text{KO1} = \text{NORMVAL} * \text{KA} / \text{KT} = \text{NORMVAL} * 65536 / (2 * \pi * \text{Fs})$$

The entire speed range cannot, in general, be represented in a 16bit word, but a long is needed. This high resolution can be useful for some particular calculations, while when, for example, the speed is used to calculate voltages, lesser resolution is enough. To reduce the overall calculation time, the most effective choice is to have a second representation of the angular speed, coherent with the voltage and current representations, which are "per unit" based. So the second representation of the angular speed is based on a normalized value:

$$\text{BASE_SPEED_R_S} = \text{MAX_OMEGA_R_S}$$

The so called `MAX_OMEGA_R_S` is the maximum angular velocity required by the application, and we will associate this to `NORMVAL`. This value is linked to the maximum frequency ($\text{MAX_OMEGA_R_S} = 2\pi * \text{MAX_FRE_HZ}$). The second representation is the following:

$$\text{Omega}[\text{internal_angular_velocity_unit2}] = \text{KO2} * \text{Omega}[\text{rad / sec}]$$

$$\text{KO2} = \text{NORMVAL} / \text{BASE_SPEED_R_S}$$

To pass from a representation to the other we have the following relationship:

$$\text{Omega}[\text{internal_angular_velocity_unit2}] = (\text{KO2} / \text{KO1}) * \text{Omega}[\text{internal_angular_velocity_unit1}]$$

$$\text{KO2} / \text{KO1} = (2 * \pi * \text{Fs}) / (65536 * \text{MAX_OMEGA_R_S})$$

$$\text{KO1} / \text{KO2} = 65536 * \text{MAX_FRE_HZ} / \text{Fs}$$

Voltage

We can start our considerations from the maximum voltage readable by the A/D converter; this value is the maximum DC bus voltage and it is related to the maximum peak phase voltage by the relation: $\text{Vout_pk} = (2/3) * \text{Vbus}$ (in case of over-modulation); this would already leave a good margin in voltage representation, but in case of deep flux weakening, the intermediate calculations can lead to higher voltage values, so we choose as the base voltage value the following:

$$\text{BASE_VOLTAGE_VOLT} = (2 \wedge K) * \text{MAX_VOLTAGE_VOLT}, \text{ with } K \text{ related with the application}$$

`MAX_VOLTAGE_VOLT` is the maximum voltage readable by the A/D converter. With normal applications, ($K = 1$) leaves a margin for the maximum phase voltage equal to 3 times Vbus , which is more than enough. The voltage representation becomes:

$$\text{Voltage}[\text{internal_voltage_unit}] = \text{KV} * \text{Voltage}[\text{Vol}]$$

$$\text{KV} = \text{NORMVAL} / \text{BASE_VOLTAGE_VOLT}$$

Current

The maximum current readable by the A/D converter is chosen as the base value:

$$\text{BASE_CURRENT_AMP} = \text{MAX_CURRENT_AMP}$$

It is represented with NORMVAL = 16384:

$$\text{Current}[\text{internal_current_unit}] = \text{KI} * \text{Current}[\text{Amp}]$$

$$\text{KI} = \text{NORMVAL} / \text{BASE_CURRENT_AMP}$$

Impedance

The base impedance value can be deduced by the base voltage and current values; in fact the extended value chosen as the base voltage keeps into account the flux weakening, and no other trick are required in case of PM motor (in case of induction motor, the current can be much higher than the ratio between voltage and the impedance due to the magnetizing inductance: this would require some modification to the representation). So we keep simply:

$$\text{BASE_IMPEDANCE_OHM} = \text{BASE_VOLTAGE_VOLT} / \text{BASE_CURRENT_AMP}$$

The internal representation is:

$$\text{Impedance}[\text{internal_impedance_unit}] = \text{KZ} * \text{Impedance}[\text{Ohm}]$$

$$\text{KZ} = \text{NORMVAL} / \text{BASE_IMPEDANCE_OHM} =$$

$$= \text{NORMVAL} * \text{BASE_CURRENT_AMP} / \text{BASE_VOLTAGE_VOLT}$$

Resistance

The resistance is expressed in function of the "base" resistance, which is kept equal to the base impedance; this leads usually in a "poor" representation of the resistance in terms of resolution, but the resistance itself is highly variable with many factors, and an higher resolution is usually not required.

$$\text{BASE_RESISTANCE_OHM} = \text{BASE_IMPEDANCE_OHM}$$

$$\text{Resistance}[\text{internal_resistance_unit}] = \text{KR} * \text{Resistance}[\text{Ohm}]$$

$$\text{KR} = \text{KZ}$$

Inductance

The base inductance value is derived from the impedance and the angular velocity:

$$\text{BASE_INDUCTANCE_HEN} = \text{BASE_IMPEDANCE_OHM} / \text{BASE_SPEED_R_S}$$

so the internal representation becomes:

$$\text{Inductance}[\text{internal_inductance_unit}] = \text{KL} * \text{Inductance}[\text{Henry}]$$

$$\begin{aligned}
 KL &= \text{NORMVAL} / \text{BASE_INDUCTANCE_HEN} = \\
 &= \text{NORMVAL} * \text{BASE_SPEED_R_S} * \text{BASE_CURRENT_AMP} / \text{BASE_VOLTAGE_VOLT}
 \end{aligned}$$

Flux

In a similar way, the "base" flux can be chosen equal to:

$$\text{BASE_FLUX_WEB} = \text{BASE_VOLTAGE_VOL} / \text{BASE_SPEED_R_S}$$

Then we can express the flux as:

$$\text{Flux}[\text{internal_flux_unit}] = KF * \text{Flux}[\text{volt} * \text{sec} / \text{rad}]$$

$$KF = \text{NORMVAL} / \text{BASE_FLUX_WEB}$$

Calculation relationships

Please find below some useful relations derived from the previous assumptions (we will indicate all the "internal_xxxx_unit" with "int"):

$$\text{Impedance}[\text{int}] = (\text{Inductance}[\text{int}] * \text{Omega}[\text{in}^2]) / \text{NORMVAL}$$

$$\text{Flux}[\text{int}] = (\text{Inductance}[\text{int}] * \text{Current}[\text{int}]) / \text{NORMVAL}$$

$$\text{Voltage}[\text{int}] = (\text{Impedance}[\text{int}] * \text{Current}[\text{int}]) / \text{NORMVAL}$$

$$\text{Voltage}[\text{int}] = (\text{Flux}[\text{int}] * \text{Omega}[\text{in}^2]) / \text{NORMVAL}$$

As you can notice, the calculations becomes particularly simple ($x/\text{NORMVAL}$ is $x \gg 14$).

18. EEPROM parameters list

Please find below the software parameters list including their full description. Each parameters located in the “customize.h” header file can be tuned by the user directly by the Graphic User Interface, without re-compiling the program.

0) (SEL_OP) default parameters setting, Used to perform special operations, like default parameter set re-loading, or current PI tuning working mode setting.

1) (RPM_MIN) minimum speed [rpm]

2) (RPM_MAX) maximum speed [rpm]

3) (R_ACC) acceleration [rpm/s]

4) (R_DEC) deceleration [rpm/s]

5) (C_POLI) polar couples

6) (I_START) start-up current (peak) [Ampere/AMP_RES]

Used to specify the peak phase current value to be used during the start-up.

7) (I_MAX) maximum phase current (peak) [Ampere/AMP_RES]

8) (R_STA) stator resistance [Ohm/OHM_RES]

9) (L_SYN) synchronous inductance [Henry/HEN_RES]

10) (PM_FLX) permanent magnets flux [Weber/WEB_RES]

The right value is needed if the exact integration flux estimation algorithm is chosen; in case of approximated integration (standard case) this value is not taken into account.

11) (KP_CUR) current loop KP

12) (KI_CUR) current loop KI

13) (KP_VEL) speed loop KP

14) (KI_VEL) speed loop KI

15) (FB_GAIN) flux amp. feedback gain

Only needed if the exact integration flux estimation algorithm is chosen; in case of approximated integration (standard case) this value is not taken into account.

16) (PHA_OFF) phase offset [deg]

It is used to add a phase offset to the phase estimation, to reach the best alignment if needed.

17) (ST_TIM) start-up acceleration time [sec/SEC_RES]

18) (FLX_FS) flux estimation filter shifts

Only needed if the approximated integration flux estimation algorithm is chosen (standard case); if exact integration method is chosen, this value is not taken into account

19. PC Graphical User Interface

Once the Motor Control PC GUI is installed based on the explanations of the Quick Start Guide, please click on the “Speed Control” button to display the following window:

The screenshot displays the Motor Control PC GUI with several key components and callouts:

- Speed Control Window:** Contains three real-time graphs: SPEED (rpm), VOLTAGE (V), and CURRENT (mA). Callout: "Motor Operation graphs - Speeds, Currents and Voltages".
- Function Tabs:** A vertical sidebar on the left with buttons for "Communication Settings", "Algorithm information", "Parameters Setting", "System Monitor", "Speed Control", and "Position Control". Callout: "Function Tabs".
- RPM CONTROL:** A circular gauge showing motor speed with a needle at 0. Callout: "Motor Control - Start, Stop, Speed and Direction DEMO Mode".
- PROPERTY MONITOR:** A gauge for "Motor speed" and a table of parameters. Callout: "Property Monitor where motor parameters can be analysed - Speed, Currents, Voltages, Torque - Operation and Parameters can be saved to Excel file".
- Parameters Setting Window:** A table for configuring motor parameters. Callout: "Motor Tuning Parameters".
- System Monitor Window:** Displays system status including clock frequency, flash/ram occupation, PWM modulation frequency, and sampling frequency. Callout: "System Parameters".

DESCRIPTION	UNIT	MIN	MAX	VALUE	VALID
00. Default Parameters Setting	-	0	32767	0	true
01. Minimum Speed	rpm	200	5000	2000	true
02. Maximum Speed	rpm	1000	20000	7000	true
03. Acceleration	rpm/s	1	10000	4000	true
04. Deceleration	rpm/s	1	10000	2000	true
05. Polar couples	-	1	4	2	true
06. Startup Current	Apk/10	0	5000		
07. Maximum Current	Apk/10	0	5000		
08. Stator Resistance	Ohm/10	0	5000	40	true

Please find below the description of the Alarm codes coming from the PC GUI:

Alarm 1:

The alarm 1 is called “EEPROM alarm” and described in the software by “EQP_ALL”. This alarm is set when one or more EEPROM parameters are higher than the maximum allowed value or lower than the minimum allowed value. The LED DL4 is quickly blinking on the main board to indicate that an alarm is set.

The maximum and minimum values are specified in the two constants tables called: "par_max[]" "par_min[]" in the "ges_eqp.h" header file. Another root cause for the alarm 1 is the EEPROM hardware failure when the error is accessed in read or write mode.

When this alarm is active, the access to the EEPROM is restricted. To reset the alarm the default parameters set should be reloaded in the EEPROM. By using the PC GUI and the parameters setting window, it becomes possible to clean the EEPROM content. The first step is to write the magic number “33” in the first parameter n°00. The second step is to reset the board by pressing the reset button on the PCB or switching off the power supply. At this point a coherent set of parameters is loaded and the alarm should disappear.

Finally, if the alarm is produced by a hardware failure of the EEPROM itself, then the board needs to be repaired.

Alarm 2:

The alarm 2 is called “hardware overcurrent” and described in the software by “FAULT_ALL”. This alarm is produced by the MCU peripheral called Port Output Enable (POE) in case of external overcurrent signal. The hardware overcurrent is producing a falling edge input on the POE pin. Furthermore, if the hardware level of the PWM output pin is not coherent with the level imposed by software, the alarm 2 will also be triggered.

The LED DL4 is quickly blinking on the main board to indicate that an alarm is set.

The only way to clear the alarm is to reset the board by using the reset button on the PCB or by switching off the supply and on again.

Finally, one of the root causes of the Alarm 2 is a hardware defect or a wrong behavior of the current control. So please also check the setting of the current PI coefficients that are stored in EEPROM or used in real-time.

Alarm 3:

The alarm 3 is called “loss of phase” and described in the software by “TRIP_ALL”. This alarm is produced when the sensorless position detection algorithm is producing inconsistent results. It means that the rotor position is unknown due to a lack of accuracy, so the motor is stopped.

The LED DL4 is quickly blinking on the main board to indicate that an alarm is set.

This alarm can be reset by setting the speed reference to zero on the PC GUI.

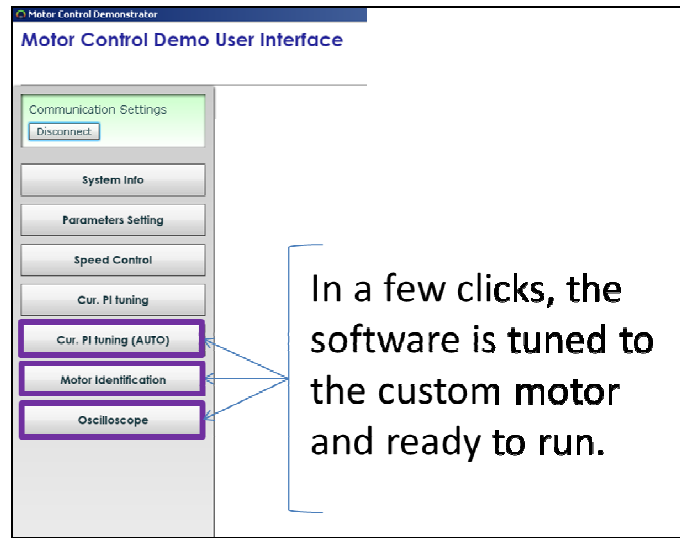
Please find below an extract of the header file “const_def.h”:

```
#define EQP_ALL          1      // EEPROM alarm code
#define FAULT_ALL       2      // overcurrent hardware alarm code (POE)
#define TRIP_ALL        3      // loss of phase alarm code
```

Finally, the PC GUI button called “parameters setting” is used to enter and modify the motor and applications parameters.

20. Motor Auto-calibration using the PC GUI

The full calibration of any 3-phase AC Brushless motor can be performed automatically using the PC Graphical User Interface. Three specific buttons are now available for and shown below:



In terms of AC Brushless motor driven in sinusoidal mode and FOC algorithm, the most important parameters to tune are:

1. Current PI parameters: Proportional K_p and Integral K_i
2. Motor parameters: Stator resistance R_s , the synchronous inductance L_s , and the Permanent Magnet flux Λ_m .

Let's tune step by step a real low voltage PMSM motor using the internal power stage with Mosfets.

Important Note: The auto-tuning embedded software is working only on the three shunts version.

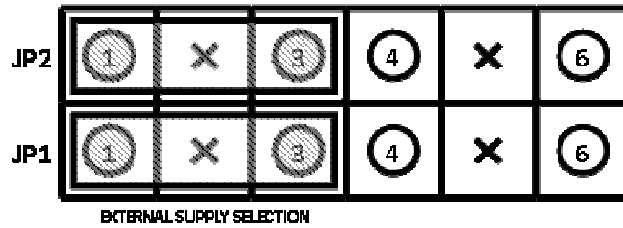
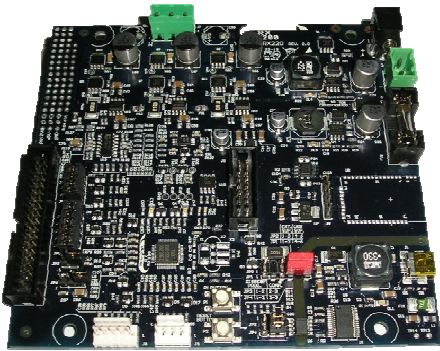
a) The BLAC Motor selected is the following one:

- Motor type: **MB057GA240**
- Maximum current: **3.5A**
- Bus Voltage: **50V**
- Maximum speed: **5000 RPM**
- Number of pole pair: **2**

MOTOR PARAMETERS @25°C			
	TOL.	UNITS	VALUE
MAX. OPERATING SPEED (S_{nl})	MAX.	R.P.M.	5000
CONTINUOUS TORQUE (T_c)	MAX.	OZ-IN	38
PEAK TORQUE (T_p)	MAX.	OZ-IN	114
CONTINUOUS CURRENT (I_c)	MAX.	AMPS	3.5
PEAK CURRENT (I_p)	MAX.	AMPS	10.6
TORQUE SENSITIVITY (K_t)	±10%	OZ-IN/AMPS	11.9
BACK EMF CONSTANT (K_e) (L-L, D.C.)	±10%	V/K R.P.M.	8.8
D.C. RESISTANCE (R_t) (L-L)	±10%	OHMS	1.12
INDUCTANCE (L) (L-L)	±15%	mH	3.19
ROTOR INERTIA (J_R)	NOM.	OZ-IN-SEC ²	0.0017
WEIGHT	NOM.	LBS	1.78



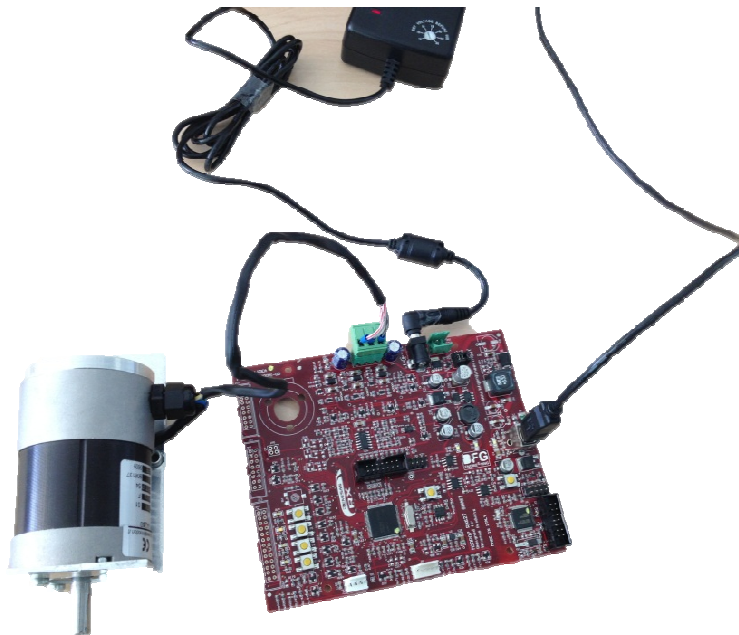
b) Let's setup the Motor control kit for 24V_{DC} external power supply: the jumper JP1 and JP2 needs to be set to 1-3 position as explained in the "Chapter 3 Power Supply selection".



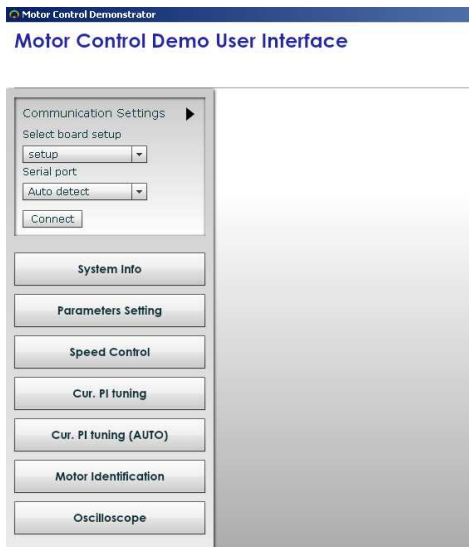
c) Let's connect the 24V_{DC} Power supply to the RX220 motor control reference kit:



d) Now, connect the USB cable to the PC and the Kit and connect the 24V to the kit and the motor to the kit:



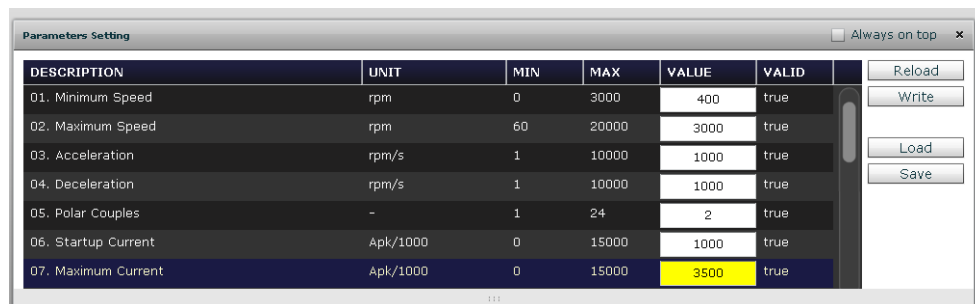
e) Launch the PC GU from the folder: "C:\Program Files\MCDemo" launch: "MotorController.exe"



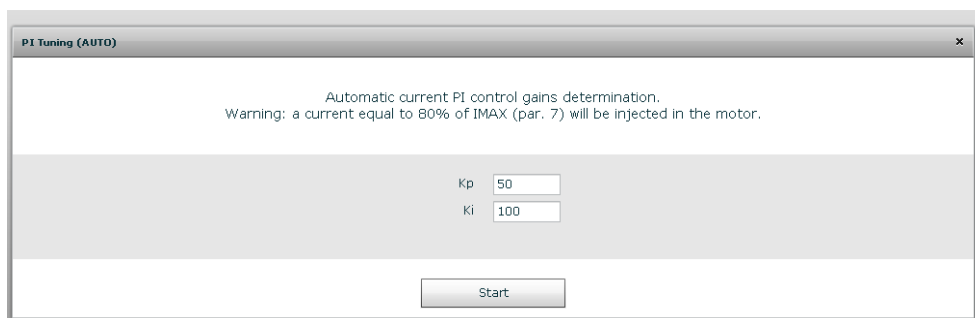
Click on the “setup” button and select “RX220 Kit” and select “Autodetect” and click on “Connect” to ensure the PC GUI is connected to the RX220 kit.

On the left hand side, the new buttons appears: “Cu. PI tuning”, “Cu. PI tuning (AUTO)”, “Motor Identification” and “Oscilloscope”.

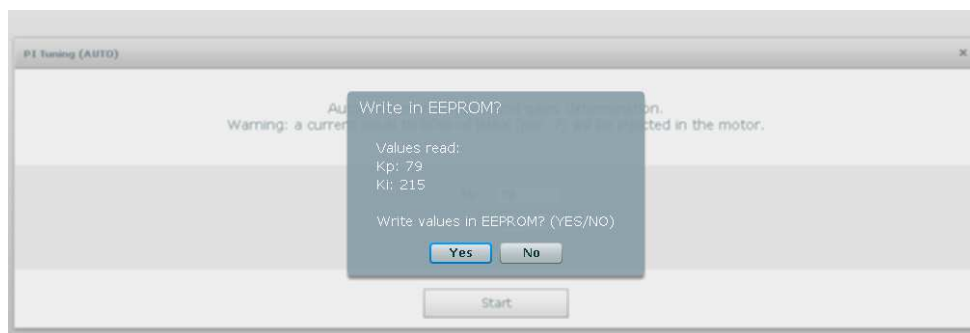
f) Set the **maximum current** (parameter n°07) as it will influence all the next steps: Click on “Parameters settings” Enter the value: 3500 (the unit is in mA) and click on “Write” to save the parameter into the EEPROM. And close the window



g) Click now on “Cu. PI tuning (AUTO)” button and press “start” to perform an automatic Current PI tuning.



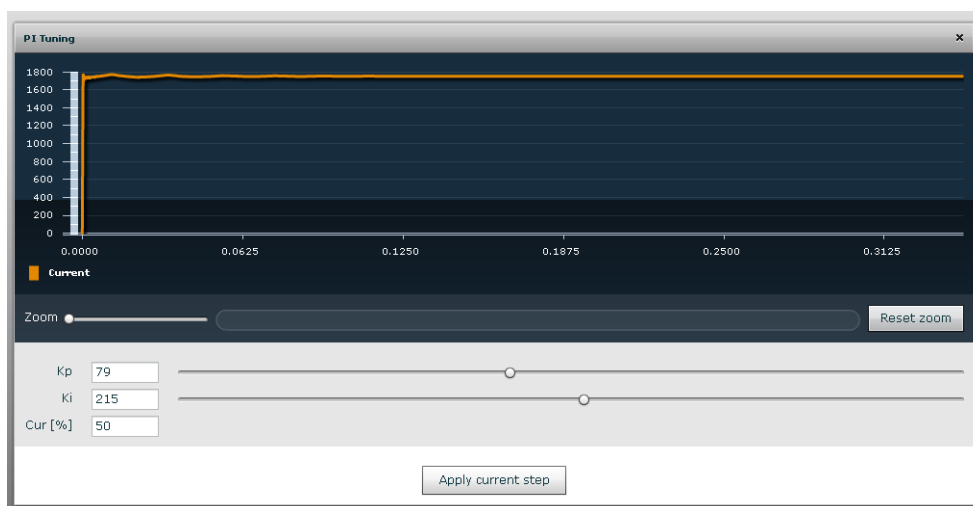
And accept the results to be programmed into the EEPROM as shown below.



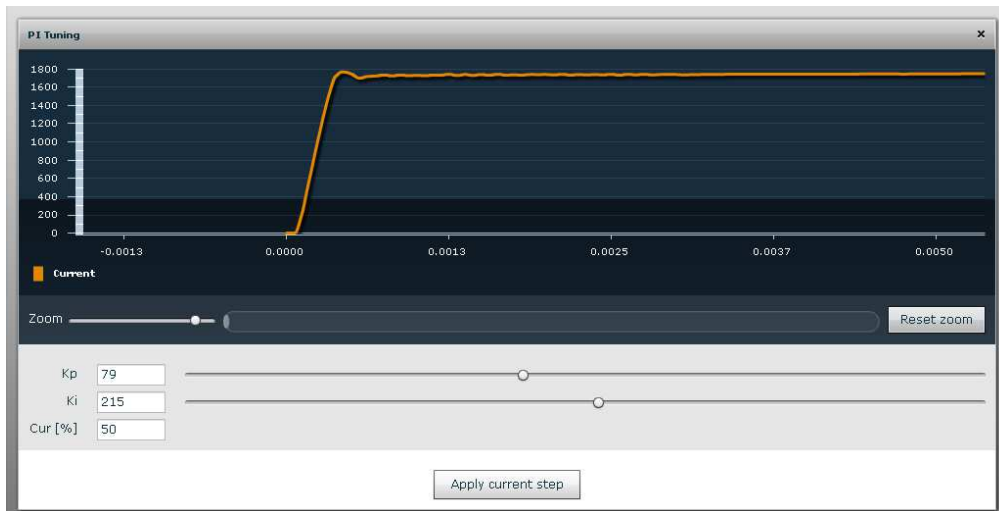
h) Now click on the button “Cu. PI tuning” to open the manual current PI tuning window and check the step answer by clicking on “Apply current step” button.



Depending on the motor, the parameters found by the automatic procedure can be too fast or too slow.



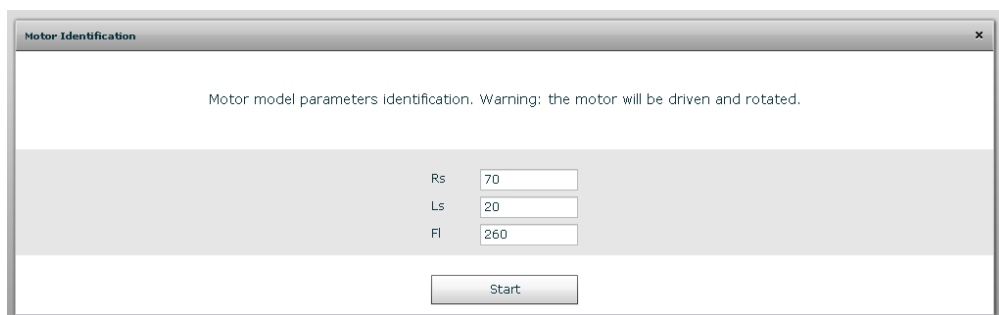
Please use the Zoom function to check the beginning of the step:



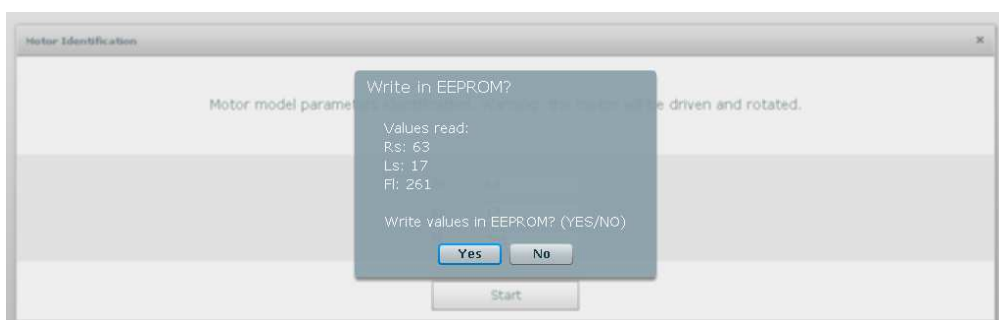
You can adjust manually the parameters to obtain an even better step response and also increase the step current level by increasing the percentage of “Cur. [%]” to 90%. The default value is 50%.

Once it’s done, the window can be closed as the proportional and integral coefficients of the PI current are tuned.

i) Perform an auto-identification of the motor parameters by clicking on “Motor Identification” and click “start”:

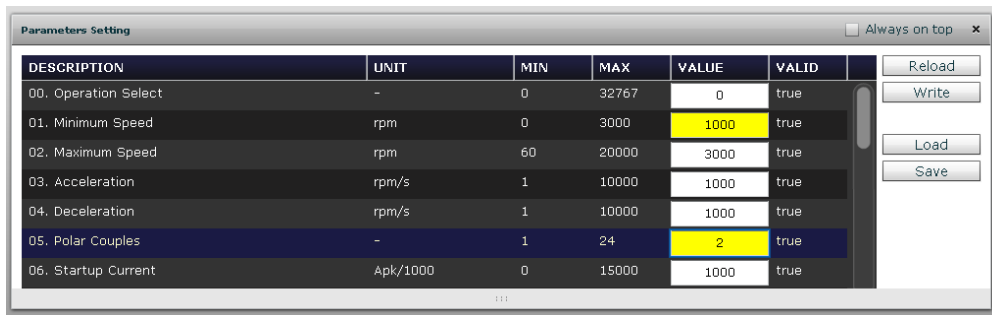


And accept the results to store them into the EEPROM.



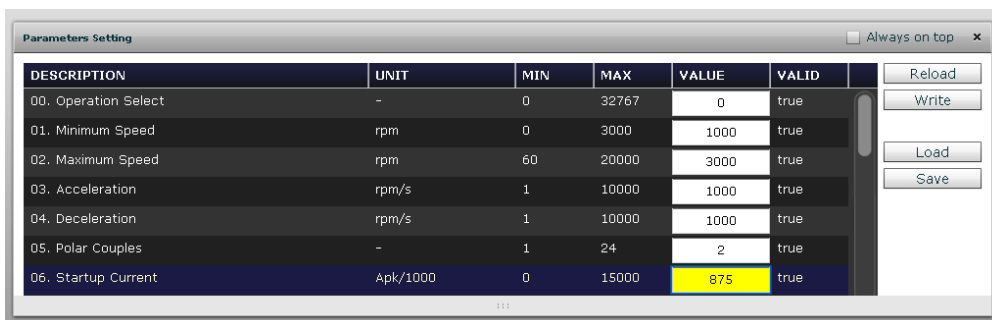
The stator resistance, the synchronous inductance and the Permanent Magnet flux have been measured and tuned.

j) Now please click on “parameters settings” and enter the number of pole pairs: 2 (parameter n°5) and enter a minimum speed or 1000 RPM (15Hz of a one pole pair motor).



DESCRIPTION	UNIT	MIN	MAX	VALUE	VALID
00. Operation Select	-	0	32767	0	true
01. Minimum Speed	rpm	0	3000	1000	true
02. Maximum Speed	rpm	60	20000	3000	true
03. Acceleration	rpm/s	1	10000	1000	true
04. Deceleration	rpm/s	1	10000	1000	true
05. Polar Couples	-	1	24	2	true
06. Startup Current	Apk/1000	0	15000	1000	true

k) Set a start-up current equal to 25% of the maximum current. In our case 25% of 3.5A is 0.875A. Please enter the value 875 into the parameter n°6 and click on the “write” button on the left.



DESCRIPTION	UNIT	MIN	MAX	VALUE	VALID
00. Operation Select	-	0	32767	0	true
01. Minimum Speed	rpm	0	3000	1000	true
02. Maximum Speed	rpm	60	20000	3000	true
03. Acceleration	rpm/s	1	10000	1000	true
04. Deceleration	rpm/s	1	10000	1000	true
05. Polar Couples	-	1	24	2	true
06. Startup Current	Apk/1000	0	15000	875	true

Let's close the window.

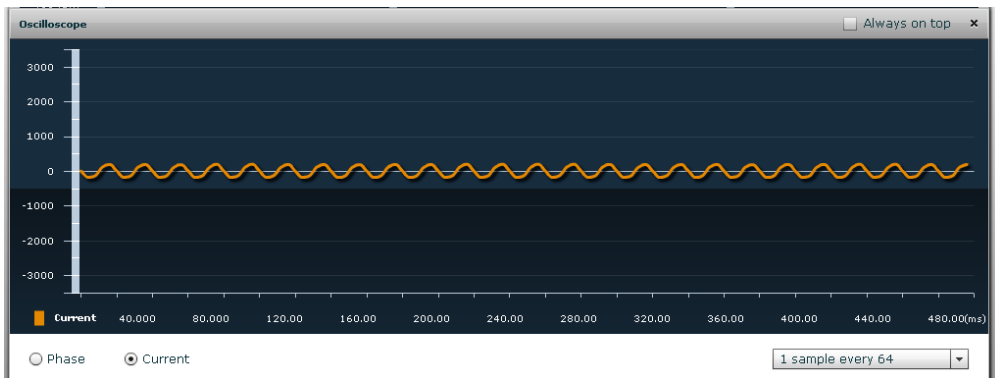
l) Please click on the button: “Speed Control”:



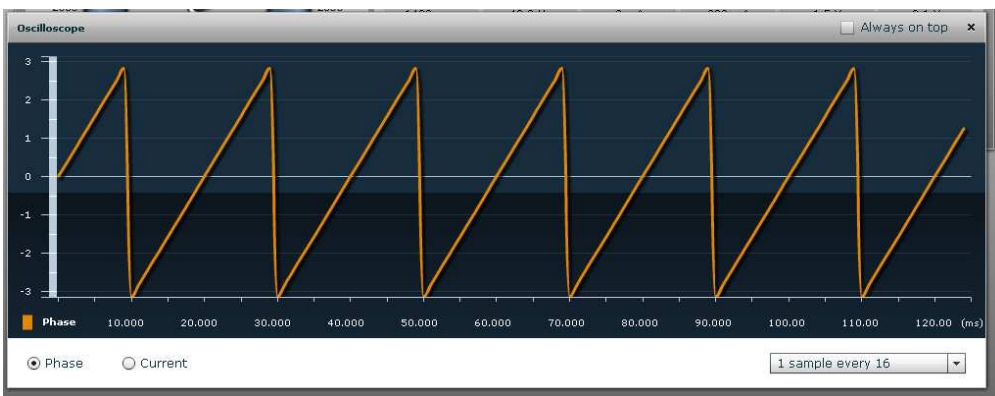
To start the motor, let's enter a speed which is 1.5 times the minimum speed, in this case **1500 RPM**



Please click on the “Oscilloscope” button to see the motor waveforms with the current in Y-axis and the time in x-axis.



You can also display the phase by clicking on “Phase” selector:

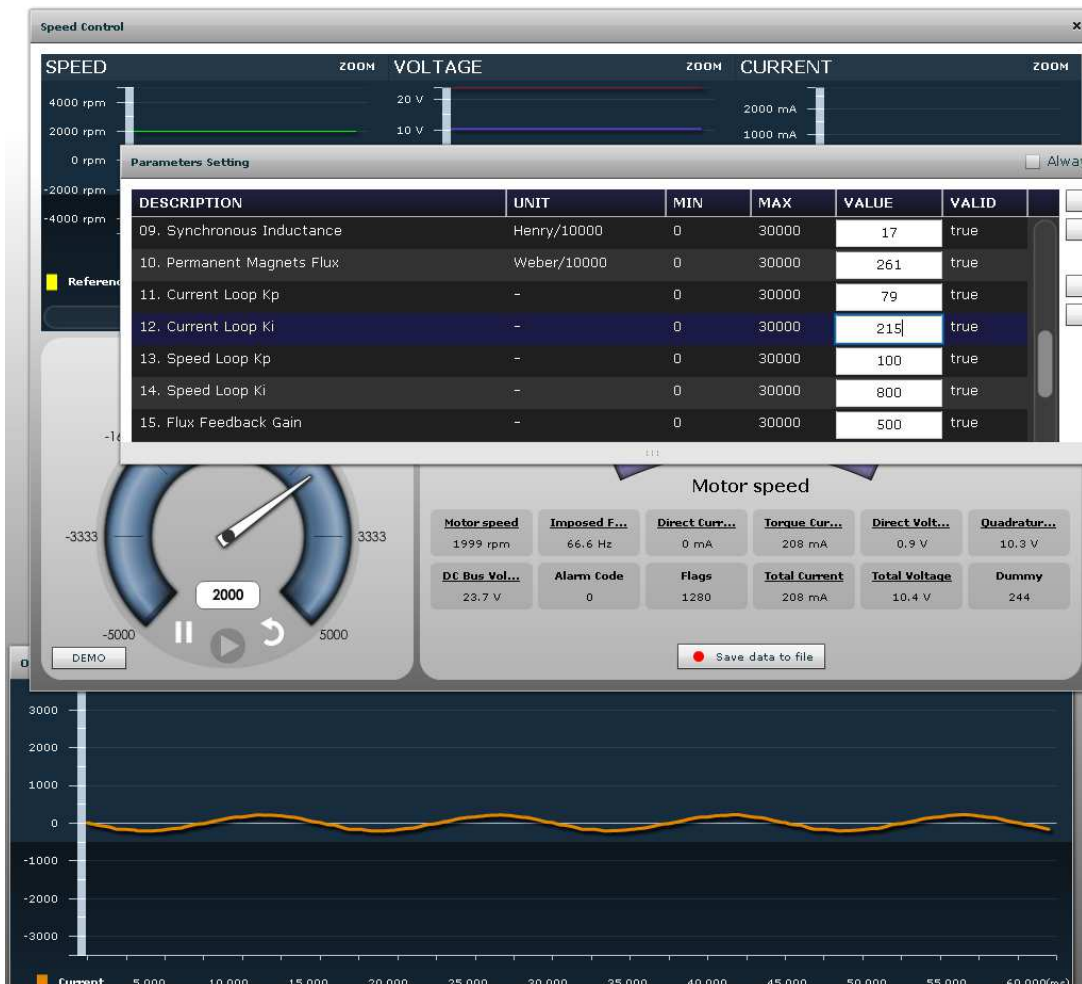


For the oscilloscope window, use an opportune time scale: “1 sample every 1” should be used for extremely fast phenomena when running at very high speed. The setting “1 sample every 128” should be used for extremely low phenomena when running at very low speed.

Let's start with an intermediate value and adjust it in order to see some periods of the current or the phase.

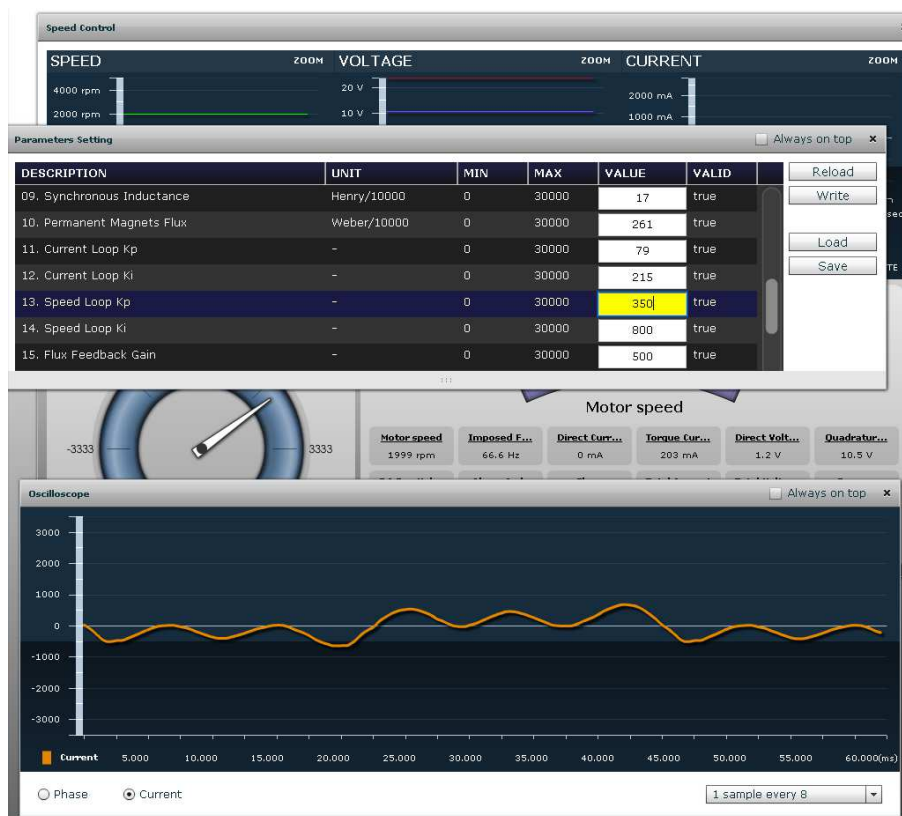
When the motor is running, you can adjust the speed PI parameters.

Please follow the procedure: while running at a medium speed range: 2 times the minimum speed.
In our example, the speed is set to 2000 RPM



Start by increasing the Parameter n°13 (Kp) until the instability that can be display in the current or phase waveform window.

Add a step of "50" and click "write" to see the effect and keep on increasing it.



In our case, at 350 it started to be very unstable, but the motor is still running. Set the speed to "0".

Then use half of the found value: 175 in our case, click on "write" and set the speed to 2000 RPM.

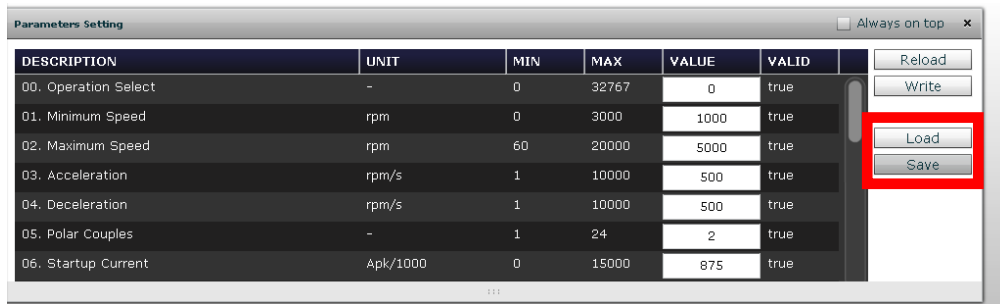
The screenshot shows the 'Parameters Setting' window with the 'Speed Loop Kp' parameter (row 13) highlighted in yellow and its value set to 175. The other parameters remain the same as in the previous screenshot.

DESCRIPTION	UNIT	MIN	MAX	VALUE	VALID
09. Synchronous Inductance	Henry/10000	0	30000	17	true
10. Permanent Magnets Flux	Weber/10000	0	30000	261	true
11. Current Loop Kp	-	0	30000	79	true
12. Current Loop Ki	-	0	30000	215	true
13. Speed Loop Kp	-	0	30000	175	true
14. Speed Loop Ki	-	0	30000	800	true
15. Flux Feedback Gain	-	0	30000	500	true

Do the same for the parameter n°14 which is the speed loop Ki parameter. Increase it until it becomes unstable. In our case the critical value is reached at 2800 for Ki, so the value to be used is: 1400.

n) Test now all the speed ranges and different rotation.

o) Finally the parameters list can be saved in a file in .CSV format for further used and can also be uploaded later on:



DESCRIPTION	UNIT	MIN	MAX	VALUE	VALID
00. Operation Select	-	0	32767	0	true
01. Minimum Speed	rpm	0	3000	1000	true
02. Maximum Speed	rpm	60	20000	5000	true
03. Acceleration	rpm/s	1	10000	500	true
04. Deceleration	rpm/s	1	10000	500	true
05. Polar Couples	-	1	24	2	true
06. Startup Current	Apk/1000	0	15000	875	true

Troubleshooting:

At the stage i) if the motor doesn't start or generate an alarm n°3, please set the speed to "0" to clear the alarm which indicates that the software lost the phase. One first test is to increase or decrease the start-up current and the minimum speed or the speed PI gains

When the motor is running, you can verify the number of pole pairs taking measurement of the effective speed, and comparing it with the imposed frequency: the number of pole pairs n is: $n = \text{freq} * 60 / \text{speed}$; if you change the number of pole pairs, remember to adjust also the minimum (and maximum) speed values. Sometimes the no load start-up is easier if the inductance parameter is set to 0.

All the procedure is tuned to manage motors which maximum current is close to the inverter capability, which is around 6Arms for the external power stage (shunt=0.05Ohm) and 3Arms for the internal power stage (shunt=0.10hm); if you try to use it for very different motors, the results will be influenced by the losses in current reading resolution.

Another possible trick when the things are very difficult, is trying to increase the flux feedback gain; sometimes I used 500 instead of 100.

21. List of motors tuned automatically using the PC GUI

Please find below a short list of AC Brushless motors tuned automatically using the auto-tuning procedure described above.

For each motor a specific text file is available to be loaded onto the PC GUI.

						
Part-name	ECI 24.42	BD35F	BLDC15P06	BLDC58-50L	MB057GA240	FL28BL38
Motor maker	EBM-PAPST	Danfoss Compressor	PMDM Minebea	Premotec	Speeder Motion	Fulling Motor
Voltage	24V	24V	12V	24V	50V	24V
Maximum Speed in RPM	3000	3500	12000	12000	5000	13000
Polar Couples	2	2	2	2	2	2
Startup Current in Apk/1000	1000	1000	1000	1000	875	200
Maximum Current Apk/1000	6000	3000	3000	3000	3500	400
Stator Resistance in Ohm/100	38	125	45	30	63	220
Synchronous inductance in Henry/10000	6	12	5	3	17	25
Permanent Magnets Flux in Weber/10000	178	333	42	52	264	30
Current PI - Prop. Coefficient: Kp	18	73	4	10	80	30
Current PI -Integ. Coefficient: Ki	40	80	10	20	215	20
Speed Loop Kp	30	30	50	40	175	120
Speed Loop Ki	400	400	100	300	1400	50
Flux Feedback Gain	400	100	400	400	100	500
Filename in csv format	EBMPAPST_ECI_24.42_24V_3000RPM	DANFOSS_BD35F_24V_3500RPM	MINEBEA_BLDC15_12V_12000RPM	PREMOTEC_BLDC58_24V_12000RPM	SPEEDERMOTION_MB057GA240_5000RPM	FULLING_FL28BL38_13000RPM

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jan 15, 2014		First Edition

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Handling of Unused Pins

Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

- The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

The state of the product is undefined at the moment when power is supplied.

- The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.

In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.

In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

Access to reserved addresses is prohibited.

- The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

- When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

Before changing from one product to another, i.e. to a product with a different type number, confirm that the change will not lead to problems.

- The characteristics of an MPU or MCU in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
 2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
 3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
 4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
 5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.
"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.
Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
 6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
 7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
 8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
 9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
 10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
 11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
 12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.
- (Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.
- (Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.



SALES OFFICES

Renesas Electronics Corporation

<http://www.renesas.com>

Refer to "<http://www.renesas.com/>" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Laviel' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141