YROTATE-IT-RX111

Rotate it! – Motor Control RX111

UM-YROTATE-IT-RX111
Rev.1.00
Oct 15, 2014

Introduction

The Renesas Motor Control Kit called YROTATE-IT-RX111, is based on the RX111 device from the powerful 32-bit RX microcontroller family.

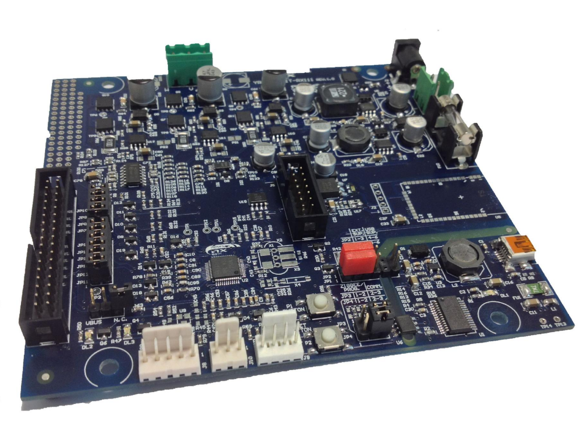
The kit enables engineers to easily test and evaluate the performance of the RX111 in a laboratory environment when driving any 3-phase Permanent Magnet Synchronous Motor (e.g. AC Brushless Motor) using an advanced sensorless Field Oriented Control algorithm. Typical applications for this type of solution are compressors, air conditioning, fans, air extractors, pumps and industrial drives.

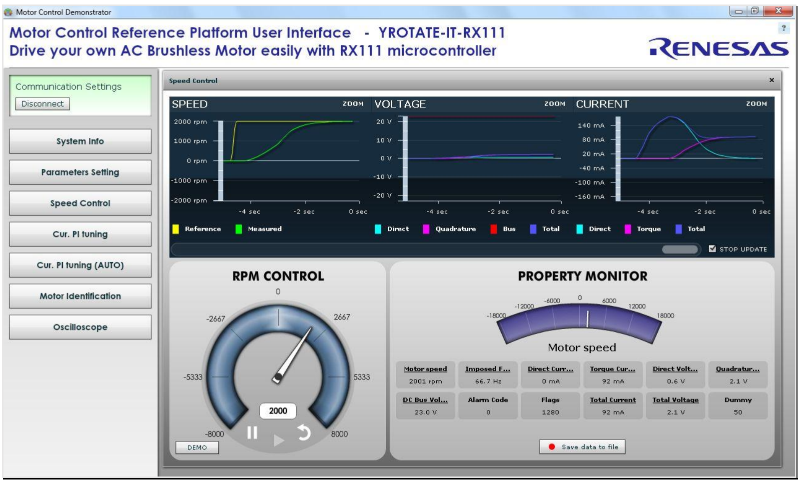
The phase current measurement is done via three shunts which offers a low cost solution, avoiding the need for an expensive current sensor or hall sensor. A single shunt current reading method is also available to secure an even more compacter bill of material.

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. Furthermore, it becomes also possible to select the best switching frequency and control frequency to adapt the control dynamics suitable to the application requirements.

The hardware is designed for easy access to key system test points and for the ability to hook up to an RX111 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-RX111 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.





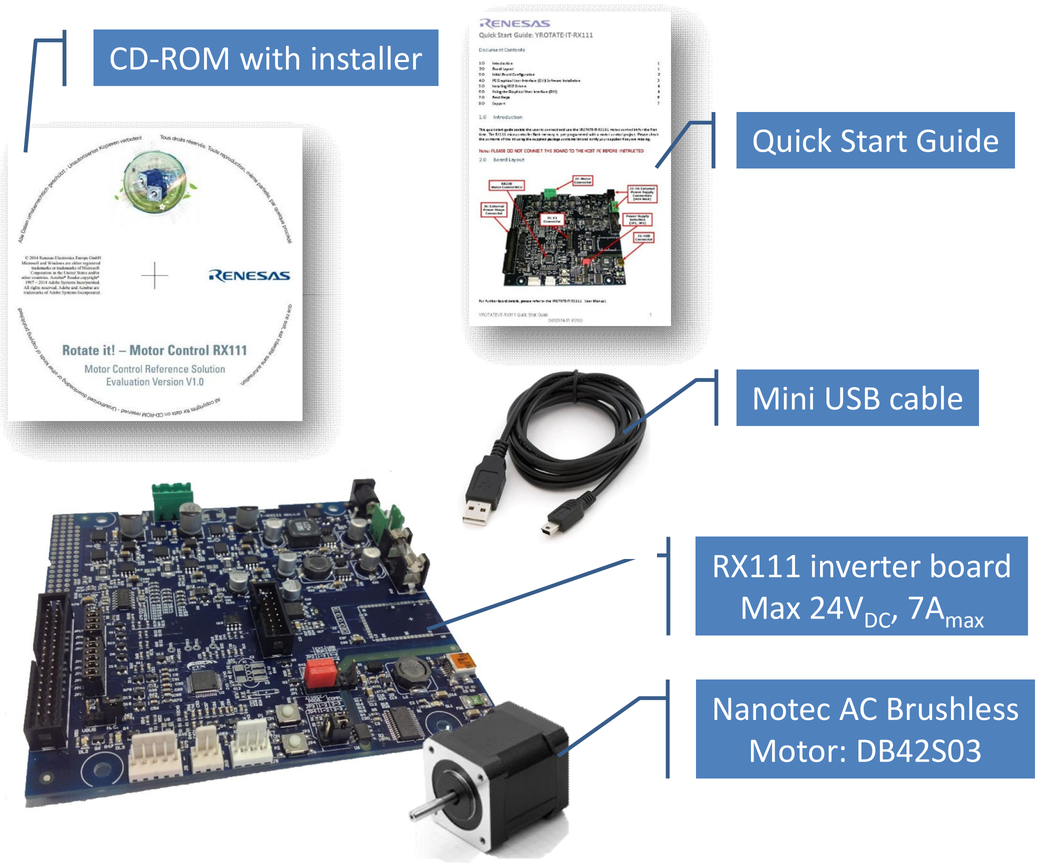
Target Device: RX111

Contents

1. Hardware overview

The inverter kit YROTATE-IT-RX111 is a single board inverter, based on the RX series microcontroller RX111 and includes a low-voltage MOSFETs power stage and a communication stage.

Please find below the content of the YROTATE-IT-RX111 kit:

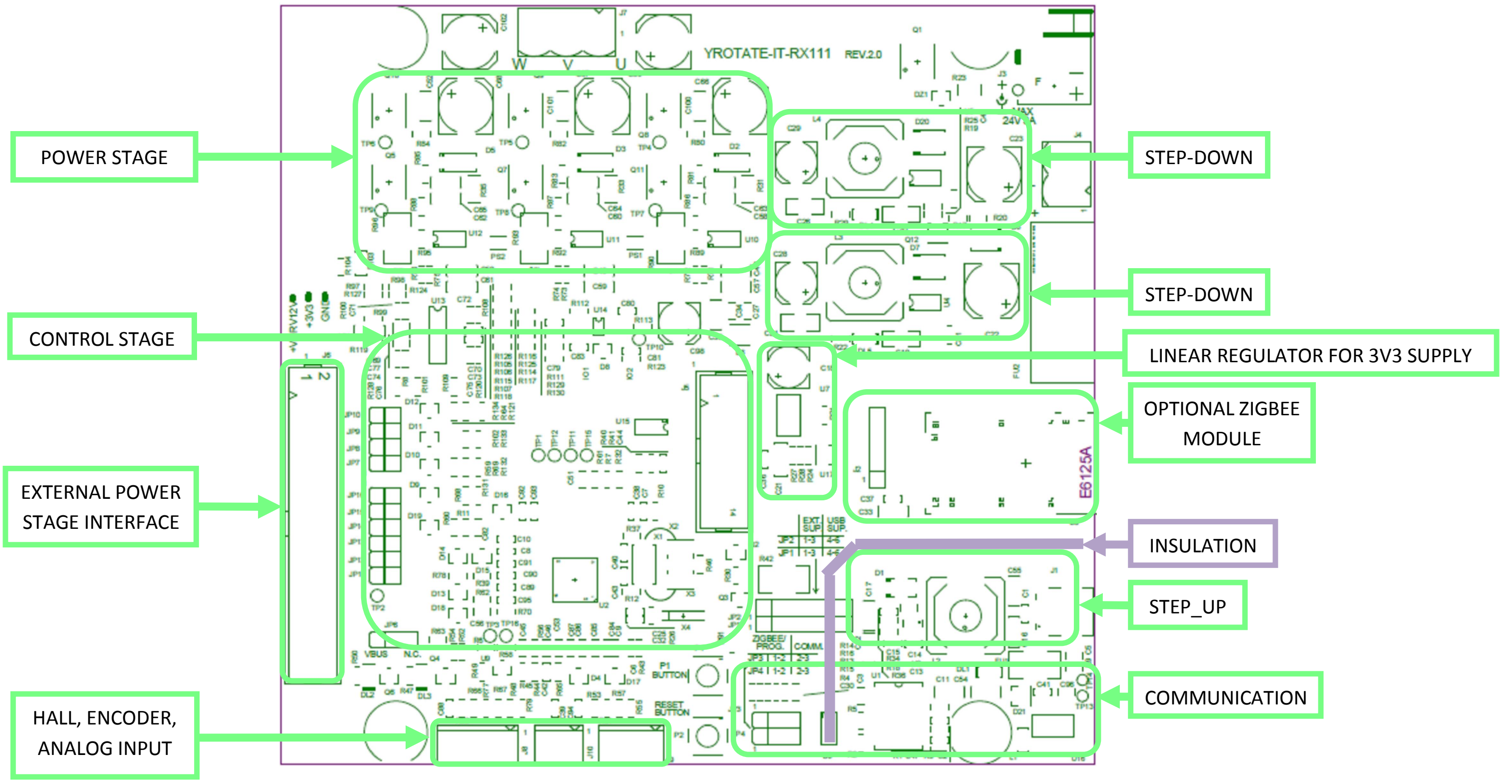


To obtain the maximum flexibility, the inverter reference kit includes:

- 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 Renesas RX111 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 delivering up to 1.5KW at 230V_{AC}
- USB power 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 24V_{DC} in case of external supply) to the power switches drive voltage (12V).
3. A step-down converter and a low dropout linear converter, from the DC bus first to 5V and then to the CPU supply voltage (3.3V).



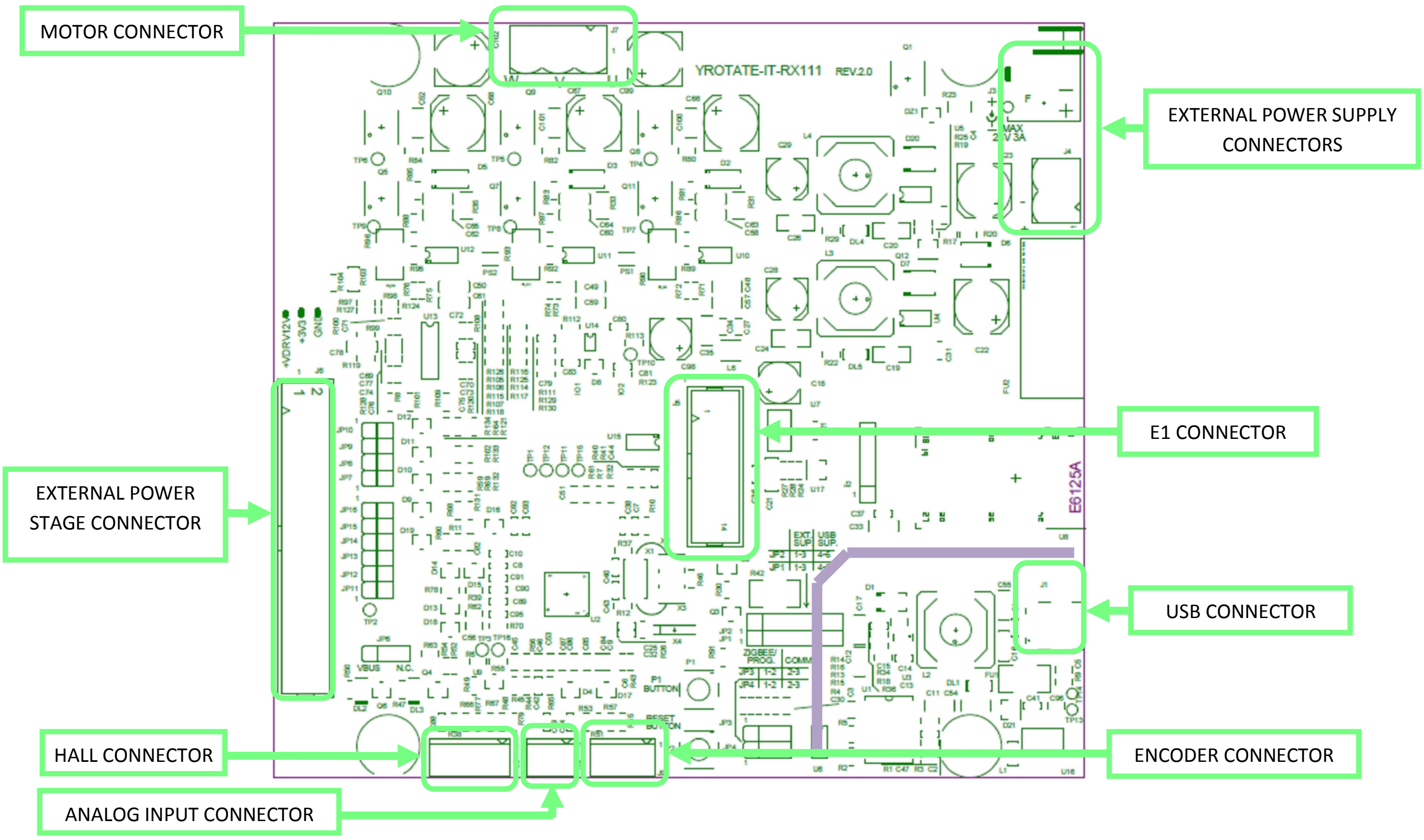
The complete schematics and design files of the inverter board are available on the CD-ROM included in the YROTATE-IT-RX111 development kit.

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 RX111.

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. The schematics and Gerber file of the power stage are available in the CD-ROM delivered with the kit.



3. Power supply selection

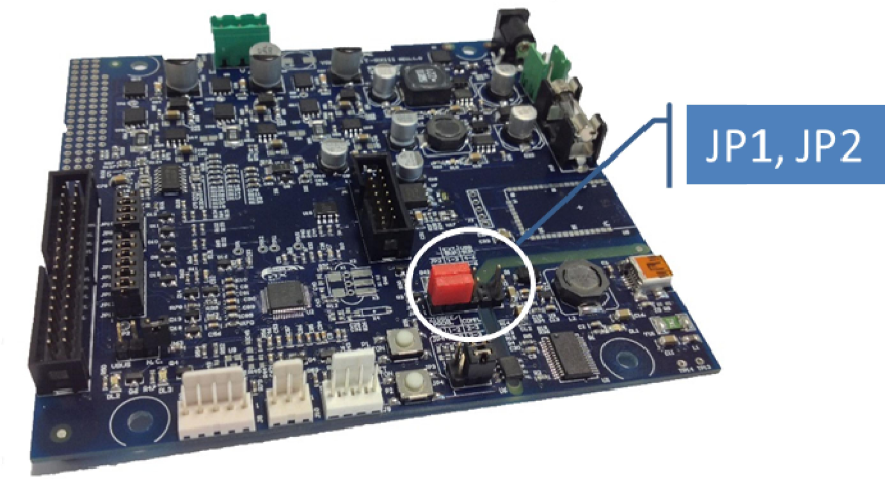
As stated before, there are two ways to supply power to the board.

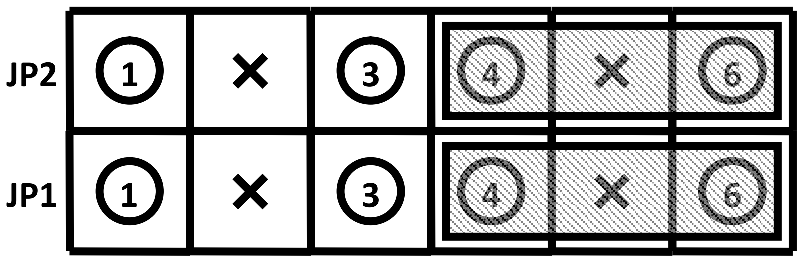
1. The first possibility is to use directly the PC USB supply. In this case the current you can give to the motor is limited by the USB current capabilities.

2. The second possibility is to use an external voltage DC source to supply the board.

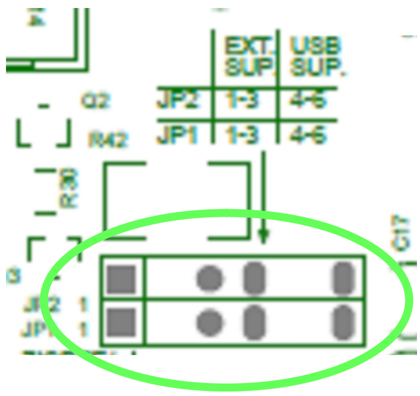
The recommended voltage value is between $12V_{DC}$ and $24V_{DC}$. In this case the communication stage is insulated from the inverter.

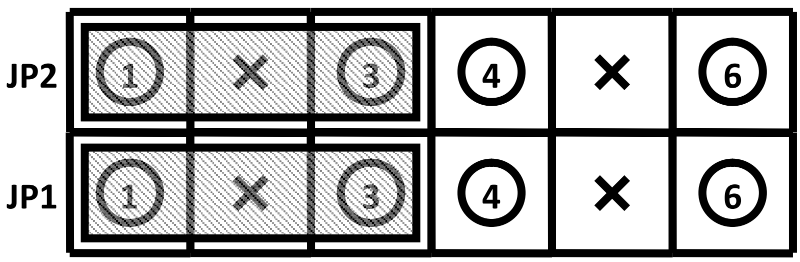
The selection between the two possibilities is made through two jumpers: **JP1** and **JP2**. Please find below the description:





PC USB SUPPLY SELECTION





EXTERNAL SUPPLY SELECTION

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.

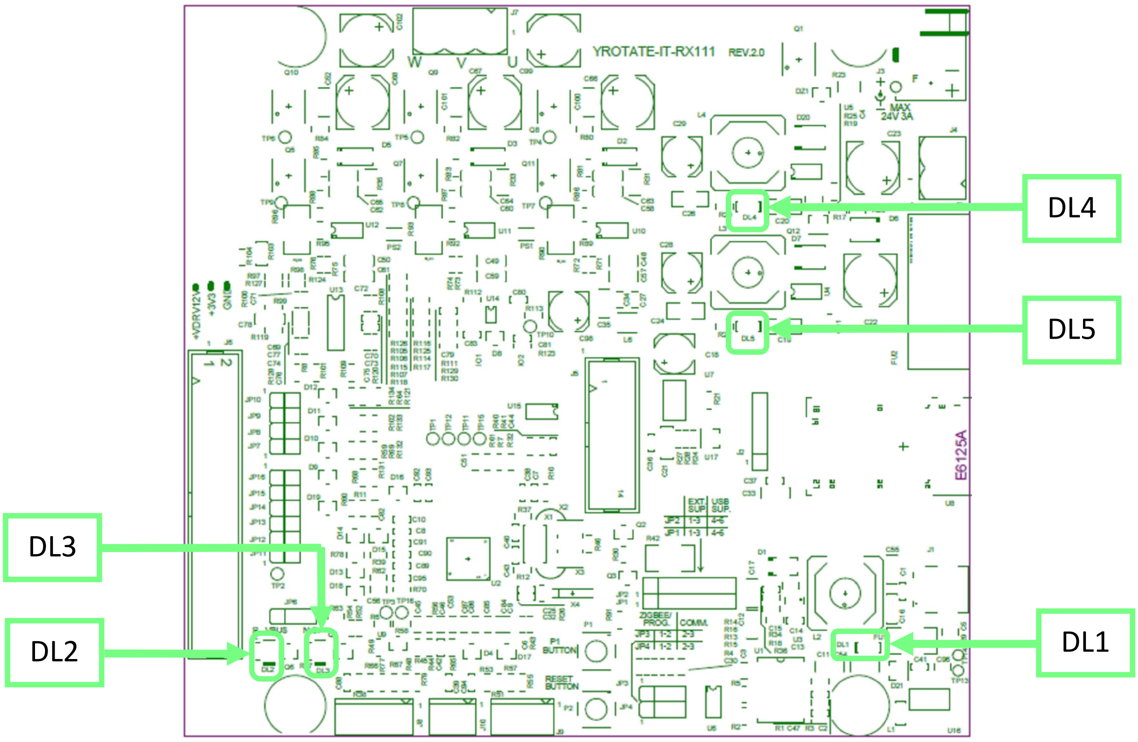
   Please notice that in this case there is no galvanic insulation between the device connected to the USB and the board.

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 functions 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 5V logic power supply; then a low drop-out linear converter produces the 3V3 logic power supply.



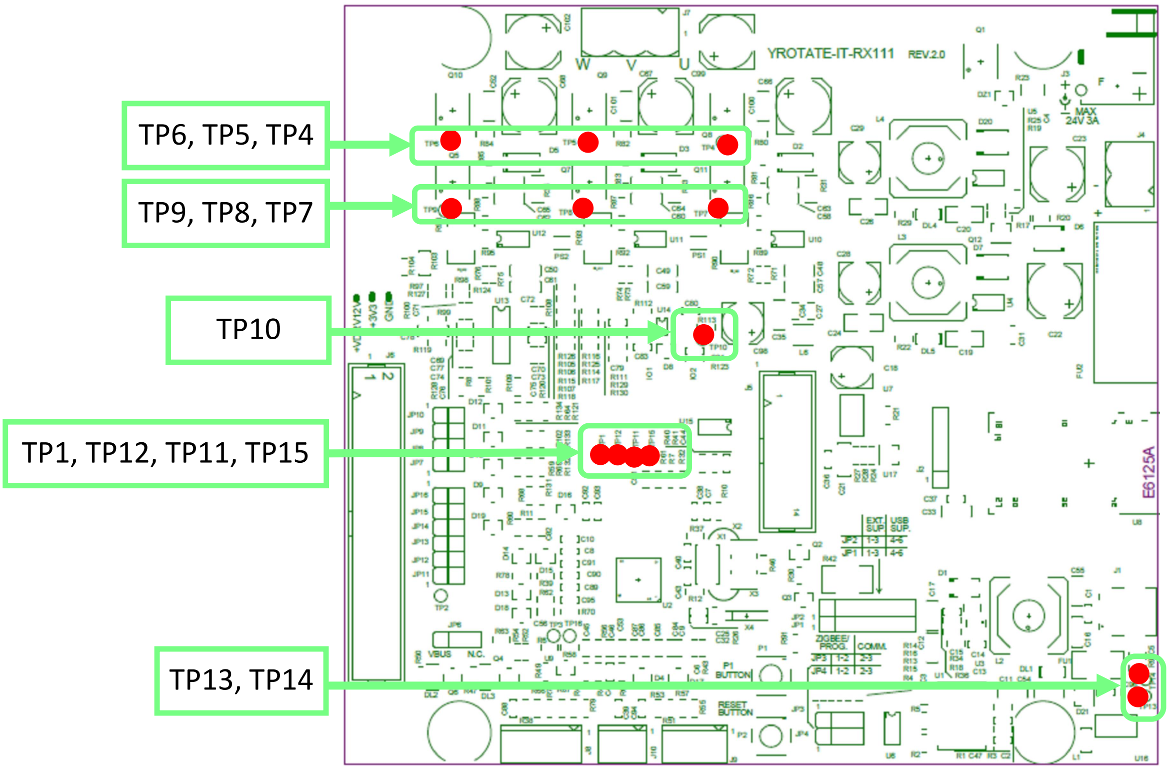
Other LEDs in the board are driven via software, in particular:

- 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 (RX111) 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 behaviour 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.



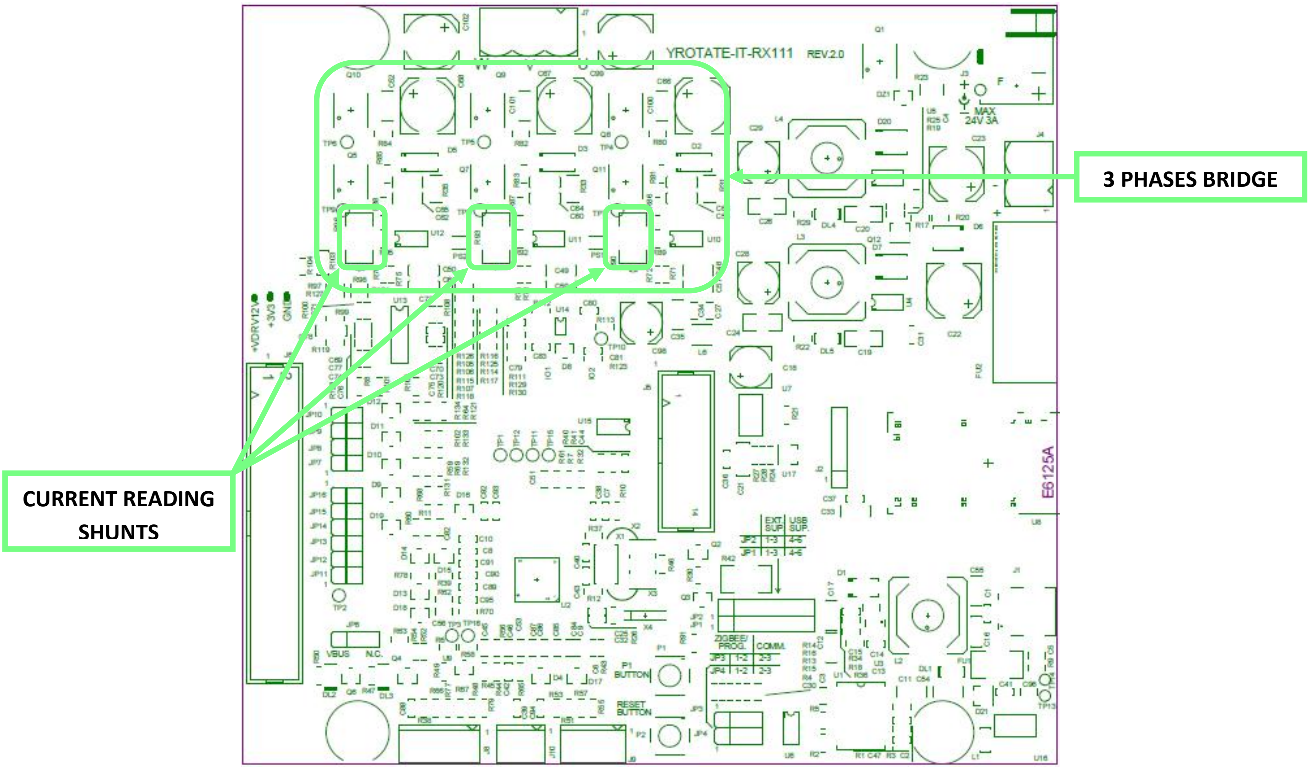
Please find below the description of the test points:

- **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 ports of the microcontroller (TP11 → USB0_DM, TP12 → USB0_DP).
- **TP15:** is connected to the board ground GND.
- **TP1 i**s a filtered PWM output which can be used to visualize the behaviour of internal variables. In the standard software TP1 is the output for the system phase, e.g. estimated rotor position.

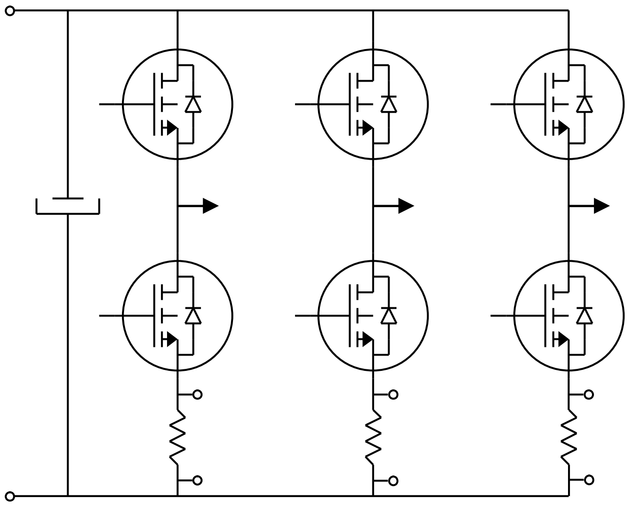
6. Internal power stage description

The power stage is a complete 3-phase bridge composed with discrete low voltage power, high current MOSFETs. The MOSFETs are the Renesas **NP75N04YUK** n-channel power MOSFETs qualified for the automotive applications.

Please refer to the data-sheet available on the CD-ROM for the switches characteristics and to the board schematics for the details on the driving circuit. The maximum current is **75A**, and the maximum voltage is **40V**.



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



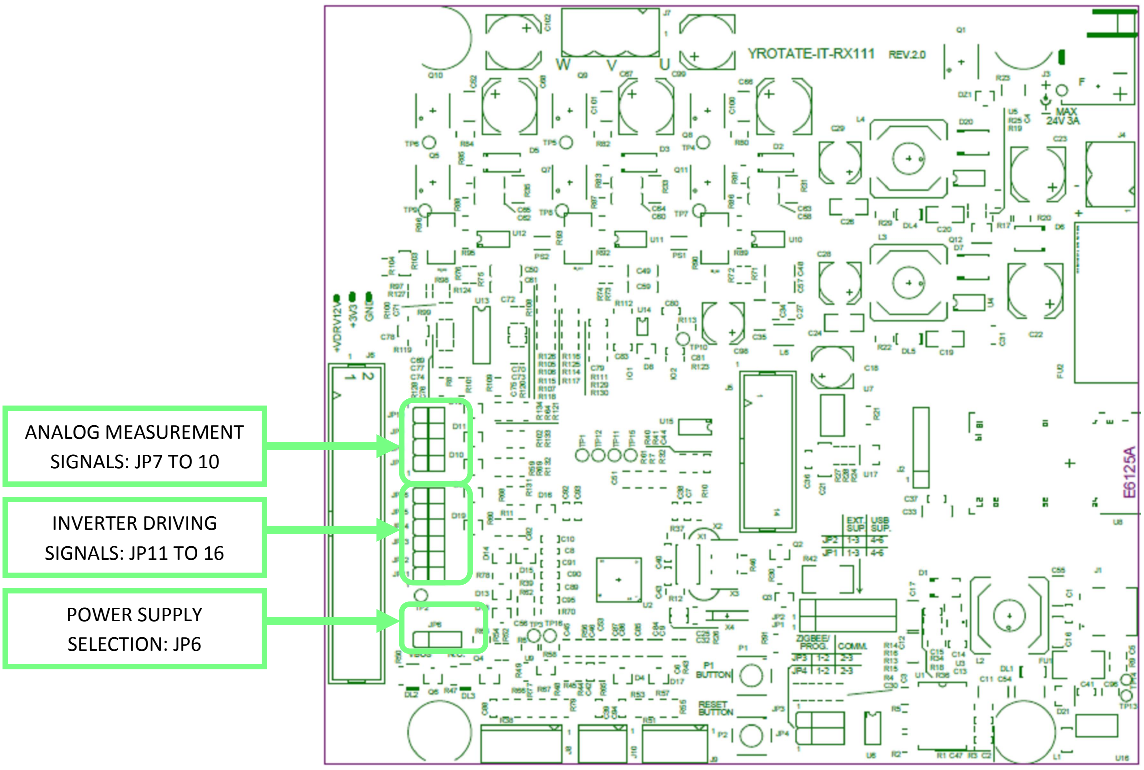
7. Interface with an external power stage

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.

- JP10: if closed, then the internal power stage DC Bus voltage is connected to the opportune A/D converter pin.
- JP7, JP8, and JP9: 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.
- 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;

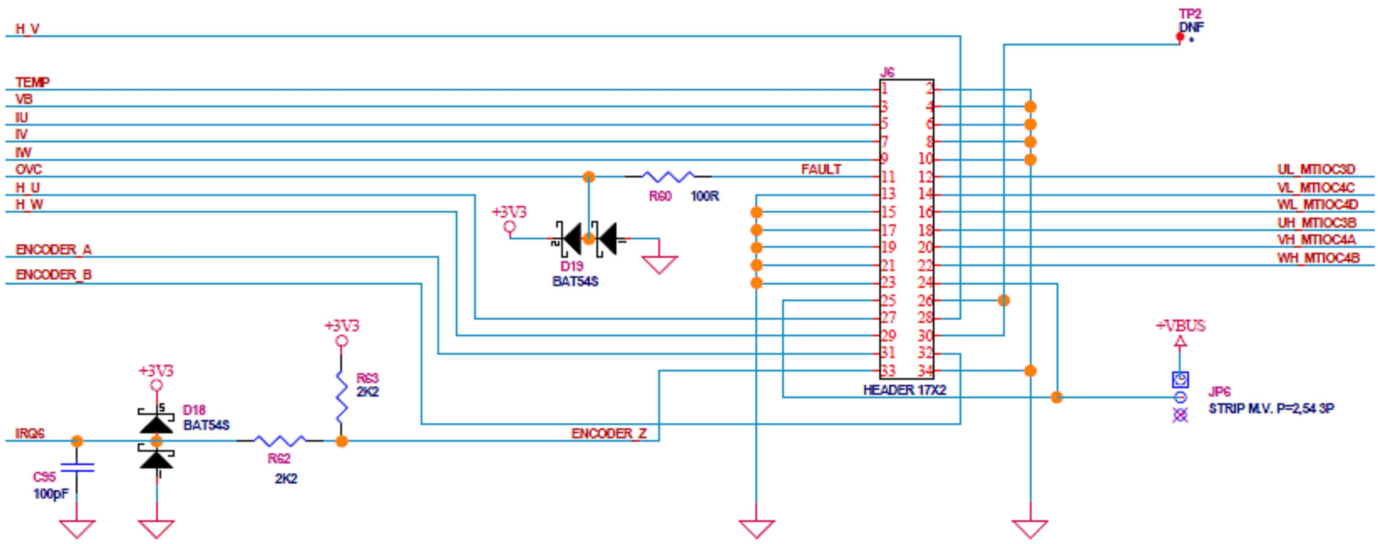
  - If pins 2 and 3 are shorted, then the external power stage low voltage supply cannot be used to supply the board.

Please be careful to take into account the following precautions:

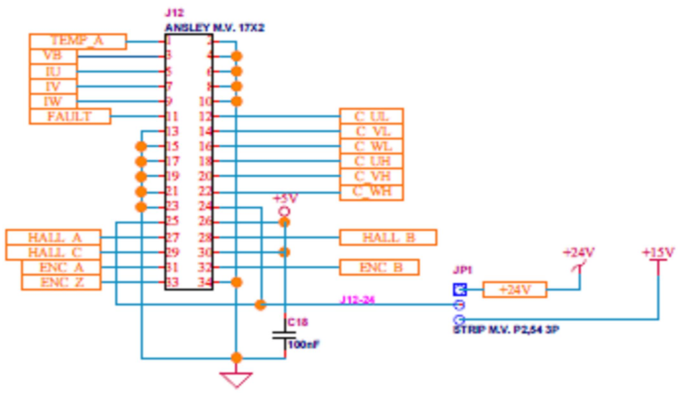
1. Please avoid to connect both the external power stage connector (J6) and to close the jumpers JP6 to JP16: this would produce short circuits between signals coming from different sources.

2. In JP6, chose the configuration with the pins 1 and 2 shorted, when external power supply board is used.

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:

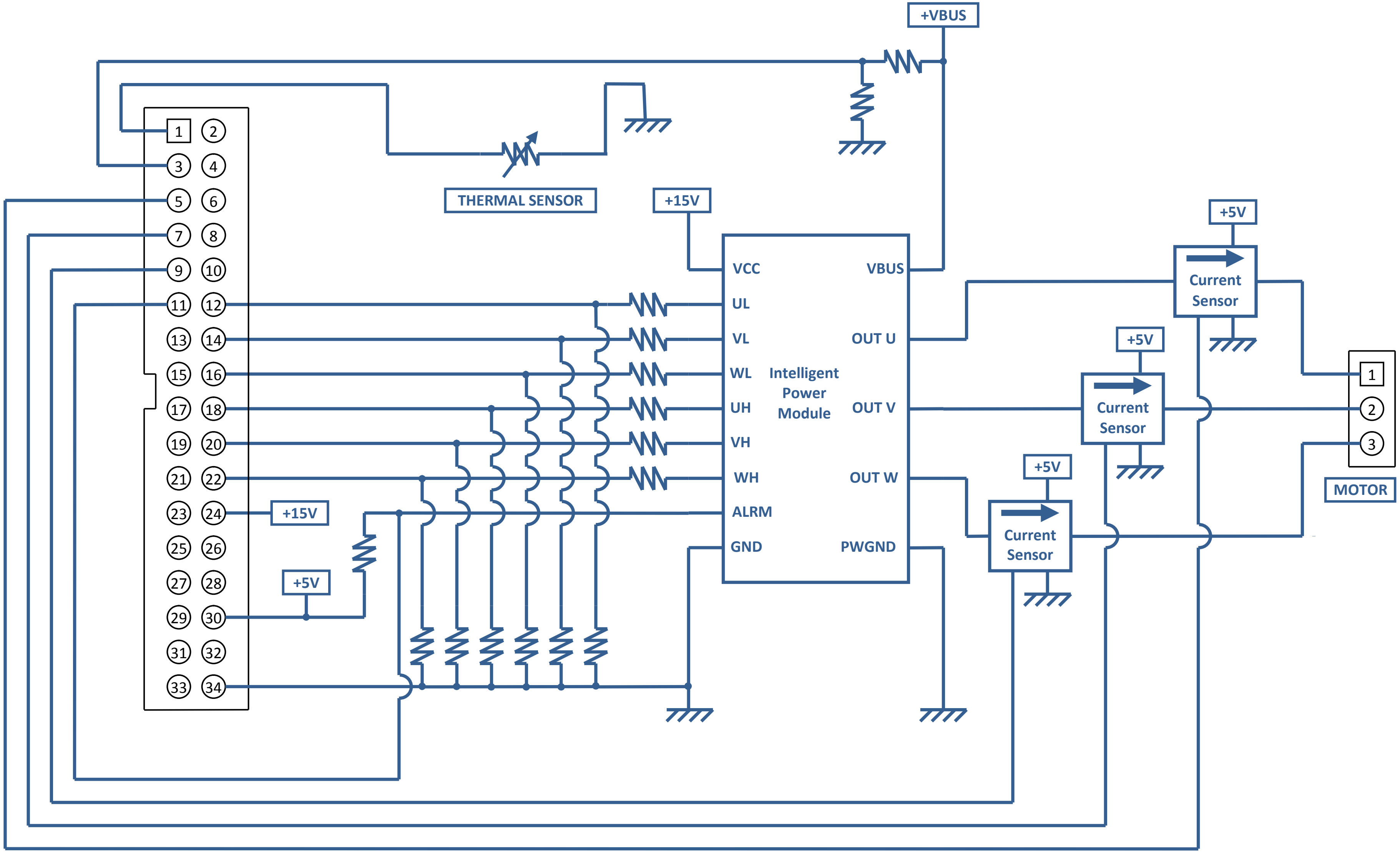
a) The PWM drive signals are directly connected to the microcontroller output pins, and there is no pull-up or pull-down resistor connected, so the polarization has to be done in the power stage. In case of alarm, the microcontroller output pins can be placed in high impedance state, so the external 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); +15 to +24V supply from power board is also used to supply the microcontroller (thanks to jumper JP6 in microcontroller board).

Please refer to the complete schematics for further details that are available on the CD-ROM delivered with the inverter kit.


+VBUS
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
THERMAL SENSOR
+15V
VCC
VBUS
UL
VL
WL
UH
VH
WH
ALRM
GND
Intelligent Power Module
OUT U
OUT V
OUT W
PWGND
+15V
+5V
+5V
Current Sensor
+5V
Current Sensor
+5V
Current Sensor
1
2
3
MOTOR

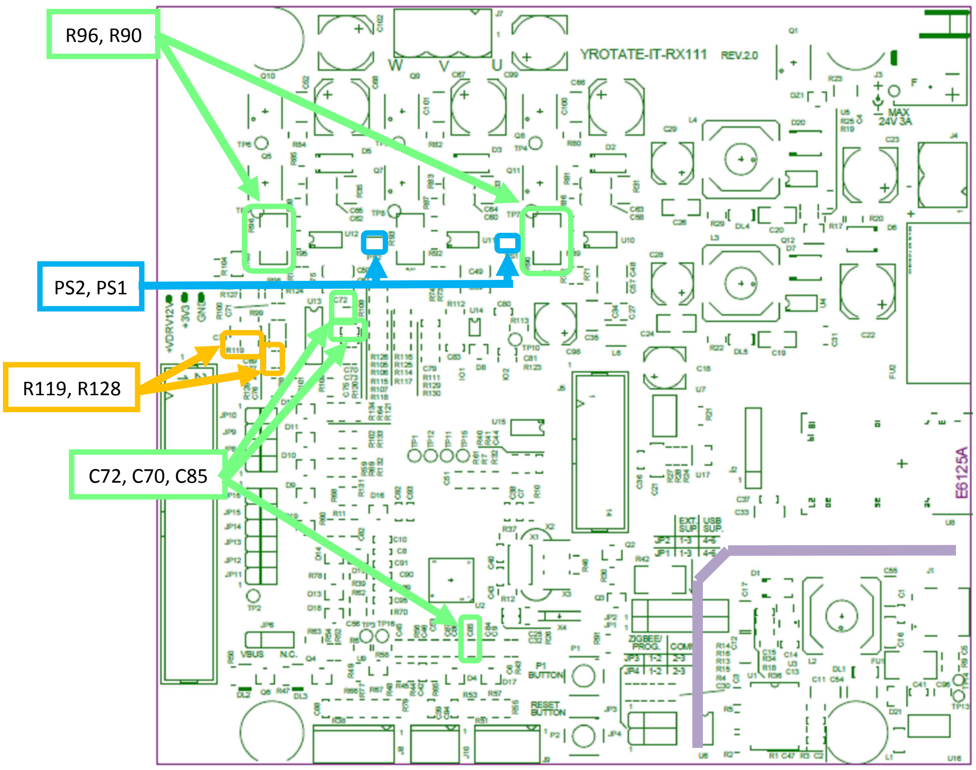
8. Single shunt current reading

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):

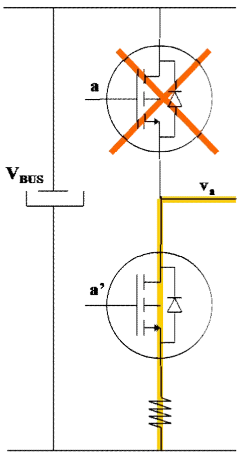
- Remove from the board R90 and R96 (they are the shunts related to the phases U and W).
- Close the soldering points PS1 and PS2 (those soldering points put the three inverter harms in common, below the lower switches and above the shunts).
- Change the value of R119 and R128 from 3K3 to 1K (this will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

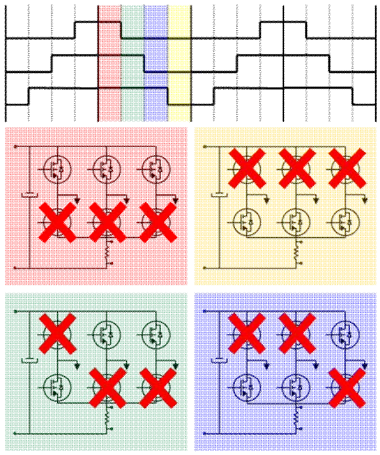
The components involved in the modifications are indicated in the figure below.

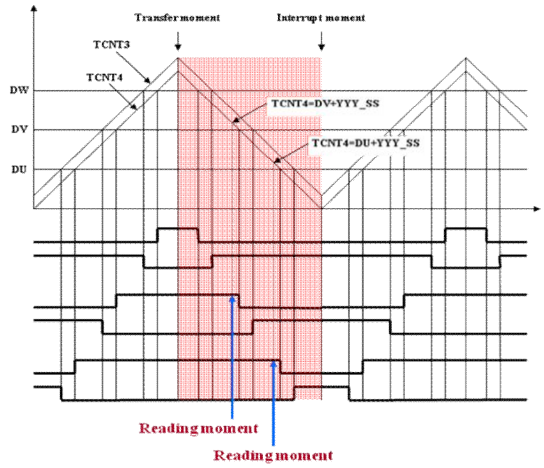


9. Current reading timing in three shunts and single shunt configurations

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 configuration

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.

By default the YROTATE-IT-RX111 kit is delivered in the three shunts configuration.

Single shunt configuration

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:

The software project delivered on the CD-ROM is designed under e²studio environment and only for three shunts configuration. The project is located in the CD-ROM folder:

..\Embedded software\E²Studio Project Source Code

The three shunts software project designed under for e²studio is called: YRotateItRX111_3s_V1

10. CD-ROM contents

The CD-ROM delivered with the YROTATE-IT-RX111 kit contains the resources described in the following table.

CD-ROM Folder	Description of Resources
Auto-tuning Video-Tutorial	Short video explaining how to easily tune any Brushless AC motor in 45 seconds using just the intuitive PC Graphical User Interface
Drivers	Drivers and setup files for the PC Graphical User Interface
YRMCKITRL78G14 Drivers	
YROTATE-IT-RX62T Drivers	
YROTATE-IT-RX220 Drivers	
YROTATE-IT-RX111 Drivers	
Embedded Software	Source files for code flashed by default into the Renesas microcontroller
E2Studio Source Code	
IEC60730 Self Test MCU software	
Manuals	Relevant documentation for the kit, the motor and the MCU
Kit Motor Specifications	
Renesas Datasheets	
Tuned Motors Specifications	
Schematics-Gerber-BoM	Schematics, Gerber files of Bill of Materials for both the main kit and the external power stage (not included as part of the kit)
External Power Stage	
Main Board	

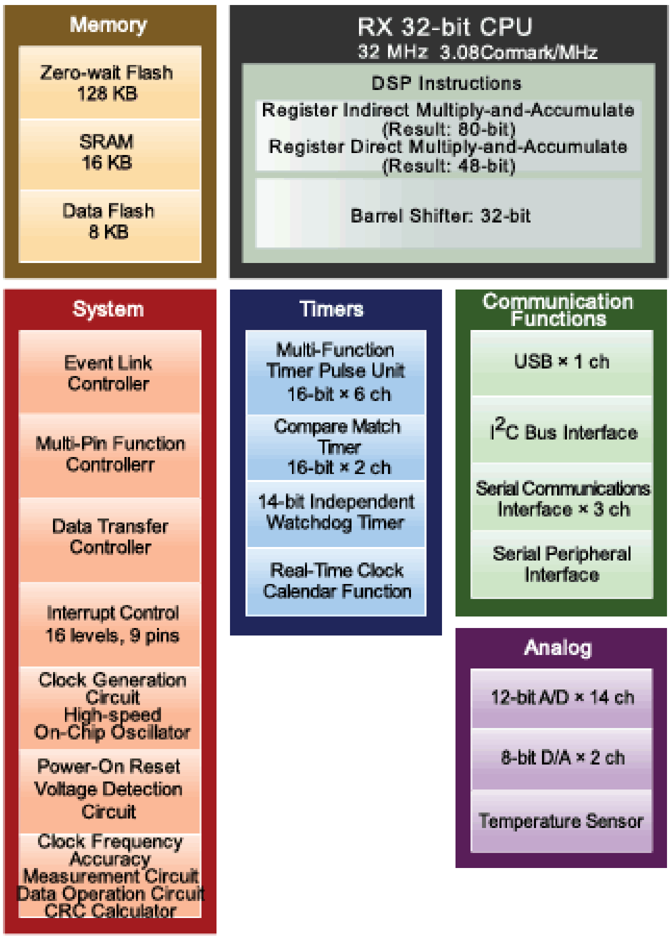
Furthermore, latest update regarding software, documentation is available on-line: www.renesas.eu/motor

11. Microcontroller RX111 short overview

The RX111 Group is a low power entry-level 32-bit microcontroller. The RX111 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 RX111 microcontrollers are as follows:

Item	Description
Wide memory line-up	16KB to 512 KB Flash, 8KB to 64 KB RAM, 8 KB data flash
Enhanced analog functions	12-bit A/D converter x max. 14 channels, D/A converter, comparator x 2 channels
Enhanced safety functions	Clock frequency Accuracy measurement Circuit (CAC), Data Operation Circuit (DOC), 14-bit Independent Watchdog Timer (IWDTa), CRC calculator (CRC)
Peripheral functions with special features	Event Link Controller (ELC), Multi-function Pin Controller (MPC)
Broad package line-up	The line-up covers from 36 to 64 pins, QFN, LGA, or QFP packages available

Please find below the RX111 microcontroller block diagram of the RX111:

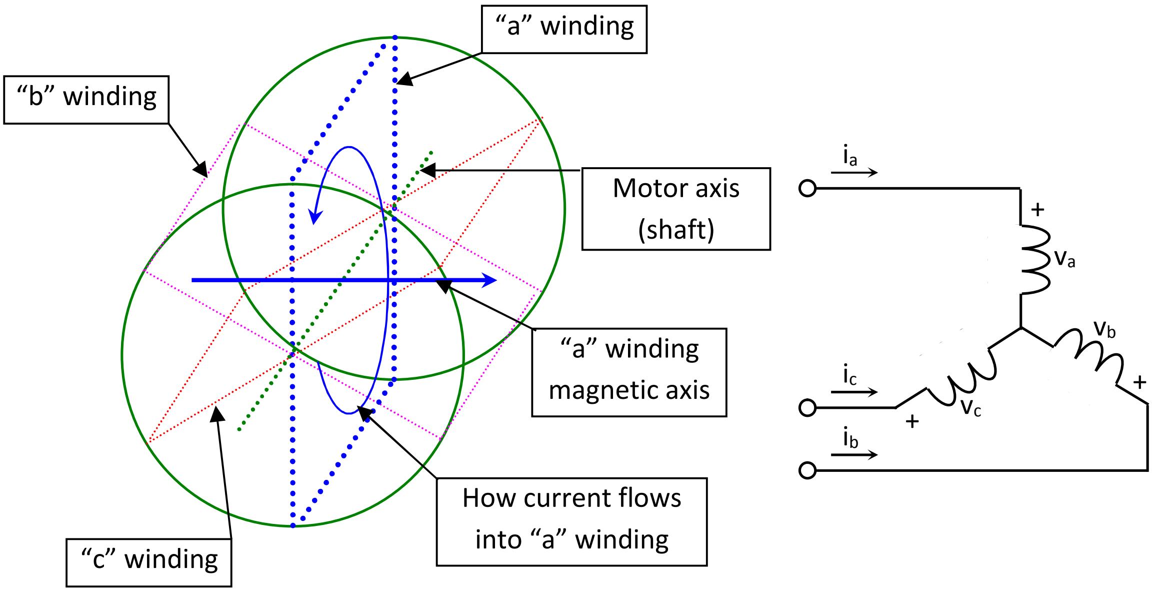


12. Permanent Magnets Brushless Motor model

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.

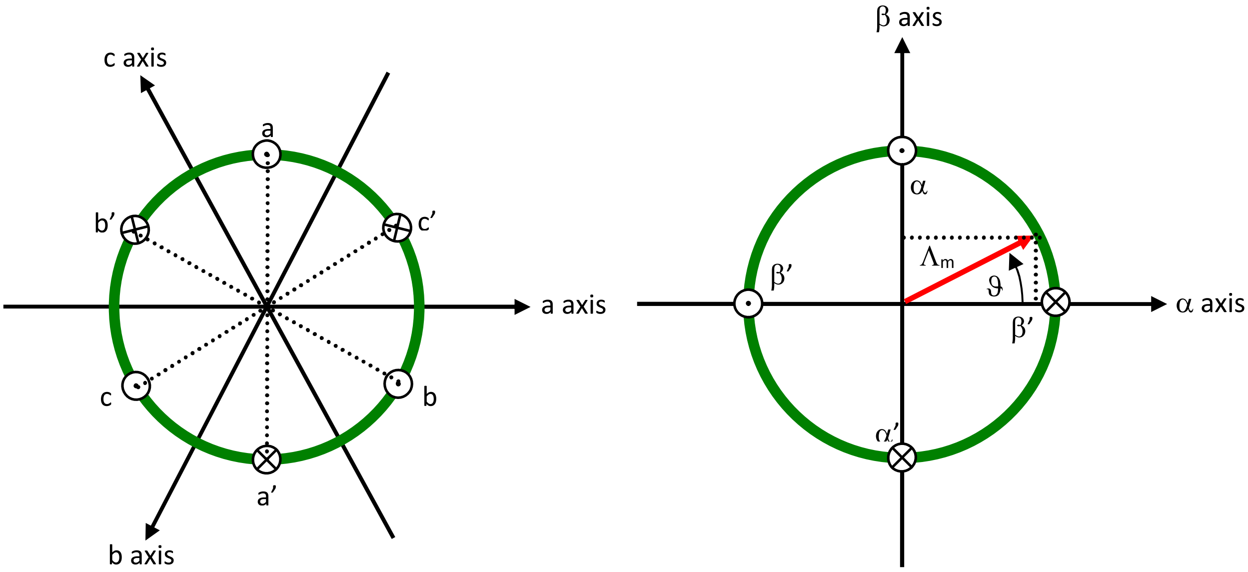
If we apply three voltages $v_a(t)$, $v_b(t)$, $v_c(t)$ to the stator windings, the relations between phase voltages and currents are:

$$v_a = R_S i_a + \frac{d\lambda_a}{dt}$$
$$v_b = R_S i_b + \frac{d\lambda_b}{dt}$$
$$v_c = R_S i_c + \frac{d\lambda_c}{dt}$$

- 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: ϑ(t) = ωt) the flux linkages derivatives can be calculated, and we obtain:

$$v_a = R_S i_a + L\frac{di_a}{dt} - \omega\Lambda_m \sin(\vartheta)$$
$$v_b = R_S i_b + L\frac{di_b}{dt} - \omega\Lambda_m \sin(\vartheta - 2\pi/3)$$
$$v_c = R_S i_b + L\frac{di_b}{dt} - \omega\Lambda_m \sin(\vartheta - 4\pi/3)$$

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:

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

After performing the derivation:

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

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).

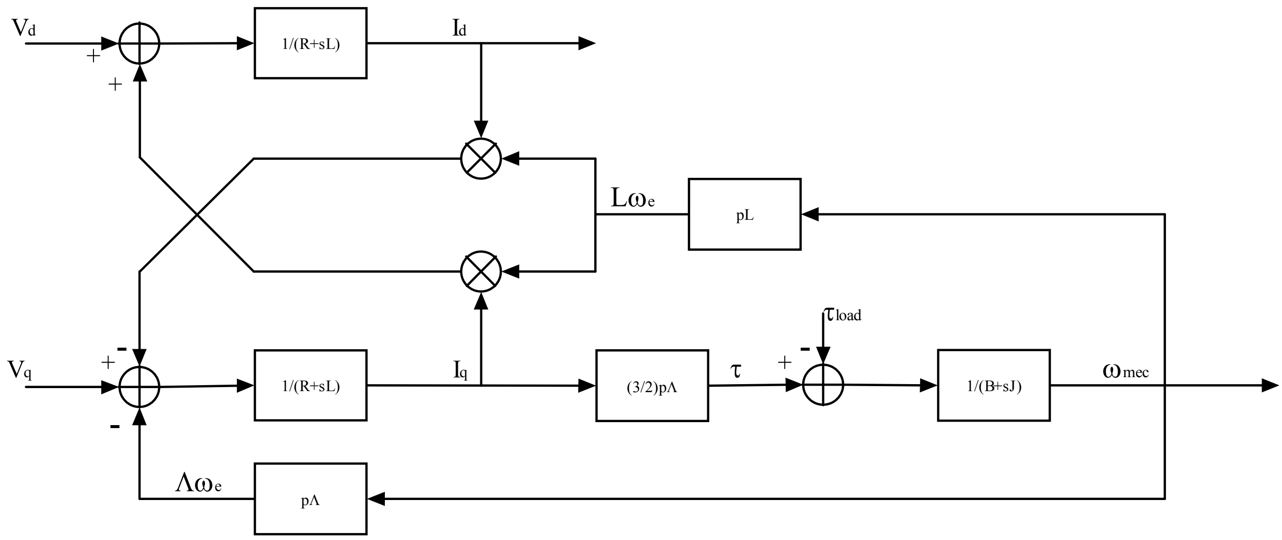
The reference frame transformations from the (α,β) system to the (d, q) system depends on the instantaneous position angle ϑ

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:

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

Furthermore:

$$\Lambda_m \cos(\vartheta) = \lambda_\alpha - L i_\alpha$$
$$\Lambda_m \sin(\vartheta) = \lambda_\beta - L i_\beta$$

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

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

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

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

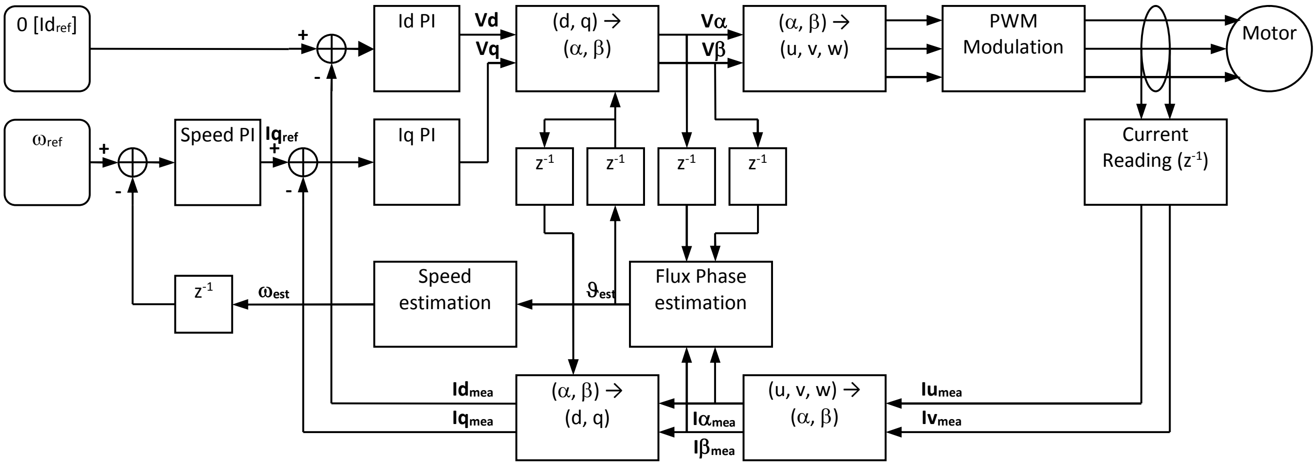
The system speed ω can be obtained as the derivative of the angle ϑ.

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

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.

13. 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.

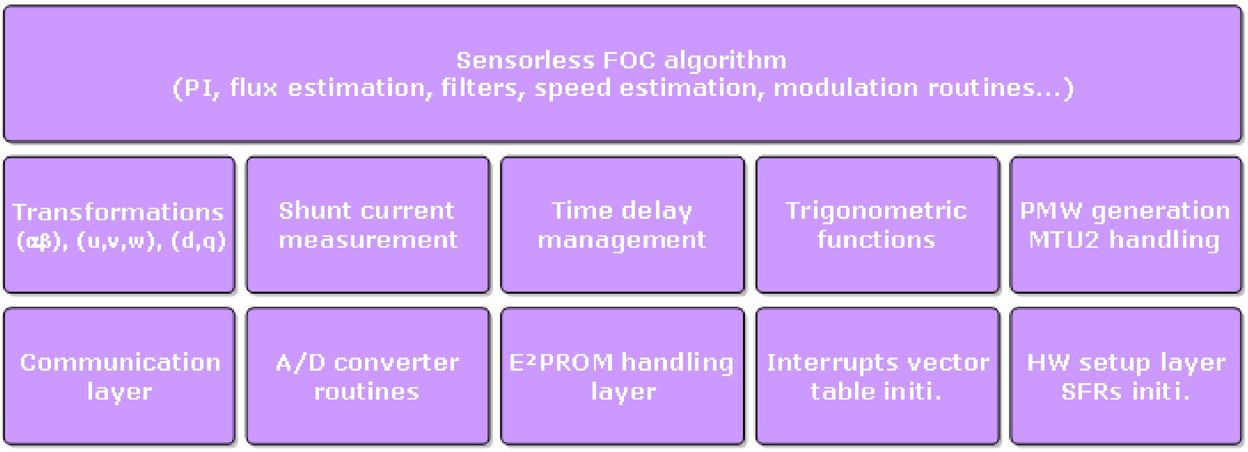
14. Software description

The software delivered in the YROTATE-IT-RX111 kit, previously described, is working on the RX111 microcontroller clocked at 32MHz and running at 3V.

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 one.

Please find below detailed information related to the software blocks of the motor control embedded software:

SOFTWARE BLOCKS



The complete software uses the following resources in the three shunts configuration, including the communication interface, the board management, the auto-tuning algorithm, the linearization of the power stage, etc.

- **FLASH memory usage: 30KB and RAM memory usage: 3KB**

The default program is called “YRotateItRX111_3s_VX” and is located on the CD-ROM:

.\Embedded software\E²Studio Project Source Code

The embedded software is by default set to 8KHz sampling frequency, i.e. 125µs for the sampling period and the PWM frequency is set to 16KHz. Such parameters can be modify dynamically using the PC GUI without recompiling the overall project changing the parameters below and resetting the board:

19	SAM_FRE_DEF	Set the sampling frequency [Hz] of the control loop
20	F_RATIO_DEF	Set the ratio between the PWM frequency and sampling frequency, e.g. if 8000 is set in the parameter #19 and 2 in the parameter #20, the PWM frequency is 16KHz.

The parameter #19 is setting the control loop speed. If 8KHz is selected by entering the value “8000”, the PWM frequency can be set to four different values depending on the motor and the applications either 8KHz, 16KHz, 24KHz or 32KHz.

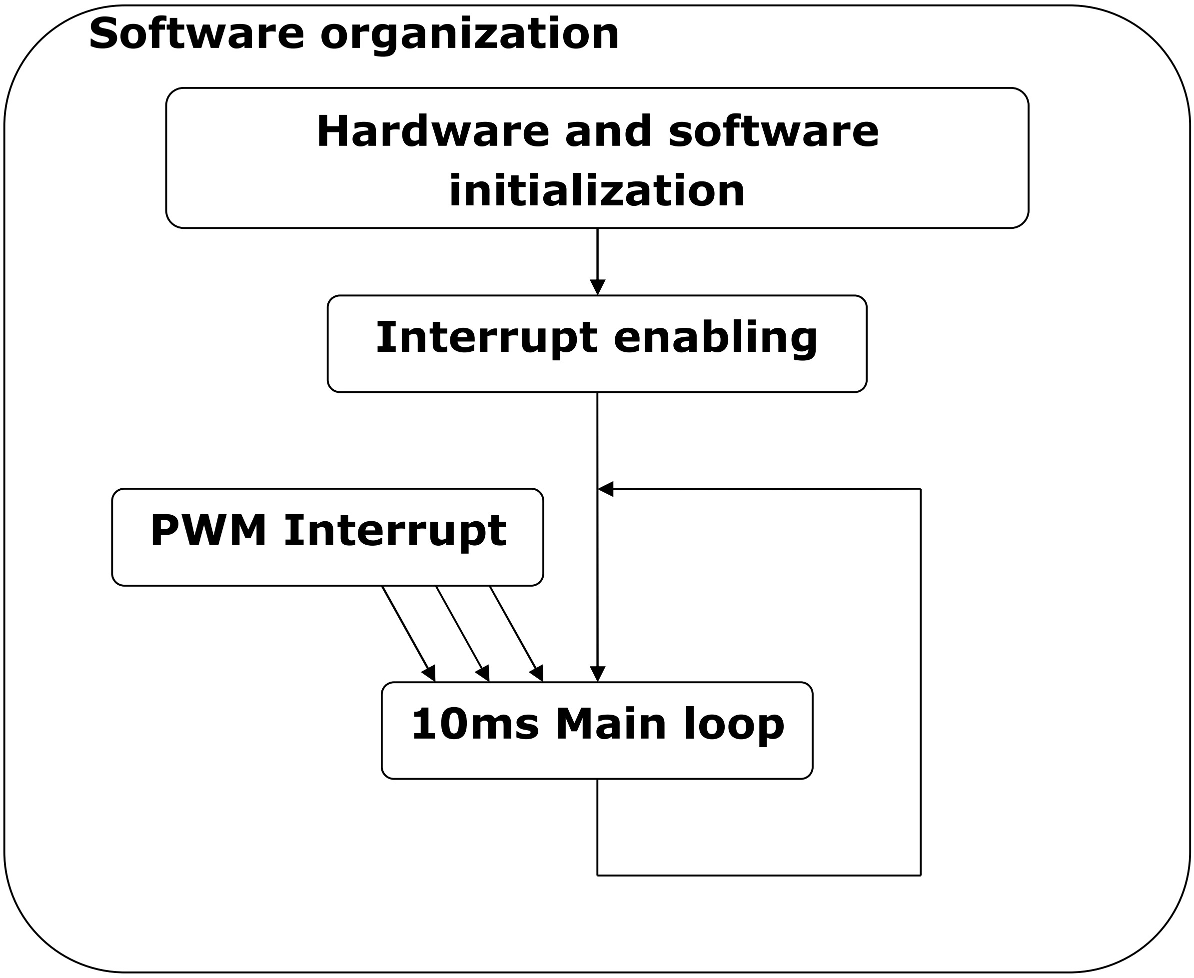
Furthermore, the execution time of the motor control interrupt including all the communication management, the Speed PI block, the Current PI block and the complete vector control algorithm and the auto-calibration mechanisms: is **67µs.**

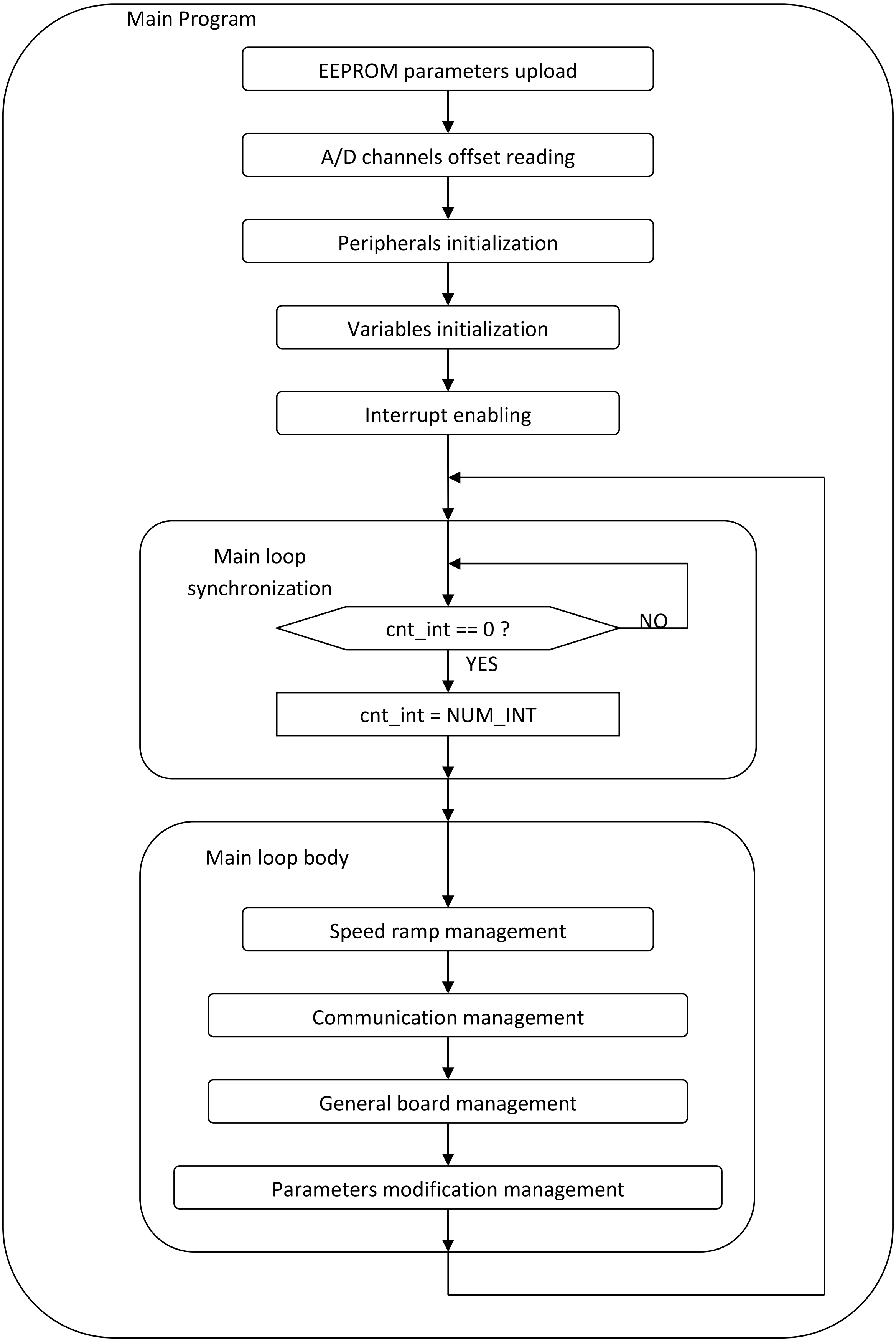
By enabling the optimization, the execution time of the software is below **55µs with the clock running at 32MHz**.

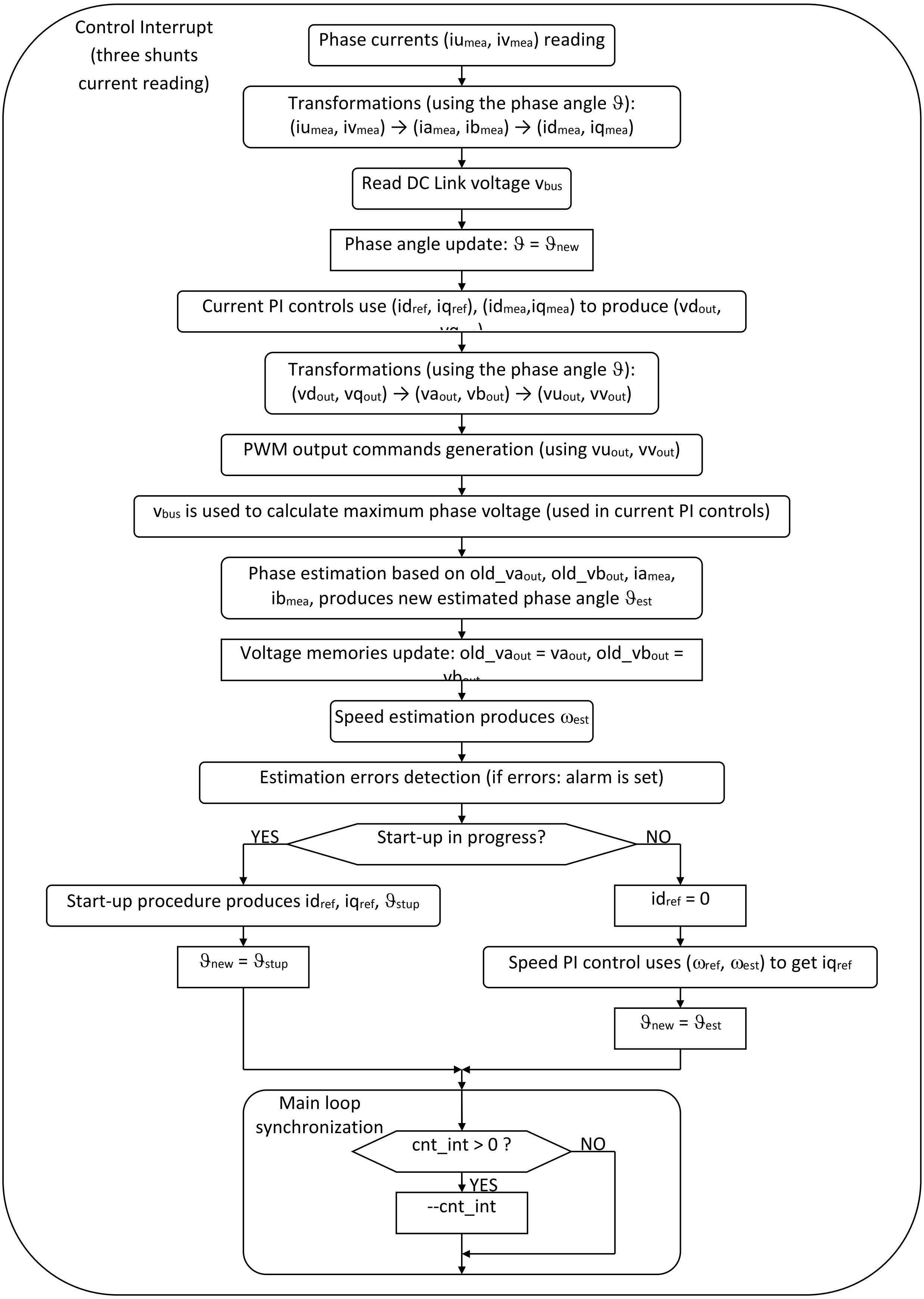
The development tools used are e²studio tool chain: **RX Family C/C++ Compiler V2.01.00.07**

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

Please fin below the flowchart for the main loop:




Main Program
EEPROM parameters upload
A/D channels offset reading
Peripherals initialization
Variables initialization
Interrupt enabling
Main loop synchronization
cnt_int == 0 ?
NO
YES
cnt_int = NUM_INT
Main loop body
Speed ramp management
Communication management
General board management
Parameters modification management

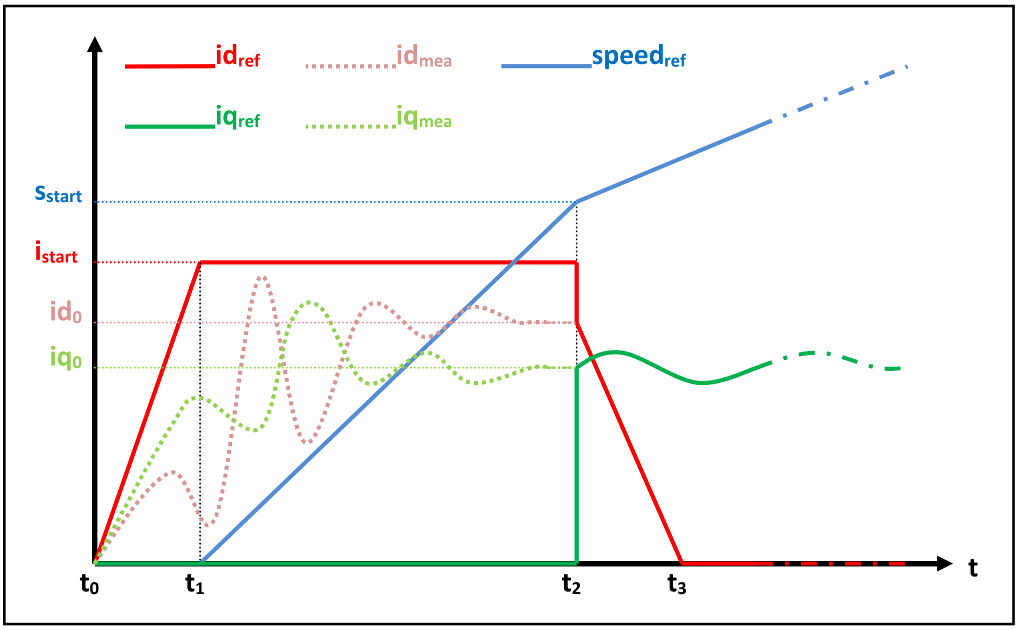

Control Interrupt
(three shunts
current reading)
Phase currents (iumea, ivmea) reading
Transformations (using the phase angle ϑ):
(iumea, ivmea) → (iamea, ibmea) → (idmea, iqmea)
Read DC Link voltage vbus
Phase angle update: ϑ = ϑnew
Current PI controls use (idref, iqref), (idmea,iqmea) to produce (vdout,
Transformations (using the phase angle ϑ):
(vdout, vqout) → (vaout, vbout) → (vuout, vvout)
PWM output commands generation (using vuout, vvout)
vbus is used to calculate maximum phase voltage (used in current PI controls)
Phase estimation based on old_vaout, old_vbout, iamea,
ibmea, produces new estimated phase angle ϑest
Voltage memories update: old_vaout = vaout, old_vbout =
Speed estimation produces ωest
Estimation errors detection (if errors: alarm is set)
YES
Start-up in progress?
NO
Start-up procedure produces idref, iqref, ϑstup
idref = 0
ϑnew = ϑstup
Speed PI control uses (ωref, ωest) to get iqref
ϑnew = ϑest
Main loop
synchronization
cnt_int > 0 ?
NO
YES
--cnt_int

15. Start-up procedure – Embedded software

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.

a) 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: id_{ref}, iq_{ref} and $speed_{ref}$ are set to zero.

b) From the moment t_0, while the iq_{ref} and the $speed_{ref}$ are maintained to zero, id_{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, id_{mea} and iq_{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, id_{mea} and iq_{mea} have no physical meaning. Even if they are not shown in the graph, the applied voltages are subjected to the same treatment (vd_{mea} and vq_{mea} are calculated in the algorithm).

c) At t = t_1, while iq_{ref} is maintained to zero and id_{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 $\vartheta_a(t)$ is obtained simply by integration of $speed_{ref}$; in the meanwhile, the phase estimation algorithm begins to align with the real system phase.

Furthermore id_{mea} and iq_{mea} begin to be similar to the real flux and torque components of the current. The real components are supposed to be id_0 and iq_0 (those values are obtained applying a low-pass filter to id_{mea} and iq_{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 ϑ_{est}.

The current references are changed to the values id_0 and iq_0, and all the PI controllers are initialized with these new values. The speed PI integral memory is initialized with the value iq_0, while the current PI integral memories are initialized with the analogous voltage values vd_0 and vq_0, obtained from vd_{mea} and vq_{mea}.

e) After $t > t_2$, the normal control is performed, based on the estimated phase ϑ_{est}; the speed reference is increased with the classical ramp; the id 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 iq current reference is obtained as output of the speed PI controller.

16. Reference system transformations in details

Find below the detailed equations used for the coordinates transformations in the embedded software for the RX111 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)

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

17. 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:

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

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

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

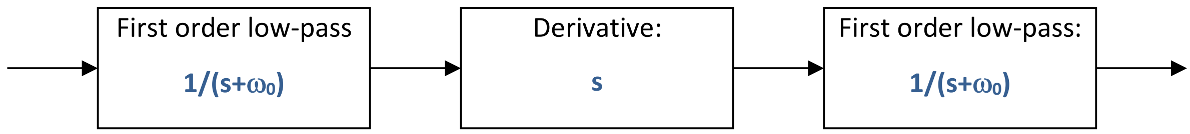
$$\omega = \frac{d}{dt} \vartheta(t)$$

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:

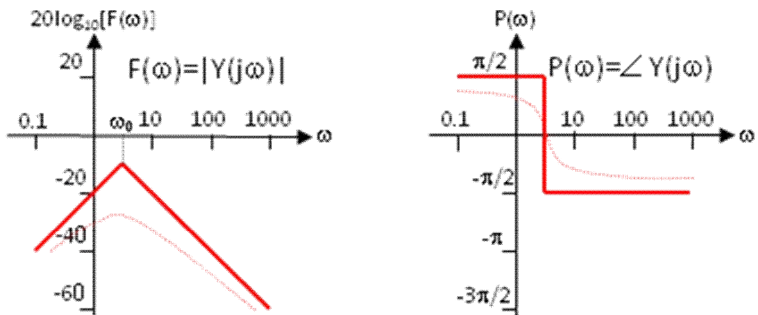
1. To use a so-called "approximated integration", which means that instead of using an integral (1/s), a special transfer function is chosen, which is very similar to the integral in certain conditions.

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:



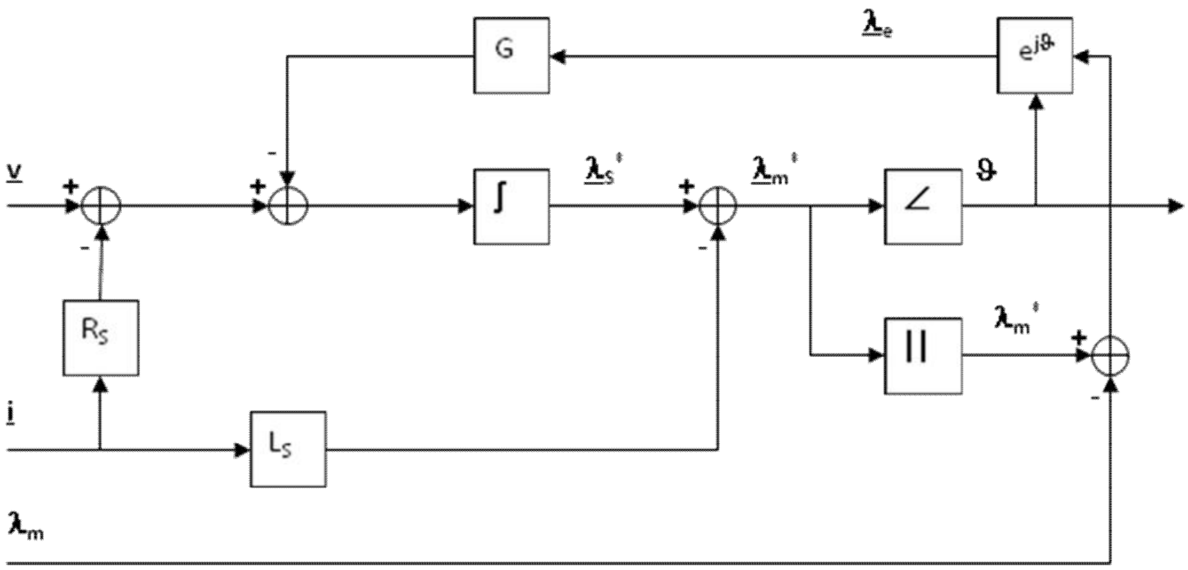
$$Y(s) = \frac{s}{(s+\omega_0)^2}$$



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

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 default embedded software delivered on the YROTATE-IT-RX111 kit, 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.

That's why, a specific parameter related to the estimator can be tuned via the PC GUI. It is the parameter n°18 "App. FE Time Constant" in ms. which is the Filter time constant used in the approximate integration flux estimation method. Please find below more technical details about it.

Regarding the Filter time constant, the BEMF integral is approximated with a function composed by two low pass filters and a derivative. In discrete time domain, the expression for the low pass filter is:

y(n)=((k-1)/k)*y(n-1)+(1/k)*x(n)
where (k*Tc) is the so called time constant of the filter and Tc is the sampling period

In the new software release for RX111: the parameter to be specified is the time constant on the filter in ms, which is much more flexible and also more immediate to understand.
In fact the cut-off frequency of the filter is directly:

1/(2*pi*time_const[s]) = 1000/(6.28*parameter_value[ms]).

Internally k is calculated as 1000*parameter_value[ms]/Tc, and the filter is implemented performing a division, so there's no restrictions on the value and the sampling frequency is now variable.
For example, if the parameter_value = 32, so the time_constant[s]=0,032 and the cutoff_freq=4,97Hz, internal k=0,032*sampling_freq.

That's why, at 8KHz, please find below some calculation about the possible value to be used to tune the estimators:

Para. 18 in ms App. FE Time Constant	Cut-off Frequency	RX220 kit previous implementation
128	1.25Hz	10
64	2.5Hz	9
32	5Hz	8
16	10Hz	7

Obviously high values of the cut-off frequency produce poor behaviour at low speed and good behaviour at high speed, and vice-versa.

18. 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, 2pi) is represented with the interval [0, 65536), with the resolution of 2pi/65536 rad.

Angle[internal_angle_unit] = KA * Angle[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.

Internal_sin(a[internal_angle_unit]) = NORMVAL * sin(a[rad]), NORMVAL = 16384

-NORMVAL <= Internal_sin() <= NORMVAL (the same for Internal_cos())

Time

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

Time[internal_time_unit] = KT * Time[sec]

KT = Fs (Fs = sampling_frequency = 1 / Ts)

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.

Omega[internal_angular_velocity_unit1] = KO1 * Omega[rad / sec] =

= (KO1 * KT / KA) * Angle[internal_angle_unit] / Time[internal_time_unit]

Since we want:

Omega[internal_angular_velocity_unit1] =

= NORMVAL * Angle[internal_angle_unit] / Time[internal_time_unit]

(→ Omega[internal_angular_velocity_unit1] = NORMVAL * (Angle(n) - Angle(n - k)) / k**)**

We obtain:

KO1 = NORMVAL * KA / KT = NORMVAL * 65536 / (2 * pi * 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:

BASE_SPEED_R_S = 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 (MAX_OMEGA_R_S = 2pi * MAX_FRE_HZ). The second representation is the following:

Omega[internal_angular_velocity_unit2] = KO2 * Omega[rad / sec]

KO2 = NORMVAL / BASE_SPEED_R_S

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

Omega[internal_angular_velocity_unit2] = (KO2 / KO1) * Omega[internal_angular_velocity_unit1]

KO2 / KO1 = (2 * pi * Fs) / (65536 * MAX_OMEGA_R_S)

KO1 / KO2 = 65536 * MAX_FRE_HZ / 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: Vout_pk = (2/3)*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:

BASE_VOLTAGE_VOLT = (2 ^ K) * MAX_VOLTAGE_VOLT, with K 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:

Voltage[internal_voltage_unit] = KV * Voltage[Vol]

KV = NORMVAL / BASE_VOLTAGE_VOLT

Current

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

BASE_CURRENT_AMP = MAX_CURRENT_AMP

It is represented with NORMVAL = 16384:

Current[internal_current_unit] = KI * Current[Amp]

KI = NORMVAL / 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:

BASE_IMPEDANCE_OHM = BASE_VOLTAGE_VOLT / BASE_CURRENT_AMP

The internal representation is:

Impedance[internal_impedance_unit] = KZ * Impedance[Ohm]

KZ = NORMVAL / BASE_IMPEDANCE_OHM =

= NORMVAL * BASE_CURRENT_AMP / 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.

BASE_RESISTANCE_OHM = BASE_IMPEDANCE_OHM

Resistance[internal_resistance_unit] = KR*Resistance[Ohm]

KR = KZ

Inductance

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

BASE_INDUCTANCE_HEN = BASE_IMPEDANCE_OHM / BASE_SPEED_R_S

so the internal representation becomes:

Inductance[internal_inductance_unit] = KL * Inductance[Henry]

KL = NORMVAL / BASE_INDUCTANCE_HEN =

= NORMVAL * BASE_SPEED_R_S * BASE_CURRENT_AMP / BASE_VOLTAGE_VOLT

Flux

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

BASE_FLUX_WEB = BASE_VOLTAGE_VOL / BASE_SPEED_R_S

Then we can express the flux as:

Flux[internal_flux_unit] = KF * Flux[volt * sec / rad]

KF = NORMVAL / 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”):

Impedance[int] = (Inductance[int] * Omega[in2]) / NORMVAL

Flux[int] = (Inductance[int] * Current[int]) / NORMVAL

Voltage[int] = (Impedance[int] * Current[int]) / NORMVAL

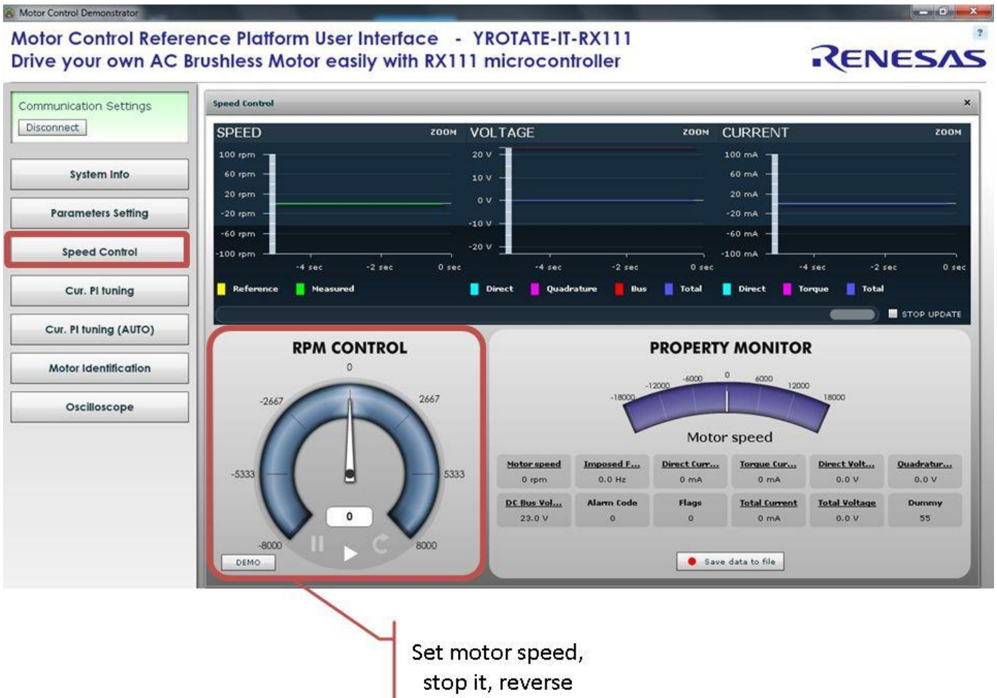
Voltage[int] = (Flux[int] * Omega[in2]) / NORMVAL

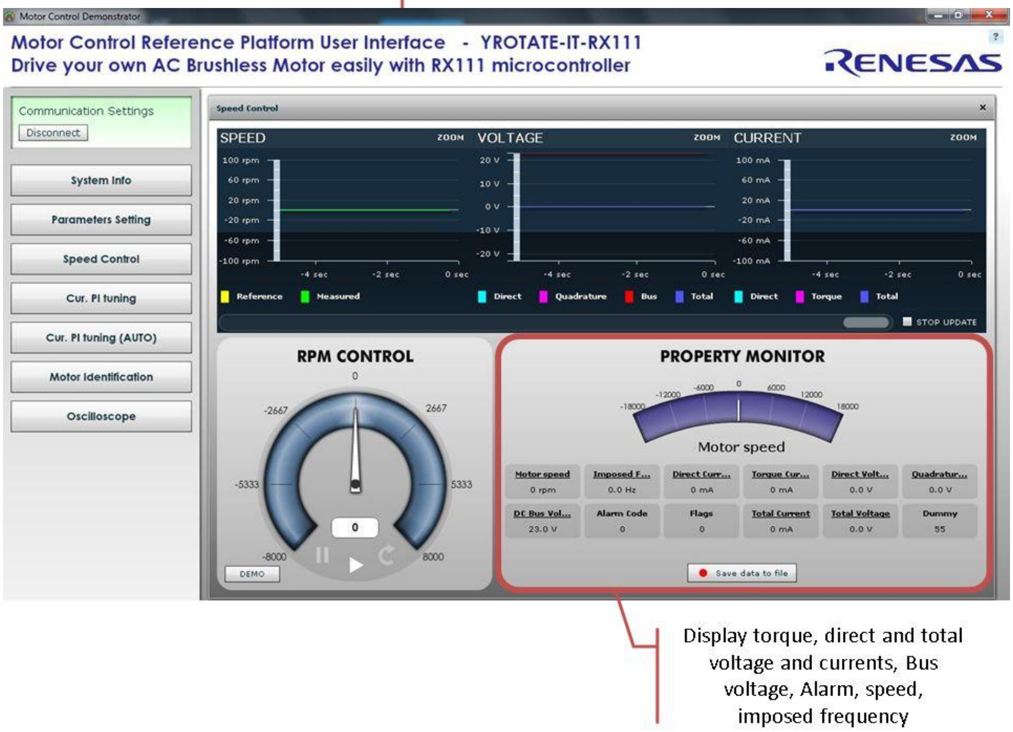
As you can notice, the calculations becomes particularly simple (x/NORMVAL is x>>14).

19. PC Graphical User Interface in details

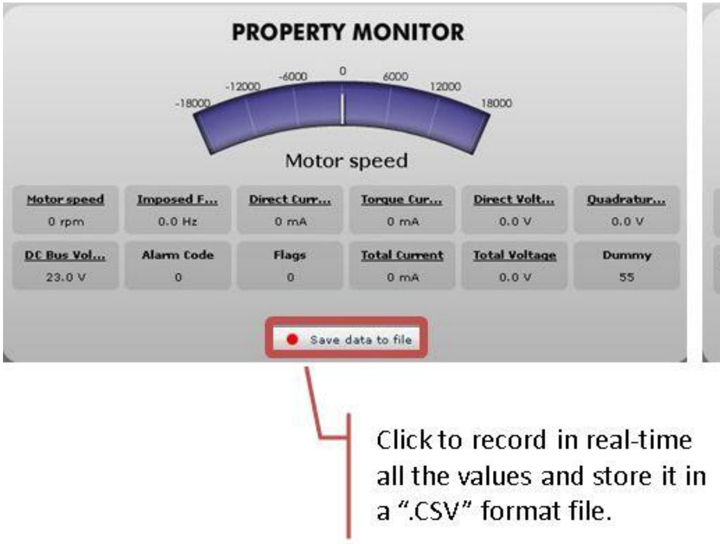
Please install the Motor Control PC GUI on your machine by following the instructions of the Quick Start Guide delivered in the YROTATE-IT-RX111 kit. After connecting the Nanotec Motor (DB42S03, 24V, 4000RPM), please connect the board RX111 and select the COM port or use the Auto-detection mechanism.

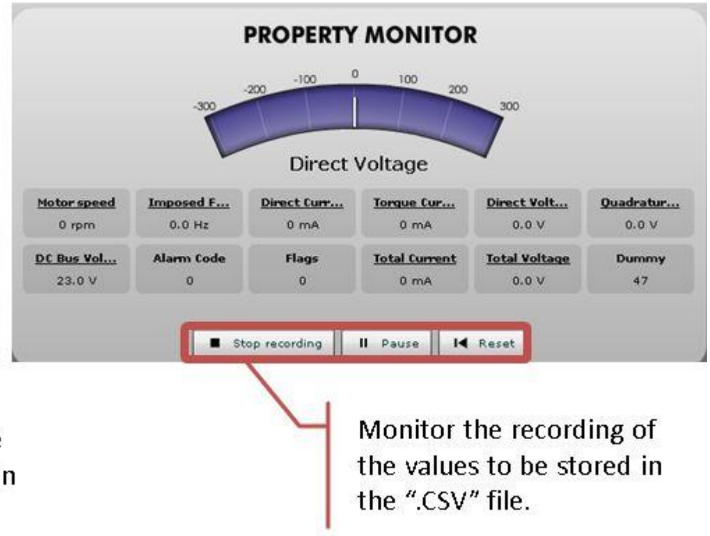
Please find below the detailed description of the PC GUI tabs and windows.



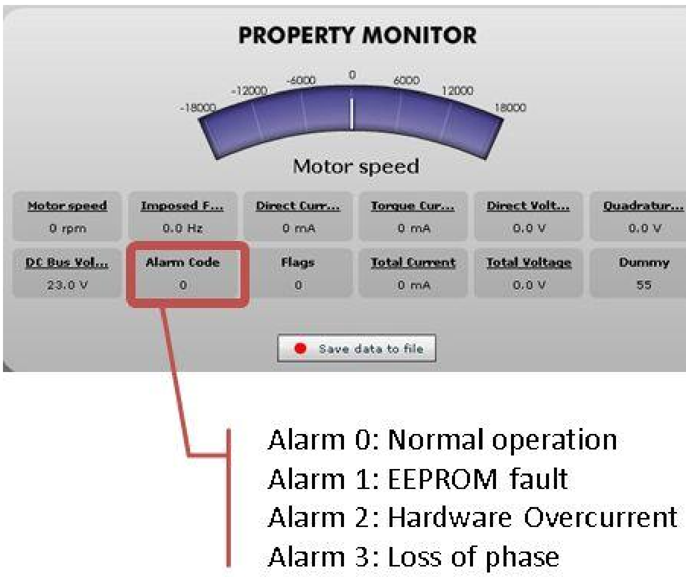


By clicking on the button “Save data to file, it becomes possible to record regularly all the values display in real-time in a file, as describe below:





Furthermore, the Speed control window displays the Alarm codes status of the board itself:



Alarm code 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.

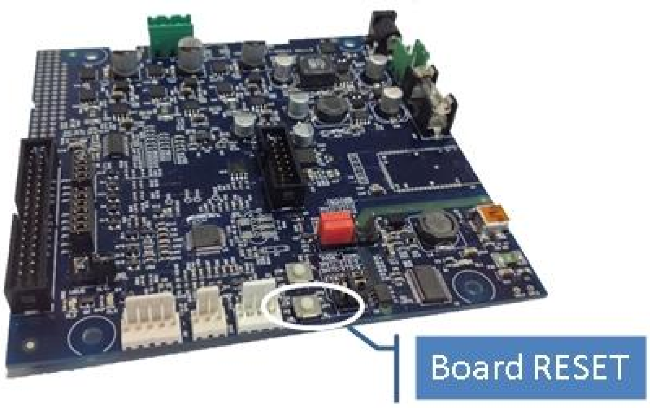
Alarm code 2:
The alarm 2 is called “hardware overcurrent” and described in the software by “FAULT_ALL”. This alarm is produced by the MCU peripheral called Port Output Enable (POE) in case of external overcurrent signal. The hardware overcurrent is producing a falling edge input on the POE pin. Furthermore, 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 behaviour 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 code 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.

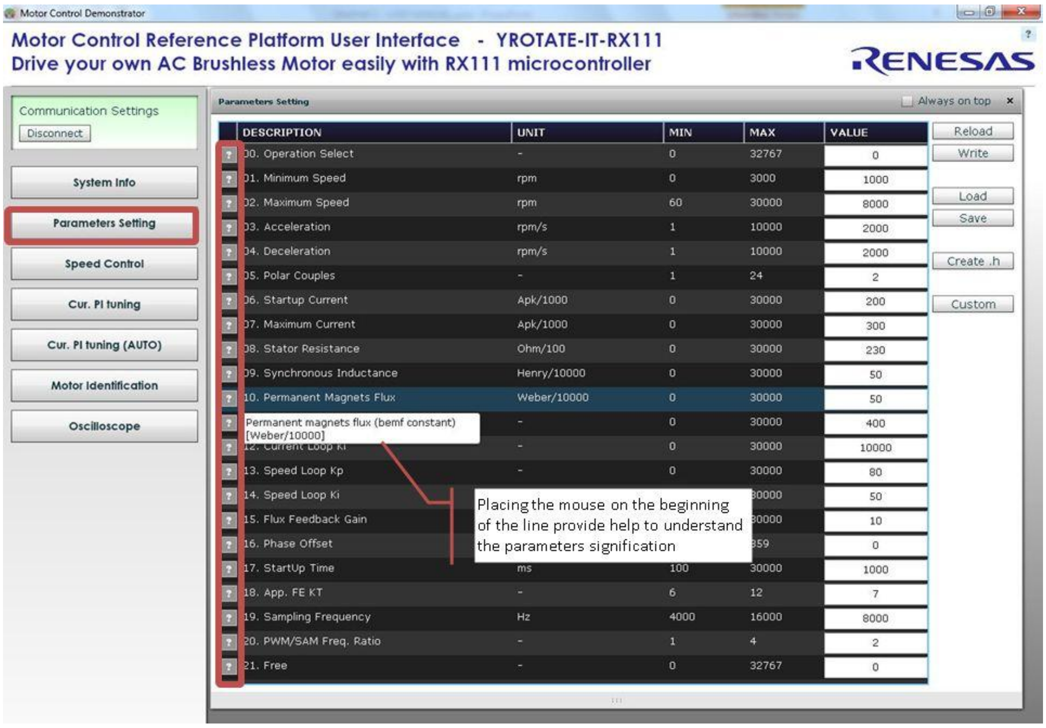
Please find below an extract of the header file “const_def.h”:

```
#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
```

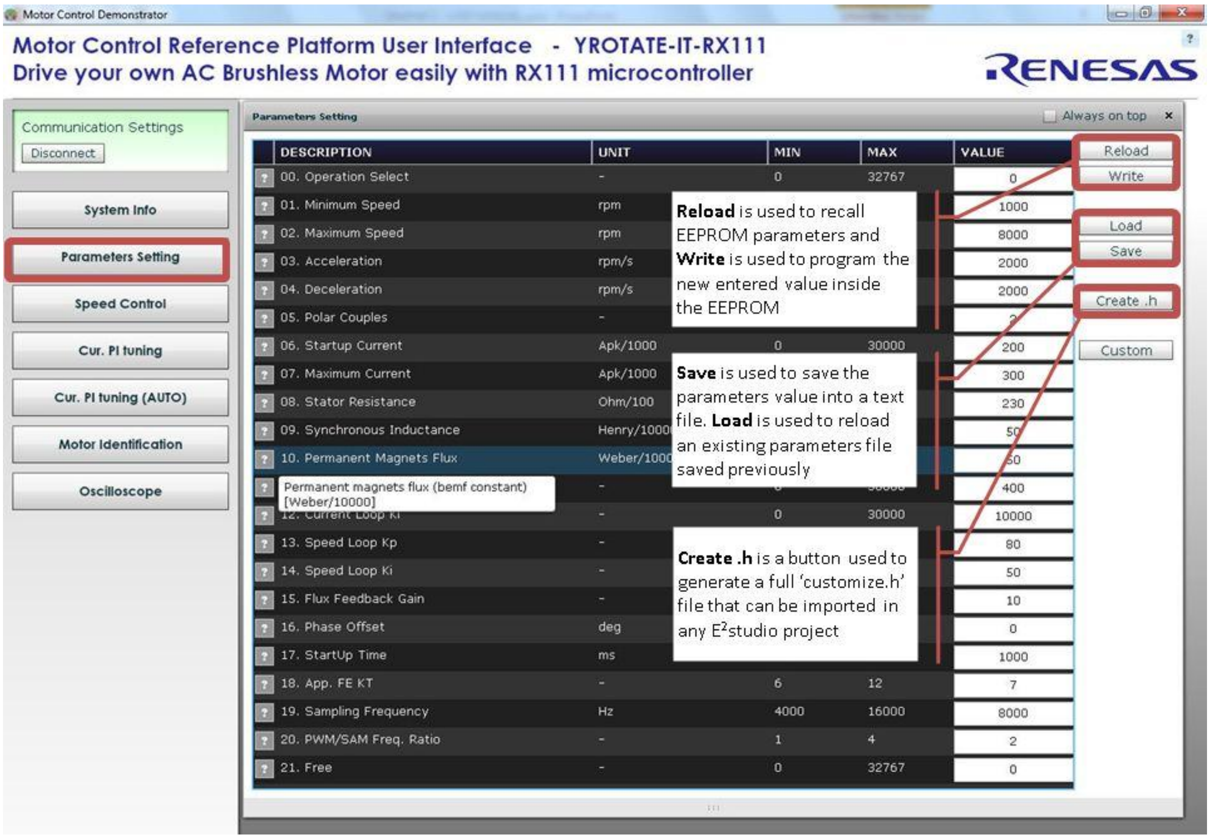
In order to reset the board in case of Alarm code 2 “Overcurrent”, please push the button as described below:



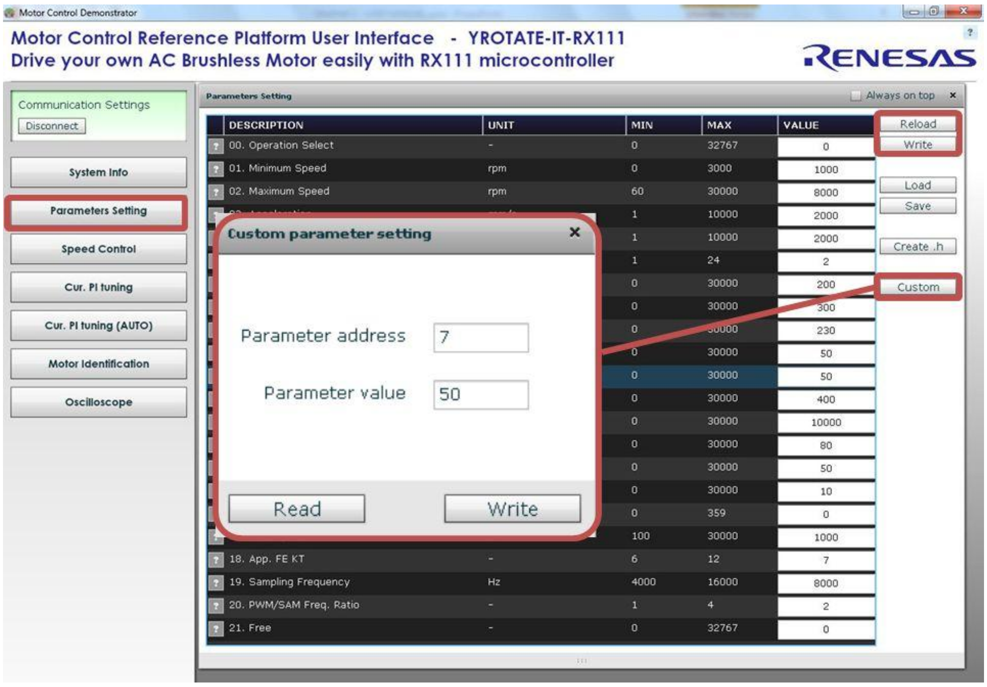
Then by clicking on the "Parameters Setting" button, the important window can be displayed showing all the parameters of the system that can be changed in real-time without having to recompile the embedded software.



The detailed description of each parameter is display when pointing the mouse on the question mark. Each parameters unit is displayed. To change one value in real-time, simply enter the new value and click on "Write" to program the new value into the EEPROM.



Furthermore, it is possible to change only one parameter at a time during fine tuning, when the motor is rotating by using the “Custom” button and enter the number of the parameter and the value inside the window as shown below.



All the parameters can be changed on the fly and after pushing the “Write” button, it’s automatically set.

But there are two exceptions related to the parameters #19 and #20:

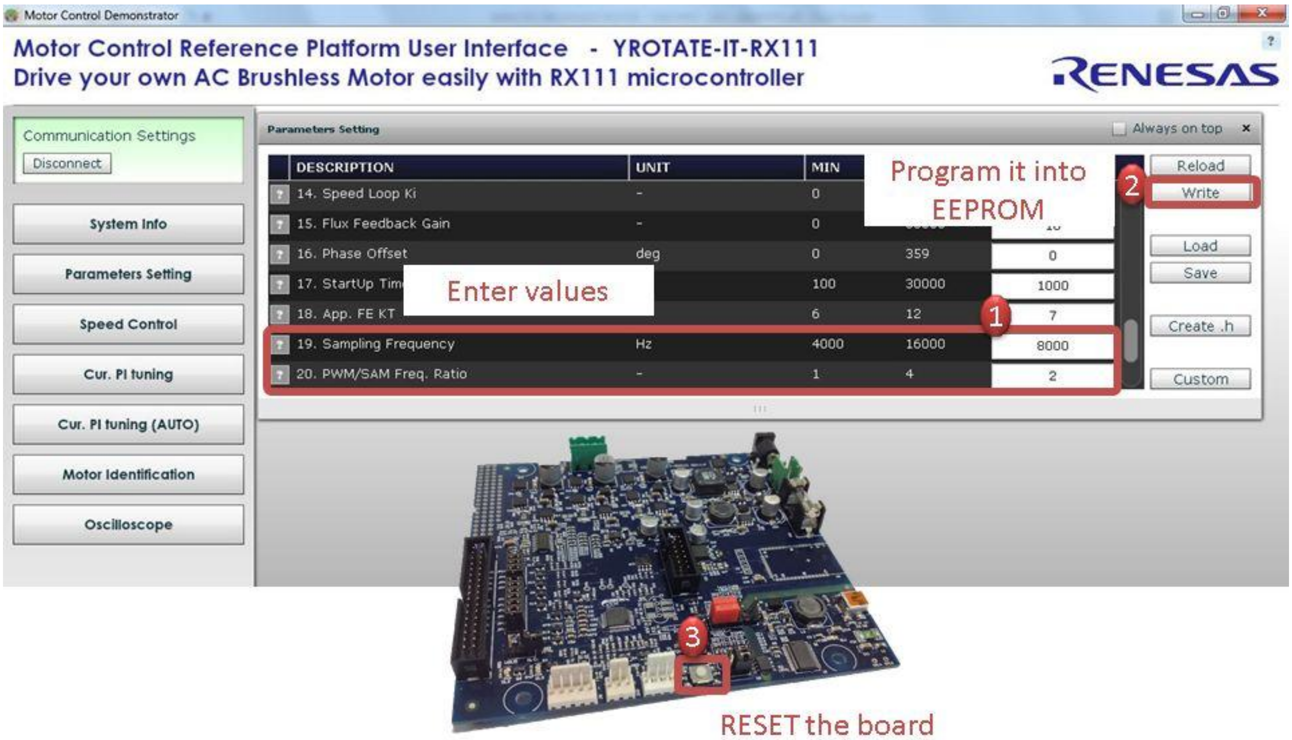
19	SAM_FRE_DEF	Set the sampling frequency [Hz] of the control loop
20	F_RATIO_DEF	Set the ratio between the PWM frequency and sampling frequency, e.g. if 8000 is set in the parameter #19 and 2 in the parameter #20, the PWM frequency is 16KHz.

The parameter #19 is setting the control loop speed. If 8KHz is selected by entering the value “8000”, the PWM frequency can be set to four different values depending on the motor and the applications either 8KHz, 16KHz, 24KHz or 32KHz.

=> It’s basically done by entering the ratio value in the parameter #20. Please find below the possible values that can be entered.

Parameter 19: Sampling freq.	Parameter 20: Ratio = 1	Parameter 20: Ratio = 2	Parameter 20: Ratio = 3	Parameter 20: Ratio = 4
4KHz	PWM freq.: **4KHz**	PWM freq.: **8KHz**	PWM freq.: **12KHz**	PWM freq.: **16KHz**
8KHz	PWM freq.: **8KHz**	PWM freq.: **16KHz**	PWM freq.: **24KHz**	PWM freq.: **32KHz**
10KHz	PWM freq.: **10KHz**	PWM freq.: **20KHz**	PWM freq.: **30KHz**	PWM freq.: **40KHz**
12KHz	PWM freq.: **12KHz**	PWM freq.: **24KHz**	PWM freq.: **36KHz**	PWM freq.: **48KHz**
14KHz	PWM freq.: **14KHz**	PWM freq.: **28KHz**	PWM freq.: **42KHz**	PWM freq.: **56KHz**
16KHz	PWM freq.: **16KHz**	PWM freq.: **32KHz**	PWM freq.: **48KHz**	PWM freq.: **64KHz**

In order to change the values of the parameters #10 and #20, please follow the description below.



After entering the new values, in this case 8KHz of sampling frequency and 16KHz of PWM frequency, click on “write” and Push the Reset button of the board.

It’s only after the Reset of the board that the new PWM and loop frequencies will be set in the embedded software.

Important Note: After setting up the new values for the parameters #19 or #20, it’s recommended to run the Auto-calibration procedure described below. It ensure the software to use the best intrinsic values and the most adapted values of the current PI coefficients.

Speed range limitations

The YROTATE-IT-RX111 kit is driving any 3-phase Permanent Magnet Motors using a sensorless vector control algorithm. So it means that there is a **minimum** speed to reach in order to run the motor properly using the three shunts current measurement methods. In the case of the Nanotec Motor DB42S03 delivered with the kit, the minimum speed is 500RPM. Below this speed, the current flowing through the three shunts are too low to be detected.

Furthermore, when the board is supplied only via the USB cable, the maximum current provided to the board is limited by the **500mA** of the USB PC port and the voltage generating by the board which is **12V**.

It means that, the first tests using the 3-phase Brushless AC motor DB42S03 from Nanotec will work properly in a specific speed range: from **500RPM** up to **2000RPM**.

The DB42S03 brushless motor is able to reach its maximum speed of 6200RPM (without load) when the power supply is 24V and up to 1A is provided. After changing the jumpers as described above and providing 24V to the board, the Nanotec motor from the YROTATE-IT-RX111 kit reach easily 6200RPM, its maximum rated speed without load. Of course, the embedded software enabling flux weakening technics, by providing more current to the board, the motor can reach **8000RPM.**

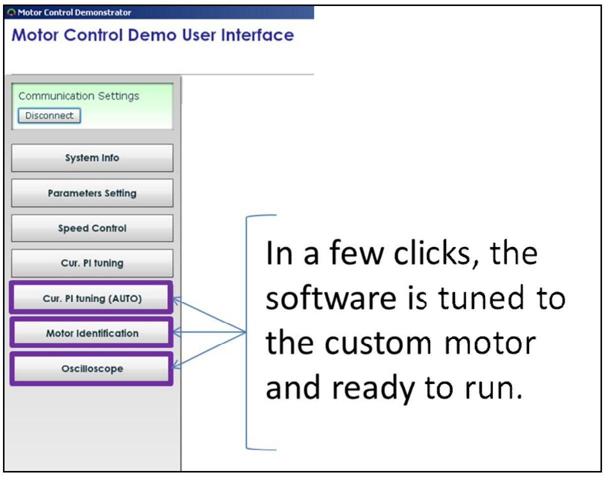
20. EEPROM parameters: detailed description

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.

Parameter number	Short name	Description
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	Set the Minimum Speed in RPM
2	RPM_MAX	Set the Maximum Speed in RPM
3	R_ACC	Set the acceleration [RPM/s]
4	R_DEC	Set the deceleration [RPM/s]
5	C_POLI	Set the number of polar couples
6	I_START	Set the 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	Set the maximum phase current (peak) [Ampere/AMP_RES]
8	R_STA	Set the stator resistance [Ohm/OHM_RES]
9	L_SYN	Set the synchronous inductance [Henry/HEN_RES]
10	PM_FLX	Set the permanent magnets flux [Weber/WEB_RES]. This value is only used when the exact integration flux estimation algorithm is selected. By default, it’s not needed as the approximated integration is selected.
11	KP_CUR	Set the Current loop Proportional coefficient: KP
12	KI_CUR	Set the Current loop Integral coefficient: KI
13	KP_VEL	Set the Speed loop Proportional coefficient: KP
14	KI_VEL	Set the Speed loop Integral coefficient: KI
15	FB_GAIN	Set the flux amplitude feedback gain. This value is only used when the exact integration flux estimation algorithm is selected. By default, it’s not needed as the approximated integration is selected
16	PHA_OFF	Set the phase offset [deg]. It is used to add a phase offset to the phase estimation, to reach better alignment
17	ST_TIM	Set the Start-up acceleration time [sec/SEC_RES]
18	FLX_FS	Filter time constant [ms]. Only needed if the approximated integration flux estimation algorithm is chosen as by default. If the exact integration method is selected, this value is not used.
19	SAM_FRE_DEF	Set the sampling frequency [Hz] of the control loop
20	F_RATIO_DEF	Set the ratio between the PWM frequency and sampling frequency, e.g. if 8000 is set in the parameter #19 and 2 in the parameter #20, the PWM frequency is 16KHz.

21. 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 **Permanent Magnet flux Λ_m**.

Please find below the auto-tuning process step by step of the Nanotec Motor DB42S03 delivered with the YROTATE-IT-RX111 kit. The DB42S03 motor is a low voltage Permanent Magnet Synchronous Motor. The auto-tuning procedure will be performed using the kit running the sensorless vector control algorithm.

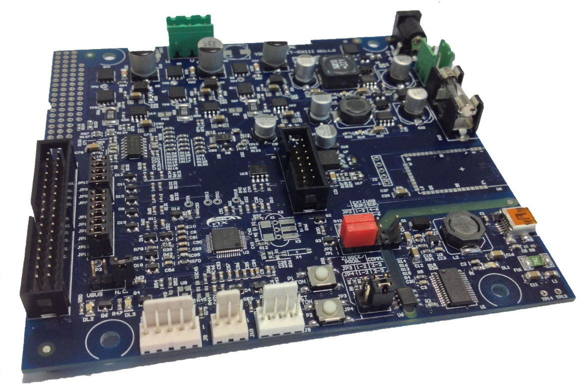
Important Note: The auto-tuning embedded software is working only on the three shunts version.

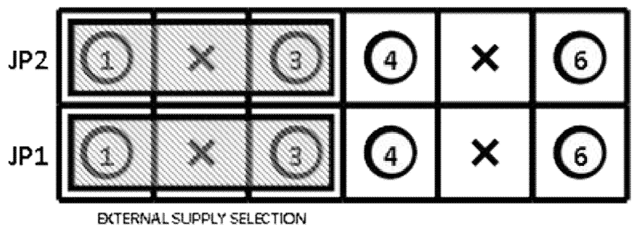
a) Please find below the specifications of the Motor delivered by the motor maker:

Motor Manufacturer: **NANOTEC** www.nanotec.com
Motor type: 3-phase AC Brushless **DB42S03**
Maximum current: **5.4A**
Bus Voltage: **24V**
Speed rated: **4000 RPM**
Number of pole pairs: **4**

SPECIFICATION \ CONNECTION	DELTA
NO. OF POL./PHASE	8/3
VOLTAGE RATED (VDC)	24
CURRENT NO LOAD/RATED/PEAK (AMP)	0.2/1.79/5.4
RESISTANCE/PHASE TO PHASE (Ohms) @25°C	1.5±15%
INDUCTANCE/PHASE TO PHASE (mH) @1KHz	2.1±20%
TORQUE RATED/PEAK (Nm) [lb-in]	0.0625/0.19 [0.553/1.68]
TORQUE/VOLTAGE CONSTANT (Nm/A)/(Vrms/KRPM)	0.035/2.78=BACK EMF ⚠1
POWER RATED (W)	26
SPEED RATED/NO LOAD (U/min)	4000/6200

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".



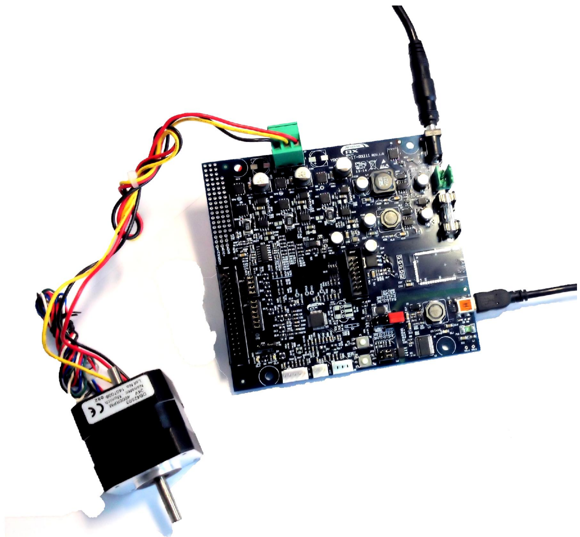


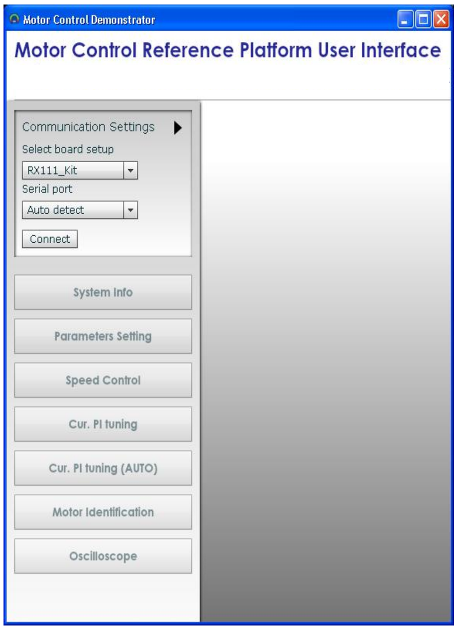
c) Let's connect the 24V_{DC} Power supply to the RX111 motor control reference kit.

The following LEDs: **DL2**, **DL4**, **DL5** are ON and the LED **DL3** is blinking to indicate that the MCU is running fine.

Important Note: The procedure below is also working with the USB power supply, but it is recommended to connect 24V_{DC} external power supply to the inverter kit. It ensure a higher resolution during the auto-tuning procedure and the extraction of the intrinsic motor parameters as the level of energy is higher.

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:





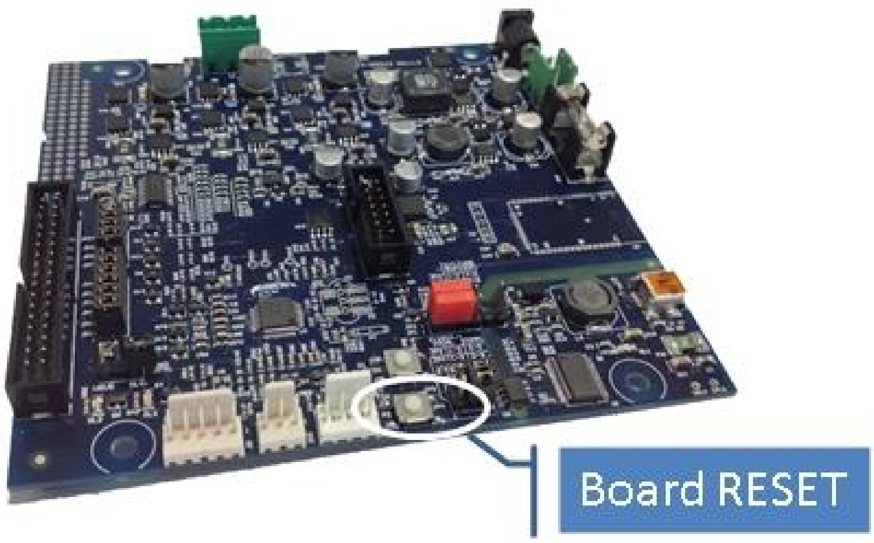
e) Launch the PC GU from the folder: "C:\Program Files\MCDEMO" launch: "MotorController.exe"
The **LED DL1** of the RX111 board is blinking rapidly showing communication between the board and the PC.

Click on the "setup" button and select "RX111 Kit" and select "Auto detect" and click on "Connect" to ensure the PC GUI is connected to the RX111 kit.

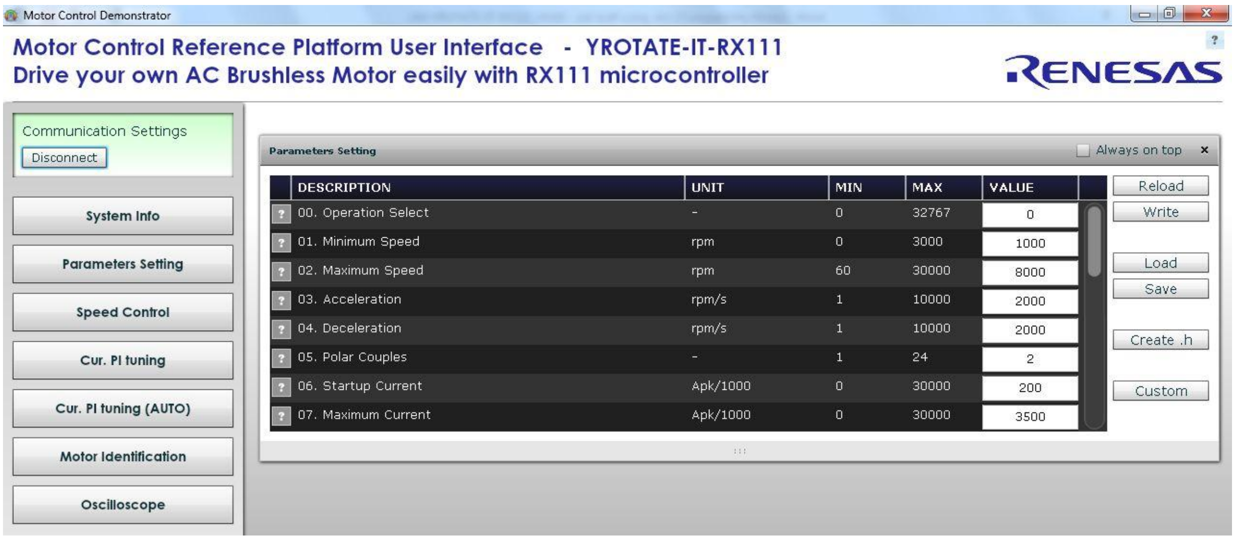
On the left hand side, the new buttons appears: “Cur. PI tuning”, “Cu. PI tuning (AUTO)”, “Motor Identification” and “Oscilloscope” which are needed for the self-calibration of the motor.

f) Clean the EEPROM content and start with the default parameters in the EEPROM.

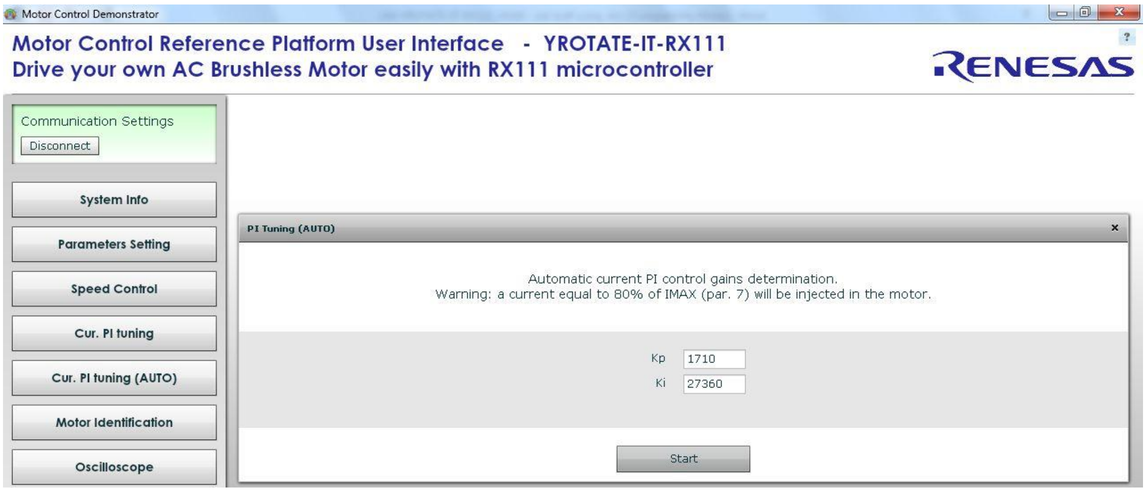
The first thing to do is to ensure that the inverter board is the default state and the default parameters are written inside. The procedure below ensures it:
Click on the “Parameters Setting” button and enter the magic value “33” in the “00. Operation Select” and push the RESET button on the board as shown:



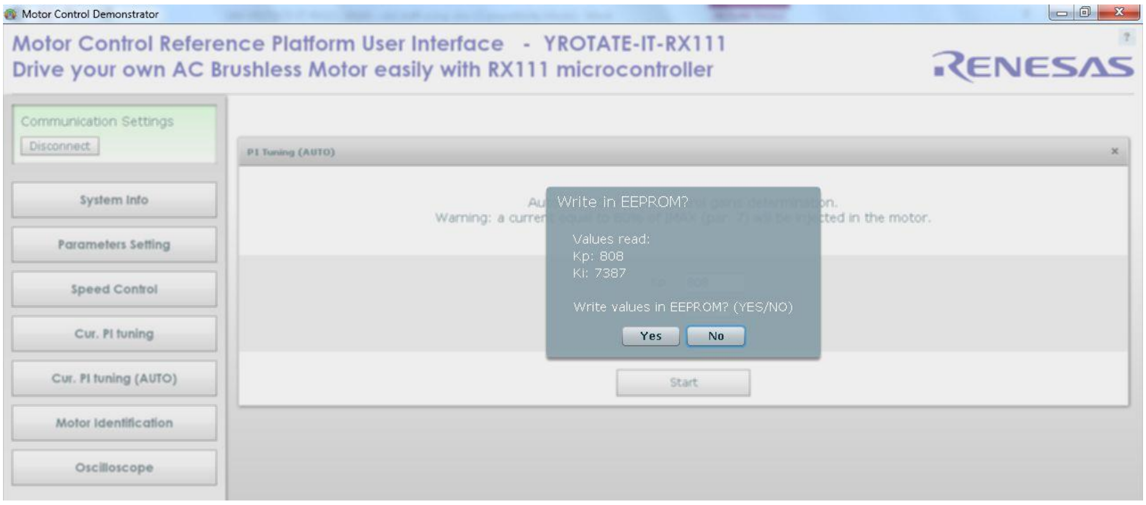
g) Set the **maximum current** (parameter n°07) as it will influence all the next steps: Click on “Parameters settings” Enter the value: **3500** (the unit is in mA) and click on “Write” to save the parameter into the EEPROM and close the parameter setting window. The maximum current parameter is fundamental for the auto-calibration. The maximum value allowed by the motor must be used to guarantee the highest resolution.



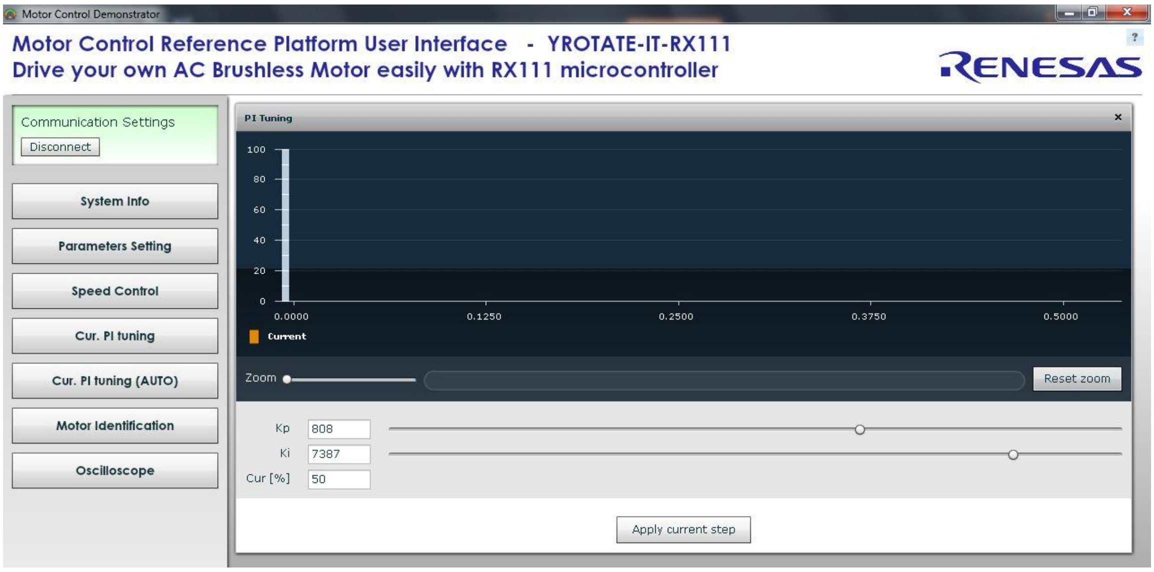
h) Click now on “Cu. PI tuning (AUTO)” button and press “start” to perform an automatic Current PI tuning. The two coefficients of the PI current block will be extracted thanks to the embedded software able to generate a step voltage and measuring the motor response.



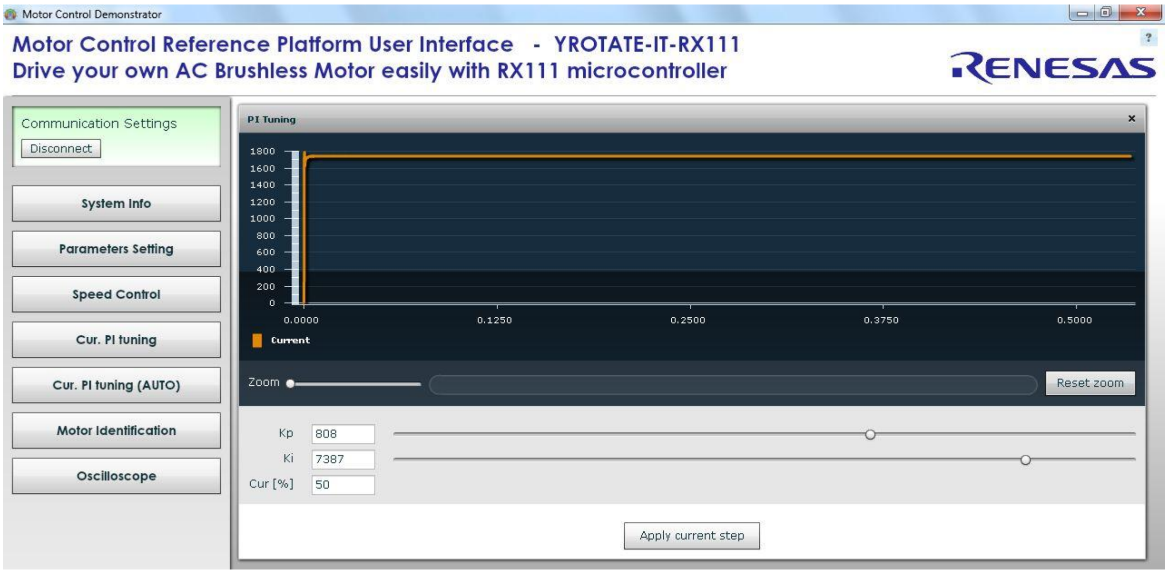
And click on "Yes" to accept the results to be programed into the EEPROM as shown below.



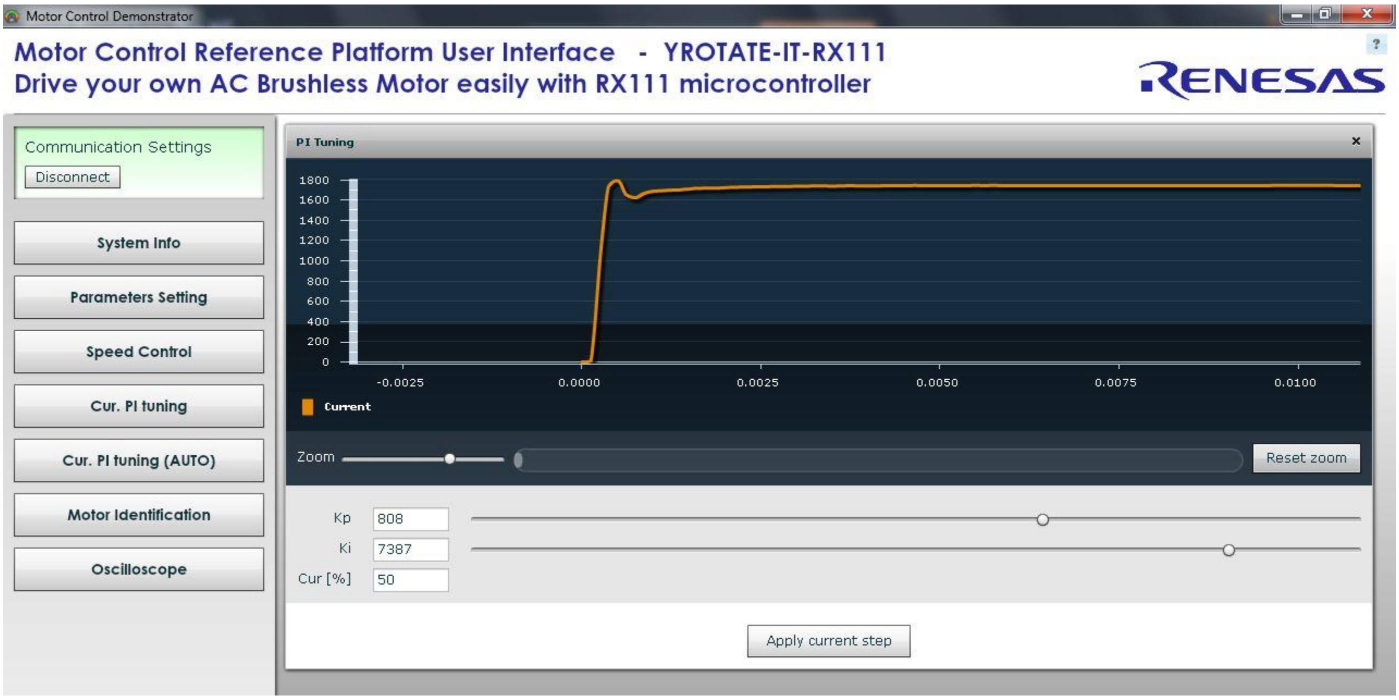
i) Now click on the button "Cu. PI tuning" to open the manual current PI tuning window and check the step answer by clicking on "Apply current step" button.



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

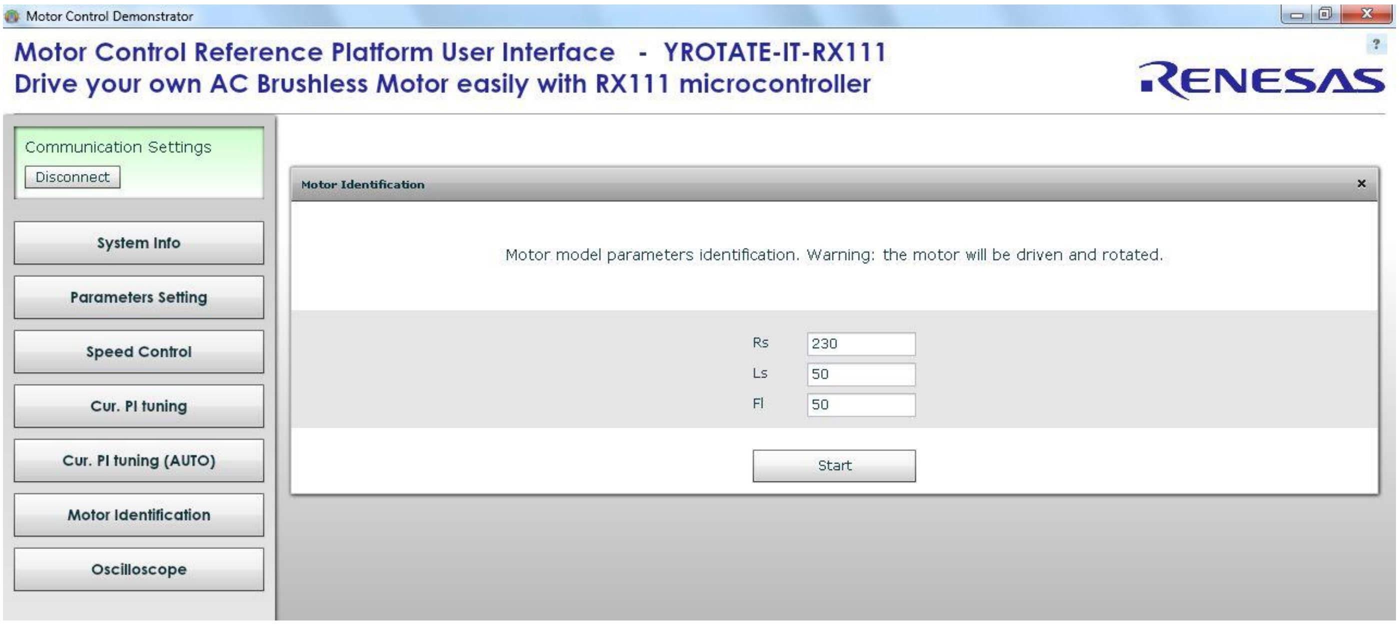


Please use the Zoom function to check the beginning of the step:



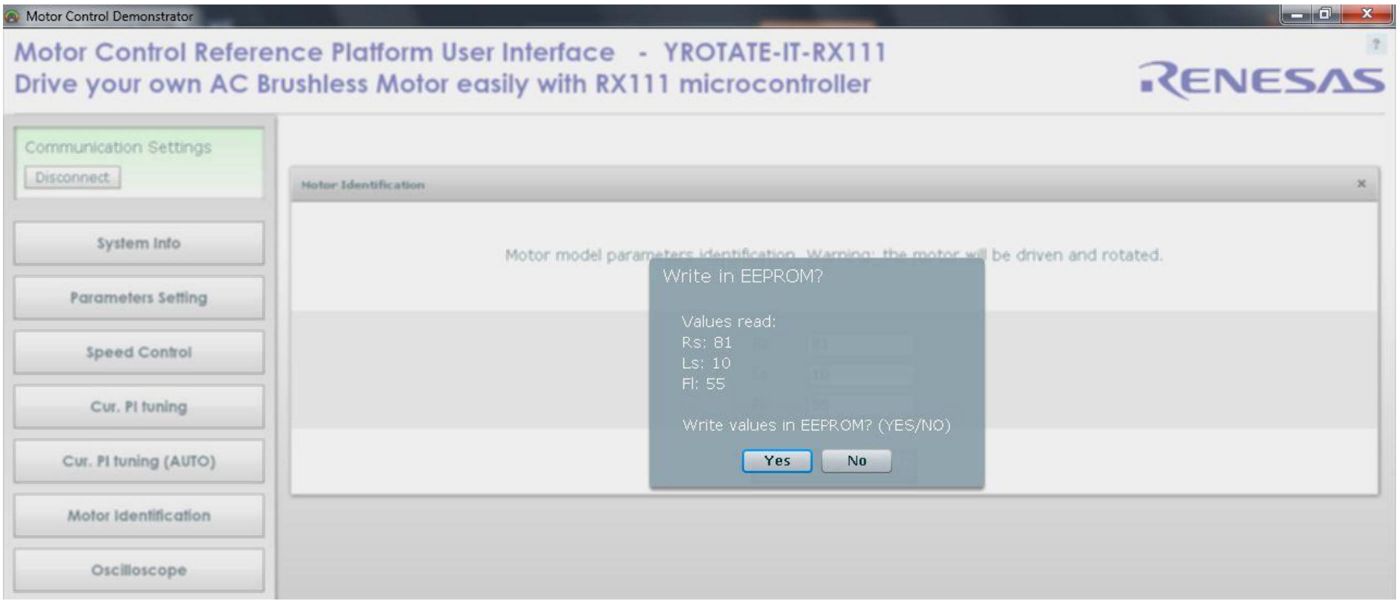
You can adjust manually the parameters to obtain an even better step response and also increase the step current level by increasing the percentage of “Cur. [%] to 90%. The default value is 50%.
Once it’s done, the window can be closed as the proportional and integral coefficients of the PI current are tuned.

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



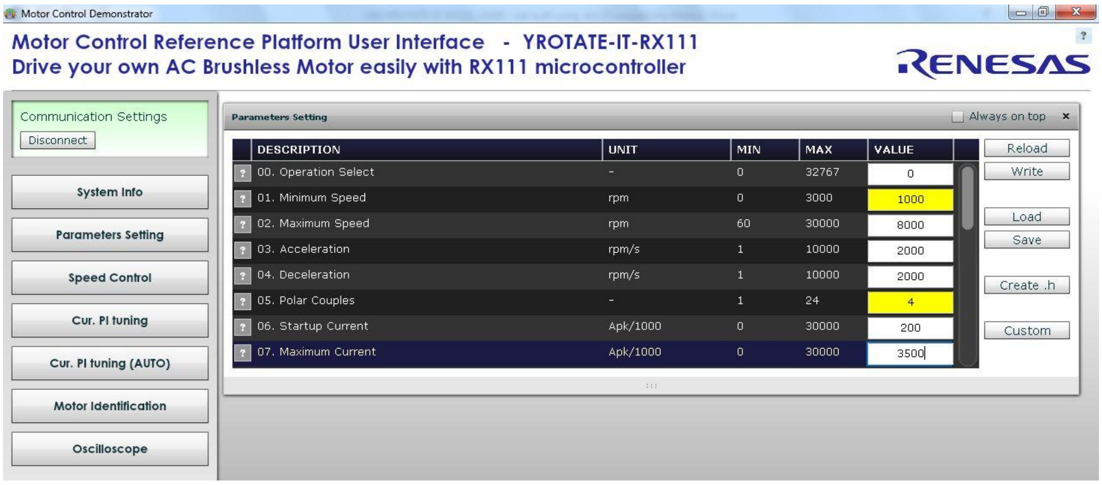
During this process the rotor should start rotating, please leave the rotor free and no loaded.

And finally accept the results to store them into the EEPROM by clicking on “yes”.

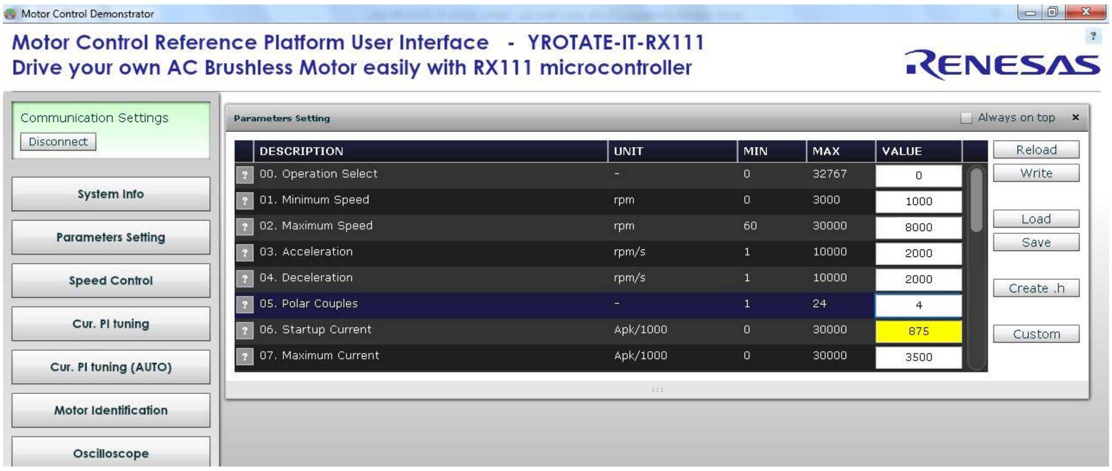


The stator resistance, the synchronous inductance and the Permanent Magnet flux have been measured and tuned.

k) Now please click on “parameters settings” and enter the number of pole pairs: **4** (parameter n°5) and enter a minimum speed or 1000 RPM



l) 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 right hand side.

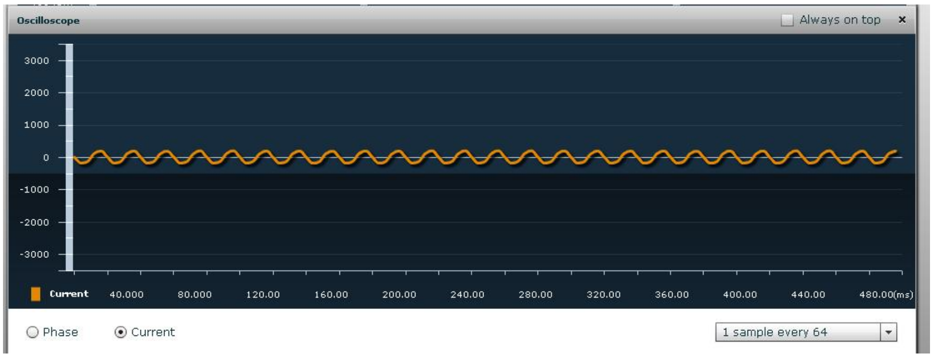


Let’s close the window.

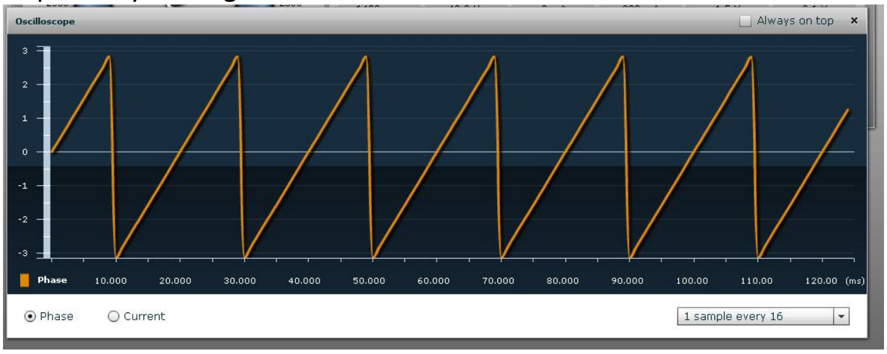
j) Now, let’s try to run the motor. Please click on the button: “Speed Control”:

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.



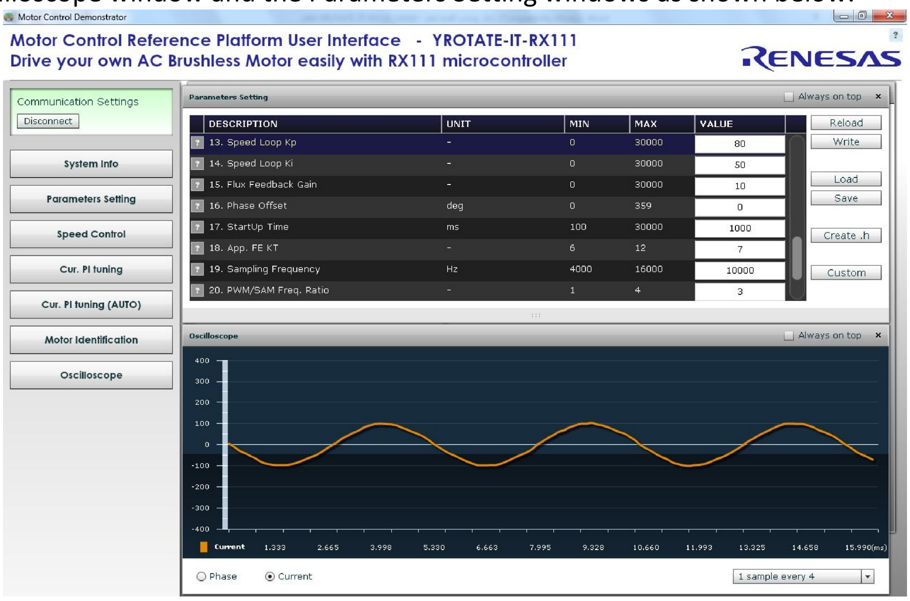
You can also display the phase by clicking on “Phase” selector:



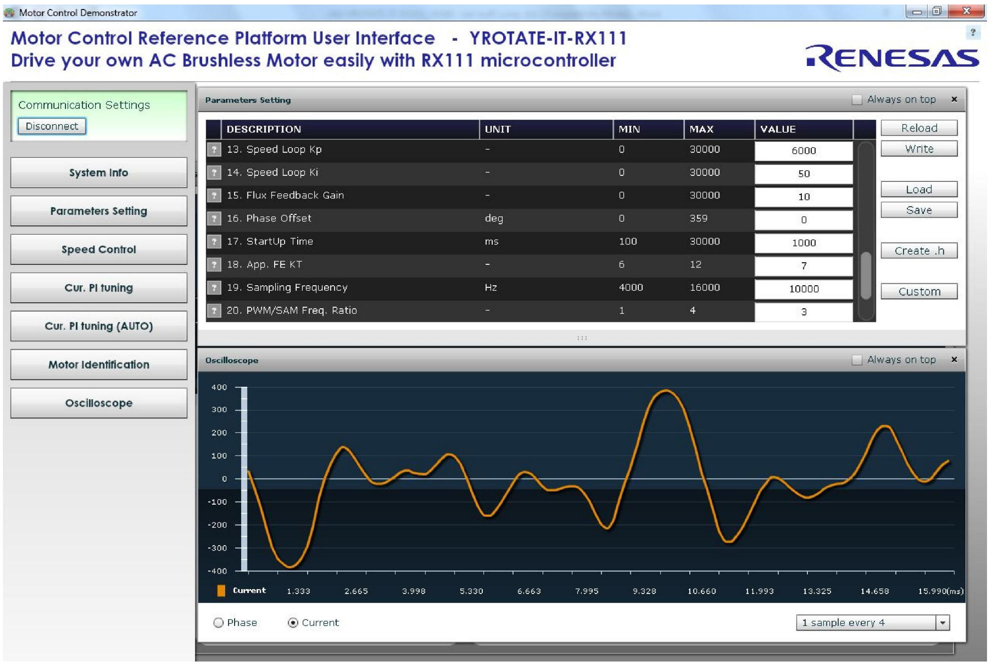
For the oscilloscope window, use an opportune time scale: “1 sample every 1” should be used for extremely fast phenomena when running at very high speed.
The setting “1 sample every 128” should be used for extremely low phenomena when running at very low speed.
Let’s start with an intermediate value and adjust it in order to see some periods of the current or the phase.

k) When the motor is running, you can adjust the two speed PI parameters: the proportional and integral terms: #13 and #14
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, the PWM frequency is set to 30KHz and the sampling frequency to 10KHz. The board was RESETTED and the auto-tuning procedure redone.

Please open the Oscilloscope window and the Parameters Setting windows as shown below.

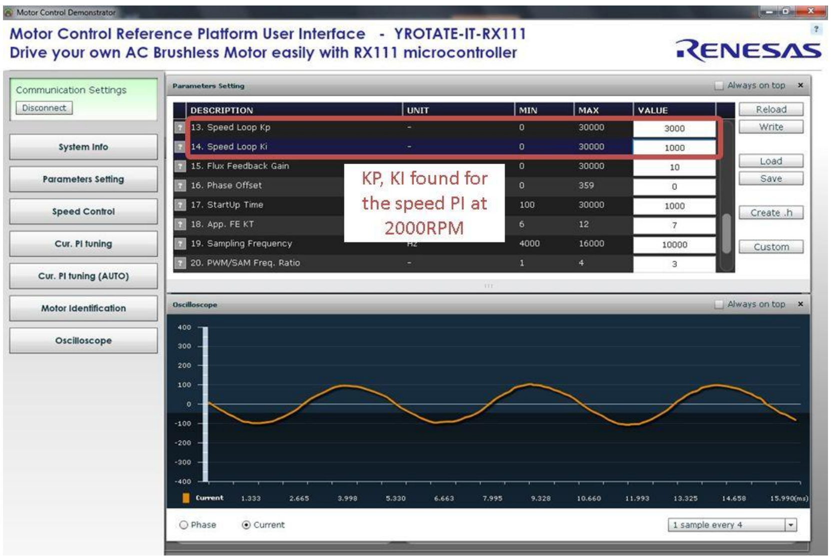


To tune the coefficients, start by increasing the Parameter n°13 (Kp) until the instability that can be display in the current or phase waveform window.
Add a step of "1000" and click "write" to see the effect and keep on increasing it.



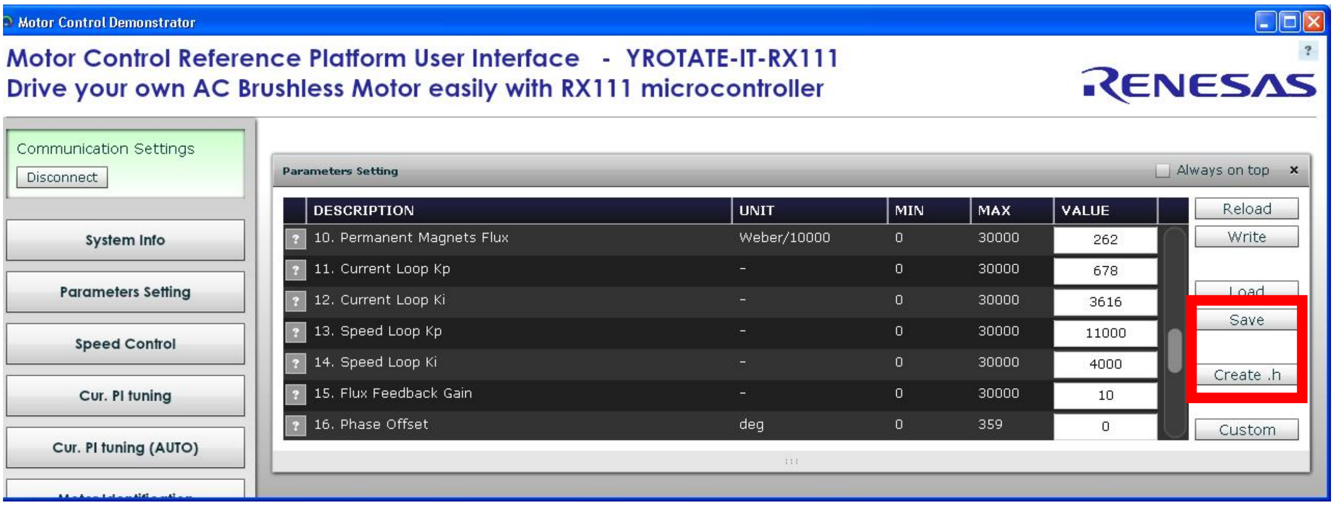
In our case, at **6000** it started to be very unstable, but the motor is still running. Set the speed to "0".
Then use half of the found value: **3000** in our case, click on "write" and set the speed to 2000 RPM.

Do the same for the **parameter n°14** (integral coefficient) which is the speed loop Ki parameter. Increase it until it becomes unstable.
In our case the critical value is reached at **2000** for Ki, so the value to be used is: **1000** (half of the value found).



l) Test the parameters found in all the speed ranges and different rotations.

m) Finally the parameters list can be saved in a file in .CSV ("Save" button) or .h file ("Create.h" button) format for further used and can also be uploaded later on:



Troubleshooting:

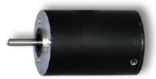
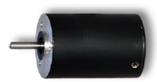
At the stage j) if the motor doesn't start or generate an alarm n°3, please set the speed to "0" to clear the alarm which indicates that the software lost the phase. One first test is to increase or decrease the start-up current and the minimum speed or the speed PI gains

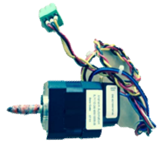
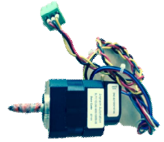
When the motor is running, you can verify the number of pole pairs taking measurement of the effective speed, and comparing it with the imposed frequency: the number of pole pairs n is: n=freq*60/speed; if you change the number of pole pairs, remember to adjust also the minimum (and maximum) speed values.
For some motors, the no-load start-up is easier if the inductance parameter is set to 0 (parameter #9)

All the procedure is tuned to manage motors which maximum current is close to the inverter capability, which is around 7A for the external power stage (shunt value is 0.05 Ohm) and 3Arms for the internal power stage (shunt value is 0.1 Ohm).
If you try to use it for very different motors, the results will be influenced by the losses in current reading resolution.

22. 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.

								
Part-name	DB42S03	DB42S03	MB057GA240	MB057GA240	FL28BL38	FL28BL38	DB87S01-S	DB87S01-S
Motor maker	NANOTEC	NANOTEC	Speeder Motion	Speeder Motion	Fulling Motor	Fulling Motor	NANOTEC	NANOTEC
Voltage	24V	24V	50V	50V	24V	24V	48V	48V
Maximum Speed in RPM	4000	4000	5000	5000	13000	13000	3000	3000
Polar Couples	4	4	2	2	2	2	4	4
Startup Current in Apk/1000	300	300	875	875	200	200	1000	1000
Maximum Current Apk/1000	3500	3500	3500	3500	400	400	3500	3500
Stator Resistance in Ohm/100	78	81	62	64	224	356	23	24
Synchronous inductance in Henry/10000	11	21	18	15	10	64	1	0
Permanent Magnets Flux in Weber/10000	55	105	250	260	51	66	184	218
Current PI - Prop. Coefficient: Kp	800	1100		748	50	50	276	486
Current PI -Integ. Coefficient: Ki	7200	3600	3461	1438	30	200	1605	1012
Speed Loop Kp	2500	2500	11000	30000	2500	2500	30000	30000
Speed Loop Ki	1000	1500	4000	15000	750	750	15000	15000
Phase Offset	0	0	0	0	0	0	0	0
Start-up Time (ms)	1000	1000	1000	1000	1000	1000	1000	1000
Filter time constant (ms)	16	16	16	16	16	16	16	16
Sampling Frequency	8000	14000	8000	14000	8000	14000	8000	14000
PWM/SAM frequency ratio	2	4	2	2	2	4	2	4

								
Part-name	**DB87S01-S**	**DB87S01-S**	**BLY171D-24V-4000**	**BLY171D-24V-4000**	**SWKX36V250W**	**SWKX36V250W**	**DB22M01**	**DB22M01**
Motor maker	**NANOTEC**	**NANOTEC**	**ANAHEIM AUTOMATION**	**ANAHEIM AUTOMATION**	**BAFANG**	**BAFANG**	**NANOTEC**	**NANOTEC**
Voltage	**48V**	**48V**	**24V**	**24V**	**36V**	**36V**	**24V**	**24V**
Maximum Speed in RPM	**3000**	**3000**	**4000**	**4000**	**285**	**285**	**4800**	**4800**
Polar Couples	**4**	**4**	**4**	**4**	**10**	**10**	**4**	**4**
Startup Current in Apk/1000	**1000**	**1000**	**700**	**700**	**2000**	**2000**	**500**	**500**
Maximum Current Apk/1000	**3500**	**3500**	**1800**	**1800**	**3500**	**3500**	**1100**	**1100**
Stator Resistance in Ohm/100	**23**	**24**	**85**	**90**	**14**	**14**	**1067**	**1160**
Synchronous inductance in Henry/10000	**1**	**0**	**23**	**11**	**0**	**0**	**0**	**0**
Permanent Magnets Flux in Weber/10000	**184**	**218**	**46**	**105**	**139**	**137**	**12**	**90**
Current PI - Prop. Coefficient: Kp	**276**	**486**	**500**	**1000**	**50**	**23**	**100**	**120**
Current PI -Integ. Coefficient: Ki	**1605**	**1012**	**4000**	**2500**	**20**	**328**	**500**	**170**
Speed Loop Kp	**30000**	**30000**	**2500**	**3500**	**30000**	**30000**	**1200**	**5000**
Speed Loop Ki	**15000**	**15000**	**1000**	**1500**	**25000**	**25000**	**500**	**5000**
Phase Offset	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Start-up Time (ms)	**1000**	**1000**	**1000**	**1000**	**3000**	**3000**	**3000**	**3000**
Filter time constant (ms)	**16**	**16**	**16**	**16**	**32**	**32**	**32**	**128**
Sampling Frequency	**8000**	**14000**	**8000**	**8000**	**8000**	**14000**	**8000**	**14000**
PWM/SAM frequency ratio	**2**	**4**	**2**	**2**	**2**	**4**	**2**	**4**

The specifications of each motor are located in the CD-ROM delivered with the YROTATE-IT-RX111 kit.

The CD-ROM contains the specifications of each motor in the folder. **.\Manuals\Tuned Motors Specifications**

Most updated information are available on the website: www.renesas.eu/motor

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Oct 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.

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 Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 2.2