



Universal motor control starter kit

Introduction

The UMC Starter Kit is considered to be a complete reference system for the universal motor drive. From a hardware and software point of view, the kit is ideal for the development of universal motor driving or, more generally, the development of phase angle control. With the supplied PC GUI, the control type can be chosen between open or closed loop (PI controller) and the control parameters easily modified.

The whole system consists of two boards (main and PC interface boards) and the PC GUI software which also serves to get measurements from the board such as current, firing angle, speed.

Contents

- 1 Getting started 3**
- 2 Phase control 5**
- 3 Description of parameters 6**
- 4 PC software 8**
 - 4.1 Acquisition 9
- 5 Firmware 11**
 - 5.1 Stand alone mode 11
 - 5.2 PC driven mode 12
 - 5.3 Frequency measurement 13
 - 5.4 Open and closed loop control 15
- 6 Hardware description 17**
 - 6.1 Testing points 20
 - 6.2 Current sense and overcurrent detection 25
 - 6.3 Using BTA12 25
 - 6.4 Working at 60Hz 25
- 7 Temperature limitation 26**
- 8 Package contents 27**
- 9 Revision history 28**

1 Getting started

Warning: The Kit is not electrically isolated from the AC input. The MCU is directly linked to the mains voltage. No insulation is ensured between the accessible parts and the high voltage. The STEVAL-IHM007V1 Kit must be used with care and only by persons qualified for working with electricity at mains voltage levels. All measurement equipment must be isolated from the mains before powering the board.

When using an oscilloscope with the Kit, it has to be isolated from the AC line. This prevents a shock from occurring as a result of touching any SINGLE point in the circuit, but does NOT prevent shocks when touching TWO or MORE points in the circuit.

When programming the MCU through the In-Dart connector, make sure the board is plugged into an insulated mains plug to avoid computer damage.

For correct operation of the UMC Starter Kit, use the following procedure (please refer to [Figure 1](#) and [Figure 2](#)):

1. Connect the two boards using the appropriate connector (SCI);
2. Connect the PC interface board to the PC serial port through a serial connector;
3. Power on to the PC interface board through JP2 or J3(5V) connector (PC interface board);
4. Attach jumper J4 (HW[Stand Alone] / SW[PC-driven]) on the main board;
5. Connect the motor to the Motor connector (main board);
6. Connect the motor feedback sensor to JP4 / Hall Sensor(Hall Effect Sensor) or J5 / Tacho(Tachometer) connectors (main board);
7. Put a jumper on SW2/Hall Tacho depending on the speed sensor used, according to the silkscreen (main board);
8. Connect the mains power to Mains connector (main board);
9. Open UMC Application window. From the **Options software** button, set the serial port and the basic / advanced user mode (see Software help for further details);
10. Click **Connect software** button. Now the kit is ready to be driven by the GUI.

In stand alone use, skip steps 1, 2, 3, 4 (no jumper on J4), 9 and 10, regulate the speed through the potentiometer POT1 (speed increases in a clockwise direction) before connecting (switching on) the main power. The motor will start running according to the default parameters and to the potentiometer value. Push SW3 ON/OFF switch to turn off (on).

The following figures show the block diagrams of the Main board and the PC Interface.

Figure 1. Main board layout

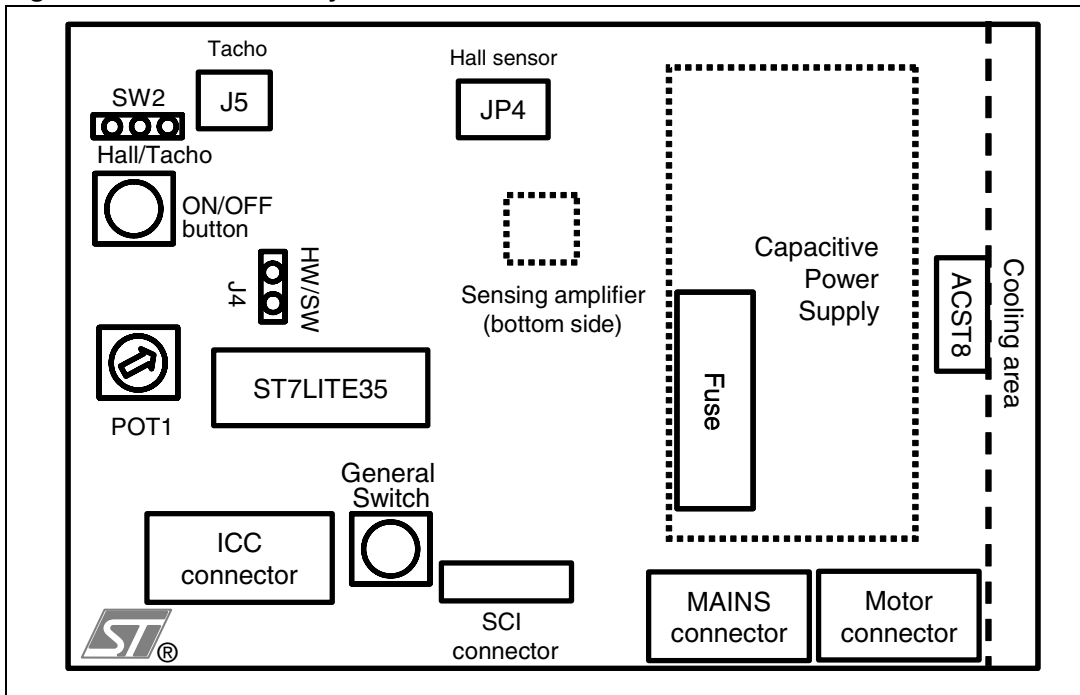
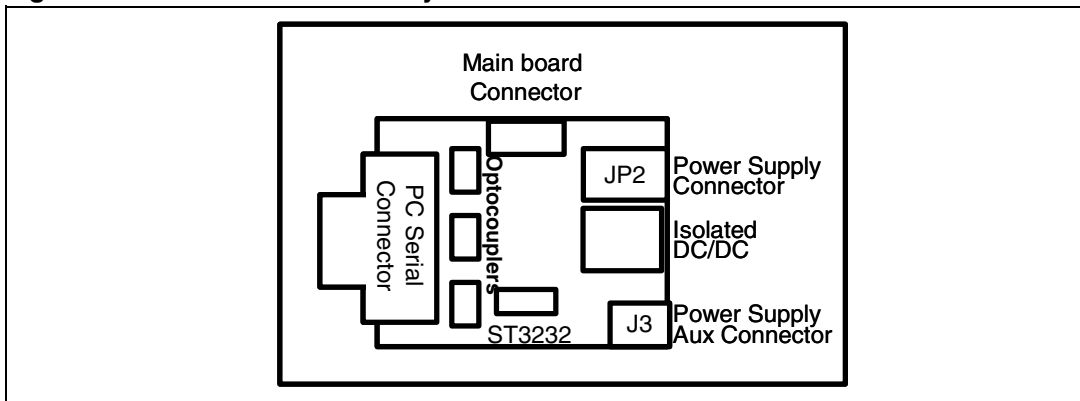


Figure 2. PC interface board layout

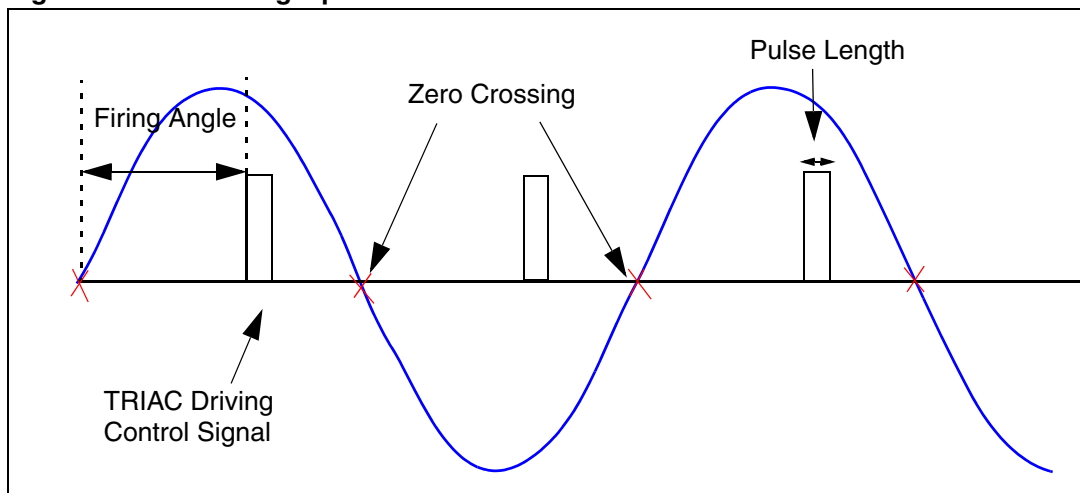


2 Phase control

The speed control of universal motor in AC mode is performed by varying the firing angle delay of a triac, thus varying the RMS voltage seen by the motor.

In detail, the triac is switched on after the mains voltage zero crossing with a delay between 0° and 180° , that is between 0 and 10 ms if working at 50Hz, or 0 and 8.3ms if working at 60Hz. The current is thus supplied from the mains to the load for a part of each half mains period: the current starts flowing through the load as soon as the triac is turned on and is stopped at the next triac current zero crossing. The power fed to the load varies depending on the firing angle delay: the lower the angle, the higher the power. The kind of control performed is symmetrical, i.e. the firing angle is kept constant within a period of the mains voltage.

Figure 3. Phase angle partialization



3 Description of parameters

In PC-driven mode all the parameters can be changed in real time (see: [Section 4 on page 8](#)). However, when the system works in stand alone mode, parameters such as maximum and minimum firing angles, pulse length, maximum current allowed, soft start duration, cycle change duration, open loop (OL) or closed loop (CL) control, target angle (if in OL), target speed (if in CL), minimum and maximum speed (CL) are set in the *UMC.h* file.

These default values can be modified in the *UMC.h* file, after which the project should be rebuilt and the microcontroller reprogrammed. Otherwise, the only parameters that can be changed while the program is running are the target angle (OL) or the target speed (CL), and this can be obtained by rotating the potentiometer POT1 on the main board (see [Figure 1](#)).

Below is a brief description of the parameters:

- **Soft Start:** this expresses in how many periods of Mains the Firing Angle reaches the Target Angle starting from Alpha Max angle.
- **Pulse Length:** duration of the firing pulse; the range is [50: 500] μ s
- **Measured Current:** selects the type of current measurement: on the mains zero crossing or the maximum value.
- **Current limitation:** if the selected current measurement type is Max current, this value sets the maximum allowed current flowing through the motor. If the load current reaches this value for a time longer than 1.5ms the motor is stopped. The user should be aware that the current is measured only during the positive wave of current if the selected measured current is the maximum value.
- **Alpha Min:** minimum firing angle allowed; range [4° : (Alpha Max-1)].
- **Alpha Max:** maximum firing angle allowed; range [(Alpha Min+1) : 167°(50Hz), 162°(60Hz)].
- **Soft Variation:** it expresses in how many periods of Mains the Firing Angle (OL) or the target speed (CL) reach a new Target Angle (OL) or a new target speed (CL) starting from the actual value.
- **Ki:** integration factor in PI speed control (CL).
- **kp:** proportional factor in PI speed control (CL).
- **Pulses Per Revolution:** it indicates the number of encoder pulses per shaft round, if this value is not correctly set, the measured speed will not be correct, thus compromising the CL control.
- **Target Speed Min/Max:** minimum/maximum target speed; the value of the changing speed knob, in PC driven mode, and, in stand alone and CL mode, the value read from the potentiometer change accordingly to these parameters.
- **Filtering:** number of acquisitions whose average gives the measurement.

Here follows an example of the *UMC.h* file with a brief description.

```

/*****
File name      : C:\temp\UMC\source\umc.h
Author        : UMC GUI Application
Description    : UMC Parameters
Creation      : Monday, November 14, 2005 12:01:35
*****/
#ifndef __C5A1372E_D06C_47ED_AEF2_67EDD76C3079__
#define __C5A1372E_D06C_47ED_AEF2_67EDD76C3079__

#define UMC_OPEN_LOOP      0x0001
#define UMC_CLOSED_LOOP   0x0002

#define UMC_MAX_CURRENT    0x0000
#define UMC_ZERO_CROSSING 0x0001

/* CONTROL SETTINGS */
#define UMC_O_C_LOOP      UMC_OPEN_LOOP      /* This has to be changed into
                                                UMC_CLOSED_LOOP if necessary*/
#define UMC_PULSES_PER_REVOLUTION 0x0008 /*Corresponding to parameter
                                                Pulses Per Revolution on the GUI*/
#define UMC_SOFT_START    0x0032 /*Corresponding to parameter
                                                Soft Start on the GUI*/
#define UMC_PULSE_LENGTH  0x00C8/*Corresponding to parameter
                                                Pulse Length on the GUI*/

#define UMC_OVER_CURRENT  0x0080 /*Corresponding to parameter
                                                Current Limitation on the GUI*/

#define UMC_ALPHA_MIN     0x000A
#define UMC_ALPHA_MAX     0x00A7
#define UMC_MEASURED_CURRENT  UMC_ZERO_CROSSING /* This has to be changed into
                                                UMC_MAX_CURRENT if necessary*/

#define UMC_SOFT_VARIATION 0x0032
#define UMC_TARGET_ANGLE  0x0000          /*Corresponding to parameter
                                                Target Angle, knob of the GUI in open loop*/
#define UMC_TARGET_FREQUENCY 0x270D      /*Corresponding to parameter
                                                Target Speed, knob of the GUI in closed loop*/

#define UMC_KI             0x0002
#define UMC_KP             0x0019
#define UMC_CL_MIN_SPEED  0x1388          /*Corresponding to parameter
                                                Min, close to the knob of the GUI in open
loop*/
#define UMC_CL_MAX_SPEED  0x2710          /*Corresponding to parameter
                                                Min, close to the knob of the GUI in open loop*/
#define UMC_CL_MIN_FREQ   (unsigned long)(UMC_CL_MIN_SPEED) *
UMC_PULSES_PER_REVOLUTION / 60
#define UMC_CL_FREQUENCY_STEP (unsigned long)(UMC_CL_MAX_SPEED - UMC_CL_MIN_SPEED) *
UMC_PULSES_PER_REVOLUTION / 60

#define UMC_FILTERING     0x0008          /*Corresponding to parameter
                                                Filterign in the Acquisition Setting of the GUI*/

#endif /* __C5A1372E_D06C_47ED_AEF2_67EDD76C3079__ */

```

For parameters not described in this paragraph, please refer to UMC GUI Help.

4 PC software

For a better tuning of parameters and for acquisition purposes, a PC interface has been developed. In *Figure 4* and *Figure 5* the Open Loop and Closed Loop windows are shown respectively.

Please launch the UMC Setup file and follow the instructions for the software installation. Once the software has been installed, select the correct COM port from the Options section and click the **connect** button to get started with the software. Then choose the Advanced or Basic Mode, where the latter case has a reduced set of user-parameters. Then you can choose both the control parameters and the control mode before starting the motor.

The communication is synchronized by means of a CTS signal (hardware handshake), which is set at a high level by firmware only when PC communication is allowed. Ensure that this signal is correctly managed by the PC otherwise communication failures may occur.

For further details please refer to the UMC Application help.

Figure 4. GUI - open loop mode

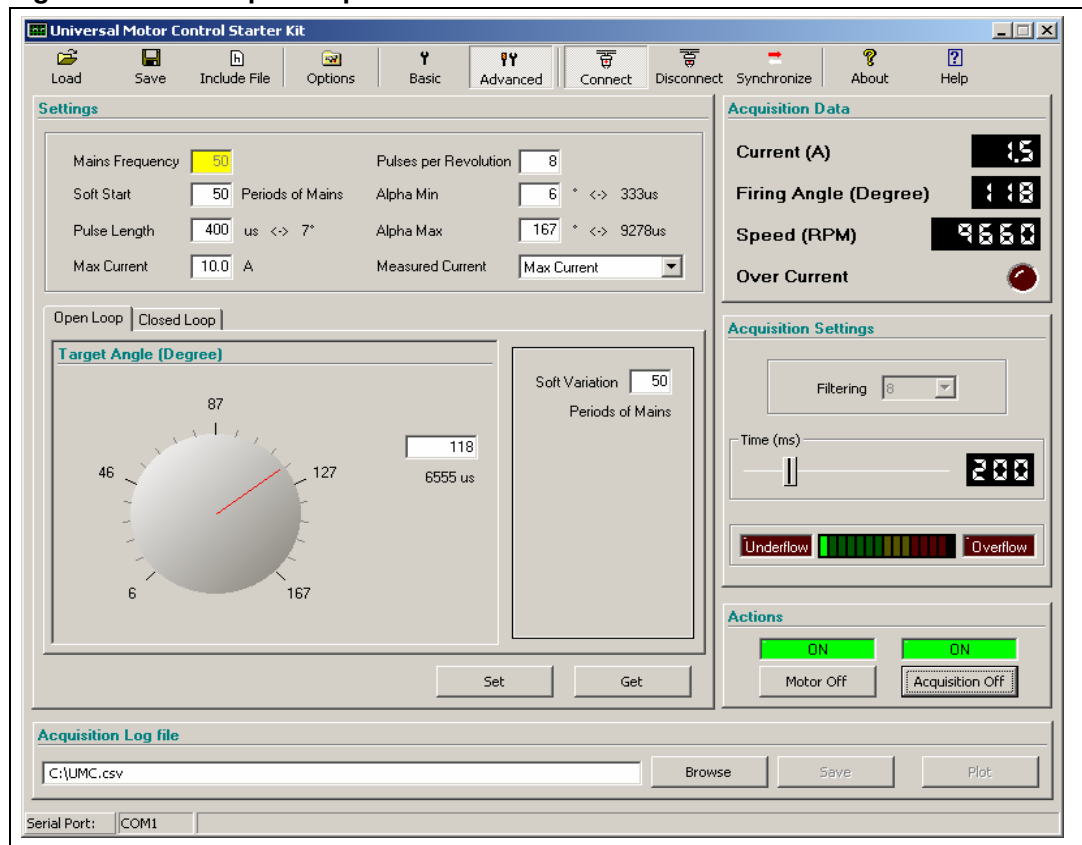
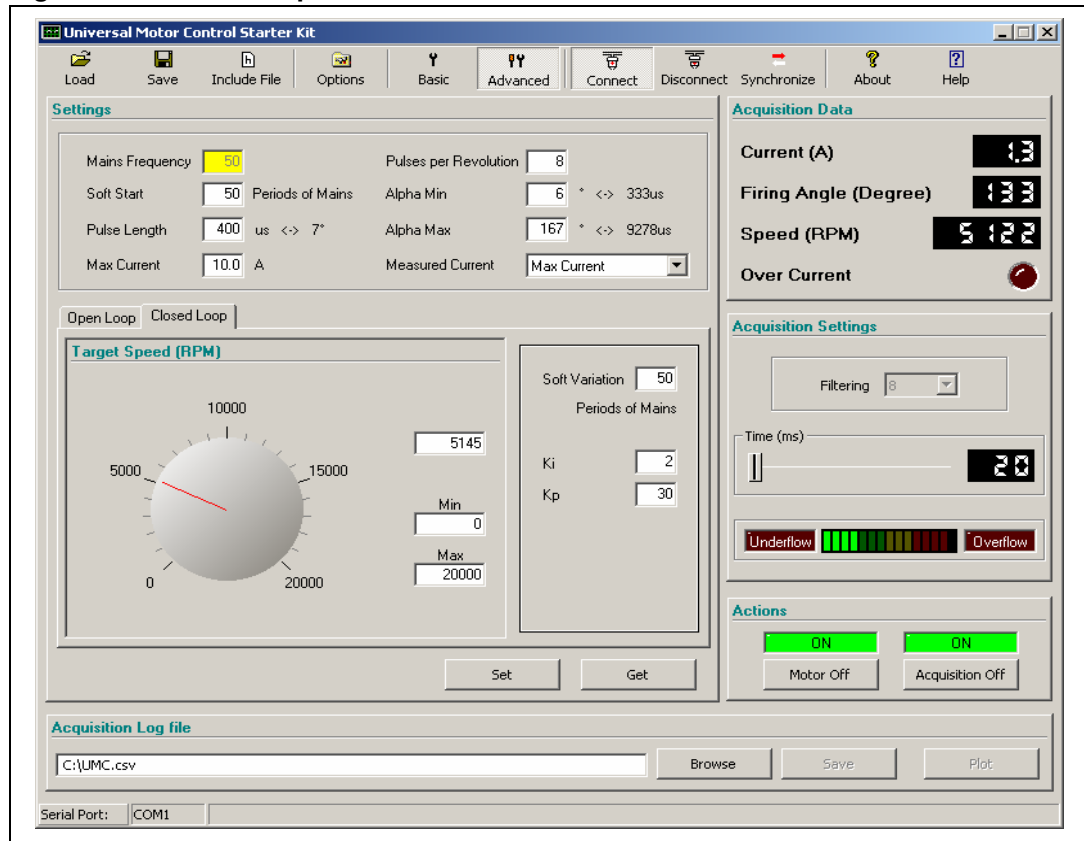


Figure 5. Closed loop mode

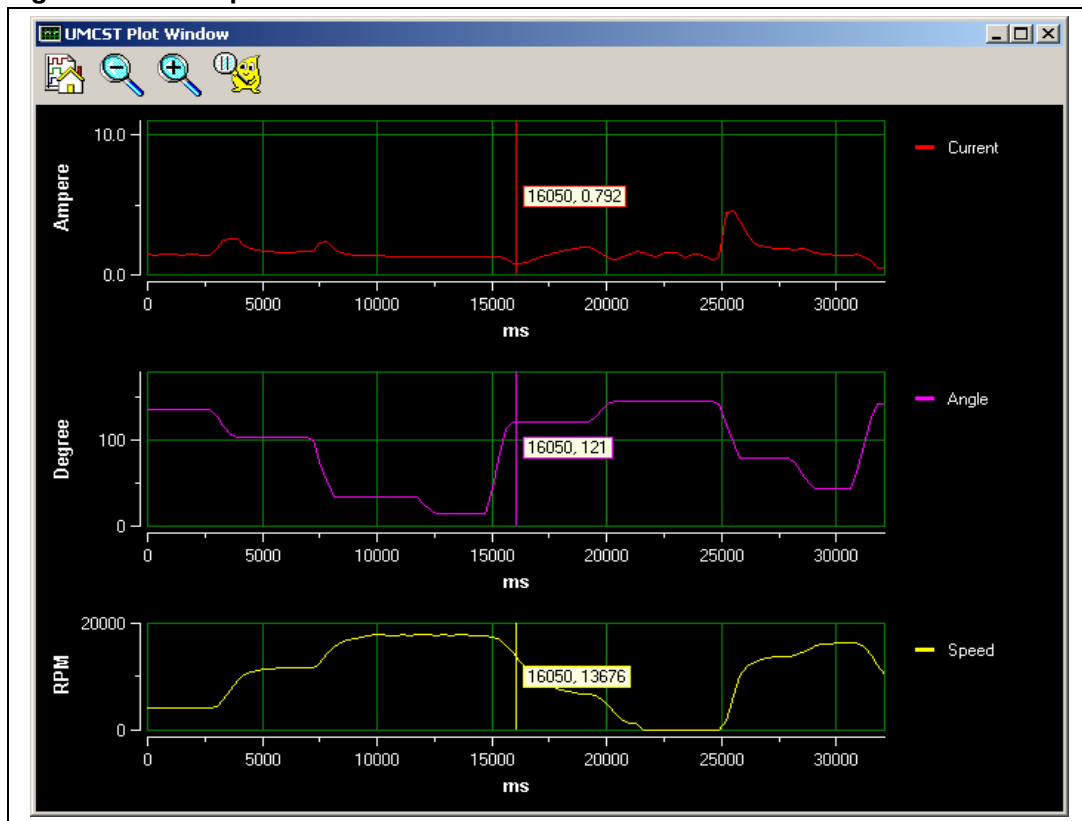


4.1 Acquisition

The acquisition takes place after having pushed the appropriate button on the right side of the GUI window. The Underflow and Overflow bars inform the user respectively whether acquisition samples are missing (communication problems), or if the application is too slow for samples to be processed. In the latter case the user should adjust the acquisition time by using the Time bar (Acquisition Settings menu) if necessary. After having stopped the acquisition, the **Plot** button is enabled. Clicking this button displays a plot based on the parameters in the Acquisition Data menu, as shown in [Figure 6](#). At the same time, a log file containing the acquired samples of the three parameters is produced.

Note: During start up and, in open loop, when the target angle is varied, the firing angle plot may result in a ramp with different slopes, the first one being steeper. Actually, the firing angle variation is linear with time and the actual slope is constant. The steeper part of the plot is in fact due to some missing acquired sampling (both in the plot and in the log file) and this is due to the fact that when the Set button is pushed, set and get commands are sent to the microcontroller preventing it from sending acquired data.

Figure 6. GUI - plot window



5 Firmware

The development toolset used is SOFTEC STVD7 ver 3.2.1. The workspace is opened by opening the *UMC.stw* file. If any modification to the source files or to the *umc.h* file has been made, the project has to be rebuilt, thus generating a new *umc.s19* file. This file is used to program the microcontroller.

The firmware has been developed in order to work both in stand alone mode (HW) and in PC driven mode (SW) depending on jumper J4. By reading the value of pin PA2 one of the two working modes is selected. Below is a brief description of the firmware flow in the two cases:

1. **Defaults Loading** (*service_routines.c*): default parameters are loaded; they can be changed by PC interface in PC driven mode.
2. **Peripheral Initialization** (*ADC_10bit.c, lite_timer_8bit.c, sci.c*):
 - ADC → ADC on, channel converted is channel 0 (hw potentiometer).
 - Lite Timer → Input Capture and Time base (1ms) interrupts enabled for tachometer or Hall effect sensor period measurement for motor speed computing.
 - SCI → 1 Start bit, 8 Data bits, 1 Stop bit, baud rate equal to 19200.
3. **Ports Initialization** (*ports.c*).
4. **Mains Frequency Detection** (*mains.c*).
5. **Reading of Pin PA2** (*mains.c*): by means of a jumper this pin is connected to ground in PC-driven mode or to a switch in stand alone mode.

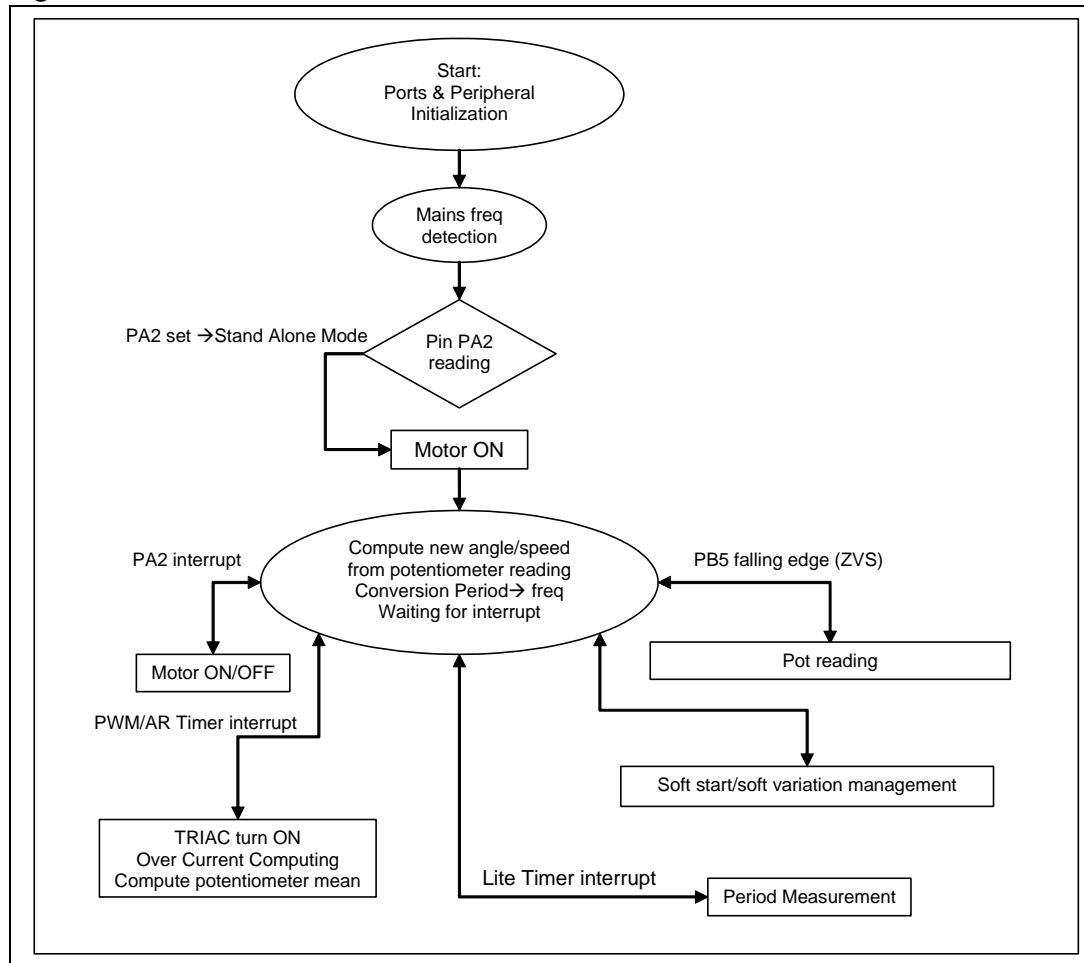
5.1 Stand alone mode

As soon as stand alone mode has been detected, the motor is turned on.

The first interrupt request is the one coming from pin PB5, the zero-crossing mains voltage detection pin, the sensitivity has been set so that both rising and falling edges of the signal generate the interrupt request. As soon as this interrupt is served, the PWM AR Timer is switched on in order to start counting the firing angle delay and then, if a falling edge occurs, the Analog to Digital Converter is set in order to convert the value coming from the potentiometer. Otherwise, on the rising edge, the A/D converter is configured for the current measuring and if required, the soft variation of the firing angle is implemented.

The triac activation is controlled by PWM AR Timer. On its interrupt routine, the measurement of the period maximum current is also performed. A flowchart for stand alone mode is shown in [Figure 7](#).

Figure 7. Firmware flowchart - stand alone mode

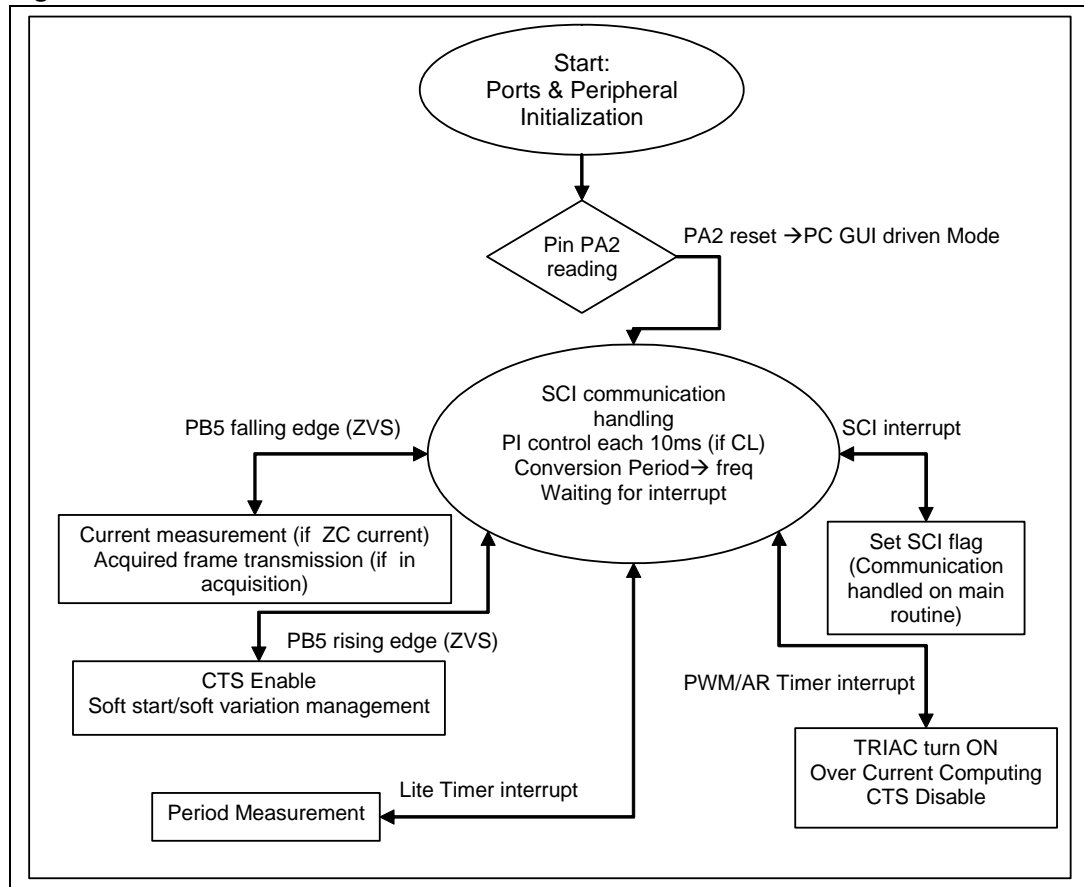


5.2 PC driven mode

The flow chart during PC driven mode is shown in [Figure 8](#).

Compared to the stand alone mode, here the SCI communication handling is present.

Figure 8. Firmware flowchart - PC driven mode



5.3 Frequency measurement

The reading of the motor speed is achieved in two steps. First, the period measurement is performed thanks to the Input Capture feature of the 8-bit Lite Timer, and then a period to frequency conversion is executed.

When a transition on the feedback signal (tachometer or Hall sensor) occurs, an interrupt is generated by the Lite Timer peripheral and the value of the Lite Timer counter is latched in the LTICR register. In order to make the system less sensitive to eventual dissymmetry on the signal duty-cycle, only high to low transitions are taken into account.

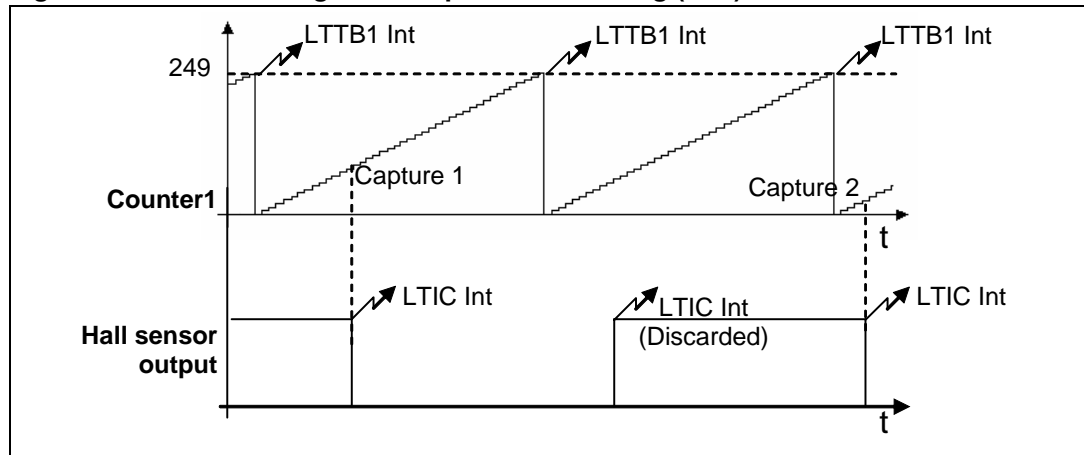
The Lite Timer Input Capture (LTIC) feature of the Lite Timer, together with the possibility of generating an interrupt when the upcounter overflows (LTTB1 Interrupt), precisely measures the period of the Hall sensor signal.

Considering, in fact, [Figure 9](#), it is possible to draw the following mathematical relationship:

$$T_H = ((250 - \text{Capture 1}) + \text{Capture 2} + 250 \cdot (N - 1)) \cdot \frac{32}{f_{osc}} \quad N > 0 \tag{5.3.1}$$

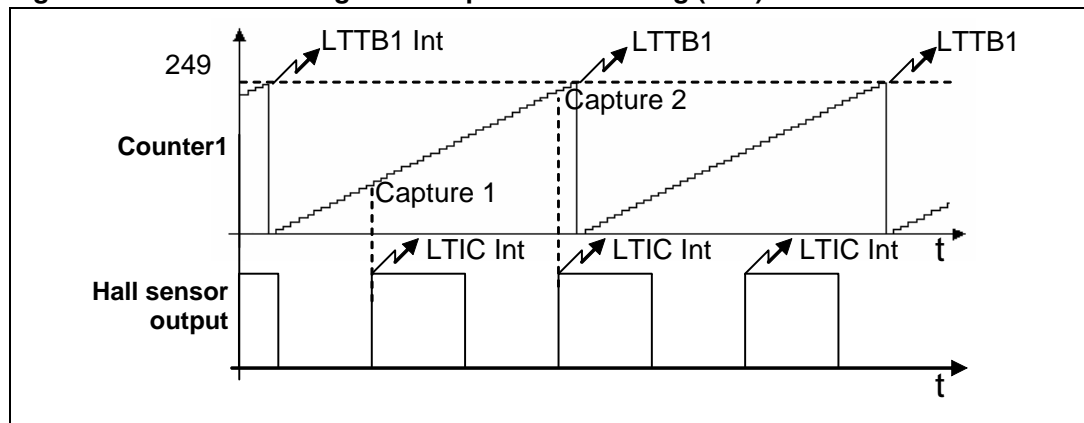
where T_H represents the feedback signal period, Capture 1 and Capture 2 indicate the values of Counter 1 at the edges of the Hall sensor signal and N is the number of LTTB1 interrupt events between the two feedback signal falling edges taken into account.

Figure 9. Feedback signal semi-period measuring (N>0)



The relation (5.3.1) is valid only under the condition N>0 and therefore it can not be used to measure frequencies higher than 500Hz. In this case the formula to be used, deduced by observing [Figure 10](#), is (5.3.2) :

Figure 10. Hall sensor signal semi-period measuring (N=0)



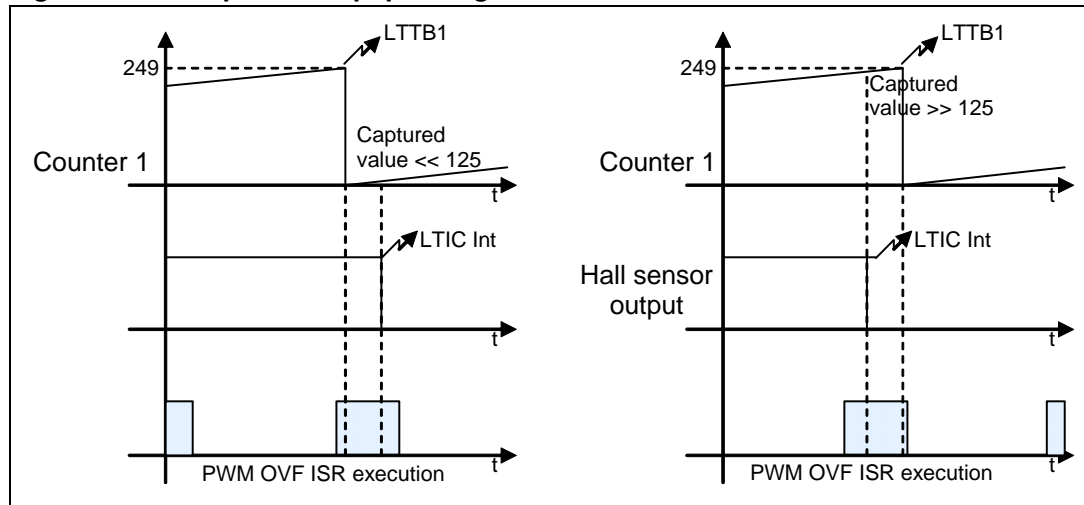
$$T_H = (Capture\ 2 - Capture\ 1) \cdot \frac{32}{f_{osc}} \quad (5.3.2)$$

Note that in both (5.3.1) and (5.3.2) T_H is computed with a resolution equal to $\frac{32}{f_{osc}}$ which is 4 μ sec at 8MHz f_{osc} .

It can be seen that in order to measure the Hall sensor semi-period correctly, the LTIC and LTTB1 ISRs must be executed by the microcontroller core in the same order in which the related interrupts events occurred. This would normally happen if no other interrupts service routines are executing. However in our software, a third interrupt source is enabled (PWM update event), so a potentially erroneous situation could arise due to the interrupt priority mechanism.

[Figure 11](#) describes two possible situations when multiple interrupts occur:

Figure 11. Multiple interrupt pending situation



Due to the higher interrupt priority of LTIC with respect to LTTB1, in both cases the LTIC ISR is executed before LTTB1 ISR.

In this case, the value stored in the LTICR is used to reconstruct the correct sequence. If the content of the LTICR is lower than 125, it is assumed that the LTTB1 event occurred *just before* an LTIC event so that the Lite Timer overflow counter must be incremented before the semi-period computation. However, if the LTICR contains a value higher than 125, it is assumed that the LTIC occurred just before a Lite timer overflow (LTTB1 event). In this case, the overflow counter must not be incremented before the semi-period computation but it will be taken into account at the next LTIC event.

After the period measurements have been carried out, they are stored in the measure array, whose depth is equal to the parameter "filtering".

Successively, in the main loop, an average of the stored measurements is performed and the period to frequency conversion is carried out. Function Compute_division() provides conversion optimization in terms of execution time for frequencies higher than 35Hz.

5.4 Open and closed loop control

When the Open Loop control strategy is selected, the firing angle is the control parameter. It can be changed by potentiometer in Stand Alone Mode or by a control in the PC GUI (refer to [Figure 4](#)) if PC-driven.

However, when the Closed Loop control strategy is selected, the control parameter is the target speed. In this case, in order to make the actual speed equal to the target speed by adjusting the firing angle, a PI controller has been implemented. The general formula for an analog PI regulator is:

$$u(t) = K_p * e(t) + K_i * \int_0^t e(\tau) d\tau \quad (5.4.1)$$

where:

$u(t)$ is the driving signal

$e(t)$ is the error signal, difference between the actual feedback frequency and the target feedback frequency.

K_p , K_I are respectively the proportional and the integral constants.

In the case of digital regulators the above equation becomes:

$$u(n) = K_p * e(n) + K_I * T * \sum_{i=0}^n e(i) \quad (5.4.2)$$

where:

$u(n)$ is the driving signal sampled at the instant nT

$e(n)$ is the error signal sampled at the instant nT

K_p , K_I , T are respectively the proportional and the integral constants

T is the sampling time.

We can also write:

$$u(n+1) = K_p * e(n+1) + K_I * T * \sum_{i=0}^{n+1} e(i) \quad (5.4.3)$$

and, subtracting equation (5.4.3) from (5.4.2):

$$u(n+1) - u(n) = K_p * [e(n+1) - e(n)] + K_I * T * \left[\sum_{i=0}^{n+1} e(i) - \sum_{i=0}^n e(i) \right]$$

simplifying:

$$\Delta u(n) = K_p * \Delta e(n) + K_I * T * e(n) \quad (5.4.4)$$

Equation (5.4.4) shows the variation of the firing angle $\Delta u(n)$ at the instant nT due to the error in the same instant (term $e(n)$) and to its variation at the instants nT and $(n-1)T$ (terms $\Delta e(n)$ and $[\Delta e(n) - \Delta e(n-1)]$).

In the "PI" routine, the error is first computed and based on it, the variation of the firing angle is calculated according to equation (5.4.4).

The coefficients must be adjusted for a given motor type. This can be easily achieved with the PC GUI.

6 Hardware description

Two separate boards have been developed: the main board and the PC interface one. In Figures 13 and 14 the schematics of the boards are shown.

At first glance, the main characteristics are:

- **A capacitive power supply** to reduce the overall cost of the system. This supply can source an average current lower than 20mA.
- **Control Unit** consisting of an ST7Lite35 device and external circuitry.
- **An ACST8-8C AC Switch.** This device is driven by means of a bipolar transistor in order to guarantee the proper working in the whole extended temperature range using only one MCU pin. In fact the turn-on current needed is equal to 35mA in nominal conditions, but it increases up to 70mA for a -20°C junction temperature. Nevertheless if no further MCU pins are needed for other purposes, the ACST switch can be directly driven using 3 HS pins connected in parallel. A noise suppressor (R7 and C7, see Figure 13) has been connected in parallel with the device in order to increase IEC61000-4-4 immunity.
- **Two speed sensor nets** for a Hall effect sensor or a Tachometer.
- **A Current Sensing circuit.**
- **A Zero Crossing Network** providing the Control Unit with a TTL signal in accordance with the zero crossing of the mains. This network is realized with a bipolar transistor, but a resistor topology could also have been used (see Figure 12), but, in this case, the firmware must handle somehow the dissymmetry compensation in order to avoid DC current injection through the motor.

Figure 12. Resistor based ZCN

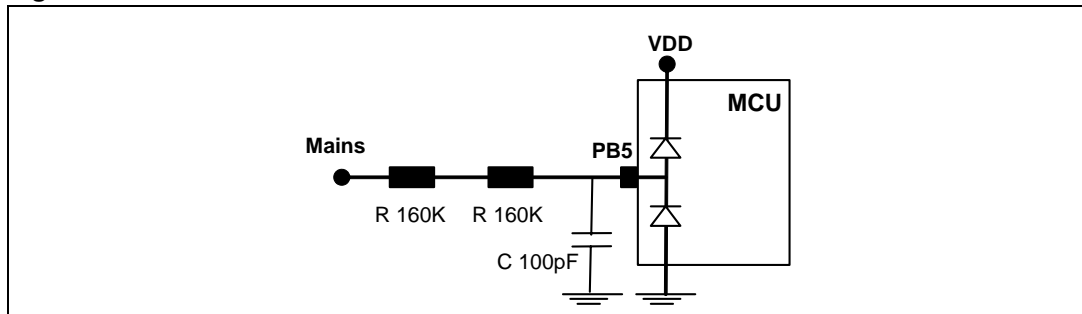


Figure 13. Main board schematic

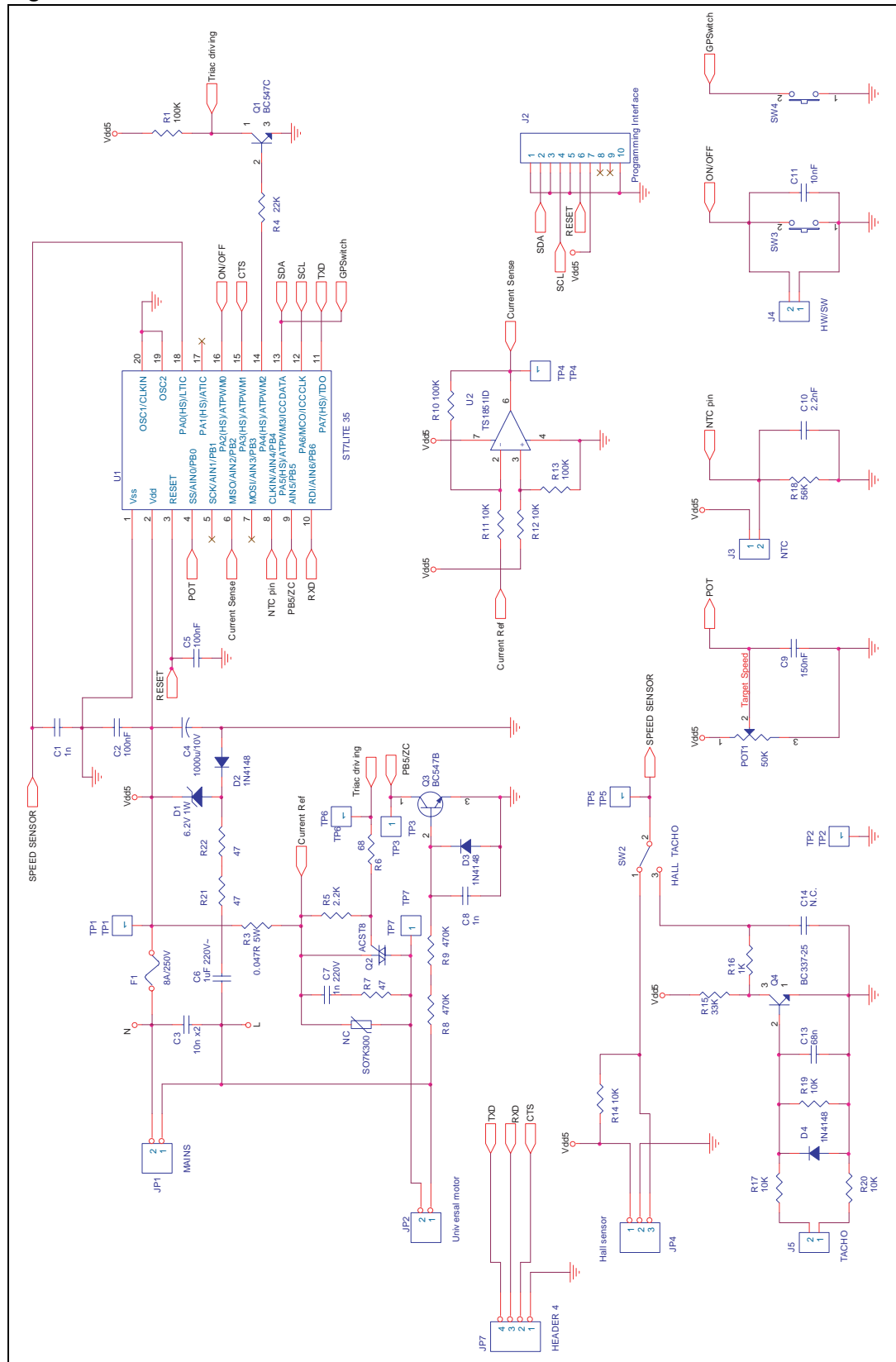
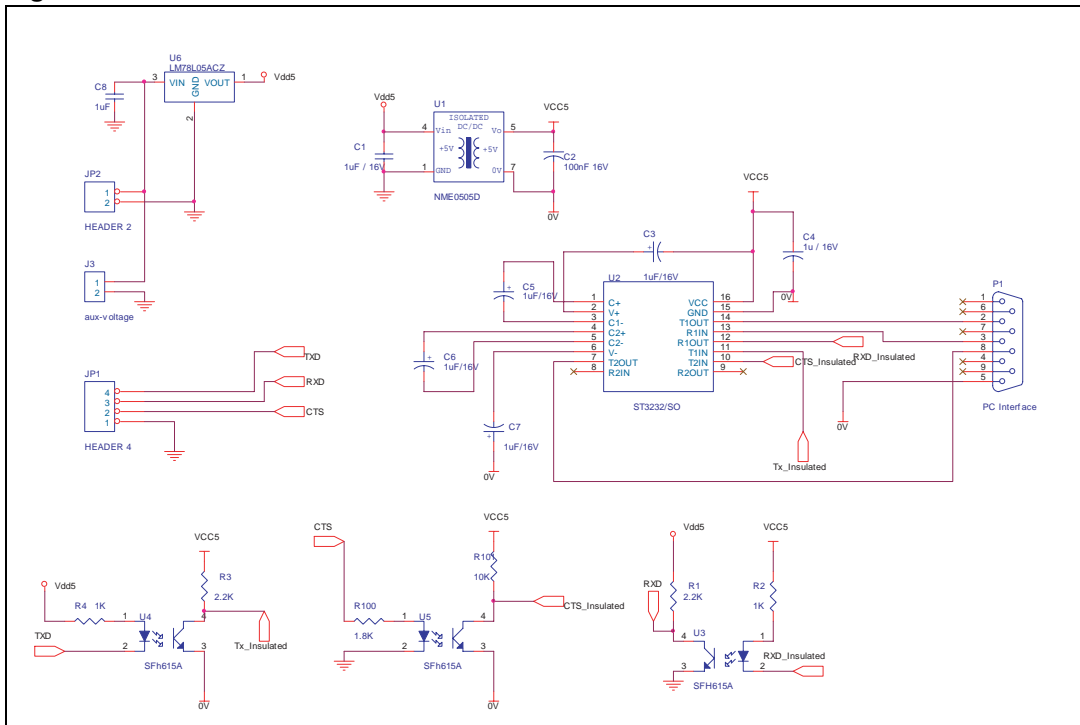
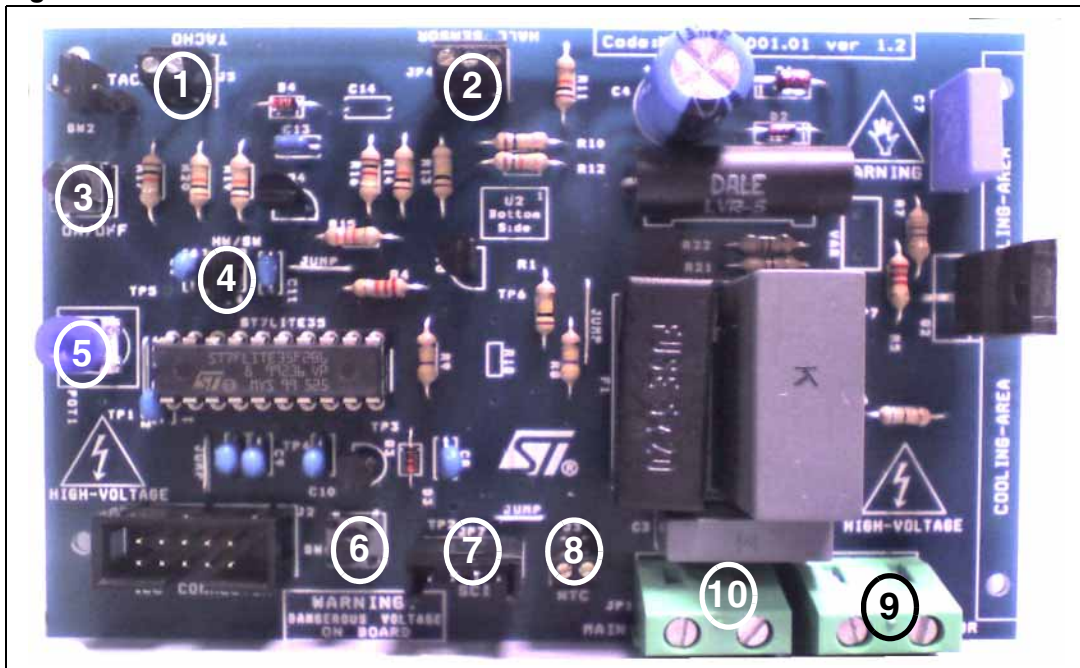


Figure 14. PC interface board schematic



In [Figure 15](#) the picture of the main board shows the main parts.

Figure 15. Main board



Note: See below for a description of labeled board components.

1. Tachometer sensor connector
2. Hall Effect sensor connector
3. Off/On push button (only in stand alone mode)
4. This jumper detects stand alone(no jumper present)/pc driven mode (jumper present)
5. Potentiometer for speed/ firing angle changing (if the system works in stand alone mode)
6. General switch
7. PC interface connector
8. NTC connector
9. Motor connector
10. Mains power connector
11. Testing points

6.1 Testing points

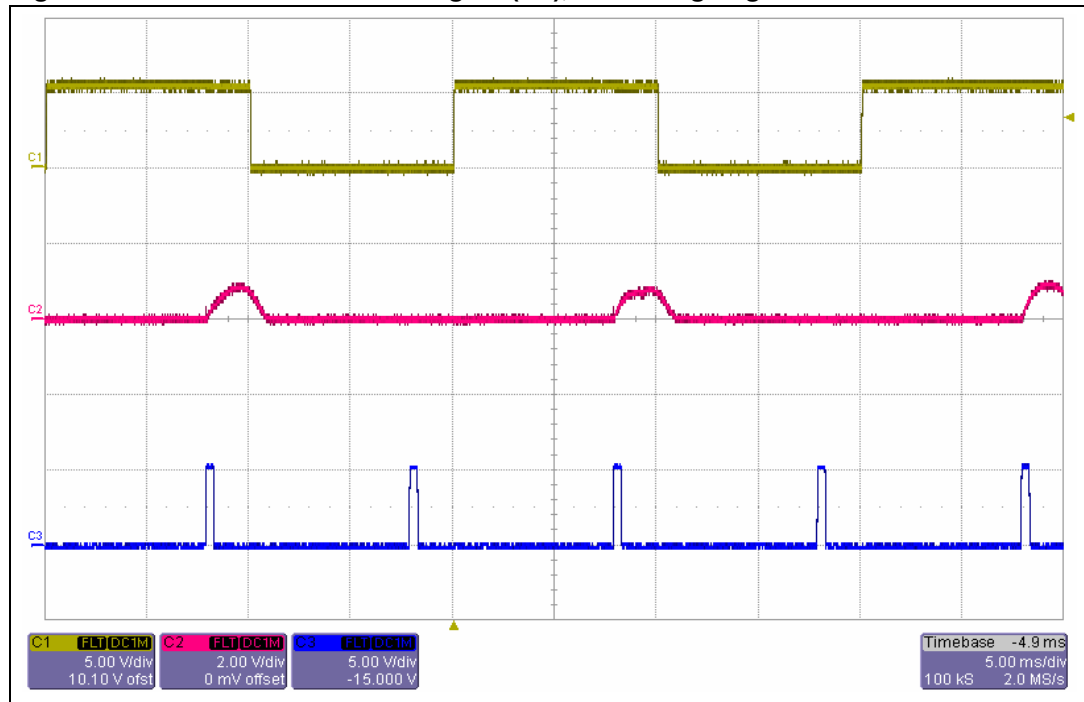
Some testing points are present on the board thus allowing the user to directly obtain and measure waveforms. Here follows a list of these points and of the expected waveforms.

- TP1: Vdd - 5V
- TP2: ground
- TP3: zero crossing signal - 0:5V square wave at 50 or 60Hz
- TP4: current sense (output signal of the OP-Amp) - see [Figure 16, 17, 18](#)
- TP5: speed sensor feedback - see [Figure 20](#)
- TP6: triac driving signal - see [Figure 21](#)
- TP7: ACST8 output - see [Figure 22, 23](#)

In all the figures below, channel C1 represents the zero crossing signal (@ 50Hz) and channel C3 pin PA4, i.e. the triac driving signal before the transistor, this in order to give an idea of the firing angle used in the examples.

Besides the waveforms on the testing points, [Figure 19](#) has been added in order to show the current on the ACST switch.

Figure 16. TP4 - Current Sense Signal (C2), 168° firing angle.



Note: The signal is different from zero only during half a period of the mains waveform.

Figure 17. TP4 - Current Sense Signal (C2), firing angle is decreasing

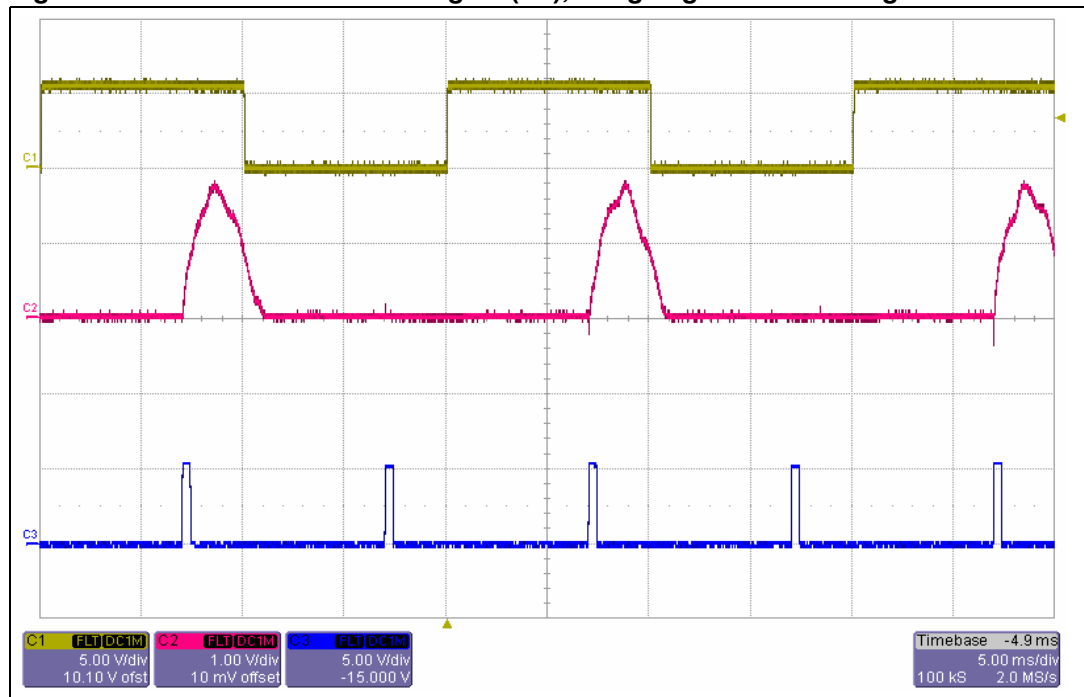


Figure 18. TP4 - Current Sense Signal (C2), 6° firing angle

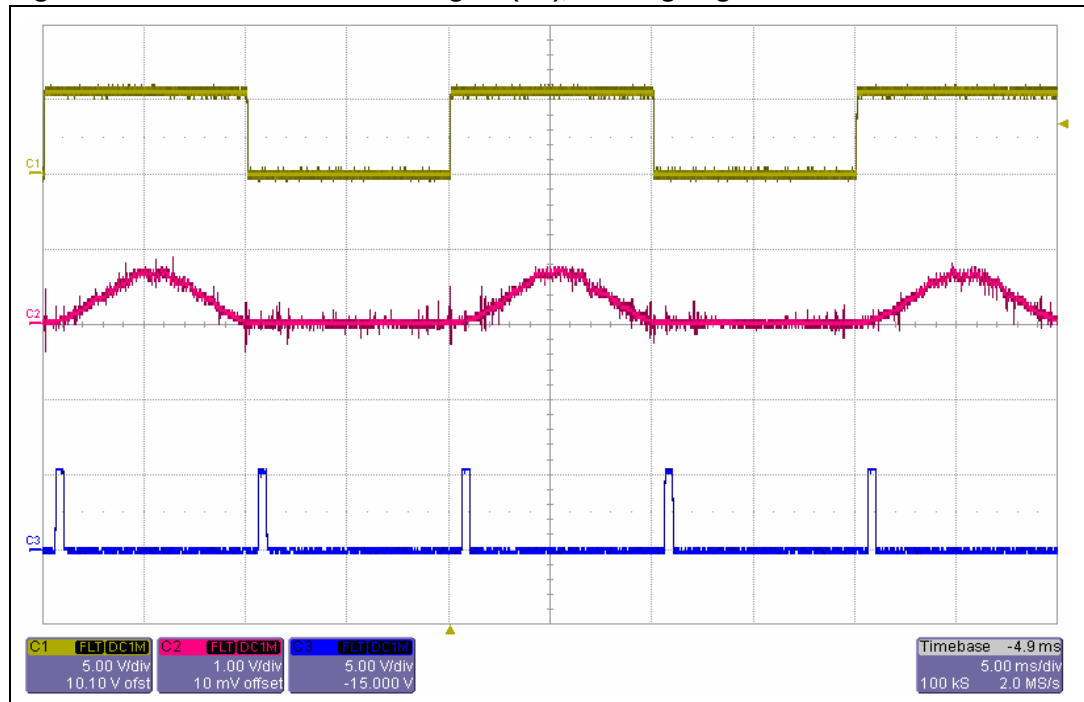


Figure 19. C1 - zero crossing signal. C2 - ACST driving signal. C4 - ACST switch current.

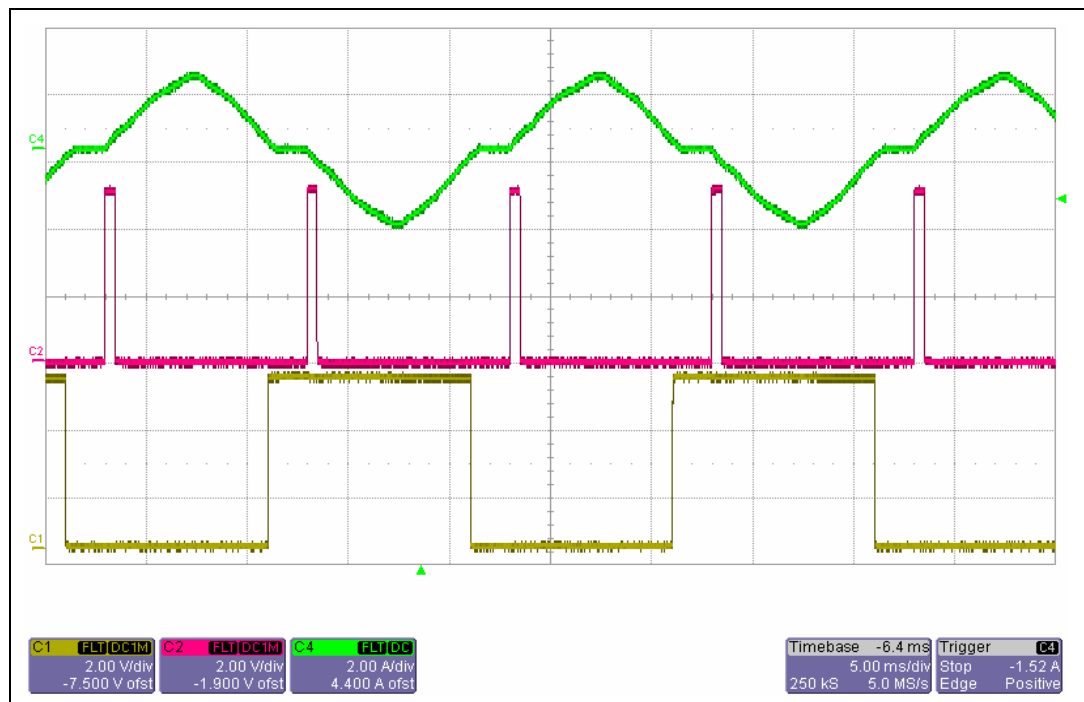


Figure 20. TP5 - Speed sensor feedback signal (C2), 3500 RPM

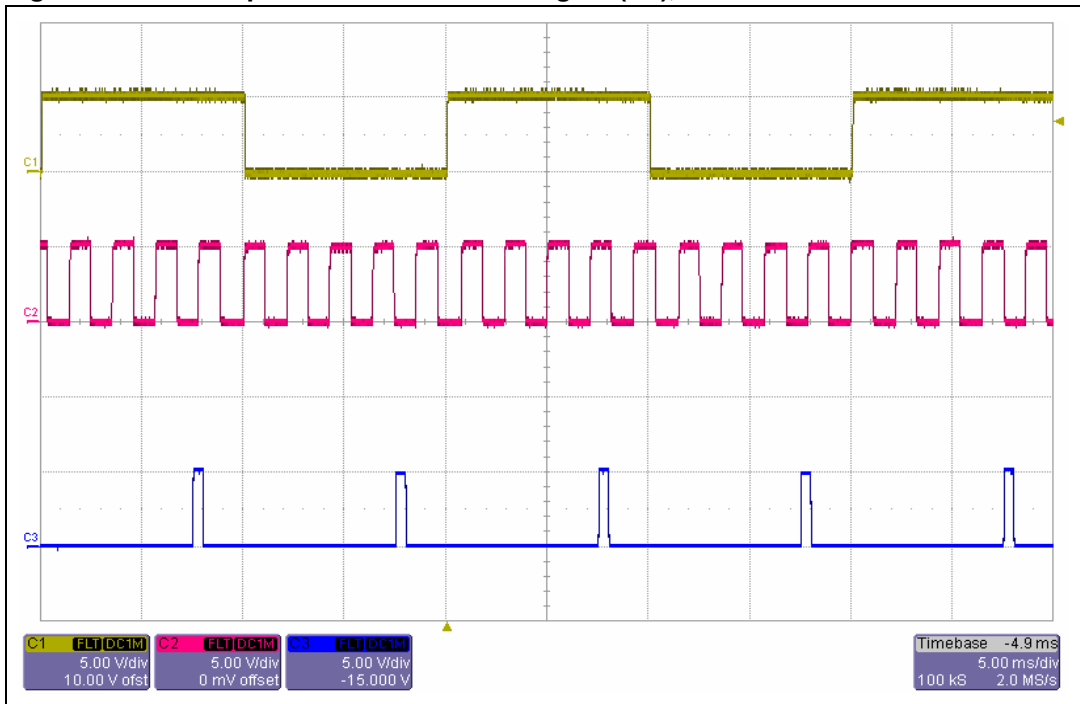


Figure 21. TP6 - Triac Driving Signal (C2)

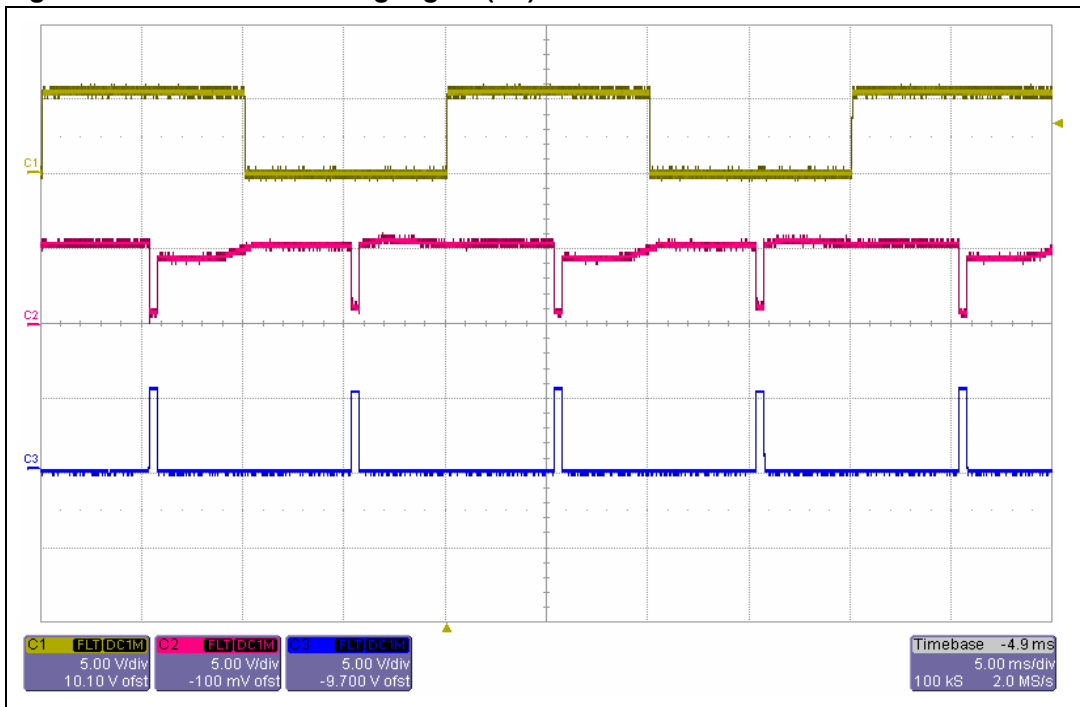


Figure 22. TP7 - ACST Output-COM voltage (C2) with firing angle equal to 167°

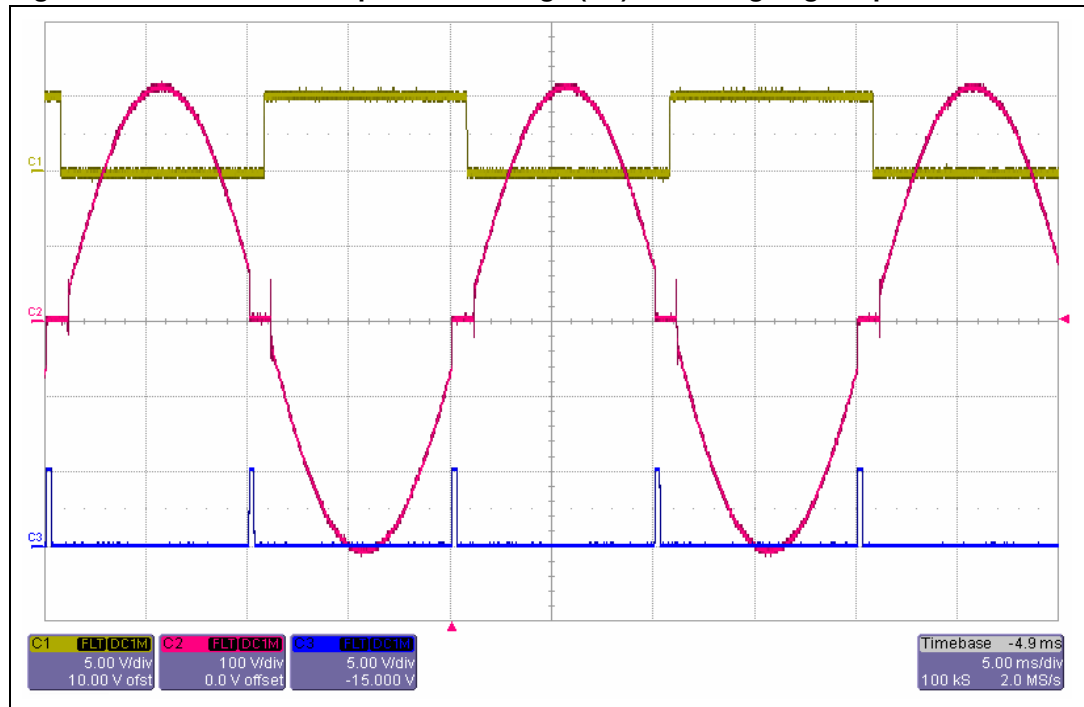
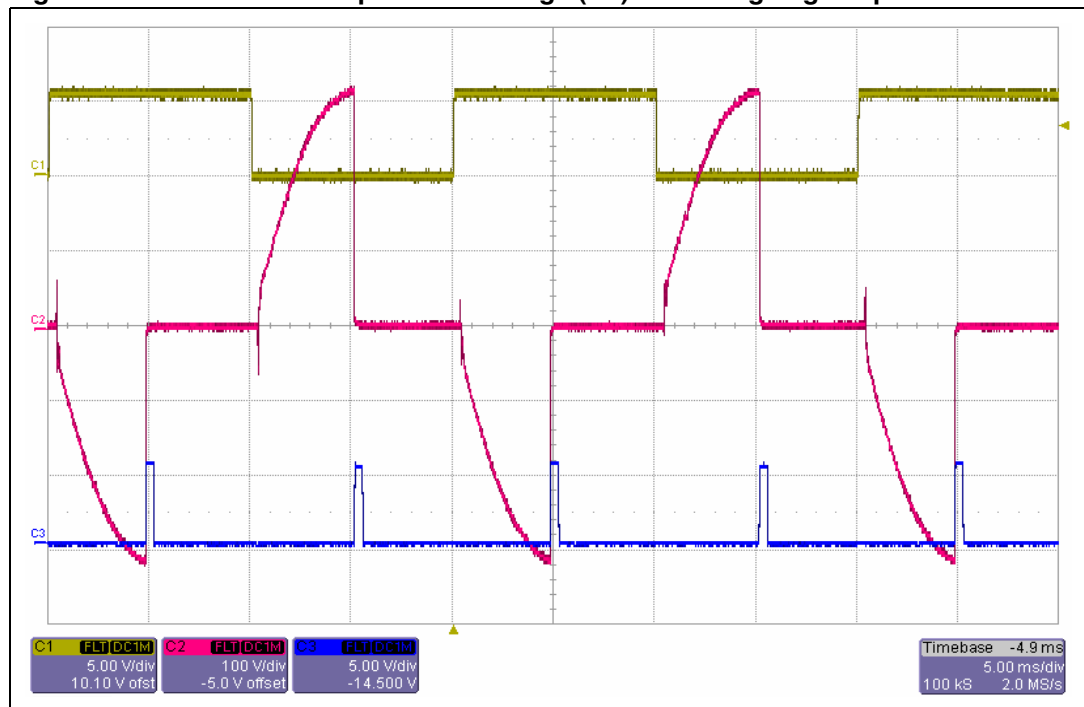


Figure 23. TP7 - ACST Output-COM voltage (C2) with firing angle equal to 90°



6.2 Current sense and overcurrent detection

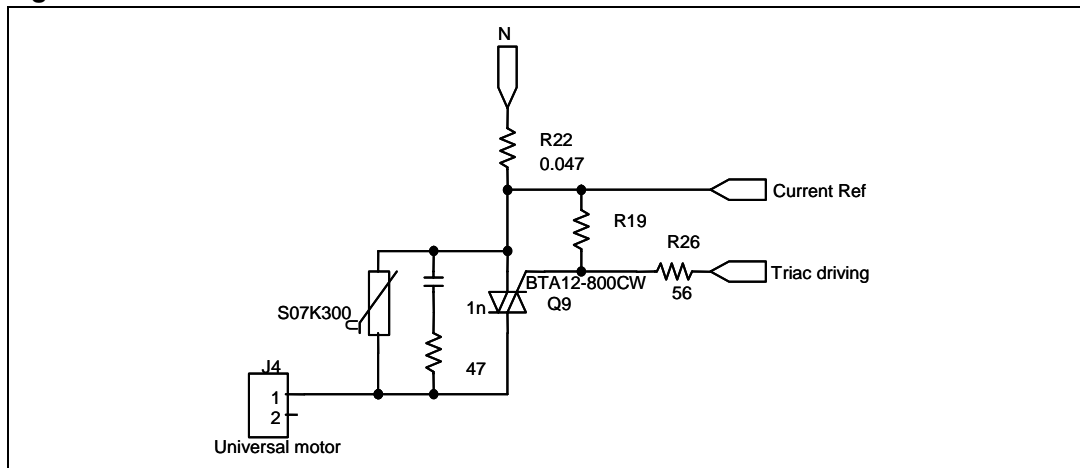
The voltage created by the current flowing through the sensing resistor R3 (0.047 Ohm) is amplified by ten so that the maximum detectable current is about 10A. The protection against overcurrent only takes place when the max current instead of the zero crossing current is measured. This parameter can be changed by software or, in stand alone mode, by modifying the *umc.h* file. In case the max current is the parameter measured, the value is acquired each 500µs and the protection takes place (motor is switched off) only if three consecutive values exceed the maximum allowed, this is in order to reduce measurement noise.

Measurement of maximum current is carried out during main power's positive half waveform.

6.3 Using BTA12

It's possible to place a BTA12-800CW (12 Amps RMS) Triac instead of the ACST8-8C. In this case a S07K300 varistor must be connected in parallel with the triac for over-voltages protection (see [Figure 24](#)).

Figure 24. Schematic variation when BTA12 is used.



6.4 Working at 60Hz

Please note that the kit works at 60Hz under nominal voltage supply conditions (i.e. 110V). For lower voltage values the capacitive power supply stage has to be modified, since it has been designed to work in the range 110V/230V.

7 Temperature limitation

Since the microcontroller uses its internal oscillator and PLL, which are temperature dependent, and since the system's correct behavior is strictly related to the oscillation frequency, the system has to be considered to operate in the temperature range that involves a maximum frequency deviation of:

- $\pm 3\%$ from the nominal 8MHz frequency if working at 50Hz or
- $-2\%:+3\%$ from the nominal 8MHz frequency if working at 60Hz

For higher frequencies the SCI communication with the PC fails, but the control still operates correctly.

Otherwise, the microcontroller needs to be recalibrated or an external oscillator must be used. Automatic real time calibration can also be implemented (refer to AN1324 for further information).

8 Package contents

The following material is supplied in the package:

- UMC Main board
- PC interface board
- One connector main board - PC interface board

9 Revision history

Table 1. Document revision history

Date	Revision	Changes
08-Sep-2006	1	Initial release.
14-Dec-2006	2	Safety warning added Section 1 on page 3

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST's terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST'S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED ST REPRESENTATIVE, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER'S OWN RISK.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

© 2006 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com

