UM0747
User manual

STEVAL-IHM026V1, motor control and touch sensing demonstration board, based on the STM8S

Introduction

This user manual describes the STEVAL-IHM026V1, a demonstration board with an embedded STM8S microcontroller.

The STEVAL-IHM026V1 demonstration board, by integrating the touch-sensing (TS) software library and the three-phase brushless direct current (BLDC) motor control (MC) software library, allows the implementation of a 3-phase brushless motor control application in which user control is managed through a few capacitive keys.

Other peripherals, not used in the demonstration firmware, have been included in the STEVAL-IHM026V1 board in order to make it flexible for further development.

The first part of this application note describes the layout and physical design guidelines to be used in touch-sensing applications. After a brief description of the TS software library and the three-phase BLDC motor control software library, the integration of both libraries within one motor control oriented firmware, including capacitive touch-sensitive control, is described.

Contents

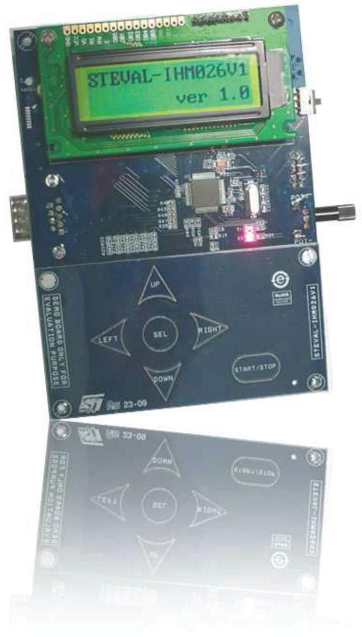
List of tables

List of figures

1 Features

- 6 V power supply
- 6 touch keys
- Motor control connector
- Sensorless or sensored motor control
- Standard or demonstration mode
- Real-time adjustment of motor control parameters
- HW provided with RS-232 connector, I^{2}C connector, IrDA transceiver
- SWIM and STice debug support
- 122 x 32 dot-matrix serial interface LCD module

Figure 1. STEVAL-IHM026V1 board



AM04436v1

2 Resistor-capacitor (RC) acquisition principle

In applications requiring user interface, touch-sensitive controls are becoming the solution of choice to replace conventional electro-mechanical switches. Among them are solutions based on SAW (surface acoustic waves), infrared, camera-based solutions and the capacitive solution, which is discussed in this user manual.

One or more sensing elements, the capacitor electrodes, are placed on the board. Each of them, together with a resistor, forms an RC network. The operating principle is that the capacitor is modified when a finger gets close to the sensing electrode, i.e. the finger changes the dielectric properties of the electrode. A capacitor is in fact characterized by the following formula:

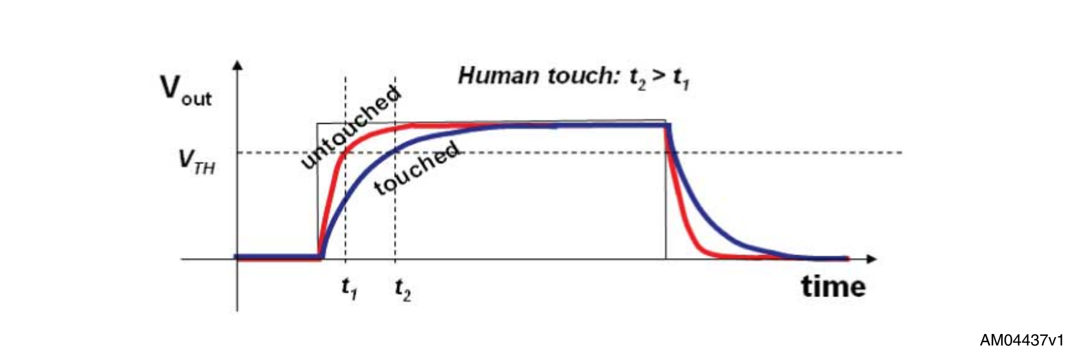
Equation 1

$$C = \frac{\varepsilon_0 \varepsilon_r A}{d}$$

where A is the electrode area, ε_r the relative dielectric constant of the insulator, ε_0 the dielectric constant of the vacuum, and d the distance between the two electrodes.

Electrode capacitance C is periodically charged and discharged through a fixed resistor. The time necessary to charge the capacitor from one reference voltage to another, measured with a timer, changes in accordance with the presence/absence of the finger (see *Figure 2*).

Figure 2. Touch acquisition



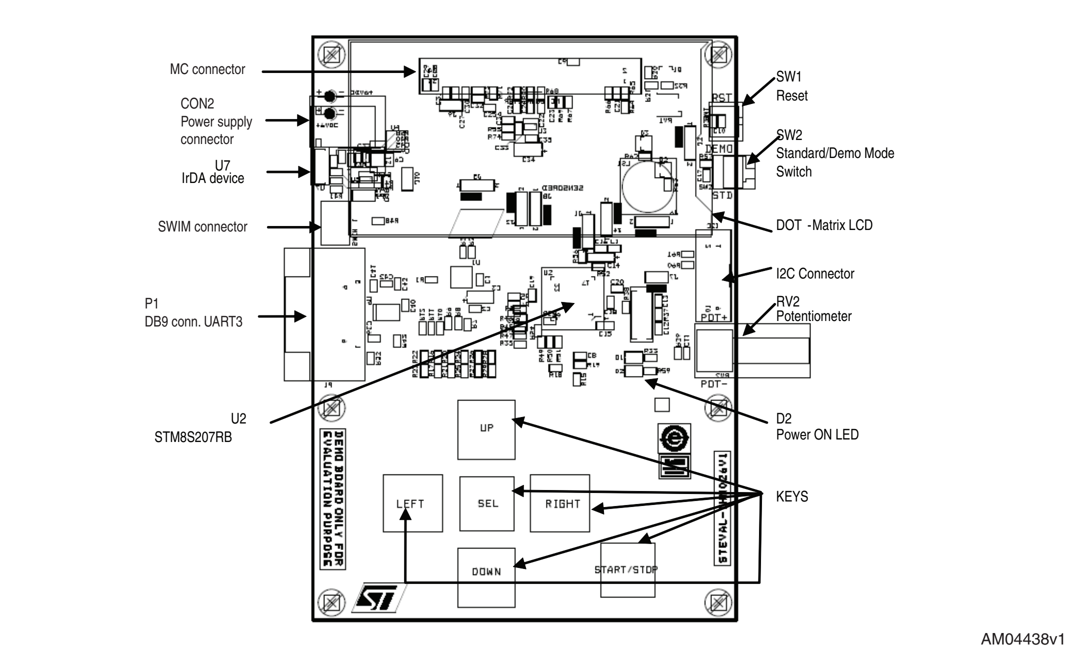
Any time variation detected is then filtered and eventually sent to a host system using dedicated I/Os or I^2C or SPI interfaces.

3 Hardware description

3.1 Hardware layout and configuration

The STEVAL-IHM026V1 demonstration board is designed around the STM8S2xx microcontroller in a 64-pin LQFP package. The hardware block diagram in *Figure 3* shows the connection between the STM8S2xx and peripherals (LCD, I^2C, UART, IrDA, LIN and motor control) and is used to locate these components on the actual board.

Figure 3. STEVAL-IHM026V1 board layout: top and bottom



3.2 Power supply

The STEVAL-IHM026V1 board is designed to be powered by a 6 V DC power supply and to be protected by the PolyZen U4 from any incorrect power connection. It is possible to configure the demonstration board to use any of following sources for the power supply.

- 6 V DC power adapter connected to the power jack on the board, 6 VDC1
- 5 V DC power from MC connector (pin 25)

The LED D2 is lit when the STEVAL-IHM026V1 board is powered on.

3.3 Clock source

The clock to the STM8S2xx microcontroller is provided by a 16 MHz oscillator Y1.

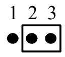
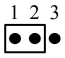
3.4 Reset source

The reset signal of STM8S2xx microcontroller is active low and the reset sources include:

- A reset button SW1
- Debugging tools from the SWIM connector

The reset pin PA0 of the STM8S2xx microcontroller is connected to either reset button SW1 or GND by the setting of jumper J3:

Table 1. Reset jumper

Jumper	Description
J3	PA0 is connected to GND when J3 is set as shown on the right. It's used to keep the STM8S2xx always on reset when STice is connected to the board. 
	PA0 is connected to the reset button when J3 is set as shown on the right. (default setting) 

3.5 Development and debug support

The debug connector available on the STEVAL-IHM026V1 board is the SWIM connector (the ERNI 4-pin connector for SWIM debugging).

3.6 Display and input devices

The 122 x 32 dot-matrix, serial interface LCD is available as a display device. See *Table 2* for description.

Table 2. LCD connector

Pin nb. in LCD connector	Description
1	Vss
2	Vdd
3	VO contrast adjust
4	Serial clock
5	Serial data input
6	Chip select
7	LED backlight +
8	LED backlight -

A potentiometer (RV2), a switch (SW2) and six capacitive keys are available as input devices (see *Section 5.2* and *5.3*).

3.7 Motor control connector

On its bottom side, the STEVAL-IHM026V1 demonstration board embeds one 3-phase brushless motor control via a dedicated 34-pin connector (the motor control connector or MC connector) which provides all required control and feedback signals to and from the motor power-driving board.

Available signals include an emergency stop, motor current, bus voltage sensing, heatsink temperature sensing coming from the motor driving board, 6 channels of PWM control signals going to the motor driving circuit, and the position/speed feedback coming from the BEMF sensing network or from the hall sensors.

The MC main connector features additional sensing signals (AC bus voltage, DC bus voltage, AC input current) for managing the PFC (power factor control) controller.

The I/O pins used on the motor control connectors are multiplexed with some peripherals on the board. Either motor control connectors or multiplexed peripherals can be enabled by setting the jumpers listed below:

Table 3. Motor control jumpers

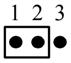
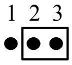
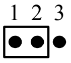
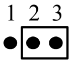
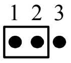
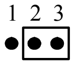
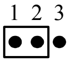
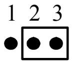
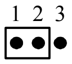
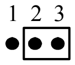
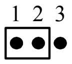
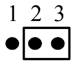
Jumper	Description	Default configuration in STEVAL-IHM026V1
J1	A) Hall sensors feedback enabled when J1 is set as shown:  B) Sensorless feedback (BEMF sensing) enabled when J1 is set as shown: 	B
J2	A) Hall sensors feedback enabled when J2 is set as shown:  B) Sensorless feedback (BEMF sensing) enabled when J2 is set as shown: 	B
J4	A) Hall sensors feedback enabled when J4 is set as shown:  B) Sensorless feedback (BEMF sensing) enabled when J4 is set as shown: 	B
J5	A) HW setup for dissipative brake when J5 is set as shown:  B) HW setup for PFC when J5 is set as shown: 	A

Table 3. Motor control jumpers (continued)

Jumper	Description	Default configuration in STEVAL-IHM026V1
J6	A) Beeper enabled when J6 is set as shown:  B) HW setup for PFC when J6 is set as shown: 	B
J8[1]	A) TIM2_CH2 enabled when J8 is set as shown (sensorless mode):  B) TIM1_CH4 enabled when J8 is set as shown (sensored mode): 	A

[1] Disregard the silkscreen relating to jumper J8 printed on the board.

Note: The PFC feature is not currently implemented by the firmware. The hardware compatibility is reserved for future development.

Table 4. Motor control connector

Description	Pin nb. in MC connector	Pin in STM8 MCU
Emergency stop	1	PE3/Tim1_BKIN
MC-UH	3	PC1/TIM1_CH1
MC -UL	5	PB0/TIM1_CH1N
MC -VH	7	PC2/TIM1_CH2
MC -VL	9	PB1/TIM1_CH2N
MC -WH	11	PC3/TIM1_CH3
MC -WL	13	PB2/TIM1_CH3N
Phase A current	15	PB6
Phase B current	17	PB5
Phase C current	19	PB4
NTC bypass relay	21	PE4
Dissipative brake	23	PD0/TIM3_CH2/
+5 V power	25	-
PFC SYNC	27	PD2/TIM3_CH1/[TIM2_CH3]
PFC PWM	29	PD0/TIM3_CH2/[TIM1_BKIN]/[CLK_CCO]
Encoder A	31	PD3/TIM2_CH2 or PF0/AIN10
Encoder B	33	PD4/TIM2_CH1 or PF2/AIN12
Bus voltage	14	PB7/AIN7
Heatsink temperature	26	PE7
+3.3 V power	28	-

Table 4. Motor control connector (continued)

Description	Pin nb. in MC connector	Pin in STM8 MCU
Encoder Index	34	PA3/TIM2_CH3 or PF3/AIN11
GND	2,4,6,8,10,12,16,18,20,22, 24,30,32	-

3.8 Other peripheral connections

Although their use is not implemented in the demonstration firmware, other interfaces have been embedded in order to make the board adaptable to user application needs. The STEVAL-IHM026V1 may be in fact interfaced with other boards by means of a serial interface, an I^{2}C interface, or by an IrDA transceiver module.

The male D-type 9-pin connector P1, available on the STEVAL-IHM026V1 board, connects to the microcontroller's USART3 port through the U6 RS232 device.

IrDA communication is supported by the IrDA transceiver U7 connected to the UART1 of the STM8S2xx. It can be enabled or disabled by setting JP1:

Table 5. IrDA related jumper

Jumper	Description
JP1	JP1 is used to shut down the IrDA transceiver. IrDA is enabled when JP1 is fitted, while IrDA is disabled when JP1 is not fitted. Resistors R41, R44, R46 have to be fitted too. Default status JP1, R41, R44, R46: NOT fitted

3.9 Touch-sensing design

3.9.1 PCB general guideline

When designing printed circuit boards (PCBs) for a capacitive sensing application, it is important to take into account more than just the circuit that is involved directly with the sensing. Not only hardware elements (capacitance, resistors, LEDs, etc.) may introduce parasitic capacitance, but also tracks could couple with sensing elements and modify them. Simple guidelines can help in achieving good performance in applications.

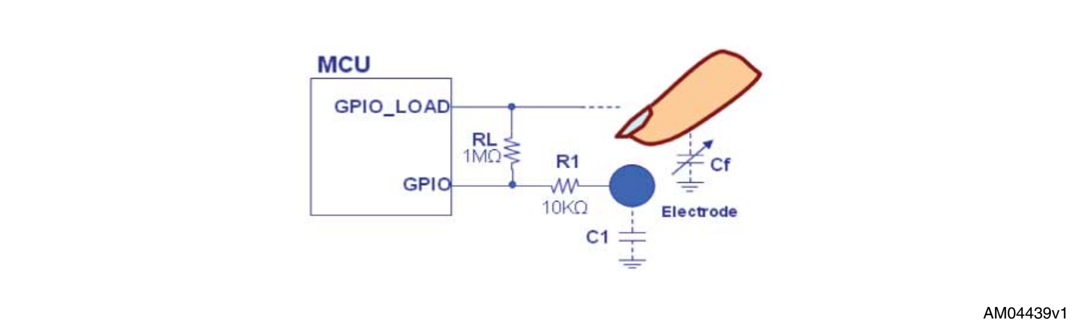
Figure 4 shows an example of implementation. The finger touch may be schematized as another capacitor Cf (few pF) in parallel with the touched existing electrode capacitance C1 on the PCB.

One I/O pin of the microcontroller, the load pin (KEY_LOAD), is used to charge the RC network formed by the capacitors C1, Cf and resistors RL, R1.

Resistor RL (whose value varies between hundreds of ohms to a few MΩ) and R1 (10 kΩ) must be placed as close as possible to the MCU. RL is used to adjust the electrode sensitivity, while resistor R1 is used to reduce noise sensitivity.

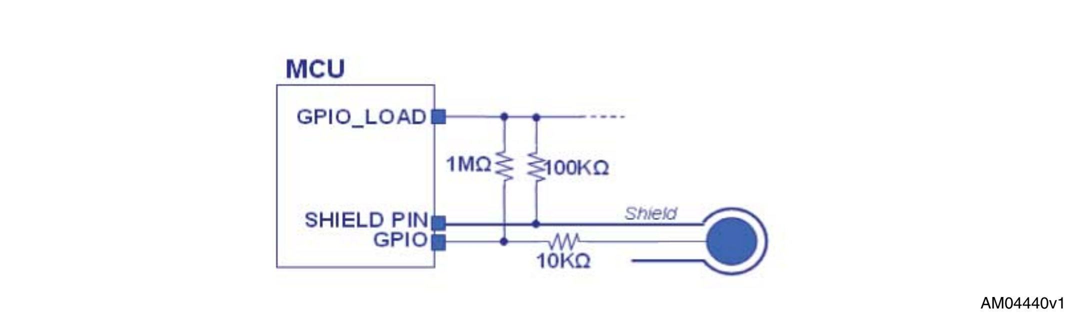
It is good practice to keep the area covered by the sensing elements and their tracks quite small. This can be achieved by reducing the distance between the controller device and the sensors to a minimum (less than 100 mm, if possible), by using thin tracks (as the PCB technology allows), and by putting sensing resistors as close as possible to the MCU (5 mm or less). Moreover adjacent sensitive tracks should be at least twice the track width from each other, while a greater distance is necessary from the ground plane or other application tracks (LEDs, power, communication peripherals, etc.).

Figure 4. RC circuit



In a noisy environment, the most sensitive area may require shielding. It could be a ground plane or an active shield. An active shield or driven shield may be implemented by the load pin or by a dedicated pin. The principle of the driven shield is to drive the shield with the same signal as the electrodes. An example of a driven shield is shown in *Figure 5*.

Figure 5. Driven shield

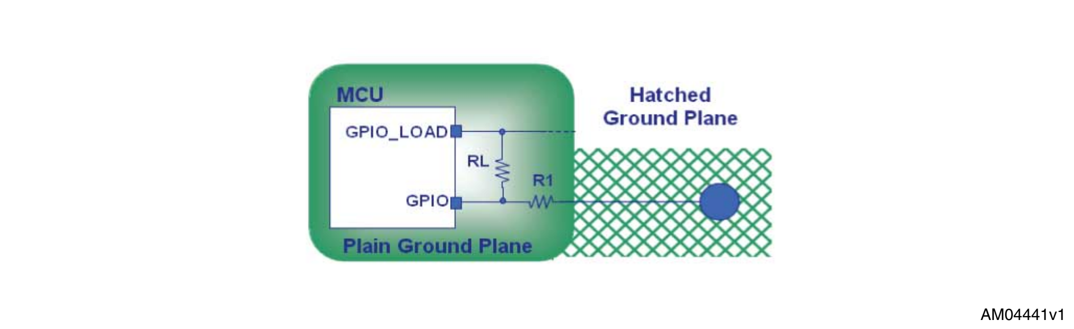


3.9.2 Power supply and ground plane

Other recommendations concern the ground plane and power supply. The power plane must never be placed under a wheel or slider, although it is possible to have it under buttons or sensing tracks, but in this case a hatched pattern with less than 40% copper on the farthest layer must be used. In noisy environments, the most sensitive area may require a ground plane (see *Figure 6*).

As for the power supply, since even small variations directly impact the RC charging time, a voltage regulator is strongly recommended.

Figure 6. Ground plane

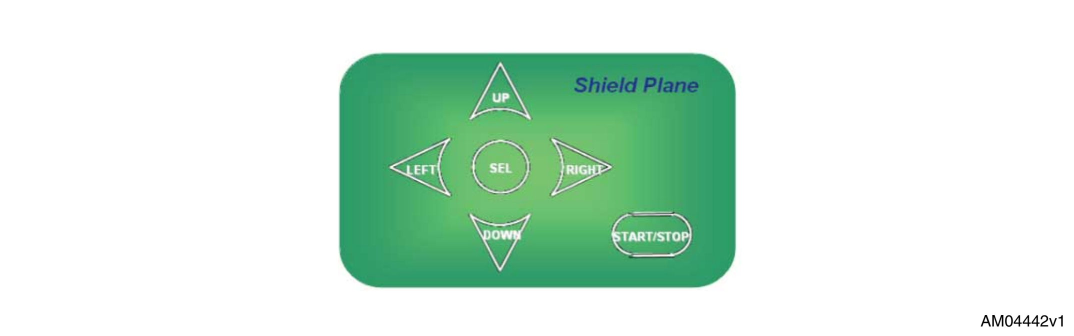


In the STEVAL-IHM026V1 board a full ground plane has been used outside the sensing area, while a full shield plane has been used around and under the sensitive tracks.

3.9.3 Electrode design

Any kind of shape can be used for the buttons since it affects only aesthetics. The size is important since it affects the sensitivity. Good practice is to make them similar to fingertip size. In the STEVAL-IHM026V1 board, no wheels or slider are present, only six keys have been implemented: up, down, left, right, SEL and start/stop (see *Figure 7*).

Figure 7. Command keys



3.9.4 Panel construction

The front panel, made of plexiglas, can be glued to the sensors with pressure sensitive adhesive (PSA). Otherwise it is possible to press the sensors against the panel without glue, but in this case moisture or small air bubbles could be trapped.

In the described application the panel is simply screwed on the PCB.

4 Firmware implementation

4.1 Touch-sensing library

4.1.1 Library description

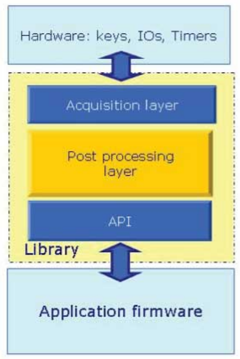
The TS library allows the STM8 microcontrollers to manage touch-sensing keys, wheels and sliders with a minimum number of external components.

The STM8 touch-sensing library:

- supports up to 24 keys distributed over three GPIO ports
- supports up to 2 wheels or sliders with two hardware implementations
- includes configurable firmware for improved adaptablility
- has advanced processing algorithms to filter out noise and to compensate temperature and power supply variations.

The core of the library consists of two state machines: the key state machine and the main state machine. The first level is the acquisition level, dedicated to the management of timers and I/O ports for RC charge/discharge time acquisition. The API level is made up of all functions that are visible from the user application (see *Figure 8*).

Figure 8. Library architecture

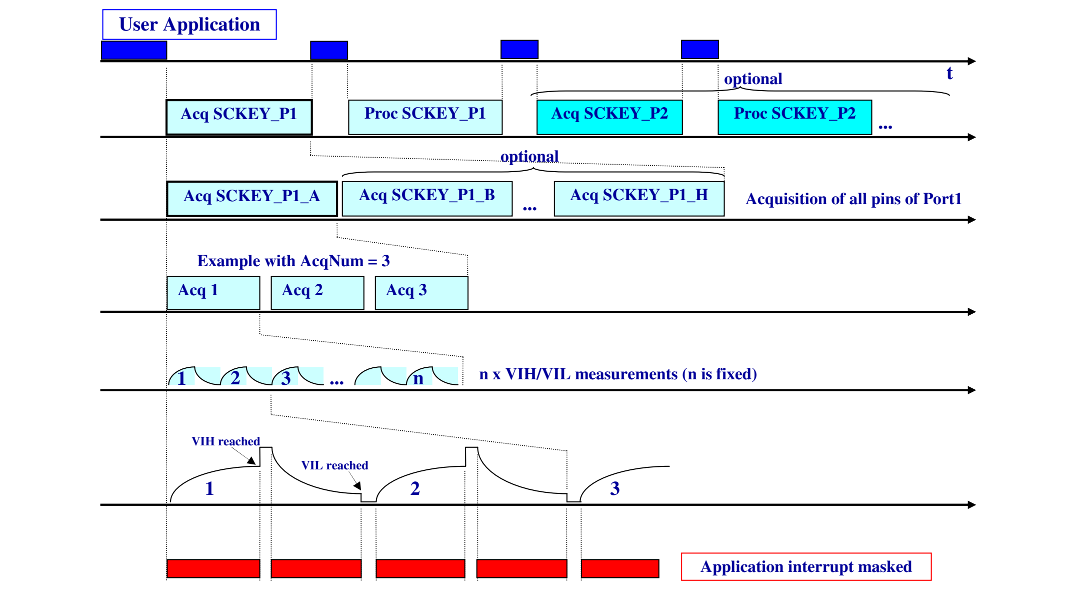


AM04443v1

Two timers are used: one is used for RC charge/discharge time measurements and the other one provides a time base for the application.

As of today, touch keys can be split among a maximum of 3 different GPIO ports. The acquisition phase of each port is followed by a processing phase. Each acquisition consists of various measures of charge/discharge times (AcqNum = 3 in the STEVAL-IHM026V1 demonstration firmware), and each measure is made up of several elementary measurements (n = 8 in the STEVAL-IHM026V1 demonstration firmware), see *Figure 9* for the acquisition sequence scheme.

Figure 9. Acquisition sequence



For a detailed TSL description see the "STM8 touch sensing library firmware on-line help manual" (stm8s_fwlib_um.chm available in TSL software package).

4.1.2 Library usage

The STM8_TSL_RC_Configuration.h file contains all configurable options for compilation. It must be modified by the user to adapt it to his specific application. Both parameters for acquisition (related to the dielectric used, to the sensitivity of the keys, to the desired resolution, etc.) and hardware configuration (microcontroller family, acquisition timer used, pins and ports linked to single and multiple sensing keys, etc.) must be set here.

Briefly, as shown below, in order to use the TS library:

1. include the STM8_TSL_RC_API.h file
2. call the TSL_Init() function to initialize the TSL state machine
3. call the TSL_Action() function in a continuous loop.

```
#include "STM8_TSL_RC_API.h"


void main(void)

{

     Application_Init();

    TSL_Init();

     for (;;)

              {

                   TSL_Action();

                   Do_Application();

     }

}
```

For TSL description details see "STM8 touch sensing library firmware on-line help manual" (stm8s_fwlib_um.chm available in TSL software package).

4.2 Three-phase BLDC motor control library overview

The STM8Sxxx three-phase BLDC motor control software library (also called in the following MC library) is a scalar motor control firmware library for 3-phase permanent magnet motors developed for the STM8Sxx microcontrollers.

These 8-bit, ST microcontrollers (STM8Sxx) come with a set of peripherals that makes them suitable for performing both permanent-magnet and AC induction motor scalar control. The control may operate in open or closed loop, in current or voltage mode. Moreover, it is possible to get rid of any speed sensor thanks to the sensorless algorithm for rotor position reconstruction based on BEMF detection.

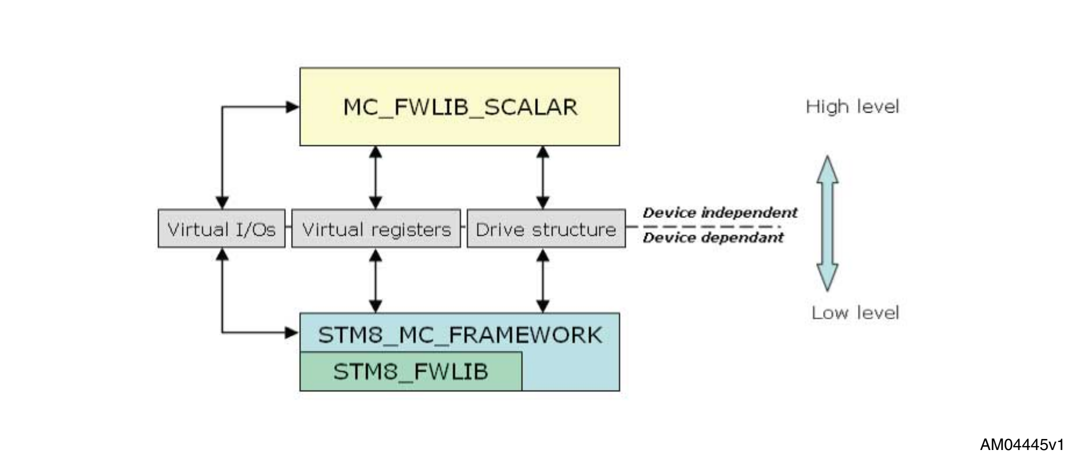
The MC software library is logically subdivided in three different parts:

- MC_FWLIB_SCALAR - includes the "high level" hardware-independent motor control modules
- STM8_FWLIB - includes the STM8 standard libraries
- STM8_MC_FRAMEWORK - contains the "low level" hardware-dependant motor control routines

The "high level" modules contain the algorithms that are device independent while the "low level" modules contain the code that is related to the hardware, so it means that only the "low level" directly interact with the peripherals and the interrupt service routines of the microcontroller.

The "high level" modules interact with the "low level" modules mainly using three "interfaces": the "virtual registers", the "virtual I/Os" and the "drive structure" (see *Figure 10*).

Figure 10. High level/low level interface



Note: For a detailed description see the “STM8Sxxx three-phase BLDC motor control software library” user manual (UM0708).

4.3 Integrating the TS library and three-phase BLDC MC library

In order to merge the two libraries into a unique firmware, some changes to both libraries must be performed.

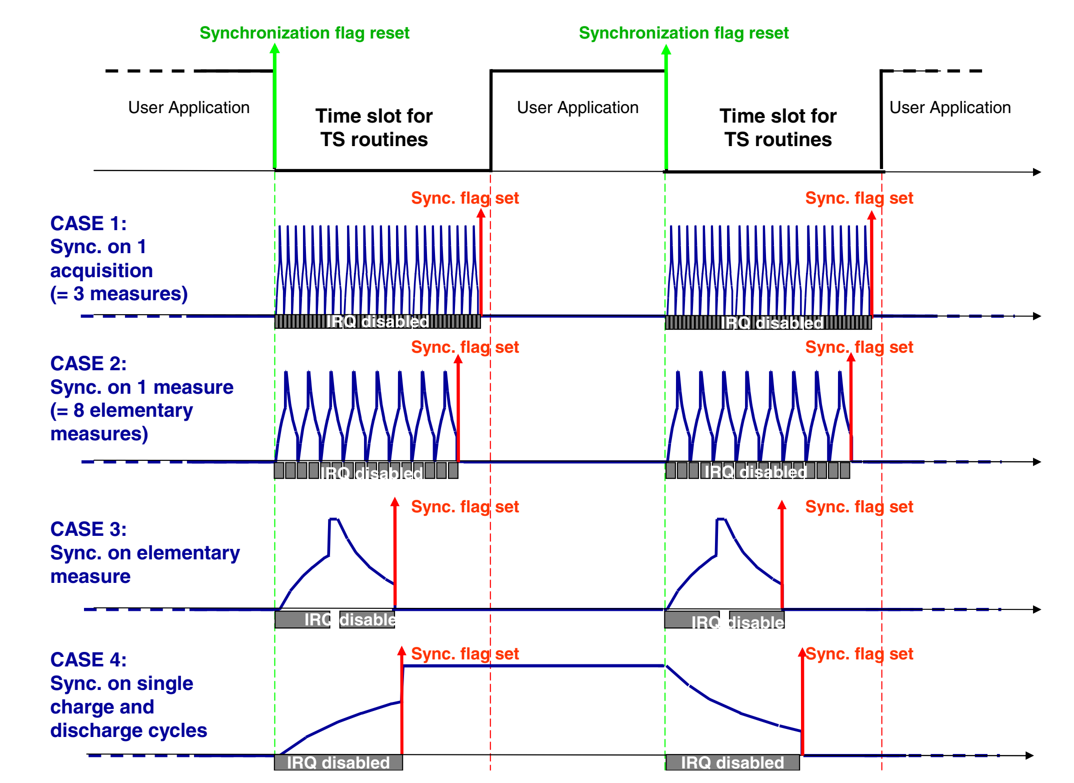
The first step is to share/merge MCU resources, then the two state machines (MC and TS) must be synchronized.

As for sharing resources, since the MC library uses the two timers TIM1, TIM2 for motor control operations and one timer TIM4 as a time base generator, letting TIM3 free, and since the TS library uses TIM3 for touch-sensing operation and TIM4 as a time base generator, this last timer (TIM4) is shared and works as a time base generator both for motor control and for touch-sensing functions. The other timers are not affected by the integration.

The idea for TS SW library integration within other libraries/firmware is to split the operations needed for the identification of keys into elementary operations and to synchronize them within the main program, as illustrated in *Figure 11* where four examples of possible synchronization are shown. The whole acquisition may in fact separate into (see also *Figure 12*):

- single complete acquisitions, each made up of "AcqNum" measures
- single measures, each made up of 8 elementary measures
- elementary measures, each made up of one charge and one discharge cycle
- single charge and discharge cycles.

Figure 11. Possible synchronizations


Synchronization flag reset
Synchronization flag reset
User Application
Time slot for TS routines
User Application
Time slot for TS routines
User Application
Sync. flag set
Sync. flag set
CASE 1: Sync. on 1 acquisition (= 3 measures)
IRQ disabled
IRQ disabled
CASE 2: Sync. on 1 measure (= 8 elementary measures)
IRQ disabled
IRQ disabled
CASE 3: Sync. on elementary measure
IRQ disable
IRQ disable
CASE 4: Sync. on single charge and discharge cycles
IRQ disabled
IRQ disabled

A synchronization flag is reset at the end of the user application routine (i.e. at beginning of the free time slot for touch-sensing routines) to enable the TS routine to start. At the end of it (i.e. at the end of a charge cycle or at the end of a whole elementary measure, etc.), the synchronization flag is set in order to prevent TS routines from continuing.

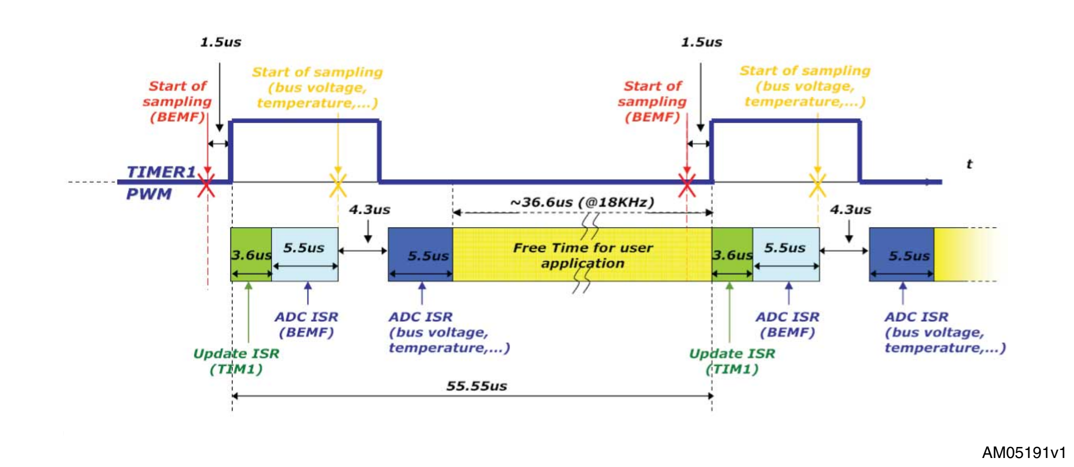
It is user's responsibility to choose how to split the TS acquisition routine according to the free time slot available, making sure that the elementary TS routine stops within that time gap.

The diagram in *Figure 12* shows the standard motor control sequences (with a 16 MHz clock oscillator). It is not commented in detail (for a detailed description see UM0708 “STM8Sxxx three-phase BLDC motor control software library” user manual). What is important for the integration within the TS library is the time freed for the user application represented by the yellow window which is the time that can be spent for the touch-sensing acquisition and detection. Thus the TS acquisition must be split and synchronized with the MC routines in order to take place only during the correct time slot.

Typical PWM frequency used for motor control runs from 10 kHz up to 20 kHz. The default value of the PWM frequency set in the STEVAL-IHM026V1 demonstration firmware is 18 kHz so the Timer1 PWM period is about 55.5 µs. The time slot available for TS routines is thus longer than 36 µs (see *Figure 12*). Since one single RC charge/discharge lasts about 20 µs, by splitting sensing acquisitions into single charge/discharge operations we are sure to not exceed the free time slot.

The synchronization is achieved by the flag IT_Sync_Flags.start. This flag - set at the end of the ADC interrupt service routine managing bus voltage acquisition, or temperature, or other (blue window in *Figure 12*) - allows one RC charge or discharge cycle to start again. If this flag is not set, the application stalls within the "TSL_IO_Acquisition" function, the function that handles RC charge/discharge time measurement.

Figure 12. Times of standard motor control tasks (at 18 kHz PWM frequency)



The timings of the motor control tasks may vary from the standard behavior described in *Figure 12*.

In fact, both the Timer1 update routine (green window in *Figure 12*) and the ADC interrupt routine (light blue window in *Figure 12*) may last longer than the given times (3.6 µs and 5.5 µs respectively).

When the conditions for motor commutation are detected, the ADC interrupt routine may last up to 20 µs (*Figure 13*), while when commutation and demagnetization are managed within the TIM1 update interrupt routine, it lasts about 21 µs, see *Figure 14* (see UM0708 “STM8Sxxx three-phase BLDC motor control software library” user manual for details).

The occurrence of these longer routines depends on the motor speed. For example at 1500 rpm each of them happens every 3.3 ms, while they occur every 1.7 ms at 3000 rpm.

In both cases, ADC ISR lasts longer or TIM1 ISR lasts longer, the free time left for the user application decreases to not allow the TS routine to take place. Thus another flag is set at the end of these longer ISRs in order to prevent the IT_Sync_Flags.start from being set.

Figure 13. Times of motor control tasks when longer ADC ISR occurs (at 18 kHz PWM frequency)

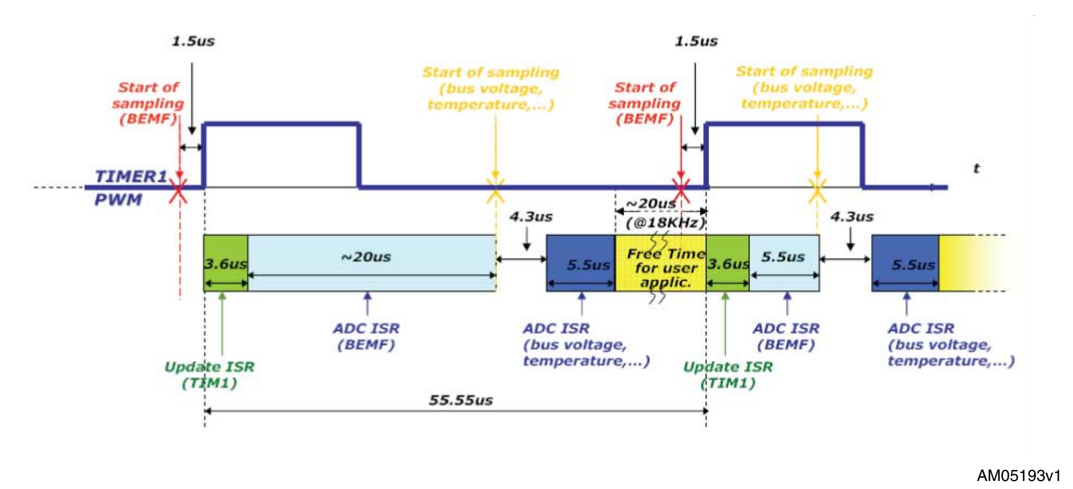
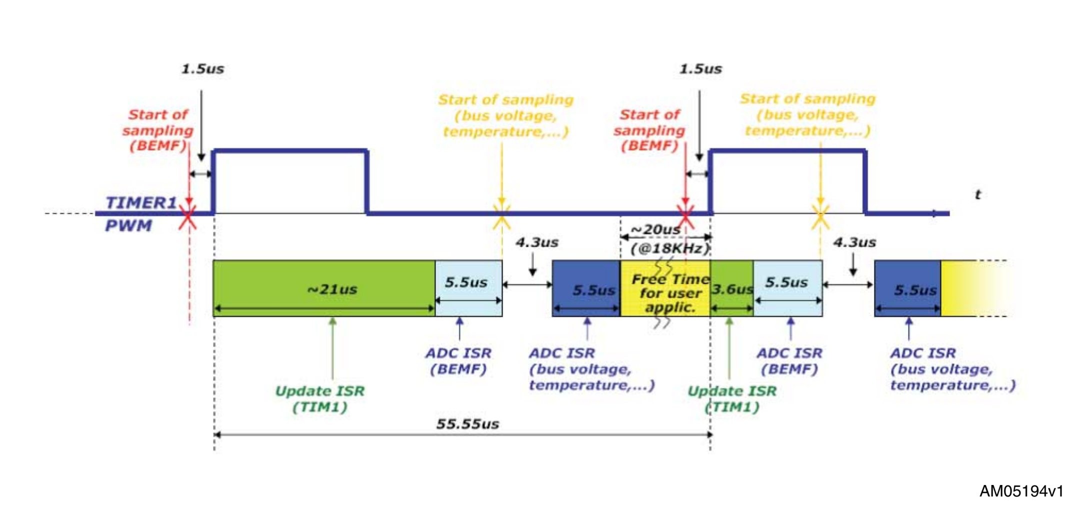
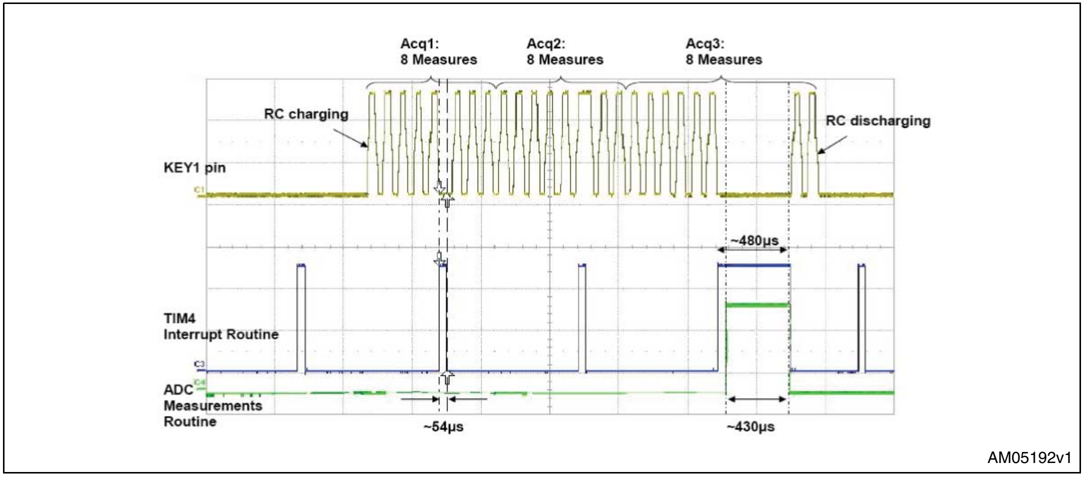


Figure 14. Times of motor control tasks when longer TIM1 ISR occurs (at 18 kHz PWM frequency)



In *Figure 12*, *13*, *14* the operations called in the TIM4 (time base generator timer) interrupt routine are not taken into account. Those operations, occurring every 1 ms, may postpone the RC charge and discharge cycles. Typically the delay is one TIMER1 PWM cycle, but each 10 ms the ADC reading operation takes place, thus postponing the charge/discharge cycles by about 500 µs (see *Figure 15*). This delay is not critical from a touch-sensing point of view.

Figure 15. Timer4 IRQ



4.4 Restrictions

In order to integrate the two libraries, the time acquisition routine of the TS library has been divided into single RC charge and discharge sections. A further division of the algorithm into smaller sections is not possible. So the motor control SW library has to assure that the free time for the user application (i.e. for capacitor charge/discharge time measurement) lasts the time necessary for a single charge or discharge time.

Considering that the typical PWM frequency used for motor control is in the range 10 kHz to 20 kHz, the stricter condition is the 20 kHz PWM frequency. In this case the free time available for the user application (yellow window in *Figure 12*), i.e. for single charge/discharge time acquisitions, lasts more than 31 µs which is still not critical for the touch-sensing algorithm (approximately 20 µs for RC charge or discharge cycles).

On the other hand, the TS algorithm may have some restrictions. First of all, the TS SW library is built to work with up to 3 independent I/O ports for keys (i.e. 24 keys) and 2 additional I/O ports for wheels/sliders. In this application wheels and sliders are not considered. If the driven shield is used, it takes one additional I/O pin per port which leads to 21 keys maximum.

Response time to user touch can be noticed only if above 150 ms. In order to keep it below this threshold, the delay due to the integration with MC SW library must be taken into account.

For example, in the STEVAL-IHM026V1 demonstration firmware, the number of acquisitions is 3 and the number of elementary measurements is 8, i.e. each single channel acquisition is made up of 24 charge cycles and 24 discharge cycles. Since those cycles are synchronized with the TIMER1 PWM (about 18 kHz), the acquisition of a single key should last about 2.66 ms. Moreover lost PWM cycles due to the Timer4 interrupt routine (*Figure 15*) - 1 PWM cycle lost every 1 ms and 10 PWM cycles lost every 10 ms - and to longer ACD ISR (*Figure 13*) and TIMER1 ISR (*Figure 14*) - 2 PWM cycles lost every 1.7 ms (worst case at 3000 rpm) - are to be put in the calculation. Thus the time to scan one single key (Ts) is approximately in the range of 2.8 ms to 3.4 ms.

Moreover an additional feature of the TS SW library has to be considered. A filtering mechanism has been put in place in order to avoid false detection due to noise similar to the debouncing mechanism used with mechanical keys and switches. In order for a key to be reported as detected, the signal must remain above the detection threshold for several consecutive sampling periods. The number of additional sampling periods is called the debounce filter (DF). A typical value for DF is 2, which means that 3 samples above the detection level are necessary to trigger detection.

This mechanism has to be considered in the computation of the maximum number of keys obtainable.

The maximum response time is approximately:

Equation 2

$$T\ response\ max = Ta(DF+1)$$

where Ta is the acquisition time for all the keys (Ta = Ts * number of keys), and DF is the debounce filter.

Roughly, in order to keep T response max below the 150 ms threshold, considering PWM Timer1 frequency = 18 kHz, DF = 2, and Ts = 3.4 ms (worst case at 18 kHz), the maximum number of keys is approximately equal to 15.

5 Demonstration board description

5.1 HW environment

The MCU components needed for touch-sensing implementation are listed below:

- one GPIO for each key (six in the present board)
- one GPIO for load pin (common for all keys)
- one GPIO to implement driven shield (optional, here implemented)
- one 16-bit timer (TIM3) for acquisition
- one standard 8-bit timer (TIM4) used as a time base for the post-processing (calibration, filtering, etc.).

On the motor control side, the resources needed are:

- one 16-bit advanced control timer able to provide 6 PWM outputs using three OC channels (TIM1)
- for sensorless drive one 16-bit general-purpose timer (TIM2) to set the reference for current regulation or limitation
- alternatively for sensored drive, the 16-bit general-purpose timer (TIM2) is used to manage the signals of the hall sensors
- one 8-bit general-purpose timer (TIM4) to manage the time base
- one 10-bit analog/digital converter (ADC) for the BEMF detection and for the remaining conversions (bus voltage, heatsink temperature, potentiometer and so on).

The board may be interfaced, by means of an MC connector (see *Section 3.2*), with a power board driving the motor.

The demonstration program allows the user to drive the Shinano motor in sensorless mode, changing a set of parameters.

The default configuration for the jumpers is indicated in the schematics in *Section 7* (see also *Table 1*).

5.2 Getting started

The user interface, consisting of an LCD display (2 x 15 characters), six keys (up, down, left, right, sel, start/stop), one potentiometer and one extra switch (SW2), has been developed for the user, allowing the real-time setting of parameters such as the operating mode and the visualization of certain drive variables.

After turn-on and at each reset of the board, a welcome message is displayed on the LCD. After a few seconds a help message is shown on the LCD. It is possible to scroll to the following help pages by touching the “RIGHT” key. The user may select the desired fields displayed by using the UP/DOWN and SEL buttons.

Setting jumper J6 in the non-default configuration (*Table 1*), the buzzer is allowed to beep when a key press has been detected.

5.3 Operating modes

Two operating modes are available: the demonstration mode and the standard or user-guided mode. The two options are selectable by means of slider SW2.

If SW2 is set in the DEMO position the motor runs so that its speed profile starts from a minimum speed up to a maximum speed in ramp time and back in the same time. A pause (wait time) is added before starting a new cycle.

Several control parameters may be set by the user (see UM0708 “STM8Sxxx three-phase BLDC motor control software library” user manual for details).

The following parameters regarding the demonstration operating mode may be chosen:

- Minimum speed expressed in rpm
- Maximum speed expressed in rpm
- Ramp rise and fall time expressed in seconds
- Wait time between ramps expressed in seconds.

In the standard mode the target speed expressed in rpm may also be set.

The parameters can also be changed while the motor is running. The target speed can also be changed by varying the position of potentiometer RV2.

6 CPU load calculation

For the calculation of the CPU load of the entire firmware (motor control and touch-sensing libraries), it has to be considered that the normal firmware flow, described in *Figure 12*, is sometimes interrupted by the TIM4 interrupt routine execution, as shown in *Figure 15*. Longer ADC and TIM1 interrupt service routines are not considered.

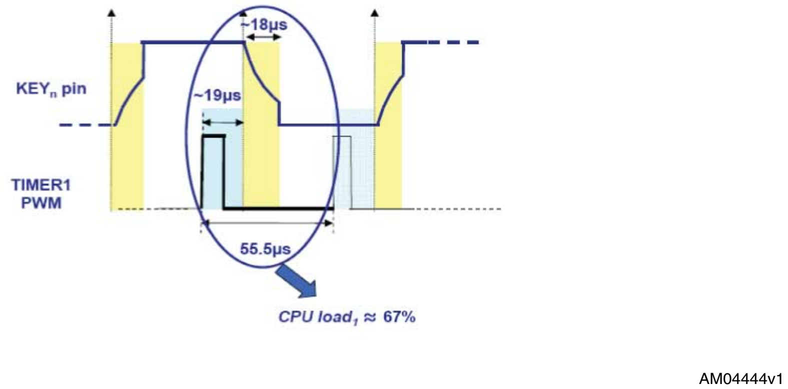
Considering the intervals during which the TIM4 interrupt routine is not executed (*Figure 16*), the CPU load is approximately equal to:

Equation 3

$$\mathrm{CPUload}_1 \approx (19\mu + 18\,\mu)/55.5\,\mu \approx 67\%$$

See *Figure 12* for the details of the motor control tasks.

Figure 16. CPU load $_1$: CPU load during one TIM1 PWM cycle



Every 1 ms the TIM4 interrupt routine, lasting t_{T4IRQ1} (approximately 56 µs) is executed, preventing the TS routines from being executed. Every 10 ms the Timer4 interrupt routine lasts t_{T4IRQ2} (about 560 µs), see *Figure 15* and *17*. During these intervals the CPU load is equal to 100%.

Considering a 10 ms time window, the time interval during which the TIM4 interrupt routine is executed is equal to:

Equation 4

$$t_{T4IRQ} = 9 \cdot t_{T4IRQ1} + t_{T4IRQ2} \approx 9 \cdot 54\mu s + 480\mu s \approx 966\mu s$$

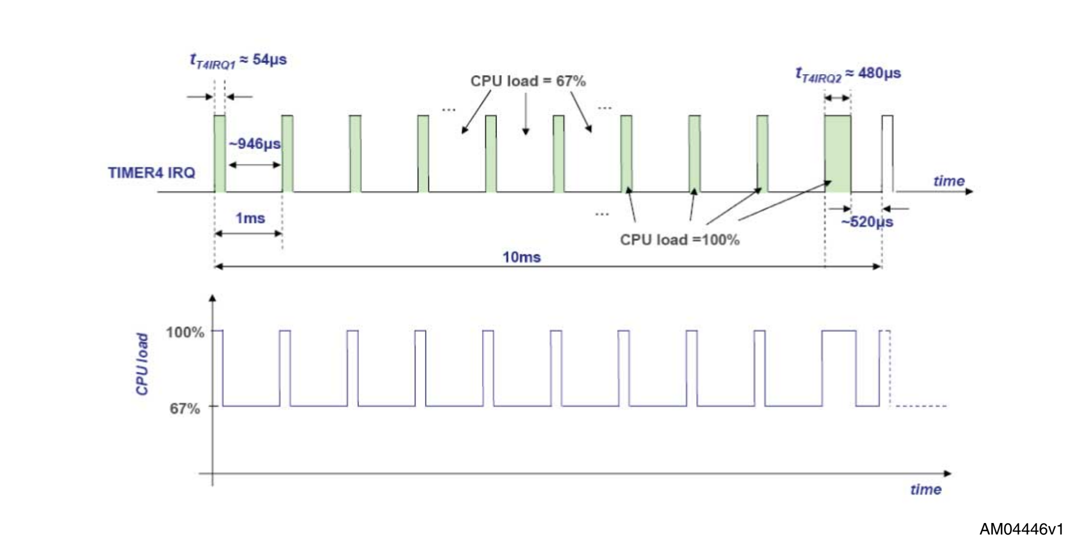
Thus, disregarding the intervals during which the longer ADC and TIM1 ISRs take place, in a 10 ms time slot, for 966 µs the CPU load is about 100%, while for the remaining 9034 µs it is about 67%, as shown in *Equation 5*.

Thus, the overall average CPU load can be computed as:

Equation 5

$$\text{CPU load} = \frac{t_{T4IRQ}}{10\text{ms}} \cdot 100\% + \frac{(10\text{ms} - t_{T4IRQ})}{10\text{ms}} \cdot 67\% \approx 70\%$$

Figure 17. CPU load calculation



7 Schematics

In this section the schematics of the STEVAL-IMH026V1 are shown in the following order:

- Peripherals, see *Figure 18*
- Microcontroller, see *Figure 19*
- Power supply, see *Figure 20*
- MC connector, see *Figure 21*
- Cap sensing input, see *Figure 22*

Figure 18. Schematic - peripherals

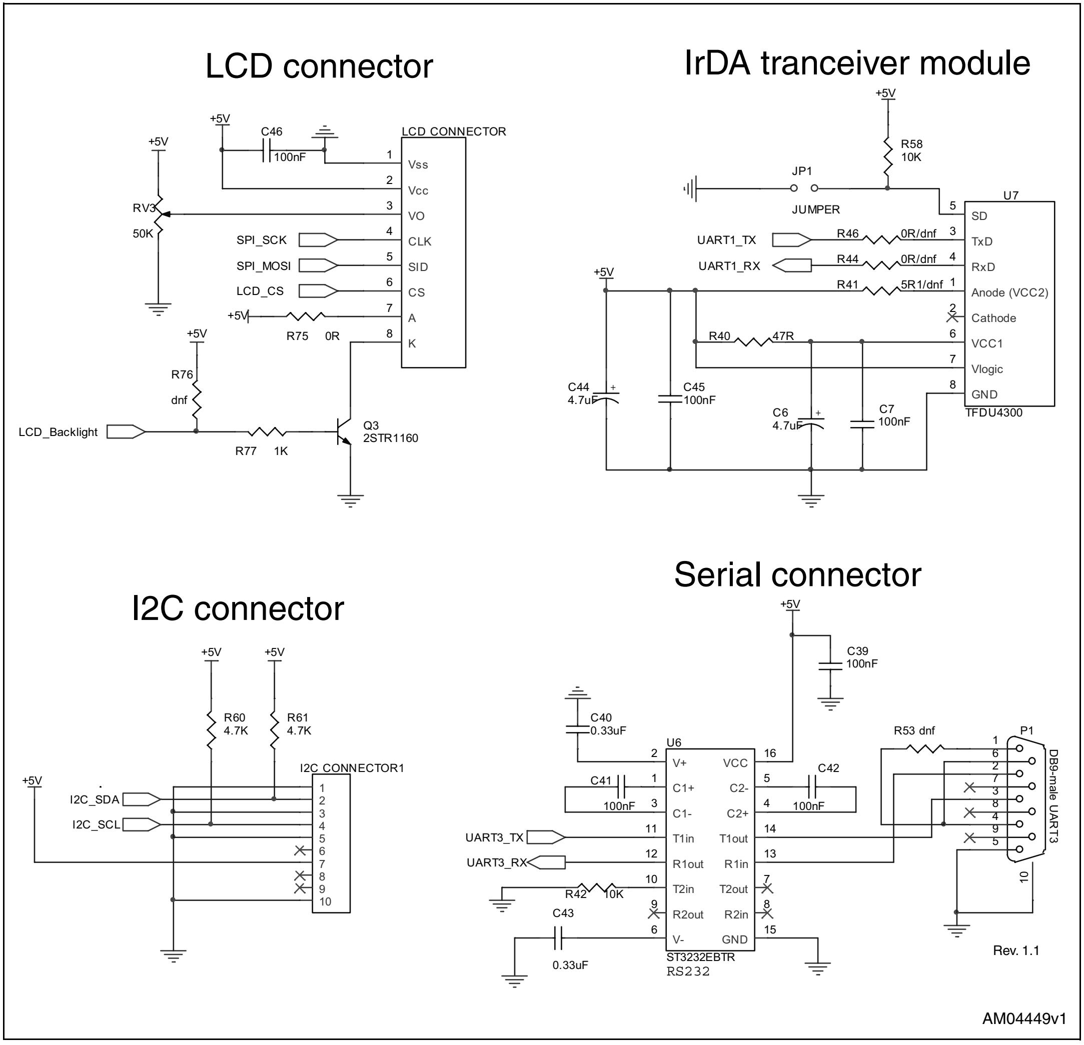


Figure 19. Schematic - microcontroller

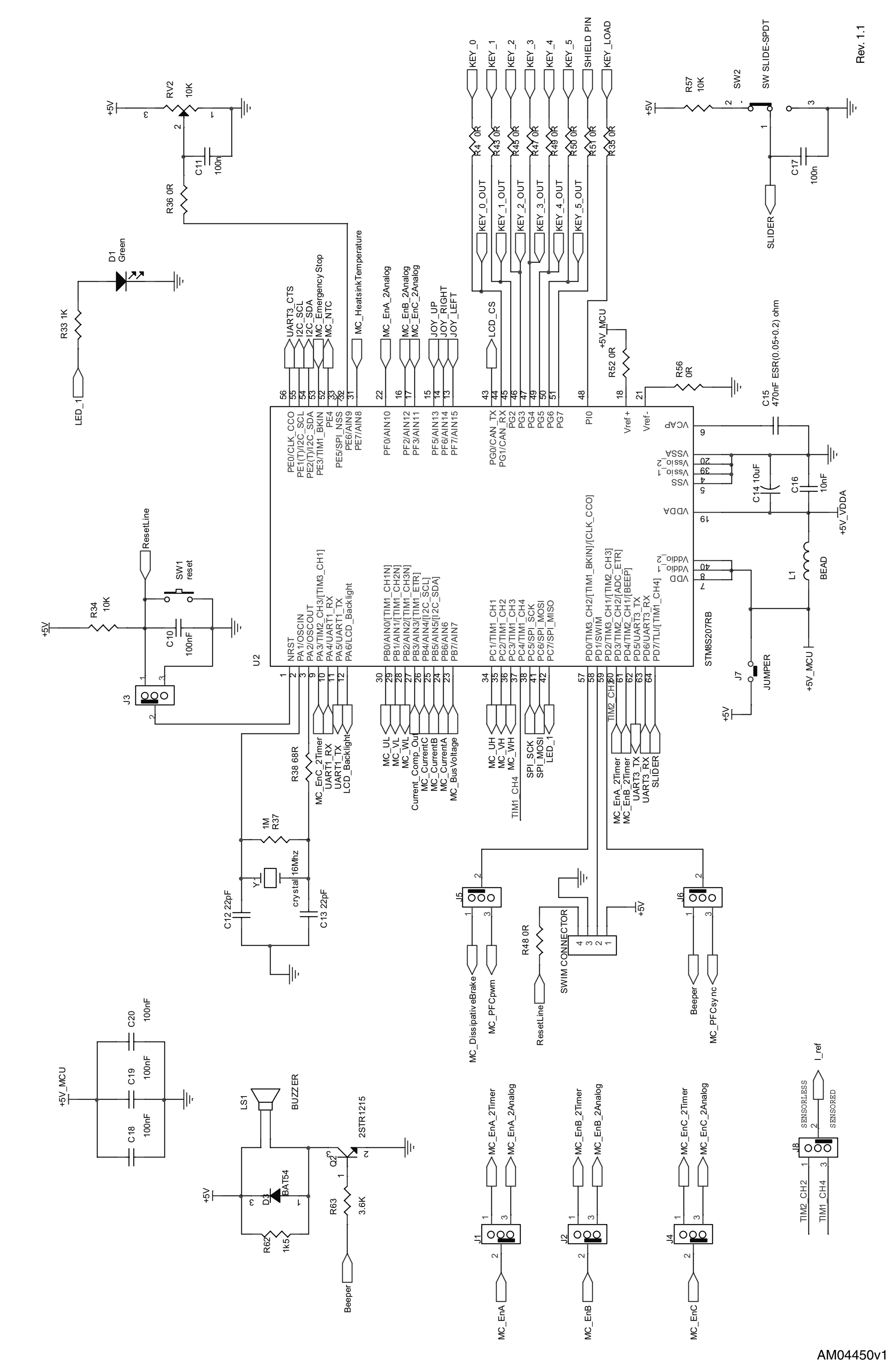


Figure 20. Schematic - power supply

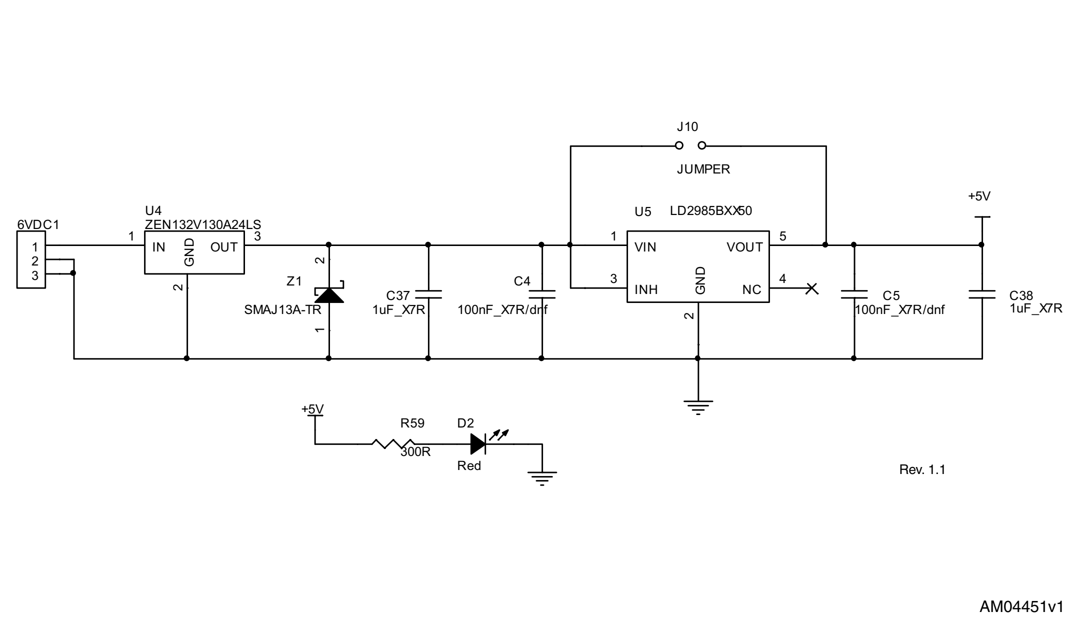


Figure 21. Schematic - MC connector

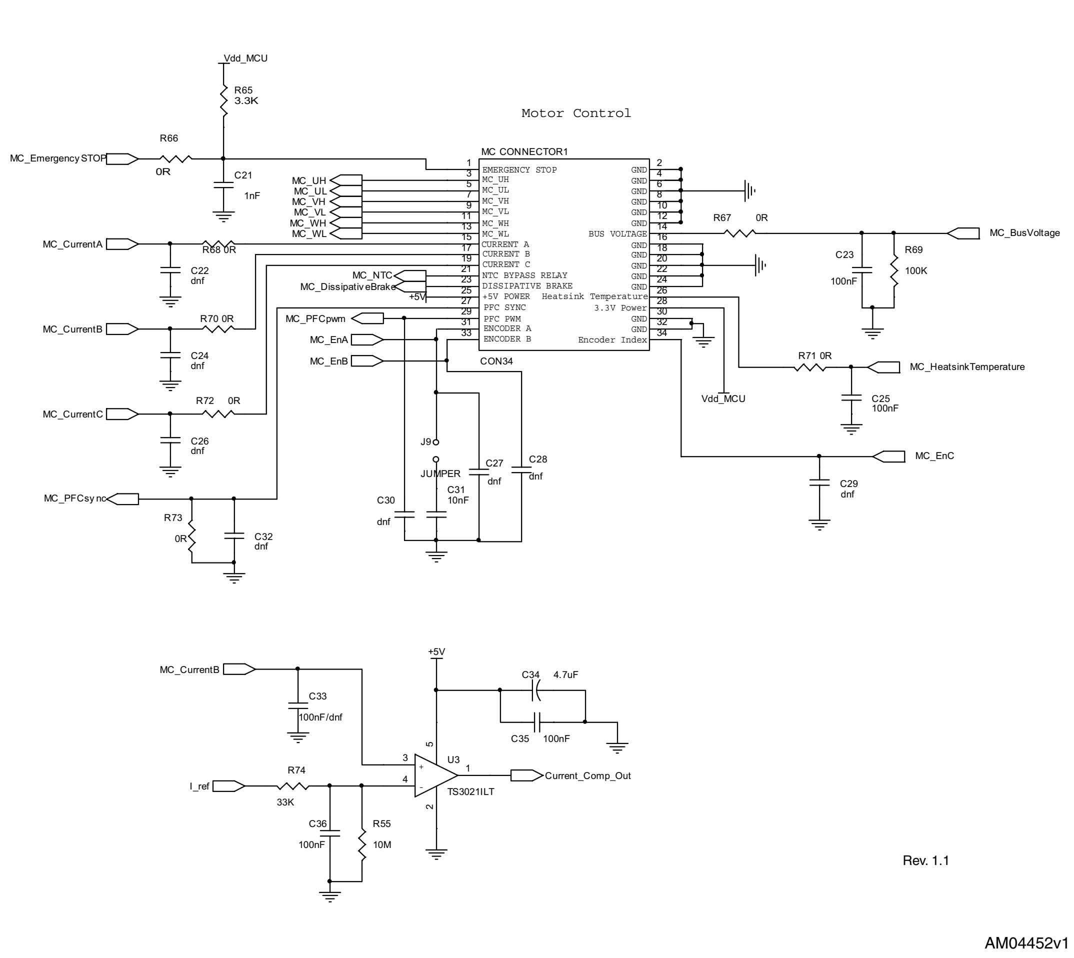
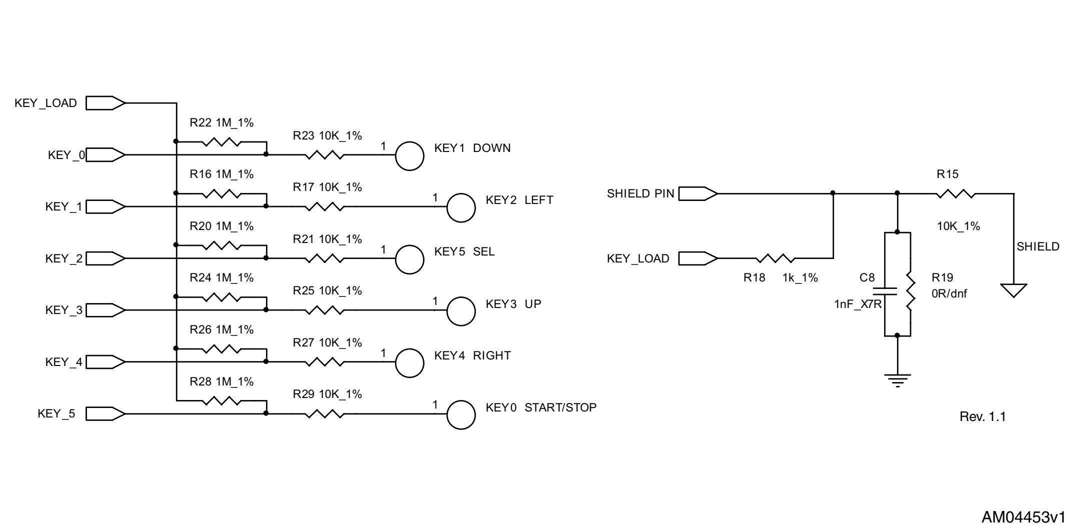


Figure 22. Schematic - cap sensing input



8 Reference documents

For further information please refer to the following documents:

1. UM0708 - STM8Sxxx three-phase BLDC motor control software library V1.0
2. STM8 touch sensing library firmware on-line help manual (stm8s_fwlib_um.chm available in TSL software package)
3. AN2927 - application note - RC acquisition principle for touch sensing applications
4. AN2869 - application note - guidelines for designing touch sensing applications

Appendix A Bill of material

Table 6. BOM

Item	Qty	Ref.	Part / value	Tolerance%	Voltage / current	Watt
1	1	C1	4.7 pF	+/- 10%	50 V	
2	17	C2,C7,C9,C10,C11,C17, C18,C19,C20,C23,C25, C35,C36,C39,C41,C42, C45	100 nF	+/- 10%	16 V	
3	2	C12,C13	22 pF	+/- 10%	50 V	
4	1	C15	470 nF ESR (0.05÷0.2) Ω	+/- 10%	16 V	
5	2	C8,C21	1 nF	+/- 10%	50 V	
6	1	C14	10 µF	+/- 20%	6,3 V	
7	2	C16,C31	10 nF	+/- 10%	50 V	
8	30	C4,C5,C22,C24,C26,C27, C28,C29,C30,C32,C33, R2,R3,R5,R6,R7,R8,R9, R10,R11,R12,R13,R14, R19,R31,R39,R44,R46, R53,R54	Not to be fitted			
9	4	C3,C6,C34,C44	4.7 µF	+/-10%	16 V	
10	2	C37,C38	1 µF	+/- 10%	16 V	
11	2	C40,C43	0.33 µF	+/- 10%	16 V	
12	1	D1	Green			
13	1	D2	Red			
14	1	D3	BAT54J			
15	1	I^{2}C connector1	CON10			
16	4	JP1,J7,J9,J10	JUMPER			
17	7	J1,J2,J3,J4,J5,J6,J8	CON3			
18	1	LCD	HXM122032-GB1			
19	1	LCD connector	CON8			
20	1	LS1	BUZZER			
21	1	L1	BEAD		200 mA	
22	1	MC connector1	CON34			
23	1	P1	DB9-male UART3			
24	1	Q1	2STR1160			
25	1	Q2	2STR1215			

Table 6. BOM (continued)

Item	Qty	Ref.	Part / value	Tolerance%	Voltage / current	Watt
26	1	RV1	50 kΩ			1/4 W
27	1	RV2	10 kΩ			50 mW
28	12	R1,R15,R17,R21,R23, R25,R27,R29,R34,R42, R57,R58	10 kΩ	+/- 1%		0.1 W
29	21	R4,R30,R35,R36,R43, R45,R47,R48,R49,R50, R51,R52,R56,R64,R66, R67,R68,R70,R71,R72, R73	0 Ω	+/- 1%		0.1 W
30	7	R16,R20,R22,R24,R26, R28,R37	1 MΩ	+/- 1%		0.1 W
31	3	R18,R32,R33	1 kΩ	+/- 1%		0.1 W
32	1	R38	68 Ω	+/- 1%		0.1 W
33	1	R40	47 Ω	+/- 1%		0.1 W
34	1	R41	5.1 Ω	+/- 1%		0.1 W
35	1	R55	10 MΩ	+/- 1%		0.1 W
36	1	R59	300 Ω	+/- 1%		0.1 W
37	2	R60,R61	4.7 kΩ	+/- 1%		0.1 W
38	1	R62	1.5 kΩ	+/- 1%		0.1 W
39	1	R63	3.6 kΩ	+/- 1%		0.1 W
40	1	R65	3.3 kΩ	+/- 1%		0.1 W
41	1	R69	100 kΩ	+/- 1%		0.1 W
42	1	R74	33 kΩ	+/- 1%		0.1 W
43	1	SWIM connector1	CON4			
44	1	SW1	Reset			
45	1	SW2	SW slide-SPDT			
47	1	U2	STM8S207RB			
48	1	U3	TS3021ILT			
49	1	U4	ZEN132V130A24LS		13.2 V	
50	1	U5	LD2985BXX50			
51	1	U6	ST3232EBTR			
52	1	U7	TFDU4300			
53	1	Y1	crystal 16 MHz			
54	1	Z1	SMAJ13A-TR			
55	1	6VDC1	CON2			

Revision history

Table 7. Document revision history

Date	Revision	Changes
27-Jan-2010	1	Initial release

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.

STMicroelectronics group of companies

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

www.st.com