



### Getting started with STM8TL53xx

---

#### Introduction

This application note complements the information in the STM8TL53xx datasheets by describing the minimum hardware and software environment required to build an application around an STM8TL53xx 8-bit microcontroller device.

A brief description of the principal hardware components is given. The power supply, reset control and ProxSense lines are described in some detail. In addition, some hardware recommendations are given. This application note also contains detailed reference design schematics with descriptions of the main components. The STM8 development tools and software toolchain are common to STM8TL53xx, STM8L, STM8S and STM8A and are presented in [Section 9](#), and [10](#). [Section 11](#) describes how to set up the STM8 development environment. Finally, [Section 12](#) provides a list of relevant documentation and online support resources.

# Contents

<b>1</b>	<b>Hardware requirements summary</b>	<b>6</b>
<b>2</b>	<b>Power supply</b>	<b>6</b>
2.1	Power supply overview	6
2.2	Main operating voltages	7
2.3	Power-on/power-down reset (POR/PDR)	8
<b>3</b>	<b>Clock management</b>	<b>9</b>
3.1	Internal clocks	9
<b>4</b>	<b>Reset control</b>	<b>10</b>
4.1	Reset management overview	10
4.1.1	Output characteristics	10
4.1.2	Input characteristics	11
4.2	Hardware reset implementation	12
<b>5</b>	<b>ProxSense line management</b>	<b>13</b>
5.1	ProxSense line management overview	13
5.2	Hardware ProxSense implementation	14
<b>6</b>	<b>Recommendations</b>	<b>15</b>
6.1	Printed circuit board	15
6.2	Component position	15
6.3	Ground and power supply ( $V_{SS}$ , $V_{DD}$ , $V_{SSIO}$ , $V_{DDIO}$ )	15
6.4	Decoupling	15
6.5	Other signals	16
6.6	Unused I/Os and features	16
<b>7</b>	<b>Reference design</b>	<b>17</b>
7.1	Component references	17
7.2	Schematics	18
<b>8</b>	<b>STM8TL5x firmware libraries</b>	<b>19</b>

8.1	STM8TL5x standard peripheral library .....	19
8.2	STM8TL5x touch sensing library .....	19
8.3	Online help .....	20
<b>9</b>	<b>STM8 development tools .....</b>	<b>21</b>
9.1	Single wire interface module (SWIM) .....	21
9.1.1	SWIM overview .....	21
9.1.2	SWIM connector pins .....	22
9.1.3	Hardware connection .....	22
9.2	RLink and STLink .....	22
<b>10</b>	<b>STM8 software toolchain .....</b>	<b>23</b>
10.1	Integrated development environment .....	24
10.2	Compiler .....	24
<b>11</b>	<b>Setting up the STM8 development environment .....</b>	<b>25</b>
11.1	Installing the tools .....	25
11.2	Using the tools .....	26
11.2.1	Project editing .....	27
11.3	Running the demonstration software .....	28
11.3.1	Compiling the project .....	28
11.3.2	Selecting the correct debug instrument .....	29
11.3.3	Connecting the hardware .....	30
11.3.4	Starting the debug session .....	31
11.3.5	Running the software .....	32
11.3.6	Follow up .....	32
<b>12</b>	<b>Documentation and online support .....</b>	<b>33</b>
<b>13</b>	<b>Revision history .....</b>	<b>34</b>

List of tables

Table 1. Component list . . . . . 17

Table 2. SWIM connector pins . . . . . 22

Table 3. Document revision history . . . . . 34



## List of figures

Figure 1.	Power supply . . . . .	7
Figure 2.	Typical layout of $V_{DD}/V_{SS}$ pair . . . . .	8
Figure 3.	Reset management . . . . .	10
Figure 4.	Output characteristics . . . . .	11
Figure 5.	Input characteristics . . . . .	11
Figure 6.	ProxSense management . . . . .	13
Figure 7.	Touch key layout example . . . . .	14
Figure 8.	Reference design . . . . .	18
Figure 9.	STM8TL5x standard peripheral firmware library online help manual . . . . .	20
Figure 10.	STM8TL5x touch sensing firmware library online help manual . . . . .	20
Figure 11.	Debug system block diagram . . . . .	21
Figure 12.	Hardware connection . . . . .	22
Figure 13.	STM8 software toolchain . . . . .	23
Figure 14.	STVD open example workspace . . . . .	26
Figure 15.	STVD MCU edit mode . . . . .	27
Figure 16.	STVD: Building the project . . . . .	28
Figure 17.	STVD: Selecting the debug instrument. . . . .	29
Figure 18.	STVD: Starting the debug session . . . . .	31
Figure 19.	STVD: Run the software . . . . .	32

# 1 Hardware requirements summary

To build an application around an STM8TL53xx device, the application board should provide the following features:

- Power supply (mandatory)
- Reset management (optional)
- ProxSense line management (optional)
- Debugging tool support: Single wire interface module (SWIM) connector (optional)

## 2 Power supply

### 2.1 Power supply overview

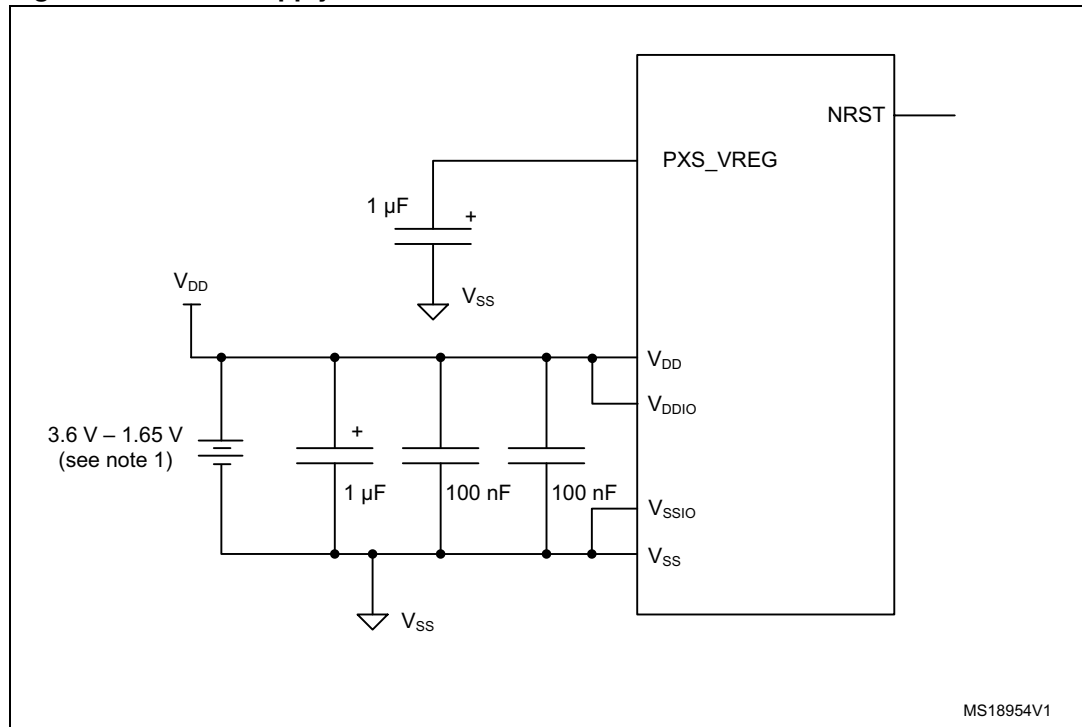
The STM8TL53xx needs to be powered by a 1.65 V to 3.6 V external source.

An on-chip power management system provides the constant digital supply to the core logic, both in normal and low power modes. This ensures that the logic consumes a constant current level over the voltage range. It is also capable of detecting voltage drops and generate a reset to avoid erratic behaviour.

The STM8TL53xx device provides:

- One pair of power supply pins ( $V_{DD}/V_{SS}$ ) for the main operating voltage (1.65 V to 3.6 V).
- Another pair of power supply pins (depending of package) ( $V_{DDIO}/V_{SSIO}$ ) for the IOs (1.65 V to 3.6 V)

The STM8TL53xx device manages the supply voltage needed by the ProxSense interface by connecting a 1  $\mu$ F capacitor to the PXS\_VREG pin (see [Figure 1](#)).

**Figure 1. Power supply**

1. The device keeps operating as long as the battery voltage is above 1.65 V and no reset is generated.

**Note:**

*The capacitors must be connected as close as possible to the device power supply pins ( $V_{DDx}$  pins).*

*The decoupling capacitor must be connected as close as possible to the ground pins ( $V_{SSx}$  pins).*

## 2.2 Main operating voltages

The STM8TL53xx devices embed an internal voltage regulator for generating the 1.8 V power supply for the core and peripherals and a second internal voltage regulator providing a stable power supply (around 1.45 V) for the ProxSense peripheral.

## 2.3 Power-on/power-down reset (POR/PDR)

The input supply to the main and low power regulators is monitored by a power-on/power-down reset circuit. The monitoring voltage begins at 0.7 V.

During power-on, the POR/PDR keeps the device under reset until the supply voltage ( $V_{DD}$ ) reach its specified working area. This internal reset is maintained for a period of ~1ms in order to wait for supply stabilization.

At power-on, a defined reset should be maintained below 0.7 V. The upper threshold for a reset release is defined in the electrical characteristics section of the product datasheets.

A hysteresis is implemented (POR > PDR) to ensure clean detection of voltage rise and fall.

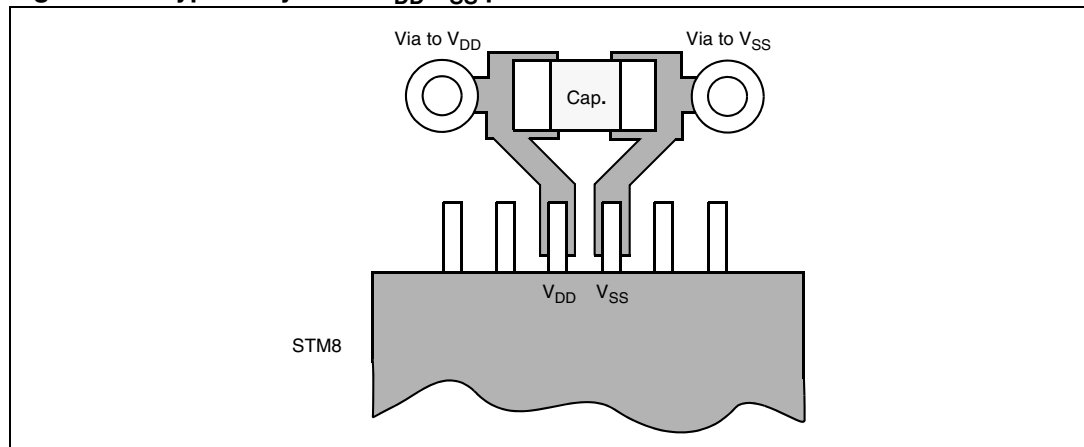
The POR/PDR also generates a reset when the supply voltage drops below the  $V_{POR/PDR}$  threshold (isolated and repetitive events).

### Recommendations

All  $V_{DD}$  and  $V_{SS}$  pins including  $V_{DDIO}$  and  $V_{SSIO}$  need to be properly connected to the power supplies. These connections, including pads, tracks and vias should have the lowest possible impedance. This is typically achieved with thick track widths and preferably dedicated power supply planes in multi-layer printed circuit boards (PCBs).

In addition, the power supply pair should be decoupled with filtering ceramic capacitors (C) at 100 nF with one chemical C (1..2  $\mu$ F) in parallel on the STM8TL53xx device. The ceramic capacitors should be placed as close as possible to the appropriate pins, or below the appropriate pins, on the opposite side of the PCB. Typical values are 10 nF to 100 nF, but exact values depend on the application needs. [Figure 2](#) shows the typical layout of such a  $V_{DD}/V_{SS}$  pair.

**Figure 2. Typical layout of  $V_{DD}/V_{SS}$  pair**



## 3 Clock management

The STM8TL53xx has no external clock so no precautionary measures are needed.

### 3.1 Internal clocks

STM8TL53xx devices have three kinds of internal clock: A high speed internal clock (HSI) running at 16 MHz, a low speed internal clock (LSI) running at 38 kHz and a high speed internal clock dedicated to the ProxSense (HSI\_PXS) running at 16 MHz. The HSI\_PXS clock runs once the ProxSense is enabled if the LowPower bit is reset. If LowPower bit is set HSI\_PXS clock runs only when an acquisition is being performed.

After reset, the CPU starts at speed of 2 MHz driven by the internal RC (HSI clock signal) divided by 8.

## 4 Reset control

### 4.1 Reset management overview

The reset pin is a 3.3 V bidirectional I/O (supplied by  $V_{DDIO}$ ). After startup it can be programmed by software to be used as a general purpose output.

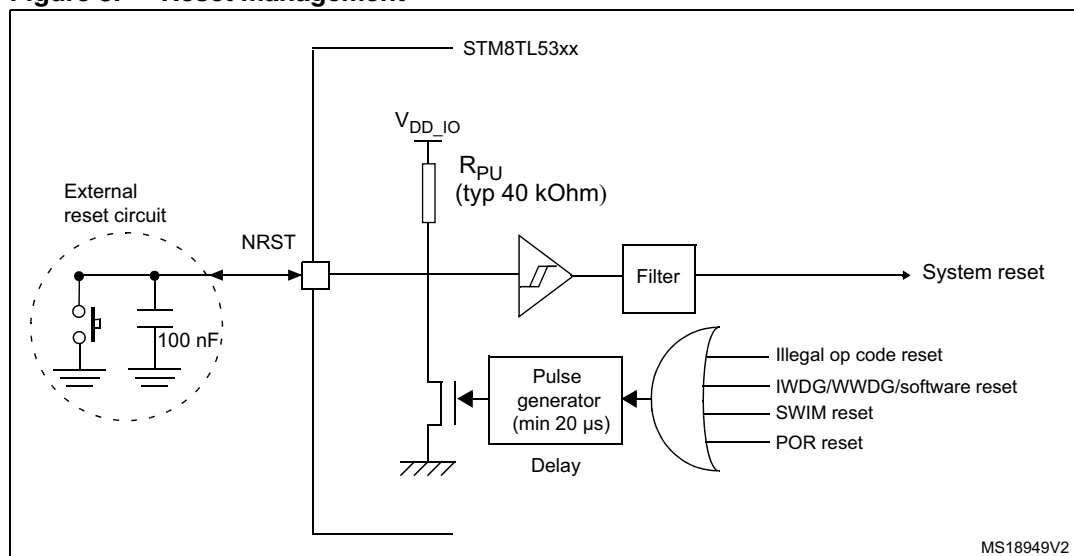
Its output buffer driving capability is fixed to  $I_{OL\_MIN} = 2 \text{ mA}$  @ 0.45 V in the 1.65 V to 3.6 V range which includes a  $\sim 40 \text{ k}$  pull-up. Output buffer is reduced to the n-channel MOSFET (NMOS). The receiver includes a glitch filter, whereas the output buffer includes a 20  $\mu\text{s}$  delay.

There are many reset sources, including:

- External reset through the NRST pin
- Power-on reset (POR): During power-on, the POR keeps the device under reset until the supply voltage ( $V_{DD}$ ) reach the right voltage level.
- Independent watchdog reset (IWDG)
- Window watchdog reset (WWDG), featuring also software reset.
- SWIM reset: An external device connected to the SWIM interface can request the SWIM block to generate a microcontroller reset.
- Illegal opcode reset: If a code to be executed does not correspond to any opcode or prebyte value, a reset is generated.

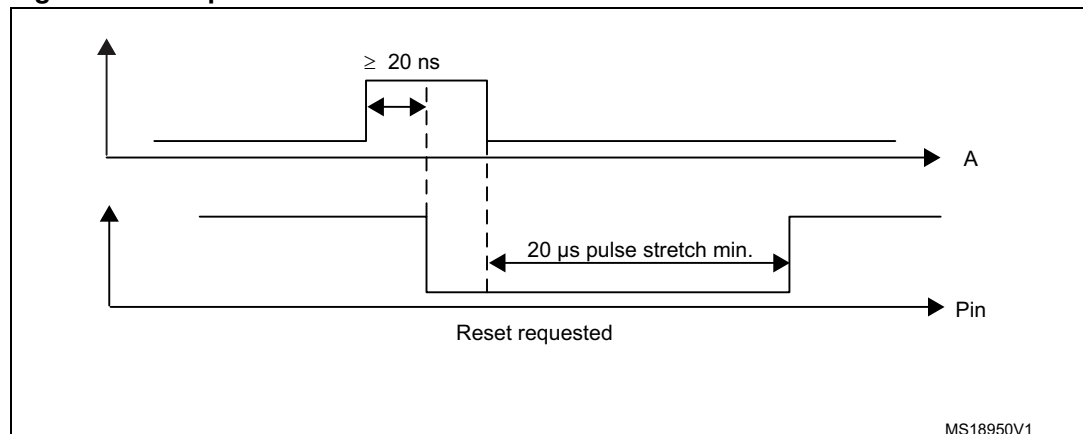
Figure 3 shows a simplified functional I/O reset schematic.

Figure 3. Reset management



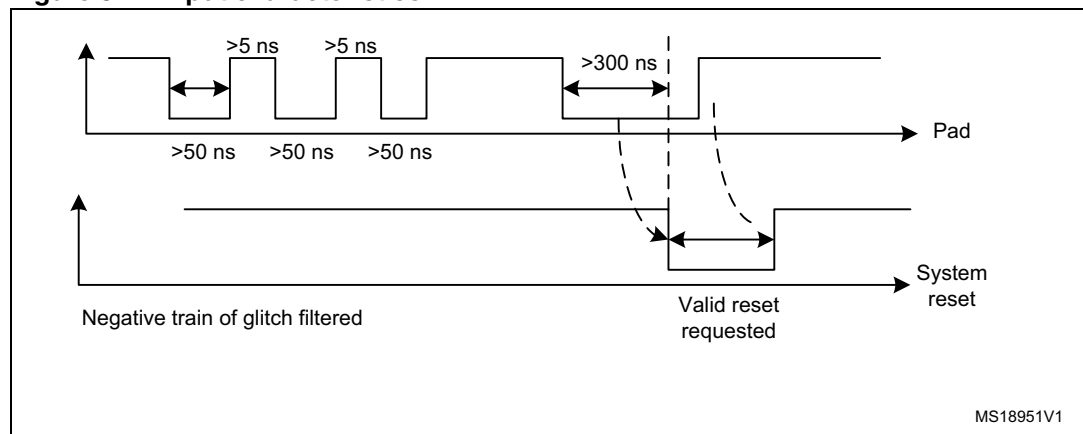
#### 4.1.1 Output characteristics

- A valid pulse on the pin is guaranteed with a  $\geq 20 \text{ ns}$  pulse duration on the internal output buffer.
- After a valid pulse is recognized, a pulse on the pin of at least 20  $\mu\text{s}$  is guaranteed starting from the falling edge of A (output of the OR between the different reset sources).

**Figure 4. Output characteristics**

#### 4.1.2 Input characteristics

- All pulses with a duration less than 50 ns are filtered
- All train/burst spikes with a ratio of 1/10 must be filtered. This means that a negative spike of up to 50 ns is always filtered, when a 5 ns interval between spikes occurs (ratio 1/10).
- All pulses with duration more than 300 ns are recognized as valid pulses

**Figure 5. Input characteristics**

## 4.2 Hardware reset implementation

The STM8TL53xx does not require an external reset circuit to power-up correctly. Only a pull-down capacitor is recommended (see [Figure 3](#)). However, charging/discharging the pull-down capacitor through an internal resistor has a negative influence on the device power consumption. Therefore, the recommended capacitor value of 100 nF can be reduced down to 10 nF to limit such power consumption.

The STM8TL53xx reset state is released 1 ms after the POR value (1.35 V to 1.65 V) is reached. At this time,  $V_{DD}$  should be in the 1.65 V to 3.6 V range.

## 5 ProxSense line management

### 5.1 ProxSense line management overview

Since the STM8TL53xx provides a ProxSense interface, it is mainly used to perform capacitance variation acquisition.

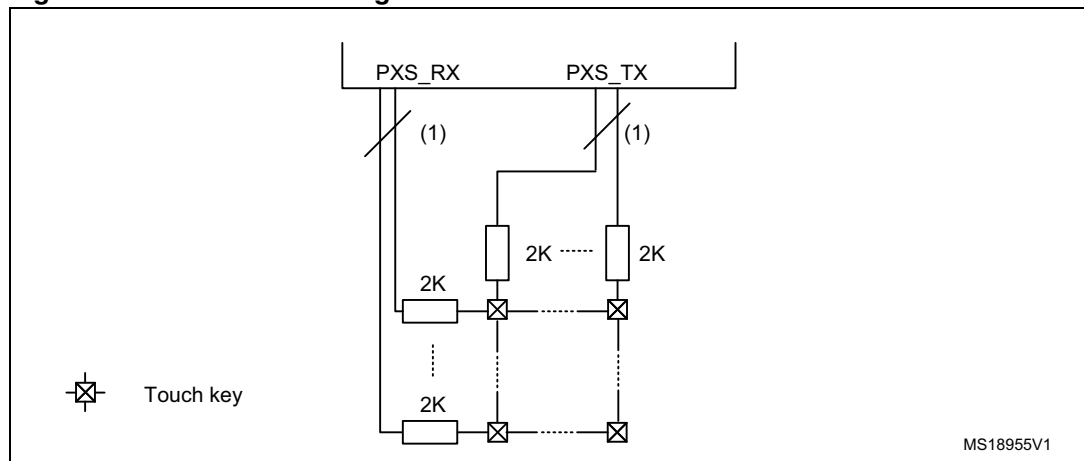
The principle of this interface is to transfer a charge from a capacitance (the electrode  $C_X$ ) to another one (the sample  $C_S$ , inside the MCU) by driving the PXS\_TX and activating some switches. This allows detection of proximity as well as touch by monitoring on PXS\_RX.

The many features of the ProxSense interface include:

- 10 independent receiver channels, allowing 10 measurements to be performed in parallel
- Each of the 10 receiver channels can be associated with two different pins, effectively allowing an application to have up to 20 receiver channels.
- Each receiver channel can be independently configured to perform projected capacitance measurements.
- The size of each  $C_S$  capacitor can be independently configured with 5 bits of resolution.

*Figure 6* shows a simplified functional schematic of the ProxSense interface.

**Figure 6. ProxSense management**



1. The receiver and transmit numbers are application dependent.

## 5.2 Hardware ProxSense implementation

The STM8TL53xx does not require any external circuitry to transfer the charge correctly. However, 2K serial resistors are recommended (see [Figure 6](#)) for ESD robustness. Several configurations are possible such as,

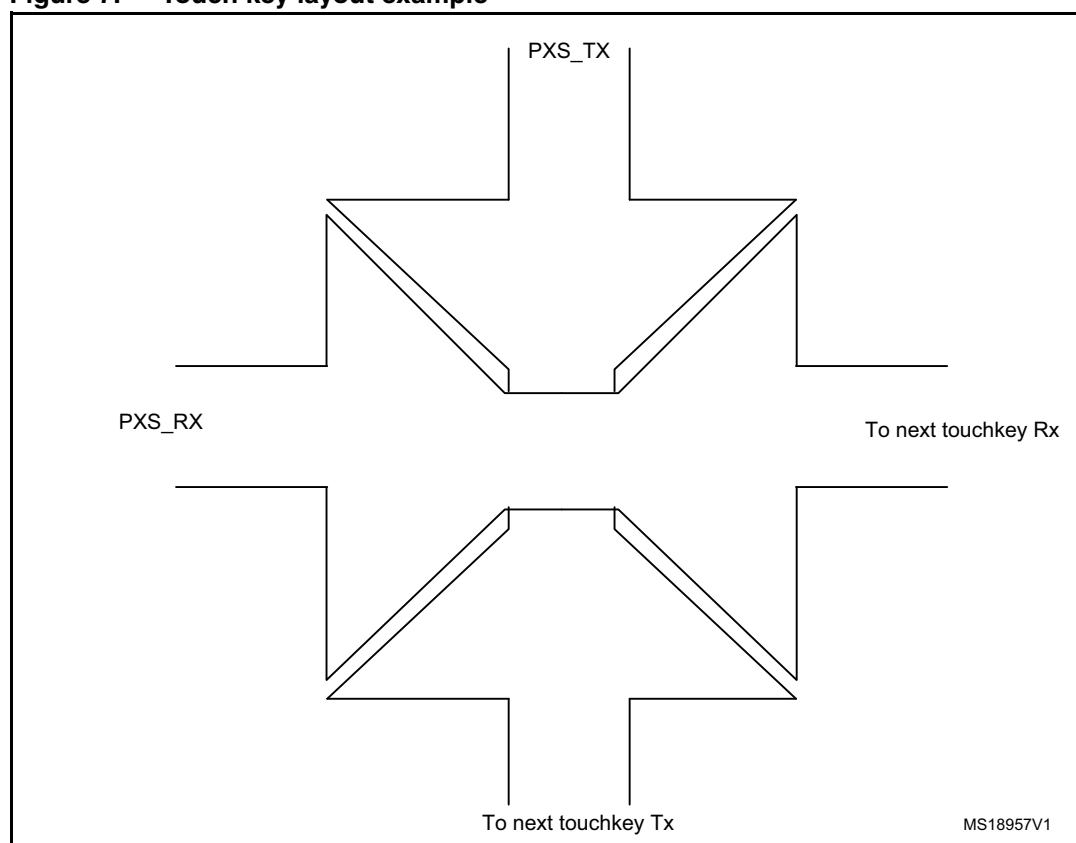
- 10 PXS\_RX with 1 PXS\_TX (10 measurements in parallel) or
- 10 PXS\_TX with 1 PXS\_RX (10 measurements in serial).

However, any combination of 20 RX and 15 TX is allowed. Up to 15 PXS\_TX are allowed to be used. Furthermore the 20 RX are organized in 2 groups, all RX belonging to the same group being acquired simultaneously.

Advanced features such as an antenna (to detect noise) and external trigger may be implemented via the PXS\_RFIN and PXS\_TRIG pins in order to increase the robustness of the application.

The footprint shown in [Figure 7](#) gives an example of a touch key implementation on a PCB.

**Figure 7. Touch key layout example**



## 6 Recommendations

### 6.1 Printed circuit board

For technical reasons, it is best to use a multi-layer PCB with a separate layer dedicated to the  $V_{SS}$  and another layer to the  $V_{DD}$  supply. This results in a good decoupling, as well as a good shielding effect. For many applications, economic requirements prohibit the use of this type of board. In this case, the most important requirement is to ensure a good structure for the  $V_{SS}$  and power supply.

### 6.2 Component position

A preliminary layout of the PCB must separate the different circuits according to their electromagnetic interference (EMI) contribution. This reduces cross-coupling on the PCB, for instance, noisy, high-current circuits, low voltage circuits, and digital components.

### 6.3 Ground and power supply ( $V_{SS}$ , $V_{DD}$ , $V_{SSIO}$ , $V_{DDIO}$ )

The  $V_{SS}$  should be distributed individually to every block (noisy, low level sensitive, and digital) with a single point for gathering all ground returns. Loops must be avoided or have a minimum surface. The power supply should be implemented close to the ground line to minimize the surface of the supply loop. This is due to the fact that the supply loop acts as an antenna, and is therefore the main emitter and receiver of EMI. All component-free surfaces of the PCB must be filled with additional grounding to create a kind of shield (especially when using single-layer PCBs).

### 6.4 Decoupling

The standard decoupler for the external power is a 1  $\mu\text{F}$  capacitor. Supplementary 100 nF capacitors must be placed as close as possible to the  $V_{SS}/V_{DD}$  and  $V_{SSIO}/V_{DDIO}$  pins of the microcontroller to reduce the area of the current loop.

As a general rule, decoupling all sensitive or noisy signals improves electromagnetic compatibility (EMC) performances.

There are two types of decouplers:

- Capacitors close to components. Inductive characteristics, which apply to all capacitors beyond a certain frequency, must be taken into account. If possible, parallel capacitors with decreasing values (0.1, 0.01,...  $\mu\text{F}$ ) should be used.
- Inductors. Although often ignored, ferrite beads, for example, are excellent inductors due to their good dissipation of EMI energy and there is no loss of DC voltage (which is not the case when simple resistors are used).

## 6.5 Other signals

When designing an application, the following areas should be closely studied to improve EMC performances:

- Noisy signals (clock)
- Sensitive signals (high impedance)
- Signals for which a temporary disturbance affects operation of the application permanently, for example, interrupts and handshaking strobe signals (but not LED commands).

A surrounding  $V_{SS}$  trace for such signals increases EMC performances, as does a shorter length or absence of noisy and sensitive traces (crosstalk effect).

For digital signals, the best possible electrical margin must be reached for the two logical states. Slow Schmitt triggers are recommended for eliminating parasitic states.

## 6.6 Unused I/Os and features

Microcontrollers are designed for a variety of applications, where often a particular application does not use 100 % of the microcontroller resources.

To avoid unnecessary power consumption (especially important for battery powered applications) and also to improve EMC performance, unused clocks, counters, or I/Os, should not be left free, I/Os should be forced externally (pull-up or pull-down to the unused I/O pins), and unused functions should be 'frozen' or disabled.

Alternatively, unused I/Os can be programmed as push-pull 'low' to keep them at a defined level without using external components. However in this case, the I/O is not driven during the power up phase, until the I/O is configured. This can add a little extra power consumption, and may be undesirable in very power sensitive applications.

The unused PXS\_RX pins should follow the same rule and can be driven to  $V_{SS}$  by resetting the PXS\_RXINSR register.

## 7 Reference design

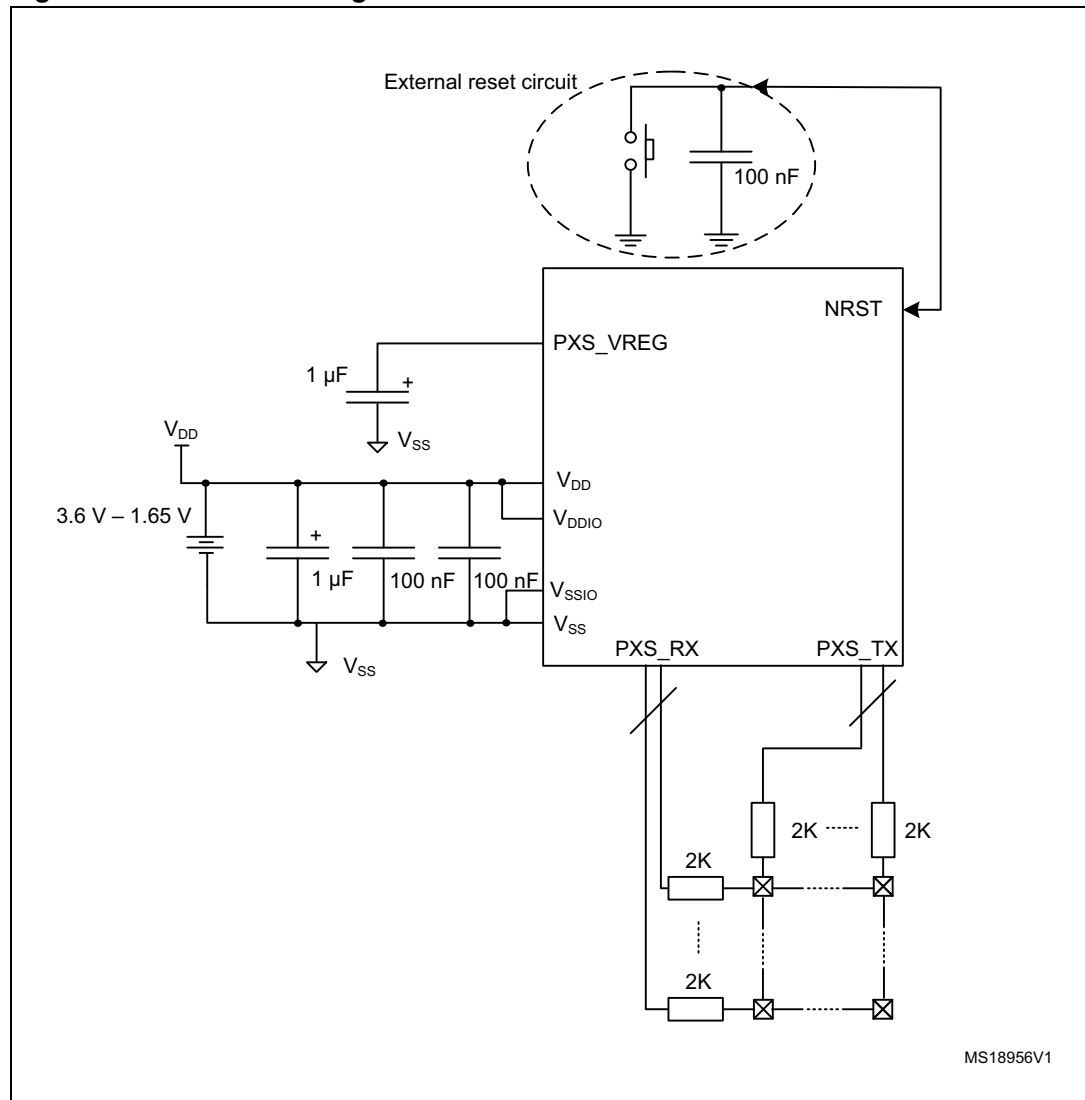
### 7.1 Component references

Table 1. Component list

ID	Component name	Reference	Quantity	Comments
1	Microcontroller	STM8TL53xx	1	Refer to the 'pinouts and pin description' and 'package characteristics' sections of the STM8TL53xx datasheets, to choose the right package
2	Battery	1.65 V to 3.6 V	1	
3	Capacitor	1 $\mu$ F	2	Decoupling capacitor
4	Capacitor	100 nF	2	Ceramic capacitor (decoupling capacitor)
Components below are optional				
5	SWIM connector	4 pins	1	
6	Resistor	2K	n	Serial resistors for PXS_TX and PXS_RX pins

## 7.2 Schematics

Figure 8. Reference design



## 8 STM8TL5x firmware libraries

In order to ease the development start-up, two firmware libraries are provided:

- The STM8TL5x standard peripheral library
- The STM8TL5x touch sensing library

### 8.1 STM8TL5x standard peripheral library

This STM8TL5x firmware library contains the standard peripheral drivers (timers, I2C, SPI, USART, watchdogs, etc...) and a complete set of source code examples for each STM8TL5x peripheral. It is written in strict ANSI-C and it is fully MISRA C 2004 compliant.

All examples can be used with four workspace and project definition files:

- One for the STVD and Cosmic C compiler
- One for the STVD and Raisonance Compiler
- One for the Raisonance integrated debugging environment and compiler (RIDE7 IDE)
- One for the IAR embedded workbench for STM8 (EWSTM8).

This enables the user to load and compile them easily into their preferred development environment.

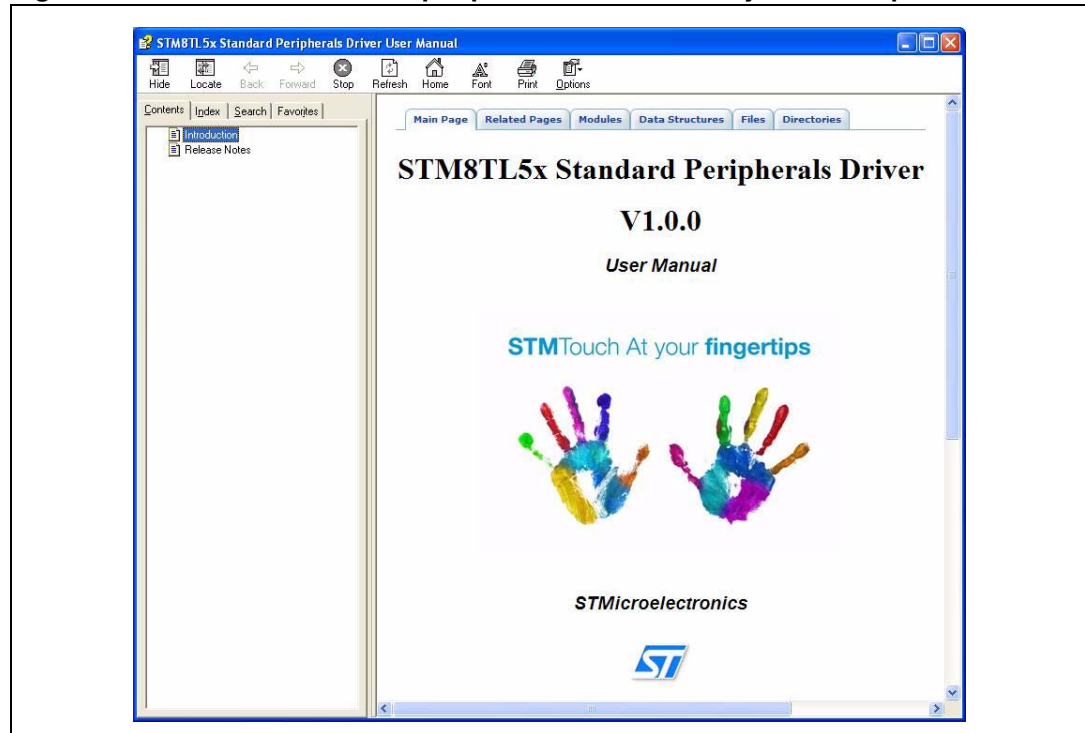
### 8.2 STM8TL5x touch sensing library

The STM8TL5x touch sensing library is dedicated to the management of the ProxSense (PXS) peripheral. It follows the same coding rules as the standard peripheral library. The STM8TL5x touch sensing library allows you to enable touch sensing capabilities on STM8TL5x devices. This simple firmware offers a complete and robust solution to manage capacitive sensing keys, wheels or sliders.

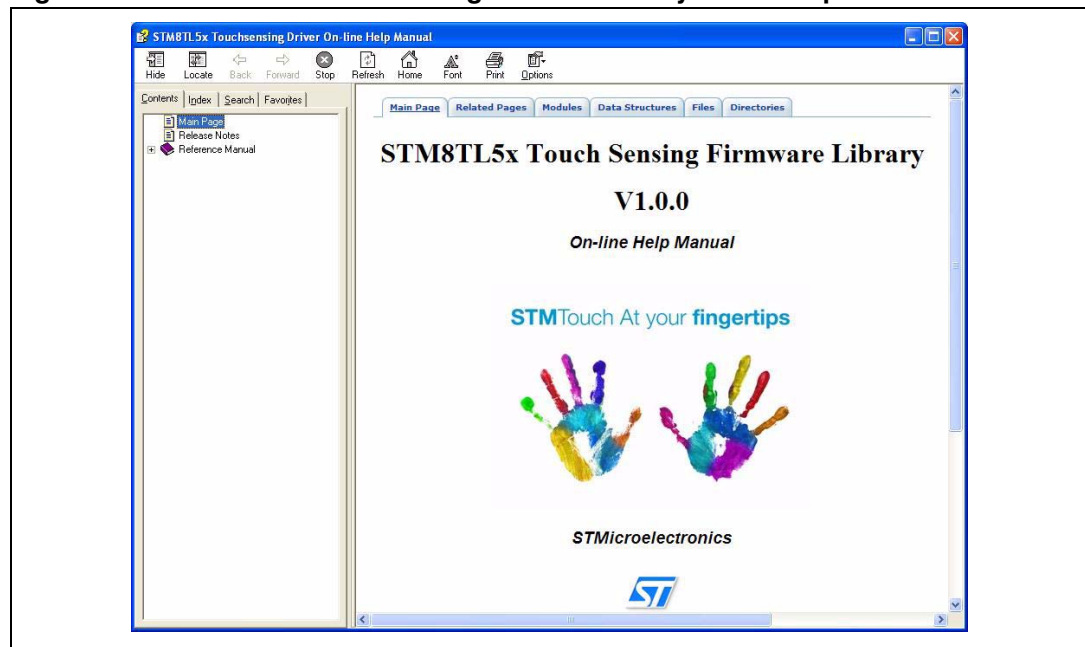
## 8.3 Online help

For each firmware library, an online help is directly available from the firmware installation directory (see [Figure 9](#) and [Figure 10](#)).

**Figure 9. STM8TL5x standard peripheral firmware library online help manual**



**Figure 10. STM8TL5x touch sensing firmware library online help manual**



## 9 STM8 development tools

Typically, the following tools are needed to get started:

- STVD, IAR worbench, or RIDE for integrated development environment
- STM8 C compiler (from Cosmic, Raisonance, or IAR)
- ST toolset from STMicroelectronics
- Firmware libraries from STMicroelectronics (STM8TL5x\_StdPeriph\_Lib and STM8TL5x\_TouchSensing\_Lib for STM8TL53).
- STM8T850 controller board from STMicroelectronics
- The debug interface ST-LINK is included in STM8T850 controller board
- If you do not use STM8T850 controller board, you may need the hardware debug interface "Rlink" from Raisonance
- STMSudio is a graphical user interface that allows sampling and viewing user variables in real time using any hardware debugging tool while the application is running.

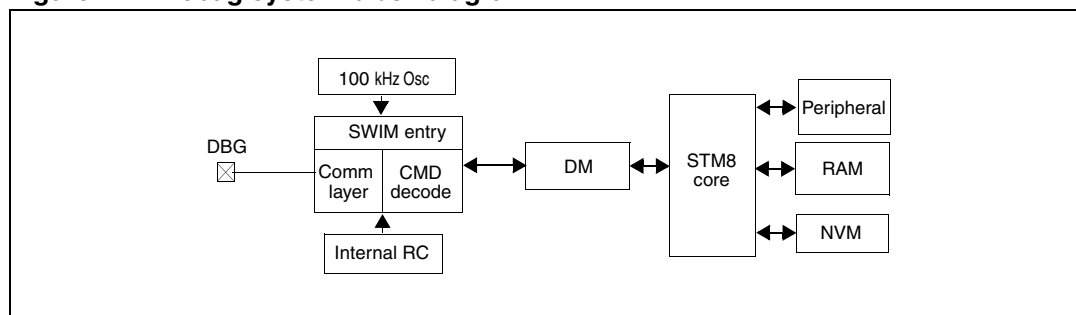
### 9.1 Single wire interface module (SWIM)

#### 9.1.1 SWIM overview

In-circuit debugging mode or in-circuit programming mode are managed through a single wire hardware interface based on an open-drain line, featuring ultra fast memory programming. Coupled with an in-circuit debugging module, the SWIM also offers a non-intrusive read/write to RAM and peripherals. This makes the in-circuit debugger extremely powerful and close in performance to a full-featured emulator.

The SWIM pin can be used as a standard I/O (with 8 mA capability) which has some restrictions if the user wants to use it for debugging. The most secure way to use it is to provide a strap option on the PCB. Please refer to the STM8 SWIM communication protocol and debug module user manual (UM0470) for more SWIM protocol details.

**Figure 11. Debug system block diagram**



### 9.1.2 SWIM connector pins

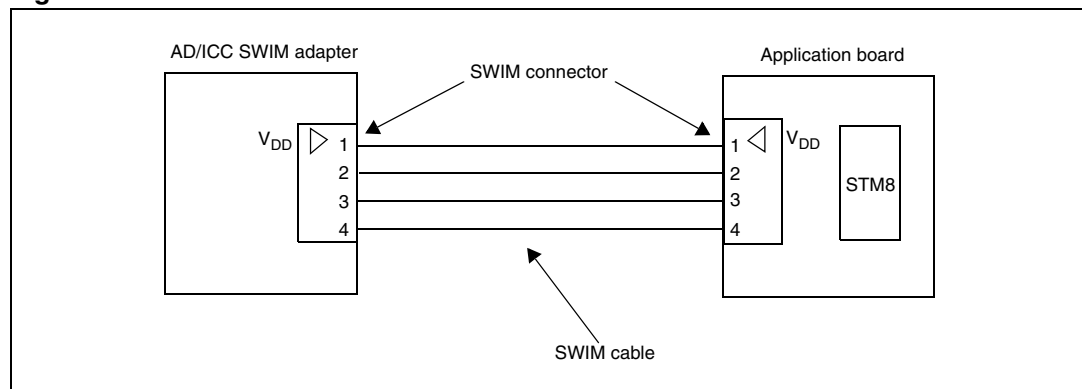
The SWIM connector pins consist of four pins as described in [Table 2](#).

**Table 2. SWIM connector pins**

Pin number	Pin name
Pin 1	V <sub>DD</sub>
Pin 2	SWIM pin
Pin 3	V <sub>SS</sub>
Pin 4	Reset

### 9.1.3 Hardware connection

**Figure 12. Hardware connection**



**Caution:** It is recommended to place the SWIM header as close as possible to the STM8TL53xx device, as this minimizes any possible signal degradation caused by long PCB tracks.

## 9.2 RLink and STLink

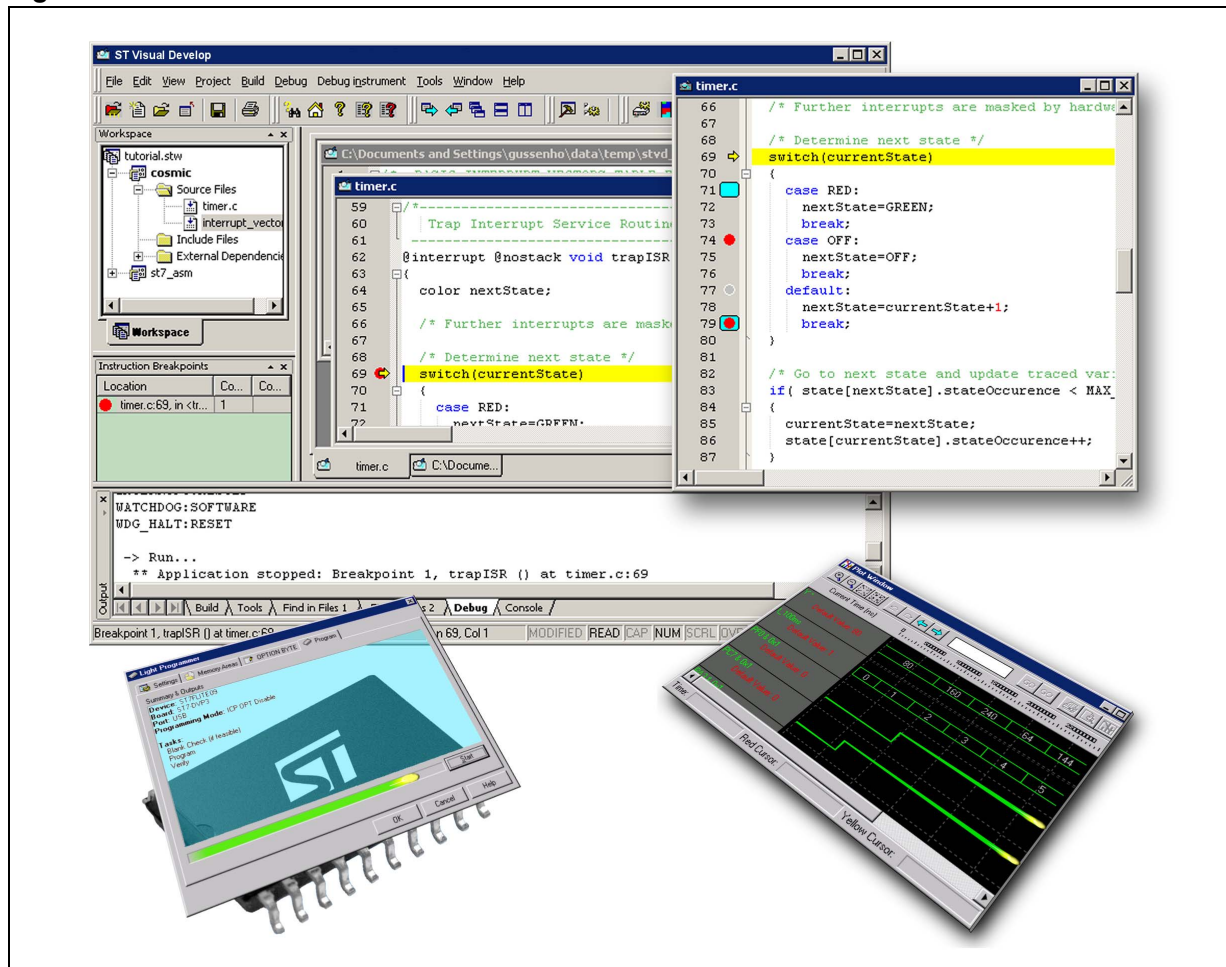
RLink and STLink are debug tools that allow any user application board with the SWIM interface to be connected to a host PC via USB for debugging and programming. See [Section 11.3.3: Connecting the hardware on page 30](#).

## 10 STM8 software toolchain

To write, compile and run the first software on an STM8TL53xx device, the following components of the software toolchain are required (see [Figure 13](#)):

- Integrated development environment
- Compiler
- Firmware library (optional, used to ease the startup)

**Figure 13. STM8 software toolchain**



## 10.1 Integrated development environment

The integrated development environment ST visual develop (STVD) provides an easy-to-use, efficient environment for start-to-finish control of application development, from building and debugging the application code to programming the microcontroller. STVD is delivered as part of the free ST toolset, which also includes the ST visual programmer (STVP) programming interface and the ST assembler linker.

To build applications, STVD provides seamless integration of C and assembly tool chains for ST including the Cosmic and Raisonance C compilers and the ST assembler linker. When debugging, STVD provides an integrated simulator (software) and supports a complete range of hardware tools including the low-cost RLink in-circuit debugger/programmer and the high-end STice emulator.

To program applications to an STM8TL53xx, the STVD also provides an interface for reading from the microcontroller memories, writing to them and verifying them. This interface is based on the ST visual programmer (STVP), and supports all the target devices and programming tools supported by STVP.

The free ST toolset for STM8 is available from STMicroelectronics homepage (see [www.st.com](http://www.st.com)).

## 10.2 Compiler

STM8TL53xx device can be programmed by a free assembler toolchain which is included in the ST toolset.

As the core is designed for optimized high-level-language support, use of a C compiler is recommended!

C compilers for STM8 are offered by the third party companies Cosmic, Raisonance, and IAR.

A free version of the C compiler with up to 32 Kbytes of generated code is available at: [www.cosmic-software.com](http://www.cosmic-software.com) and [www.raisonance.com](http://www.raisonance.com).

# 11 Setting up the STM8 development environment

## 11.1 Installing the tools

All software tools are delivered with a setup wizard which guides the user through the installation process. It is recommended to install the tools in the following order:

1. C compiler
2. ST toolset
3. STM8TL5x firmware libraries

ST-LINK does not need any dedicated software installation in the STM8 development environment because the necessary drivers are delivered with the ST toolset.

The R-link drivers must be launched separately as follows:

*Start/Programs/STtoolset/Setup/Install Rlink driver.*

## 11.2 Using the tools

Once the tools installation is complete, the ST visual develop (STVD) integrated development environment can be launched.

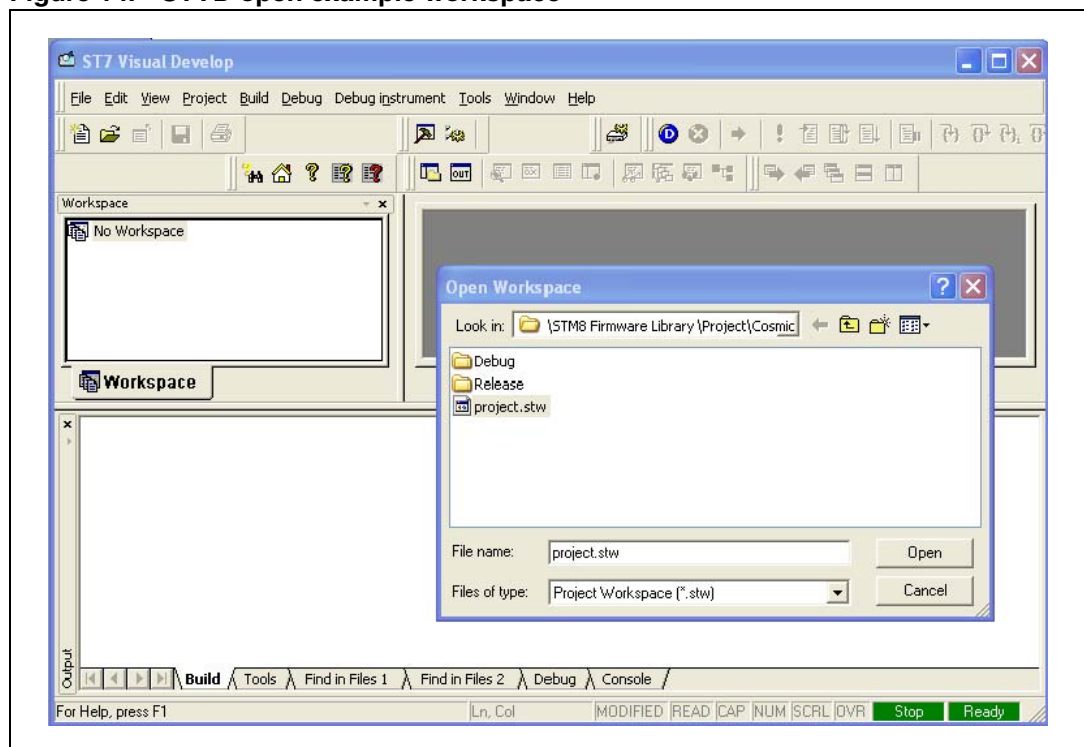
The user then has the choice to generate either a new workspace with a new project or to open an existing workspace. If using the STVD for the first time, it is recommended to open an existing project from one of the STM8TL5x firmware libraries.

**Note:** *Even if you are not intending to use the library, an existing library project can be used as a template to configure all the compiler options. Enter your own code after main().*

The STM8TL5x standard peripheral firmware library includes several examples for each peripheral plus one workspace containing a blank project which is ready to receive your C code. It is located in the firmware subdirectory \Project\Template (see [Figure 14](#)). You can choose between STVD\Cosmic, STVD\Raisonance, RIDE, or EWSTM8.

The STM8TL5x Touch Sensing library provides all functions required for an easy and quick development of your own touch sensing application, using the full set of STM8TL5x touch sensing features. The STM8TL5x Touch Sensing library contains the touch sensing drivers dedicated to the STM8TL5x and its ProxSense peripheral but also a set of examples showing the STM8TL5x performances.

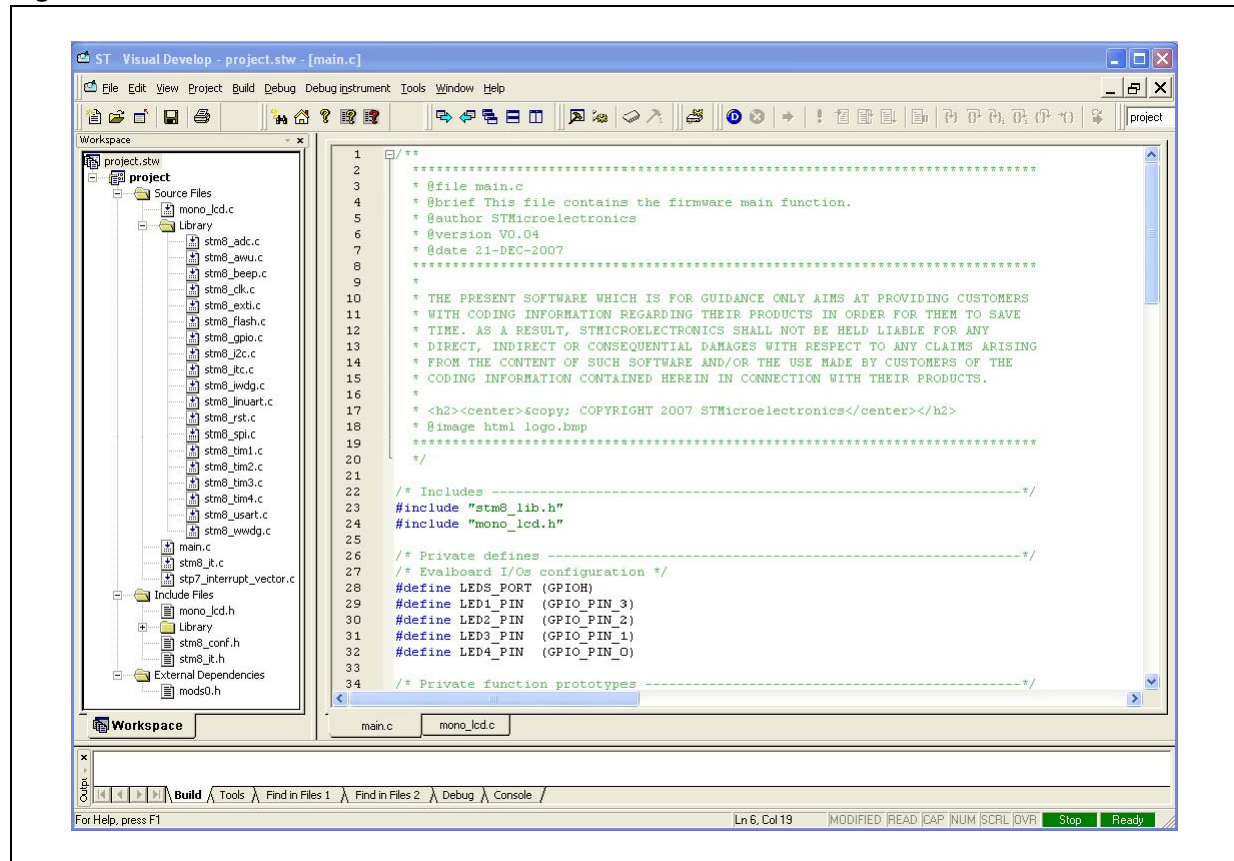
**Figure 14. STVD open example workspace**



## 11.2.1 Project editing

All project source files are visible and can be edited (see [Figure 15](#)).

**Figure 15. STVD MCU edit mode**



## 11.3 Running the demonstration software

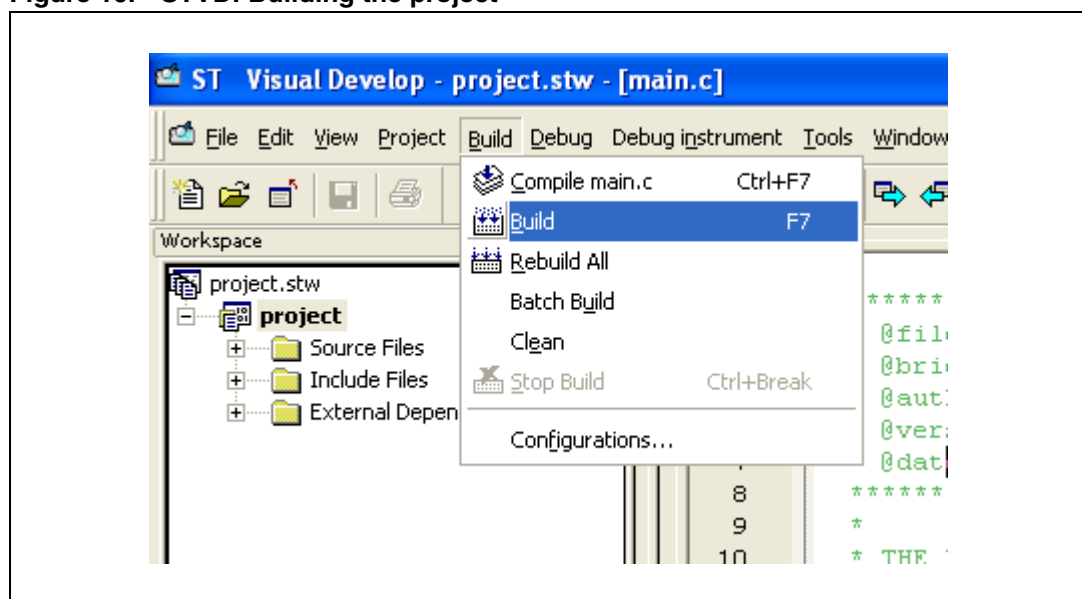
- Choose STM8TL53-EVAL firmware
- Open the desired project workspace within the chosen demonstration firmware package.

To run the demonstration software on the STM8T850 controller board, the project has to be compiled and the correct HW tool must be selected before the debug session can be started.

### 11.3.1 Compiling the project

The project can be compiled using the 'Build' function in the 'Build' menu (see [Figure 16](#)).

**Figure 16. STVD: Building the project**

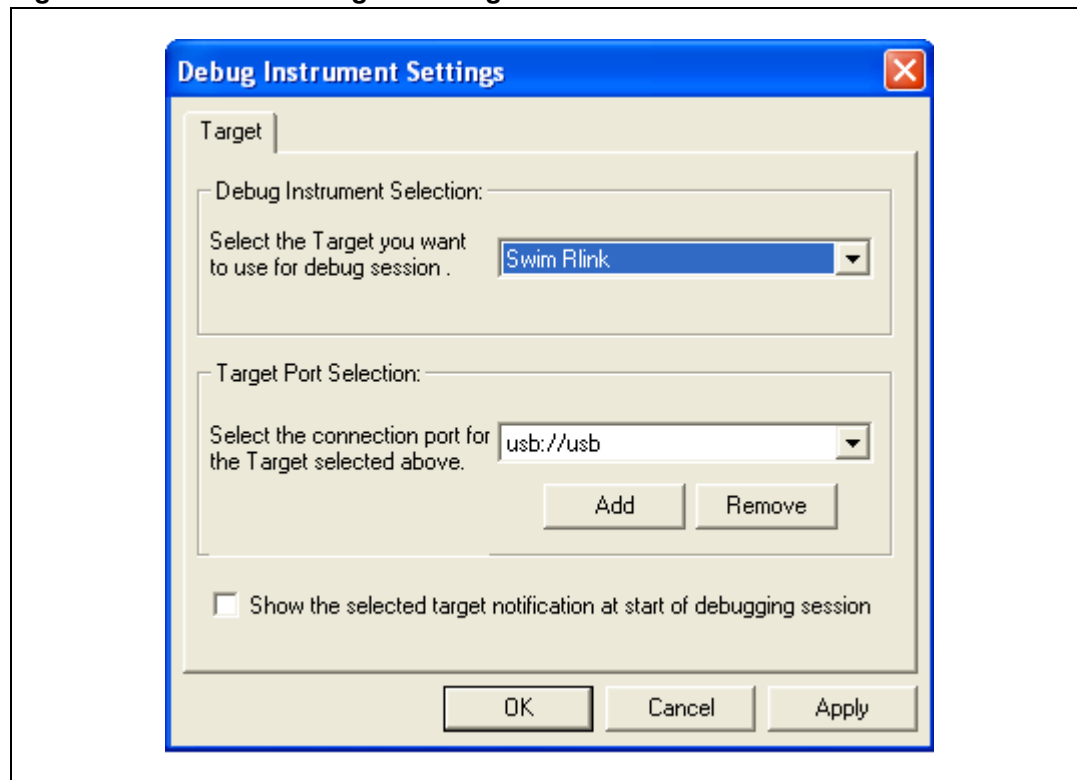


### 11.3.2 Selecting the correct debug instrument

In the example below, the Rlink tool is used for communicating via the SWIM interface with the on-board debug module of the STM8.

The Rlink tool can be selected from the 'Debug Instrument Selection' list in the 'Debug Instrument Settings' dialog (see [Figure 17](#)).

**Figure 17. STVD: Selecting the debug instrument**



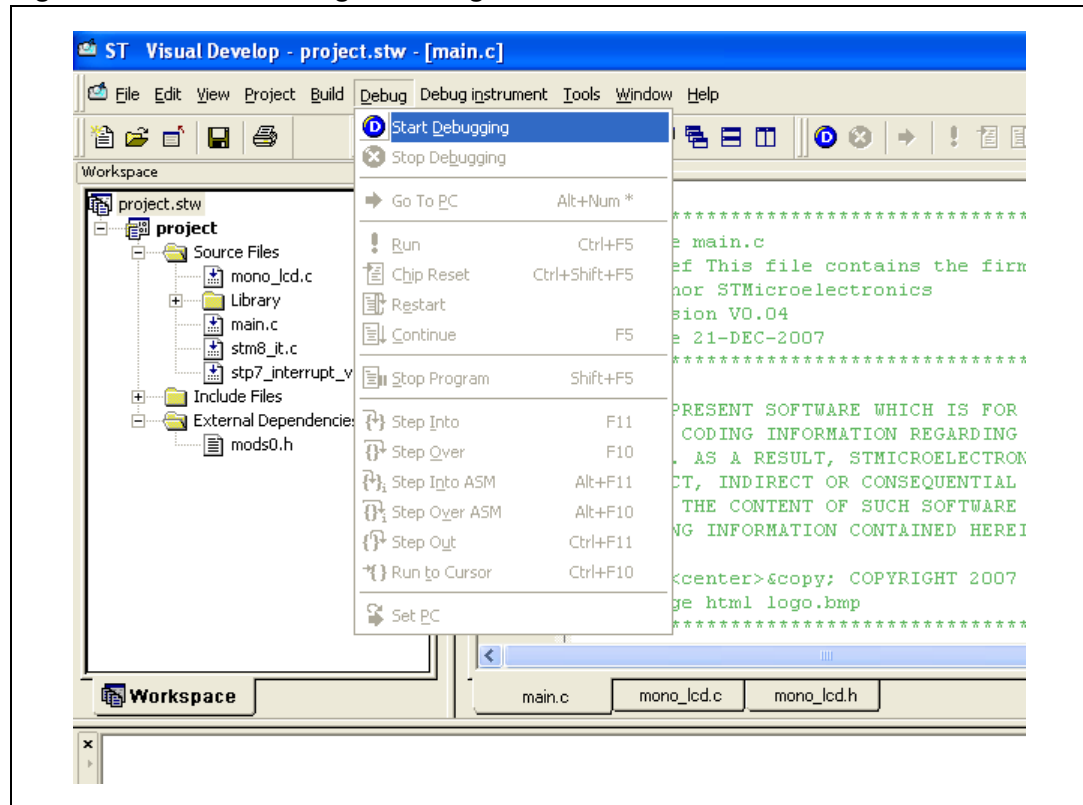
### 11.3.3 Connecting the hardware

The debug tool, STLink, is included on the STM8T850 controller board. You can connect the PC to the USB connector. This connection ensures the debug connection and the power. If the jumpers on the boards are no longer in the default position, please read the evaluation board user manuals to select power and debug support jumpers.

### 11.3.4 Starting the debug session

Debug mode can be entered by the command 'Debug Start Debugging' (see [Figure 18](#)).

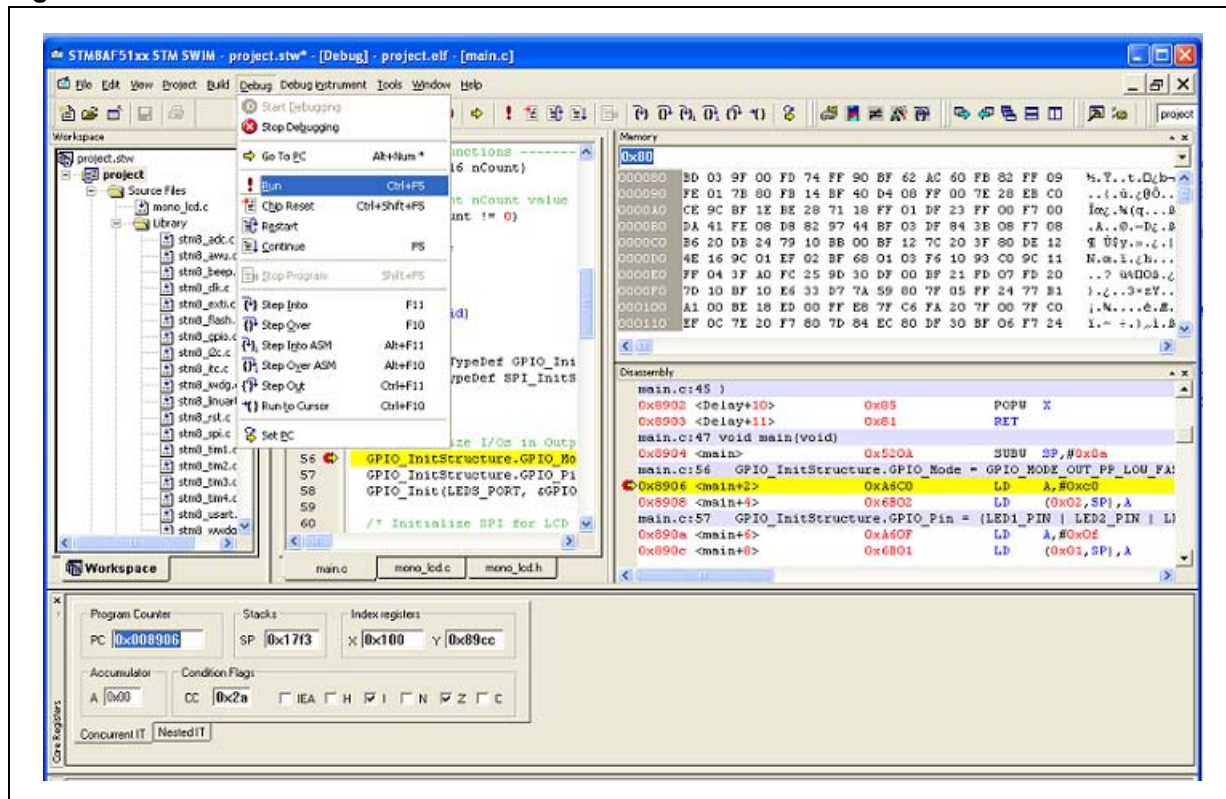
**Figure 18. STVD: Starting the debug session**



### 11.3.5 Running the software

After entering debug mode, the software can be started by the run command in the menu 'Debug Run' (see [Figure 19](#)).

Figure 19. STVD: Run the software



### 11.3.6 Follow up

Step by step, additional peripherals of STM8TL53xx devices can be run, following on from the initial debug session described above.

Many features of STM8TL53xx devices are supported by dedicated hardware on the STM8T850 controller board. The necessary software drivers, including STM8TL5x peripheral drivers (USART, I2C, SPI) and driver for the touch sensing modules (buttons, wheel, slider), are delivered in the STM8L85x firmware standard peripheral library and the STM8TL5x touch sense library.

## 12 Documentation and online support

Documentation resources related to tool usage includes:

### Application

- STM8TL53xx datasheet.
- How to program Flash memory and data EEPROM on STM8TL53xx microcontrollers (PM0212).
- STM8TL53xx reference manual (RM0312)
- STM8 CPU programming manual (PM0044)
- AN2869 Guidelines for designing touch sensing applications

### Tools

- STM8TL5x firmware standard peripheral library and release note (detailed descriptions of the library are included as help files).
- STM8TL5x firmware touch sensing library and release note (detailed descriptions of the library are included as help files).
- Cosmic, Raisonance, or IAR C compiler user manual
- STM8T850 controller board user manual (UM1081).
- STM8TL53-EVAL firmware
- ST Visual Develop tutorial (included as help files in the ST-toolchain)
- ST Visual Develop (STVD) user manual
- STM8 SWIM communication protocol and debug module user manual (UM0470)

The microcontroller discussion forum on [www.st.com](http://www.st.com) can be used by developers to exchange ideas. It is the best place to find different application ideas. In addition, the website has a knowledge base of FAQs for microcontrollers, which provide answers to many queries and solutions to many problems.

## 13 Revision history

**Table 3. Document revision history**

Date	Revision	Changes
10-Mar-2011	1	Initial release
06-Dec-2011	2	STM8TL53xx product name update Changed references to associated documents Updated <a href="#">Section 8: STM8TL5x firmware libraries on page 19</a> Updated <a href="#">Section 11.2: Using the tools on page 26</a> .

**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 TWO AUTHORIZED ST REPRESENTATIVES, 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.

© 2011 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 - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

[www.st.com](http://www.st.com)

