

UM0671 User manual

STM8/128-EV/TS STM8S touch sensing evaluation board

Introduction

The STM8S touch sensing evaluation kit (STM8/128-EV/TS) provides a platform that introduces users to STMicroelectronics capacitive touch sensing firmware library.

The kit contains an STM8S touch sensing (TS) evaluation daughterboard (STM8Sxxx-TS1) in addition to the STM8/128-EVAL board.

The STM8S touch sensing evaluation daughterboard provides an evaluation platform for resistor-capacitor (RC) touch sensing technology for an implementation using 5 keys and one slider.

The STM8S TS evaluation kit provides a software solution for transforming any 8-bit STM8 microcontroller (MCU) into a capacitive touchkey controller.

For further details about the touch sensing software library, please read the technical documentation available on www.st.com/touch-sense-sw-lib.



Figure 1. STM8S touch sensing evaluation kit

February 2010

Contents

1	Ove	rview	3
2	Gett	ing started with the touch sensing evaluation kit	1
	2.1	Evaluation kit contents	4
	2.2	Using the evaluation kit	4
	2.3	User interface	5
3	Eval	uation kit board settings	7
	3.1	STM8S touch sensing daughterboard	7
		3.1.1 Dielectric	3
		3.1.2 Daughterboard MCU pin functions	3
		3.1.3 Daughterboard power supply1	1
		3.1.4 SWIM connections	1
		3.1.5 Analysis connectors (J2 and J3)12	2
		3.1.6 External supply and communication12	2
	3.2	STM8S2xx evaluation motherboard settings	3
4	Adva	anced evaluation using a debugging environment	5
	4.1	Running the evaluation firmware in debug mode	5
	4.2	Entering debug mode using RLink 15	5
	4.3	Exploring key structures 20)
Appen	dix A	STM8Sxxx-TS1 daughterboard schematics	ł
Revisio	on histo	ry 22	2



1 Overview

This solution enables designers, comfortable with the use of standard microcontrollers, to create higher-end "look and feel" user interfaces by replacing conventional electromechanical switches with touch sensing controls.

Designers can combine touch sensing functions with traditional MCU features (communication, LED control, beeper, LCD control, etc.).

The touch sensing firmware library is part of the application firmware.

Maturity, robustness, flexibility and performance make this solution simple to implement with a low "time to market" period making it easy to develop all types of applications including mobile phones, cooking appliances and printers only to name a few.

The STM8S TS evaluation kit comes pre-programmed with evaluation firmware that manages 5 touch keys and one slider all available on the touch sensing daughterboard. Using this kit, the user can easily evaluate the touch sensing software features and performances by monitoring sensing parameters using an LCD display interface.

In development mode, designers are able to debug, modify, adapt or integrate the touch sensing library in application firmware using a USB debugging and programming tools:

- Raisonance RLink debugger/programmer for ST microcontrollers
- STice in-circuit emulation system
- ST MCU Toolset with ST Visual Develop (STVD) IDE and ST Visual Programmer (STVP) programming interface



UM0671

2 Getting started with the touch sensing evaluation kit

2.1 Evaluation kit contents

The STM8S touch sensing evaluation kit (STM8/128-EV/TS) contains:

- STM8S touch sensing evaluation daughterboard (STM8Sxxx-TS1)
- STM8/128-EVAL board
- AC/DC power supply and its AC adaptors
- MCU selection guide
- User manual (this document)

IMPORTANT: The STM8 TS library, STM8 Toolset, STM8S firmware library and related documentation are available at www.st.com/touch-sense-sw-lib

2.2 Using the evaluation kit

After connecting the motherboard to the mains supply, the evaluation kit is ready for use.

Just use the joystick to navigate through the menu options as described in Section 2.3: User interface.

The evaluation firmware enables the user to quickly evaluate the main features of the touch sensing library.



Figure 2. TS evaluation kit assembly

4/23



2.3 User interface

The joystick is the main user interface used to display keys values and state or to modify the main touch sensing library parameters: de-bounce filter, detection time-out, low power mode, DES setting, etc.

Note: The 5 touch sensing keys are also for navigating through the touch sensing menu.



Figure 3. Navigation scheme

 After power on, to quickly display the key and slider states (display option 1), move the joystick once to the right (→) and twice towards the bottom (↓).



Display options

Use the joystick to navigate through the sub-menus as shown in *Figure 3* to do the following:

- 1. Display the state of keys (K1, K2, K3, K4 and K5) and slider on the same screen.
- 2. Display only the state of the selected key (Kx) (Idle or Detected).
- 3. Display the signal value and the reference threshold of the selected Key (Kx).
- 4. Display the number of samples rejected by the noise filtering system.
- 5. Display the state of the slider (Idle or Detected).
- 6. Display the signal value and the position of the slider.
- 7. Display/change slider resolution (default is 4 bits, may be set to up to 7 bits).

Parameter setting options

Use the joystick to navigate through the sub-menus as shown in *Figure 3* to set the following options:

1. Detection time-out

This feature automatically recalibrates keys after a fixed duration of continuous touch detection. This prevents the keys from becoming 'stuck on' due to foreign objects or other sudden influences. This is known as the detection time-out feature.

After recalibration, the keys will continue to operate normally, even if partially or fully obstructed.

Infinite timeout (detection time-out value = 0) is useful in applications where a prolonged detection can occur and where the output must reflect the detection regardless of its duration.

2. De-bounce filter

The de-bounce filter reduces the effects of low frequency noise on key states. This mechanism requires a specified number of measurements that qualify as detections (and these must occur consecutively) or the detection will not be reported.

In a similar manner, the end of a touch (loss of signal) also has to be confirmed over several measurements. This process acts as a type of "de-bounce" mechanism against noise.

The typical value is equal to 2.

3. Low power mode

To reduce device power consumption, this feature inserts a low power mode window between each key acquisition period.

This window duration is programmable with the following typical values (in milliseconds): 0, 1, 2, 4, 8, 16, 32, 64, and 128.

4. Detection Exclusion System

The Detection Exclusion System (DES) prevents multiple keys from responding to a single touch. This can happen with closely spaced keys.

Once a key is considered as "touched", all other keys are locked in an untouched state. To unlock these keys, the touched key must return to an untouched state.



3 Evaluation kit board settings

3.1 STM8S touch sensing daughterboard



Figure 4. STM8S TS daughterboard overview

STM8S MCU

This board uses a STM8S microcontroller (STM8S207K6T6C) in a 32-pin LQFP package.

Keys

The 5 touchkeys (electrodes) are made of a simple copper surface.



Slider

The slider consists of 5 elementary juxtaposed electrodes.

Analysis connectors

All electrode and driven shield signals are available through two connectors (J2 and J3) for analysis and monitoring.

I²C communication connector

In the event of I^2C communication, the daughterboard provides a connector (J6) for the I^2C data and clock signals for interfacing with the STM8S microcontroller.

SWIM connector and setting jumper

A single-wire interface module (SWIM) interface (J5) with its associated jumper (W1) for analysis and development.

3.1.1 Dielectric

A 1.5-mm thick Plexiglas panel (*Figure 5*) is used as a dielectric between the electrodes and the touch surface. The user can replace this panel by another kind of dielectric with a different thickness and/or material. Consequently, new firmware parameters will have to be tuned.



Figure 5. Interchangeable dielectric panel

3.1.2 Daughterboard MCU pin functions

Table 1 describes the application functions assigned to each pin of the daughterboard microcontroller.

The motherboard provides certain resources for the daughterboard such as the LCD, power supply, LEDs, joystick, buzzer, etc.

For more information, see Section 3.2: STM8S2xx evaluation motherboard settings.

Table 1.Daughterboard MCU pin description

Pin no.	Pin name	Application usage	Option	Configuration
1	RST			
2	PA1	LED4	Disconnect	R17
3	PA2	Backlight		



Pin no.	Pin name	Application usage	Option	Configuration
4	VSS			
5	VCAP			
6	VDD			
7	VDDIO_1			
8	PF4	Load		
9	VDDA			
10	VSSA			
11	PB5	LED2	I ² C SDA	R43
12	PB4	LED3	I ² C SCL	R42
13	PB3	Joy Down		
14	PB2	Joy Left		
15	PB1	Joy Right		
16	PB0	Joy Up		
17	PE5	LCD CS		
18	PC1	Key K1		
19	PC2	Key K2		
20	PC3	Кеу КЗ		
21	PC4	Buzzer	Driven shield keys	R4/R46
22	PC5	SPI SCK		
23	PC6	SPI MOSI		
24	PC7	Key K4		
25	PD0	Key K5		
26	PD1	DB SWIM connector	MB SWIM connector	W1
27	PD2	Slider S5		
28	PD3	Slider S4		
29	PD4	Slider S3		
30	PD5	Slider S2		
31	PD6	Slider S1		
32	PD7	Driven shield slider	LED1	R45/R44

 Table 1.
 Daughterboard MCU pin description (continued)

Note:

Options are selected using 0-ohm resistors.





Table 2. Daughter/motherboard CN1 and CN5 header connections

10/23



3.1.3 Daughterboard power supply

By default, the daughterboard is powered through the motherboard. The 3.3 V regulator on the daughterboard supplies the daughterboard MCU.

The MCU current consumption (I_{DD}) can be measured by removing jumper W2.

3.1.4 SWIM connections

The STM8 debug system interface allows a debugging or programming tool to be connected to the MCU through a single-wire bidirectional communication based on open-drain line.

This single-wire interface module (SWIM) module allows non-intrusive read/write accesses to be performed on-the-fly to the RAM and peripheral registers, for debug purposes.

The SWIM module can also perform a MCU device software reset and can also be used by as a standard I/O port with some restrictions.

Jumper W1 is used to configure SWIM settings as described in Table 3.

Table 3.W1 jumper description

Configuration	Description
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	Uses STM8S TS daughterboard SWIM connector (default setting)
$\begin{array}{c}1 & 2 & 3\\ \bullet & \bullet \end{array}$	Connect PD1of STM8S TS daughterboard device to resource of STM8S/128- EVAL board.

For more information, please refer to user manual UM0470: STM8 SWIM communication protocol and debug module.

Figure 6. SWIM connector (top view)



Table 4. SWIM connector pin description

Pin number	Description	Pin number	Description
1	VDD	2	PD1
3	GND	4	PA0 (RESET)

Note:

The SWIM connector of the STM8/128-EVAL board cannot be used.



Application designers can use connectors J2 and J3 to analyze electrode and driven shield signals on the daughterboard.

Note: The user should take into account possible probe capacitance disturbance and should consider re-calibrating the device before use.

Table 5.	J2 connector	pin description
----------	--------------	-----------------

Pin number	Description	Pin number	Description
Shield	Slider 1 active shield	K3	Key 3 electrode
K1	Key 1 electrode	K4	Key 4 electrode
K2	Key 2 electrode	K5	Key 5 electrode

Table 6.J3 connector pin description

Pin number	Description	Pin number	Description
S1_1	Slider 1 electrode 1	S1_4	Slider 1 electrode 4
S1_2	Slider 1 electrode 2	S1_5	Slider 1 electrode 5
S1_3	Slider 1 electrode 3	Shield	Slider 1 active shield

3.1.6 External supply and communication

Connector J6 provides I²C data and clock signal pins for communication with the daughterboard microcontroller. The user can use these pins to develop his own communication protocol.

Resistors R36 and R37 are available as I^2C pull-up resistors and can be mounted by the user if needed.

Note: If the device is supplied by pin 1 and 5, jumper W2 must not be connected.

I description
1

Pin number	Description	Pin number	Description
1	VSS	4	I ² C SDA
2	PA1	5	VDD
3	I ² C SCL		

Pin 2 is connected to PA1 and is available for user implementation. For example, it may be used for acquisition synchronization.

3.2 STM8S2xx evaluation motherboard settings



Figure 7. TS motherboard overview

When using the STM8S TS daughterboard assembled with the STM8/128-EVAL (mother) board, the following settings must be implemented:

- To maintain STM8S TS daughterboard MCU functionality, the Reset Source (JP1) jumper must be set to the "STice" position (*Table 8*) on the STM8/128-EVAL board.
- To supply power from the supply jack (CN6) to both mother and daughterboards connected on connectors CN5 and CN1:
 - On STM8S TS daughterboard, connect both pins on jumper W2 (Section 3.1.3).
 - On STM8/128-EVAL board, set jumper JP3 as shown in *Table 8*. (STM8S TS daughterboard must have not its own power supply connected.)

For more information about the STM8/128-EVAL board, please refer to UM0482: STM8/128-EVAL evaluation board user manual.



Jumper	Description	Configuration
JP1	Set to "STice" to keep motherboard MCU Reset pin low to enable correct TS daughterboard MCU functionality. (Default configuration)	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$
JP3	Connect both PSU and DTB jumpers to supply power supply the TS daughterboard. (Default configuration)	DTB PSU

Table 8. STM8/128-EVAL board settings

14/23



4 Advanced evaluation using a debugging environment

4.1 Running the evaluation firmware in debug mode

Designers can easily run the evaluation firmware in Debug mode using ST debugging and programming tools.

- Hardware tools to be ordered separately:
 - Raisonance RLink debugger/programmer for ST microcontrollers (www.raisonance.com/)
 - STice in-circuit emulation system
- Software tools to be downloaded separately:
 - ST MCU Toolset with ST Visual Develop (STVD) IDE and ST Visual Programmer (STVP) programming interface
 - STM8 Cosmic C compiler (www.cosmic-software.com/)

In Debug mode, designers can perform an in-depth evaluation of the firmware and visualize touch sensing parameters. Designers can even customize the touch sensing library to their specific needs by creating their own application firmware.

For further information about ST software, STM8 microcontrollers or debugging tools, please read the associated documentation or ask your local ST support team for a training session. More information is available at www.st.com/touch-sense-sw-lib

4.2 Entering debug mode using RLink

This section describes the steps necessary to set up a platform for evaluating and developing TS firmware.

Note: For links to tools and downloads, please refer to ST's microcontroller web site at www.st.com/mcu/.

- 1. Download and install the *ST MCU Toolset* with ST Visual Develop (STVD) IDE and ST Visual Programmer (STVP) programming interface (one single download).
- 2. Download and install the STM8 cosmic C compiler.
- 3. Install the Rlink USB driver using one of the two following methods:
 - Insert the RAISONANCE CD-ROM into your CD drive and exit from Autorun.
 Explore the CD-ROM contents and execute the *RLinkUSBInstall.exe* file in the *D:\Driver\RlinkDrv* directory.
 - Download the *RLinkUSBInstall.exe* file from the Raisonance web site and execute the file.
- 4. Restart your computer to apply these changes.
- 5. Install the RLink hardware:
 - a) Place jumpers on the "SWIM" and "ADAPT" RLink configuration pins of the Rlink USB adaptor as shown in *Figure 8*.
 - b) Connect the SWIM-STM8 adaptor to the RLink USB adaptor.
 - c) Connect the SWIM cable between the touch sensing daughterboard (J5 SWIM connector) and the SWIM-STM8 adaptor as shown in *Figure 9*.
 - d) Connect the USB cable between your PC and the RLink USB adaptor.
 - e) Power the TS daughterboard through the STM8S2xx evaluation board.











- 6. Download the STM8 touch sensing library from www.st.com/touch-sense-sw-lib
- 7. Launch the ST Visual Develop (STVD) integrated development environment.
- 8. Load the STM8 TS evaluation firmware (*Figure 10*):
 - In the "File" menu, click "Open workspace" and select/open the file STM8S20xK_TS1_EVAL_FW.stw.
 (Default path file @ \STM8S20xK_TS1_EVAL_FW\Project\STVD\Cosmic)



Eile Edit View Project	Build Debug Debug instrum	ent <u>T</u> ools <u>W</u> indo	w Help	International and a council	e in II ei	
		300		12 92 99 99 A A A		
Workstoace	- 11 192 194 196 43 (0* 63	i 01 01 01 1				Steel
No Workspace						
		Open Work	space			2
		Look in:) Cosmic			* == •
		Release				
		<u> </u> STM8520	K_ISI_EVAL_FW.stw			
<u> </u>						
Workspace						
*						
		File name:	STM8S20xK_TS1_EVAL_F	V.stw		Open
					1.1	

Figure 10. Loading the TS evaluation firmware

9. In the "Project" menu, select "Setting" to define the C cosmic location directory (*Figure 11*).

Figure 11. Project settings

🖹 😅 🝵 🔲 🖨 🗍 🗖 🚧 📗 Workspace	Settings for: Debug	General Debug MCU Selection C Compiler Assembler	- © 🕮 🛎 🛎
版 STM8520ik_TS1_EVAL_FW.stw 法一翻 stm0520isk_ts1_eval_fw	€ ∰ stm8s20ak_ts1_eval_fw	Defaults Toolset Info Toolset STM8 Cormic Project specific toolset path Root path C:VProgram File:\COSMIC\UCXSTM8_1 Toolset sub-path(felative to the Root path) Bin path Include path Hstm8 Library path Lib Output directory Debug	

- 10. Build the project by compiling and linking all the source code.
 - a) In the "Build" menu, select "Rebuild all".
 - b) When the process is completed, check that there are no errors and no warning messages in the Output windows.



Eile Edit View Proje	t <u>B</u> uild <u>D</u> ebug Debu	g instrument Tools	s <u>W</u> indow						
Workspace	Compile Build AL Batch Build Clgan Clgan	Ctrl+F7 F7							
 	Configurations.								
- S									
Workspace									
<pre>www.space *</pre>	tm8_ts_lib\sour c ogram Files\COS \stm8s20xk_ts1_ Build step \stm8s20xk_ts1_	cces\stm8_ts SHIC\CXSTH8_ eval_fw.sm8 eval_fw.s19	l_rc_timerdri 16K\Lib" -o Debug\stm8s2	er.c: ebug\stm8s20xk_t: xk_ts1_eval_fw.sn	s1_eval_fw.s	m8 −mDebug\s	tm8s20xk_ts1_e	eval_fw.map De	bu
workspace x	tm8_ts_lib\sour c ogram Files\CO2 \stm8s20xk_ts1_ Build step \stm8s20xk_ts1_ _eval_fw.elf -	cces\stm8_ts HIC\CXSTM8_ eval_fw.sm8 eval_fw.s19 0 error(s),	l_rc_timerdri 16K\Lib" -o Debug\stm8s2 0 warning(s)	er.c: ebug\stm8s20xk_t; xk_ts1_eval_fw.sn	si_eval_fw.s n8	m8 −mDebug\s	tm8s20xk_ts1_e	eval_fw.map De	bu
workpace Aunning Linke clnk -1"C:\Pr cvdwarf Debug Running Post- chex -0 Debug stm8s20xk_ts1 d	tm8_ts_lib\sour c ogram Files\COS stm8s20xk_ts1_ stm8s20xk_ts1_ _eval_fw.elf - fw.elf -	cces\stm8_ts SHIC\CXSTM8_ eval_fw.sm8 eval_fw.sm9 0 error(s), s1) FindinFile:	l_rc_timerdri 16K\Lib" -o Debug\stm8s2 0 warning(s) s2 <u>}</u> Debug <u>}</u> Cor	rer.c: rebug\stm8s20xk_t: 1xk_ts1_eval_fw.sn oke /	si_eval_fw.s	m8 -mDebug\s	tm8s20xk_ts1_e	eval_źw.map De	bu

Figure 12. Building the project

- 11. Select RLink as the debugging tool.
 - In the "Debug instrument" menu, click "Target setting" and select "SWIM RLink" as shown in *Figure 13*.

Figure 13. Debug instrument settings

ST Visual Develop - STM8520xK_TS1_EVAL_FW.stw Ele Edt Vew Project Build Debug Ipstrument Icols Window Help
Debug Instrument Settings C Workspace Tagel Tagel Debug Instrument Selector: Select the Tagel you want to use for debug sersion. Swim Flink. Her Page Star Debug for Simulator Stop Swim Flink. Image: Simulator Simulator Stop Carget Port Selector: Swim Flink Varies Connection port for the Taget selected above. Add Permove Add Permove Now the selected taget notification at start of debugging session

- 12. Start the debugging process.
 - In the "Debug" menu, select "Start debugging" and click "Run" (or press CTRL-F5) to start the application in Debug mode.



🖆 STM8S205K6 STM SWIM - STM8S20xK_TS1_EVAL	_FW.stw* - [I	- [Debug] - stm8s20xk_ts1_eval_fw.elf - [main.c]	BX
🖆 Eile Edit View Project Build Debug Debug instrument	<u>T</u> ools <u>W</u> indow	ow Help	8 ×
👔 💣 💼 📮 🚭 🗠 start Debugging			
Stop Debugging		etm9e20vk te1 v Dahun v 🕾 🕮 🖽 👗 🤐	
Workspace	Alt+Num *		× ¥
	Childers		
stmoszuxk_tst_eval_eval_eval	Lichthier	2 0x6001	
	ri+onir(++5	**************************************	
Restart		**************************************	
The monu c	F5	* @file main.c Ox6008	
The main c		* @brief This file contains the main function. Ox600a	
iovstick button d	Shift+F5	* @author STMicroelectronics - MCD Application 🧔 🛛 🛛 🛛 🛛 🔿	
functions c De chan take	E11	Team 0x600e	
demo ro stm8s.c	FII	* @version V1.0.0 0x6011	
themo help.c	F10	* @date 15-AUG-2008 0x6014	
demo about.c 🔁 Step Into ASM	Alt+F11	₽ 0x6016	
stm8 interrupt v 🔂 Step Over ASM	Alt+F10	**************************************	
E STM85_STANDAR (Step Out	Ctrl+E11	0x601e	
E STM8_T5_LIB #D Run to Gunne	CHUCIO	* 0x601f	
Include Files	CUNHFID	* THE PRESENT FIRMWARE WHICH IS FOR GUIDANCE 7 0x6022	
stm8_tsl_rc_conf 👷 set PC		ONLY AIMS AT PROVIDING CUSTOMERS 0x6026	
mono_lcd.h		* WITH CODING INFORMATION REGARDING THEIR 7 Ox602a	
stm8s_conf.h		PRODUCTS IN ORDER FOR THEM TO SAVE	
menu.h	12	NOT DE MELO LADIE FOR MW	
main.h	12	A DIRECT INDIRECT OR CONSPONENTIAL DAWACES	
joystick_button.h		WITH DESDECT TO MAY CLAIMS ADISING	
🔄 📋 functions.h		* FOON THE CONTENT OF CHICH FIDENIADE AND (OD THE - W ON COD)	
Workspace	demo	no_help.c main.c	
Variable	Value	Type Address	
C SUKeyinto	{{State = {whole	hole = Uxb, b = {LALIBRATIUN = Ux1, IDLE = struct { union { unsigned char Ux13b	
± [U]	{State = {whole	ole = 0xb, b = {CALIBRATION = 0x1, IDLE = 0 struct { union { unsigned char 0x13b	
	{State = {whole	ole = 0xd, b = {LALIBHATION = 0x1, IDLE = 0 struct { union { unsigned char 0x148	
	totate = (whole	ute = uxee, b = (LALIDHATION = UXU, IDLE = struct (union (unsigned char UX155	
+ [4]	(State = (whole	ole = 0x10, 0 = (CALIDDATION = 0x0, IDLE = struct (union (unsigned char 0x162	
	(orace = twhole	ore – oxao, o – (cascronia novi = oxo, rolle = struct (union (unsigned char UX16r	
2 whether the state of the stat			
Watch 1 A watch 2 A watch 3 A Watch	4 /		
Heset the chip and run the program		Ln 1, Col 1 MODIFIED READ CAP NUM SCRL OVR Stop	Ready
🧉 start 🔰 🗵 🖸 🖿 🙆 😂 🖿 📀	🔘 Inbo	nbox - Microsoft 🖳 TSSL getting star 🤷 STM85205K6 ST 🚺 100% 📲 🕥 🖏 🔃 🕏	14:45

Figure 14. Debug mode



4.3 Exploring key structures

All key and slider data structures can be monitored through the STVD watch window.

The main "touch sensing" structures are "sSCKeyInfo" and "sMCKeyInfo".

To learn more about library variables and function descriptions, please refer to the CHM file available at <installation path>\STM8_TS_LIB\stm8_tsl_um.chm

Figure 15. STVD watch window

🖆 STM85205K6 STM SWIM - STM8520xk		ichannelkey.c]	_ 🗆 🗙	
🖆 Eile Edit View Project Build Debug instrument Iools Window Help				
12 2 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1				
			1	
1102 - ((<u> </u>	
1103 phckeyst	ruct->channel[1].IntegratorCounter = phokeystruct->Direct	:ionunangeintegrator;		
1104 4 phokeyst	ruct->setting.s.bikeciiow - 1, // wew ullection accepted	anciciockwise.	-	
main.cSTM8_TSL_Rstm8_t	st_rc_iod menu.h menu.c stm8_tst_rc_mult			
× Variable	Value	Type	Address	
SMCKevInfo	{{State = {whole = 2, b = {CALIBRATION = 0, IDLE = 1, DETECTED = 0, ERROR = 0, PRE	struct { union { unsigned char whole: struct {	0x13d	
Ē [0]	{State = {whole = 2, b = {CALIBRATION = 0, IDLE = 1, DETECTED = 0. ER	struct { union { unsigned char whole:	0x13d	
E State	{whole = 2, b = {CALIBRATION = 0, IDLE = 1, DETECTED = 0, ERROR = 0, PRE_STATE	union { unsigned char whole; struct { unsigned	0x13d	
	{whole = 515, b = {IMPLEMENTED = 1, ENABLED = 1, DETECTED = 0, CHANGED = 0, P	union { unsigned short whole; struct { unsigned	0x13e	
Counter	0	unsigned char	0x140	
DESGroup	0	unsigned char	0x141	
Channel	{{LastMeas = 1258, LastMeasRejectNb = 0, Reference = 1257, IntegratorCo	struct { unsigned short LastMeas; unsign	0x142	
Ē. [0]	{LastMeas = 1258, LastMeasRejectNb = 0, Reference = 1257, IntegratorCo	struct { unsigned short LastMeas; unsign	0x142	
	{LastMeas = 1434, LastMeasRejectNb = 0, Reference = 1434, IntegratorCounter = 0, ECSR	struct { unsigned short LastMeas; unsigned char La	0x149	
⊡ [2]	{LastMeas = 1501, LastMeasRejectNb = 0, Reference = 1501, IntegratorCo	struct { unsigned short LastMeas; unsign	0x150	
	{LastMeas = 1505, LastMeasRejectNb = 0, Reference = 1506, IntegratorCounter = 0, ECSR	struct { unsigned short LastMeas; unsigned char La	0x157	
	{LastMeas = 1403, LastMeasRejectNb = 0, Reference = 1405, IntegratorCo	struct { unsigned short LastMeas; unsign	0x15e	
Position	0	unsigned char	0x165	
UnScaledPosition	0	unsigned char	0x166	
DetectThreshold	60	signed char	0x167	
EndDetectThreshold	20	signed char	0x168	
RecalibrationThreshold	-20	signed char	0x169	
Resolution	4	unsigned char	Ox16a	
DirectionChangeIntegrator	1	unsigned char	0x16b	
DirectionLhangeThreshold	30	unsigned char	Ux16c	
SSUKeyInfo	((State = (whole = 2, b = (CALIBRATION = 0, IDLE = 1, DETECTED = 0, ERRUR = 0, PRE	struct { union { unsigned char whole; struct {	0x175	
	(State = (Whole = 2, D = (LALIBRATION = 0, IDLE = 1, DETECTED = 0, ERROR = 0, PRE	struct { union { unsigned char whole; struct {	0x175	
C State	(WHILE = 2, U = (UALIDRATION = 0, IULE = 1, UETEUTEUTEUTEUTEUTEUTEUTEUTEUTEUTEUTEUTE	union (unsigned char whole; struct (unsigned	08175	
Counter	WINDE = 3, D = VIMITLEMENTED = 1, ENABLED = 1, DETECTED = 0, CHANGED = 0, POS	union t unsigned short whole; struct t unsigned	0.170	
DESGraue	0	unsigned char	0.170	
t Channel	v { astMeas - 470 astMeasBejectNb - 0 Beference - 471 IntegratorCount	struct { unsigned short astMeas: unsign	0x175	
DetectThreshold	12	signed char	0v181	
EndDetectThreshold	7	signed char	0v182	
BecalibrationThreshold		signed char	0x183	
+	{State = {whole = 2, b = {CALIBBATION = 0, IDLE = 1, DETECTED = 0, EBBOB = 0, PBE	struct { union { unsigned char whole: struct {	0x184	
±[2]	{State = {whole = 2, b = {CALIBRATION = 0, IDLE = 1, DETECTED = 0, FRBOR = 0, PRF	struct { union { unsigned char whole; struct {	0x193	
	{State = {whole = 2, b = {CALIBRATION = 0, IDLE = 1, DETECTED = 0. ERROR = 0. PRE	struct { union { unsigned char whole; struct {	0x1a2	
± [4]	{State = {whole = 2, b = {CALIBRATION = 0, IDLE = 1, DETECTED = 0, ERROR = 0, PRE	struct { union { unsigned char whole; struct {	0x1b1	
£				
Watch 1 / Watch 2 / Watch	ch 3 X Watch 4 /	•		
For Help, press F1	Lo 1104 Col 1		Beadu	
Participation 1	junitos, con	hard been been been been been been	11	

20/23







UM0671

Revision history

Date	Revision	Changes
04-Feb-2009	1	Initial release.
09-Mar-2009	2	Updated Figure 3: Navigation scheme on page 5 and Display options on page 6.
20-Mar-2009	3	Corrected reference to STM8S touch sensing evaluation kit from "STM8/128-EVAL/TS" to "STM8/128-EV/TS".
26-Feb-2010	4	Updated values of C3 and C9 in <i>Figure 16 on page 21</i> .

Table 9.Document revision history

22/23



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 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.

© 2010 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

