
Getting started with touch sensing control on STM32 microcontrollers

Introduction

This document helps customers to quickly locate information regarding touch sensing on STM32 microcontrollers.

It is applicable to STM32F0, STM32F3, STM32L0, STM32L1 and STM32L4 Series products. It lists all the existing application notes and user manuals covering touch sensing. It indicates where the key aspects of touch sensing are documented.

It also explains how to build touch sensing applications on STM32L0538-DISCO and STM32F072B-DISCO discovery boards using the STM32CubeMx graphical interface.

1 General information

This document applies to Arm®-based devices.

Note: Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



2 Terminology and principle

2.1 Terminology

The touch sensing most relevant acronyms are described below:

- Acquisition mode
 - CT: Charge-Transfer acquisition principle. This mode is used on STM32 microcontrollers.
- Touch sensing STM32 peripheral
 - TSC: touch sensing controller peripheral
 - Bank: set of channels acquired simultaneously
 - Channel: elementary acquisition item
 - Group: set of 1..3 channels plus 1 sampling capacitor (Cs)
- Sensors
 - Touchkey or TKey: single channel sensor
 - Linear sensor: multi-channels sensor with the electrodes positioned in a linear way
 - Rotary sensor: multi-channels sensor with the electrodes positioned in a circular way
 - Active shield: track running along or copper plane surrounding the sensor track and/or sensor itself. Active shield is driven similarly to the sensor. Improve noise robustness without decreasing the sensitivity.
- STM32 software
 - TSL: touch sensing library
 - Delta: difference between the measure and the reference
 - Measure or meas: current signal measured on a channel
 - Reference or ref: reference signal based on the average of a sample of measures
 - DTO: detection time out. Time out is defined by TSLPRM DTO. See TSLPRM DTO in tsl_conf.h file.
 - DXS: detection exclusion system. Exclusion system is defined by TSLPRM_USE_DXS. See TSLPRM_USE_DXS in tsl_conf.h file.
 - ECS: environment change system. See TSLPRM_ECS_DELAY in tsl_conf.h file.
- Hardware Involved
 - Cx: sensor capacitance (typical value is few pF)
 - Cp: parasitic capacitance (typical value few pF)
 - Ct: equivalent touch capacitance
 - Cs/Cskey/Csshield: sampling capacitor (typical value from 2.2 to 100nF)
 - Rs/Rskey/Rsshield: serial resistor, ESD protection (typical value from 100Ohms to 10K)

2.2 Principle

The STM32 touch sensing feature is based on charge transfer.

The surface charge transfer acquisition principle consists in charging a sensor capacitance (Cx) and in transferring the accumulated charge into a sampling capacitor (Cs).

This sequence is repeated until the voltage across Cs reaches V_{IH} .

The number of charge transfers required to reach the threshold is a direct representation of the size of the electrode capacitance. When the sensor is touched, the sensor capacitance to the earth is increased. This mean the C voltage reaches V_{IH} with less count and the measurement value decreases. When this measurement goes below a threshold, a detection is reported by the TSL. The schematic below do not take into account the parasitic capacitor.

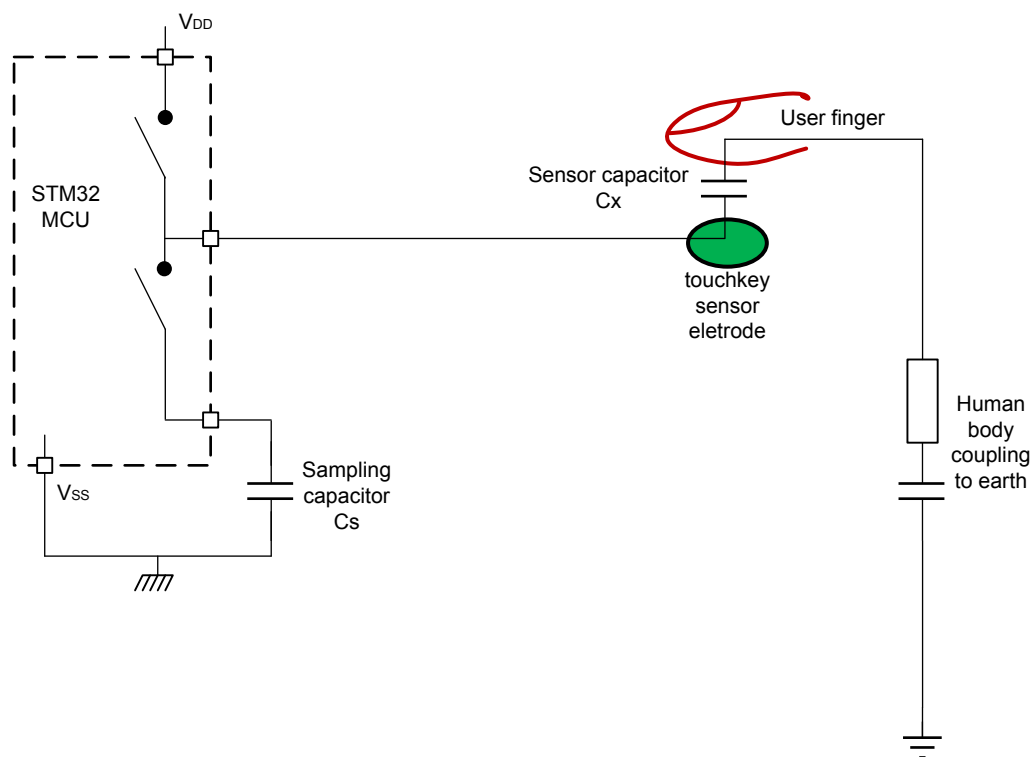
Figure 1. Change transfer principle


Table 1. [Change transfer principle documentation](#) gives a list of documents containing information about the change transfer principle.

Table 1. Change transfer principle documentation

Id	Title	Chapters
AN4299	Guidelines to improve conducted noise robustness on STM32F0/F3/L0/L4 series touch sensing applications	Surface charge transfert acquisition principle overview
AN4310	Sampling capacitor selection guide for MCU based touch sensing applications	Charge transfert acquisition principle overview
AN4312	Guidelines for designing touch sensing applications with surface sensors	Capacitive sensing technology in ST
AN4316	Tuning a STMTouch-based application	Charge transfer periode tuning
OLT	STM32L4 On Line Training	Touch sensing controller (TSC)

3 Document reference

Figure 2. Main documentation tree shows the main documentation tree related to TSC and TSL.

Figure 2. Main documentation tree

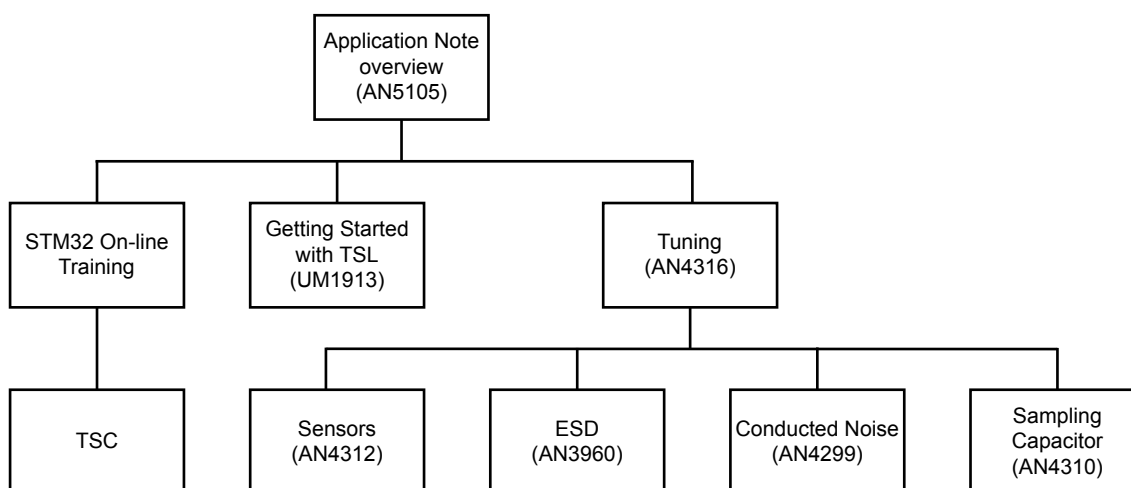


Table 2. References documentation

Document name	Document title
UM1913	Developing applications on STM32Cube with STMTouch® touch sensing library
AN3960	ESD considerations for touch sensing applications
AN4299	Guidelines to improve conducted noise robustness on STM32F0/F3/L0/L4 series touch sensing applications
AN4310	Sampling capacitor selection guide for MCU based touch sensing applications
AN4312	Guidelines for designing touch sensing applications with surface sensors
AN4316	Tuning a STMTouch-based application

4 STM32L4 touch sensing controller online presentation

An online training is available under our website www.st.com.

Insert the "STM32L4 Online Training" string in the "Search" function and press enter.

To find it use the function "Search" and insert the strings "STM32L4 Online Training". Figure 4. STM32L4 Touch Sensing Controller online training shows the online page available .

Figure 3. STM32L4 online training

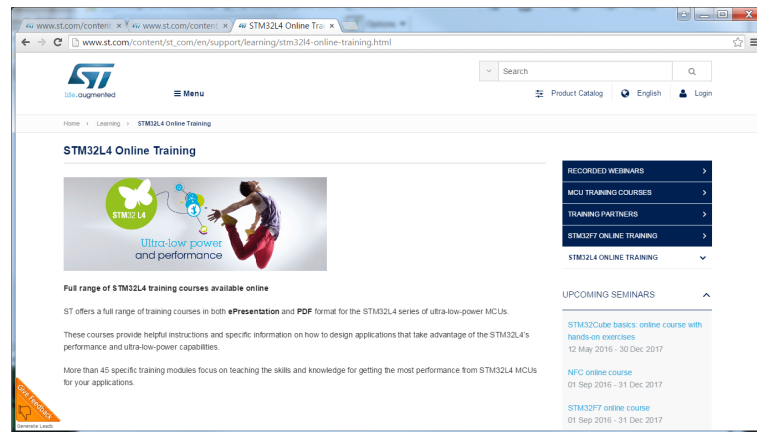
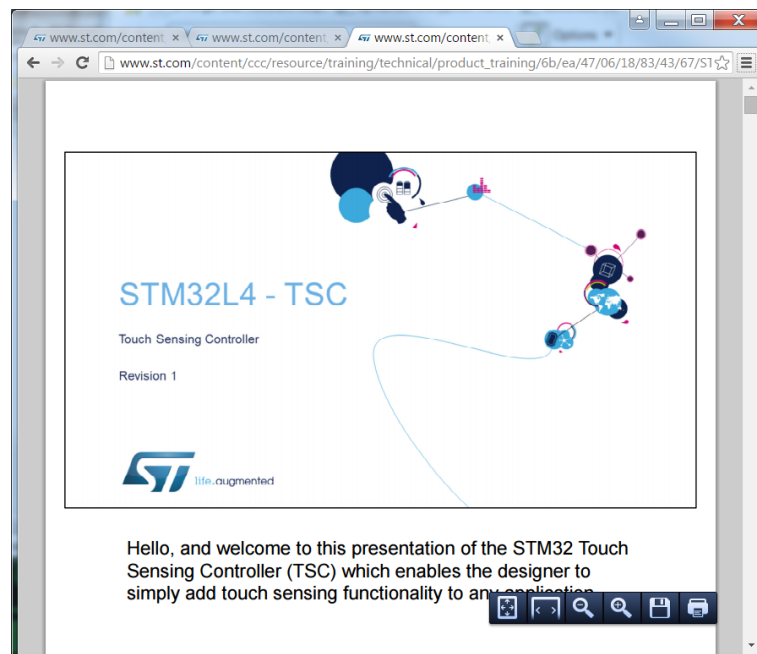


Figure 4. STM32L4 Touch Sensing Controller online training



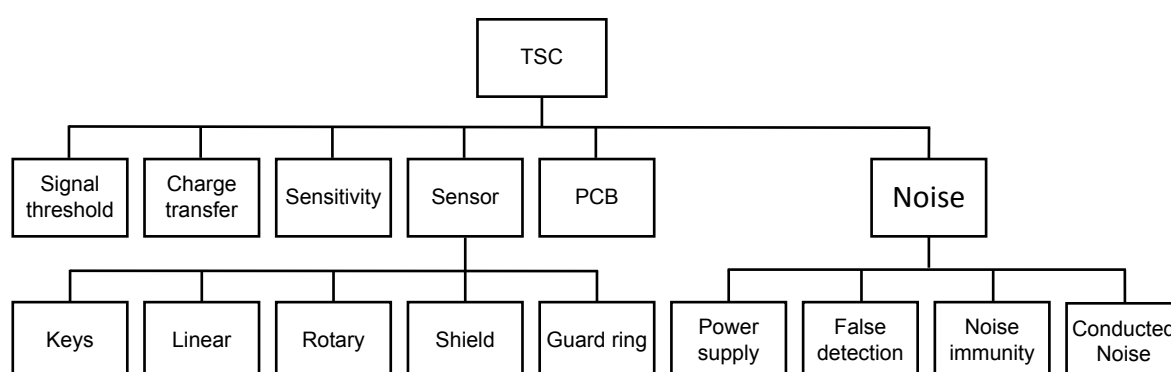
5 Main characteristics

5.1 Description

The following [Figure 5. TSC characteristics](#) shows all touch sensing controller (TSC) characteristics and their correlation.

The TSC main characteristics are described in the following pages.

Figure 5. TSC characteristics



5.2 Signal threshold

To tune the detection thresholds, it must determine the sensitivity of each touchkey. For each touchkey, can be used few parameters to adjust these signal thresholds.

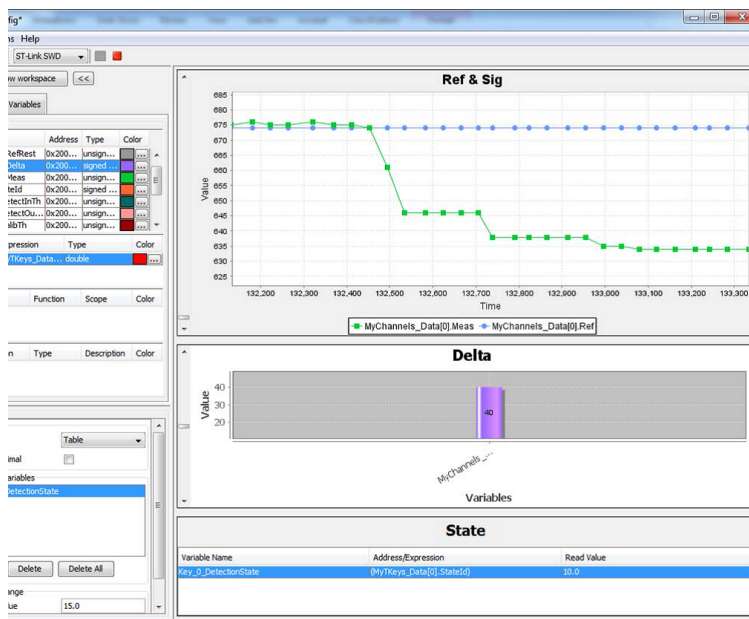
For debug purpose, it can get touchkey parameters using printf or STMStudio tool:

```

for (Index = 0; Index < NUMBER_OF_TOUCHKEYS; Index++)
{
    printf("K%1d [%2d] [%4d %3d %3d %4d] %d %d %d %d %d"
        , Index
        , MyTKeys[Index].p_Data->StateId
        , MyTKeys[Index].p_ChD->Ref
        , MyTKeys[Index].p_ChD->RefRest
        , MyTKeys[Index].p_ChD->Delta
        , MyTKeys[Index].p_ChD->Meas
        , MyTKeys[Index].p_Param->ProxInTh
        , MyTKeys[Index].p_Param->ProxOutTh
        , MyTKeys[Index].p_Param->DetectInTh
        , MyTKeys[Index].p_Param->DetectOutTh
        , MyTKeys[Index].p_Param->CalibTh
    );
}
  
```

Note: *ProxInTh and ProxOutTh are defined for proximity detection feature only, when TSLPRM_USE_PROX = 1.*

Figure 6. STMStudio outputs



- On software side:
 - Relevant information are located in `tsl_conf.h` and `tscl_user.c` files.
 - Threshold (`xx_TH`) can be adjust in `tsl_conf_tsc.h` file.:

See below an example:

```
#define TSLPRM_TKEY_DETECT_IN_TH (64)
#define TSLPRM_TKEY_DETECT_OUT_TH (60)
#define TSLPRM_TKEY_CALIB_TH (56)
#define TSLPRM_LINROT_DETECT_IN_TH (50)
#define TSLPRM_LINROT_DETECT_OUT_TH (40)
```

- The TSL api, `tsl_user_SetThresholds`, located in `tsl_user.c` allows to adjust each channel independently. See below an example:

```
void tsl_user_SetThresholds(void)
{
  /* USER CODE BEGIN Tsl_user_SetThresholds */
  /* Example: Decrease the Detect thresholds for the TKEY 0*/
  MyTKeys_Param[0].DetectInTh -= 10;
  MyTKeys_Param[0].DetectOutTh -= 10;
  /* USER CODE END Tsl_user_SetThresholds */
}
```

Table 3. [Signal threshold usage documentation](#) gives a list of documents containing information about the signal threshold usage.

Table 3. Signal threshold usage documentation

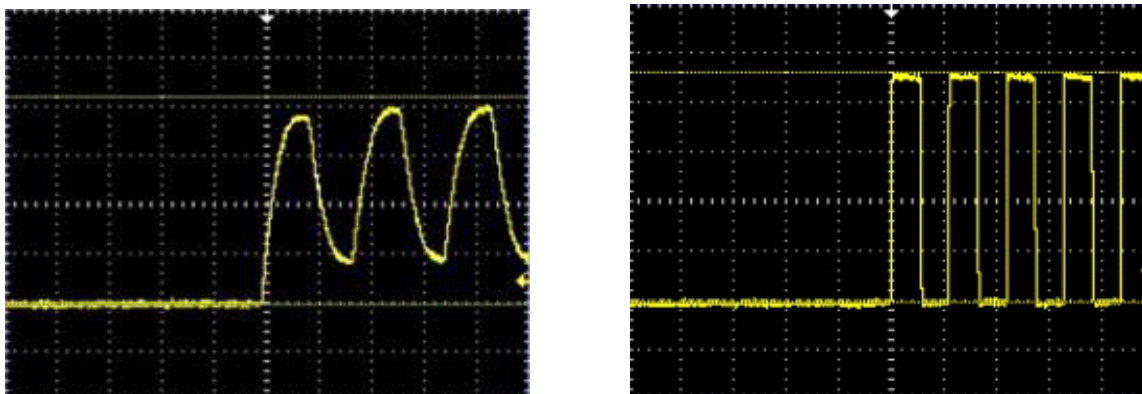
Id	Title	Chapters
UM1913	Developing applications on STM32Cube with STMTouch® touch sensing library	Debug with STMStudio
AN4316	Tuning a STMTouch-based application	Monitoring STMTouch driver variables using STMStudio Tuning of the Thresholds Touchkeys thresholds Linear and Rotary touch sensors thresholds

5.3 Charge transfer

The acquisition is based on the measurement of the sensor channel capacitance.

To ensure that the Cx capacitance is correctly charged, it is necessary to monitor the pin connected to the sensor. On sensors and shield sides, it must observe a complete Charge/Discharge cycle.

Figure 7. Incomplete and complete charge transfert cycle



In this example, to complete the charge transfer cycles, the following parameter must be modified as below:

- INCREASE:
 - htsc.Init.PulseGeneratorPrescaler
 - htsc.Init.CTPulseHighLength
 - htsc.Init.CTPulseLowLength
- DECREASE:
 - Sysclk

Table 4. Charge transfer documentation gives a list of documents containing information about the charge transfer.

Table 4. Charge transfer documentation

Id	Title	Chapters
AN4299	Guidelines to improve conducted noise robustness on STM32F0/F3/L0/L4 series touch sensing applications	Active shield
AN4316	Tuning a STMTouch-based application	Charge transfer period tuning

5.4 Sensitivity

Sensitivity is a key point in touch sensing applications. The sensitivity can be improved by:

- Reduce air gap
- Reduce panel thickness
- Choose dielectric with higher ϵ_R
- GND plane must not too close from shield and sensors
- Avoid metallic paint near shield and sensors

Table 5. Sensitivity documentation gives a list of documents containing information about the sensitivity.

Table 5. Sensitivity documentation

Id	Title	Chapter
AN1913	Developing applications on STM32Cube with STMTouch® touch sensing library	GPIO mode (table)
AN4312	Guidelines for designing touch sensing applications with surface sensors	Air Gap: <ul style="list-style-type: none"> • Reduce air gap Changing the Panel material: <ul style="list-style-type: none"> • Reduce Panel thickness • Choose dielectric with higher ϵ_R Metal chassis: <ul style="list-style-type: none"> • GND not too closed from Shield and Sensors • Avoid Metallic paint near Shield and Sensors Mechanical construction and PCB to panel bonding. Surface sensor design
AN4316	Tuning a STMTouch-based application	All chapter

Dielectric example

Table 6. Dielectric constants of common materials used in a panel construction

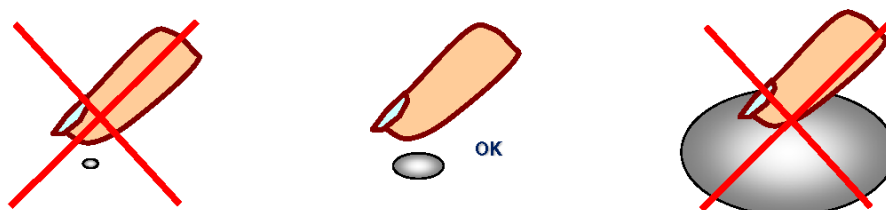
Material	ϵ_R
Air	1.00059
Glass	4 to 10
Sapphire glass	9 to 11
Mica	4 to 8
Nylon	3
Plexiglass	3.4
Polyethylene	2.2
Polystyrene	2.56
Polyethylene terephthalate (PET)	3.7
FR4 (fiberglass + epoxy	4.2
PMMA (Poly methyl methacrylate)	2.6 to 4
Typical PSA	2.0 - 3.0 (approximately)

5.5 Sensors

- It is recommended to use the same shape for all electrodes.
- The touchkeys can be customized by the drawing on the panel. TSL compensates capacitance differences.
- Acquisition time and processing parameters can be optimized when electrodes have similar capacitance.

Sensor size example

Figure 8. Sensor size



5.5.1 Key

- Key sensors are used in common application
- You can get deeper key information in following documents:

Table 7. [Key documentation](#) gives a list of documents containing information about the key.

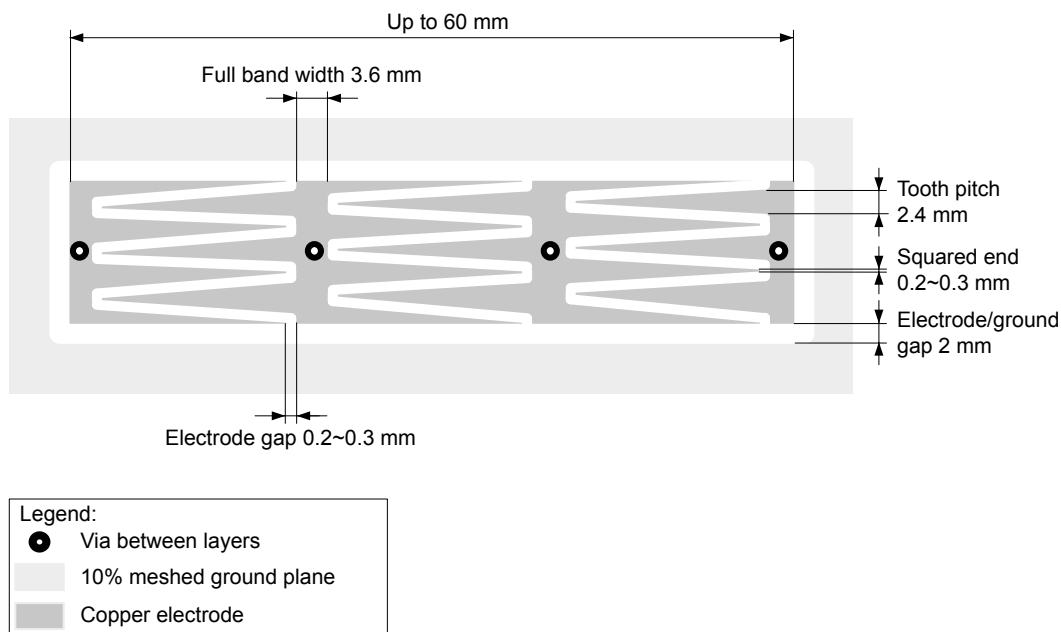
Table 7. Key documentation

Id	Title	Chapters
UM1913	Developing applications on STM32Cube with STMTouch® touch sensing library	Touchkey sensor
AN4312	Guidelines for designing touch sensing applications with surface sensors	Touchkey sensor

5.5.2 Linear or slider

A linear is a set of contiguous capacitive electrodes. [Figure 9. Interlaced linear touch sensor with 3 channels / 4 electrodes \(half-ended electrodes design\)](#) shows a slider used on a discovery board.

Figure 9. Interlaced linear touch sensor with 3 channels / 4 electrodes (half-ended electrodes design)



[Table 8. Linear touch sensor documentation](#) gives a list of documents containing information about the linear touch sensor.

Table 8. Linear touch sensor documentation

Id	Title	Chapters
UM1913	Developing applications on STM32Cube with STMTouch® touch sensing library	Linear and rotary touch sensors
AN4312	Guidelines for designing touch sensing applications with surface sensors	Linear sensor

5.5.3

Rotary or wheel

A rotary is a set of contiguous capacitive electrodes.

Figure 10. Interlaced patterned rotary sensor with 3 channels / 3 electrodes

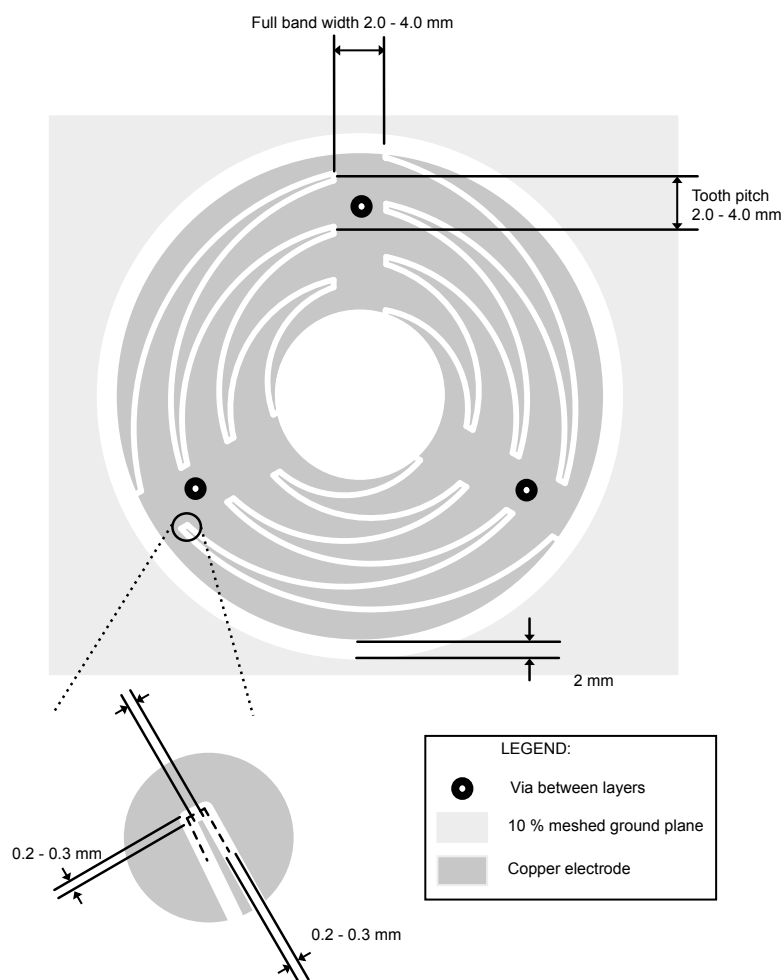


Table 9. [Rotary sensor documentation](#) gives a list of documents containing information about the rotary sensor.

Table 9. Rotary sensor documentation

Id	Title	Chapters
UM1913	Developing applications on STM32Cube with STMTouch® touch sensing library	Linear and rotary touch sensors
AN4312	Guidelines for designing touch sensing applications with surface sensors	Rotary sensor

5.5.4 Active shield or driven shield

Active shield or driven shield. (this name is used in some application notes) drives the shield plane with the same signal as the electrode.

There are several advantages using Active Shield instead of a grounded shield:

- The parasitic capacitance between the electrode and the shield no longer needs to be charged.
- Protect the touch electrodes from a noise source.
- Increase system stability and performance when a moving metal part is close to the electrode.

Figure 11. Active shield principle

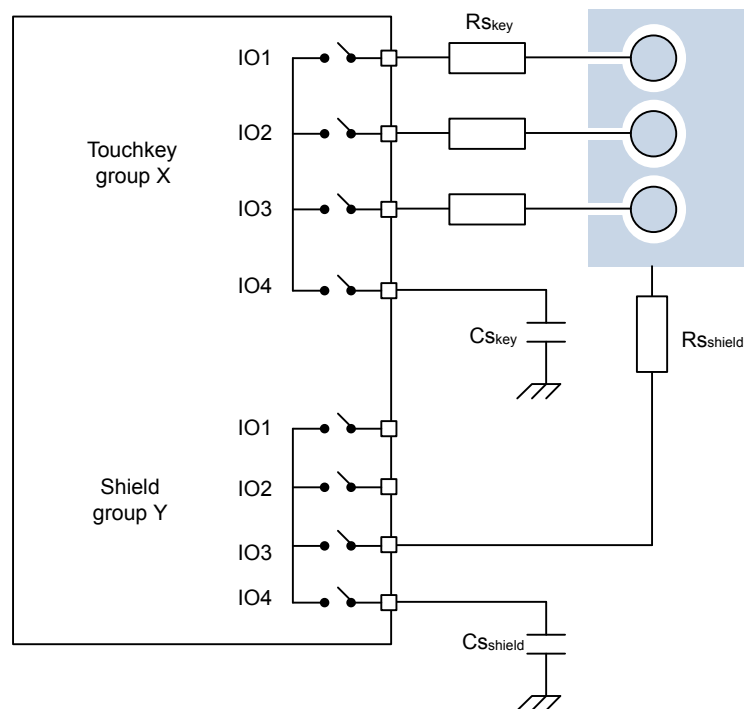


Table 10. Active shield documentation gives a list of documents containing information about the active shield.

Table 10. Active shield documentation

Id	Title	Chapters
AN4299	Guidelines to improve conducted noise robustness on STM32F0/F3/L0/L4 series touch sensing applications	Active Shield
AN4312	Guidelines for designing touch sensing applications with surface sensors	Driven shield
AN4316	Tuning a STMTouch-based application	Shield adjustment
OLT	STM32L4 Online Training	Touch sensing controller (TSC)

5.6 Layout and PCB

Rules to follow to improve TSC systems

5.6.1 Led rules

Figure 12. Led layout example

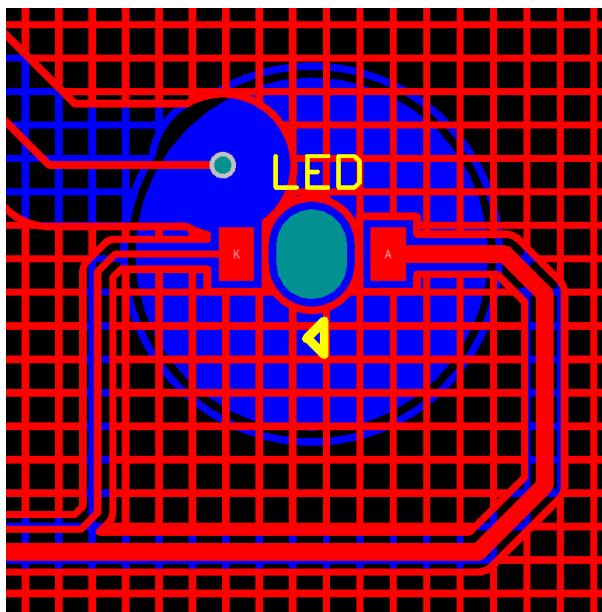


Figure 13. Example of cases where a LED bypass capacitor is required

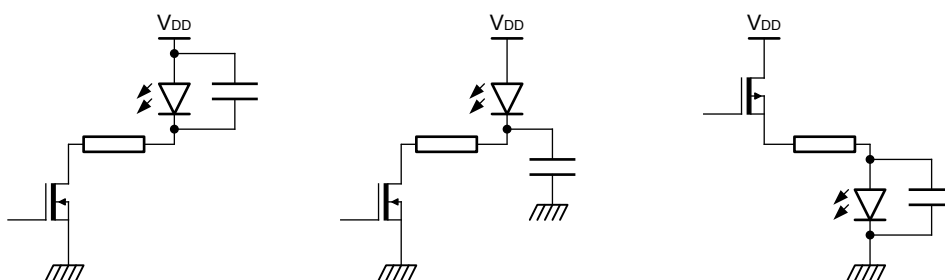


Table 11. Led rules documentation gives a list of documents containing information about led rules.

Table 11. Led rules documentation

Id	Title	Chapters
AN4312	Guidelines for designing touch sensing applications with surface sensors	<ul style="list-style-type: none"> LEDs and Sensors. Placing of LEDs close to sensor

5.6.2 Electrode not located on PCB

It is possible but it is not recommended, because when the electrode it isn't located on PCB, the sensitivity decreases and additional extra parasitic capacitances are added.

Figure 14. Electrode not located on PCB example

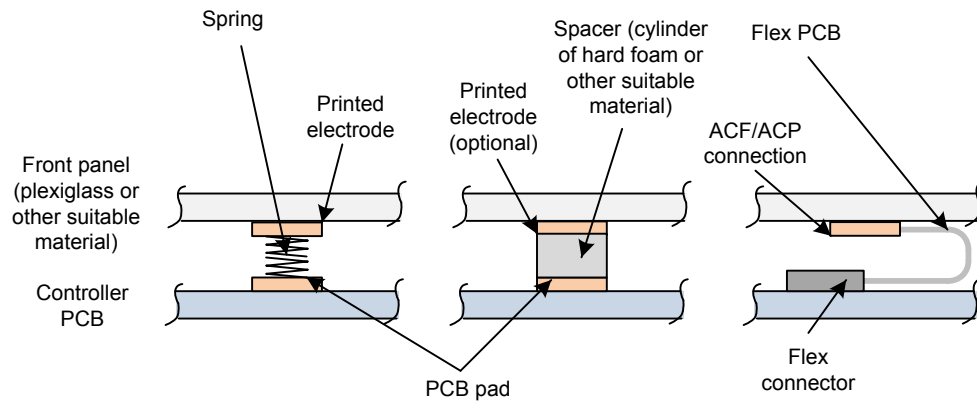


Table 12. [Electrode documentation](#) gives a list of documents containing information about the electrode .

Table 12. Electrode documentation

Id	Title	Chapters
AN4312	Guidelines for designing touch sensing applications with surface sensors	Using electrodes separated from the PCB

5.6.3 Ground, shield and sensors

Table 13 gives a list of documents containing information about the layout .

Table 13. Layout documentation

Id	Title	Chapters
AN4312	Guidelines for designing touch sensing applications with surface sensors	<ul style="list-style-type: none"> PCB and Layout Ground considerations Rotary and linear sensor recommendations

Figure 15 shows the ground plane and the signal tracks.

Figure 15. Hatched ground and signal tracks

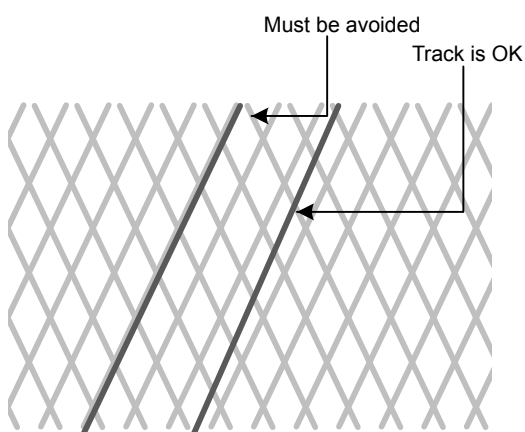


Figure 16. Ground plane example

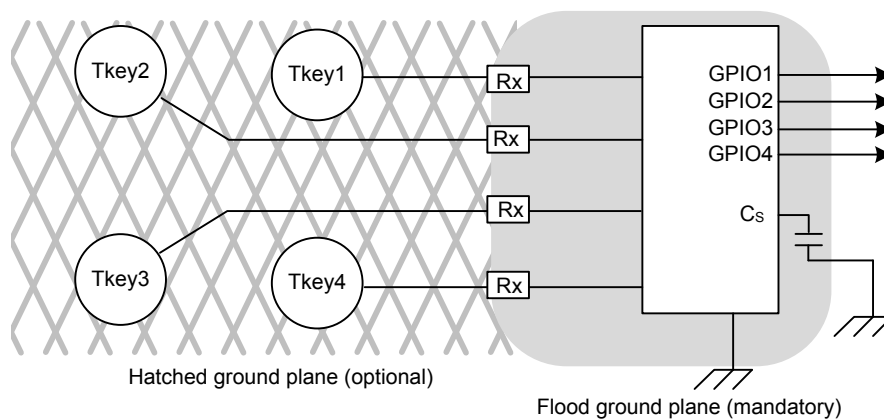


Figure 17. Track routing

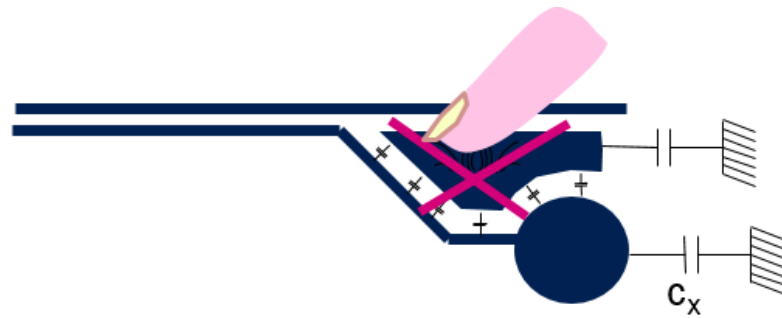


Figure 18. Track routing recommendation

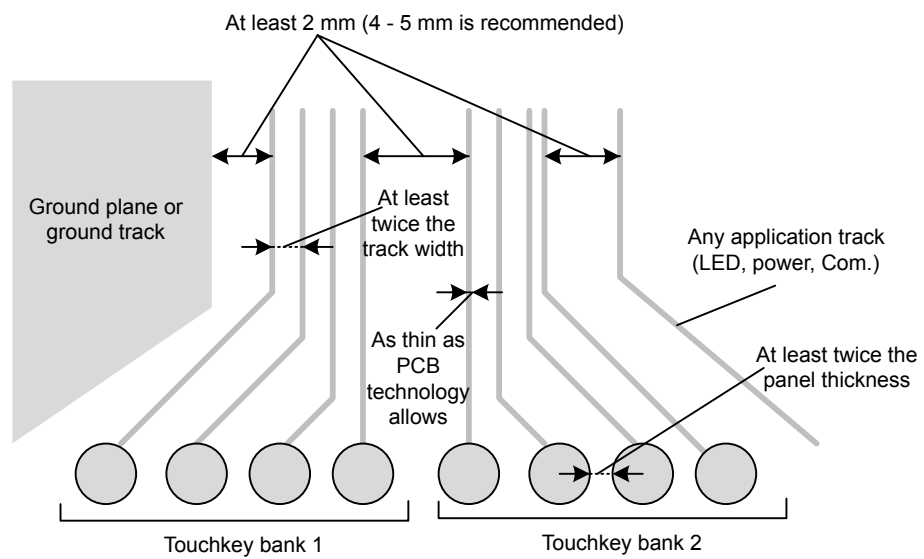
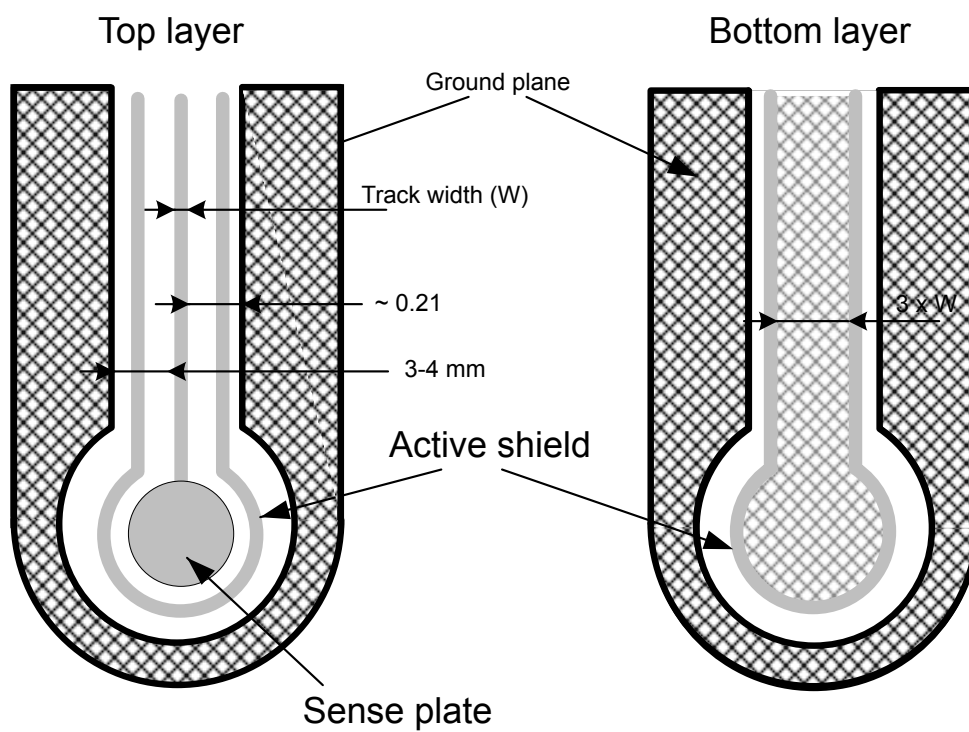


Figure 19. Shield



5.6.4

FAQ

System keys points:

- Direct connection between earth and board ground is required to avoid conducted noise issues.
- Conductive painting on the front panel must be avoid.
- Robust mechanical assembly is required.

Layout keys points:

- GND plane is mandatory under MCU, sampling capacitors and up to serial resistors
- Hatched GND plane recommended for sensor traces from both sides of the PCB:
 - minimize parasitic capacitance
 - mesh plane possible with 25% to 40% copper
- Route the sensors and ground on the same layer while the components and other tracks are routed on the other layers

Driven shield, or Active shield, is recommended.

- If there will be LEDs close to sensors, to indicate a touch event, they must be bypassed by a capacitor. typically 10 nF.
- External LDO regulator should be used to power the MCU only to provide a stable power supply voltage without any ripple, especially all the switching components like transistors, LEDs, in the application mustn't be powered from the same voltage. This regulator should not be placed close to the sensors and their tracks, but close to MCU.
- It is strongly recommended to dedicate pins to be used as touch sensors and do not share them with other features

R_S and C_S keys points:

- PPS or NPO sampling capacitors are recommended. Possible X5R or X7R.
- Never use tantalium sampling capacitors.
- Serial ESD 10 K (down to 1 K) resistors are recommended to be placed as close as possible to the MCU
- No track crossing or via between these resistors and the MCU
- The value of sampling capacitor of active shield should be different than the value of the sampling capacitors used for acquisition.
- The capacitance of active shield is higher (larger area) than C_X of a single touch sensing channel. In order to achieve the same waveform on active shield and active touch sensing channel, the ratios C_S/C_X of active shield and active touch sensing channel (touchkey). therefore, the CS of the active shield should also have higher value ($k \times C_S$ of touch sensing channel).

Sensor key points:

- Other traces must not cross the touch sensing traces or the whole touch sensing area
- The touch sensing traces should be as thin as technology allows and as short as possible.
 - No longer than 10 cm
- The space between traces and GND plane should be ideally 5 mm
- TC pins are more robust against external interference than FT:
- Consider modification of PCB layout to allow connection of external VDD clamping diode to touch sensing electrode traces.
 - Use low-capacitance diode like BAR18, BAS70 with $C_{max} = 2$ pF.
 - In case it is later needed, add pads and connection to the PCB without assembling components.
- Floating panes must never be placed close to the sensors.

5.7 Noise

Noise is a key point for touch sensing applications. Noise can come from Power supply.

5.7.1 Power supply

Main rules to follow:

- Place Buzzer and LED before LDO.
- Place LDO close to MCU.

Figure 20. Typical power supply schematic

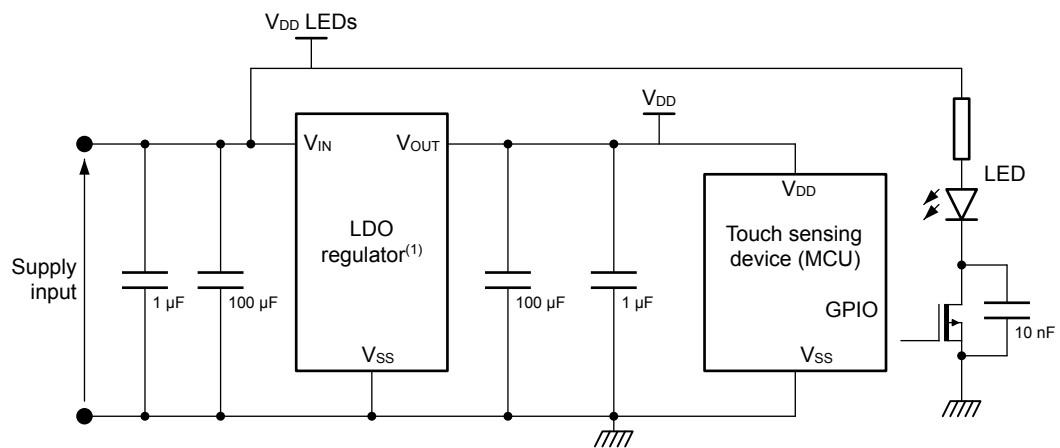


Table 14. Power supply documentation gives a list of documents containing information about the power supply.

Table 14. Power supply documentation

Id	Title	Chapters
AN4312	Guidelines for designing touch sensing applications with surface sensors	Power supply

5.7.2 False detection

To avoid false detection TSL embed ECS, DXS and DTO algorithms.

Table 15. False detection documentation gives a list of documents containing information about the false detection.

Table 15. False detection documentation

Id	Title	Chapters
UM1913	Developing applications on STM32Cube with STMTouch® touch sensing library	<ul style="list-style-type: none"> • Environment Change System (ECS) <ul style="list-style-type: none"> – Power supply voltage, temperature and air humidity • Detection Exclusion System (DXS) • Detection Time Out (DTO)

5.7.3 Noise immunity

Noise filtering can be done on hardware and software (TSL) sides.

[Table 16. Noise immunity documentation](#) gives a list of documents containing information about the noise immunity.

Table 16. Noise immunity documentation

Id	Title	Chapters
UM1913	Developing applications on STM32Cube with STMTouch® touch sensing library	Noise filters
AN4299	Guidelines to improve conducted noise robustness on STM32F0/F3/L0/L4 series touch sensing applications	How to improve noise immunity
OLT	STM32L4 Online Training	Touch sensing controller (TSC)

5.7.4 Conducted noise

- Touch sensing systems requires the conducted noise immunity.
- A key point is the signal to noise ratio (SNR).
- The test condition to be followed by the user is described in the standard IEC61000-4-6.

[Table 17. Conducted noise documentation](#) gives a list of documents containing information about the conducted noise.

Table 17. Conducted noise documentation

Id	Title	Chapters
AN4299	Guidelines to improve conducted noise robustness on STM32F0/F3/L0/L4 series touch sensing applications	All chapters

6 Tuning

For tuning purpose dedicated application note are available.

Sensors

[Table 18. Sensors documentation](#) gives a list of documents containing information about the sensor.

Table 18. Sensors documentation

Id	Title	Chapters
AN4312	Guidelines for designing touch sensing applications with surface sensors	All chapters

ESD

[Table 19. ESD documentation](#) gives a list of documents containing information about the ESD

Table 19. ESD documentation

Id	Title	Chapters
AN3960	ESD considerations for touch sensing applications	All chapters

CN

[Table 20. Conducted noise documentation](#) gives a list of documents containing information about the conducted noise.

Table 20. Conducted noise documentation

Id	Title	Chapters
AN4299	Guidelines to improve conducted noise robustness on STM32F0/F3/L0/L4 series touch sensing applications	All chapters

CS

[Table 21. Sampling capacitor documentation](#) gives a list of documents containing information about the sampling capacitor.

Table 21. Sampling capacitor documentation

Id	Title	Chapters
AN4310	Sampling capacitor selection guide for MCU based touch sensing applications	All chapters

7 Getting started TSC with STM32CubeMX

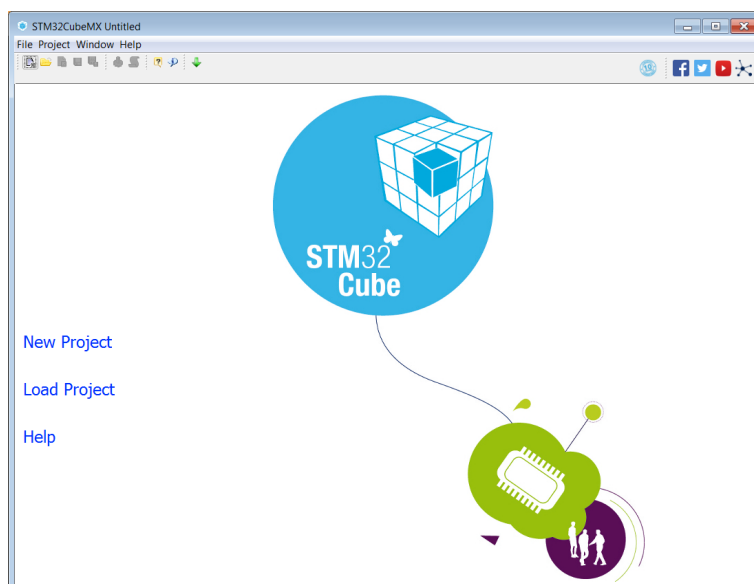
7.1 Uses cases

How to set-up an TSC application based on TSL is explained in the following two examples. These examples describe the way to set-up TLS on STM32F072B-DISCO and STM32L0538-DISCO discovery boards.

This description can be used as example to set-up others TSC series such us L4, F3, L0, L1 and L4.

An STM32CubeMX new feature is available from version 4.24.0. This new feature can help to speed-up TSL, TouchSensingLib, installation.

Figure 21. Main project panel



7.2 Discovery board: STM32F072B-DISCO

The STM32F072 Discovery kit helps the user to discover the STM32F072, which has the full set of features available in the STM32F0 Series, and to develop his applications easily. It includes everything required for beginners and experienced users to get started quickly.

Based on the STM32F072RBT6, it includes an ST-LINK/V2 embedded debug tool interface, an ST MEMS gyroscope, LEDs, push-buttons, linear touch sensor, RF EEPROM connector and a USB mini-B connector.

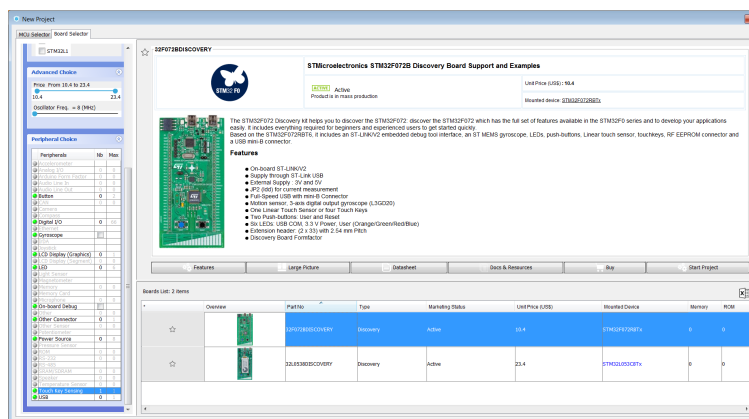
This discovery board provide a three channels linear (or slider) sensor. the main characteristics of these sensor are:

- On-board ST-LINK/V2
- Supply through ST-Link USB
- External Supply: 3V and 5V
- JP2 (Idd) for current measurement
- Full-Speed USB with mini-B Connector
- Motion sensor, 3-axis digital output gyroscope (L3GD20)
- One Linear Touch Sensor or four Touch Keys
- Two Push-buttons: User and Reset
- Six LEDs: USB COM, 3.3 V Power, User (Orange/Green/Red/Blue)
- Extension header: (2 x 33) with 2.54 mm Pitch
- Discovery Board Formfactor

7.2.1 STM32F072B-DISCO board selection

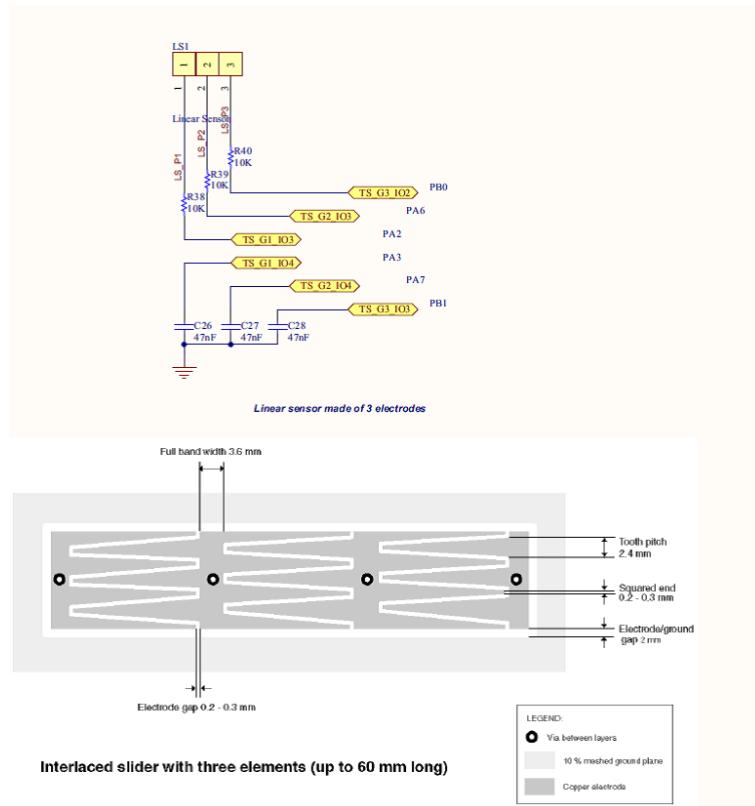
Start to select STM32F072B-DISCO board.

Figure 22. STM32F072B-DISCO board selection



To start Linear Touch Sensor channel acquisition at the same time, three groups are used. (See Figure 23. STM32F072B-DISCO board schematics

Figure 23. STM32F072B-DISCO board schematics



7.2.2

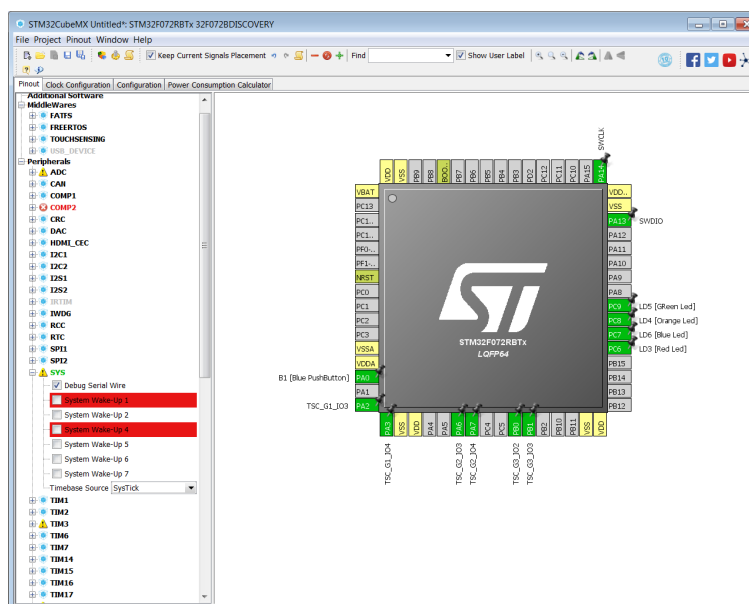
STM32F072B-DISCO TSC group and sensor activation

To activate the TSC group, sampling capacitors and sensor channels follows the below steps:

- activate TSC according schematics information.
- deactivate unrelevant peripheral like USB, SPI, NCF(L0), EPaper(L0), MFX(L0)

SWD peripheral must be set according to [Figure 24](#)

Figure 24. STM32F072B-DISCO pinout SWD



TSC peripheral must be set according to [Figure 25](#).

Figure 25. STM32F072B-DISCO pinout TSC

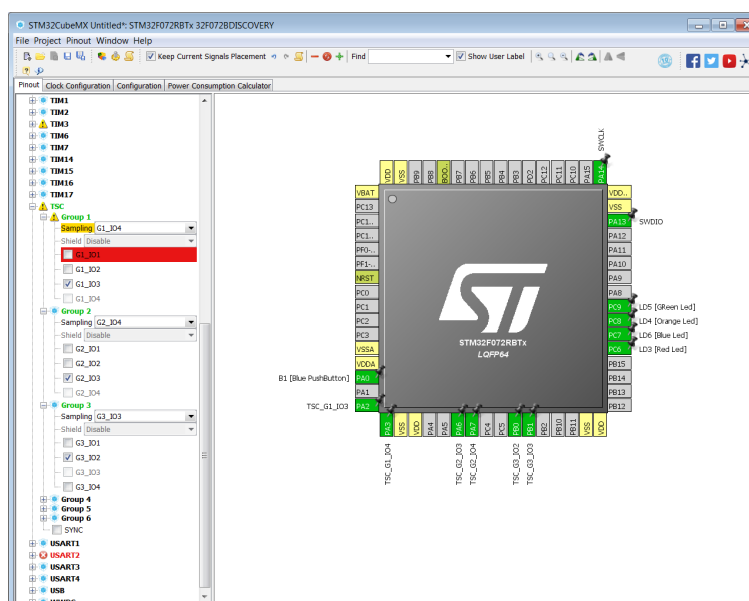
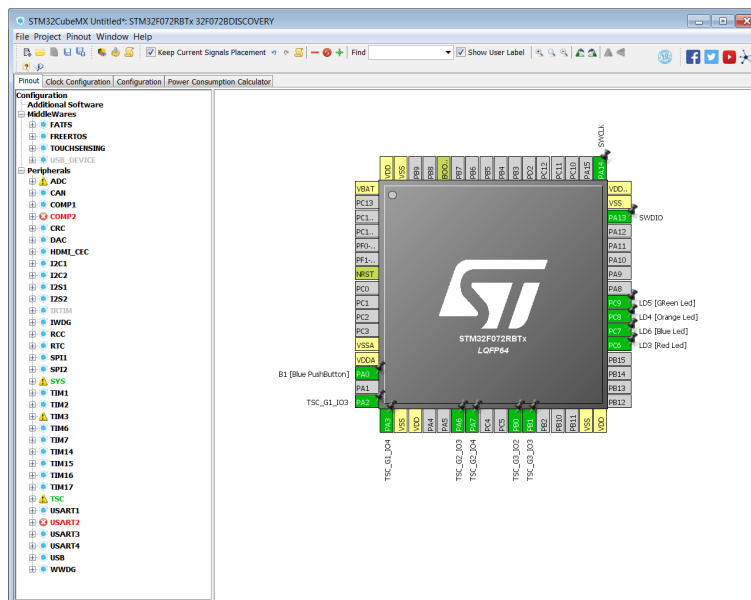


Figure 26 shows the results obtained.

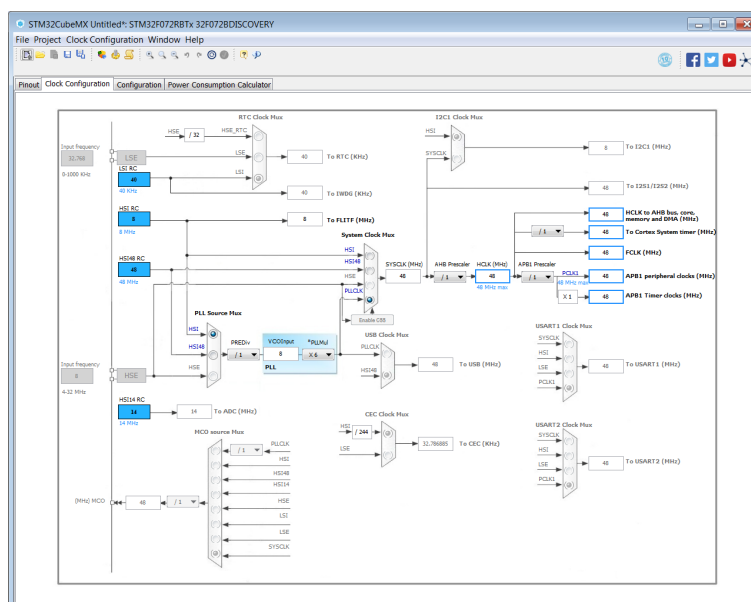
Figure 26. STM32F072B-DISCO pinout overview



7.2.3 STM32F072B-DISCO clock tree

It uses the default clock tree setting.

Figure 27. STM32F072B-DISCO clock configuration

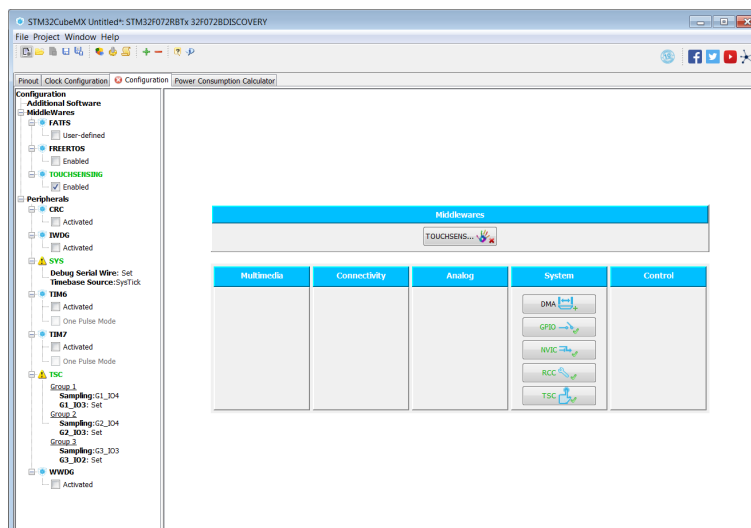


7.2.4

STM32F072B-DISCO touchsensing library

To activate the TLS usage, switch on TOUCHSENSING box configuration.

Figure 28. TOUCHSENSING box configuration



Select three channels Linear slider and assign dedicated Gx_I Oy (see previous chapter or schematics for details).

- For training purpose, we can use three channels Linear slider as three keys sensors.
- Select three keys and assign dedicated Gx_I Oy (see previous chapter or schematics for details).

Figure 29 to Figure 33 show these steps.

Figure 29. STM32F072B-DISCO sensor selection

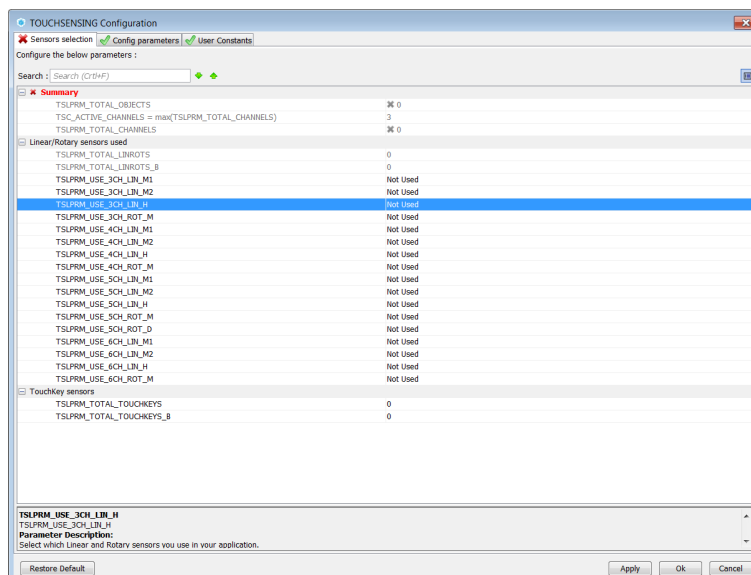


Figure 30. STM32F072B-DISCO sensor selection step2

TOUCHSENSING Configuration

Sensors selection ☒ Config parameters ☒ User Constants

Configure the below parameters :

Search Search (Ctrl+F)

Summary

TSUPRM_TOTAL_OBJECTS	36 0
TSC_ACTIVE_CHANNELS = max(TSUPRM_TOTAL_CHANNELS)	3
TSUPRM_TOTAL_CHANNELS	36 0
Linear/Rotary sensors used	
TSUPRM_TOTAL_LINROTS	0
TSUPRM_TOTAL_LINROTS_B	0
TSUPRM_USE_3CH_LIN_M1	Not Used
TSUPRM_USE_3CH_LIN_M2	Not Used
TSUPRM_USE_3CH_LIN_H	Used
TSUPRM_USE_3CH_LIN_H_NBR	0
TSUPRM_USE_3CH_LIN_H_B_NBR	0
TSUPRM_USE_3CH_ROT_M	Not Used
TSUPRM_USE_4CH_LIN_M1	Not Used
TSUPRM_USE_4CH_LIN_M2	Not Used
TSUPRM_USE_4CH_LIN_H	Not Used
TSUPRM_USE_4CH_ROT_M	Not Used
TSUPRM_USE_5CH_LIN_M1	Not Used
TSUPRM_USE_5CH_LIN_M2	Not Used
TSUPRM_USE_5CH_LIN_H	Not Used
TSUPRM_USE_5CH_ROT_M	Not Used
TSUPRM_USE_6CH_LIN_M1	Not Used
TSUPRM_USE_6CH_LIN_M2	Not Used
TSUPRM_USE_6CH_LIN_H	Not Used
TSUPRM_USE_6CH_ROT_M	Not Used
Touchkey sensors	
TSUPRM_TOTAL_TOUCHKEYS	0
TSUPRM_TOTAL_TOUCHKEYS_B	0

TSUPRM_USE_3CH_LIN_H
 TSUPRM_USE_3CH_LIN_H
 Parameter Description:
 Select which Linear and Rotary sensors you use in your application.

Restore Default Apply Ok Cancel

Figure 31. STM32F072B-DISCO sensor selection step3

TOUCHSENSING Configuration

Sensors selection ☒ Config parameters ☒ User Constants

Configure the below parameters :

Search Search (Ctrl+F)

Summary

TSUPRM_TOTAL_OBJECTS	1
TSC_ACTIVE_CHANNELS = max(TSUPRM_TOTAL_CHANNELS)	3
TSUPRM_TOTAL_CHANNELS	3
Linear/Rotary sensors used	
TSUPRM_TOTAL_LINROTS	1
TSUPRM_TOTAL_LINROTS_B	0
TSUPRM_USE_3CH_LIN_M1	Not Used
TSUPRM_USE_3CH_LIN_M2	Not Used
TSUPRM_USE_3CH_LIN_H	Used
TSUPRM_USE_3CH_LIN_H_NBR	1
TSUPRM_USE_3CH_LIN_H_NBR1_CH1	G1_X03
TSUPRM_USE_3CH_LIN_H_NBR1_CH2	G2_X03
TSUPRM_USE_3CH_LIN_H_NBR1_CH3	G3_X02
TSUPRM_USE_3CH_LIN_H_B_NBR	0
TSUPRM_USE_3CH_ROT_M	Not Used
TSUPRM_USE_4CH_LIN_M1	Not Used
TSUPRM_USE_4CH_LIN_M2	Not Used
TSUPRM_USE_4CH_LIN_H	Not Used
TSUPRM_USE_4CH_ROT_M	Not Used
TSUPRM_USE_5CH_LIN_M1	Not Used
TSUPRM_USE_5CH_LIN_M2	Not Used
TSUPRM_USE_5CH_LIN_H	Not Used
TSUPRM_USE_5CH_ROT_M	Not Used
TSUPRM_USE_6CH_LIN_M1	Not Used
TSUPRM_USE_6CH_LIN_M2	Not Used
TSUPRM_USE_6CH_LIN_H	Not Used
TSUPRM_USE_6CH_ROT_M	Not Used
Touchkey sensors	
TSUPRM_TOTAL_TOUCHKEYS	0
TSUPRM_TOTAL_TOUCHKEYS_B	0

TSUPRM_USE_3CH_LIN_H_NBR1_CH1
 TSUPRM_USE_3CH_LIN_H_NBR1_CH1
 * This parameter has automatically changed after your last modification.

Restore Default Apply Ok Cancel

Figure 32. STM32F072B-DISCO sensor selection step4

TOUCHSENSING Configuration

Sensors selection: ☒ Config parameters: ☒ User Constants: ☒

Configure the below parameters:

Search:

TSUPRM_TKEY_DETECT_INL_TH	120
TSUPRM_TKEY_DETECT_OUT_TH	110
TSUPRM_TKEY_CALB_TH	120
TSUPRM_COEFF_TH	0
Thresholds for Linear and Rotary sensors	
TSUPRM_LINROT_PROX_INL_TH	10
TSUPRM_LINROT_PROX_OUT_TH	5
TSUPRM_LINROT_DETECT_INL_TH	80
TSUPRM_LINROT_DETECT_OUT_TH	75
TSUPRM_LINROT_CALB_TH	80
TSUPRM_LINROT_USE_NORMDELTA	0
Linear/Rotary sensors position	
TSUPRM_LINROT_RESOLUTION	7
TSUPRM_LINROT_DIR_CHG_POS	10
TSUPRM_LINROT_DIR_CHG_DEB	1
Debounce counters	
TSUPRM_DEBOUNCE_PROX	2
TSUPRM_DEBOUNCE_DETECT	2
TSUPRM_DEBOUNCE_RELEASE	2
TSUPRM_DEBOUNCE_CALB	3
TSUPRM_DEBOUNCE_ERROR	3
Environment Change System (ECS)	
TSUPRM_ECS_K_SLOW	10
TSUPRM_ECS_K_FAST	20
TSUPRM_ECS_DELAY	500
Detection Time Out (DTO)	
TSUPRM DTO	0
Detection Exclusion System (DXS)	
TSUPRM_USE_DXS	0
Miscellaneous parameters	
TSUPRM_TXL_FREQ	1000
TSUPRM_DELAY_DISCHARGE_ALL	1000
TSUPRM_DDEFF	TSC_DDEFF_OUT_PP_LOW

TSUPRM_LINROT_RESOLUTION
TSUPRM_LINROT_RESOLUTION must be between 1 and 8.
Parameter Description:
Position resolution in number of bits (range=1..8)

Restore Default Apply Ok Cancel

Figure 33. STM32F072B-DISCO sensor selection step5

TOUCHSENSING Configuration

Sensors selection: ☒ Config parameters: ☒ User Constants: ☒

Configure the below parameters:

Search:

Version and modes	
TouchSensing version	2.2.0
Optional features	
TSUPRM_USE_MEAS	1
TSUPRM_USE_PROX	1
TSUPRM_USE_ZONE	0
Acquisition limits	
TSUPRM_ACQ_MIN	10
TSUPRM_ACQ_MAX	TSC_MCV_B191
Calibration	
TSUPRM_CALB_SAMPLES	4
TSUPRM_CALB_DELAY	0
Thresholds for Touchkey sensors	
TSUPRM_TKEY_PROX_INL_TH	10
TSUPRM_TKEY_PROX_OUT_TH	5
TSUPRM_TKEY_DETECT_INL_TH	120
TSUPRM_TKEY_DETECT_OUT_TH	110
TSUPRM_TKEY_CALB_TH	120
TSUPRM_COEFF_TH	0
Thresholds for Linear and Rotary sensors	
TSUPRM_LINROT_PROX_INL_TH	10
TSUPRM_LINROT_PROX_OUT_TH	5
TSUPRM_LINROT_DETECT_OUT_TH	35
TSUPRM_LINROT_CALB_TH	40
TSUPRM_LINROT_USE_NORMDELTA	0
Linear/Rotary sensors position	
TSUPRM_LINROT_RESOLUTION	7
TSUPRM_LINROT_DIR_CHG_POS	10
TSUPRM_LINROT_DIR_CHG_DEB	1
Debounce counters	
TSUPRM_DEBOUNCE_PROX	2
TSUPRM_DEBOUNCE_DETECT	2
TSUPRM_DEBOUNCE_RELEASE	2

TSUPRM_LINROT_DETECT_INL_TH
TSUPRM_LINROT_DETECT_INL_TH must be between 0 and 255.
Parameter Description:
Linear/Rotary Detect state input threshold (range=0..255)

Restore Default Apply Ok Cancel

7.2.5 STM32F072B-DISCO software project generation

Now, it is possible to generate the complete software project based on TSC HAL and TSL.

Figure 34 to Figure 37 show all these steps.

Figure 34. STM32F072B-DISCO software generation step1

Project Settings

Project Name: STM32F072B-DISCO-M4.24.0

Project Location: C:\Users\richardo\Desktop\TSC_Evolution

Toolchain Folder Location: C:\Users\richardo\Desktop\TSC_Evolution\STM32F072B-DISCO-M4.24.0\

Toolchain / IDE: EWARM

Linker Settings

Minimum Heap Size: 0x200

Minimum Stack Size: 0x400

Mcu and Firmware Package

Mcu Reference: STM32F072RBTx

Firmware Package Name and Version: STM32Cube_FW_F0_V1.9.0

Use Default Firmware Location: ☒

C:\Users\richardo\STM32Cube\Repository\STM32Cube_FW_F0_V1.9.0

Figure 35. STM32F072B-DISCO software generation step2

Project Settings

STM32Cube Firmware Library Package

☒ Copy all used libraries into the project folder

☐ Copy only the necessary library files

☐ Add necessary library files as reference in the toolchain project configuration file

Generated files

☐ Generate peripheral initialization as a pair of '.c/.h' files per peripheral

☐ Backup previously generated files when re-generating

☒ Keep User Code when re-generating

☒ Delete previously generated files when not re-generated

HAL Settings

☒ Set all free pins as analog (to optimize the power consumption)

☐ Enable Full Assert

Template Settings

Select a template to generate customized code

Settings...

Figure 36. STM32F072B-DISCO software generation step3

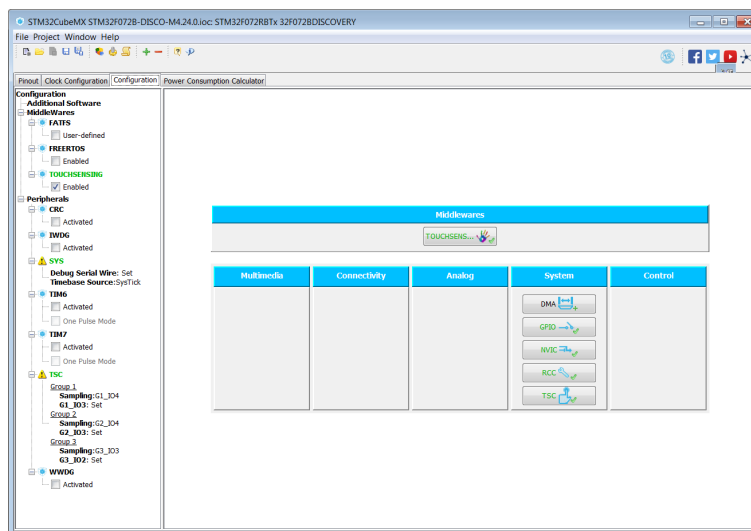
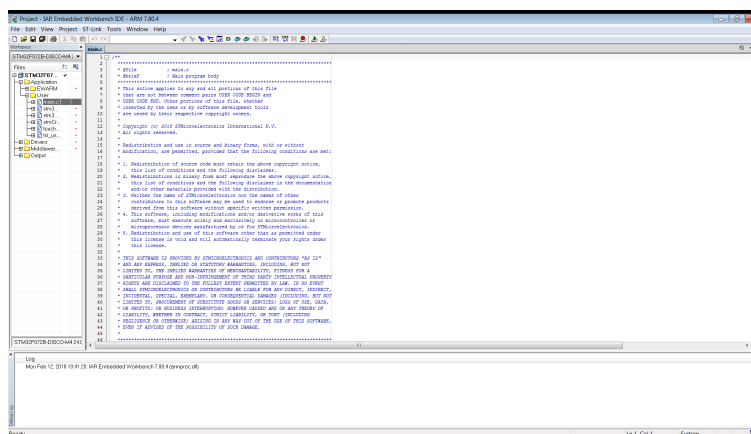


Figure 37. STM32F072B-DISCO IDE workspace



7.2.6

STM32F072B-DISCO software basic algorithm

The user needs now to write the main application loop.

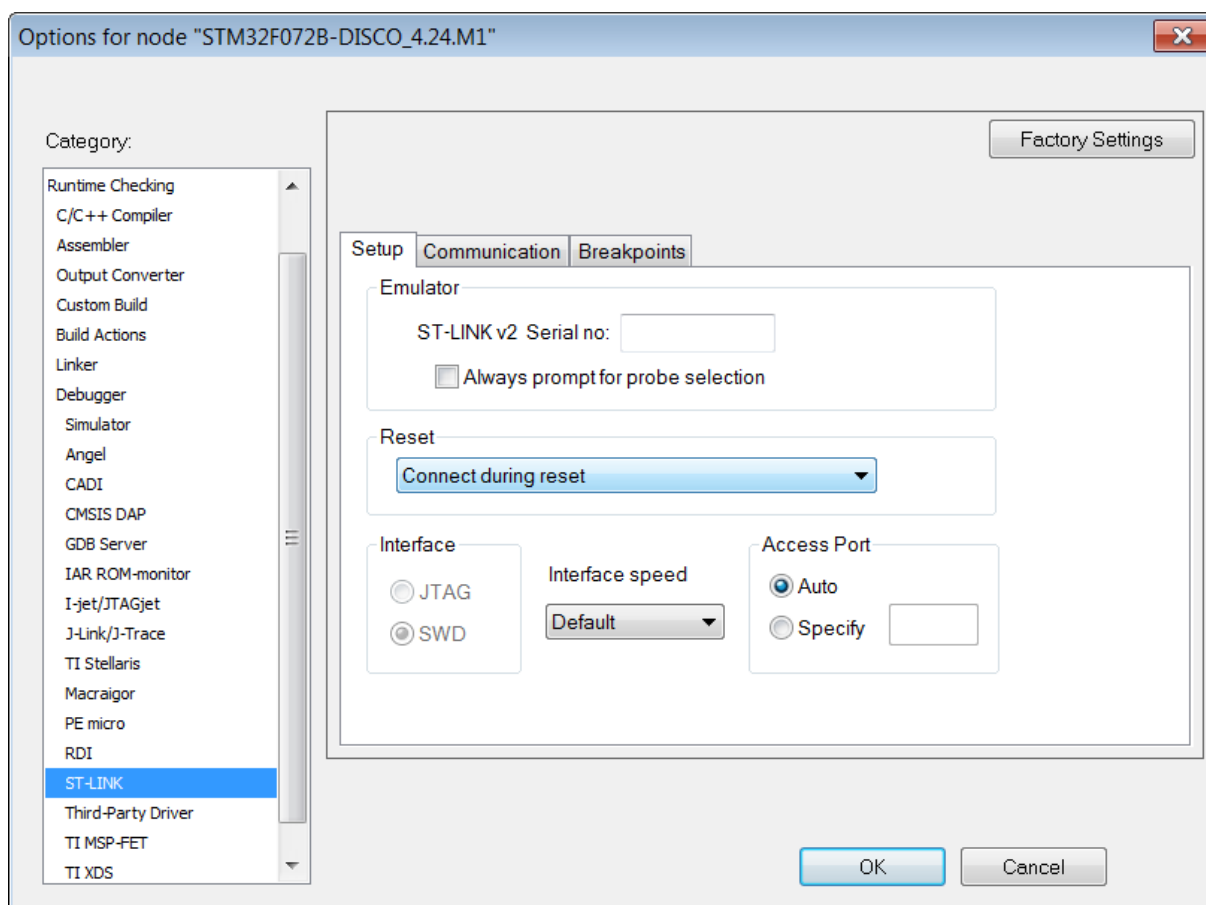
Example to show keys usage instead of slider usage.

- Open your IDE and in main.c file add the following lines:

```
/* USER CODE BEGIN 3 */
extern TSL_LinRot_T MyLinRots[];
static uint32_t cnt=0;
tsl_user_status_t status = TSL_USER_STATUS_BUSY;
status = tsl_user_Exec();
if(TSL_USER_STATUS_BUSY == status)
{
    // Nothing to do
    if(cnt++%50==0){
        HAL_GPIO_TogglePin(LD3_GPIO_Port, LD3_Pin);
    }
    HAL_Delay(1);
}
else
{
    if(MyLinRots[0].p_Data->StateId == TSL_STATEID_DETECT)
    {
        //TSLPRM_LINROT_RESOLUTION
        if(MyLinRots[0].p_Data->Position >= 5 && MyLinRots[0].p_Data->Position < 50)
        {
            HAL_GPIO_WritePin(LD4_GPIO_Port, LD4_Pin, GPIO_PIN_SET);
            HAL_GPIO_WritePin(LD6_GPIO_Port, LD6_Pin, GPIO_PIN_RESET);
            HAL_GPIO_WritePin(LD5_GPIO_Port, LD5_Pin, GPIO_PIN_RESET);
        }
        if(MyLinRots[0].p_Data->Position >= 50 && MyLinRots[0].p_Data->Position < 80)
        {
            HAL_GPIO_WritePin(LD6_GPIO_Port, LD6_Pin, GPIO_PIN_SET);
            HAL_GPIO_WritePin(LD4_GPIO_Port, LD4_Pin, GPIO_PIN_RESET);
            HAL_GPIO_WritePin(LD5_GPIO_Port, LD5_Pin, GPIO_PIN_RESET);
        }
        if(MyLinRots[0].p_Data->Position >= 80 && MyLinRots[0].p_Data->Position < 120)
        {
            HAL_GPIO_WritePin(LD5_GPIO_Port, LD5_Pin, GPIO_PIN_SET);
            HAL_GPIO_WritePin(LD4_GPIO_Port, LD4_Pin, GPIO_PIN_RESET);
            HAL_GPIO_WritePin(LD6_GPIO_Port, LD6_Pin, GPIO_PIN_RESET);
        }
    }
    else //if(MyLinRots[0].p_Data->StateId == TSL_STATEID_RELEASE)
    {
        HAL_GPIO_WritePin(LD4_GPIO_Port, LD4_Pin, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(LD5_GPIO_Port, LD5_Pin, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(LD6_GPIO_Port, LD6_Pin, GPIO_PIN_RESET);
    }
}
}
/* USER CODE END 3 */
```

Take care of ST-Link setup, see [Figure 38. STM32F072B-DISCO setup](#).

Figure 38. STM32F072B-DISCO setup



Now the system is functional and ready to be used. Led will blink according finger position on slider.

7.3 Discovery board: STM32L0538-DISCO

The STM32L053 discovery kit helps you to discover the ultra-low-power microcontrollers of the STM32L0 series. It offers everything required for beginners and experienced users to get started quickly and develop applications easily.

Based on an STM32L053C8T6, it includes an ST-LINK/V2-1 embedded debug tool interface, linear touch sensor, IDD current measurement, 2.04" E-paper display, NFC connector for PLUG-CR95HF-B board, LEDs, pushbuttons and a USB Mini-B connector.

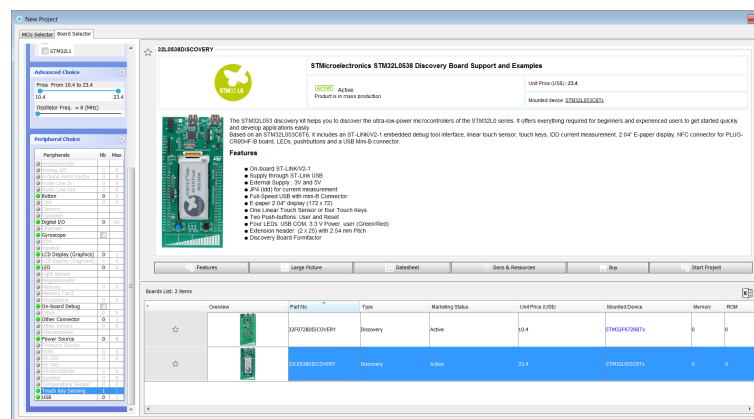
This discovery board provide a three channels linear (or slider) sensor. Their main characteristics are:

- On-board ST-LINK/V2-1
- Supply through ST-Link USB
- External Supply : 3V and 5V
- JP4 (Idd) for current measurement
- Full-Speed USB with mini-B Connector
- E-paper 2.04" display (172 x 72)
- One Linear Touch Sensor or four Touch Keys
- Two Push-buttons: User and Reset
- Four LEDs: USB COM, 3.3 V Power, user (Green/Red)
- Extension header: (2 x 25) with 2.54 mm Pitch
- Discovery Board Formfactor

7.3.1 STM32L0538-DISCO board selection

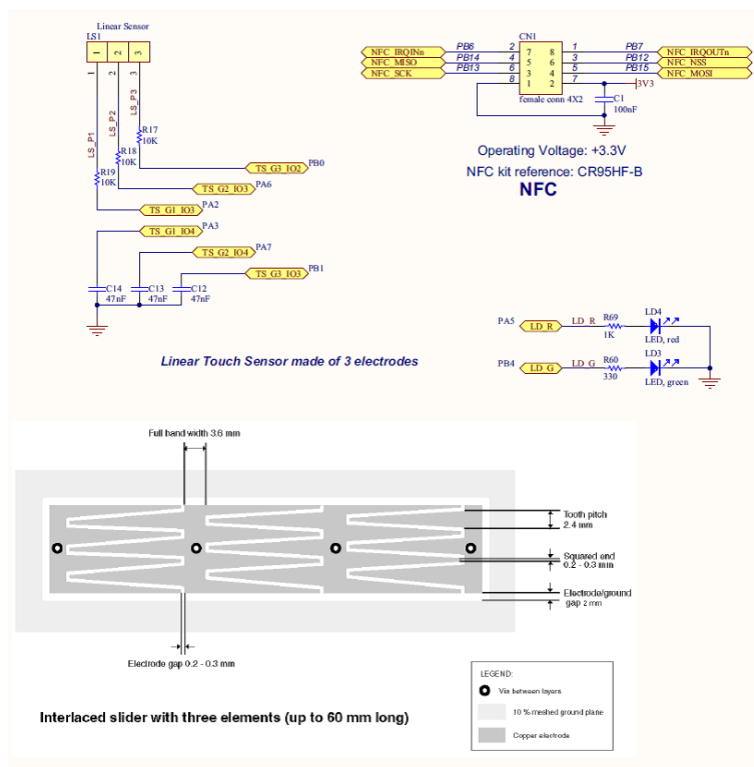
Start to select STM32L0538-DISCO board.

Figure 39. STM32L0538-DISCO board selection



To start linear touch sensor channel acquisition at the same time, three groups are used.

Figure 40. STM32L0538-DISCO board schematics



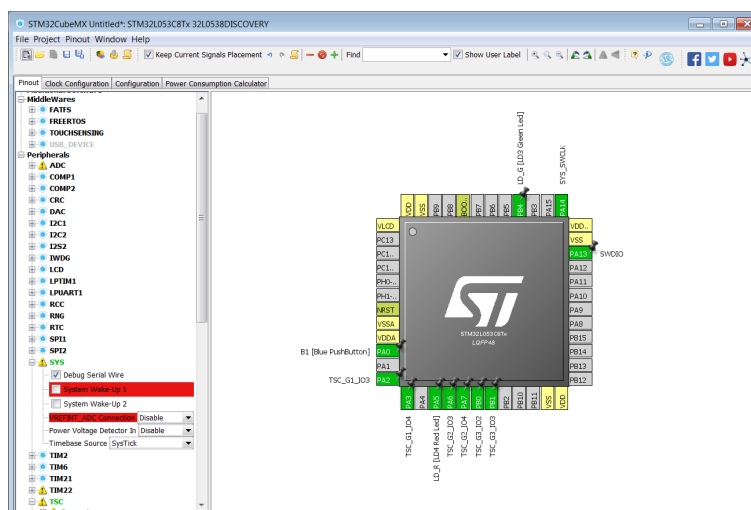
7.3.2

STM32L0538-DISCO TSC group and sensor activation

To activate the TSC group, sampling capacitors and sensor channels follows the below steps:

- Activate TSC according schematics information.
 - You can deactivate irrelevant peripheral like USB, SPI, NCF(L0), EPaper(L0), MFX(L0)
- SWD peripheral must be set according to [Figure 41](#).

Figure 41. Pinout SWD



TSC peripheral must be set according to [Figure 42](#).

Figure 42. Pinout TSC

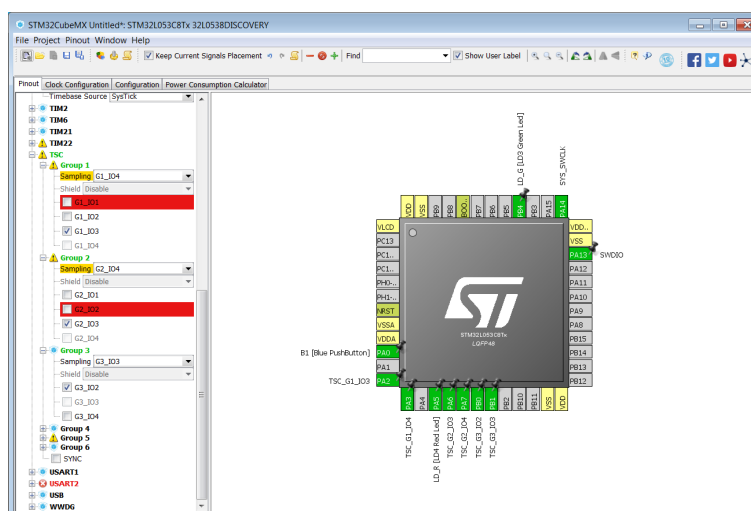
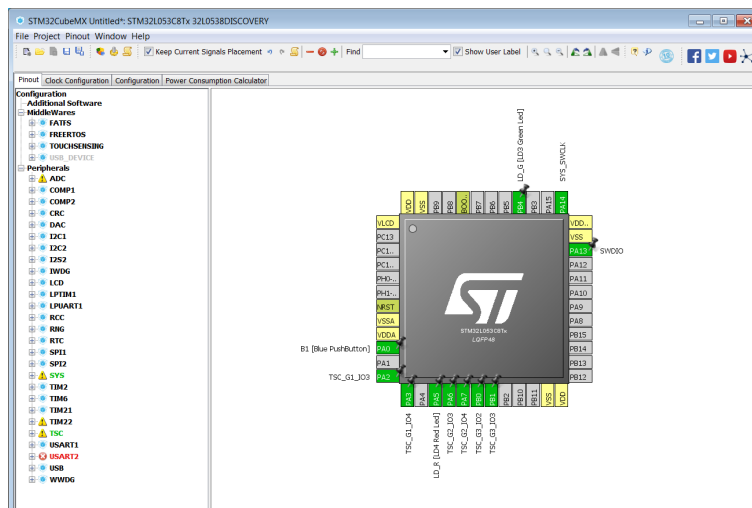


Figure 43 shows the results obtained.

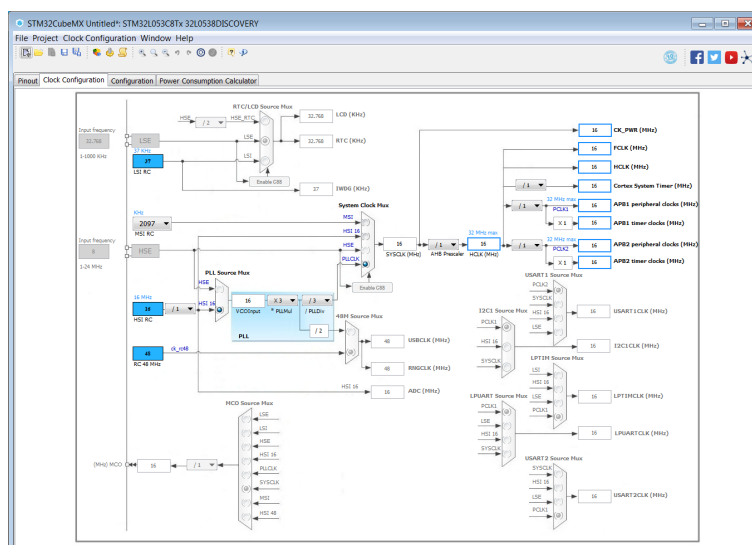
Figure 43. Pinout overview



7.3.3 STM32L0538-DISCO clock tree

It uses the default clock tree setting.

Figure 44. Clock configuration

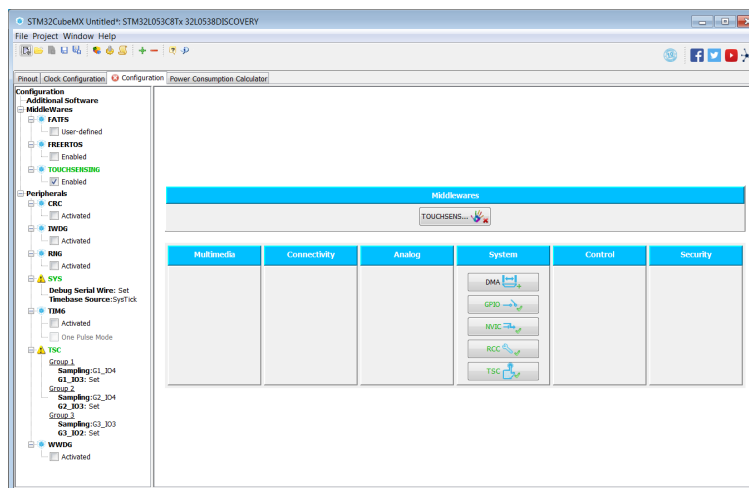


7.3.4

STM32L0538-DISCO touchsensing library

To activate the TLS usage, switch on TOUCHSENSING box configuration.

Figure 45. TOUCHSENSING box configuration



Select 3 channels Linear slider and assign dedicated Gx_I/Oy (see previous chapter or schematics for details).

For training purpose, the user can:

- use three channels linear slider as three keys sensors
- Select three keys and assign dedicated Gx_I/Oy (see previous chapter or schematics for details).

Follow Figure 46 to Figure 50 to set sensors.

Figure 46. STM32L0538-DISCO sensor selection step1

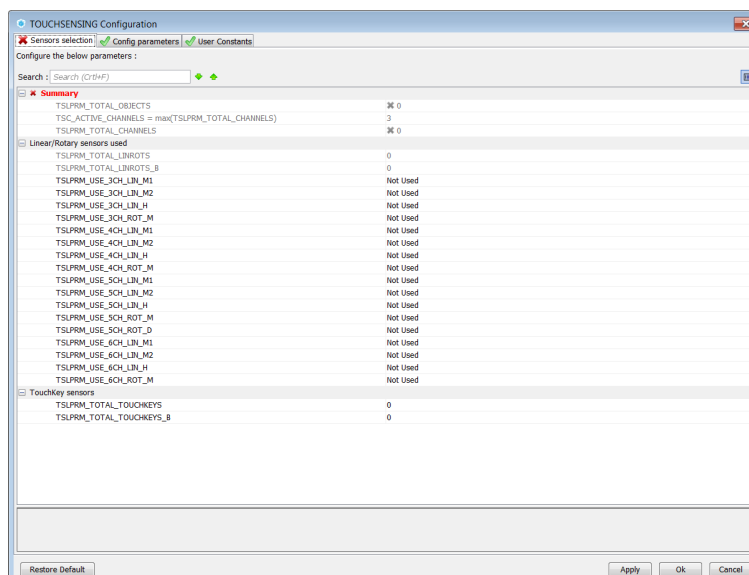


Figure 47. STM32L0538-DISCO sensor selection step2

Figure 48. STM32L0538-DISCO sensor selection step3

Figure 49. STM32L0538-DISCO sensor selection step4

Project Settings

Project Name: STM32L0538-DISCO-4.24.0

Project Location: C:\Users\richardo\Desktop\TSC_Evolution\

Toolchain Folder Location: C:\Users\richardo\Desktop\TSC_Evolution\STM32L0538-DISCO-4.24.0\

Toolchain / IDE: EWARM

Linker Settings

Minimum Heap Size: 0x200

Minimum Stack Size: 0x400

Mcu and Firmware Package

Mcu Reference: STM32L053C8Tx

Firmware Package Name and Version: STM32Cube_FW_L0_V1.10.0

☒ Use Default Firmware Location

C:/Users/richardo/STM32Cube/Repository/STM32Cube_FW_L0_V1.10.0

Ok Cancel

Figure 50. STM32L0538-DISCO sensor selection step5

Project Settings

STM32Cube Firmware Library Package

☒ Copy all used libraries into the project folder

☐ Copy only the necessary library files

☐ Add necessary library files as reference in the toolchain project configuration file

Generated files

☐ Generate peripheral initialization as a pair of '.c/.h' files per peripheral

☐ Backup previously generated files when re-generating

☒ Keep User Code when re-generating

☒ Delete previously generated files when not re-generated

HAL Settings

☒ Set all free pins as analog (to optimize the power consumption)

☐ Enable Full Assert

Template Settings

Select a template to generate customized code

Settings...

Ok Cancel

7.3.5 STM32L0538-DISCO software project generation

Now, it is possible to generate the complete software project based on TSC HAL and TSL.
See details in [Figure 51](#) to [Figure 55](#).

Figure 51. STM32L0538-DISCO software generation step1

Project Settings

Project | Code Generator | Advanced Settings

Project Settings

Project Name
STM32L0538-DISCO-4.24.0

Project Location
C:\Users\richardo\Desktop\TSC_Evolution\ Browse

Toolchain Folder Location
C:\Users\richardo\Desktop\TSC_Evolution\STM32L0538-DISCO-4.24.0\

Toolchain / IDE
EWARM ☐ Generate Under Root

Linker Settings

Minimum Heap Size
0x200

Minimum Stack Size
0x400

Mcu and Firmware Package

Mcu Reference
STM32L053C8Tx

Firmware Package Name and Version
STM32Cube_FW_L0_V1.10.0

☒ Use Default Firmware Location
C:/Users/richardo/STM32Cube/Repository/STM32Cube_FW_L0_V1.10.0 Browse

Ok Cancel

Figure 52. STM32L0538-DISCO software generation step2

Project Settings

Project | Code Generator | Advanced Settings

STM32Cube Firmware Library Package

☒ Copy all used libraries into the project folder
☐ Copy only the necessary library files
☐ Add necessary library files as reference in the toolchain project configuration file

Generated files

☐ Generate peripheral initialization as a pair of '.c/.h' files per peripheral
☐ Backup previously generated files when re-generating
☒ Keep User Code when re-generating
☒ Delete previously generated files when not re-generated

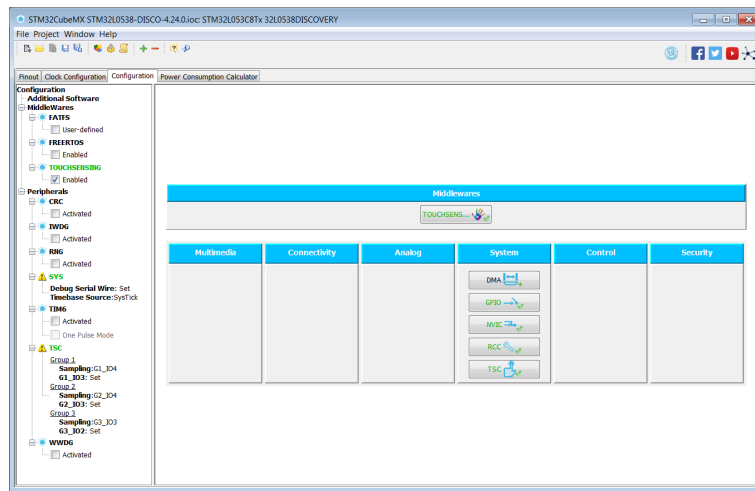
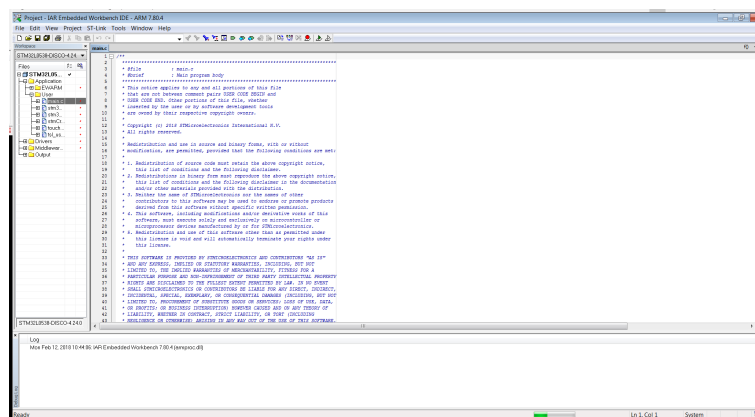
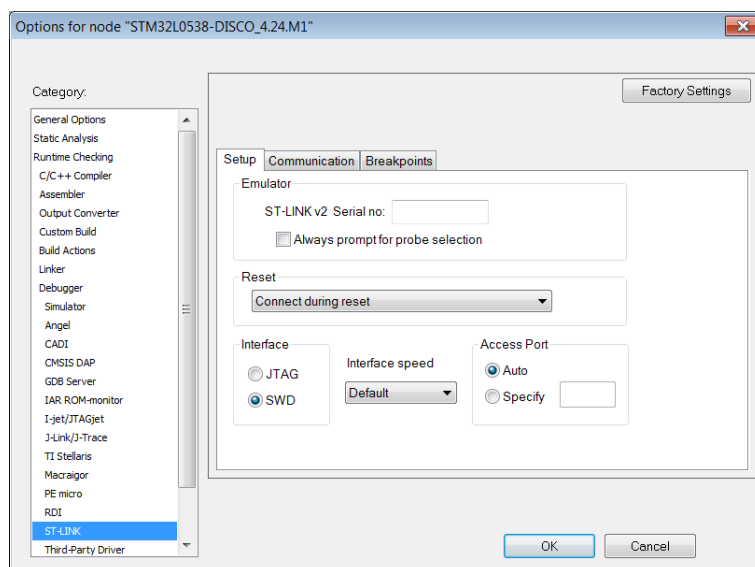
HAL Settings

☒ Set all free pins as analog (to optimize the power consumption)
☐ Enable Full Assert

Template Settings

Select a template to generate customized code Settings...

Ok Cancel

Figure 53. STM32L0538-DISCO complete project overview

Figure 54. STM32L0538-DISCO IDE workspace

Figure 55. SWD settings


7.3.6 STM32L0538-DISCO software basic algorithm

Below is showed an example to show keys usage instead of slider usage.

- Open your IDE and in main.c file add the following lines:

```
/* USER CODE BEGIN 3 */
extern TSL_TouchKey_T MyTKeys[];
static uint32_t cnt=0;
tsl_user_status_t status = TSL_USER_STATUS_BUSY;

status = tsl_user_Exec();
if(TSL_USER_STATUS_BUSY == status)
{
    // Nothing to do
    if(cnt++%50==0){
    }
    HAL_Delay(1);
}
else
{
    HAL_GPIO_WritePin(LD_R_GPIO_Port, LD_R_Pin, GPIO_PIN_RESET); //00
    HAL_GPIO_WritePin(LD_G_GPIO_Port, LD_G_Pin, GPIO_PIN_RESET);
    if(MyTKeys[0].p_Data->StateId == TSL_STATEID_DETECT)
    {
        HAL_GPIO_WritePin(LD_R_GPIO_Port, LD_R_Pin, GPIO_PIN_SET); //11
        HAL_GPIO_WritePin(LD_G_GPIO_Port, LD_G_Pin, GPIO_PIN_SET);
    }
    if(MyTKeys[1].p_Data->StateId == TSL_STATEID_DETECT)
    {
        HAL_GPIO_WritePin(LD_R_GPIO_Port, LD_R_Pin, GPIO_PIN_SET); //01
        HAL_GPIO_WritePin(LD_G_GPIO_Port, LD_G_Pin, GPIO_PIN_RESET);
    }
    if(MyTKeys[2].p_Data->StateId == TSL_STATEID_DETECT)
    {
        HAL_GPIO_WritePin(LD_R_GPIO_Port, LD_R_Pin, GPIO_PIN_RESET); //01
        HAL_GPIO_WritePin(LD_G_GPIO_Port, LD_G_Pin, GPIO_PIN_SET);
    }
}
}
/* USER CODE BEGIN 3 */
```

Now the system is functional and ready to be used.

The Led is blink according to the position of the on slider.

Revision history

Table 22. Document revision history

Date	Version	Changes
19-Sep-2018	1	Initial release.

Contents

1	General information	2
2	Terminology and principle	3
2.1	Terminology	3
2.2	Principle	3
3	Document reference	5
4	STM32L4 touch sensing controller online presentation	6
5	Main characteristics	7
5.1	Description	7
5.2	Signal threshold	7
5.3	Charge transfer	9
5.4	Sensitivity	10
5.5	Sensor	10
5.5.1	Key	11
5.5.2	Linear or slider	12
5.5.3	Rotary or wheel	13
5.5.4	Active shield or driven shield	14
5.6	Layout and PCB	15
5.6.1	Led rules	15
5.6.2	Electrode not located on PCB	16
5.6.3	Ground, shield and sensors	17
5.6.4	FAQ	20
5.7	Noise	21
5.7.1	Power supply	21
5.7.2	False detection	21
5.7.3	Noise immunity	22
5.7.4	Conducted noise	22
6	Tuning	23
7	Getting started TSC with STM32CubeMX	24
7.1	Uses cases	24

7.2	Discovery board: STM32F072B-DISCO	25
7.2.1	STM32F072B-DISCO board selection	25
7.2.2	TSC group and sensor activation	27
7.2.3	STM32F072B-DISCO clock tree	28
7.2.4	STM32F072B-DISCO touchsensing library	29
7.2.5	STM32F072B-DISCO software project generation	32
7.2.6	Software basic algorithm	34
7.3	Discovery board: STM32L0538-DISCO	36
7.3.1	STM32L0538-DISCO board selection	36
7.3.2	STM32L0538-DISCO TSC group and sensor activation	38
7.3.3	STM32L0538-DISCO clock tree	39
7.3.4	STM32L0538-DISCO touchsensing library	40
7.3.5	STM32L0538-DISCO software project generation	43
7.3.6	STM32L0538-DISCO software basic algorithm	45
Revision history		46

List of tables

Table 1.	Change transfer principle documentation	4
Table 2.	References documentation	5
Table 3.	Signal threshold usage documentation	9
Table 4.	Charge transfer documentation	9
Table 5.	Sensitivity documentation	10
Table 6.	Dielectric constants of common materials used in a panel construction	10
Table 7.	Key documentation	11
Table 8.	Linear touch sensor documentation	12
Table 9.	Rotary sensor documentation	13
Table 10.	Active shield documentation	14
Table 11.	Led rules documentation	15
Table 12.	Electrode documentation	16
Table 13.	Layout documentation	17
Table 14.	Power supply documentation	21
Table 15.	False detection documentation	21
Table 16.	Noise immunity documentation	22
Table 17.	Conducted noise documentation	22
Table 18.	Sensors documentation	23
Table 19.	ESD documentation	23
Table 20.	Conducted noise documentation	23
Table 21.	Sampling capacitor documentation	23
Table 22.	Document revision history	46

List of figures

Figure 1.	Charge transfer principle	4
Figure 2.	Main documentation tree	5
Figure 3.	STM32L4 online training	6
Figure 4.	STM32L4 Touch Sensing Controller online training	6
Figure 5.	TSC characteristics	7
Figure 6.	STMStudio outputs	8
Figure 7.	Incomplete and complete charge transfert cycle	9
Figure 8.	Sensor size	11
Figure 9.	Interlaced linear touch sensor with 3 channels / 4 electrodes (half-ended electrodes design)	12
Figure 10.	Interlaced patterned rotary sensor with 3 channels / 3 electrodes	13
Figure 11.	Active shield principle	14
Figure 12.	Led layout example	15
Figure 13.	Example of cases where a LED bypass capacitor is required	15
Figure 14.	Electrode not located on PCB example	16
Figure 15.	Hatched ground and signal tracks	17
Figure 16.	Ground plane example	17
Figure 17.	Track routing	18
Figure 18.	Track routing recommendation	18
Figure 19.	Shield	19
Figure 20.	Typical power supply schematic	21
Figure 21.	Main project panel.	24
Figure 22.	STM32F072B-DISCO board selection	25
Figure 23.	STM32F072B-DISCO board schematics	26
Figure 24.	STM32F072B-DISCO pinout SWD	27
Figure 25.	STM32F072B-DISCO pinout TSC	27
Figure 26.	STM32F072B-DISCO pinout overview	28
Figure 27.	STM32F072B-DISCO clock configuration	28
Figure 28.	TOUCHSENSING box configuration	29
Figure 29.	STM32F072B-DISCO sensor selection	29
Figure 30.	STM32F072B-DISCO sensor selection step2	30
Figure 31.	STM32F072B-DISCO sensor selection step3	30
Figure 32.	STM32F072B-DISCO sensor selection step4	31
Figure 33.	STM32F072B-DISCO sensor selection step5	31
Figure 34.	STM32F072B-DISCO software generation step1	32
Figure 35.	STM32F072B-DISCO software generation step2	32
Figure 36.	STM32F072B-DISCO software generation step3	33
Figure 37.	STM32F072B-DISCO IDE workspace	33
Figure 38.	STM32F072B-DISCO setup	35
Figure 39.	STM32L0538-DISCO board selection	36
Figure 40.	STM32L0538-DISCO board schematics	37
Figure 41.	Pinout SWD	38
Figure 42.	Pinout TSC	38
Figure 43.	Pinout overview	39
Figure 44.	Clock configuration	39
Figure 45.	TOUCHSENSING box configuration	40
Figure 46.	STM32L0538-DISCO sensor selection step1	40
Figure 47.	STM32L0538-DISCO sensor selection step2	41
Figure 48.	STM32L0538-DISCO sensor selection step3	41
Figure 49.	STM32L0538-DISCO sensor selection step4	42
Figure 50.	STM32L0538-DISCO sensor selection step5	42
Figure 51.	STM32L0538-DISCO software generation step1	43
Figure 52.	STM32L0538-DISCO software generation step2	43

Figure 53.	STM32L0538-DISCO complete project overview.	44
Figure 54.	STM32L0538-DISCO IDE workspace	44
Figure 55.	SWD settings	44

IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST's terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers' products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2018 STMicroelectronics – All rights reserved