UM1913
User manual

How to develop applications on STM32Cube with STMTouch touch sensing library

Introduction

STM32Cube is an STMicroelectronics original initiative to improve designer productivity significantly by reducing development effort, time, and cost. STM32Cube covers the whole STM32 portfolio. STM32Cube includes (see *Section 2* for more details):

- A set of user-friendly software development tools to cover project development from the conception to the realization, among which STM32CubeMX, a graphical software configuration tool, STM32CubeIDE, an all-in-one development tool, STM32CubeCLT, an all-in-one command-line development toolset, STM32CubeProgrammer (STM32CubeProg), a programming tool, and STM32CubeMonitor, a set of monitoring tools.
- STM32Cube MCU and MPU Packages, comprehensive embedded-software platforms specific to each microcontroller and microprocessor series (such as STM32CubeU0 for the STM32U0 series), which include STM32Cube hardware abstraction layer (HAL), STM32Cube low-layer APIs, a consistent set of middleware components, and all embedded software utilities.
- STM32Cube Expansion Packages, which contain embedded software components that complement the functionalities of the STM32Cube MCU and MPU Packages with middleware extensions and applicative layers, and examples.

This user manual describes the STMTouch touch sensing library (TSL) that is part of the STM32Cube and includes:

- A complete register address mapping with all bits, bitfields and registers declared in C
- A collection of routines and data structures covering all functions to manage touch sensing technology

The source code is developed using the ANSI-C standard is MISRA C® 2004 compliant. Writing the whole library in 'Strict ANSI-C' makes it independent from the development tools. Only the startup files depend on the development tools.

Table 1. Applicable products

Type	Software package (associated STM32 series)
STM32Cube MCU Package	STM32CubeF0 (STM32F0), STM32CubeF3 (STM32F3), STM32CubeL0 (STM32L0), STM32CubeL1 (STM32L1), STM32CubeL4 (STM32L4/L4+), STM32CubeL5 (STM32L5), STM32CubeU0 (STM32U0), STM32CubeU3 (STM32U3), STM32CubeU5 (STM32U5), STM32CubeWB (STM32WB)

Contents

List of tables

List of figures

1 General information

The document applies to STM32 MCUs Arm®[a]-based devices.

arm

1.1 Acronyms and abbreviations

The table below summarizes all acronyms and abbreviations used in this document.

Table 2. Terms and acronyms

Name	Definition
Bank	Group of channels acquired simultaneously
Channel	Elementary acquisition item
Cs	Charge-transfer sampling capacitance
Ct	Equivalent touch capacitance
CT	Charge-transfer acquisition principle
Cx	Equivalent sensor capacitance
Delta	Difference between the measure and the reference
DTO	Detection time-out
DXS	Detection exclusion system
ECS	Environment change system
Linear sensor	Multi-channels sensor with electrodes positioned in a linear way
LinRot sensor	Linear or rotary touch sensor
Measure or Meas	Current signal measured on a channel
Reference or Ref	Reference signal initialized during the calibration and then regularly updated by the ECS
Rotary sensor	Multi-channels sensor with electrodes positioned in a circular way
Rs	ESD protection serial resistor
Sensor or object	Any touch sensor (such as touchkey, linear or rotary)
Timer acquisition mode	Acquisition using two timers and PWM signals (also called hardware acquisition mode). *Note: Only available on STM32L1 series microcontrollers*
Touchkey or TKey sensor	Single channel sensor
TSC	Touch sensing controller peripheral

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

1.2 Related documents

[1] MISRA C 2004 Guidelines for the use of the C language in critical systems.

[2] *How to design surface sensors for touch sensing applications on STM32 MCUs* (AN4312)

[3] *How to improve conducted noise robustness for touch sensing applications on STM32 MCUs* (AN4299)

[4] *How to tune touch sensing applications on STM32 MCUs* (AN4316)

1.3 Naming conventions

The following naming conventions are used in the STMTouch touch sensing library source files:

- Source and header files are in lower-case and preceded by 'tsl' or 'tsl_'.
- The microcontroller family is added at the end of the file name if needed.
- Functions, globals, typedefs and defines are preceded by 'TSL'.
- Constants are written in upper case and preceded by 'TSLPRM_'.
- Constants used in one file are defined within this file only.
- Constants used in more than one file are defined in a header file.
- Typedef names are suffixed with '_T'.
- Enum typedefs are suffixed with '_enum_T'.
- Functions are named according to the 'TSL_[module]_[function]' scheme:
  - [module]: abbreviation of the file (such as acq, tim, dxs)
  - [function]: the first letter in each word is in upper case.

1.4 Coding rules

This section describes the coding rules used in the STMTouch touch sensing library source files.

1.4.1 General

- Source code complies with ANSI C standard.
- No warning after compilation. Any warning that cannot be eliminated is commented in the source code.
- ANSI standard data types are used and defined in the ANSI C header file <stdint.h>.
- No blocking code is present and all required waiting loops (polling loops) are controlled by a timeout.

1.4.2 Variable types

Specific variable types are already defined with a fixed type and size:

- The types that are used by all modules are defined in the *tsl_types.h* file.
- Other variable types are defined in their corresponding module header file.

1.4.3 Peripheral registers

The peripheral registers are accessed using the pointers described in the CMSIS device peripheral access layer header file.

1.5 MISRA C 2004 compliance

1.5.1 Generalities

The C programming language importance grows for embedded systems. However, when it comes to developing code for safety-critical applications, this language has many drawbacks. There are several unspecified, implementation-defined, and undefined aspects of the C language that make it unsuited for developing safety-critical systems.

The motor industry software reliability association describes a subset of the C language well suited for developing safety-critical systems in document *[1]*.

The STMTouch touch sensing library has been developed to be MISRA C 2004 compliant.

The following section describes how the STMTouch touch sensing library complies with MISRA C 2004 (as described in section ‘Claiming compliance’ of document *[1]*:

- A compliance matrix has been completed. It shows how compliance has been enforced.
- The whole STMTouch touch sensing library source code is compliant with MISRA C 2004 rules.
- Deviations are documented. A list of all instances of rules not being followed is maintained. For each instance, there is an appropriately signed-off deviation.
- All the issues listed in section ‘The programming language and coding context’ of document *[1]*, that need to be checked during the firmware development phase, have been addressed during the development of the STMTouch touch sensing library and appropriate measures have been taken.

1.5.2 Compliance matrix

The compliance of the STMTouch touch sensing library with MISRA C 2004 has been checked in two ways:

- using PC-lint tool for C/C++ (NT) versus 8.00v, copyright gimpel software 1985-2006
- performing regular code reviews

The following table lists the MISRA C 2004 rules that are frequently violated in the code:

Table 3. MISRA C 2004 rules not followed

MISRA C 2004 rule number	Required / advisory	Summary	Reason of deviance
1.1 1.2	Required	All code must conform to ISO 9899:1990 standard C, with no extensions permitted.	Compilers extensions are enabled. Comments starting with “//” symbol for code readability.
5.4	Required	A tag name must be a unique identifier.	Due to the usage of objects methods

Table 3. MISRA C 2004 rules not followed (continued)

MISRA C 2004 rule number	Required / advisory	Summary	Reason of deviance
8.1	Required	No prototype seen. Functions must always have prototype declarations and the prototype must be visible at both the function definition and call.	This rule is violated as there is no function prototypes for the object methods.
10.1 10.2	Required	The value of an expression of integer/floating type must not be implicitly converted to a different underlying type.	Code complexity
10.3	Required	The value of a complex expression of integer type may only be cast to a type that is narrower and of the same signedness as the underlying type of the expression.	Code complexity
10.5	Required	If the bitwise operators are applied to an operand of underlying type unsigned char or unsigned short, the result must be immediately cast to the underlying type of the operand.	Use shift on signed quantity for the linear/rotary position.
11.3	Advisory	A cast must not be performed between a pointer type and an integral type.	Needed when addressing memory-mapped registers
12.7	Required	Bitwise operators must not be applied to operands whose underlying type is signed.	Shift of signed value needed
14.3	Required	Before preprocessing, a null statement must only occur on a line by itself.	Use of macros to simplify the code.
14.5	Required	The continue statement must not be used.	Used to optimize the code speed execution
19.11	Required	All macro identifiers in preprocessor directives must be defined before use, except in `ifdef` and `ifndef` preprocessor directives and the `defined()` operator.	All parameters are checked in the check_config files.

2 What is STM32Cube?

STM32Cube is an STMicroelectronics original initiative to improve designer productivity significantly by reducing development effort, time, and cost. STM32Cube covers the whole STM32 portfolio.

STM32Cube includes:

- A set of user-friendly software development tools to cover project development from conception to realization, among which are:
  - STM32CubeMX, a graphical software configuration tool that allows the automatic generation of C initialization code using graphical wizards
  - STM32CubeIDE, an all-in-one development tool with peripheral configuration, code generation, code compilation, and debug features
  - STM32CubeCLT, an all-in-one command-line development toolset with code compilation, board programming, and debug features
  - STM32CubeProgrammer (STM32CubeProg), a programming tool available in graphical and command-line versions
  - STM32CubeMonitor (STM32CubeMonitor, STM32CubeMonPwr, STM32CubeMonRF, STM32CubeMonUCPD), powerful monitoring tools to fine-tune the behavior and performance of STM32 applications in real time
- STM32Cube MCU and MPU Packages, comprehensive embedded-software platforms specific to each microcontroller and microprocessor series (such as STM32CubeU3 for the STM32U3 series), which include:
  - STM32Cube hardware abstraction layer (HAL), ensuring maximized portability across the STM32 portfolio
  - STM32Cube low-layer APIs, ensuring the best performance and footprints with a high degree of user control over hardware
  - A consistent set of middleware components such as ThreadX, FileX, LevelX, NetX Duo, USBX, USB PD, touch library, network library, mbed-crypto, TFM, and OpenBL
  - All embedded software utilities with full sets of peripheral and applicative examples
- STM32Cube Expansion Packages, which contain embedded software components that complement the functionalities of the STM32Cube MCU and MPU Packages with:
  - Middleware extensions and applicative layers
  - Examples running on some specific STMicroelectronics development boards

3 STMTouch touch sensing library

3.1 Supported microcontrollers and development tools

3.1.1 Supported microcontrollers

This STMTouch touch sensing library version supports the following MCUs and acquisition modes:

- STM32 MCUs embedding the TSC (touch sensing controller) peripheral (all series in *Table 1* except STM32L1 series):
  - Surface charge-transfer acquisition principle managed by the TSC
  - Up to 24 channels (eight groups of three channels maximum)
  - Up to eight channels that can be acquired simultaneously
  - Spread spectrum feature
  - Programmable charge-transfer frequency and maximum count value
- STM32L1 series: The surface charge-transfer acquisition principle is managed by:
  - Two timers plus a routing interface (hardware acquisition mode). This mode is not supported on STM32L1 series microcontrollers featuring 256-Kbyte or less memory.
  - GPIOs plus a routing interface (software acquisition mode). This mode is supported by all microcontrollers.
  - Up to 34 channels
  - Up to 11 channels that can be acquired simultaneously

3.1.2 Development tools

The STM32 MCUs are supported by a full range of development solutions from lead suppliers that deliver start-to-finish control of application development from a single integrated development environment.

The STMTouch touch sensing library has been developed with the following toolchains:

- IAR Systems® - IAR Embedded Workbench® (EWARM)
- Keil® - Microcontroller Development Kit (MDK-ARM)
- STMicroelectronics - STM32CubeIDE

For more details about the compilers versions used, see the STM32Cube MCU Package release note.

3.2 Package description

The following snapshots show an example of installation inside the STM32CubeF0 package.

Figure 1. Installation folder 1/2 (library)

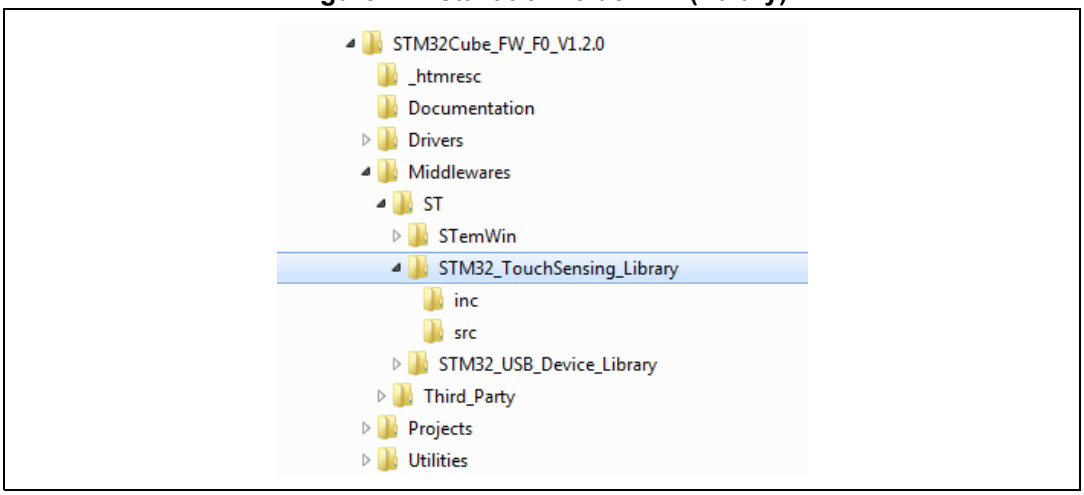
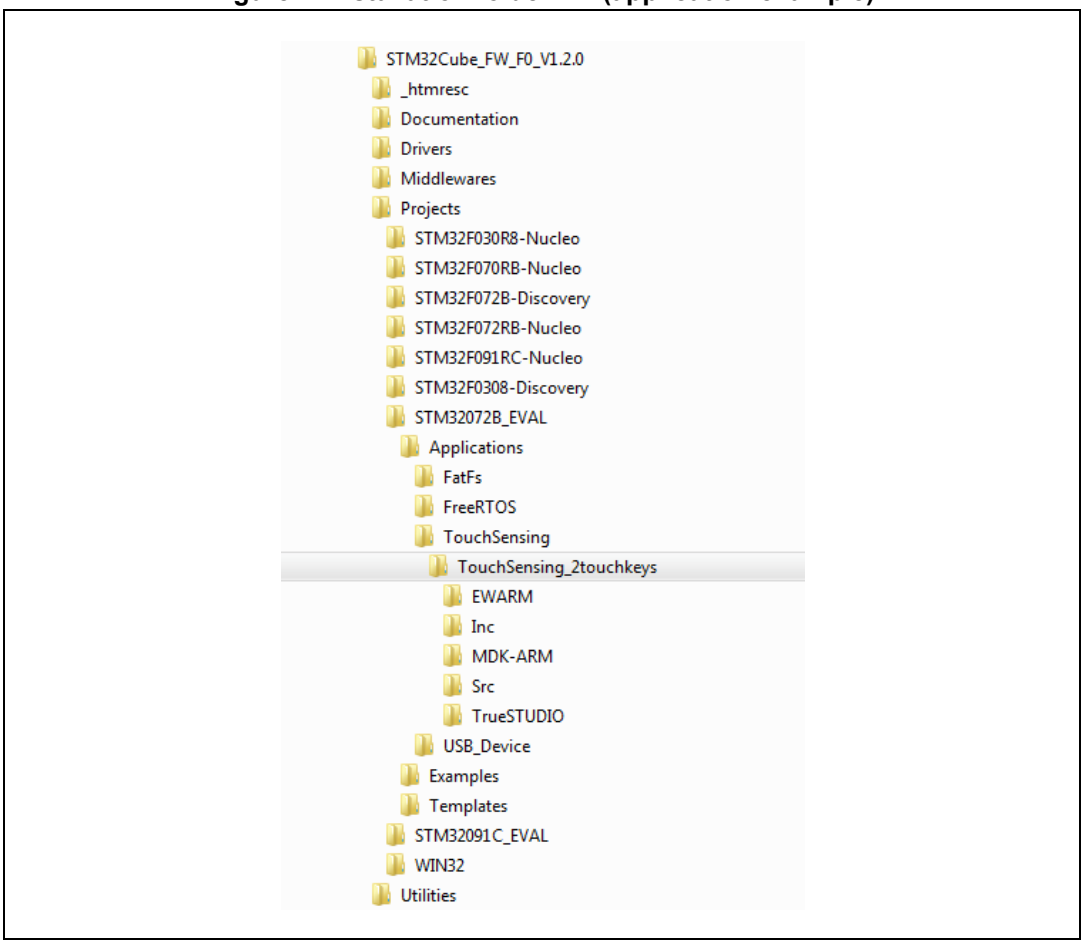


Figure 2. Installation folder 2/2 (application example)



3.3 Main features

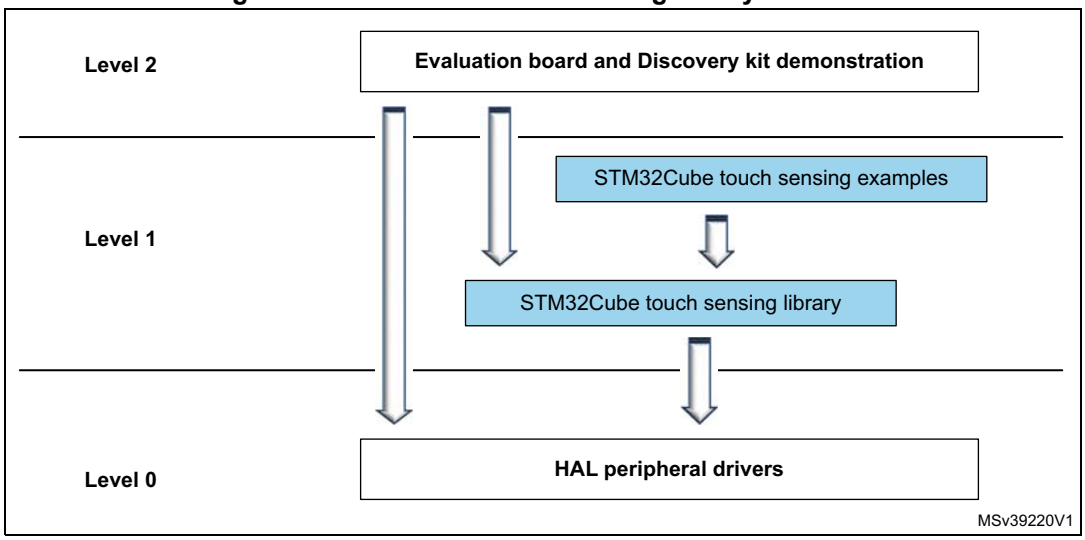
- Proximity, touchkeys, linear and rotary touch sensors supported
- Environment change system (ECS)
- Detection time out (DTO)
- Detection exclusion system (DXS)
- Noise filter
- Unlimited number of sensors
- Modular architecture allowing easy addition of new acquisitions or sensors
- Each sensor having its own state machine
- Simplified timing management
- Management of error during acquisition

3.4 Architecture

3.4.1 Overview

The following figure shows the interactions between the STMTouch touch sensing library and the other firmware layers.

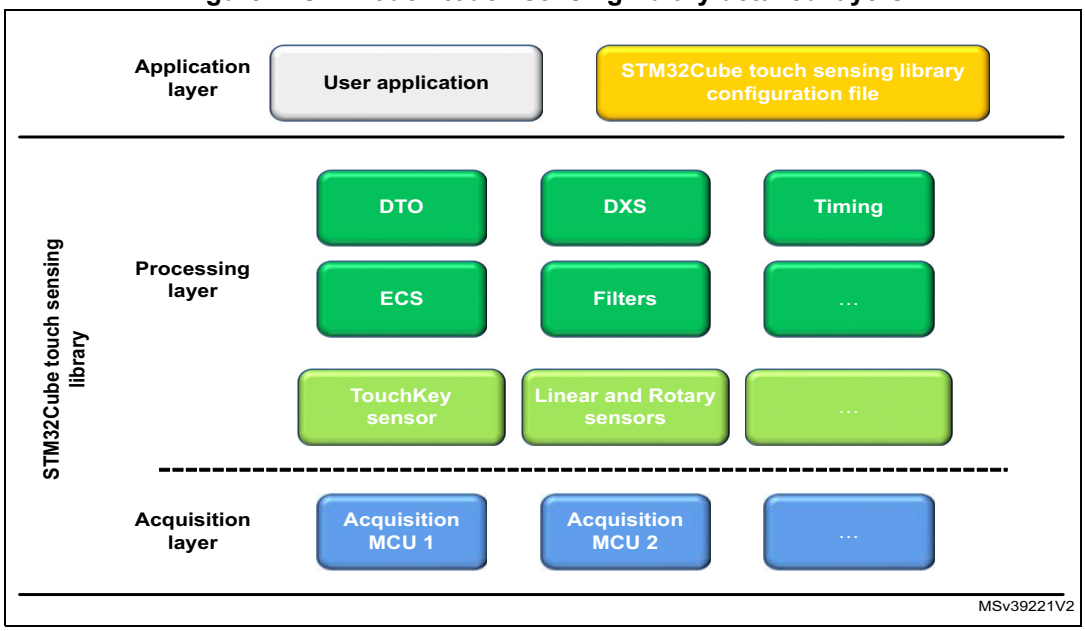
Figure 3. STM32Cube touch sensing library overview



3.4.2 STMTouch touch sensing library layers

The following figure shows a more detailed view of the different STM32Cube touch sensing library layers.

Figure 4. STMTouch touch sensing library detailed layers



The STMTouch touch sensing library is composed of the following main layers:

- acquisition layer
- processing layer
- configuration layer

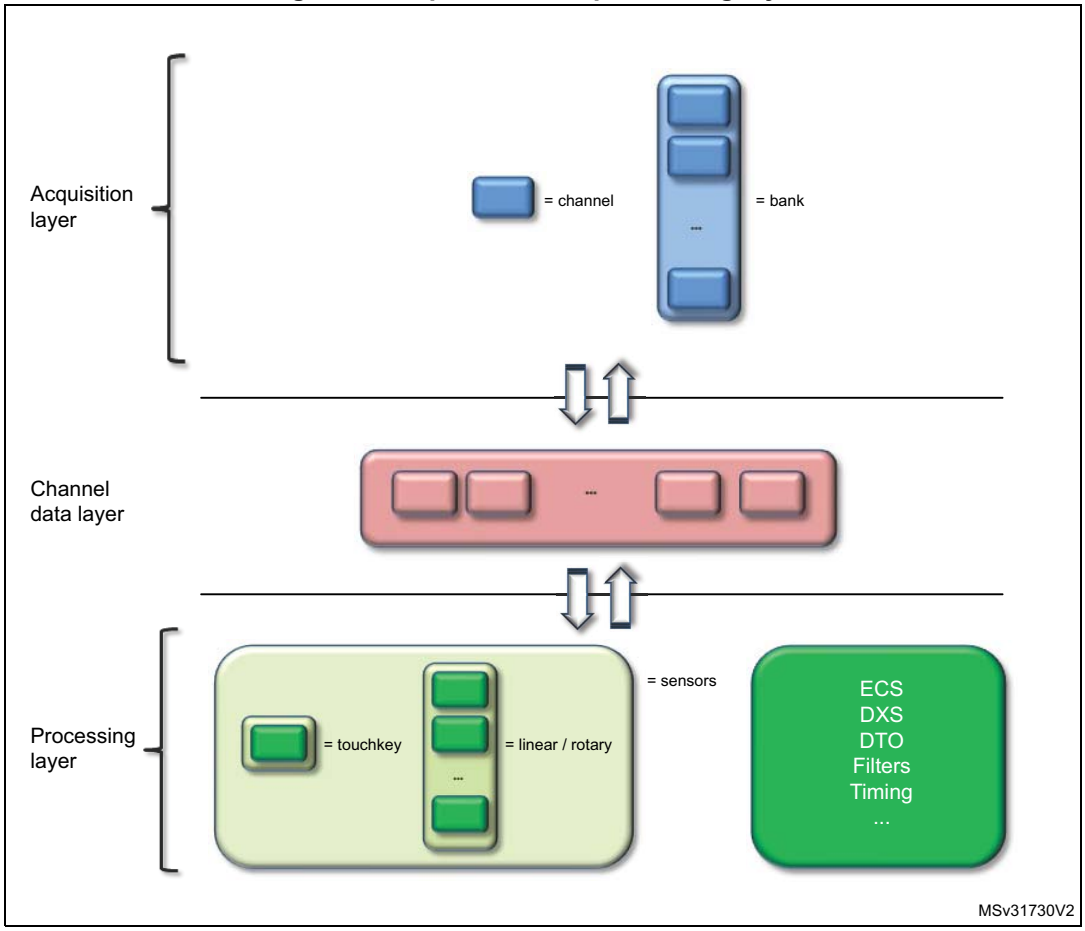
The configuration layer corresponds to what the user needs to write in the application code in order to correctly use the STMTouch touch sensing library. This includes all the channel and sensor declarations and the parameters for example.

The acquisition and processing layers are described in more details in the next section.

3.4.3 Acquisition and processing layers

The following figure details the acquisition and processing layers and the different elements used in each layer.

Figure 5. Acquisition and processing layers



The acquisition layer role is to perform the acquisition of the different channels. The result of the acquisition (measure and flags) is stored inside the channel data layer. All the information are accessed by the processing layer.

The acquisition layer has only access to the channels and banks. It does not have access to the sensors.

The channel data layer role is to share information between the acquisition and processing layers. It stores the result of the acquisition (measure) for each channel and store different informations coming from the processing layer (such as reference, delta or flags).

Located in RAM, the ChannelData structure is the only interface between the acquisition and processing layers.

This processing layer consists in executing each sensor state machine, executing the different data processing like ECS, DXS, DTO, and storing any useful information for the acquisition layer inside the channel data area.

The processing layer does not have direct access to the channels and banks. This access is made through the sensors.

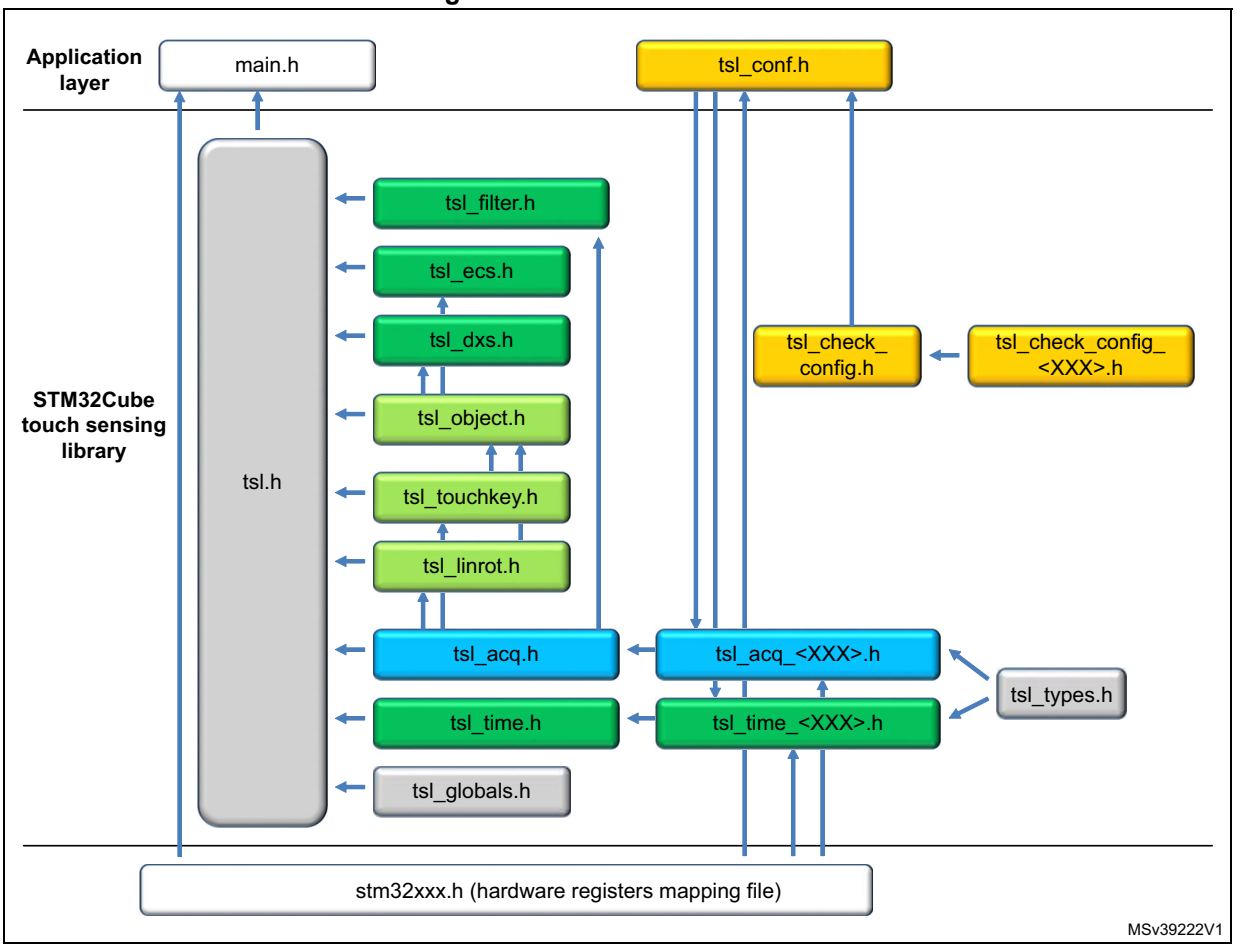
3.4.4 Header files inclusion

The figure below provides a global view of the STMTouch touch sensing library use and the interaction between the different header files.

In the actual version of the STMTouch touch sensing library, the <XXX> is equal to “tsc” or “stm32l1xx”.

Note: To simplify the drawing, only the most important links are shown. For example the tsl_globals.h file is also included in different files.

Figure 6. Header files inclusion



3.5 Channel

3.5.1 Principle

A channel is the basic element that is used to store several information like:

- where the source measurement can be found after the acquisition is performed (TSC_IOGxCR registers for TSC acquisition)
- where are stored the measure, the reference, the delta or the flags

3.5.2 Resources

A channel is defined by the following data structures:

- `TSL_ChannelSrc_T`: contains all information about the source measurement (such as index of the register containing the measurement or masks)
- `TSL_ChannelDest_T`: contains all information about the measurement destination (index in the channel data array).
- `TSL_ChannelData_T`: contains all data for the channel (such as measure, delta or reference)

The channel depends on the acquisition technology. This is why the contents of these structures are not common for all acquisitions. They are declared in each acquisition header files (*tsl_acq_<XXX>.h*):

- t*sl_acq_stm32l1xx_hw.h* for STM32L1 using the hardware acquisition mode
- *tsl_acq_stm32l1xx_sw.h* for STM32L1 using the software acquisition mode
- *tsl_acq_tsc.h* for any STM32 featuring the TSC peripheral

The maximum number of channels is only limited by the device (memory size and channels supported).

The user must declare all the channel arrays in the application code. It can be done directly in the *main.c* file or in any other file.

3.5.3 Parameters

- `TSLPRM_TOTAL_CHANNELS`

3.5.4 Example

The channel structure must be declared in the application code.

Example of channel source array declaration for microcontrollers featuring TSC peripheral. This structure must always be placed in the ROM.

```
const TSL_ChannelSrc_T MyChannels_Src[TSLPRM_TOTAL_CHANNELS] =
{ { CHANNEL_0_SRC },
  { CHANNEL_1_SRC },
  { CHANNEL_2_SRC }};
```

Example of channel destination array declaration for microcontrollers featuring TSC peripheral. This structure must always be placed in the ROM.

```
const TSL_ChannelDest_T MyChannels_Dest[TSLPRM_TOTAL_CHANNELS] =
{ { CHANNEL_0_DEST },
  { CHANNEL_1_DEST },
  { CHANNEL_2_DEST }};
```

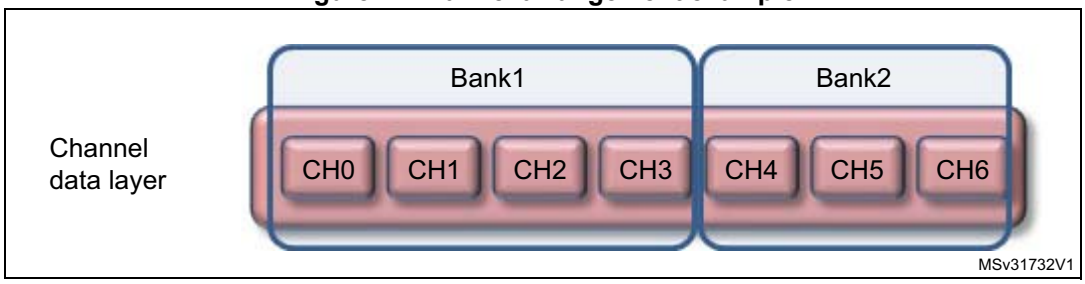
Note: The `CHANNEL_x_SRC` and `CHANNEL_x_DEST` are `#define` constants and are used for readability. The values are acquisition dependent.

Example of channel data array declaration (channel data layer). This structure must always be placed in the RAM.

```
TSL_ChannelData_T MyChannels_Data[TSLPRM_TOTAL_CHANNELS];
```

Warning: When several banks are present, it is mandatory to declare all channels of each bank consecutively in the source and destination structures.

Figure 7. Channel arrangement example



Example of channel source array declaration for microcontrollers featuring TSC peripheral.

```
CONST TSL_ChannelSrc_T MyChannels_Src[TSLPRM_TOTAL_CHANNELS] =
{
// Bank 1
{ CHANNEL_0_SRC, CHANNEL_0_IO_MSK, CHANNEL_0_GRP_MSK },
{ CHANNEL_1_SRC, CHANNEL_1_IO_MSK, CHANNEL_1_GRP_MSK },
{ CHANNEL_2_SRC, CHANNEL_2_IO_MSK, CHANNEL_2_GRP_MSK },
{ CHANNEL_3_SRC, CHANNEL_3_IO_MSK, CHANNEL_3_GRP_MSK },
// Bank 2
{ CHANNEL_4_SRC, CHANNEL_4_IO_MSK, CHANNEL_4_GRP_MSK },
{ CHANNEL_5_SRC, CHANNEL_5_IO_MSK, CHANNEL_5_GRP_MSK },
{ CHANNEL_6_SRC, CHANNEL_6_IO_MSK, CHANNEL_6_GRP_MSK }
};
```

3.6 Bank

3.6.1 Principle

A bank is a group of channels that are acquired simultaneously. The number of channels in the bank is variable.

3.6.2 Resources

The bank data are held by only one structure: `TSL_Bank_T`.

The bank depends also on the acquisition technology. Structures are declared in each acquisition header files (*tsl_acq_<XXX>.h*).

The maximum number of banks is only limited by the device.

The user must declare all the bank arrays in the application code. It can be done directly in the *main.c* file or in any other file.

The banks are used mainly by the functions described below. Some functions are common whatever the device and acquisition technology. Some others are dependent on the device.

Common functions:

- `TSL_acq_BankGetResult()`
- `TSL_acq_BankCalibrate()`

Device-dependent functions:

- `TSL_acq_BankConfig()`
- `TSL_acq_BankStartAcq()`
- `TSL_acq_BankWaitEOC()`

3.6.3 Parameters

- `TSLPRM_TOTAL_BANKS`

3.6.4 Example

Example of three banks declaration for microcontrollers featuring TSC peripheral:

```
CONST TSL_Bank_T MyBanks[TSLPRM_TOTAL_BANKS] = {
  {&MyChannels_Src[0], &MyChannels_Dest[0], MyChannels_Data,
BANK_0_NBCHANNELS, BANK_0_MSK_CHANNELS, BANK_0_MSK_GROUPS},
  {&MyChannels_Src[1], &MyChannels_Dest[1], MyChannels_Data,
BANK_1_NBCHANNELS, BANK_1_MSK_CHANNELS, BANK_1_MSK_GROUPS},
  {&MyChannels_Src[2], &MyChannels_Dest[2], MyChannels_Data,
BANK_2_NBCHANNELS, BANK_2_MSK_CHANNELS, BANK_2_MSK_GROUPS}
};
```

3.7 Objects

3.7.1 Principle

The term “object” or “sensor” stands for any sensor type (touchkeys, linear and rotary touch sensors) supported by the STMTouch touch sensing library.

3.7.2 Resources

All processing that affect the sensors in general are defined in the following files:

- *tsl_object.c*
- *tsl_object.h*

The functions are:

- `TSL_obj_GroupInit()`
- `TSL_obj_GroupProcess()`
- `TSL_obj_SetGlobalObj()`

A sensor is described by the structures:

- TSL_Object_T
- TSL_ObjectGroup_T

3.7.3 Parameters

- TSLPRM_TOTAL_OBJECTS

3.7.4 Example

All touchkeys, linear and rotary touch sensors (described after) used in the application must be described first as 'generic' sensors or objects.

Example:

```
// Mix of touchkeys and Linear touch sensors
const TSL_Object_T MyObjects[TSLPRM_TOTAL_OBJECTS] =
{
  // TKeys
  { TSL_OBJ_TOUCHKEYB, (TSL_TouchKeyB_T *)&MyTKeys[0]   },
  { TSL_OBJ_TOUCHKEYB, (TSL_TouchKeyB_T *)&MyTKeys[1]   },
  // Linear touch sensors
  { TSL_OBJ_LINEARB,   (TSL_LinRotB_T   *)&MyLinRots[0] }
};
```

These objects must be placed in the ROM.

Once this is done, it is necessary to create at least one group of sensors. Groups of sensors are used by the different processing routines (such as ECS or DXS).

These groups of objects must be placed in the RAM.

Example:

```
TSL_ObjectGroup_T MyObjGroup_All = {
 MyObjects,
 3,
 0,
 TSL_STATE_NOT_CHANGED
};
```

Then, all the sensors must be initialized and “processed”. This is done in the main function of the application:

```
int main(void) {
  ...
  TSL_obj_GroupInit(&MyObjGroup_All);
  ...
  while (1) {
    ...
    TSL_obj_GroupProcess(&MyObjGroup_All);
    ...
  }
}
```

3.8 Touchkey sensor

3.8.1 Principle

The touchkey sensor is composed of only one channel. It acts as a simple button with two states RELEASE and DETECT (or TOUCH if DXS is enabled).

3.8.2 Resources

All the functions related to this sensor are described in the files:

- *tsl_touchkey.c*
- *tsl_touchkey.h*

Two types of touchkey sensor are available:

- Basic: defined by the `TSL_TouchKeyB_T` structure
- Extended: defined by the `TSL_TouchKey_T` structure

Two functions (called methods) are used to initialize the sensor parameters and to run the sensor state machine:

- `TSL_tkey_Init()`
- `TSL_tkey_Process()`

The difference between the basic and extended types concerns the methods and sensor state machine used:

- For a basic sensor, the methods and state machine are those used in the `TSL_Params` structure.
- For an extended sensor, the methods and state machine are those declared in their own structure.

3.8.3 Parameters

- `TSLPRM_TOTAL_TKEYS`

3.8.4 Example

The user must declare these methods in the application code.

Note: User initialization and process functions can also be used instead:

```
const TSL_TouchKeyMethods_T MyTKeys_Methods =
{
  TSL_tkey_Init,
  TSL_tkey_Process
};
```

The declaration of the touchkey sensor is done by the user in the application code.

Example with a basic sensor:

```
// "Basic" touchkeys: Always placed in ROM
const TSL_TouchKeyB_T MyTKeys[TSLPRM_TOTAL_TKEYS] =
{
  { &MyTKeys_Data[0], &MyTKeys_Param[0], &MyChannels_Data[0] },
  { &MyTKeys_Data[1], &MyTKeys_Param[1], &MyChannels_Data[1] },
```

```
  { &MyTKeys_Data[2], &MyTKeys_Param[2], &MyChannels_Data[2] }
};
```

Example with an extended sensor:

```
// "Extended" TouchKeys: Always placed in ROM
const TSL_TouchKey_T MyTKeys[TSLPRM_TOTAL_TKEYS] =
{
  { &MyTKeys_Data[0], &MyTKeys_Param[0], &MyChannels_Data[0],
MyTKeys_StateMachine, &MyTKeys_Methods },
  { &MyTKeys_Data[1], &MyTKeys_Param[1], &MyChannels_Data[1],
MyTKeys_StateMachine, &MyTKeys_Methods },
  { &MyTKeys_Data[2], &MyTKeys_Param[2], &MyChannels_Data[2],
MyTKeys_StateMachine, &MyTKeys_Methods }
};
```

3.9 Linear and rotary touch sensors

3.9.1 Principle

The linear and rotary touch sensors are like a touchkey sensor except that they are composed of a variable number of channels. The difference between the linear and rotary touch sensors is how the electrodes are organized together.

The linear and rotary touch sensors have additional fields in their structure compared to touchkey sensors:

- Number of channels
- Delta coefficient table
- Position offset table
- Sector computation parameter
- Position correction parameter for linear sensor

Note: The last three fields are used to calculate the position.

3.9.2 Number of channels

Only 1, 3, 4, 5, and 6 channels are supported by the STMTouch touch sensing library. Additional number of channels can be added by the end user.

Note: A linear touch sensor with one channel is equivalent to one touchkey sensor. When an application uses both touchkey, linear and rotary sensors, it is better to use touchkeys with a 1-channel linear touch sensor. In this case, the gain in memory size is important as the touchkey sensor state machine is not used.

3.9.3 Delta coefficient table

The delta coefficient table is used to adjust each channel of the linear and rotary touch sensors. Each value is a 16-bit integer. The MSB is the integer part. The LSB is the real part.

Examples:

- To apply a 1.10 factor:
  - MSB = 0x01
  - LSB = 0x1A (0.10 x 256 = 25.6, rounded to 26 = 0x1A)
- To apply a 1.00 factor:
  - MSB = 0x01
  - LSB = 0x00
- To apply a 0.90 factor:
  - MSB = 0x00
  - LSB = 0xE6 (0.90 x 256 = 230.4, rounded to 230 = 0xE6)

This results in the following delta coefficient table:

```
CONST uint16_t MyLinRot0_DeltaCoeff[3] = {0x011A, 0x0100, 0x00E6};
```

The number of delta coefficient table is not limited. The same delta coefficient table can be shared by several linear and rotary touch sensors.

3.9.4 Electrode placement

The placement (design) of the electrodes can be done in three different manners:

- Mono-electrode design

  The number of electrodes is equivalent to the number of channels. This design is used for linear and rotary touch sensors.

  Abbreviations: LIN_M1, LIN_M2, and ROT_M

  Examples:
  - CH1 CH2 CH3
  - CH1 CH2 CH3 CH4
  - CH1 CH2 CH3 CH4 CH5
- Dual-electrode design

  All the electrodes are duplicated and interlaced together in order to increase the touch area.

  This design is used for linear and rotary touch sensors composed with at least five channels.

  Abbreviation: ROT_D

  Examples with 5 channels:
  - CH1 CH2 CH3 CH4 CH5 CH1 CH3 CH5 CH2 CH4
  - CH1 CH2 CH3 CH4 CH5 CH2 CH4 CH1 CH3 CH5
  - CH1 CH2 CH3 CH4 CH5 CH3 CH1 CH4 CH2 CH5
- Half-ended electrode design

  The first electrode is duplicated and the replica is placed at the end. The size of the first and last electrode is half the size of the other electrodes. This design is used for linear

sensors only. The 0 and 255 positions are obtained more easily compared to the mono-electrode design.

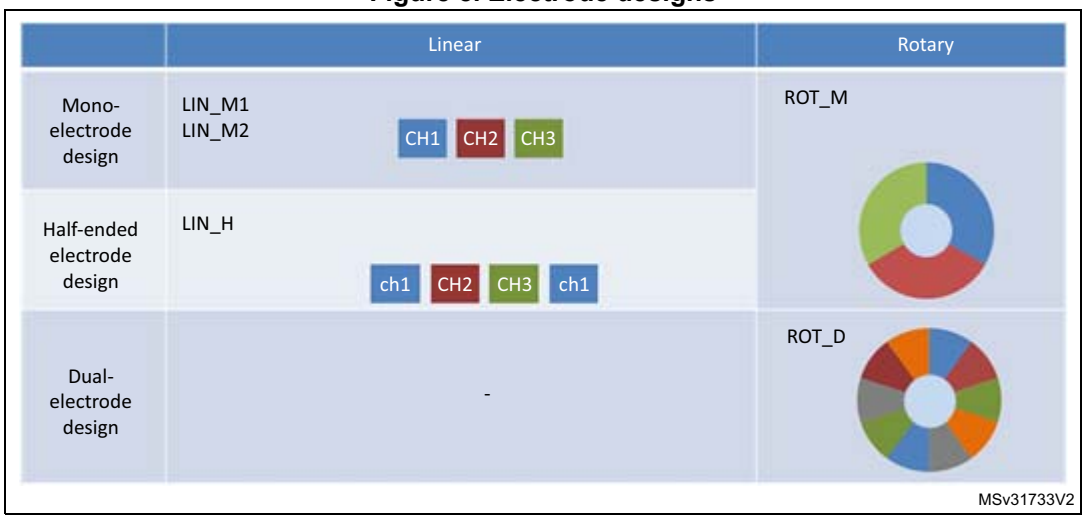
Abbreviation: LIN_H

Examples:

– ch1 CH2 CH3 ch1
– ch1 CH2 CH3 CH4 ch1
– ch1 CH2 CH3 CH4 CH5 ch1

The following figure summarizes the different electrode designs on linear and rotary touch sensors:

Figure 8. Electrode designs



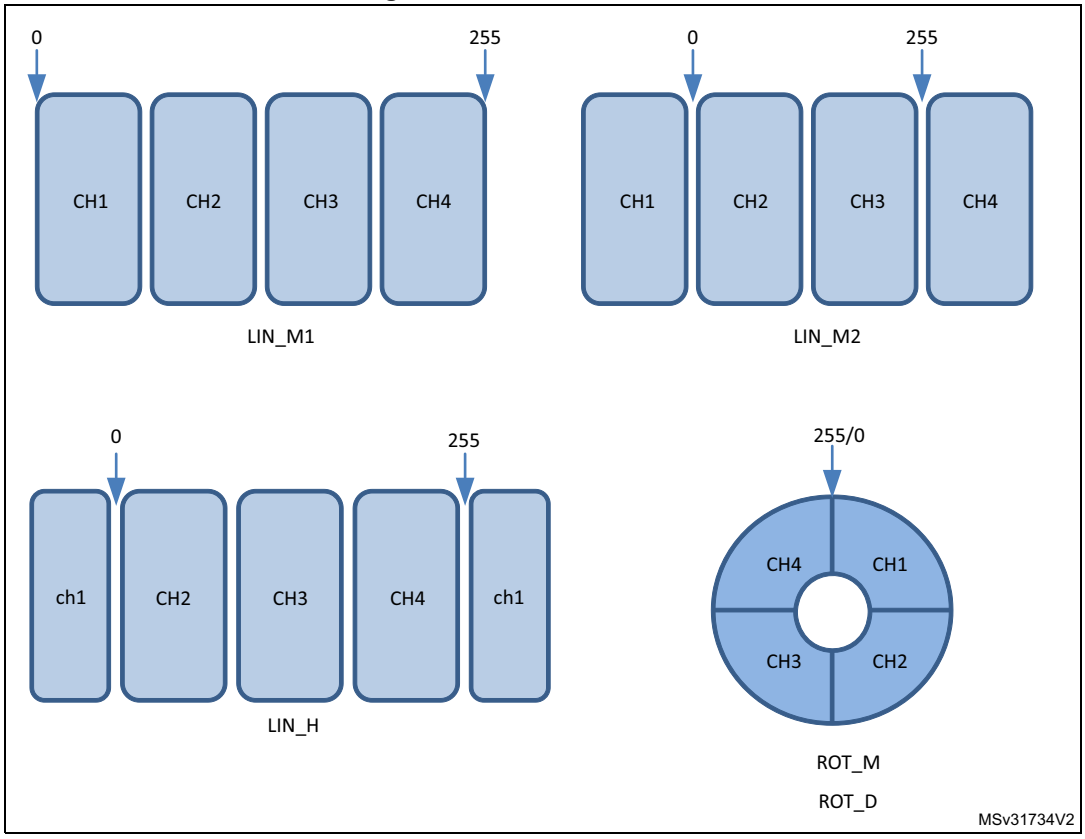
Positions 0 and 255

Special care must be taken for the 0 and 255 positions on linear sensors. These positions are placed differently depending on the electrode design used:

- LIN_M1: The 0 and 255 positions are placed completely at the sensor extremities. These positions can be obtain with difficulty if the electrodes are too big or if they are separated by an important space.
- LIN_M2, LIN_H: The 0 position is placed between the first and second electrodes. The 255 position is placed between the last two electrodes.
- ROT_M and ROT_D: the 0 and 255 positions are always placed between the first and the last electrodes.

The following figure summarizes the different placements of the 0 and 255 positions with 4-channel sensors:

Figure 9. Positions 0 and 255



The following table summarizes the different linear and rotary touch sensors electrode designs supported by the STMTouch touch sensing library:

Table 4. Supported linear and rotary touch sensors

Number of channels	LIN_M1	LIN_M2	LIN_H	ROT_M	ROT_D
3	Yes	Yes	Yes	Yes	No
4	Yes	Yes	Yes	Yes	No
5	Yes	Yes	Yes	Yes	Yes
6	Yes	Yes	Yes	Yes	No

Each supported electrode design is described by the following fields in the `TSL_LinRot_T` or `TSL_LinRotB_T` structures:

- Position offset table
- Sector computation parameter
- Position correction parameter for linear sensor

These fields are defined in the *tsl_linrot.c* and *tsl_linrot.h* files and follow the naming convention:

- position offset table: `TSL_POSOFF_nCH_[LIN|ROT]_[M1|M2|H|D]`
- sector computation parameter: `TSL_SCTCOMP_nCH_[LIN|ROT]_[M1|M2|H|D]`
- position correction parameter for linear sensor: `TSL_POSCORR_nCH_LIN_[M1|M2|H|D]`

with:

- n = number of channels
- LIN = linear sensor
- ROT = rotary sensor
- M1 = mono-electrode design with 0/255 position at extremities
- M2 = mono-electrode design
- H = half-ended electrode design
- D = dual-electrode design

In order to gain memory space, each table is only compiled if its corresponding parameter is set in the configuration file:

`TSLPRM_USE_nCH_[LIN|ROT]_[M1|M2|H|D]`

3.9.5 Resources

All the functions related to this sensor are described in the files:

- *tsl_linrot.c*
- *tsl_linrot.h*

Two types of linear and rotary sensor are available:

- basic: defined by the `TSL_LinRotB_T` structure
- extended: defined by the `TSL_LinRot_T` structure

The difference between basic and extended is the same as for the touchkey sensor.

Three functions (called methods) are used to initialized the sensor parameters, run the sensor state machine and calculate the position.

- `TSL_linrot_Init()`
- `TSL_linrot_Process()`
- `TSL_linrot_CalcPos()`

3.9.6 Parameters

- `TSLPRM_TOTAL_LINROTS`

3.9.7 Examples

The user must declared these methods in the application code.

Note: User initialization and process functions can also be used instead:

```
CONST TSL_LinRotMethods_T MyLinRots_Methods =
{
  TSL_linrot_Init,
  TSL_linrot_Process,
  TSL_linrot_CalcPos
};
```

The declaration of the linear and rotary sensor is done by the user in the application code in the same manner as for touchkey sensor.

Example with two basic linear touch sensors, one with three half-ended channels and the other with five mono-electrode channels:

```
CONST TSL_LinRotB_T MyLinRots[2] =
{
  // LinRot sensor 0
  &MyLinRots_Data[0],
  &MyLinRots_Param[0],
  &MyChannels_Data[CHANNEL_9_DEST],
  3, // Number of channels
  MyLinRot0_DeltaCoeff, // Delta coefficient table
  (TSL_tsignPosition_T *)TSL_POSOFF_3CH_LIN_H, // Position table
  TSL_SCTCOMP_3CH_LIN_H, // Sector compensation
  TSL_POSCORR_3CH_LIN_H, // Position correction
  // LinRot sensor 1
  &MyLinRots_Data[1],
  &MyLinRots_Param[1],
  &MyChannels_Data[CHANNEL_12_DEST],
  5, // Number of channels
  MyLinRot1_DeltaCoeff, // Delta coefficient table
  (TSL_tsignPosition_T *)TSL_POSOFF_5CH_LIN_M2, // Position table
  TSL_SCTCOMP_5CH_LIN_M2, // Sector compensation
  TSL_POSCORR_5CH_LIN_M2  // Position correction
};
```

Example of one extended (having its own state machine and methods) linear touch sensor with three half-ended channels:

```
CONST TSL_LinRot_T MyLinRots[1] =
{
  // LinRot sensor 0
  &MyLinRots_Data[0],
  &MyLinRots_Param[0],
  &MyChannels_Data[CHANNEL_0_DEST],
  3, // Number of channels
  MyLinRot0_DeltaCoeff,
```

```
  (TSL_tsignPosition_T *)TSL_POSOFF_3CH_LIN_H,
  TSL_SCTCOMP_3CH_LIN_H,
  TSL_POSCORR_3CH_LIN_H,
  MyLinRots_StateMachine, // Specific state machine
  &MyLinRots_Methods // Specific methods
};
```

Example of one extended rotary touch sensor with three mono-electrode channels:

```
CONST TSL_LinRot_T MyLinRots[0] =
{
  // LinRot sensor 0
  &MyLinRots_Data[0],
  &MyLinRots_Param[0],
  &MyChannels_Data[CHANNEL_0_DEST],
  3, // Number of channels
  MyLinRot0_DeltaCoeff,
  (TSL_tsignPosition_T *)TSL_POSOFF_3CH_ROT_M,
  TSL_SCTCOMP_3CH_ROT_M,
  0, // No position correction needed on a Rotary sensor
  MyLinRots_StateMachine, // Specific state machine
  &MyLinRots_Methods // Specific methods
};
```

3.10 Main state machine

The main state machine is managed by the user in the application layer. A set of functions are available to accomplish this task. The main state machine can be defined with polling or with interrupt mode, using one or several banks. The modularity of the STMTouch touch sensing library allows also the application code to be inserted between acquisition and processing tasks. Several examples are given below.

The functions to use for the acquisition are:

- `TSL_acq_BankConfig()`
- `TSL_acq_BankStartAcq()`
- `TSL_acq_BankWaitEOC()`
- `TSL_acq_BankGetResult()`

These functions are device dependent and are described in the *tsl_acq_<XXX>.c* files.

The functions to use for the processing are:

- `TSL_obj_GroupProcess()`
- `TSL_ecs_Process()`
- `TSL_dxs_FirstObj()`

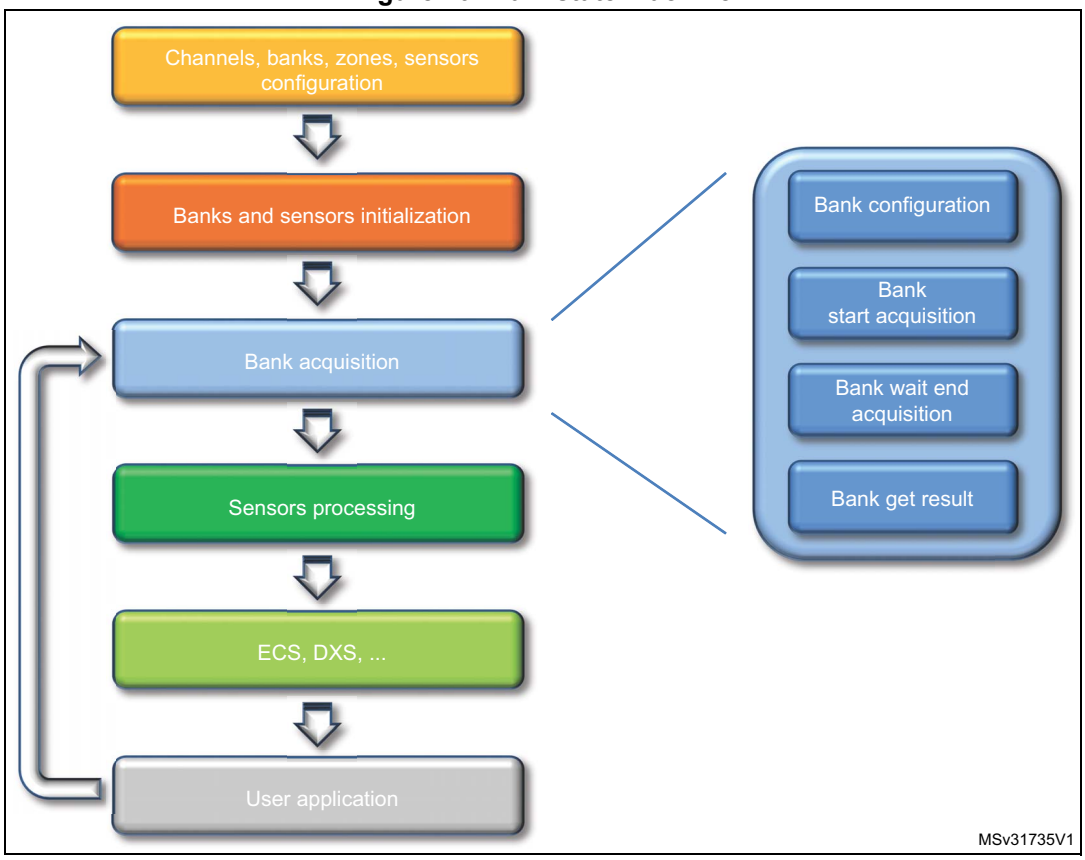
Other functions that can be used during the processing are:

- `TSL_tim_CheckDelay_ms()`
- `TSL_obj_SetGlobalObj()`
- `TSL_tkey_GetStateId()`

- `TSL_tkey_GetStateMask()`
- `TSL_linrot_SetStateOff()`
- `TSL_linrot_SetStateCalibration()`

The main state machine principle is illustrated by the figure below:

Figure 10. Main state machine



The main state-machine steps are:

1. The channel, bank and sensor configuration step is used to declare all the different elements. This is done in the global declaration section in the main application file. See the section associated to each element for more details.
2. The bank and sensor initialization step is used to initialize the STMTouch touch sensing library modules. The sensor parameters are initialized with their default value defined in the configuration files.
3. The bank acquisition step is composed of the following substeps:
   - configuration: used to configure all channels of the bank
   - start acquisition: used to launch the measurement on all channels of the bank
   - wait end acquisition: used to wait the end of acquisition of all channels of the bank
   - get result: used to read all the channel measurements and to store them in the channel data layer

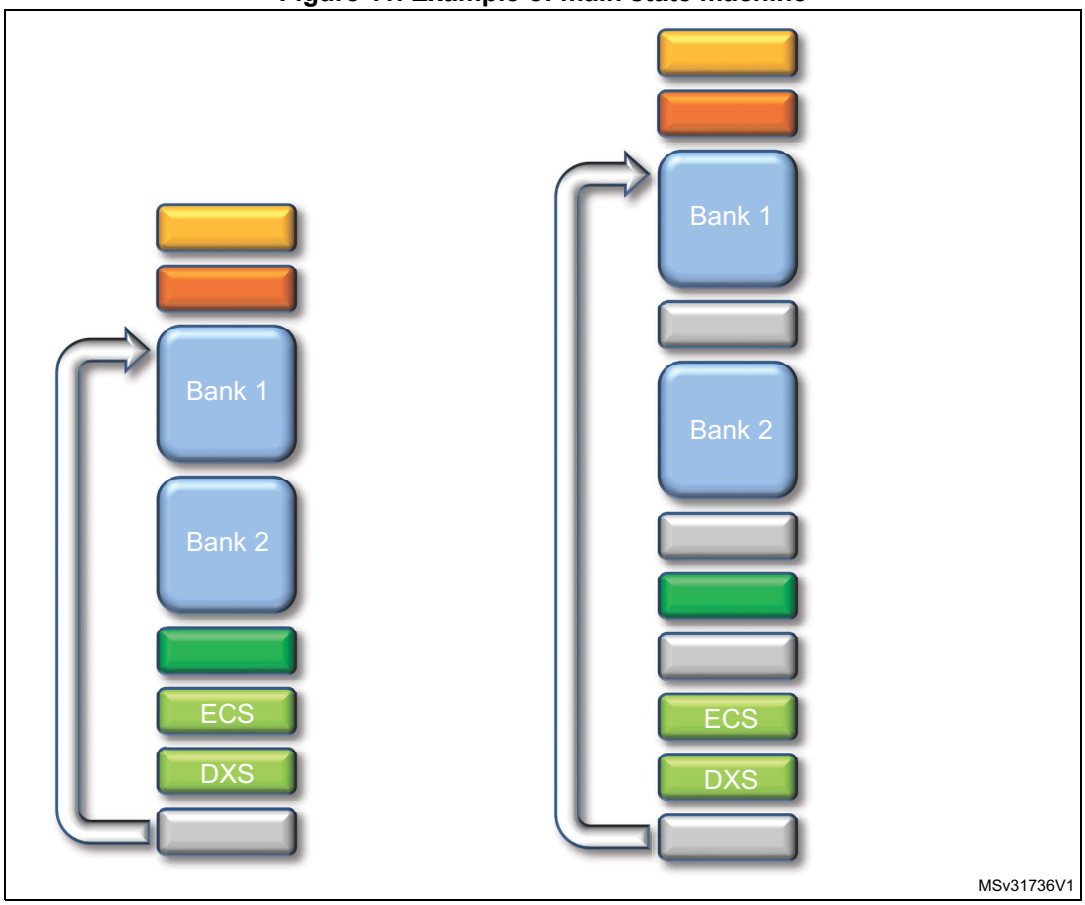
4. The sensors processing step is used to execute the state machine of the sensors.

Note: The debouncing, DTO and re-calibration are automatically performed inside this step.

5. The ECS, DXS step is used to execute other algorithms that are not performed in the sensor state machine (such as ECS, DXS, or other filters). This step is optional and can be executed at certain time intervals (mainly for ECS).
6. The user application step is used to execute the application layer (such as read the sensor state, decide actions to perform, or manage ERROR states). The user application can also be placed between other steps, for example between the sensor processing and the ECS/DXS steps.

There are various way of performing the main state machine. The following figure gives some examples with two banks.

Figure 11. Example of main state machine



3.11 Sensors state machine

3.11.1 Overview

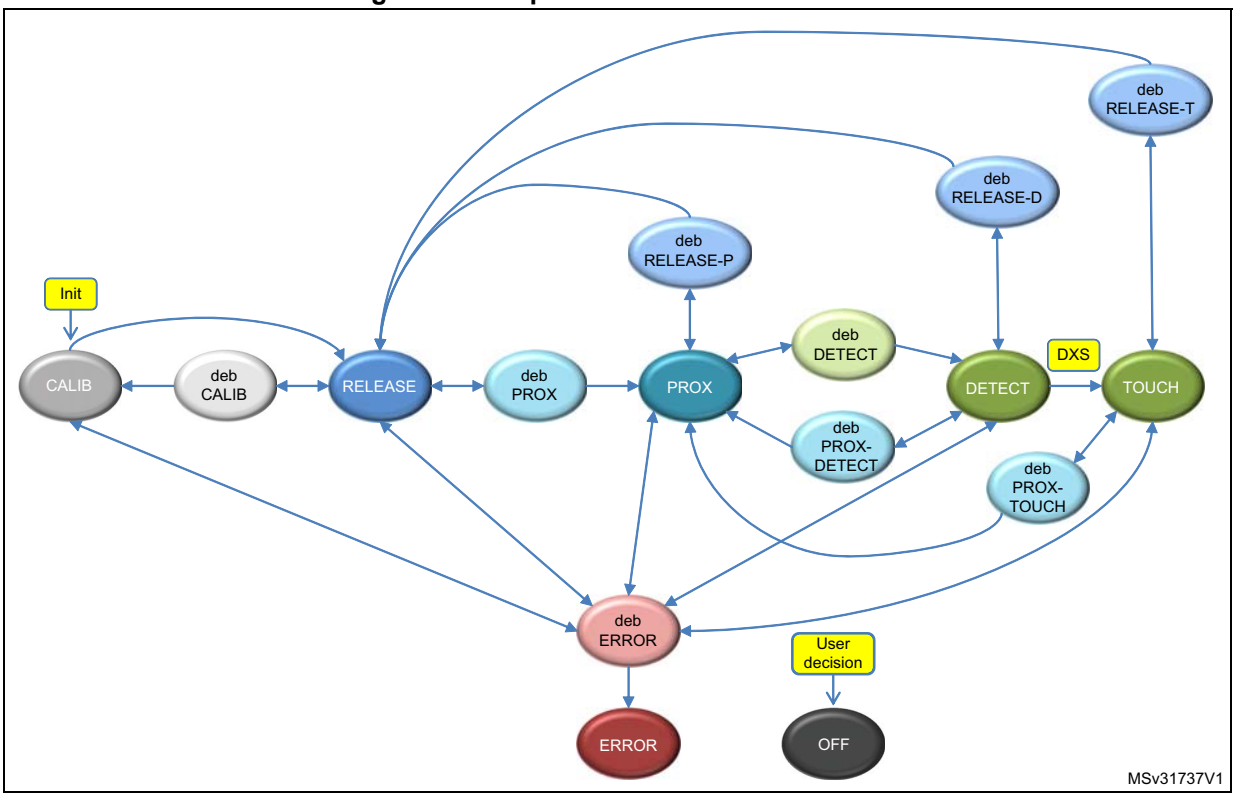
The state machine is managed in the following files:

- *tsl_touchkey.c* and *tsl_touchkey.h* for the touchkey sensors
- *tsl_linrot.c* and *tsl_linrot.h* for the linear and rotary touch sensors

There is a total of 20 states defined in the `TSL_StateId_enum_T` structure.

The following figure shows the simplified state machine used by any sensor (for clarity not all the connections between states are shown).

Figure 12. Simplified sensor state machine



3.11.2 States constant table

Each state ID is associated to a mask and a function. The association `STATE_ID-mask-function` is made in the user application code using a constant table of the `TSL_State_T` type. The name of this table is free and the user can give any name. If no function is needed, simply put a zero instead of the function name.

Example of touchkey sensor state machine:

```
// Touchkeys state machine
const TSL_State_T MyTKeys_StateMachine[] =
{
//-------------------------------------------------------------------
// ID      MASK                                FUNCTION
//-------------------------------------------------------------------
// Calibration states
/*  0 */ { TSL_STATEMASK_CALIB, TSL_tkey_CalibrationStateProcess },
/*  1 */ { TSL_STATEMASK_DEB_CALIB, TSL_tkey_DebCalibrationStateProcess },
// RELEASE states
/*  2 */ { TSL_STATEMASK_RELEASE,           TSL_tkey_ReleaseStateProcess },
#if TSLPRM_USE_PROX > 0
```

```
/*  3 */ { TSL_STATEMASK_DEB_RELEASE_PROX,
TSL_tkey_DebReleaseProxStateProcess },
#else
/*  3 */ { TSL_STATEMASK_DEB_RELEASE_PROX, 0 },
#endif
/*  4 */ { TSL_STATEMASK_DEB_RELEASE_DETECT,
TSL_tkey_DebReleaseDetectStateProcess },
/*  5 */ { TSL_STATEMASK_DEB_RELEASE_TOUCH,
TSL_tkey_DebReleaseTouchStateProcess },
#if TSLPRM_USE_PROX > 0
// Proximity states
/*  6 */ { TSL_STATEMASK_PROX,               TSL_tkey_ProxStateProcess },
/*  7 */ { TSL_STATEMASK_DEB_PROX,          TSL_tkey_DebProxStateProcess },
/*  8 */ { TSL_STATEMASK_DEB_PROX_DETECT,
TSL_tkey_DebProxDetectStateProcess },
/*  9 */ { TSL_STATEMASK_DEB_PROX_TOUCH,
TSL_tkey_DebProxTouchStateProcess },
#else
/*  6 */ { TSL_STATEMASK_PROX,               0 },
/*  7 */ { TSL_STATEMASK_DEB_PROX,           0 },
/*  8 */ { TSL_STATEMASK_DEB_PROX_DETECT,    0 },
/*  9 */ { TSL_STATEMASK_DEB_PROX_TOUCH,     0 },
#endif
// DETECT states
/* 10 */ { TSL_STATEMASK_DETECT,             TSL_tkey_DetectStateProcess },
/* 11 */ { TSL_STATEMASK_DEB_DETECT,        TSL_tkey_DebDetectStateProcess
},
// TOUCH state
/* 12 */ { TSL_STATEMASK_TOUCH,              TSL_tkey_TouchStateProcess },
// ERROR states
/* 13 */ { TSL_STATEMASK_ERROR,              MyTKeys_ErrorStateProcess },
/* 14 */ { TSL_STATEMASK_DEB_ERROR_CALIB,    TSL_tkey_DebErrorStateProcess
},
/* 15 */ { TSL_STATEMASK_DEB_ERROR_RELEASE,  TSL_tkey_DebErrorStateProcess
},
/* 16 */ { TSL_STATEMASK_DEB_ERROR_PROX,     TSL_tkey_DebErrorStateProcess
},
/* 17 */ { TSL_STATEMASK_DEB_ERROR_DETECT,   TSL_tkey_DebErrorStateProcess
},
/* 18 */ { TSL_STATEMASK_DEB_ERROR_TOUCH,    TSL_tkey_DebErrorStateProcess
},
// Other states
/* 19 */ { TSL_STATEMASK_OFF,                MyTKeys_OffStateProcess }
};
```

The STMTouch touch sensing library contains all the functions needed to manage each state. The user can also copy and adapt one or several functions to fit the application requirements.

Example:

```
/*  0 */ { TSL_STATEMASK_CALIB, MyTkeys_CalibrationStateProcess },
```

Note: The two functions used to manage the ERROR and OFF states are not part of the STMTouch touch sensing library. These functions are managed by the application.

For linear and rotary sensor state machine, it is the same principle. The functions used to manage each state start with the prefix `TSL_linrot_`:

```
CONST TSL_State_T MyLinRots_StateMachine[] =
{
// Calibration states
/*  0 */ { TSL_STATEMASK_CALIB, TSL_linrot_CalibrationStateProcess },
```

3.11.3 State details

The tables below show the detail of how each state is entered following the thresholds measured.

Table 5. Detailed sensor states 1/2

Previous state	all excepted 13	all excepted 13	2p,10p,12p,3, 4p,5p,7,8,9, 11p	2,4,11	2p,6,4p,7,8,1 1p	DXS,5	DXS,5p,9	2,2p,1	2,2p,6,10, 10p,12,12p ,0,14..18
State number	**2**	**2p**	**6**	**10**	**10p**	**12**	**12p**	**0**	**13**
Current state	**RELEASE**	**RELEASE with PROX**	**PROX**	**DETECT**	**DETECT with PROX**	**TOUCH**	**TOUCH with PROX**	**CALIB**	**ERROR**
Delta									
DETECT IN Th	deb DETECT or DETECT+DTO	deb DETECT or DETECT+DTO	deb DETECT or DETECT+DTO	same or CALIB if DTO	same or CALIB if DTO	same or CALIB if DTO	same or CALIB if DTO	RELEASE or ERROR	same
-	same	deb PROX or PROX+DTO	same or CALIB if DTO	deb RELEASE-DETECT or RELEASE	deb PROX-DETECT or PROX+DTO	deb RELEASE-TOUCH or RELEASE	deb PROX-TOUCH or PROX+DTO		
DETECT OUT Th PROX IN Th									
-		same							
PROX OUT Th			deb RELEASE-PROX or RELEASE		deb RELEASE-DETECT or RELEASE		deb RELEASE-TOUCH or RELEASE		
CALIB Th	deb CALIB or CALIB	deb CALIB or CALIB							
if ACQ ERROR	deb ERROR or ERROR	deb ERROR or ERROR	deb ERROR or ERROR	deb ERROR or ERROR	deb ERROR or ERROR	deb ERROR or ERROR	deb ERROR or ERROR	deb ERROR or ERROR	same

Table 6. Detailed sensor states 2/2

<table>
<tr><th>Previous state</th><th>6</th><th>10</th><th>10p,8</th><th>12</th><th>12p,9</th><th>2p,11p</th><th>10p</th><th>12p</th><th>2</th><th>2p,6,7</th><th>2,2p</th><th>2,2p,6,10,10p,12,12p,0</th></tr>
<tr><th>State number</th><th>3</th><th>4</th><th>4p</th><th>5</th><th>5p</th><th>7</th><th>8</th><th>9</th><th>11</th><th>11p</th><th>1</th><th>14..18</th></tr>
<tr><th>Current state</th><th>deb RELEASE-PROX</th><th>deb RELEASE-DETECT</th><th>deb RELEASE-DETECT with PROX</th><th>deb RELEASE-TOUCH</th><th>deb RELEASE-TOUCH with PROX</th><th>deb PROX</th><th>deb PROX-DETECT</th><th>deb PROX-TOUCH</th><th>deb DETECT</th><th>deb DETECT with PROX</th><th>deb CALIB</th><th>deb ERROR</th></tr>
<tr><td colspan="14">Delta</td></tr>
<tr><td>DETECT IN Th</td><td rowspan="3">PROX</td><td rowspan="3">DETECT</td><td rowspan="3">DETECT</td><td rowspan="3">TOUCH</td><td rowspan="3">TOUCH</td><td>deb DETECT or DETECT+ DTO</td><td rowspan="3">DETECT</td><td rowspan="3">TOUCH</td><td>same or DETECT+ DTO</td><td>same or DETECT+ DTO</td><td rowspan="4">RELEASE</td><td rowspan="6">RE LEASE PROX DETECT TOUCH CALIB</td></tr>
<tr><td>-</td><td rowspan="2">same or PROX+ DTO</td><td rowspan="3">RELEASE</td><td>deb PROX or PROX+ DTO</td></tr>
<tr><td>DETECT OUT Th PROX IN Th</td><td rowspan="3">same or RELEASE</td><td rowspan="2">PROX</td><td rowspan="3">same or RELEASE</td><td rowspan="2">PROX</td><td rowspan="2">same or PROX+ DTO</td><td rowspan="2">same or PROX+ DTO</td></tr>
<tr><td>-</td><td rowspan="3">RELEASE</td></tr>
<tr><td>PROX OUT Th</td><td rowspan="2">same or RELEASE</td><td rowspan="2">same or RELEASE</td><td rowspan="2">deb RELEASE-DETECT or RELEASE</td><td rowspan="2">deb RELEASE-TOUCH or RELEASE</td><td rowspan="2">RELEASE</td></tr>
<tr><td>CALIB Th</td><td>same or CALIB</td></tr>
<tr><td>if ACQ ERROR</td><td>PROX</td><td>DETECT</td><td>DETECT</td><td>TOUCH</td><td>TOUCH</td><td>RELEASE</td><td>DETECT</td><td>TOUCH</td><td>RELEASE</td><td>RELEASE</td><td>RELEASE</td><td>ERROR</td></tr>
</table>

3.11.4 CALIBRATION state

It consists in calculating the reference for all the channels of a sensor. An average of some measurements is done.

The number of measurement samples to use for the calibration is defined by the `TSLPRM_CALIB_SAMPLES` parameter.

After reset, the initialization method of each object is called. This method initializes the sensor parameters and then goes in the CALIBRATION state. After the calibration is done, the sensor goes in RELEASE or ERROR state.

Related functions:

- `TSL_tkey_CalibrationStateProcess()`
- `TSL_linrot_CalibrationStateProcess()`
- `TSL_tkey_SetStateCalibration()`
- `TSL_linrot_SetStateCalibration()`

Calibration delay

If a noise filter is used, it is necessary to wait a certain amount of measurement samples before to start the reference calculation. This number of samples to wait is defined by the `TSLPRM_CALIB_DELAY` parameter.

Re-calibration

If the calibration threshold is reached while in RELEASE state, a new calibration is performed. This re-calibration prevents the application to get stuck if something touches permanently the sensor, like a drop of water for example or if the sensor is touched upon power-on.

3.11.5 RELEASE state

Corresponds to the idle state of the sensor when no presence is detected.

Related functions:

- `TSL_tkey_ReleaseStateProcess()`
- `TSL_linrot_ReleaseStateProcess()`

3.11.6 PROXIMITY state

This state is optional and is enabled or disabled using the `TSLPRM_USE_PROX` parameter.

Related functions:

- `TSL_tkey_ProxStateProcess()`
- `TSL_linrot_ProxStateProcess()`

3.11.7 DETECT state

It is the “normal” state when the sensor is touched.

Related functions:

- `TSL_tkey_DetectStateProcess()`
- `TSL_linrot_DetectStateProcess()`

3.11.8 TOUCH state

Same as DETECT state, except that this state is entered only by the DXS processing. If the DXS is not used, this state is never entered.

Related functions:

- `TSL_tkey_TouchStateProcess()`
- `TSL_linrot_TouchStateProcess()`

3.11.9 ERROR state

It is used to catch all acquisition errors detected in the other states. The management of this state must be performed at application level.

3.11.10 OFF state

It is used to inform the acquisition module to stop the burst and/or acquisition on the sensor channels. The management of this state must be performed at application level.

3.11.11 DEBOUNCE state

The debounce is optional and is enabled/disabled using the different debounce counters parameters: `TSLPRM_DEBOUNCE_PROX`, `TSLPRM_DEBOUNCE_DETECT`, `TSLPRM_DEBOUNCE_RELEASE`, `TSLPRM_DEBOUNCE_CALIB`, `TSLPRM_DEBOUNCE_ERROR`

The debounce is off if the corresponding parameter is equal to zero.

3.11.12 Reading the current state

The current state can be obtained by using the following functions:

- For touchkey sensor:
  - `TSL_tkey_GetStateId()`
  - `TSL_tkey_GetStateMask()`
- For linear and rotary sensor:
  - `TSL_linrot_GetStateId()`
  - `TSL_linrot_GetStateMask()`

`TSL_tkey_IsChanged()` or `TSL_linrot_IsChanged()` are used to check if a sensor state has changed.

The state can also directly be read inside the sensor data structure:

```
if MyTKeys[0].p_Data->StateId == TSL_STATEID_DETECT)
```

3.11.13 Enabling a specific state

It is possible to enter directly in the CALIBRATION, OFF, and OFF-with-burst-only states. The OFF-with-burst-only state consists in only bursting the electrode without performing acquisition on it. It can be used in specific cases to improve the robustness against noise or to keep optimum sensor sensitivity.

This is done by using the following functions:

- For touchkey sensor:
  - `TSL_tkey_SetStateCalibration()`
  - `TSL_tkey_SetStateOff()`
  - `TSL_tkey_SetStateBurstOnly()`
- For linear and rotary sensor:
  - `TSL_linrot_SetStateCalibration()`
  - `TSL_linrot_SetStateOff()`
  - `TSL_linrot_SetStateBurstOnly()`

3.12 Environment change system (ECS)

3.12.1 Principle

Power supply voltage, temperature, and air humidity may induce a slow variation of the measured signal. The environment change system (ECS) is used to adapt the reference to these environment changes.

The ECS processing is based on an infinite response digital low-pass filter of the first order (IIR filter):

$$Y(n) = K \times X(n) + (1 - K) \times Y(n - 1)$$

with:

- Y = reference
- X = acquisition value (last measurement)
- K = coefficient

The higher value for K, the faster the response time is. Two default K coefficients are available to obtain fast and slow responses.

The sampling frequency is programmable using a timing utility routine (see example below).

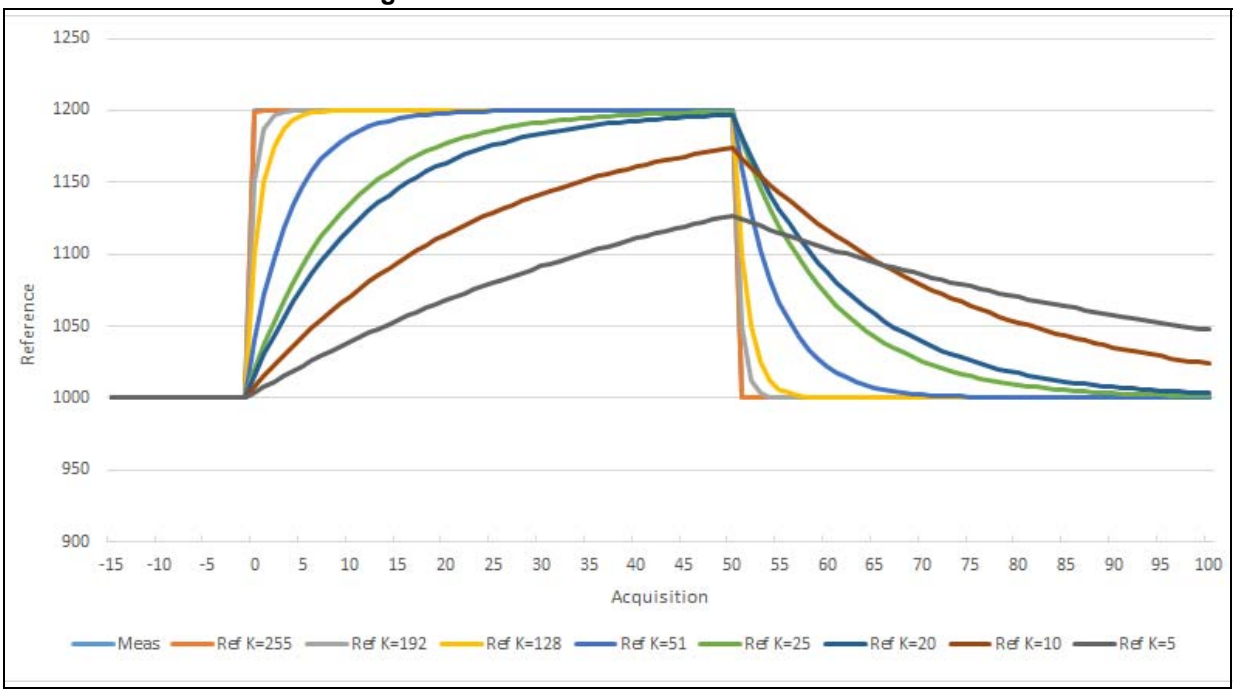
If the sensor is in PROX, DETECT or TOUCH states, the ECS is disabled for the duration of the detection timeout or for the duration of the touch (whichever ends first).

When the ECS is disabled, Yn = Yn - 1.

As soon as the recalibration times out or the detection ends, the filter is set active again.

The figure below shows the K filter effect to staircase on reference.The horizontal axis represents the number of calls to `ECS_process()` API. So the response time depends on the ECS_DELAY.

Figure 13. Reference versus K filter value



3.12.2 Resources

The ECS functions are provided in the files:

- *tsl_ecs.c*
- *tsl_ecs.h*

The functions are:

- `TSL_ecs_Process()`: main function to be used by the user
- `TSL_ecs_CalcK()`: additional function
- `TSL_ecs_ProcessK()`: additional function

3.12.3 Parameters

- `TSLPRM_ECS_K_FAST`
- `TSLPRM_ECS_K_SLOW`
- `TSLPRM_ECS_DELAY`

3.12.4 Use example

The ECS processing is usually performed in the main state machine at regular time intervals defined by the user. But it can also be done in interrupt routines. It must be performed after the sensor state machine is processed.

The ECS is activated only when all the sensors are in RELEASE, ERROR, or OFF states, with at least one sensor in RELEASE state. It can also be delayed from milliseconds to few seconds.

The ECS processing is performed on a group of sensors defined by the user. Different groups can be created. The ECS can be applied on these groups with different K coefficients. The user decides the best thing to do for the application.

The simplest way is to call the `TSL_ecs_Process()` function in the main application loop, using the default K coefficients defined in the configuration file: *TSL_ecs_Process(&MyObjGroup)*.

To call this function at regular time intervals, the timing routine `TSL_ecs_Process()` can be used.

Example with ECS executed every 100 ms:

```
TSL_tTick_ms_T time_ECS_tick;
int main(void) {
  while (1) {
    ...
    // ECS every 100 ms
    if (TSL_tim_CheckDelay_ms(100, &time_ECS_tick) == TSL_STATUS_OK)
    {
      TSL_ecs_Process(&MyObjGroup);
    }
    ...
  }
}
```

The `TSL_ecs_Process()` function allows the use of a K coefficient different than the default value:

```
    if (TSL_tim_CheckDelay_ms(100, &time_ECS_tick) == TSL_STATUS_OK)
    {
      if ((MyObjGroup->StateMask & TSL_STATE_RELEASE_BIT_MASK) &&
         !(MyObjGroup->StateMask & TSL_STATEMASK_ACTIVE))
      {
        TSL_ecs_ProcessK(&MyObjGroup, 120);
      }
    }
```

To update `TSL_ecs_Process()`, the system tick handler must be updated as follows:

```
void SysTick_Handler(void)
{
  /* USER CODE BEGIN SysTick_IRQn 0 */

  /* USER CODE END SysTick_IRQn 0 */
  HAL_IncTick();
  HAL_SYSTICK_IRQHandler();
  /* USER CODE BEGIN SysTick_IRQn 1 */
```

```
  // TSL timing for ECS, DTO, ...
  TSL_tim_ProcessIT();

  /* USER CODE END SysTick_IRQn 1 */
}
```

3.13 Detection exclusion system (DXS)

3.13.1 Principle

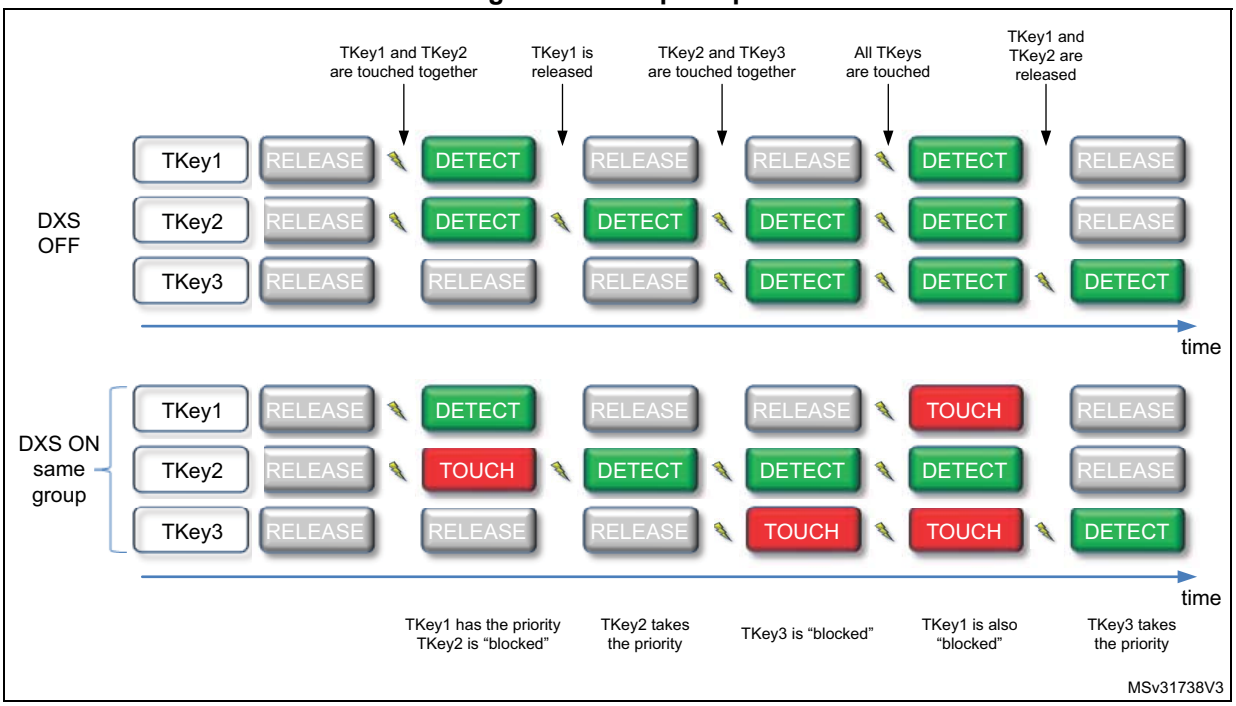
The DXS processing is used to prevent several sensors to be in the DETECT state at the same time. This may happen if the sensors are close to each other or if their sensitivity is too high. This can be useful also in some applications to prevent the user to touch at the same time several sensors with “opposite” meaning (volume up and volume down for example).

The first sensor in the group of sensors has the priority and enters in the DETECT state (with the `DxSLock` flag set). The other sensors are “blocked” and enter instead in the TOUCH state.

Note: A particular care must be taken when designing sensors that are shared between multiple DXS groups. The sensor that is assigned in the DETECT state depends on the sensors position in the DXS groups, and also on the order of the DXS groups processing. See the examples 1 and 2 below for more detail.

The figure below illustrates the difference in behavior for a group of three sensors (touchkeys), part of the same DXS group, when the DXS is off and on (touchkeys can be replaced by linear or rotary sensors).

Figure 14. DXS principle



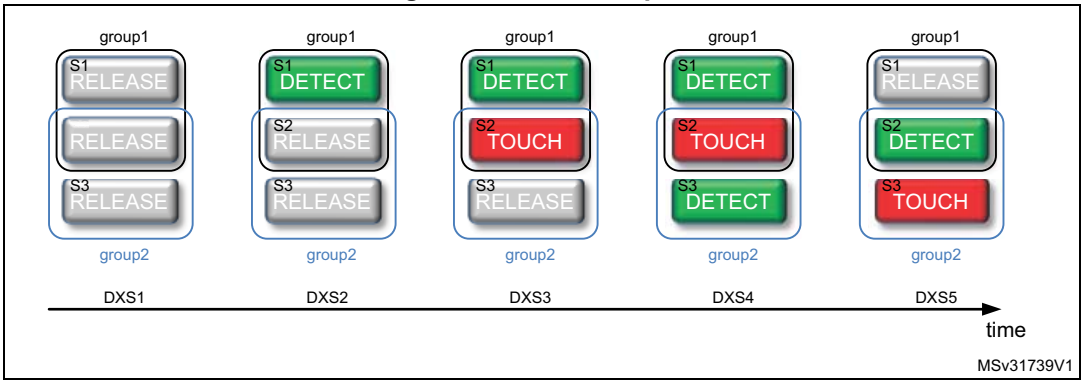
Example 1: Three sensors with one shared between two groups

In this example, the group1 is composed of the two sensors S1 and S2 in this order, and the group2 of the two sensors S2 and S3 in this order.

The DXS groups are processed in this order: group1 first and then group2.

In the DXS5 step, S2 goes in DETECT state instead of S3. This is simply because S2 is placed first in the group2.

Figure 15. DXS example 1



Example 2: Four sensors with one shared between three groups

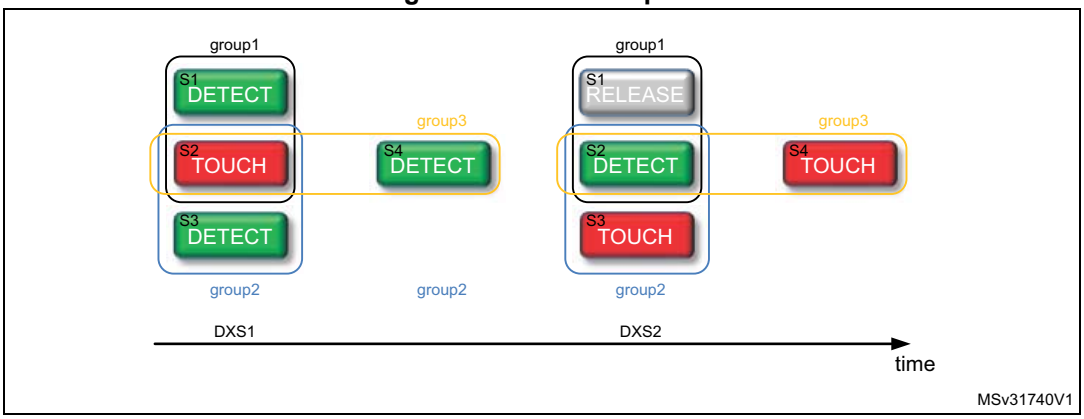
In this example, the group1 is composed of the two sensors S1 and S2 in this order, the group2 of the two sensors S2 and S3 in this order, and the group3 of the two sensors S2 and S4 in this order.

The DXS groups are processed in this order: group1 first, then group2 and finally group3.

In the DXS2 step, S2 takes the priority over S3 and S4.

To summarize, the decision to be in DETECT state depends on the sensors placement inside the group and also on the order of the groups processing.

Figure 16. DXS example 2



3.13.2 Resources

The DXS functions are provided in the files:

- *tsl_dxs.c*
- *tsl_dxs.h*

The function to use is `TSL_dxs_FirstObj()`.

3.13.3 Parameter

- `TSLPRM_USE_DXS`

3.13.4 Use example

The DXS processing is performed usually in the main state machine but it can also be done in interrupt routines.

Warning: The DXS must be absolutely performed after the sensors state machine is processed, that is after the call to the `TSL_obj_GroupProcess()` function (see the main state machine for more details).

The DXS processing is performed on a group of sensors defined by the user. Different groups of DXS can be created.

It is up to the user to decide the best partitioning for the application.

Example:

```
int main(void) {
  while (1) {
    ...
    TSL_obj_GroupProcess(&MyObjGroup1);
    TSL_obj_GroupProcess(&MyObjGroup2);
    TSL_dxs_FirstObj(&MyObjGroup1);
    TSL_dxs_FirstObj(&MyObjGroup2);
    ...
  }
}
```

3.14 Detection time out (DTO)

3.14.1 Principle

The detection time out (DTO) introduces a simple way to cope with water film and any obstacle that may come in contact with a sensor. It introduces a maximum duration for the DETECTED state of any sensor called the DTO.

After this period of time, the sensor is automatically re-calibrated. This allows the sensor to be made touch sensitive again, even if the obstacle or the liquid film is still present on the application front panel.

This feature is application dependent and the time out must be tuned according to the user interface specifications.

The DTO is applied on the PROX, DETECT and TOUCH states and can be disabled.

3.14.2 Resources

The DTO functions are provided in the files:

- *tsl_touchkey.c*
- *tsl_touchkey.h*
- *tsl_linrot.c*
- *tsl_linrot.h*

The functions used by the DTO are:

- `TSL_tkey_DTOGetTime()`
- `TSL_linrot_DTOGetTime()`
- `TSL_tim_CheckDelay_sec()`

Note: The user does not need to call these functions to perform the DTO.

3.14.3 Parameter

- `TSLPRM_DTO`

3.14.4 Example

The DTO is automatically performed inside the sensor state machine. The user does not need to call any function in the application code.

The DTO is disabled by writing zero in the `TSLPRM_DTO` parameter.

3.15 Noise filters

3.15.1 Principle

The STMTouch touch sensing library has been designed to facilitate the implementation of different noise filters. These filters can be used for many purpose and can range from very simple design to very complicated.

3.15.2 Resources

The filters are defined in the files:

- *tsl_filter.c*
- *tsl_filter.h*

Each filter is described by a function:

- `TSL_filt_MeasFilter()`: filter on measurement values
- `TSL_filt_DeltaFilter()`: filter on delta values

3.15.3 Parameter

There is no parameter for the filter module.

3.15.4 Example

The filter functions can be called at anytime in the main application. In order to speed up the execution time and to gain RAM space, the measure and delta filters are called by the `TSL_acq_BankGetResult()` function.

Examples:

```
// Apply a filter on the measures only
TSL_acq_BankGetResult(0, TSL_filt_MeasFilter, 0);
// Get the measures without applying any filter
TSL_acq_BankGetResult(0, 0, 0);
```

Note: The user can also create the application filter functions.

3.16 Timing management

3.16.1 Principle

The STMTouch touch sensing library needs an internal clock (timing), in particular for the ECS and DTO processing.

The timing process consists to increment a global variable at a regular interval. Different functions are then used to compare the current “time” and to check if a certain delay has elapsed.

The SysTick is used as timebase for the STMTouch touch sensing library. Its initialization must be done in the user code layer. Usually it is already done by the `HAL_Init` function. The `TSLPRM_TICK_FREQ` parameter must be set accordingly.

3.16.2 Resources

The common timing routines are described in the files:

- *tsl_time.c*
- *tsl_time.h*

Functions are:

- `TSL_tim_ProcessIT()`
- `TSL_tim_CheckDelay_ms()`
- `TSL_tim_CheckDelay_sec()`

3.16.3 Parameter

- `TSLPRM_TICK_FREQ`: the value must be in line with the SysTick frequency that is initialized in the user code.

3.16.4 Example

The `TSL_tim_CheckDelay_ms()` function can be used in the main application code to execute some code (for example the ECS) at a regular interval.

Example:

```
TSL_tTick_ms_T time_ECS_tick;
TSL_tTick_ms_T time_LED_tick;
int main(void) {
  TSL_Init(MyBanks); // The timing starts...
  while (1) {
    ...
    // Launch the ECS every 100 ms
    if (TSL_tim_CheckDelay_ms(100, &time_ECS_tick) == TSL_STATUS_OK)
    {
      TSL_ecs_Process(&MyObjGroup);
    }
    // Toggle LED every 500 ms
    if (TSL_tim_CheckDelay_ms(500, &time_LED_tick) == TSL_STATUS_OK)
    {
      ToggleLED();
    }
    ...
  }
}
```

3.17 Parameters

All the parameters are described in the *tsl_conf.h* file.

Note: The tsl_conf_<XXX>_template.h file present in the STM32_TouchSensing_Library/inc folder must be copied in the application project inc/tsl_conf.h and adapted to the application (number of channels, banks, debounce, or DTO for example).

The structure `TSL_Params_T` *is used* to hold some parameters that are common to all sensors. These parameters can be changed by the user while the application is running.

All common parameters are verified (presence and value range) in the *tsl_check_config.h* file.

All device specific parameters are verified in the *tsl_check_config_<XXX>.h* file.

3.18 Sensors acquisition timings

Reminders regarding terminology used in this section are given below:

- Bank: set of channels acquired simultaneously belonging to different groups
- Channel: elementary acquisition item (a GPIO from a group connected to a sensor)
- Group (also known as TSC group): set of up to four GPIOs defined as one sampling capacitor (Cs) with up to three channels or an active shield

- Shield: set of track and hatched plane used to increase noise robustness
- Active shield: a channel connected to an hatched plane driven simultaneously to sensor channels belonging to the same bank. An active shield requires its own sampling capacitor.
- Passive shield: a plane preferably hatched connected to the ground

This section details sequences for system with the following configurations:

- Touchkey sensor only
- Linear or rotary sensor only
- Mixed configuration using touchkey, and/or linear, and/or rotary sensors

3.18.1 Acquisition timing using touchkey sensors

The examples detailed in the tables below summarize dependencies between group, bank, sensor number, and acquisition timing, for STM32F0, STM32L4, and STM32WB series.

For example:

- Three sensors can be acquired in:
  - 1 time using 3 groups and 1 bank
  - 2 times using 2 groups and 2 banks
  - 3 times using 1 groups and 3 banks
- Five sensors can be acquired in:
  - 1 time using 5 groups and 1 bank
  - 2 times using 3 groups and 2 banks
  - 3 times using 2 groups and 3 banks
- Six sensors can be acquired in:
  - 1 time using 6 groups and 1 bank
  - 2 times using 3 groups and 2 banks
  - 3 times using 2 groups and 3 banks

Note: In Table 8, Table 11, Table 14, Table 16, Table 18, Table 21, and Table 24 below, an empty cell indicates that no event is reported for the corresponding bank, channel, or shield during the corresponding period.

Using up to three touchkey sensors on the same group

As a group is able to handle up to three channels, this use case can be handled using only one group. An extra group can also be added for the shield.

The main advantages of this configuration are the following:

- reduced number of used GPIOs
- only one sampling capacitor required for three touchkeys

Table 7 gives an example based on G1 and G2 and *Table 8* details the corresponding acquisition time line.

Table 7. Example based on G1/G2 (3 touchkey sensors)

Group	I/O	Channel[1]	Sampling	Touchkey	Bank
G1	IO1	-	CS = 22 nF	-	-
	IO2	CH1	-	K1	B1
	IO3	CH2	-	K2	B2
	IO4	CH3	-	K3	B3
G2	IO1	-	CS = 47 nF	-	-
	IO2	SHIELD	-	-	-
	IO3	-	-	-	-
	IO4	-	-	-	-

1. Green is used for CHx and blue for SHIELD.

Table 8. Example G1/G2 (3 touchkey sensors) - Acquisition time line

Time	T0	T1	T2	T3	T4	T5	T6	T7	T8
B1/B2/B3	↑(1) P1(3) I(4)	P2 I	P3 I eoa(2)	↑ P1 I	P2 I	P3 I eoa	↑ P1 I	P2 I	P3 I eoa
K1/CH1	Γ(5)			Γ			Γ		
K2/CH2		Γ			Γ			Γ	
K3/CH3			Γ			Γ			Γ
SHIELD	Γ	Γ	Γ	Γ	Γ	Γ	Γ	Γ	Γ

1. ↑ = start of bank Bx acquisition.
2. eoa = end of Bx acquisition.
3. Px = program groups and channels for Bank x.
4. I = interrupt.
5. Γ = charge transfer cycle (green for CHx and blue for SHIELD).

The table below provides a list of features for this example. This is the most simple use case. In this case, the sensor acquisition is sequential.

Table 9. Example G1/G2 (3 touchkey sensors) - Synthesis

Features	Value
Sampling capacitance	1 + 1 (shield)
Pins without shield	4
Pins with shield	6
Cost	Low
Response time	3 * T0

Using up to three touchkey sensors on three groups

To speed up the acquisition, the configuration one group per touchkey is better. An extra group can also be added for the shield.

Table 10 gives an example based on G1, G2, G3 and G4 (3 touchkey sensors) and *Table 11* details the corresponding acquisition time line.

Table 10. Example based on G1/G2/G3/G4 (3 touchkey sensors)

Group	I/O	Channel(1)	Sampling	Touchkey	Bank
G1	IO1	-	CS = 22 nF	-	-
	IO2	CH1	-	K1	B1
	IO3	-	-	-	-
	IO4	-	-	-	-
G2	IO1	-	CS = 22 nF	-	-
	IO2	CH2	-	K2	B1
	IO3	-	-	-	-
	IO4	-	-	-	-
G3	IO1	-	CS = 22 nF	-	-
	IO2	CH3	-	K3	B1
	IO3	-	-	-	-
	IO4	-	-	-	-
G4	IO1	-	CS = 47 nF	-	-
	IO2	SHIELD	-	-	-
	IO3	-	-	-	-
	IO4	-	-	-	-

1. Green is used for CHx and blue for SHIELD.

Table 11. Example G1/G2/G3/G4 (3 touchkey sensors) - Acquisition time line

Time	T0	T1	T2	T3	T4	T5	T6	T7	T8
B1	↑(1) eoa(2)	↑ eoa	↑ eoa	↑ eoa	↑ eoa	↑ eoa	↑ eoa	↑ eoa	↑ eoa
	P1(3) I(4)	P1 I	P1 I	P1 I	P1 I	P1 I	P1 I	P1 I	P1 I
K1/CH1	Γ(5)	Γ	Γ	Γ	Γ	Γ	Γ	Γ	Γ
K2/CH2	Γ	Γ	Γ	Γ	Γ	Γ	Γ	Γ	Γ
K3/CH3	Γ	Γ	Γ	Γ	Γ	Γ	Γ	Γ	Γ
SHIELD	Γ	Γ	Γ	Γ	Γ	Γ	Γ	Γ	Γ

1. ↑ = start of bank Bx acquisition.
2. eoa = end of Bx acquisition.
3. Px = program groups and channels for Bank x.
4. I = interrupt.

5. Γ = charge transfer cycle (green for CHx and blue for SHIELD).

The table below provides a list of features for this example. This is the way to acquire all sensors at the same time.

Table 12. Example G1/G2/G3/G4 (3 touchkey sensors) - Synthesis

Feature	Value
Sampling capacitance	3+ 1 (shield)
Pins without shield	6
Pins with shield	8
Cost	Medium
Response time	1 * T0

Note: The pins not used inside a group (such as IO3/4 in the example above) must not be used for analog features.

Using more than three touchkey sensors

To handle more than three touchkey sensors, more than one group is needed as the maximum number of channels per group is three. An extra group can also be used for the shield.

Table 13 gives an example based on G1, G2, G3 and G4 with 9 touchkey sensors and *Table 14* details the corresponding acquisition time line.

Table 13. Example based on G1/G2/G3/G4 (9 touchkey sensors)

Group	I/O	Channel(1)	Sampling	Key	Bank
G1	IO1	-	CS = 22 nF	-	-
	IO2	CH1	-	K1	B1
	IO3	CH2	-	K2	B2
	IO4	CH3	-	K3	B3
G2	IO1	-	CS = 22 nF	-	-
	IO2	CH4	-	K4	B1
	IO3	CH5	-	K5	B2
	IO4	CH6	-	K6	B3
G3	IO1	-	CS = 22 nF	-	-
	IO2	CH7	-	K7	B1
	IO3	CH8	-	K8	B2
	IO4	CH9	-	K9	B3

Table 13. Example based on G1/G2/G3/G4 (9 touchkey sensors) (continued)

Group	I/O	Channel(1)	Sampling	Key	Bank
G4	IO1	-	CS = 47 nF	-	-
	IO2	SHIELD	-	-	-
	IO3	-	-	-	-
	IO4	-	-	-	-

1. Green is used for CHx and blue for SHIELD.

Table 14. Example G1/G2/G3/G4 (9 touchkey sensors) - Acquisition time line

Time	T0	T1	T2	T3	T4	T5	T6	T7	T8
B1/B2/B3	↑(1) P1(3) I(4)	P2 I	P3 I	eoa(2) ↑ P1 I	P2 I	P3 I	eoa ↑ P1 I	P2 I	P3 I eoa
K1/CH1	Γ(5)			Γ			Γ		
K2/CH2		Γ			Γ			Γ	
K3/CH3			Γ			Γ			Γ
K4/CH4	Γ			Γ			Γ		
K5/CH5		Γ			Γ			Γ	
K6/CH6			Γ			Γ			Γ
K7/CH7	Γ			Γ			Γ		
K8/CH8		Γ			Γ			Γ	
K9/CH9			Γ			Γ			Γ
SHIELD	Γ	Γ	Γ	Γ	Γ	Γ	Γ	Γ	Γ

1. ↑ = start of bank Bx acquisition.
2. eoa = end of Bx acquisition.
3. Px = program groups and channels for Bank x.
4. I = interrupt.
5. Γ = charge transfer cycle (green for CHx and blue for SHIELD).

The table below provides a list of features for this example. This is the most complex use case.

Table 15. Example G1/G2/G3/G4 (9 keys)- Synthesis

Feature	Value
Sampling capacitance	3+ 1 (shield)
Pins without shield	12
Pins with shield	14
Cost	Medium
Response time	3 * T0

Using only few specific touchkey sensors

If only some touchkey sensors must be acquired, the bank helps to start the acquisition only on these specific touchkeys.

The table below shows an example of the acquisition time line in this case.

Table 16. Example G1/G2/G3/G4 (specific touch key sensors only) - Acquisition time line

Time	T0	T1	T2	T3	T4	T5	T6	T7	T8
B1/B2/B3	↑(1)eoa(2) P1(3) I(4)		↑ eoa P1 I	↑ eoa P1 I	↑ eoa P2 I	P3 I	↑ eoa P1 I	P2 I	P3 I
K1/CH1	Γ(5)		Γ	Γ			Γ		
K2/CH2					Γ			Γ	
K3/CH3						Γ			Γ
K4/CH4	Γ		Γ	Γ			Γ		
K5/CH5					Γ			Γ	
K6/CH6						Γ			Γ
K7/CH7	Γ		Γ	Γ			Γ		
K8/CH8					Γ			Γ	
K9/CH9						Γ			Γ
SHIELD	Γ		Γ	Γ	Γ	Γ	Γ	Γ	Γ

1. ↑ = start of bank Bx acquisition.
2. eoa = end of Bx acquisition.
3. Px = program groups and channels for Bank x.
4. I = interrupt.
5. Γ = charge transfer cycle (green for CHx and blue for SHIELD).

3.18.2 Acquisition timing using linear or rotary sensors

To handle linear or rotary sensors in a proper way regarding sensitivity, it is recommended to split linear and rotary channels on various groups.

For example, three groups needed to handle three linear/rotary channels.

Using one linear/rotary sensor

Three groups are used to handle the three channels. An extra group can also be used for the shield.

Table 17 gives an example based on G1, G2, G3 and G4 (one linear sensor) and *Table 18* details the corresponding acquisition time line.

Table 17. Example based on G1/G2/G3/G4 (1 linear sensor)

Group	I/O	Channel[1]	Sampling	Linear	Bank
G1	IO1	-	CS = 47 nF	-	-
	IO2	CH1	-	S1	B1
	IO3	-	-	-	-
	IO4	-	-	-	-
G2	IO1	-	CS = 47 nF	-	-
	IO2	CH2	-	S1	B1
	IO3	-	-	-	-
	IO4	-	-	-	-
G3	IO1	-	CS = 47 nF	-	-
	IO2	CH3	-	S1	B1
	IO3	-	-	-	-
	IO4	-	-	-	-
G4	IO1	-	CS = 47 nF	-	-
	IO2	SHIELD	-	-	-
	IO3	-	-	-	-
	IO4	-	-	-	-

1. Green is used for CHx and blue for SHIELD.

Table 18. Example G1/G2/G3/G4 (1 sensor) - Acquisition time line

Time	T0	T1	T2	T3	T4	T5	T6	T7	T8
B1	↑[1] eoa[2] P1[3] I[4]	↑ eoa P1 I	↑ eoa P1 I	↑ eoa P1 I	↑ eoa P1 I	↑ eoa P1 I	↑ eoa P1 I	↑ eoa P1 I	↑ eoa P1 I
S1/CH1	Γ[5]	Γ	Γ	Γ	Γ	Γ	Γ	Γ	Γ
S1/CH2	Γ	Γ	Γ	Γ	Γ	Γ	Γ	Γ	Γ
S1/CH3	Γ	Γ	Γ	Γ	Γ	Γ	Γ	Γ	Γ
SHIELD	Γ	Γ	Γ	Γ	Γ	Γ	Γ	Γ	Γ

1. ↑ = start of bank Bx acquisition.
2. eoa = end of Bx acquisition.
3. Px = program groups and channels for Bank x.
4. I = interrupt.
5. Γ = charge transfer cycle (green for CHx and blue for SHIELD).

The table below provides a list of features for this example.

Table 19. Example G1/G2/G3/G4 (1 sensor) - Synthesis

Feature	Value
Sampling capacitance	3+ 1 (shield)
Pins without shield	6
Pins with shield	8
Cost	Medium
Response time	1 * T0

Using two linear and one rotary sensors

The three sensors allows the handling of nine channels.

Table 20 gives an example based on G1, G2, G3 and G4 (three sensors) and *Table 21* details the corresponding acquisition time line.

Table 20. Example based on G1/G2/G3/G4 (3 L/R sensors)

Group	I/O	Channel(1)	Sampling	Linear/Rotary	Bank
G1	IO1	-	CS = 47 nF	-	-
	IO2	CH1	-	S1	B1
	IO3	CH4	-	S2	B2
	IO4	CH7	-	R1	B3
G2	IO1	-	CS = 47 nF	-	-
	IO2	CH2	-	S1	B1
	IO3	CH5	-	S2	B2
	IO4	CH8	-	R1	B3
G3	IO1	-	CS = 47 nF	-	-
	IO2	CH3	-	S1	B1
	IO3	CH6	-	S2	B2
	IO4	CH9	-	R1	B3
G4	IO1	-	CS = 47 nF	-	-
	IO2	SHIELD	-	-	-
	IO3	-	-	-	-
	IO4	-	-	-	-

1. Green is used for CHx and blue for SHIELD.

Table 21. Example G1/G2/G3/G4 (3 L/R sensors) - Acquisition time line

Time	T0	T1	T2	T3	T4	T5	T6	T7	T8
B1	↑(1)eoa(2) P1(3) I(4)			↑ eoa P1 I			↑ eoa P1 I		
B2		↑ eoa P2 I			↑ eoa P2 I			↑ eoa P2 I	
B3			↑ eoa P3 I			↑ eoa P3 I			↑ eoa P3 I
S1/CH1	Γ(5)			Γ			Γ		
S1/CH2	Γ			Γ			Γ		
S1/CH3	Γ			Γ			Γ		
S2/CH4		Γ			Γ			Γ	
S2/CH5		Γ			Γ			Γ	
S2/CH6		Γ			Γ			Γ	
R1/CH7			Γ			Γ			Γ
R1/CH8			Γ			Γ			Γ
R1/CH9			Γ			Γ			Γ
SHIELD	Γ	Γ	Γ	Γ	Γ	Γ	Γ	Γ	Γ

1. ↑ = start of bank Bx acquisition.
2. eoa = end of Bx acquisition.
3. Px = program groups and channels for Bank x.
4. I = interrupt.
5. Γ = charge transfer cycle (green for CHx and blue for SHIELD).

The table below provides a list of features for this example.

Table 22. Example G1/G2/G3/G4 (3 L/R sensors) - Synthesis

Feature	Value
Sampling capacitance	3+ 1 (shield)
Pins without shield	12
Pins with shield	14
Cost	Medium
Response time	3 * T0

3.18.3 Acquisition timing using touchkey, linear and rotary sensors

Table 23 gives an example based on G1, G2, G3 and G4, with three touchkey, one linear and one rotary sensors and *Table 24* details the corresponding acquisition time line.

Table 23. Example based on G1, G2, G3 and G4 (3 touchkey, 1 linear, 1 rotary sensors)

Group	I/O	Channel(1)	Sampling	Touchkey	Linear/Rotary	Bank
G1	IO1	-	CS = 47 nF	-	-	-
	IO2	CH1	-	-	S1	B1
	IO3	CH4	-	K1	-	B2
	IO4	CH7	-	-	R1	B3
G2	IO1	-	CS = 47 nF	-	-	-
	IO2	CH2	-	-	S1	B1
	IO3	CH5	-	K2	-	B2
	IO4	CH8	-	-	R1	B3
G3	IO1	-	CS = 47 nF	-	-	-
	IO2	CH3	-	-	S1	B1
	IO3	CH6	-	K3	-	B2
	IO4	CH9	-	-	R1	B3
G4	IO1	-	CS = 47 nF	-	-	-
	IO2	SHIELD	-	-	-	-
	IO3	-	-	-	-	-
	IO4	-	-	-	-	-

1. Green is used for CHx and blue for SHIELD.

Table 24. Example G1/G2/G3/G4 (3 touchkey, 1 linear, 1 rotary sensors) - Acquisition time line

Time	T0	T1	T2	T3	T4	T5	T6	T7	T8
B1	↑(1)eoa(2) P1(3) I(4)			↑ eoa P1 I			↑ eoa P1 I		
B2		↑ eoa P2 I			↑ eoa P2 I			↑ eoa P2 I	
B3			↑ eoa P3 I			↑ eoa P3 I			↑ eoa P3 I
S1/CH1	Γ(5)			Γ			Γ		
S1/CH2	Γ			Γ			Γ		
S1CH3	Γ			Γ			Γ		
K1/CH4		Γ			Γ			Γ	
K2/CH5		Γ			Γ			Γ	
K3/CH6		Γ			Γ			Γ	
R1/CH7			Γ			Γ			Γ
R1/CH8			Γ			Γ			Γ
R1/CH9			Γ			Γ			Γ
SHIELD	Γ	Γ	Γ	Γ	Γ	Γ	Γ	Γ	Γ

1. ↑ = start of bank Bx acquisition.
2. eoa = end of Bx acquisition.
3. Px = program groups and channels for Bank x.
4. I = interrupt.
5. Γ = charge transfer cycle (green for CHx and blue for SHIELD).

The table below provides a list of features for this example.

Table 25. Example G1/G2/G3/G4 (3 touchkey, 1 linear, 1 rotary sensors) - Synthesis

Feature	Value
Sampling capacitance	3+ 1 (shield)
Pins without shield	12
Pins with shield	14
Cost	Medium
Response time	3 * T0

3.19 Error management

Top level error management can be done at user application level.

The APIs are described in *tsl_user.c* where we split keys and linrot error cases.

The customer decides which action to be done in these cases (such as reboot, re-init or alarm).

The corresponding APIs are detailed in the below code:

```
/**
  * @brief  Executed when a sensor is in Error state
  * @param  None
  * @retval None
  */
  void MyTKeys_ErrorStateProcess(void)
{
  /* Add here your own processing when a sensor is in Error state */
}
void MyLinRots_ErrorStateProcess(void)
{
  /* Add here your own processing when a sensor is in Error state */
}

/**
  * @brief  Executed when a sensor is in Off state
  * @param  None
  * @retval None
  */
void MyTKeys_OffStateProcess(void)
{
  /* Add here your own processing when a sensor is in Off state */
}

void MyLinRots_OffStateProcess(void)
{
  /* Add here your own processing when a sensor is in Off state */
}
```

4 Devices with TSC peripheral

This section concerns all STM32 microcontrollers that include the touch sensing controller peripheral (TSC).

4.1 Acquisition

The acquisition is done in the files:

- *tsl_acq_tsc.c*
- *tsl_acq_tsc.h*

Functions used by the application layer and that are device dependent:

- `TSL_acq_BankConfig()`
- `TSL_acq_BankStartAcq()`
- `TSL_acq_BankWaitEOC()`
- `TSL_acq_GetMeas()`

The other functions in this file are for internal use and the user does not need to call them directly.

The device selection must be done at the end of the *tsl_conf.h* file:

```
#include "stm32f0xx.h" /* Select the file corresponding to the device in use
(i.e. stm32f3xx.h, stm32f0xx.h, ...) */
```

4.2 Timings

The timing management is done in the files:

- *tsl_time.c*
- *tsl_time.h*

The SysTick is used to generate a timebase for the ECS and DTO modules. It must be initialized in the user code (already done by the `HAL_init` function).

4.3 Parameters

The parameters are described in the *tsl_conf_tsc_template.h* file (to be copied in the project and rename in *tsl_conf.h*).

Parameters are checked in the *tsl_check_config_tsc.h* file.

4.4 MCU resources

The table below shows the peripherals used by the STMTouch touch sensing library on any STM32 microcontroller with the TSC peripheral. Care must be taken when using them to avoid any unwanted behavior.

Table 26. MCU resources used

Peripheral	Function
GPIOs	Acquisition
SysTick	Time base for ECS and DTO
Touch sensing controller (TSC)	Acquisition

4.5 Memory footprint

Conditions

- IAR ANSI C/C++ compiler/linker V7.40.3.8902 for Arm
- Compiler optimization: high size
- Counted files: *tsl*.o*
- STMTouch touch sensing library options: ECS = ON, DTO = ON, DXS = OFF, PROX = OFF
- Each sensor has its own parameters placed in RAM.

The following table summarizes the memory footprint with different configurations and for the various MCUs.

Table 27. STM32 memory footprint(1)

Channels	Banks	Sensors	STM32F0/L0		STM32F3/L4/L4+/L5 STM32U5/WB	
			ROM (Kbytes)	RAM (bytes)	ROM (Kbytes)	RAM (bytes)
1	1	1 TKey	3.0	100	2.8	100
2	1	2 TKeys	3.0	120	2.8	120
2	2	2 TKeys	3.0	120	2.8	120
24	3	24 TKeys	4.0	620	3.8	620
3	1	1 Linear-3ch	4.1	130	3.8	130
15	3	12 TKeys + 1 Linear-3ch	6.2	420	5.7	420
24	3	18 TKeys + 2 Linear-3ch	6.5	610	6.0	610

1. The content of this table is provided only for information.

4.6 STM32F0 series

4.6.1 Available touch sensing channels

The tables below provide an overview of the available touch sensing channels for the STM32F0 series microcontrollers.

The content of these tables is provided only for information.

Warning: For n available pins in an I/O group, one pin must be used as sampling capacitor and n-1 pins are used as channels.

Note: The I/O group cannot be used if the number of available pins in less or equal to one.

Table 28. STM32F0 20-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32F042Fx
			TSSOP20
G1	TSC_G1_IO1	PA0	X(2)
	TSC_G1_IO2	PA1	X
	TSC_G1_IO3	PA2	X
	TSC_G1_IO4	PA3	X
G2	TSC_G2_IO1	PA4	(3)
	TSC_G2_IO2	PA5	(3)
	TSC_G2_IO3	PA6	X
	TSC_G2_IO4	PA7	X
G4	TSC_G4_IO1	PA9	X
	TSC_G4_IO2	PA10	X
	TSC_G4_IO3	PA11	X
	TSC_G4_IO4	PA12	X
Maximum number of sensors			8

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

Table 29. STM32F0 28-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32F042Gx	STM32F048Gx
			UFQFPN28	
G1	TSC_G1_IO1	PA0	X(2)	X
	TSC_G1_IO2	PA1	X	X
	TSC_G1_IO3	PA2	X	X
	TSC_G1_IO4	PA3	X	X
G2	TSC_G2_IO1	PA4	(3)	(3)
	TSC_G2_IO2	PA5	(3)	(3)
	TSC_G2_IO3	PA6	X	X
	TSC_G2_IO4	PA7	X	X
G3	TSC_G3_IO2	PB0	X	-
	TSC_G3_IO3	PB1	X	-
G4	TSC_G4_IO1	PA9	X	X
	TSC_G4_IO2	PA10	X	X
	TSC_G4_IO3	PA11	X	X
	TSC_G4_IO4	PA12	X	X
G5	TSC_G5_IO1	PB3	X	X
	TSC_G5_IO2	PB4	X	X
	TSC_G5_IO3	PB6	X	X
	TSC_G5_IO4	PB7	X	X
Maximum number of sensors			12	11

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. ‘X’ I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

Table 30. STM32F0 32-pin packages - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32F042/051Kx	
			LQFP32	UFQFPN32
G1	TSC_G1_IO1	PA0	X(2)	X
	TSC_G1_IO2	PA1	X	X
	TSC_G1_IO3	PA2	X	X
	TSC_G1_IO4	PA3	X	X

Table 30. STM32F0 32-pin packages - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32F042/051Kx	
			LQFP32	UFQFPN32
G2	TSC_G2_IO1	PA4	(3)	(3)
	TSC_G2_IO2	PA5	(3)	(3)
	TSC_G2_IO3	PA6	X	X
	TSC_G2_IO4	PA7	X	X
G3	TSC_G3_IO2	PB0	X	X
	TSC_G3_IO3	PB1	X	X
	TSC_G3_IO4	PB2	-	X
G4	TSC_G4_IO1	PA9	X	X
	TSC_G4_IO2	PA10	X	X
	TSC_G4_IO3	PA11	X	X
	TSC_G4_IO4	PA12	X	X
G5	TSC_G5_IO1	PB3	X	X
	TSC_G5_IO2	PB4	X	X
	TSC_G5_IO3	PB6	X	X
	TSC_G5_IO4	PB7	X	X
Maximum number of sensors			12	13

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

Table 31. STM32F0 36-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32F042/051Tx	STM32F048/058Tx
			WLCSP36	
G1	TSC_G1_IO1	PA0	X(2)	X
	TSC_G1_IO2	PA1	X	X
	TSC_G1_IO3	PA2	X	X
	TSC_G1_IO4	PA3	X	X
G2	TSC_G2_IO1	PA4	(3)	(3)
	TSC_G2_IO2	PA5	(3)	(3)
	TSC_G2_IO3	PA6	X	X
	TSC_G2_IO4	PA7	X	X

Table 31. STM32F0 36-pin package - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32F042/051Tx	STM32F048/058Tx
			WLCSP36	
G3	TSC_G3_IO2	PB0	X	X
	TSC_G3_IO3	PB1	X	X
	TSC_G3_IO4	PB2	X	-
G4	TSC_G4_IO1	PA9	X	X
	TSC_G4_IO2	PA10	X	X
	TSC_G4_IO3	PA11	X	X
	TSC_G4_IO4	PA12	X	X
G5	TSC_G5_IO1	PB3	X	X
	TSC_G5_IO2	PB4	X	X
	TSC_G5_IO3	PB6	X	X
	TSC_G5_IO4	PB7	X	X
Maximum number of sensors			13	12

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

Table 32. STM32F0 48-pin packages - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32F042Cx	STM32F051/ 071/072/091Cx	STM32F058 /078/098Cx	STM32F048Cx
			LQFP48 - UFQFPN48			UFQFPN48
G1	TSC_G1_IO1	PA0	X(2)	X	X	X
	TSC_G1_IO2	PA1	X	X	X	X
	TSC_G1_IO3	PA2	X	X	X	X
	TSC_G1_IO4	PA3	X	X	X	X
G2	TSC_G2_IO1	PA4	(3)	(3)	(3)	(3)
	TSC_G2_IO2	PA5	(3)	(3)	(3)	(3)
	TSC_G2_IO3	PA6	X	X	X	X
	TSC_G2_IO4	PA7	X	X	X	X
G3	TSC_G3_IO2	PB0	X	X	X	X
	TSC_G3_IO3	PB1	X	X	X	X
	TSC_G3_IO4	PB2	X	-	X	-

Table 32. STM32F0 48-pin packages - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32F042Cx	STM32F051/ 071/072/091Cx	STM32F058 /078/098Cx	STM32F048Cx
			LQFP48 - UFQFPN48			UFQFPN48
G4	TSC_G4_IO1	PA9	X	X	X	X
	TSC_G4_IO2	PA10	X	X	X	X
	TSC_G4_IO3	PA11	X	X	X	X
	TSC_G4_IO4	PA12	X	X	X	X
G5	TSC_G5_IO1	PB3	X	X	X	X
	TSC_G5_IO2	PB4	X	X	X	X
	TSC_G5_IO3	PB6	X	X	X	X
	TSC_G5_IO4	PB7	X	X	X	X
G6	TSC_G6_IO1	PB11	-	X	X	-
	TSC_G6_IO2	PB12	-	X	X	-
	TSC_G6_IO3	PB13	-	X	X	-
	TSC_G6_IO4	PB14	-	X	X	-
Maximum number of sensors			13	16	15	12

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

Table 33. STM32F0 49-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32F071/072Cx	STM32F078Cx
			WLCSP49	
G1	TSC_G1_IO1	PA0	X(2)	X
	TSC_G1_IO2	PA1	X	X
	TSC_G1_IO3	PA2	X	X
	TSC_G1_IO4	PA3	X	X
G2	TSC_G2_IO1	PA4	(3)	(3)
	TSC_G2_IO2	PA5	(3)	(3)
	TSC_G2_IO3	PA6	X	X
	TSC_G2_IO4	PA7	X	X
G3	TSC_G3_IO2	PB0	X	X
	TSC_G3_IO3	PB1	X	X
	TSC_G3_IO4	PB2	X	-

Table 33. STM32F0 49-pin package - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32F071/072Cx	STM32F078Cx
			WLCSP49	
G4	TSC_G4_IO1	PA9	X	X
	TSC_G4_IO2	PA10	X	X
	TSC_G4_IO3	PA11	X	X
	TSC_G4_IO4	PA12	X	X
G5	TSC_G5_IO1	PB3	X	X
	TSC_G5_IO2	PB4	X	X
	TSC_G5_IO3	PB6	X	X
	TSC_G5_IO4	PB7	X	X
G6	TSC_G6_IO1	PB11	X	X
	TSC_G6_IO2	PB12	X	X
	TSC_G6_IO3	PB13	X	X
	TSC_G6_IO4	PB14	X	X
Maximum number of sensors			16	15

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

Table 34. STM32F0 64-pin packages - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32F051/071 /072/091Rx	STM32F058 /078/098Rx
			LQFP64 - UFBGA64 - WLCSP64	
G1	TSC_G1_IO1	PA0	X(2)	X
	TSC_G1_IO2	PA1	X	X
	TSC_G1_IO3	PA2	X	X
	TSC_G1_IO4	PA3	X	X
G2	TSC_G2_IO1	PA4	(3)	(3)
	TSC_G2_IO2	PA5	(3)	(3)
	TSC_G2_IO3	PA6	X	X
	TSC_G2_IO4	PA7	X	X
G3	TSC_G3_IO1	PC5	X	X
	TSC_G3_IO2	PB0	X	X
	TSC_G3_IO3	PB1	X	X
	TSC_G3_IO4	PB2	X	-

Table 34. STM32F0 64-pin packages - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32F051/071 /072/091Rx	STM32F058 /078/098Rx
			LQFP64 - UFBGA64 - WLCSP64	
G4	TSC_G4_IO1	PA9	X	X
	TSC_G4_IO2	PA10	X	X
	TSC_G4_IO3	PA11	X	X
	TSC_G4_IO4	PA12	X	X
G5	TSC_G5_IO1	PB3	X	X
	TSC_G5_IO2	PB4	X	X
	TSC_G5_IO3	PB6	X	X
	TSC_G5_IO4	PB7	X	X
G6	TSC_G6_IO1	PB11	X	X
	TSC_G6_IO2	PB12	X	X
	TSC_G6_IO3	PB13	X	X
	TSC_G6_IO4	PB14	X	X
Maximum number of sensors			17	16

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

Table 35. STM32F0 100-pin packages - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32F071 /072/091Vx	STM32F078/098Rx
			LQFP100 - UFBGA100	
G1	TSC_G1_IO1	PA0	X(2)	X
	TSC_G1_IO2	PA1	X	X
	TSC_G1_IO3	PA2	X	X
	TSC_G1_IO4	PA3	X	X
G2	TSC_G2_IO1	PA4	(3)	(3)
	TSC_G2_IO2	PA5	(3)	(3)
	TSC_G2_IO3	PA6	X	X
	TSC_G2_IO4	PA7	X	X

Table 35. STM32F0 100-pin packages - Touch sensing channels(1) (continued)

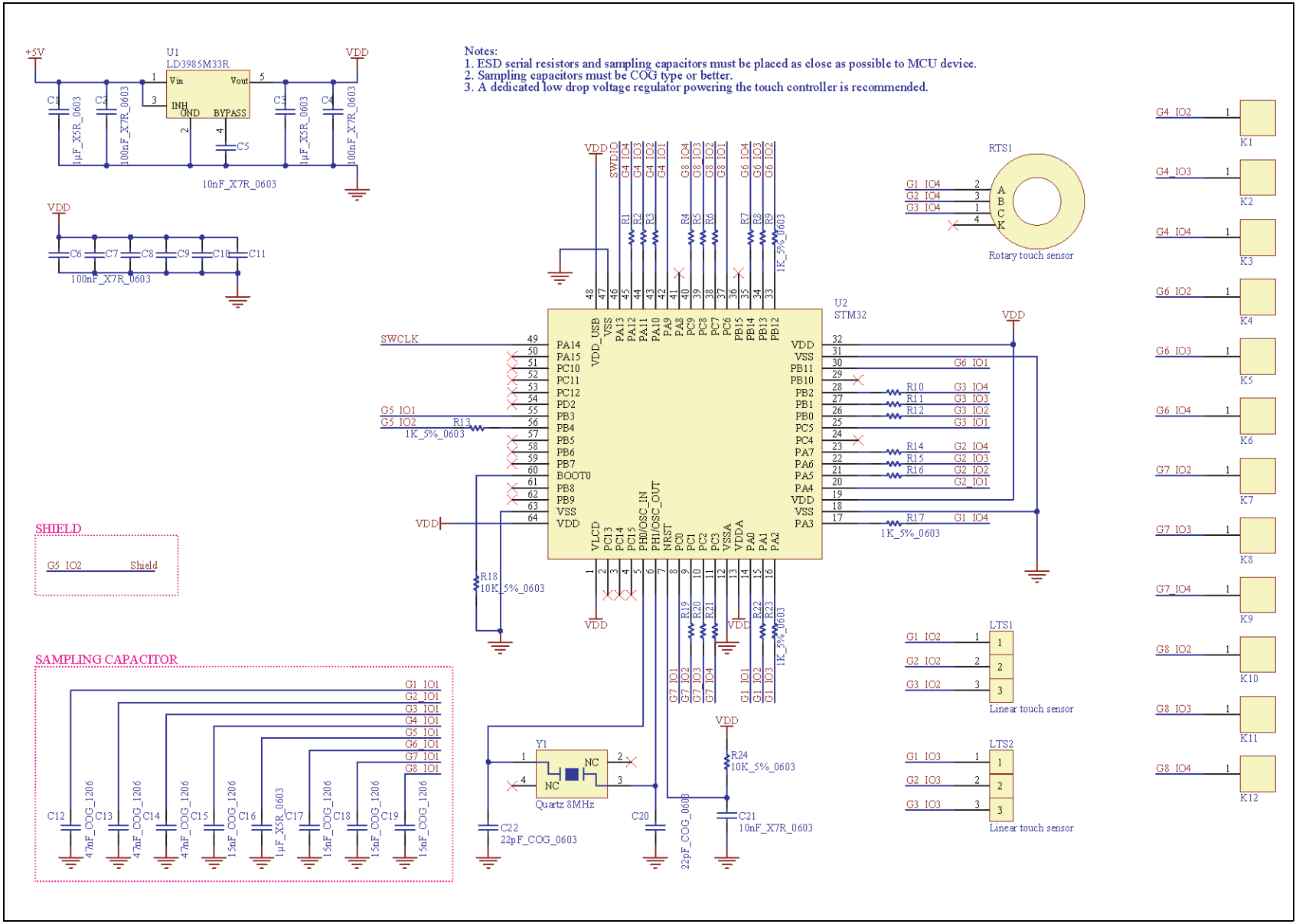
Group	Capacitive sensing signal name	Pin name	STM32F071 /072/091Vx	STM32F078/098Rx
			LQFP100 - UFBGA100	
G3	TSC_G3_IO1	PC5	X	X
	TSC_G3_IO2	PB0	X	X
	TSC_G3_IO3	PB1	X	X
	TSC_G3_IO4	PB2	X	-
G4	TSC_G4_IO1	PA9	X	X
	TSC_G4_IO2	PA10	X	X
	TSC_G4_IO3	PA11	X	X
	TSC_G4_IO4	PA12	X	X
G5	TSC_G5_IO1	PB3	X	X
	TSC_G5_IO2	PB4	X	X
	TSC_G5_IO3	PB6	X	X
	TSC_G5_IO4	PB7	X	X
G6	TSC_G6_IO1	PB11	X	X
	TSC_G6_IO2	PB12	X	X
	TSC_G6_IO3	PB13	X	X
	TSC_G6_IO4	PB14	X	X
G7	TSC_G7_IO1	PE2	X	X
	TSC_G7_IO2	PE3	X	X
	TSC_G7_IO3	PE4	X	X
	TSC_G7_IO4	PE5	X	X
G8	TSC_G8_IO1	PD12	X	X
	TSC_G8_IO2	PD13	X	X
	TSC_G8_IO3	PD14	X	X
	TSC_G8_IO4	PD15	X	X
Maximum number of sensors			23	22

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

4.6.2 Hardware implementation example

Figure 17 shows an example of hardware implementation on STM32F0 series microcontrollers.

Figure 17. STM32F0 series hardware implementation example



4.7 STM32F3 series microcontrollers

4.7.1 Available touch sensing channels

The tables below provide an overview of the available touch sensing channels for the STM32F3 series microcontrollers.

The content of these tables is provided only for information.

Warning: For n available pins in an I/O group, one pin must be used as sampling capacitor and n-1 pins are used as channels.

Note: The I/O group cannot be used if the number of available pins in less or equal to one.

Table 36. STM32F3 32-pin packages - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32F301/303 /334Kx	STM32F301/302 /303/334Kx	STM32F318Kx
			LQFP32	UFQFPN32	
G1	TSC_G1_IO1	PA0	X(2)	X	X
	TSC_G1_IO2	PA1	X	X	X
	TSC_G1_IO3	PA2	(3)	(3)	(3)
	TSC_G1_IO4	PA3	X	X	X
G2	TSC_G2_IO1	PA4	(3)	(3)	(3)
	TSC_G2_IO2	PA5	(3)	(3)	(3)
	TSC_G2_IO3	PA6	(3)	(3)	(3)
	TSC_G2_IO4	PA7	X	X	X
G3	TSC_G3_IO2	PB0	X	-	-
	TSC_G3_IO3	PB1	(3)	-	-
G4	TSC_G4_IO1	PA9	X	X	X
	TSC_G4_IO2	PA10	X	X	X
	TSC_G4_IO3	PA13	X	X	X
	TSC_G4_IO4	PA14	X	X	X
G5	TSC_G5_IO1	PB3	X	X	X
	TSC_G5_IO2	PB4	X	X	X
	TSC_G5_IO3	PB6	X	X	X
	TSC_G5_IO4	PB7	X	X	-
Maximum number of sensors			11	10	9

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. ‘X’ I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

Table 37. STM32F3 48-pin package 1/2 - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32F301/302 /303/334Cx	STM32F318/328 /358Cx
			LQFP48	
G1	TSC_G1_IO1	PA0	X(2)	X
	TSC_G1_IO2	PA1	X	X
	TSC_G1_IO3	PA2	(3)	(3)
	TSC_G1_IO4	PA3	X	X
G2	TSC_G2_IO1	PA4	(3)	(3)
	TSC_G2_IO2	PA5	(3)	(3)
	TSC_G2_IO3	PA6	(3)	(3)
	TSC_G2_IO4	PA7	X	X
G3	TSC_G3_IO2	PB0	X	X
	TSC_G3_IO3	PB1	(3)	(3)
	TSC_G3_IO4	PB2	X	-
G4	TSC_G4_IO1	PA9	X	X
	TSC_G4_IO2	PA10	X	X
	TSC_G4_IO3	PA13	X	X
	TSC_G4_IO4	PA14	X	X
G5	TSC_G5_IO1	PB3	X	X
	TSC_G5_IO2	PB4	X	X
	TSC_G5_IO3	PB6	X	X
	TSC_G5_IO4	PB7	X	X
G6	TSC_G6_IO1	PB11	X	X
	TSC_G6_IO2	PB12	(3)	(3)
	TSC_G6_IO3	PB13	X	X
	TSC_G6_IO4	PB14	X	X
Maximum number of sensors			15	14

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

Table 38. STM32F3 48-pin package 2/2 - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32F373/378Cx
			LQFP48
G1	TSC_G1_IO1	PA0	X(2)
	TSC_G1_IO2	PA1	X
	TSC_G1_IO3	PA2	X
	TSC_G1_IO4	PA3	X
G2	TSC_G2_IO1	PA4	(3)
	TSC_G2_IO2	PA5	(3)
	TSC_G2_IO3	PA6	X
	TSC_G2_IO4	PA7	-
G3	TSC_G3_IO3	PB0	X
	TSC_G3_IO4	PB1	(3)
G4	TSC_G4_IO1	PA9	X
	TSC_G4_IO2	PA10	X
	TSC_G4_IO3	PA13	X
	TSC_G4_IO4	PA14	X
G5	TSC_G5_IO1	PB3	X
	TSC_G5_IO2	PB4	X
	TSC_G5_IO3	PB6	X
	TSC_G5_IO4	PB7	X
G6	TSC_G6_IO1	PB14	X
	TSC_G6_IO2	PB15	X
	TSC_G6_IO3	PD8	X
Maximum number of sensors			12

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. ‘X’ I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

Table 39. STM32F3 49-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32F301 /302Cx	STM32F303 334Cx	STM32F318Cx
			WLCSP49		
G1	TSC_G1_IO1	PA0	X(2)	X	X
	TSC_G1_IO2	PA1	X	X	X
	TSC_G1_IO3	PA2	(3)	(3)	(3)
	TSC_G1_IO4	PA3	X	X	X
G2	TSC_G2_IO1	PA4	(3)	(3)	(3)
	TSC_G2_IO2	PA5	(3)	(3)	(3)
	TSC_G2_IO3	PA6	(3)	(3)	(3)
	TSC_G2_IO4	PA7	X	X	X
G3	TSC_G3_IO1	PC5	-	X	-
	TSC_G3_IO2	PB0	X	X	X
	TSC_G3_IO3	PB1	(3)	(3)	(3)
	TSC_G3_IO4	PB2	X	X	-
G4	TSC_G4_IO1	PA9	X	X	X
	TSC_G4_IO2	PA10	X	X	X
	TSC_G4_IO3	PA13	X	X	X
	TSC_G4_IO4	PA14	X	X	X
G5	TSC_G5_IO1	PB3	X	X	X
	TSC_G5_IO2	PB4	X	X	X
	TSC_G5_IO3	PB6	X	X	X
	TSC_G5_IO4	PB7	X	X	X
G6	TSC_G6_IO1	PB11	X	X	X
	TSC_G6_IO2	PB12	(3)	(3)	(3)
	TSC_G6_IO3	PB13	X	X	X
	TSC_G6_IO4	PB14	X	X	X
Maximum number of sensors			15	16	14

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. ‘X’ I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

Table 40. STM32F3 64-pin package 1/2 - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32F301/302 /303/334Rx	STM32F358Rx
			LQFP64	
G1	TSC_G1_IO1	PA0	X(2)	X
	TSC_G1_IO2	PA1	X	X
	TSC_G1_IO3	PA2	(3)	(3)
	TSC_G1_IO4	PA3	X	X
G2	TSC_G2_IO1	PA4	(3)	(3)
	TSC_G2_IO2	PA5	(3)	(3)
	TSC_G2_IO3	PA6	(3)	(3)
	TSC_G2_IO4	PA7	X	X
G3	TSC_G3_IO1	PC5	X	X
	TSC_G3_IO2	PB0	X	X
	TSC_G3_IO3	PB1	(3)	(3)
	TSC_G3_IO4	PB2	X	-
G4	TSC_G4_IO1	PA9	X	X
	TSC_G4_IO2	PA10	X	X
	TSC_G4_IO3	PA13	X	X
	TSC_G4_IO4	PA14	X	X
G5	TSC_G5_IO1	PB3	X	X
	TSC_G5_IO2	PB4	X	X
	TSC_G5_IO3	PB6	X	X
	TSC_G5_IO4	PB7	X	X
G6	TSC_G6_IO1	PB11	X	X
	TSC_G6_IO2	PB12	(3)	(3)
	TSC_G6_IO3	PB13	X	X
	TSC_G6_IO4	PB14	X	X
Maximum number of sensors			16	15

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. ‘X’ I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

Table 41. STM32F3 64-pin package 2/2 - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32F373/378Rx
			LQFP64
G1	TSC_G1_IO1	PA0	X(2)
	TSC_G1_IO2	PA1	X
	TSC_G1_IO3	PA2	(3)
	TSC_G1_IO4	PA3	X
G2	TSC_G2_IO1	PA4	(3)
	TSC_G2_IO2	PA5	(3)
	TSC_G2_IO3	PA6	(3)
	TSC_G2_IO4	PA7	X
G3	TSC_G3_IO1	PC4	X
	TSC_G3_IO2	PC5	X
	TSC_G3_IO3	PB0	X
	TSC_G3_IO4	PB1	(3)
G4	TSC_G4_IO1	PA9	X
	TSC_G4_IO2	PA10	X
	TSC_G4_IO3	PA13	X
	TSC_G4_IO4	PA14	X
G5	TSC_G5_IO1	PB3	X
	TSC_G5_IO2	PB4	X
	TSC_G5_IO3	PB6	X
	TSC_G5_IO4	PB7	X
G6	TSC_G6_IO1	PB14	X
	TSC_G6_IO2	PB15	X
	TSC_G6_IO3	PD8	X
Maximum number of sensors			15

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

Table 42. STM32F3 66-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32F378Rx
			WLCSP66
G1	TSC_G1_IO1	PA0	X(2)
	TSC_G1_IO2	PA1	X
	TSC_G1_IO3	PA2	(3)
	TSC_G1_IO4	PA3	X
G2	TSC_G2_IO1	PA4	(3)
	TSC_G2_IO2	PA5	(3)
	TSC_G2_IO3	PA6	(3)
	TSC_G2_IO4	PA7	X
G3	TSC_G3_IO1	PC4	X
	TSC_G3_IO2	PC5	X
	TSC_G3_IO3	PB0	X
	TSC_G3_IO4	PB1	(3)
G4	TSC_G4_IO1	PA9	X
	TSC_G4_IO2	PA10	X
	TSC_G4_IO3	PA13	X
	TSC_G4_IO4	PA14	X
G5	TSC_G5_IO1	PB3	X
	TSC_G5_IO2	PB4	X
	TSC_G5_IO3	PB6	X
	TSC_G5_IO4	PB7	X
G6	TSC_G6_IO1	PB14	X
	TSC_G6_IO2	PB15	X
	TSC_G6_IO3	PD8	X
Maximum number of sensors			15

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. ‘X’ I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

Table 43. STM32F3 100-pin packages 1/2 - Touch sensing channels[1]

Group	Capacitive sensing signal name	Pin name	STM32F302 /303Vx	STM32F358 /398Vx
			LQFP100 - UFBGA100 - WLCSP100	LQFP100
G1	TSC_G1_IO1	PA0	X[2]	X
	TSC_G1_IO2	PA1	X	X
	TSC_G1_IO3	PA2	(3)	(3)
	TSC_G1_IO4	PA3	X	X
G2	TSC_G2_IO1	PA4	(3)	(3)
	TSC_G2_IO2	PA5	(3)	(3)
	TSC_G2_IO3	PA6	(3)	(3)
	TSC_G2_IO4	PA7	X	X
G3	TSC_G3_IO1	PC5	X	X
	TSC_G3_IO2	PB0	X	X
	TSC_G3_IO3	PB1	(3)	(3)
	TSC_G3_IO4	PB2	X	-
G4	TSC_G4_IO1	PA9	X	X
	TSC_G4_IO2	PA10	X	X
	TSC_G4_IO3	PA13	X	X
	TSC_G4_IO4	PA14	X	X
G5	TSC_G5_IO1	PB3	X	X
	TSC_G5_IO2	PB4	X	X
	TSC_G5_IO3	PB6	X	X
	TSC_G5_IO4	PB7	X	X
G6	TSC_G6_IO1	PB11	X	X
	TSC_G6_IO2	PB12	(3)	(3)
	TSC_G6_IO3	PB13	X	X
	TSC_G6_IO4	PB14	X	X
G7	TSC_G7_IO1	PE2	X	X
	TSC_G7_IO2	PE3	X	X
	TSC_G7_IO3	PE4	X	X
	TSC_G7_IO4	PE5	X	X

Table 43. STM32F3 100-pin packages 1/2 - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32F302 /303Vx	STM32F358 /398Vx
			LQFP100 - UFBGA100 - WLCSP100	LQFP100
G8	TSC_G8_IO1	PD12	X	X
	TSC_G8_IO2	PD13	X	X
	TSC_G8_IO3	PD14	X	X
	TSC_G8_IO4	PD15	X	X
Maximum number of sensors			22	21

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

Table 44. STM32F3 100-pin packages 2/2 - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32F373 /378Vx
			LQFP100 - UFBGA100
G1	TSC_G1_IO1	PA0	X(2)
	TSC_G1_IO2	PA1	X
	TSC_G1_IO3	PA2	(3)
	TSC_G1_IO4	PA3	X
G2	TSC_G2_IO1	PA4	(3)
	TSC_G2_IO2	PA5	(3)
	TSC_G2_IO3	PA6	(3)
	TSC_G2_IO4	PA7	X
G3	TSC_G3_IO1	PC4	X
	TSC_G3_IO2	PC5	X
	TSC_G3_IO3	PB0	X
	TSC_G3_IO4	PB1	(3)
G4	TSC_G4_IO1	PA9	X
	TSC_G4_IO2	PA10	X
	TSC_G4_IO3	PA13	X
	TSC_G4_IO4	PA14	X

Table 44. STM32F3 100-pin packages 2/2 - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32F373 /378Vx
			LQFP100 - UFBGA100
G5	TSC_G5_IO1	PB3	X
	TSC_G5_IO2	PB4	X
	TSC_G5_IO3	PB6	X
	TSC_G5_IO4	PB7	X
G6	TSC_G6_IO1	PB14	X
	TSC_G6_IO2	PB15	X
	TSC_G6_IO3	PD8	X
	TSC_G6_IO4	PD9	X
G7	TSC_G7_IO1	PE2	X
	TSC_G7_IO2	PE3	X
	TSC_G7_IO3	PE4	X
	TSC_G7_IO4	PE5	X
G8	TSC_G8_IO1	PD12	X
	TSC_G8_IO2	PD13	X
	TSC_G8_IO3	PD14	X
	TSC_G8_IO4	PD15	X
Maximum number of sensors			22

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

Table 45. STM32F3 14-pin packages - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32F302/303Zx
			LQFP144
G1	TSC_G1_IO1	PA0	X(2)
	TSC_G1_IO2	PA1	X
	TSC_G1_IO3	PA2	(3)
	TSC_G1_IO4	PA3	X
G2	TSC_G2_IO1	PA4	(3)
	TSC_G2_IO2	PA5	(3)
	TSC_G2_IO3	PA6	(3)
	TSC_G2_IO4	PA7	X
G3	TSC_G3_IO1	PC5	X
	TSC_G3_IO2	PB0	X
	TSC_G3_IO3	PB1	(3)
	TSC_G3_IO4	PB2	X
G4	TSC_G4_IO1	PA9	X
	TSC_G4_IO2	PA10	X
	TSC_G4_IO3	PA13	X
	TSC_G4_IO4	PA14	X
G5	TSC_G5_IO1	PB3	X
	TSC_G5_IO2	PB4	X
	TSC_G5_IO3	PB6	X
	TSC_G5_IO4	PB7	X
G6	TSC_G6_IO1	PB11	X
	TSC_G6_IO2	PB12	(3)
	TSC_G6_IO3	PB13	X
	TSC_G6_IO4	PB14	X
G7	TSC_G7_IO1	PE2	X
	TSC_G7_IO2	PE3	X
	TSC_G7_IO3	PE4	X
	TSC_G7_IO4	PE5	X
G8	TSC_G8_IO1	PD12	X
	TSC_G8_IO2	PD13	X
	TSC_G8_IO3	PD14	X
	TSC_G8_IO4	PD15	X
Maximum number of sensors			22

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. ‘X’ I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

4.7.2 Hardware implementation example

Figure 18 shows an example of hardware implementation on STM32F3 series microcontrollers.

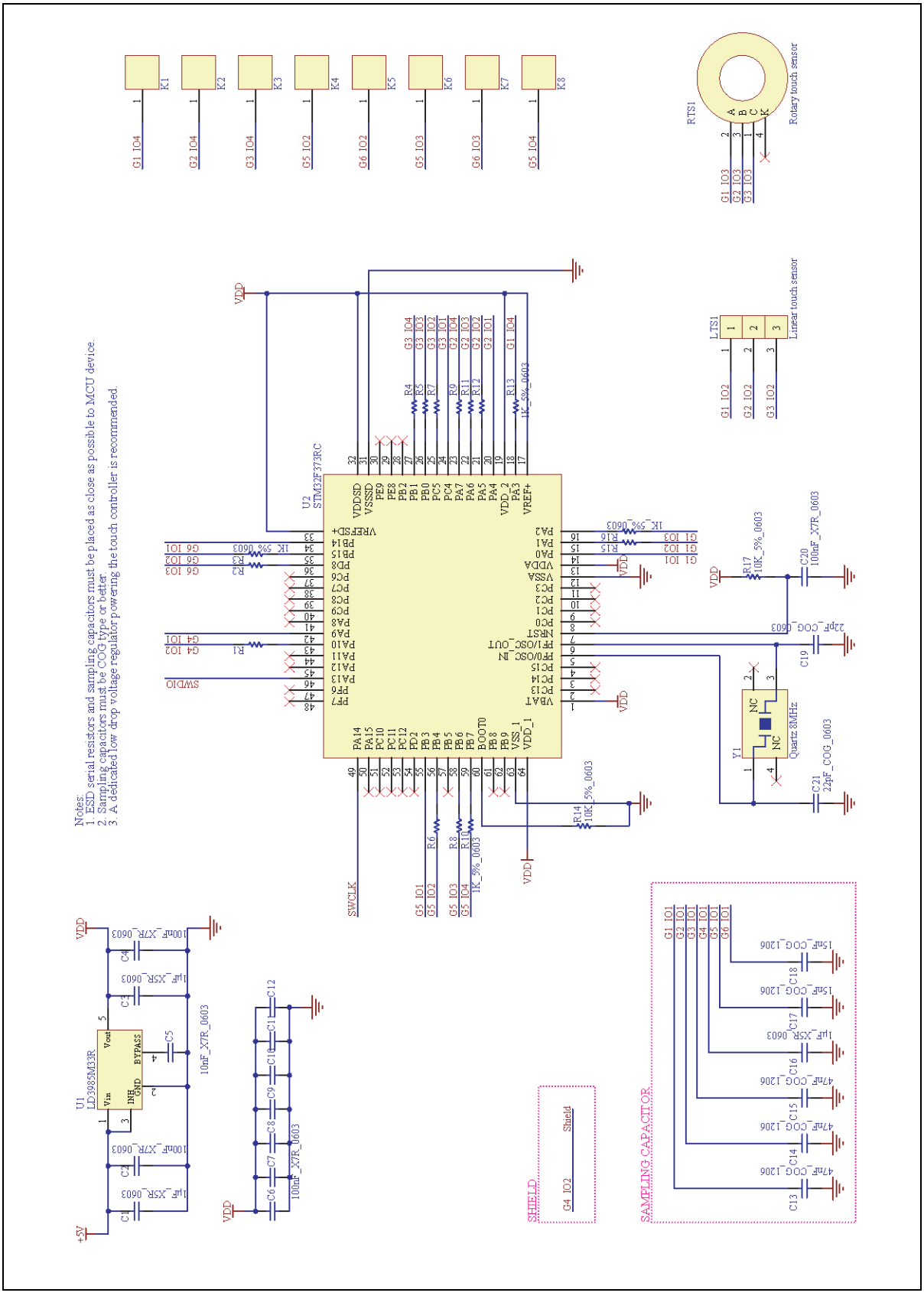


Figure 18. STM32F3 series hardware implementation example

4.8 STM32L0 series microcontrollers

4.8.1 Available touch sensing channels

The tables below provide an overview of the available touch sensing channels for the STM32L0 series microcontrollers.

The content of these tables is provided only for information.

Warning: For n available pins in an I/O group, one pin must be used as sampling capacitor and n-1 pins are used as channels.

Note: The I/O group cannot be used if the number of available pins in less or equal to one.

Table 46. STM32L0 32-pin packages - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L052 /062Kx	STM32L072 /082Kx	STM32L052 /062Kx	STM32L072 /082Kx
			LQFP32		UFQFPN32	
G1	TSC_G1_IO1	PA0	X(2)	X	X	X
	TSC_G1_IO2	PA1	X	X	X	X
	TSC_G1_IO3	PA2	X	X	X	X
	TSC_G1_IO4	PA3	X	X	X	X
G2	TSC_G2_IO1	PA4	(3)	X	(3)	X
	TSC_G2_IO2	PA5	X	X	X	X
	TSC_G2_IO3	PA6	X	X	X	X
	TSC_G2_IO4	PA7	X	X	X	X
G3	TSC_G3_IO2	PB0	X	X	X	X
	TSC_G3_IO3	PB1	X	X	X	X
	TSC_G3_IO4	PB2	-	-	X	-
G4	TSC_G4_IO1	PA9	X	X	X	X
	TSC_G4_IO2	PA10	X	X	X	X
	TSC_G4_IO3	PA11	X	X	X	X
	TSC_G4_IO4	PA12	X	X	X	X
G5	TSC_G5_IO1	PB3	X	X	X	-
	TSC_G5_IO2	PB4	X	X	X	X
	TSC_G5_IO3	PB6	X	X	X	X
	TSC_G5_IO4	PB7	X	X	X	X
Maximum number of sensors			13	13	14	12

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.

3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

Table 47. STM32L0 36-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L052Tx
			WLCSP36
G1	TSC_G1_IO1	PA0	X(2)
	TSC_G1_IO2	PA1	X
	TSC_G1_IO3	PA2	X
	TSC_G1_IO4	PA3	X
G2	TSC_G2_IO1	PA4	(3)
	TSC_G2_IO2	PA5	X
	TSC_G2_IO3	PA6	X
	TSC_G2_IO4	PA7	X
G3	TSC_G3_IO2	PB0	X
	TSC_G3_IO3	PB1	X
	TSC_G3_IO4	PB2	X
G4	TSC_G4_IO1	PA9	X
	TSC_G4_IO2	PA10	X
	TSC_G4_IO3	PA11	X
	TSC_G4_IO4	PA12	X
G5	TSC_G5_IO1	PB3	X
	TSC_G5_IO2	PB4	X
	TSC_G5_IO3	PB6	X
	TSC_G5_IO4	PB7	X
Maximum number of sensors			14

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. ‘X’ I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

Table 48. STM32L0 48-pin packages - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L052/053 /062/063Cx	STM32L072/073 /082/083Cx
			LQFP48 - UFQFPN48	
G1	TSC_G1_IO1	PA0	X(2)	X
	TSC_G1_IO2	PA1	X	X
	TSC_G1_IO3	PA2	X	X
	TSC_G1_IO4	PA3	X	X
G2	TSC_G2_IO1	PA4	(3)	X
	TSC_G2_IO2	PA5	X	X
	TSC_G2_IO3	PA6	X	X
	TSC_G2_IO4	PA7	X	X
G3	TSC_G3_IO2	PB0	X	X
	TSC_G3_IO3	PB1	X	X
	TSC_G3_IO4	PB2	X	X
G4	TSC_G4_IO1	PA9	X	X
	TSC_G4_IO2	PA10	X	X
	TSC_G4_IO3	PA11	X	X
	TSC_G4_IO4	PA12	X	X
G5	TSC_G5_IO1	PB3	X	X
	TSC_G5_IO2	PB4	X	X
	TSC_G5_IO3	PB6	X	X
	TSC_G5_IO4	PB7	X	X
G6	TSC_G6_IO1	PB11	X	X
	TSC_G6_IO2	PB12	X	X
	TSC_G6_IO3	PB13	X	X
	TSC_G6_IO4	PB14	X	X
Maximum number of sensors			17	17

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. ‘X’ I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

Table 49. STM32L0 49-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L072/073/082Cx
			WLCSP49 - EWLCSP49
G1	TSC_G1_IO1	PA0	X(2)
	TSC_G1_IO2	PA1	X
	TSC_G1_IO3	PA2	X
	TSC_G1_IO4	PA3	X
G2	TSC_G2_IO1	PA4	X
	TSC_G2_IO2	PA5	X
	TSC_G2_IO3	PA6	X
	TSC_G2_IO4	PA7	X
G3	TSC_G3_IO2	PB0	X
	TSC_G3_IO3	PB1	X
	TSC_G3_IO4	PB2	X
G4	TSC_G4_IO1	PA9	X
	TSC_G4_IO2	PA10	X
	TSC_G4_IO3	PA11	X
	TSC_G4_IO4	PA12	X
G5	TSC_G5_IO1	PB3	X
	TSC_G5_IO2	PB4	X
	TSC_G5_IO3	PB6	X
	TSC_G5_IO4	PB7	X
G6	TSC_G6_IO1	PB11	X
	TSC_G6_IO2	PB12	X
	TSC_G6_IO3	PB13	X
	TSC_G6_IO4	PB14	X
G7	TSC_G7_IO1	PC0	X
	TSC_G7_IO2	PC1	X
	TSC_G7_IO3	PC2	X
Maximum number of sensors			19

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. ‘X’ I/Os can be used as sampling capacitor, sensor, or active shield.

Table 50. STM32L0 64-pin packages - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L052/053 /063Rx	STM32L072/073 /083Rx	STM32L052 /053Rx	STM32L072/073 /083Rx
			LQFP64		TFBGA64	TFBGA64 UFBGA64
G1	TSC_G1_IO1	PA0	X(2)	X	X	X
	TSC_G1_IO2	PA1	X	X	X	X
	TSC_G1_IO3	PA2	X	X	X	X
	TSC_G1_IO4	PA3	X	X	X	X
G2	TSC_G2_IO1	PA4	(3)	X	(3)	(3)
	TSC_G2_IO2	PA5	X	X	X	X
	TSC_G2_IO3	PA6	X	X	X	X
	TSC_G2_IO4	PA7	X	X	X	X
G3	TSC_G3_IO1	PC5	X	X	X	X
	TSC_G3_IO2	PB0	X	X	X	X
	TSC_G3_IO3	PB1	X	X	X	X
	TSC_G3_IO4	PB2	X	X	X	X
G4	TSC_G4_IO1	PA9	X	X	X	X
	TSC_G4_IO2	PA10	X	X	X	X
	TSC_G4_IO3	PA11	X	X	X	X
	TSC_G4_IO4	PA12	X	X	X	X
G5	TSC_G5_IO1	PB3	X	X	X	X
	TSC_G5_IO2	PB4	X	X	X	X
	TSC_G5_IO3	PB6	X	X	X	X
	TSC_G5_IO4	PB7	X	X	X	X
G6	TSC_G6_IO1	PB11	X	X	X	X
	TSC_G6_IO2	PB12	X	X	X	X
	TSC_G6_IO3	PB13	X	X	X	X
	TSC_G6_IO4	PB14	X	X	X	X
G7	TSC_G7_IO1	PC0	X	X	X	X
	TSC_G7_IO2	PC1	X	X	X	X
	TSC_G7_IO3	PC2	X	X	X	X
	TSC_G7_IO4	PC3	X	X	-	-

Table 50. STM32L0 64-pin packages - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32L052/053 /063Rx	STM32L072/073 /083Rx	STM32L052 /053Rx	STM32L072/073 /083Rx
			LQFP64		TFBGA64	TFBGA64 UFBGA64
G8	TSC_G8_IO1	PC6	X	X	X	X
	TSC_G8_IO2	PC7	X	X	X	X
	TSC_G8_IO3	PC8	X	X	X	X
	TSC_G8_IO4	PC9	X	X	X	X
Maximum number of sensors			24	24	23	23

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

Table 51. STM32L0 100-pin packages - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L072/073/083Vx
			UFBGA100 - LQFP100
G1	TSC_G1_IO1	PA0	X(2)
	TSC_G1_IO2	PA1	X
	TSC_G1_IO3	PA2	X
	TSC_G1_IO4	PA3	X
G2	TSC_G2_IO1	PA4	X
	TSC_G2_IO2	PA5	X
	TSC_G2_IO3	PA6	X
	TSC_G2_IO4	PA7	X
G3	TSC_G3_IO1	PC5	X
	TSC_G3_IO2	PB0	X
	TSC_G3_IO3	PB1	X
	TSC_G3_IO4	PB2	X
G4	TSC_G4_IO1	PA9	X
	TSC_G4_IO2	PA10	X
	TSC_G4_IO3	PA11	X
	TSC_G4_IO4	PA12	X
G5	TSC_G5_IO1	PB3	X
	TSC_G5_IO2	PB4	X
	TSC_G5_IO3	PB6	X
	TSC_G5_IO4	PB7	X

Table 51. STM32L0 100-pin packages - Touch sensing channels(1) (continued)

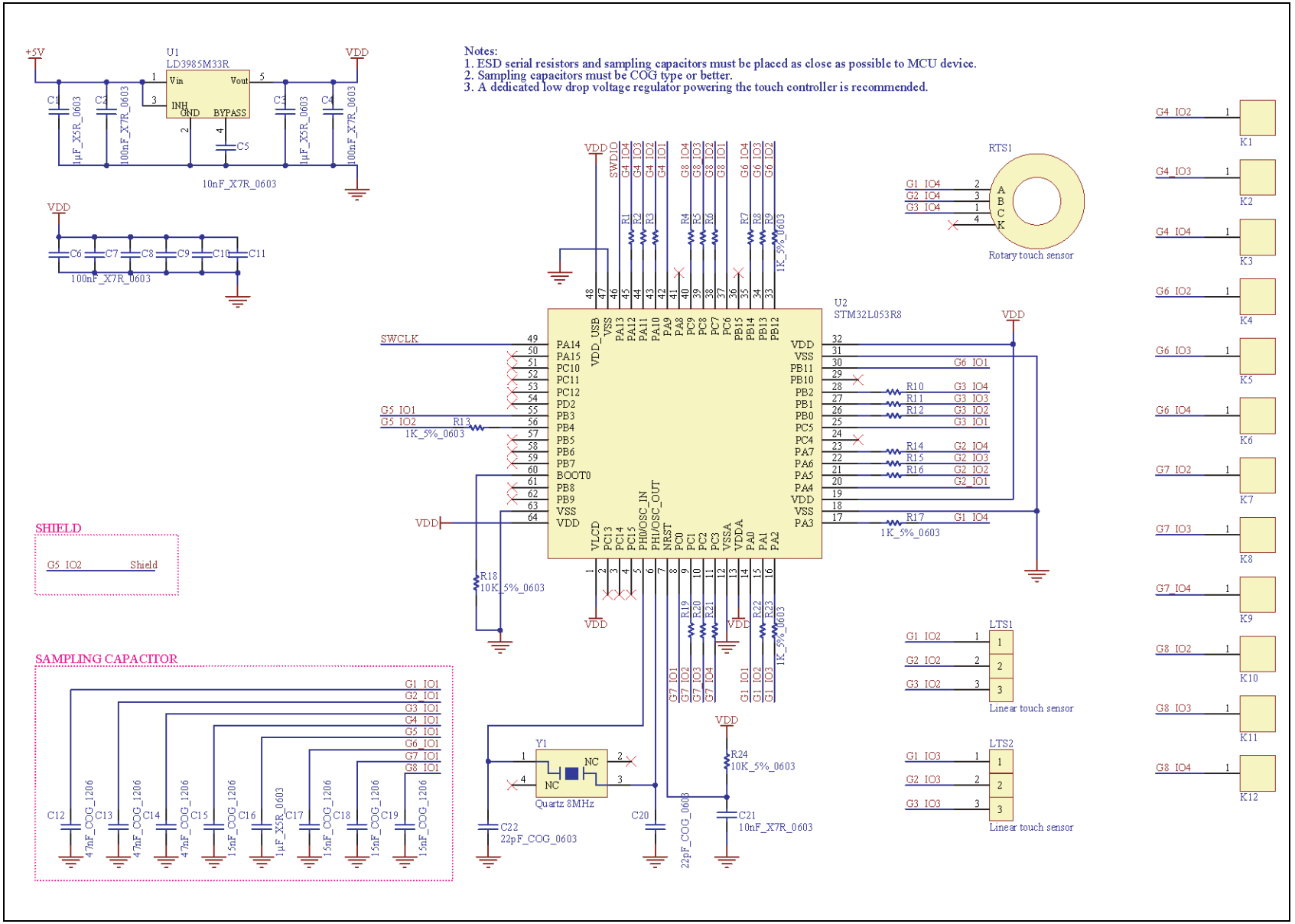
Group	Capacitive sensing signal name	Pin name	STM32L072/073/083Vx
			UFBGA100 - LQFP100
G6	TSC_G6_IO1	PB11	X
	TSC_G6_IO2	PB12	X
	TSC_G6_IO3	PB13	X
	TSC_G6_IO4	PB14	X
G7	TSC_G7_IO1	PC0	X
	TSC_G7_IO2	PC1	X
	TSC_G7_IO3	PC2	X
	TSC_G7_IO4	PC3	X
G8	TSC_G8_IO1	PC6	X
	TSC_G8_IO2	PC7	X
	TSC_G8_IO3	PC8	X
	TSC_G8_IO4	PC9	X
Maximum number of sensors			24

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.

4.8.2 Hardware implementation example

Figure 19 shows an example of hardware implementation on STM32L0 series microcontrollers.

Figure 19. STM32L0 series hardware implementation example



4.9 STM32L4 series microcontrollers

4.9.1 Available touch sensing channels

The tables below provide an overview of the available touch sensing channels for the STM32L4 series microcontrollers.

The content of these tables is provided only for information.

Warning: For n available pins in an I/O group, one pin must be used as sampling capacitor and n-1 pins are used as channels.

Note: The I/O group cannot be used if the number of available pins in less or equal to one.

Table 52. STM32L4 32-pin packages - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L412/422Kx	STM32L431/432/442Kx
			LQFP32 - UFQFPN32	UFQFPN32
G2	TSC_G2_IO1	PB4	(2)	(3)
	TSC_G2_IO2	PB5	(3)	(3)
	TSC_G2_IO3	PB6	X(4)	X
	TSC_G2_IO4	PB7	X	(3)
Maximum number of sensors			2	1

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.
3. All I/Os with this note offer a reduced touch sensing SNR. It is recommended to use these I/Os as active shield or sampling capacitor (see the product errata sheet for information).
4. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.

Table 53. STM32L4 36-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L412/422Tx
			WLCSP36
G2	TSC_G2_IO1	PB4	(2)
	TSC_G2_IO2	PB5	(3)
	TSC_G2_IO3	PB6	X(4)
	TSC_G2_IO4	PB7	X
Maximum number of sensors			1

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

3. All I/Os with this note offer a reduced touch sensing SNR. It is recommended to use these I/Os as active shield or sampling capacitor (see the product errata sheet for information).
4. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.

Table 54. STM32L4 48-pin packages - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L412/ 422Cx	STM32L431/ 433/443Cx	STM32L451/ 452/462Cx
			LQFP48 - UFQFPN48		
G1	TSC_G1_IO1	PB12	X(2)	X	(4)
	TSC_G1_IO2	PB13	X	X	X
	TSC_G1_IO3	PB14	X	X	X
	TSC_G1_IO4	PB15	X	X	X
G2	TSC_G2_IO1	PB4	(3)	(4)	(4)
	TSC_G2_IO2	PB5	(4)	(4)	(4)
	TSC_G2_IO3	PB6	X	X	X
	TSC_G2_IO4	PB7	X	(4)	(4)
Maximum number of sensors			5	4	4

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.
4. All I/Os with this note offer a reduced touch sensing SNR. It is recommended to use these I/Os as active shield or sampling capacitor (see the product errata sheet for information).

Table 55. STM32L4 49-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L431/433Cx
			WLCSP49
G1	TSC_G1_IO1	PB12	X(2)
	TSC_G1_IO2	PB13	X
	TSC_G1_IO3	PB14	X
	TSC_G1_IO4	PB15	X
G2	TSC_G2_IO1	PB4	(3)
	TSC_G2_IO2	PB5	(3)
	TSC_G2_IO3	PB6	X
	TSC_G2_IO4	PB7	(3)
Maximum number of sensors			4

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.

3. All I/Os with this note offer a reduced touch sensing SNR. It is recommended to use these I/Os as active shield or sampling capacitor (see the product errata sheet for information).

Table 56. STM32L4 64-pin packages - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L412/ 422Rx	STM32L431/433 /471/475/476/486Rx	STM32L451/ 452/462Rx
			LQFP64 - UFBGA64	LQFP64 - UFBGA64 - WLCSP64	
G1	TSC_G1_IO1	PB12	X(2)	X	(4)
	TSC_G1_IO2	PB13	X	X	X
	TSC_G1_IO3	PB14	X	X	X
	TSC_G1_IO4	PB15	X	X	X
G2	TSC_G2_IO1	PB4	(3)	(4)	(4)
	TSC_G2_IO2	PB5	(4)	(4)	(4)
	TSC_G2_IO3	PB6	X	X	X
	TSC_G2_IO4	PB7	X	(4)	(4)
G3	TSC_G3_IO1	PA15	(3)	(3)	(3)
	TSC_G3_IO2	PC10	X	X	X
	TSC_G3_IO3	PC11	X	X	X
	TSC_G3_IO4	PC12	X	X	X
G4	TSC_G4_IO1	PC6	X	X	X
	TSC_G4_IO2	PC7	X	X	X
	TSC_G4_IO3	PC8	X	X	X
	TSC_G4_IO4	PC9	X	X	X
Maximum number of sensors			11	10	10

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.
4. All I/Os with this note offer a reduced touch sensing SNR. It is recommended to use these I/Os as active shield or sampling capacitor (see the product errata sheet for information).

Table 57. STM32L4 72-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L473/485/486Gx
			WLCSP72
G1	TSC_G1_IO1	PB12	X(2)
	TSC_G1_IO2	PB13	X
	TSC_G1_IO3	PB14	X
	TSC_G1_IO4	PB15	X

Table 57. STM32L4 72-pin package - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32L473/485/486Gx
			WLCSP72
G2	TSC_G2_IO1	PB4	(3)
	TSC_G2_IO2	PB5	(3)
	TSC_G2_IO3	PB6	X
	TSC_G2_IO4	PB7	(3)
G3	TSC_G3_IO1	PA15	(4)
	TSC_G3_IO2	PC10	X
	TSC_G3_IO3	PC11	X
	TSC_G3_IO4	PC12	X
G4	TSC_G4_IO1	PC6	X
	TSC_G4_IO2	PC7	X
	TSC_G4_IO3	PC8	X
	TSC_G4_IO4	PC9	X
Maximum number of sensors			10

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing SNR. It is recommended to use these I/Os as active shield or sampling capacitor (see the product errata sheet for information).
4. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

Table 58. STM32L4 81-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L476Mx
			WLCSP81
G1	TSC_G1_IO1	PB12	X(2)
	TSC_G1_IO2	PB13	X
	TSC_G1_IO3	PB14	X
	TSC_G1_IO4	PB15	X
G2	TSC_G2_IO1	PB4	(3)
	TSC_G2_IO2	PB5	(3)
	TSC_G2_IO3	PB6	X
	TSC_G2_IO4	PB7	(3)

Table 58. STM32L4 81-pin package - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32L476Mx
			WLCSP81
G3	TSC_G3_IO1	PA15	(4)
	TSC_G3_IO2	PC10	X
	TSC_G3_IO3	PC11	X
	TSC_G3_IO4	PC12	X
G4	TSC_G4_IO1	PC6	X
	TSC_G4_IO2	PC7	X
	TSC_G4_IO3	PC8	X
	TSC_G4_IO4	PC9	X
Maximum number of sensors			10

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing SNR. It is recommended to use these I/Os as active shield or sampling capacitor (see the product errata sheet for information).
4. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

Table 59. STM32L4 100-pin packages - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L431/433/443/471/475/476/486/496/4A6Vx	STM32L451/452/462Vx	STM32L496/4A6Vx	
			LQFP100 - UFBGA100		LQFP100	WLCSP100
G1	TSC_G1_IO1	PB12	X(2)	(3)	X	X
	TSC_G1_IO2	PB13	X	X	X	X
	TSC_G1_IO3	PB14	X	X	X	X
	TSC_G1_IO4	PB15	X	X	X	X
G2	TSC_G2_IO1	PB4	(3)	(3)	(3)	(3)
	TSC_G2_IO2	PB5	(3)	(3)	(3)	(3)
	TSC_G2_IO3	PB6	X	X	X	X
	TSC_G2_IO4	PB7	(3)	(4)	(3)	(3)
G3	TSC_G3_IO1	PA15	(4)	X	(3)	(4)
	TSC_G3_IO2	PC10	X	X	X	X
	TSC_G3_IO3	PC11	X	X	X	X
	TSC_G3_IO4	PC12	X	X	X	X

Table 59. STM32L4 100-pin packages - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32L431/433/443/471/475/476/486/496/4A6Vx	STM32L451/452/462Vx	STM32L496/4A6Vx	
			LQFP100 - UFBGA100		LQFP100	WLCSP100
G4	TSC_G4_IO1	PC6	X	X	X	X
	TSC_G4_IO2	PC7	X	X	X	X
	TSC_G4_IO3	PC8	X	X	X	X
	TSC_G4_IO4	PC9	X	X	X	X
G5	TSC_G5_IO1	PE10	X	X	X	X
	TSC_G5_IO2	PE11	X	X	X	X
	TSC_G5_IO3	PE12	X	X	X	X
	TSC_G5_IO4	PE13	X	X	X	X
G6	TSC_G6_IO1	PD10	X	X	X	-
	TSC_G6_IO2	PD11	X	X	X	-
	TSC_G6_IO3	PD12	X	X	X	-
	TSC_G6_IO4	PD13	X	X	X	-
G7	TSC_G7_IO1	PE2	X	X	X	X
	TSC_G7_IO2	PE3	X	X	X	X
	TSC_G7_IO3	PE4	X	X	X	X
	TSC_G7_IO4	PE5	X	X	X	X
Maximum number of sensors			19	19	19	16

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. ‘X’ I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing SNR. It is recommended to use these I/Os as active shield or sampling capacitor (see the product errata sheet for information).
4. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

Table 60. STM32L4 115-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L496Wx
			WLCSP115
G1	TSC_G1_IO1	PB12	X(2)
	TSC_G1_IO2	PB13	X
	TSC_G1_IO3	PB14	X
G2	TSC_G2_IO1	PB4	(3)
	TSC_G2_IO2	PB5	(3)
	TSC_G2_IO3	PB6	X
	TSC_G2_IO4	PB7	(3)

Table 60. STM32L4 115-pin package - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32L496Wx
			WLCSP115
G3	TSC_G3_IO1	PA15	X
	TSC_G3_IO2	PC10	X
	TSC_G3_IO3	PC11	X
	TSC_G3_IO4	PC12	X
G4	TSC_G4_IO1	PC6	X
	TSC_G4_IO2	PC7	X
	TSC_G4_IO3	PC8	X
	TSC_G4_IO4	PC9	X
G5	TSC_G5_IO1	PE10	X
	TSC_G5_IO2	PE11	X
	TSC_G5_IO3	PE12	X
	TSC_G5_IO4	PE13	X
G7	TSC_G7_IO1	PE2	X
	TSC_G7_IO2	PE3	X
	TSC_G7_IO3	PE4	X
	TSC_G7_IO4	PE5	X
G8	TSC_G8_IO3	PG0	X
	TSC_G8_IO4	PG1	X
Maximum number of sensors			16

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing SNR. It is recommended to use these I/Os as active shield or sampling capacitor (see the product errata sheet for information).

Table 61. STM32L4 132-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L471/476/486Qx	STM32L496/4A6Qx
			UFBGA132	
G1	TSC_G1_IO1	PB12	X(2)	X
	TSC_G1_IO2	PB13	X	X
	TSC_G1_IO3	PB14	X	X
	TSC_G1_IO4	PB15	X	X

Table 61. STM32L4 132-pin package - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32L471/476/486Qx	STM32L496/4A6Qx
			UFBGA132	
G2	TSC_G2_IO1	PB4	(3)	(3)
	TSC_G2_IO2	PB5	(3)	(3)
	TSC_G2_IO3	PB6	X	X
	TSC_G2_IO4	PB7	(3)	(3)
G3	TSC_G3_IO1	PA15	(4)	(4)
	TSC_G3_IO2	PC10	X	X
	TSC_G3_IO3	PC11	X	X
	TSC_G3_IO4	PC12	X	X
G4	TSC_G4_IO1	PC6	X	X
	TSC_G4_IO2	PC7	X	X
	TSC_G4_IO3	PC8	X	X
	TSC_G4_IO4	PC9	X	X
G5	TSC_G5_IO1	PE10	X	X
	TSC_G5_IO2	PE11	X	X
	TSC_G5_IO3	PE12	X	X
	TSC_G5_IO4	PE13	X	X
G6	TSC_G6_IO1	PD10	X	X
	TSC_G6_IO2	PD11	X	X
	TSC_G6_IO3	PD12	X	X
	TSC_G6_IO4	PD13	X	X
G7	TSC_G7_IO1	PE2	X	X
	TSC_G7_IO2	PE3	X	X
	TSC_G7_IO3	PE4	X	X
	TSC_G7_IO4	PE5	X	X
G8	TSC_G8_IO1	PF14	X	(3)
	TSC_G8_IO2	PF15	X	(3)
	TSC_G8_IO3	PG0	X	X
	TSC_G8_IO4	PG1	X	X
Maximum number of sensors			22	21

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. ‘X’ I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing SNR. It is recommended to use these I/Os as active shield or sampling capacitor (see the product errata sheet for information).
4. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

Table 62. STM32L4 144-pin packages - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L471/476/486Zx	STM32L496/4A6Zx
			LQFP144 - UFBGA144	UFBGA144
G1	TSC_G1_IO1	PB12	X(2)	X
	TSC_G1_IO2	PB13	X	X
	TSC_G1_IO3	PB14	X	X
	TSC_G1_IO4	PB15	X	X
G2	TSC_G2_IO1	PB4	(3)	(3)
	TSC_G2_IO2	PB5	(3)	(3)
	TSC_G2_IO3	PB6	X	X
	TSC_G2_IO4	PB7	(3)	(3)
G3	TSC_G3_IO1	PA15	(4)	(4)
	TSC_G3_IO2	PC10	X	X
	TSC_G3_IO3	PC11	X	X
	TSC_G3_IO4	PC12	X	X
G4	TSC_G4_IO1	PC6	X	X
	TSC_G4_IO2	PC7	X	X
	TSC_G4_IO3	PC8	X	X
	TSC_G4_IO4	PC9	X	X
G5	TSC_G5_IO1	PE10	X	X
	TSC_G5_IO2	PE11	X	X
	TSC_G5_IO3	PE12	X	X
	TSC_G5_IO4	PE13	X	X
G6	TSC_G6_IO1	PD10	X	X
	TSC_G6_IO2	PD11	X	X
	TSC_G6_IO3	PD12	X	X
	TSC_G6_IO4	PD13	X	X
G7	TSC_G7_IO1	PE2	X	X
	TSC_G7_IO2	PE3	X	X
	TSC_G7_IO3	PE4	X	X
	TSC_G7_IO4	PE5	X	X
G8	TSC_G8_IO1	PF14	X	(3)
	TSC_G8_IO2	PF15	X	(3)
	TSC_G8_IO3	PG0	X	X
	TSC_G8_IO4	PG1	X	X
Maximum number of sensors			22	21

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing SNR. It is recommended to use these I/Os as active shield or sampling capacitor (see the product errata sheet for information).
4. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

Table 63. STM32L4 169-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L496//4A6Gx
			UFBGA169
G1	TSC_G1_IO1	PB12	X(2)
	TSC_G1_IO2	PB13	X
	TSC_G1_IO3	PB14	X
	TSC_G1_IO4	PB15	X
G2	TSC_G2_IO1	PB4	(3)
	TSC_G2_IO2	PB5	(3)
	TSC_G2_IO3	PB6	X
	TSC_G2_IO4	PB7	(3)
G3	TSC_G3_IO1	PA15	(4)
	TSC_G3_IO2	PC10	X
	TSC_G3_IO3	PC11	X
	TSC_G3_IO4	PC12	X
G4	TSC_G4_IO1	PC6	X
	TSC_G4_IO2	PC7	X
	TSC_G4_IO3	PC8	X
	TSC_G4_IO4	PC9	X
G5	TSC_G5_IO1	PE10	X
	TSC_G5_IO2	PE11	X
	TSC_G5_IO3	PE12	X
	TSC_G5_IO4	PE13	X
G6	TSC_G6_IO1	PD10	X
	TSC_G6_IO2	PD11	X
	TSC_G6_IO3	PD12	X
	TSC_G6_IO4	PD13	X
G7	TSC_G7_IO1	PE2	X
	TSC_G7_IO2	PE3	X
	TSC_G7_IO3	PE4	X
	TSC_G7_IO4	PE5	X

Table 63. STM32L4 169-pin package - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32L496//4A6Gx
			UFBGA169
G8	TSC_G8_IO1	PF14	(3)
	TSC_G8_IO2	PF15	(3)
	TSC_G8_IO3	PG0	X
	TSC_G8_IO4	PG1	X
Maximum number of sensors			21

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing SNR. It is recommended to use these I/Os as active shield or sampling capacitor (see the product errata sheet for information).
4. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

4.9.2 Hardware implementation example

Figure 20 shows an example of hardware implementation on STM32L4 series microcontrollers.

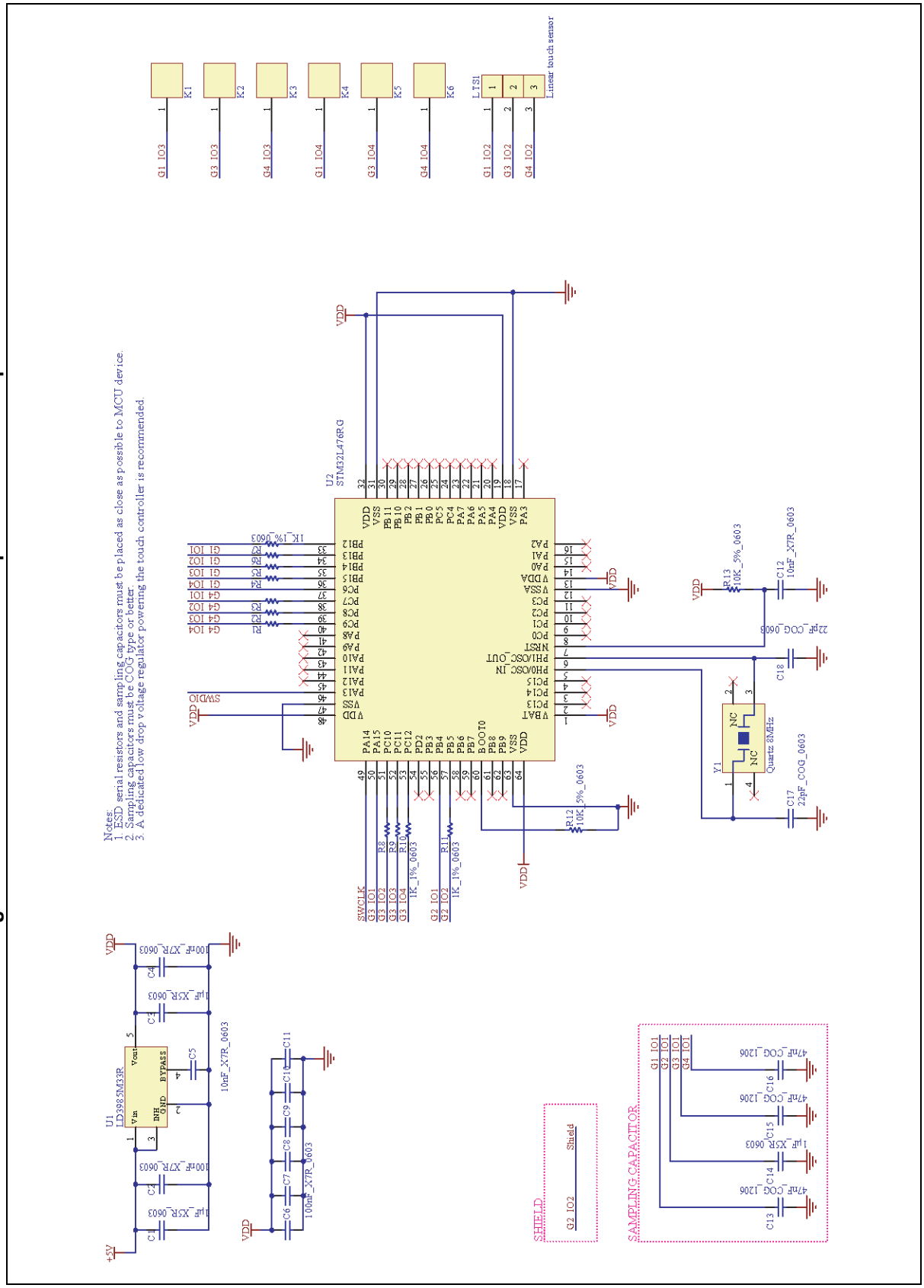


Figure 20. STM32L4 series hardware implementation example

4.10 STM32L4+ series microcontrollers

4.10.1 Available touch sensing channels

The tables below provide an overview of the available touch sensing channels for the STM32L4+ series microcontrollers.

The content of these tables is provided only for information.

Warning: For n available pins in an I/O group, one pin must be used as sampling capacitor and n-1 pins are used as channels.

Note: The I/O group cannot be used if the number of available pins in less or equal to one.

Table 64. STM32L4+ 48-pin packages - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L4P5/4Q5Cx
			LQFP48 - UFQFPN48
G1	TSC_G1_IO1	PB12	X(2)
	TSC_G1_IO2	PB13	X
	TSC_G1_IO3	PB14	X
	TSC_G1_IO4	PB15	X
G2	TSC_G2_IO1	PB4	(3)
	TSC_G2_IO2	PB5	(4)
	TSC_G2_IO3	PB6	X
	TSC_G2_IO4	PB7	(4)
Maximum number of sensors			4(5)

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.
4. All I/Os with this note offer a reduced touch sensing SNR. It is recommended to use these I/Os as active shield or sampling capacitor (see the product errata sheet for information).
5. Four sensors for all products except STM32L4Q5CGTxP/STM32L4Q5CGUxP for which there are maximum five sensors.

Table 65. STM32L4+ 64-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L4P5/4Q5Rx
			LQFP64
G1	TSC_G1_IO1	PB12	X(2)
	TSC_G1_IO2	PB13	X
	TSC_G1_IO3	PB14	X
	TSC_G1_IO4	PB15	X
G2	TSC_G2_IO1	PB4	X(3) or (4)
	TSC_G2_IO2	PB5	(5)
	TSC_G2_IO3	PB6	X
	TSC_G2_IO4	PB7	(5)
G3	TSC_G3_IO1	PA15	X(6) or (4)
	TSC_G3_IO2	PC10	X
	TSC_G3_IO3	PC11	X
	TSC_G3_IO4	PC12	X
G4	TSC_G4_IO1	PC6	X
	TSC_G4_IO2	PC7	X
	TSC_G4_IO3	PC8	X
	TSC_G4_IO4	PC9	X
Maximum number of sensors			10(7)

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. ‘X’ I/Os can be used as sampling capacitor, sensor, or active shield.
3. This I/O is X for STM32L4Q5RGTxP and (4) for all the other products.
4. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.
5. All I/Os with this note offer a reduced touch sensing SNR. It is recommended to use these I/Os as active shield or sampling capacitor (see the product errata sheet for information).
6. This I/O is X for STM32L4Q5RGTxP and (4) for all the others devices.
7. Ten sensors for all products except STM32L4Q5RGTxP for which there are maximum 11 sensors.

Table 66. STM32L4+ 100-pin packages - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L4 P5VGYxP/ R9V(G-I)Tx/ S9VITx	STM32L4 P5V(G-E)Tx/ P5VGTxP/ Q5VGTx/ R5V(G-I)Tx/ R7VITx/ S5VITx/ S7VITx	STM32L4 Q5VGTxP	STM32L4 P5V(G-E)Yx/ Q5VGYx	STM32L4 Q5VGYxP
			LQFP100 WLCSP100	LQFP100		WLCSP100	
G1	TSC_G1_IO1	PB12	X(2)	X	X	X	X
	TSC_G1_IO2	PB13	X	X	X	X	X
	TSC_G1_IO3	PB14	X	X	X	X	X
	TSC_G1_IO4	PB15	X	X	X	X	X
G2	TSC_G2_IO1	PB4	(3)	(3)	X	(3)	X
	TSC_G2_IO2	PB5	(4)	(4)	(4)	(4)	(4)
	TSC_G2_IO3	PB6	X	X	X	X	X
	TSC_G2_IO4	PB7	(4)	(4)	(4)	(4)	(4)
G3	TSC_G3_IO1	PA15	(3)	(3)	X	(3)	
	TSC_G3_IO2	PC10	X	X	X	X	X
	TSC_G3_IO3	PC11	X	X	X	X	X
	TSC_G3_IO4	PC12	X	X	X	X	X
G4	TSC_G4_IO1	PC6	X	X	X	X	X
	TSC_G4_IO2	PC7	X	X	X	X	X
	TSC_G4_IO3	PC8	X	X	X	X	X
	TSC_G4_IO4	PC9	X	X	X	X	X
G5	TSC_G5_IO1	PE10	X	X	X	X	X
	TSC_G5_IO2	PE11	X	X	X	X	X
	TSC_G5_IO3	PE12	X	X	X	X	X
	TSC_G5_IO4	PE13	X	X	X	X	X
G6	TSC_G6_IO1	PD10	-	X	X	-	-
	TSC_G6_IO2	PD11	-	X	X	-	-
	TSC_G6_IO3	PD12	-	X	X	-	-
	TSC_G6_IO4	PD13	-	X	X	-	-

Table 66. STM32L4+ 100-pin packages - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32L4 P5VGYxP/ R9V(G-I)Tx/ S9VITx	STM32L4 P5V(G-E)Tx/ P5VGTxP/ Q5VGTx/ R5V(G-I)Tx/ R7VITx/ S5VITx/ S7VITx	STM32L4 Q5VGTxP	STM32L4 P5V(G-E)Yx/ Q5VGYx	STM32L4 Q5VGYxP
			LQFP100 WLCSP100	LQFP100		WLCSP100	
G7	TSC_G7_IO1	PE2	X	X	X	X	X
	TSC_G7_IO2	PE3	X	X	X	X	X
	TSC_G7_IO3	PE4	X	X	X	X	X
	TSC_G7_IO4	PE5	X	X	X	X	X
Maximum number of sensors			16	19	20	16	17

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.
4. All I/Os with this note offer a reduced touch sensing SNR. It is recommended to use these I/Os as active shield or sampling capacitor (see the product errata sheet for information).

Table 67. STM32L4+ 132-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L4P5/4Q5/4R5/4S5Qx
			UFBGA132
G1	TSC_G1_IO1	PB12	X(2)
	TSC_G1_IO2	PB13	X
	TSC_G1_IO3	PB14	X
	TSC_G1_IO4	PB15	X
G2	TSC_G2_IO1	PB4	X(3) or (4)
	TSC_G2_IO2	PB5	(5)
	TSC_G2_IO3	PB6	X
	TSC_G2_IO4	PB7	(5)
G3	TSC_G3_IO1	PA15	X(6) or (4)
	TSC_G3_IO2	PC10	X
	TSC_G3_IO3	PC11	X
	TSC_G3_IO4	PC12	X

Table 67. STM32L4+ 132-pin package - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32L4P5/4Q5/4R5/4S5Qx
			UFBGA132
G4	TSC_G4_IO1	PC6	X
	TSC_G4_IO2	PC7	X
	TSC_G4_IO3	PC8	X
	TSC_G4_IO4	PC9	X
G5	TSC_G5_IO1	PE10	X
	TSC_G5_IO2	PE11	X
	TSC_G5_IO3	PE12	X
	TSC_G5_IO4	PE13	X
G6	TSC_G6_IO1	PD10	X
	TSC_G6_IO2	PD11	X
	TSC_G6_IO3	PD12	X
	TSC_G6_IO4	PD13	X
G7	TSC_G7_IO1	PE2	X
	TSC_G7_IO2	PE3	X
	TSC_G7_IO3	PE4	X
	TSC_G7_IO4	PE5	X
G8	TSC_G8_IO1	PF14	X
	TSC_G8_IO2	PF15	X
	TSC_G8_IO3	PG0	X
	TSC_G8_IO4	PG1	X
Maximum number of sensors			22(7)

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. This I/O is X for STM32L4Q5QGIxP and (4) for the other devices.
4. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.
5. All I/Os with this note offer a reduced touch sensing SNR. It is recommended to use these I/Os as active shield or sampling capacitor (see the product errata sheet for information).
6. This I/O is X for STM32L4Q5QGIxP and (4) for the others devices.
7. 22 sensors for all products except STM32L4Q5QGIxP for which there are maximum 23 sensors.

Table 68. STM32L4+ 144-pin packages - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L4R5/4R7/4R9Zx	STM32L4Q5Zx
			LQFP144 - UFBGA144 - WLCSP144	LQFP144
G1	TSC_G1_IO1	PB12	X(2)	X
	TSC_G1_IO2	PB13	X	X
	TSC_G1_IO3	PB14	X	X
	TSC_G1_IO4	PB15	X	X
G2	TSC_G2_IO1	PB4	(3)	X
	TSC_G2_IO2	PB5	(4)	(4)
	TSC_G2_IO3	PB6	X	X
	TSC_G2_IO4	PB7	(4)	(4)
G3	TSC_G3_IO1	PA15	(3)	X
	TSC_G3_IO2	PC10	X	X
	TSC_G3_IO3	PC11	X	X
	TSC_G3_IO4	PC12	X	X
G4	TSC_G4_IO1	PC6	X	X
	TSC_G4_IO2	PC7	X	X
	TSC_G4_IO3	PC8	X	X
	TSC_G4_IO4	PC9	X	X
G5	TSC_G5_IO1	PE10	X	X
	TSC_G5_IO2	PE11	X	X
	TSC_G5_IO3	PE12	X	X
	TSC_G5_IO4	PE13	X	X
G6	TSC_G6_IO1	PD10	X	X
	TSC_G6_IO2	PD11	X	X
	TSC_G6_IO3	PD12	X	X
	TSC_G6_IO4	PD13	X	X
G7	TSC_G7_IO1	PE2	X	X
	TSC_G7_IO2	PE3	X	X
	TSC_G7_IO3	PE4	X	X
	TSC_G7_IO4	PE5	X	X
G8	TSC_G8_IO1	PF14	X	X
	TSC_G8_IO2	PF15	X	X
	TSC_G8_IO3	PG0	X	X
	TSC_G8_IO4	PG1	X	X
Maximum number of sensors			22	23

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.
4. All I/Os with this note offer a reduced touch sensing SNR. It is recommended to use these I/Os as active shield or sampling capacitor (see the product errata sheet for information).

Table 69. STM32L4+ 169-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L4P5/4Q5/4R5/4S5Qx
			UFBGA169
G1	TSC_G1_IO1	PB12	X(2)
	TSC_G1_IO2	PB13	X
	TSC_G1_IO3	PB14	X
	TSC_G1_IO4	PB15	X
G2	TSC_G2_IO1	PB4	X(3) or (4)
	TSC_G2_IO2	PB5	(5)
	TSC_G2_IO3	PB6	X
	TSC_G2_IO4	PB7	(5)
G3	TSC_G3_IO1	PA15	X(6) or (4)
	TSC_G3_IO2	PC10	X
	TSC_G3_IO3	PC11	X
	TSC_G3_IO4	PC12	X
G4	TSC_G4_IO1	PC6	X
	TSC_G4_IO2	PC7	X
	TSC_G4_IO3	PC8	X
	TSC_G4_IO4	PC9	X
G5	TSC_G5_IO1	PE10	X
	TSC_G5_IO2	PE11	X
	TSC_G5_IO3	PE12	X
	TSC_G5_IO4	PE13	X
G6	TSC_G6_IO1	PD10	X
	TSC_G6_IO2	PD11	X
	TSC_G6_IO3	PD12	X
	TSC_G6_IO4	PD13	X
G7	TSC_G7_IO1	PE2	X
	TSC_G7_IO2	PE3	X
	TSC_G7_IO3	PE4	X
	TSC_G7_IO4	PE5	X

Table 69. STM32L4+ 169-pin package - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32L4P5/4Q5/4R5/4S5Qx
			UFBGA169
G8	TSC_G8_IO1	PF14	X
	TSC_G8_IO2	PF15	X
	TSC_G8_IO3	PG0	X
	TSC_G8_IO4	PG1	X
Maximum number of sensors			22(7)

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. This I/O is X for STM32L4Q5AGIxP and (4) for the other devices.
4. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.
5. All I/Os with this note offer a reduced touch sensing SNR. It is recommended to use these I/Os as active shield or sampling capacitor (see the product errata sheet for information).
6. This I/O is X for STM32L4Q5AGIxP and (4) for the others devices.
7. 22 sensors for all products except STM32L4Q5AGIxP for which there are maximum 23 sensors.

4.11 STM32L5 series microcontrollers

4.11.1 Available touch sensing channels

The tables below provide an overview of the available touch sensing channels for the STM32L5 series microcontrollers.

The content of these tables is provided only for information.

Warning: For n available pins in an I/O group, one pin must be used as sampling capacitor and n-1 pins are used as channels.

Note: The I/O group cannot be used if the number of available pins in less or equal to one.

Table 70. STM32L5 48-pin packages - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L552/562Cx
			LQFP48 - UFQFPN48
G1	TSC_G1_IO1	PB12	X(2)
	TSC_G1_IO2	PB13	X
	TSC_G1_IO3	PB14	X
G2	TSC_G2_IO1	PB4	(3)
	TSC_G2_IO2	PB5	(4)
	TSC_G2_IO3	PB6	(4)
	TSC_G2_IO4	PB7	(4)
Maximum number of sensors			2

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.
4. All I/Os with this note offer a reduced touch sensing SNR. It is recommended to use these I/Os as active shield or sampling capacitor (see the product errata sheet for information).

Table 71. STM32L5 64-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L5 52RETxQ	STM32L5 52R(C-E)Tx/ 52RETxP/ 62RETx/ 62RETxP	STM32L5 62RETxQ
			LQFP64		
G1	TSC_G1_IO1	PB12	-	X	-
	TSC_G1_IO2	PB13	X(2)	X	X
	TSC_G1_IO3	PB14	X	X	X
G2	TSC_G2_IO1	PB4	(3)	(3)	X
	TSC_G2_IO2	PB5	(4)	(4)	(4)
	TSC_G2_IO3	PB6	(4)	(4)	(4)
	TSC_G2_IO4	PB7	(4)	(4)	(4)
G3	TSC_G3_IO2	PC10	X	X	X
	TSC_G3_IO3	PC11	X	X	X
	TSC_G3_IO4	PC12	X	X	X
G4	TSC_G4_IO1	PC6	X	X	X
	TSC_G4_IO2	PC7	X	X	X
	TSC_G4_IO3	PC8	X	X	X
	TSC_G4_IO4	PC9	X	X	X
Maximum number of sensors			6	7	7

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.
4. All I/Os with this note offer a reduced touch sensing SNR. It is recommended to use these I/Os as active shield or sampling capacitor (see the product errata sheet for information).

Table 72. STM32L5 81-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L5 52MEYxP/ 62MEYxP	STM32L5 52MEYxQ/ 62MEYxQ
			WLCSP81	
G1	TSC_G1_IO1	PB12	X(2)	X
	TSC_G1_IO2	PB13	X	X
	TSC_G1_IO3	PB14	X	X

Table 72. STM32L5 81-pin package - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32L5 52MEYxP/ 62MEYxP	STM32L5 52MEYxQ/ 62MEYxQ
			WLCSP81	
G2	TSC_G2_IO1	PB4	(3)	(3)
	TSC_G2_IO2	PB5	(4)	(4)
	TSC_G2_IO3	PB6	(4)	(4)
	TSC_G2_IO4	PB7	(4)	(4)
G3	TSC_G3_IO2	PC10	X	X
	TSC_G3_IO3	PC11	X	X
	TSC_G3_IO4	PC12	X	X
G4	TSC_G4_IO1	PC6	X	X
	TSC_G4_IO2	PC7	X	X
	TSC_G4_IO3	PC8	X	X
	TSC_G4_IO4	PC9	X	X
Maximum number of sensors			7	7

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. ‘X’ I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.
4. All I/Os with this note offer a reduced touch sensing SNR. It is recommended to use these I/Os as active shield or sampling capacitor (see the product errata sheet for information).

Table 73. STM32L5 100-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L5 52VETx/ 62VETx	STM32L5 52V(C-E)TxQ/ 62VETxQ
			LQFP100	
G1	TSC_G1_IO1	PB12	X(2)	-
	TSC_G1_IO2	PB13	X	X
	TSC_G1_IO3	PB14	X	X
G2	TSC_G2_IO1	PB4	(3)	(3)
	TSC_G2_IO2	PB5	(4)	(4)
	TSC_G2_IO3	PB6	(4)	(4)
	TSC_G2_IO4	PB7	(4)	(4)

Table 73. STM32L5 100-pin package - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32L5 52VETx/ 62VETx	STM32L5 52V(C-E)TxQ/ 62VETxQ
			LQFP100	
G3	TSC_G3_IO2	PC10	X	X
	TSC_G3_IO3	PC11	X	X
	TSC_G3_IO4	PC12	X	X
G4	TSC_G4_IO1	PC6	X	X
	TSC_G4_IO2	PC7	X	X
	TSC_G4_IO3	PC8	X	X
	TSC_G4_IO4	PC9	X	X
G5	TSC_G5_IO1	PE10	X	X
	TSC_G5_IO2	PE11	X	X
	TSC_G5_IO3	PE12	X	X
	TSC_G5_IO4	PE13	X	X
G6	TSC_G6_IO1	PD10	X	X
	TSC_G6_IO2	PD11	X	X
	TSC_G6_IO3	PD12	X	X
	TSC_G7_IO4	PD13	X	X
G7	TSC_G7_IO1	PE2	X	X
	TSC_G7_IO2	PE3	X	X
	TSC_G7_IO3	PE4	X	X
	TSC_G7_IO4	PE5	X	X
Maximum number of sensors			16	15

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. ‘X’ I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.
4. All I/Os with this note offer a reduced touch sensing SNR. It is recommended to use these I/Os as active shield or sampling capacitor (see the product errata sheet for information).

Table 74. STM32L5 132-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L5 62QEIx	STM32L5 52Q(C-E)IxQ/ 52QEIx/ 52QEIxP/ 62QEIxP/ 62QEIxQ
			UFBGA132	
G1	TSC_G1_IO1	PB12	X(2)	-
	TSC_G1_IO2	PB13	X	X
	TSC_G1_IO3	PB14	X	X
G2	TSC_G2_IO1	PB4	X	(3)
	TSC_G2_IO2	PB5	(4)	(4)
	TSC_G2_IO3	PB6	(4)	(4)
	TSC_G2_IO4	PB7	(4)	(4)
G3	TSC_G3_IO2	PC10	X	X
	TSC_G3_IO3	PC11	X	X
	TSC_G3_IO4	PC12	X	X
G4	TSC_G4_IO1	PC6	X	X
	TSC_G4_IO2	PC7	X	X
	TSC_G4_IO3	PC8	X	X
	TSC_G4_IO4	PC9	X	X
G5	TSC_G5_IO1	PE10	X	X
	TSC_G5_IO2	PE11	X	X
	TSC_G5_IO3	PE12	X	X
	TSC_G5_IO4	PE13	X	X
G6	TSC_G6_IO1	PD10	X	X
	TSC_G6_IO2	PD11	X	X
	TSC_G6_IO3	PD12	X	X
	TSC_G7_IO4	PD13	X	X
G7	TSC_G7_IO1	PE2	X	X
	TSC_G7_IO2	PE3	X	X
	TSC_G7_IO3	PE4	X	X
	TSC_G7_IO4	PE5	X	X

Table 74. STM32L5 132-pin package - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32L5 62QEIx	STM32L5 52Q(C-E)IxQ/ 52QEIx/ 52QEIxP/ 62QEIxP/ 62QEIxQ
			UFBGA132	
G8	TSC_G8_IO1	PF14	X	X
	TSC_G8_IO2	PF15	X	X
	TSC_G8_IO3	PG0	X	X
	TSC_G8_IO4	PG1	X	X
Maximum number of sensors			20	19

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.
4. All I/Os with this note offer a reduced touch sensing SNR. It is recommended to use these I/Os as active shield or sampling capacitor (see the product errata sheet for information).

Table 75. STM32L5 144-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L5 552ETx/ 562ETx	STM32L5 52Z(C-E)TxQ/ 62ZETxQ
			LQFP144	
G1	TSC_G1_IO1	PB12	X(2)	-
	TSC_G1_IO2	PB13	X	X
	TSC_G1_IO3	PB14	X	X
G2	TSC_G2_IO1	PB4	(3)	(3)
	TSC_G2_IO2	PB5	(4)	(4)
	TSC_G2_IO3	PB6	(4)	(4)
	TSC_G2_IO4	PB7	(4)	(4)
G3	TSC_G3_IO2	PC10	X	X
	TSC_G3_IO3	PC11	X	X
	TSC_G3_IO4	PC12	X	X
G4	TSC_G4_IO1	PC6	X	X
	TSC_G4_IO2	PC7	X	X
	TSC_G4_IO3	PC8	X	X
	TSC_G4_IO4	PC9	X	X

Table 75. STM32L5 144-pin package - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32L5 552ETx/ 562ETx	STM32L5 52Z(C-E)TxQ/ 62ZETxQ
			LQFP144	
G5	TSC_G5_IO1	PE10	X	X
	TSC_G5_IO2	PE11	X	X
	TSC_G5_IO3	PE12	X	X
	TSC_G5_IO4	PE13	X	X
G6	TSC_G6_IO1	PD10	X	X
	TSC_G6_IO2	PD11	X	X
	TSC_G6_IO3	PD12	X	X
	TSC_G7_IO4	PD13	X	X
G7	TSC_G7_IO1	PE2	X	X
	TSC_G7_IO2	PE3	X	X
	TSC_G7_IO3	PE4	X	X
	TSC_G7_IO4	PE5	X	X
G8	TSC_G8_IO1	PF14	X	X
	TSC_G8_IO2	PF15	X	X
	TSC_G8_IO3	PG0	X	X
	TSC_G8_IO4	PG1	X	X
Maximum number of sensors			19	18

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. ‘X’ I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.
4. All I/Os with this note offer a reduced touch sensing SNR. It is recommended to use these I/Os as active shield or sampling capacitor (see the product errata sheet for information).

4.12 STM32U0 series microcontrollers

4.12.1 Available touch sensing channels

The tables below provide an overview of the available touch sensing channels for the STM32U0 series microcontrollers.

The content of these tables is provided only for information.

Warning: For n available pins in an I/O group, one pin must be used as sampling capacitor and n-1 pins are used as channels.

Note: The I/O group cannot be used if the number of available pins in less or equal to one.

Table 76. STM32U0 20-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32U0 31F4Px/31F6Px/31F8Px
			TSSOP-20
G2	TSC_G2_IO1	PB4	X(2)
	TSC_G2_IO2	PB5	X
	TSC_G2_IO3	PB6	X
	TSC_G2_IO4	PB7	X
G5	TSC_G5_IO1	PA6	X
	TSC_G5_IO2	PB0	X
G7	TSC_G7_IO1	PA8	X
	TSC_G7_IO2	PA9	X
	TSC_G7_IO3	PA10	X
	TSC_G7_IO4	PA13	X
Maximum number of sensors			7

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.

Table 77. STM32U0 27-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32U0 31F4Px/31F6Px/31F8Px
			WLCSP27
G2	TSC_G2_IO2	PB5	X(2)
	TSC_G2_IO3	PB6	X
	TSC_G2_IO4	PB7	X

Table 77. STM32U0 27-pin package - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32U0 31F4Px/31F6Px/31F8Px
			WLCSP27
G5	TSC_G5_IO1	PA6	X
	TSC_G5_IO2	PB0	X
G7	TSC_G7_IO1	PA8	X
	TSC_G7_IO2	PA9	X
	TSC_G7_IO3	PA10	X
	TSC_G7_IO4	PA13	X
Maximum number of sensors			7

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.

Table 78. STM32U0 32-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32U0 31K4Ux/31K6Ux/31K8Ux 73K8Ux/73KBUx/73KCUx/ 83KCUx
			UFQFPN32
G2	TSC_G2_IO1	PB4	X(2)
	TSC_G2_IO2	PB5	X
	TSC_G2_IO3	PB6	X
	TSC_G2_IO4	PB7	X
G3	TSC_G3_IO1	PA15	X
	TSC_G3_IO4	PA14	X
G5	TSC_G5_IO1	PA6	X
	TSC_G5_IO2	PB0	X
G7	TSC_G7_IO1	PA8	X
	TSC_G7_IO2	PA9	X
	TSC_G7_IO3	PA10	X
	TSC_G7_IO4	PA13	X
Maximum number of sensors			8

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.

Table 79. STM32U0 48-pin packages - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32U0 31C6Tx/31C8Tx/ 73C8Tx/73CBTx/ 73CCTx/83CCTx	STM32U0 31C6Ux/31C8Ux/ 73C8Ux/73CBUx/ 73CCUx/83CCUx
			LQFP48 - UFQFPN48	
G1	TSC_G1_IO1	PB12	X(2)	X
	TSC_G1_IO2	PB13	X	X
	TSC_G1_IO3	PB14	X	X
	TSC_G1_IO4	PB15	X	X
G2	TSC_G2_IO1	PB4	X	X
	TSC_G2_IO2	PB5	X	X
	TSC_G2_IO3	PB6	X	X
	TSC_G2_IO4	PB7	X	X
G3	TSC_G3_IO1	PA15	X	X
	TSC_G3_IO4	PA14	X	X
G5	TSC_G5_IO1	PA6	X	X
	TSC_G5_IO2	PB0	X	X
	TSC_G5_IO3	PB10	X	X
	TSC_G5_IO4	PB11	X	X
G7	TSC_G7_IO1	PA8	X	X
	TSC_G7_IO2	PA9	X	X
	TSC_G7_IO3	PA10	X	X
	TSC_G7_IO4	PA13	X	X
Maximum number of sensors			13	13

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.

Table 80. STM32U0 64-pin packages - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32U0 31R6Ix/31R8Ix/ 73R8Ix/73RBIx/ 73RCIx/83RCIx	STM32U0 31R6Tx/31R8Tx/ 73R8Tx/73RBTx/ 73RCTx/83RCTx
			LQFP64 – UFBGA64	
G1	TSC_G1_IO1	PB12	X(2)	X
	TSC_G1_IO2	PB13	X	X
	TSC_G1_IO3	PB14	X	X
	TSC_G1_IO4	PB15	X	X

Table 80. STM32U0 64-pin packages - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32U0 31R6Ix/31R8Ix/ 73R8Ix/73RBIx/ 73RCIx/83RCIx	STM32U0 31R6Tx/31R8Tx/ 73R8Tx/73RBTx/ 73RCTx/83RCTx
			LQFP64 – UFBGA64	
G2	TSC_G2_IO1	PB4	X	X
	TSC_G2_IO2	PB5	X	X
	TSC_G2_IO3	PB6	X	X
	TSC_G2_IO4	PB7	X	X
G3	TSC_G3_IO1	PA15	X	X
	TSC_G3_IO2	PC10	X	X
	TSC_G3_IO3	PC11	X	X
	TSC_G3_IO4	PA14	X	X
G4	TSC_G4_IO1	PC6	X	X
	TSC_G4_IO2	PC7	X	X
	TSC_G4_IO3	PC8	X	X
	TSC_G4_IO4	PC9	X	X
G5	TSC_G5_IO1	PA6	X	X
	TSC_G5_IO2	PB0	X	X
	TSC_G5_IO3	PB10	X	X
	TSC_G5_IO4	PB11	X	X
G7	TSC_G7_IO1	PA8	X	X
	TSC_G7_IO2	PA9	X	X
	TSC_G7_IO3	PA10	X	X
	TSC_G7_IO4	PA13	X	X
Maximum number of sensors			18	18

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. ‘X’ I/Os can be used as sampling capacitor, sensor, or active shield.

Table 81. STM32U0 80-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32U0 73M8Tx/73MBTx/73MCTx/ 83MCTx
			LQFP80
G1	TSC_G1_IO1	PB12	X(2)
	TSC_G1_IO2	PB13	X
	TSC_G1_IO3	PB14	X
	TSC_G1_IO4	PB15	X
G2	TSC_G2_IO1	PB4	X
	TSC_G2_IO2	PB5	X
	TSC_G2_IO3	PB6	X
	TSC_G2_IO4	PB7	X
G3	TSC_G3_IO1	PA15	X
	TSC_G3_IO2	PC10	X
	TSC_G3_IO3	PC11	X
	TSC_G3_IO4	PA14	X
G4	TSC_G4_IO1	PC6	X
	TSC_G4_IO2	PC7	X
	TSC_G4_IO3	PC8	X
	TSC_G4_IO4	PC9	X
G5	TSC_G5_IO1	PA6	X
	TSC_G5_IO2	PB0	X
	TSC_G5_IO3	PB10	X
	TSC_G5_IO4	PB11	X
G6	TSC_G6_IO1	PD10	X
	TSC_G6_IO2	PD11	X
	TSC_G6_IO3	PD12	X
	TSC_G6_IO4	PD13	X
G7	TSC_G7_IO1	PA8	X
	TSC_G7_IO2	PA9	X
	TSC_G7_IO3	PA10	X
	TSC_G7_IO4	PA13	X
Maximum number of sensors			21

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. ‘X’ I/Os can be used as sampling capacitor, sensor, or active shield.

Table 82. STM32U0 81-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32U0 73M8Ix/73MBIx/73MCIx/ 83MCIx
			UFBGA81
G1	TSC_G1_IO1	PB12	X(2)
	TSC_G1_IO2	PB13	X
	TSC_G1_IO3	PB14	X
	TSC_G1_IO4	PB15	X
G2	TSC_G2_IO1	PB4	X
	TSC_G2_IO2	PB5	X
	TSC_G2_IO3	PB6	X
	TSC_G2_IO4	PB7	X
G3	TSC_G3_IO1	PA15	X
	TSC_G3_IO2	PC10	X
	TSC_G3_IO3	PC11	X
	TSC_G3_IO4	PA14	X
G4	TSC_G4_IO1	PC6	X
	TSC_G4_IO2	PC7	X
	TSC_G4_IO3	PC8	X
	TSC_G4_IO4	PC9	X
G5	TSC_G5_IO1	PA6	X
	TSC_G5_IO2	PB0	X
	TSC_G5_IO3	PB10	X
	TSC_G5_IO4	PB11	X
G6	TSC_G6_IO1	PD10	X
	TSC_G6_IO2	PD11	X
	TSC_G6_IO3	PD12	X
	TSC_G6_IO4	PD13	X
G7	TSC_G7_IO1	PA8	X
	TSC_G7_IO2	PA9	X
	TSC_G7_IO3	PA10	X
	TSC_G7_IO4	PA13	X
Maximum number of sensors			21

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. ‘X’ I/Os can be used as sampling capacitor, sensor, or active shield.

4.13 STM32U3 series microcontrollers

4.13.1 Available touch sensing channels

The tables below provide an overview of the available touch sensing channels for the STM32U3 series microcontrollers.

The content of these tables is provided only for information.

Warning: For n available pins in an I/O group, one pin must be used as sampling capacitor and n-1 pins are used as channels.

Note: The I/O group cannot be used if the number of available pins in less or equal to one.

Table 83. STM32U3 48-pin packages - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32U3 75CETx/75CEUx 75CGTx/75CGUx 85CGTx/85CGUx	STM32U3 75CETxQ/75CEUxQ 75CGTxQ/75CGUxQ 85CGTxQ/85CGUxQ
			LQFP48 - UFQFPN48	
G1	TSC_G1_IO1	PB12	X(2)	X
	TSC_G1_IO2	PB13	X	X
	TSC_G1_IO3	PB14	X	X
	TSC_G1_IO4	PB15	X	X
G2	TSC_G2_IO1	PB4	X	X
	TSC_G2_IO2	PB5	X	X
	TSC_G2_IO3	PB6	X	X
	TSC_G2_IO4	PB7	X	X
Maximum number of sensors			6	6

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. ‘X’ I/Os can be used as sampling capacitor, sensor, or active shield.

Table 84. STM32U3 64-pin packages - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32U3 75RETx/75RElx 75RGTx/75RGlx 85RGTx/85RGlx	STM32U3 75RETxQ/75RElxQ 75RGTxQ/75RGlxQ 85RGTxQ/85RGlxQ
			LQFP64 – UFBGA64	
G1	TSC_G1_IO1	PB12	X(2)	X
	TSC_G1_IO2	PB13	X	X
	TSC_G1_IO3	PB14	X	X
	TSC_G1_IO4	PB15	X	X
G2	TSC_G2_IO1	PB4	X	X
	TSC_G2_IO2	PB5	X	X
	TSC_G2_IO3	PB6	X	X
	TSC_G2_IO4	PB7	X	X
G3	TSC_G3_IO1	PA15	X	X
	TSC_G3_IO2	PC10	X	X
	TSC_G3_IO3	PC11	X	X
	TSC_G3_IO4	PC12	X	X
G4	TSC_G4_IO1	PC6	X	X
	TSC_G4_IO2	PC7	X	X
	TSC_G4_IO3	PC8	X	X
	TSC_G4_IO4	PC9	X	X
Maximum number of sensors			12	12

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. ‘X’ I/Os can be used as sampling capacitor, sensor, or active shield.

Table 85. STM32U3 100-pin packages - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32U3 75VETx/75VElx 75VGTx/75VGlx 85VGTx/85VGlx	STM32U3 75VETxQ/75VElxQ 75VGTxQ/75VGlxQ 85VGTxQ/85VGlxQ
			LQFP100 – UFBGA100	
G1	TSC_G1_IO1	PB12	X(2)	X
	TSC_G1_IO2	PB13	X	X
	TSC_G1_IO3	PB14	X	X
	TSC_G1_IO4	PB15	X	X

Table 85. STM32U3 100-pin packages - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32U3 75VETx/75VEIx 75VGTx/75VGIx 85VGTx/85VGIx	STM32U3 75VETxQ/75VEIxQ 75VGTxQ/75VGIxQ 85VGTxQ/85VGIxQ
			LQFP100 – UFBGA100	
G2	TSC_G2_IO1	PB4	X	X
	TSC_G2_IO2	PB5	X	X
	TSC_G2_IO3	PB6	X	X
	TSC_G2_IO4	PB7	X	X
G3	TSC_G3_IO1	PA15	X	X
	TSC_G3_IO2	PC10	X	X
	TSC_G3_IO3	PC11	X	X
	TSC_G3_IO4	PC12	X	X
G4	TSC_G4_IO1	PC6	X	X
	TSC_G4_IO2	PC7	X	X
	TSC_G4_IO3	PC8	X	X
	TSC_G4_IO4	PC9	X	X
G5	TSC_G5_IO1	PE10	X	X
	TSC_G5_IO2	PE11	X	X
	TSC_G5_IO3	PE12	X	X
	TSC_G5_IO4	PE13	X	X
G6	TSC_G6_IO1	PD10	X	X
	TSC_G6_IO2	PD11	X	X
	TSC_G6_IO3	PD12	X	X
	TSC_G6_IO4	PD13	X	X
G7	TSC_G7_IO1	PE2	X	X
	TSC_G7_IO2	PE3	X	X
	TSC_G7_IO3	PE4	X	X
	TSC_G7_IO4	PE5	X	X
Maximum number of sensors			21	21

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. ‘X’ I/Os can be used as sampling capacitor, sensor, or active shield.

Table 86. STM32U3 32-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32U3 75KEUx/75KGUx 85KGUx
			UQFN32
G2	TSC_G2_IO1	PB4	X(2)
	TSC_G2_IO2	PB5	X
	TSC_G2_IO3	PB6	X
	TSC_G2_IO4	PB7	X
Maximum number of sensors			3

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.

Table 87. STM32U3 52-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32U3 75CEYxQ/75CGYxQ 85CGYxQ
			WLCSP52
G1	TSC_G1_IO2	PB13	X(2)
	TSC_G1_IO3	PB14	X
	TSC_G1_IO4	PB15	X
G2	TSC_G2_IO1	PB4	X
	TSC_G2_IO2	PB5	X
	TSC_G2_IO3	PB6	X
	TSC_G2_IO4	PB7	X
Maximum number of sensors			5

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.

Table 88. STM32U3 68-pin package (1 of 2) - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32U3 75REYxQ/75RGYxQ 85RGYxQ
			WLCSP68
G1	TSC_G1_IO1	PB12	X(2)
	TSC_G1_IO2	PB13	X
	TSC_G1_IO3	PB14	X
	TSC_G1_IO4	PB15	X
G2	TSC_G2_IO1	PB4	X
	TSC_G2_IO2	PB5	X
	TSC_G2_IO3	PB6	X
	TSC_G2_IO4	PB7	X
G3	TSC_G3_IO1	PA15	X
	TSC_G3_IO2	PC10	X
	TSC_G3_IO3	PC11	X
	TSC_G3_IO4	PC12	X
G4	TSC_G4_IO1	PC6	X
	TSC_G4_IO2	PC7	X
	TSC_G4_IO3	PC8	X
	TSC_G4_IO4	PC9	X
G6	TSC_G6_IO3	PD12	X
	TSC_G6_IO4	PD13	X
Maximum number of sensors			13

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.

Table 89. STM32U3 68-pin package (2 of 2) - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32U3 75REYxG/75RGYxG 85RGYxG
			WLCSP68
G1	TSC_G1_IO2	PB13	X(2)
	TSC_G1_IO3	PB14	X
	TSC_G1_IO4	PB15	X

Table 89. STM32U3 68-pin package (2 of 2) - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32U3 75REYxG/75RGYxG 85RGYxG
			WLCSP68
G2	TSC_G2_IO1	PB4	X
	TSC_G2_IO2	PB5	X
	TSC_G2_IO3	PB6	X
	TSC_G2_IO4	PB7	X
Maximum number of sensors			5

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.

4.14 STM32U5 series microcontrollers

4.14.1 Available touch sensing channels

The tables below provide an overview of the available touch sensing channels for the STM32U5 series microcontrollers.

The content of these tables is provided only for information.

Warning: For n available pins in an I/O group, one pin must be used as sampling capacitor and n-1 pins are used as channels.

Note: The I/O group cannot be used if the number of available pins in less or equal to one.

Table 90. STM32U5 48-pin packages - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32U5 75CGTx/75CGUx/ 75CITx/75CIUx/ 85CITx/85CIUx	STM32U5 75CGTxQ/75CGUxQ/ 75CITxQ/75CIUxQ/ 85CITxQ/85CIUxQ
			LQFP48 - UFQFPN48	
G1	TSC_G1_IO1	PB12	X(2)	-
	TSC_G1_IO2	PB13	X	X
	TSC_G1_IO3	PB14	X	X

Table 90. STM32U5 48-pin packages - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32U5 75CGTx/75CGUx/ 75CITx/75CIUx/ 85CITx/85CIUx	STM32U5 75CGTxQ/75CGUxQ/ 75CITxQ/75CIUxQ/ 85CITxQ/85CIUxQ
			LQFP48 - UFQFPN48	
G2	TSC_G2_IO1	PB4	X	X
	TSC_G2_IO2	PB5	X	X
	TSC_G2_IO3	PB6	X	X
	TSC_G2_IO4	PB7	X	X
Maximum number of sensors			5	4

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.

Table 91. STM32U5 64-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32U5 75RGTx/75RITx/ 85RITx	STM32U5 75RGTxQ/75RITxQ/ 85RITxQ
			LQFP64	
G1	TSC_G1_IO1	PB12	X(2)	-
	TSC_G1_IO2	PB13	X	X
	TSC_G1_IO3	PB14	X	X
	TSC_G1_IO4	PC3	X	X
G2	TSC_G2_IO1	PB4	X	X
	TSC_G2_IO2	PB5	X	X
	TSC_G2_IO3	PB6	X	X
	TSC_G2_IO4	PB7	X	X
G3	TSC_G3_IO1	PC2	X	X
	TSC_G3_IO2	PC10	X	X
	TSC_G3_IO3	PC11	X	X
	TSC_G3_IO4	PC12	X	X
G4	TSC_G4_IO1	PC6	X	X
	TSC_G4_IO2	PC7	X	X
	TSC_G4_IO3	PC8	X	X
	TSC_G4_IO4	PC9	X	X
Maximum number of sensors			12	11

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.

2. ‘X’ I/Os can be used as sampling capacitor, sensor, or active shield.

Table 92. STM32U5 90-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32U575/585Ox
			WLCSP90
G1	TSC_G1_IO2	PB13	X(2)
	TSC_G1_IO3	PB14	X
	TSC_G1_IO4	PC3	X
G2	TSC_G2_IO1	PB4	X
	TSC_G2_IO2	PB5	X
	TSC_G2_IO3	PB6	X
	TSC_G2_IO4	PB7	X
G3	TSC_G3_IO1	PC2	X
	TSC_G3_IO2	PC10	X
	TSC_G3_IO3	PC11	X
	TSC_G3_IO4	PC12	X
G4	TSC_G4_IO1	PC6	X
	TSC_G4_IO2	PC7	X
	TSC_G4_IO3	PC8	X
	TSC_G4_IO4	PC9	X
G7	TSC_G7_IO2	PE3	X
	TSC_G7_IO3	PE4	X
	TSC_G7_IO4	PE5	X
Maximum number of sensors			13

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. ‘X’ I/Os can be used as sampling capacitor, sensor, or active shield.

Table 93. STM32U5 100-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32U5 75VGTx/75VITx/ 85VITx	STM32U5 75VGTxQ/75VITxQ/ 85VITxQ
			LQFP100	
G1	TSC_G1_IO1	PB12	X(2)	-
	TSC_G1_IO2	PB13	X	X
	TSC_G1_IO3	PB14	X	X
	TSC_G1_IO4	PC3	X	X

Table 93. STM32U5 100-pin package - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32U5 75VGTx/75VITx/ 85VITx	STM32U5 75VGTxQ/75VITxQ/ 85VITxQ
			LQFP100	
G2	TSC_G2_IO1	PB4	X	X
	TSC_G2_IO2	PB5	X	X
	TSC_G2_IO3	PB6	X	X
	TSC_G2_IO4	PB7	X	X
G3	TSC_G3_IO1	PC2	X	X
	TSC_G3_IO2	PC10	X	X
	TSC_G3_IO3	PC11	X	X
	TSC_G3_IO4	PC12	X	X
G4	TSC_G4_IO1	PC6	X	X
	TSC_G4_IO2	PC7	X	X
	TSC_G4_IO3	PC8	X	X
	TSC_G4_IO4	PC9	X	X
G5	TSC_G5_IO1	PE10	X	X
	TSC_G5_IO2	PE11	X	X
	TSC_G5_IO3	PE12	X	X
	TSC_G5_IO4	PE13	X	X
G6	TSC_G6_IO1	PD10	X	X
	TSC_G6_IO2	PD11	X	X
	TSC_G6_IO3	PD12	X	X
	TSC_G6_IO4	PD13	X	X
G7	TSC_G7_IO1	PE2	X	X
	TSC_G7_IO2	PE3	X	X
	TSC_G7_IO3	PE4	X	X
	TSC_G7_IO4	PE5	X	X
Maximum number of sensors			21	20

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. ‘X’ I/Os can be used as sampling capacitor, sensor, or active shield.

Table 94. STM32U5 132-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32U575/585Qx
			UFBGA132
G1	TSC_G1_IO1	PB12	X(2)
	TSC_G1_IO2	PB13	X
	TSC_G1_IO3	PB14	X
	TSC_G1_IO4	PC3	X
G2	TSC_G2_IO1	PB4	X
	TSC_G2_IO2	PB5	X
	TSC_G2_IO3	PB6	X
	TSC_G2_IO4	PB7	X
G3	TSC_G3_IO1	PC2	X
	TSC_G3_IO2	PC10	X
	TSC_G3_IO3	PC11	X
	TSC_G3_IO4	PC12	X
G4	TSC_G4_IO1	PC6	X
	TSC_G4_IO2	PC7	X
	TSC_G4_IO3	PC8	X
	TSC_G4_IO4	PC9	X
G5	TSC_G5_IO1	PE10	X
	TSC_G5_IO2	PE11	X
	TSC_G5_IO3	PE12	X
	TSC_G5_IO4	PE13	X
G6	TSC_G6_IO1	PD10	X
	TSC_G6_IO2	PD11	X
	TSC_G6_IO3	PD12	X
	TSC_G6_IO4	PD13	X
G7	TSC_G7_IO1	PE2	X
	TSC_G7_IO2	PE3	X
	TSC_G7_IO3	PE4	X
	TSC_G7_IO4	PE5	X
G8	TSC_G8_IO1	PF14	X
	TSC_G8_IO2	PF15	X
	TSC_G8_IO3	PG0	X
	TSC_G8_IO4	PG1	X
Maximum number of sensors			24

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.

Table 95. STM32U5 144-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32U5 75ZGTx/75ZITx/ 85ZITx	STM32U5 75ZGTxQ/75ZITx/Q 85ZITxQ
			LQFP144	
G1	TSC_G1_IO1	PB12	X(2)	-
	TSC_G1_IO2	PB13	X	X
	TSC_G1_IO3	PB14	X	X
	TSC_G1_IO4	PC3	X	X
G2	TSC_G2_IO1	PB4	X	X
	TSC_G2_IO2	PB5	X	X
	TSC_G2_IO3	PB6	X	X
	TSC_G2_IO4	PB7	X	X
G3	TSC_G3_IO1	PC2	X	X
	TSC_G3_IO2	PC10	X	X
	TSC_G3_IO3	PC11	X	X
	TSC_G3_IO4	PC12	X	X
G4	TSC_G4_IO1	PC6	X	X
	TSC_G4_IO2	PC7	X	X
	TSC_G4_IO3	PC8	X	X
	TSC_G4_IO4	PC9	X	X
G5	TSC_G5_IO1	PE10	X	X
	TSC_G5_IO2	PE11	X	X
	TSC_G5_IO3	PE12	X	X
	TSC_G5_IO4	PE13	X	X
G6	TSC_G6_IO1	PD10	X	X
	TSC_G6_IO2	PD11	X	X
	TSC_G6_IO3	PD12	X	X
	TSC_G6_IO4	PD13	X	X
G7	TSC_G7_IO1	PE2	X	X
	TSC_G7_IO2	PE3	X	X
	TSC_G7_IO3	PE4	X	X
	TSC_G7_IO4	PE5	X	X

Table 95. STM32U5 144-pin package - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32U5 75ZGTx/75ZITx/ 85ZITx	STM32U5 75ZGTxQ/75ZITx/Q 85ZITxQ
			LQFP144	
G8	TSC_G8_IO1	PF14	X	X
	TSC_G8_IO2	PF15	X	X
	TSC_G8_IO3	PG0	X	X
	TSC_G8_IO4	PG1	X	X
Maximum number of sensors			24	23

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.

Table 96. STM32U5 169-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32U575/585Ax
			UFBGA169
G1	TSC_G1_IO1	PB12	X(2)
	TSC_G1_IO2	PB13	X
	TSC_G1_IO3	PB14	X
	TSC_G1_IO4	PC3	X
G2	TSC_G2_IO1	PB4	X
	TSC_G2_IO2	PB5	X
	TSC_G2_IO3	PB6	X
	TSC_G2_IO4	PB7	X
G3	TSC_G3_IO1	PC2	X
	TSC_G3_IO2	PC10	X
	TSC_G3_IO3	PC11	X
	TSC_G3_IO4	PC12	X
G4	TSC_G4_IO1	PC6	X
	TSC_G4_IO2	PC7	X
	TSC_G4_IO3	PC8	X
	TSC_G4_IO4	PC9	X

Table 96. STM32U5 169-pin package - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32U575/585Ax
			UFBGA169
G5	TSC_G5_IO1	PE10	X
	TSC_G5_IO2	PE11	X
	TSC_G5_IO3	PE12	X
	TSC_G5_IO4	PE13	X
G6	TSC_G6_IO1	PD10	X
	TSC_G6_IO2	PD11	X
	TSC_G6_IO3	PD12	X
	TSC_G6_IO4	PD13	X
G7	TSC_G7_IO1	PE2	X
	TSC_G7_IO2	PE3	X
	TSC_G7_IO3	PE4	X
	TSC_G7_IO4	PE5	X
G8	TSC_G8_IO1	PF14	X
	TSC_G8_IO2	PF15	X
	TSC_G8_IO3	PG0	X
	TSC_G8_IO4	PG1	X
Maximum number of sensors			24

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.

4.15 STM32WB series microcontrollers

4.15.1 Available touch sensing channels

The tables below provide an overview of the available touch sensing channels for the STM32WB series microcontrollers.

The content of these tables is provided only for information.

Warning: For n available pins in an I/O group, one pin must be used as sampling capacitor and n-1 pins are used as channels.

Note: The I/O group cannot be used if the number of available pins in less or equal to one.

Table 97. STM32WB 48-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32WB 15CCUxE	STM32WB 10CCUx/15CCUx
			UFQFPN48	
G1	TSC_G1_IO1	PB12	X(2)	-
	TSC_G1_IO2	PB13	X	-
G2	TSC_G2_IO1	PB4	X	X
	TSC_G2_IO2	PB5	(3)	(3)
	TSC_G2_IO3	PB6	X	X
	TSC_G2_IO4	PB7	(4)	(4)
G3	TSC_G3_IO1	PA15	X	-
	TSC_G3_IO2	PB10	X	-
	TSC_G3_IO3	PC11	X	-
G7	TSC_G7_IO1	PA13	X	X
	TSC_G7_IO2	PA10	X	X
	TSC_G7_IO3	PB8	X	X
	TSC_G7_IO4	PB9	X	X
Maximum number of sensors			8	5

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. ‘X’ I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.
4. All I/Os with this note offer a reduced touch sensing SNR. It is recommended to use these I/Os as active shield or sampling capacitor (see the product errata sheet for information).

Table 98. STM32WB 49-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32WB15CCYx
			WLCSP49
G2	TSC_G2_IO1	PB4	X(2)
	TSC_G2_IO2	PB5	(3)
	TSC_G2_IO3	PB6	X
	TSC_G2_IO4	PB7	(4)
G7	TSC_G7_IO1	PA13	X
	TSC_G7_IO2	PA10	X
	TSC_G7_IO3	PB8	X
Maximum number of sensors			4

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.
4. All I/Os with this note offer a reduced touch sensing SNR. It is recommended to use these I/Os as active shield or sampling capacitor (see the product errata sheet for information).

Table 99. STM32WB 68-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32WB55Rx
			VFQFPN68
G1	TSC_G1_IO1	PB12	X(2)
	TSC_G1_IO2	PB13	X
	TSC_G1_IO3	PB14	X
	TSC_G1_IO4	PB15	X
G2	TSC_G2_IO1	PB4	(3)
	TSC_G2_IO2	PB5	(3)
	TSC_G2_IO3	PB6	(3)
	TSC_G2_IO4	PB7	(3)
G3	TSC_G3_IO1	PA15	X
	TSC_G3_IO2	PC10	X
	TSC_G3_IO3	PC11	X
	TSC_G3_IO4	PC12	(4)
Maximum number of sensors			6

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing SNR. It is recommended to use these I/Os as active shield or sampling capacitor (see the product errata sheet for information).

4. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

Table 100. STM32WB 86-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32WB55Mx
			LGA86
G1	TSC_G1_IO1	PB12	X(2)
	TSC_G1_IO2	PB13	X
	TSC_G1_IO3	PB14	X
	TSC_G1_IO4	PB15	X
G2	TSC_G2_IO1	PB4	(3)
	TSC_G2_IO2	PB5	(3)
	TSC_G2_IO3	PB6	(3)
	TSC_G2_IO4	PB7	(3)
G3	TSC_G3_IO1	PA15	X
	TSC_G3_IO2	PC10	X
	TSC_G3_IO3	PC11	X
	TSC_G3_IO4	PC12	(4)
G4	TSC_G4_IO1	PC6	X
	TSC_G4_IO2	PC7	X
	TSC_G4_IO3	PC8	X
	TSC_G4_IO4	PC9	X
G5	TSC_G5_IO1	PD4	X
	TSC_G5_IO2	PD5	X
	TSC_G5_IO3	PD6	X
	TSC_G5_IO4	PD7	X
G6	TSC_G6_IO1	PD10	X
	TSC_G6_IO2	PD11	X
	TSC_G6_IO3	PD12	X
	TSC_G6_IO4	PD13	X
G7	TSC_G7_IO1	PE2	X
	TSC_G7_IO2	PE1	X
	TSC_G7_IO3	PE0	X
	TSC_G7_IO4	PB9	X
Maximum number of sensors			18

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. ‘X’ I/Os can be used as sampling capacitor, sensor, or active shield.

3. All I/Os with this note offer a reduced touch sensing SNR. It is recommended to use these I/Os as active shield or sampling capacitor (see the product errata sheet for information).
4. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

Table 101. STM32WB 100-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32WB55Vx
			WLCSP100
G1	TSC_G1_IO1	PB12	X(2)
	TSC_G1_IO2	PB13	X
	TSC_G1_IO3	PB14	X
	TSC_G1_IO4	PB15	X
G2	TSC_G2_IO1	PB4	(3)
	TSC_G2_IO2	PB5	(3)
	TSC_G2_IO3	PB6	(3)
	TSC_G2_IO4	PB7	(3)
G3	TSC_G3_IO1	PA15	X
	TSC_G3_IO2	PC10	X
	TSC_G3_IO3	PC11	X
	TSC_G3_IO4	PC12	(4)
G4	TSC_G4_IO1	PC6	X
	TSC_G4_IO2	PC7	X
	TSC_G4_IO3	PC8	X
	TSC_G4_IO4	PC9	X
G5	TSC_G5_IO1	PD4	X
	TSC_G5_IO2	PD5	X
	TSC_G5_IO3	PD6	X
	TSC_G5_IO4	PD7	X
G6	TSC_G6_IO1	PD10	X
	TSC_G6_IO2	PD11	X
	TSC_G6_IO3	PD12	X
	TSC_G6_IO4	PD13	X
G7	TSC_G7_IO1	PE2	X
	TSC_G7_IO2	PE1	X
	TSC_G7_IO3	PE0	X
	TSC_G7_IO4	PB9	X
Maximum number of sensors			18

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing SNR. It is recommended to use these I/Os as active shield or sampling capacitor (see the product errata sheet for information).
4. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

Table 102. STM32WB 129-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32WB55Vx
			UFBGA129
G1	TSC_G1_IO1	PB12	X(2)
	TSC_G1_IO2	PB13	X
	TSC_G1_IO3	PB14	X
	TSC_G1_IO4	PB15	X
G2	TSC_G2_IO1	PB4	(3)
	TSC_G2_IO2	PB5	(3)
	TSC_G2_IO3	PB6	(3)
	TSC_G2_IO4	PB7	(3)
G3	TSC_G3_IO1	PA15	X
	TSC_G3_IO2	PC10	X
	TSC_G3_IO3	PC11	X
	TSC_G3_IO4	PC12	(4)
G4	TSC_G4_IO1	PC6	X
	TSC_G4_IO2	PC7	X
	TSC_G4_IO3	PC8	X
	TSC_G4_IO4	PC9	X
G5	TSC_G5_IO1	PD4	X
	TSC_G5_IO2	PD5	X
	TSC_G5_IO3	PD6	X
	TSC_G5_IO4	PD7	X
G6	TSC_G6_IO1	PD10	X
	TSC_G6_IO2	PD11	X
	TSC_G6_IO3	PD12	X
	TSC_G6_IO4	PD13	X

Table 102. STM32WB 129-pin package - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32WB55Vx
			UFBGA129
G7	TSC_G7_IO1	PE2	X
	TSC_G7_IO2	PE1	X
	TSC_G7_IO3	PE0	X
	TSC_G7_IO4	PB9	X
Maximum number of sensors			18

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing SNR. It is recommended to use these I/Os as active shield or sampling capacitor (see the product errata sheet for information).
4. All I/Os with this note offer a reduced touch sensing sensitivity. These I/Os can be used only as sampling capacitor.

5 STM32L1 with hardware assist

This section concerns the STM32L1 series microcontrollers that do not include the TSC.These microcontrollers support two different acquisition modes: hardware and software.

5.1 Acquisition

The STM32L1 hardware acquisition mode (using two timers) is done in the files:

- *tsl_acq_stm32l1xx_hw.c*
- *tsl_acq_stm32l1xx_hw.h*

Warning: This acquisition mode is only available for the STM32L1 devices featuring a minimum of 384 Kbytes of Flash memory.

The STM32L1 software acquisition mode is done in the files:

- *tsl_acq_stm32l1xx_sw.c*
- *tsl_acq_stm32l1xx_sw.h*

This acquisition is available for all STM32L1microcontrollers.

Note: The hardware acquisition mode is selected per default for the STM32L1 devices featuring a minimum of 384 Kbytes of Flash memory. For the software acquisition mode, the user must add the `TSLPRM_STM32L1XX_SW_ACQ` *constant in the toolchain compiler preprocessor.*

The following functions are used by the application layer and are device dependent:

- `TSL_acq_BankConfig()`
- `TSL_acq_BankStartAcq()`
- `TSL_acq_BankWaitEOC()`
- `TSL_acq_GetMeas()`

The other functions in xxx_hw/sw.c/h files are for internal use and the user does not need to call them directly.

5.2 Timings

The timing management is done in the files:

- *tsl_time.c*
- *tsl_time.h*

The SysTick is used to generate a timebase for ECS and DTO modules. It must be initialized in the user code (already done by the `HAL_init` function).

5.3 Parameters

The STM32L1 specific parameters are:

- described in *tsl_conf_stm32l1xx_template.h* file (to be copied in project and renamed in *tsl_conf.h*)
- checked in *tsl_check_config_stm32l1xx.h* file

5.4 Memory footprint

Conditions

- IAR ANSI C/C++ compiler/linker V7.40.3.8902 for Arm
- Compiler optimization: high size
- Counted files: tsl*.o
- STM32 touch sensing library options: ECS = ON, DTO = ON, DXS = OFF, PROX = OFF
- Each sensor has its own parameters placed in RAM.

The following tables summarize the memory footprint with different configurations

Table 103. STM32L1 series with hardware acquisition mode memory footprint(1)

Channels	Banks	Sensors	ROM (Kbytes)	RAM (bytes)
1	1	1 TKey	5.3	340
2	1	2 TKeys	5.3	360
2	2	2 TKeys	5.5	360
24	3	24 TKeys	6.2	870
3	1	1 Linear-3ch	6.3	370
15	3	12 TKeys + 1 Linear-3ch	8.3	660
24	3	18 TKeys + 2 Linear-3ch	8.5	850

1. The content of this table is provided only for information.

Table 104. STM32L1 series with software acquisition mode memory footprint(1)

Channels	Banks	Sensors	ROM (Kbytes)	RAM (bytes)
1	1	1 TKey	5.5	410
2	1	2 TKeys	5.5	430
2	2	2 TKeys	5.5	430
24	3	24 TKeys	6.2	930
3	1	1 Linear-3ch	6.5	440
15	3	12 TKeys + 1 Linear-3ch	8.2	730
24	3	18 TKeys + 2 Linear-3ch	8.5	920

1. The content of this table is provided only for information.

5.5 MCU resources

The tables below show peripherals that are used by the STMTouch touch sensing library on STM32L1 devices. Care must be taken when using them to avoid any unwanted behavior.

Table 105. MCU resources used on STM32L1 with hardware acquisition

Peripheral	Function
GPIOs	Acquisition
SysTick	Time base for ECS and DTO
2 timers (TIM9, TIM11)	Acquisition
Routing interface	Acquisition

Table 106. MCU resources used on STM32L1 with software acquisition

Peripheral	Function
GPIOs	Acquisition
SysTick	Time base for ECS and DTO
Routing interface	Acquisition

5.6 Available touch sensing channels

The tables below provide an overview of the available touch sensing channels for the STM32L1 series microcontrollers.

The content of these tables is provided only for information.

Warning: For n available pins in an I/O group, one pin must be used as sampling capacitor and n-1 pins are used as channels.

Note: The I/O group cannot be used if the number of available pins in less or equal to one.

Table 107. STM32L1 48-pin packages - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L1 51CCTx/51CCUx/ 51C(6-8-B)TxA/51C(6-8-B)UxA/ 52CCTx/52CCUx/ 52C(6-8-B)TxA/52C(6-8-B)UxA	STM32L1 51C(6-8-B)Tx/51C(6-8-B)Ux/ 52C(6-8-B)Tx/52C(6-8-B)Ux
			LQFP48 - UFQFPN48	
G1	TSC_G1_IO1	PA0-WKUP1	X(2)	X
	TSC_G1_IO2	PA1	X	X
	TSC_G1_IO3	PA2	X	X
	TSC_G1_IO4	PA3	(3)	(3)
G2	TSC_G2_IO1	PA6	X	X
	TSC_G2_IO2	PA7	X	X
G3	TSC_G3_IO1	PB0	(3)	X
	TSC_G3_IO2	PB1	X	X
	TSC_G3_IO3	PB2	X	-
G4	TSC_G4_IO1	PA8	X	X
	TSC_G4_IO2	PA9	X	X
	TSC_G4_IO3	PA10	X	X
G5	TSC_G5_IO1	PA13	X	X
	TSC_G5_IO2	PA14	X	X
	TSC_G5_IO3	PA15	X	X
G6	TSC_G6_IO1	PB4	X	X
	TSC_G6_IO2	PB5	X	X
	TSC_G6_IO3	PB6	X	-
	TSC_G7_IO4	PB7	X	-
G7	TSC_G7_IO1	PB12	X	X
	TSC_G7_IO2	PB13	X	X
	TSC_G7_IO3	PB14	X	X
	TSC_G7_IO4	PB15	X	X
Maximum number of sensors			16	13

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. ‘X’ I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. Only one of these I/Os can be used as sampling capacitor. The others cannot be used.

Table 108. STM32L1 63-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L151/152Ux
			WLCSP63
G1	TSC_G1_IO1	PA0-WKUP1	X(2)
	TSC_G1_IO2	PA1	X
	TSC_G1_IO3	PA2	X
	TSC_G1_IO4	PA3	(3)
G2	TSC_G2_IO1	PA6	X
	TSC_G2_IO2	PA7	X
G3	TSC_G3_IO1	PB0	(3)
	TSC_G3_IO2	PB1	X
	TSC_G3_IO3	PB2	X
G4	TSC_G4_IO1	PA8	X
	TSC_G4_IO2	PA9	X
	TSC_G4_IO3	PA10	X
G5	TSC_G5_IO1	PA13	X
	TSC_G5_IO2	PA14	X
	TSC_G5_IO3	PA15	X
G6	TSC_G6_IO1	PB4	X
	TSC_G6_IO2	PB5	X
	TSC_G6_IO3	PB6	X
	TSC_G7_IO4	PB7	X
G7	TSC_G7_IO1	PB12	X
	TSC_G7_IO2	PB13	X
	TSC_G7_IO3	PB14	X
	TSC_G7_IO4	PB15	X
G8	TSC_G8_IO1	PC0	X
	TSC_G8_IO2	PC1	X
	TSC_G8_IO3	PC2	X
	TSC_G8_IO4	PC3	(3)
G9	TSC_G9_IO1	PC4	X
	TSC_G9_IO2	PC5	X

Table 108. STM32L1 63-pin package - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32L151/152Ux
			WLCSP63
G10	TSC_G10_IO1		X
	TSC_G10_IO2		X
	TSC_G10_IO3		X
	TSC_G10_IO4		X
Maximum number of sensors			23

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. Only one of these I/Os can be used as sampling capacitor. The others cannot be used.

Table 109. STM32L1 64-pin packages - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L1 51RCTx/51RCYx/ 51R(6-8-B)TxA/ 51RETx/51RDYx/ 51RCTxA/51RDTx/ 52RCTxA/ 52RCTx/52RETx/ 52R(6-8-B)TxA/ 52RDTx/52RDYx/ 62RCTx/ 62RCTxA/62RDTx/ 62RDYx/62RETx	STM32L1 51R(6-8-B)HxA/ 52R(6-8-B)HxA	STM32L1 51R(6-8-B)Hx/ 52R(6-8-B)Hx	STM32L1 51R(6-8-B)Tx/ 52R(6-8-B)Tx
			LQFP64 - WLCSP64	TFBGA64		LQFP64
G1	TSC_G1_IO1	PA0-WKUP1	X(2)	X	X	X
	TSC_G1_IO2	PA1	X	X	X	X
	TSC_G1_IO3	PA2	X	X	X	X
	TSC_G1_IO4	PA3	(3)	(3)	(3)	(3)
G2	TSC_G2_IO1	PA6	X	X	X	X
	TSC_G2_IO2	PA7	X	X	X	X
G3	TSC_G3_IO1	PB0	(3)	(3)	(3)	(3)
	TSC_G3_IO2	PB1	X	X	X	X
	TSC_G3_IO3	PB2	X	X	-	-
G4	TSC_G4_IO1	PA8	X	X	X	X
	TSC_G4_IO2	PA9	X	X	X	X
	TSC_G4_IO3	PA10	X	X	X	X

Table 109. STM32L1 64-pin packages - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32L1 51RCTx/51RCYx/ 51R(6-8-B)TxA/ 51RETx/51RDYx/ 51RCTxA/51RDTx/ 52RCTxA/ 52RCTx/52RETx/ 52R(6-8-B)TxA/ 52RDTx/52RDYx/ 62RCTx/ 62RCTxA/62RDTx/ 62RDYx/62RETx	STM32L1 51R(6-8-B)HxA/ 52R(6-8-B)HxA	STM32L1 51R(6-8-B)Hx/ 52R(6-8-B)Hx	STM32L1 51R(6-8-B)Tx/ 52R(6-8-B)Tx
			LQFP64 - WLCSP64	TFBGA64		LQFP64
G5	TSC_G5_IO1	PA13	X	X	X	X
	TSC_G5_IO2	PA14	X	X	X	X
	TSC_G5_IO3	PA15	X	X	X	X
G6	TSC_G6_IO1	PB4	X	X	X	X
	TSC_G6_IO2	PB5	X	X	X	X
	TSC_G6_IO3	PB6	X	X	-	-
	TSC_G7_IO4	PB7	X	X	-	-
G7	TSC_G7_IO1	PB12	X	X	X	X
	TSC_G7_IO2	PB13	X	X	X	X
	TSC_G7_IO3	PB14	X	X	X	X
	TSC_G7_IO4	PB15	X	X	X	X
G8	TSC_G8_IO1	PC0	X	X	X	X
	TSC_G8_IO2	PC1	X	X	X	X
	TSC_G8_IO3	PC2	X	X	X	X
	TSC_G8_IO4	PC3	(3)	-	-	(3)
G9	TSC_G9_IO1	PC4	X	X	X	X
	TSC_G9_IO2	PC5	X	X	X	X
G10	TSC_G10_IO1	PC6	X	X	X	X
	TSC_G10_IO2	PC7	X	X	X	X
	TSC_G10_IO3	PC8	X	X	X	X
	TSC_G10_IO4	PC9	X	X	X	X
Maximum number of sensors			23	22	19	20

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. Only one of these I/Os can be used as sampling capacitor. The others cannot be used.

Table 110. STM32L1 100-pin packages - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L1 51VCHx/51VCTx/51V(8-B)TxA/ 51V(8-B)HxA/52V(8-B)HxA/ 52VCHx/52VCTx/52V(8-B)/ 62VCHx/62VCTx/	STM32L1 51V(8-B)Hx/ 51V(8-B)Tx/ 52V(8-B)Hx/ 52V(8-B)Tx
			LQFP100 - UFBGA100	
G1	TSC_G1_IO1	PA0-WKUP1	X(2)	X
	TSC_G1_IO2	PA1	X	X
	TSC_G1_IO3	PA2	X	X
	TSC_G1_IO4	PA3	(3)	(3)
G2	TSC_G2_IO1	PA6	X	X
	TSC_G2_IO2	PA7	X	X
G3	TSC_G3_IO1	PB0	(3)	(3)
	TSC_G3_IO2	PB1	X	X
	TSC_G3_IO3	PB2	X	-
G4	TSC_G4_IO1	PA8	X	X
	TSC_G4_IO2	PA9	X	X
	TSC_G4_IO3	PA10	X	X
G5	TSC_G5_IO1	PA13	X	X
	TSC_G5_IO2	PA14	X	X
	TSC_G5_IO3	PA15	X	X
G6	TSC_G6_IO1	PB4	X	X
	TSC_G6_IO2	PB5	X	X
	TSC_G6_IO3	PB6	X	-
	TSC_G7_IO4	PB7	X	-
G7	TSC_G7_IO1	PB12	X	X
	TSC_G7_IO2	PB13	X	X
	TSC_G7_IO3	PB14	X	X
	TSC_G7_IO4	PB15	X	X
G8	TSC_G8_IO1	PC0	X	X
	TSC_G8_IO2	PC1	X	X
	TSC_G8_IO3	PC2	X	X
	TSC_G8_IO4	PC3	(3)	(3)
G9	TSC_G9_IO1	PC4	X	X
	TSC_G9_IO2	PC5	X	X

Table 110. STM32L1 100-pin packages - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32L1 51VCHx/51VCTx/51V(8-B)TxA/ 51V(8-B)HxA/52V(8-B)HxA/ 52VCHx/52VCTx/52V(8-B)/ 62VCHx/62VCTx/	STM32L1 51V(8-B)Hx/ 51V(8-B)Tx/ 52V(8-B)Hx/ 52V(8-B)Tx
			LQFP100 - UFBGA100	
G10	TSC_G10_IO1	PC6	X	X
	TSC_G10_IO2	PC7	X	X
	TSC_G10_IO3	PC8	X	X
	TSC_G10_IO4	PC9	X	X
Maximum number of sensors			23	20

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. Only one of these I/Os can be used as sampling capacitor. The others cannot be used.

Table 111. STM32L1 104-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L1 51VDYxX/51VEYx/52VEYx/ 62VDYxX/62VEYx
			WLCSP104
G1	TSC_G1_IO1	PA0-WKUP1	X(2)
	TSC_G1_IO2	PA1	X
	TSC_G1_IO3	PA2	X
	TSC_G1_IO4	PA3	(3)
G2	TSC_G2_IO1	PA6	X
	TSC_G2_IO2	PA7	X
G3	TSC_G3_IO1	PB0	(3)
	TSC_G3_IO2	PB1	X
	TSC_G3_IO3	PB2	X
G4	TSC_G4_IO1	PA8	X
	TSC_G4_IO2	PA9	X
	TSC_G4_IO3	PA10	X
G5	TSC_G5_IO1	PA13	X
	TSC_G5_IO2	PA14	X
	TSC_G5_IO3	PA15	X

Table 111. STM32L1 104-pin package - Touch sensing channels(1) (continued)

Group	Capacitive sensing signal name	Pin name	STM32L1 51VDYxX/51VEYx/52VEYx/ 62VDYxX/62VEYx
			WLCSP104
G6	TSC_G6_IO1	PB4	X
	TSC_G6_IO2	PB5	X
	TSC_G6_IO3	PB6	X
	TSC_G7_IO4	PB7	X
G7	TSC_G7_IO1	PB12	X
	TSC_G7_IO2	PB13	X
	TSC_G7_IO3	PB14	X
	TSC_G7_IO4	PB15	X
G8	TSC_G8_IO1	PC0	X
	TSC_G8_IO2	PC1	X
	TSC_G8_IO3	PC2	X
	TSC_G8_IO4	PC3	(3)
G9	TSC_G9_IO1	PC4	X
	TSC_G9_IO2	PC5	X
G10	TSC_G10_IO1	PC6	X
	TSC_G10_IO2	PC7	X
	TSC_G10_IO3	PC8	X
	TSC_G10_IO4	PC9	X
Maximum number of sensors			23

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. Only one of these I/Os can be used as sampling capacitor. The others cannot be used.

Table 112. STM32L1 132-pin package - Touch sensing channels(1)

Group	Capacitive sensing signal name	Pin name	STM32L1 51QCHx/51QDHx/51QEHx/ 52QCHx/52QDHx/52QEHx/ 62QCHx/62QDHx/
			UFBGA132
G1	TSC_G1_IO1	PA0-WKUP1	X(2)
	TSC_G1_IO2	PA1	X
	TSC_G1_IO3	PA2	X
	TSC_G1_IO4	PA3	(3)
G2	TSC_G2_IO1	PA6	X
	TSC_G2_IO2	PA7	X
	TSC_G2_IO3	PF15	X
	TSC_G2_IO4	PG0	(3)
	TSC_G2_IO5	PG1	(3)
G3	TSC_G3_IO1	PB0	(3)
	TSC_G3_IO2	PB1	X
	TSC_G3_IO3	PB2	X
	TSC_G3_IO4	PF11	X
	TSC_G2_IO5	PF12	X
G4	TSC_G4_IO1	PA8	X
	TSC_G4_IO2	PA9	X
	TSC_G4_IO3	PA10	X
G5	TSC_G5_IO1	PA13	X
	TSC_G5_IO2	PA14	X
	TSC_G5_IO3	PA15	X
G6	TSC_G6_IO1	PB4	X
	TSC_G6_IO2	PB5	X
	TSC_G6_IO3	PB6	X
	TSC_G7_IO4	PB7	X
G7	TSC_G7_IO1	PB12	X
	TSC_G7_IO2	PB13	X
	TSC_G7_IO3	PB14	X
	TSC_G7_IO4	PB15	X
	TSC_G7_IO5	PG2	(3)
	TSC_G7_IO6	PG3	(3)
	TSC_G7_IO7	PG4	(3)

Table 112. STM32L1 132-pin package - Touch sensing channels(1) (continued)

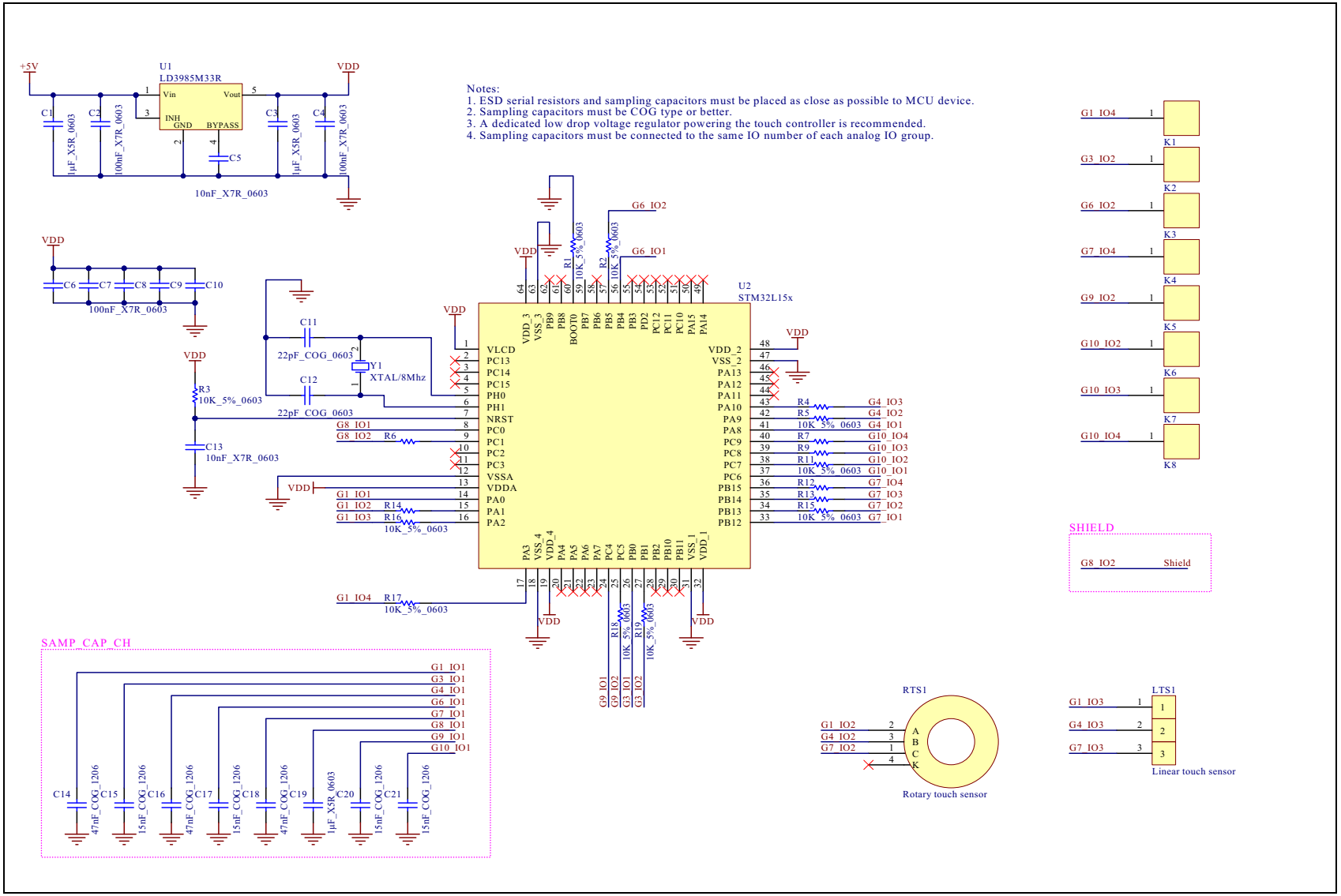
Group	Capacitive sensing signal name	Pin name	STM32L1 51QCHx/51QDHx/51QEHx/ 52QCHx/52QDHx/52QEHx/ 62QCHx/62QDHx/
			UFBGA132
G8	TSC_G8_IO1	PC0	X
	TSC_G8_IO2	PC1	X
	TSC_G8_IO3	PC2	X
	TSC_G8_IO4	PC3	(3)
G9	TSC_G9_IO1	PC4	X
	TSC_G9_IO2	PC5	X
	TSC_G9_IO3	PF13	X
	TSC_G9_IO4	PF14	X
G10	TSC_G10_IO1	PC6	X
	TSC_G10_IO2	PC7	X
	TSC_G10_IO3	PC8	X
	TSC_G10_IO4	PC9	X
G11	TSC_G11_IO1	PF6	X
	TSC_G11_IO2	PF7	X
	TSC_G11_IO3	PF8	X
	TSC_G11_IO4	PF9	X
	TSC_G11_IO5	PF10	X
Maximum number of sensors			34

1. One sampling capacitor mandatory for each group (and no more than one). No more than one active shield per group.
2. 'X' I/Os can be used as sampling capacitor, sensor, or active shield.
3. All I/Os with this note offer a reduced touch sensing sensitivity. Only one of these I/Os can be used as sampling capacitor. The others cannot be used.

5.7 Hardware implementation example

Figure 21 shows an example of hardware implementation on STM32L1 series microcontrollers.

Figure 21. STM32L1 series hardware implementation example



6 Getting started

6.1 Create the application

Start with an application example present in the STM32Cube package of the device. Take an example that is close in term of number of channels/sensors with the target application. Copy/paste the example in the same parent folder and rename it according to the target application. Then modify the files as described below.

6.1.1 Toolchain compiler preprocessor section

The device must be written in the toolchain compiler preprocessor section of the project. These defines are the same as those used by the STM32Cube (see the *stm<xxx>.h* map file).

Note: The hardware acquisition mode is selected by default for the STM32L1 with a minimum of 384-Kbyte Flash memory. For the software acquisition mode, the user must add the `TSLPRM_STM32L1XX_SW_ACQ` *constant in the toolchain compiler preprocessor.*

6.1.2 *tsl_conf.h* file

tsl_conf.h contains all the STMTouch touch sensing library parameters. The following edits must be done:

1. Change the number of channels, banks, sensors according to the application.
2. Change the common parameters (such as thresholds, debounce, ECS, or DTO).
3. Change the parameters specific to the device.

6.1.3 *main.c and main.h* files

main.c and *main.h* contain the application code itself (such as LEDs and LCD management) These files also includes the call to the STMTouch touch sensing library initialization and action functions.

6.1.4 *tsl_user*.c and *tsl_user*.h files

tsl_user.c and *tsl_user.h* contain the STMTouch touch sensing library configuration (such as channel, bank, or sensor definitions).These files also includes the STMTouch touch sensing library initialization (`TSL_user_Init`) and action (`TSL_user_Action`) functions.

Create the channel variables using the structures (**mandatory**):

- `TSL_ChannelSrc_T`
- `TSL_ChannelDest_T`
- `TSL_ChannelData_T`

Create the bank variables using the structure (**mandatory**):

- `TSL_Bank_T`

Create the touchkey variables using the structures (optional):

- `TSL_TouchKeyData_T`
- `TSL_TouchKeyParam_T`
- `TSL_State_T`

- TSL_TouchKeyMethods_T
- TSL_TouchKeyB_T
- TSL_TouchKey_T

Create the linear and rotary touch sensor variables using the structures (optional):

- TSL_LinRotData_T
- TSL_LinRotParam_T
- TSL_State_T
- TSL_LinRotMethods_T
- TSL_LinRotB_T
- TSL_LinRot_T

Create the generic sensor (object) variables using the structures (**mandatory**):

- TSL_Object_T
- TSL_ObjectGroup_T

The TSL_user_Init() function contains the initialization of the STMTouch touch sensing library. Modify this function to take into account the application bank array name and object group names.

The TSL_user_Exec() function contains the main state machine. Modify it also if several object groups must be processed, or the ECS period must be changed, for example.

6.2 Debug with STM Studio

The STM Studio software is very useful to observe variables of the STMTouch touch sensing library. Its powerful features allow a better understanding of how the sensors behave and help to find the better parameters to apply.

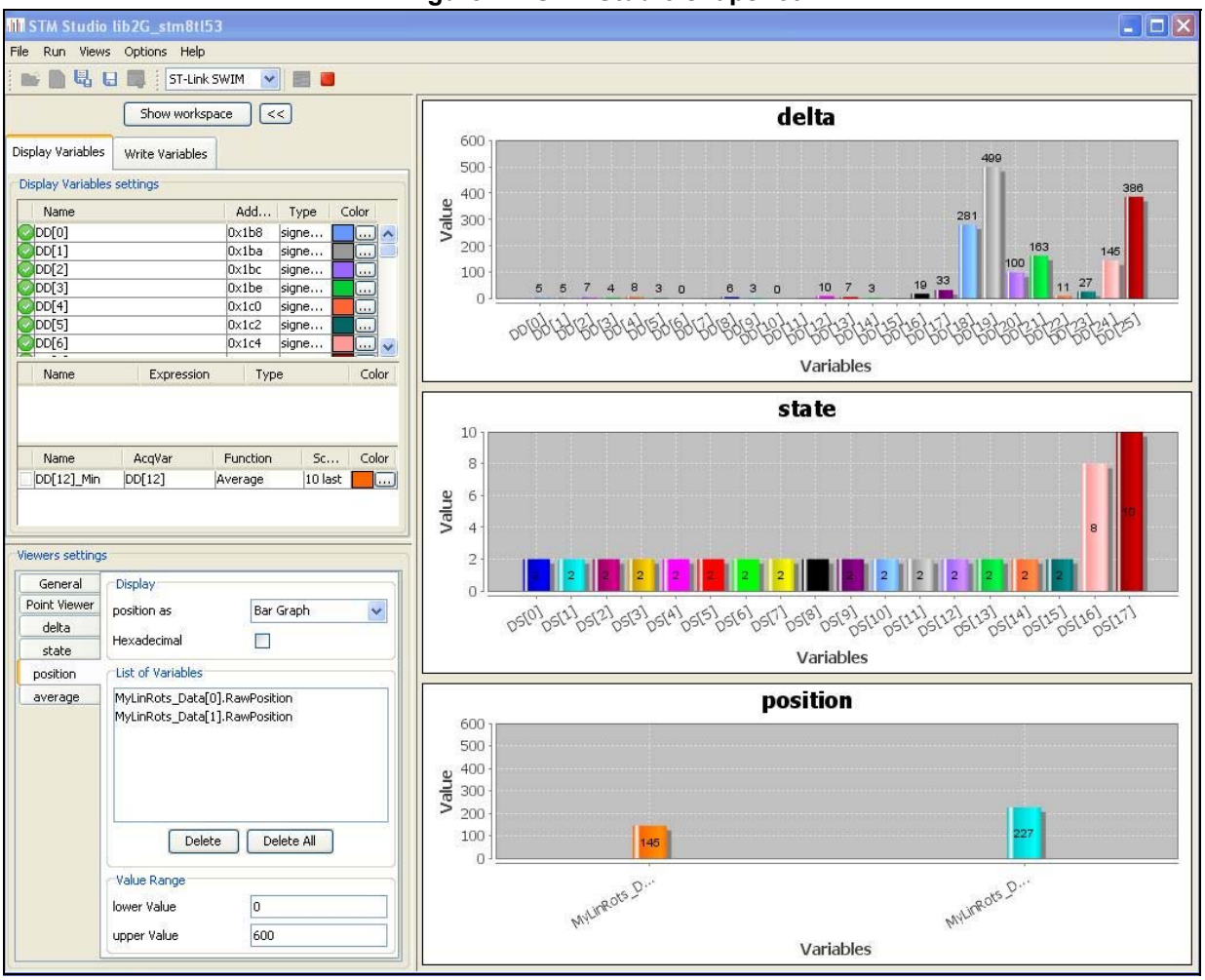
This section does not intend to explain how to use this tool, but gives some advice to better understand and debug the user application.

This is a non-exhaustive list of the STMTouch touch sensing library variables to observe:

- channel measure, reference and delta: These variables are present inside the TSL_ChannelData_T structure. This is useful to adjust the thresholds parameters.
- sensor state present in the TSL_TouchKeyData_T and TSL_LinRotData_T structures. This is useful to adjust the Debounce, ECS and DTO parameters.
- linear and rotary touch sensor position in the TSL_LinRotData_T structure

The following snapshot is an example of data visualization on STM Studio:

Figure 22. STM Studio snapshot



6.3 Debug with STM32CubeMonitor

The STM32CubeMonitor tool can be used to follow the variables of touch sensing libraries.

This scalable tool can be used to go deeper during debug phases.

6.4 Low-power strategy

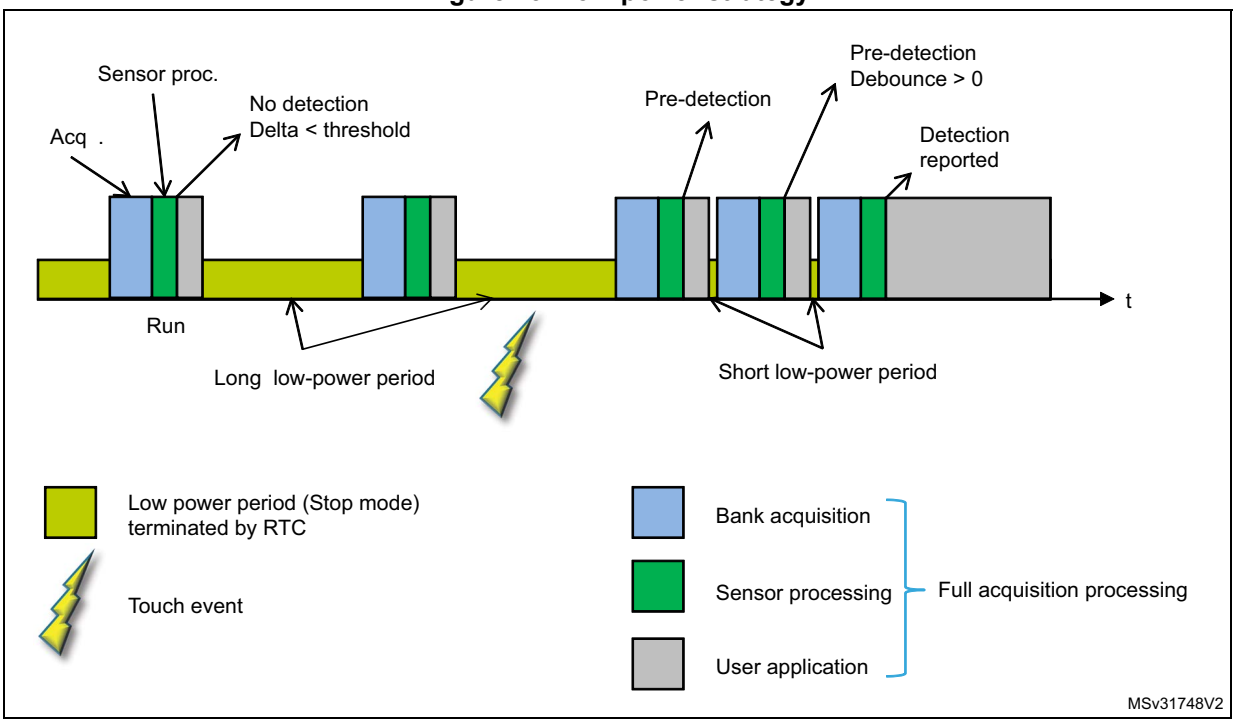
Figure 23 shows the acquisition sequencing for a single-bank acquisition to optimize the power consumption of the device.

To reduce the power consumption, the acquisitions are sequenced with a long delay in between. During this delay, the device can be in low-power mode (Stop mode). This delay can be shortened or even removed between two consecutive acquisitions when the delta

becomes greater than a detection threshold (proximity or touch). The long delay is restored if all the sensors return in RELEASE state.

For an optimum power consumption, the acquisition must be performed with the device in Sleep or Low-power sleep mode, and with the optimum TSC peripheral clock frequency (not too low or too high). The sensor processing must be performed at the highest possible frequency in order to minimize the processing duration. The user application processing must be done at the optimum CPU frequency to offer the best trend between task duration and power consumption.

Figure 23. Low-power strategy



This approach allows power consumption saving without increasing the response time. The maximum response time is obtained when a touch occurs during the sensor processing. It can be expressed as followed:

Max response time = long low-power period + (n) x short low-power period + (n+2) x full acquisition processing - bank acquisition

with n being the debounce value.

6.5 Tips and tricks

6.5.1 Recommendations to increase the noise immunity on the PCB

To ensure a correct operation in noisy environment, the floating nets must be avoided (tracks, copper elements, or conductive frames for example).

As a consequence:

- All unused touch controller I/Os must be either configured to output push-pull low or externally tied to ground.
- The `TSLPRM_TSC_IODEF` parameter must also be configured to the output push-pull low state.
- it is recommended to drive the sampling capacitor common node using a standard I/O of the touch controller configured in output push-pull low mode.
- A capacitor-input filter (pi filter) can also be added on each channel line.

6.5.2 Bank definition

For optimum sensitivity and position reporting, all the channels composing a linear or a rotary touch sensor must be acquired simultaneously. This means that all the channels must belong to the same bank.

Note: The library allows the definition of a linear or a rotary touch sensor with channels belonging to different banks. This configuration induces a loss of sensitivity and a higher noise level. Depending on the acquisition time, it is also possible to observe a position change when removing the finger from the sensor.

6.5.3 Channel assignment

It is recommended to assign GPIOs offering the same sensitivity level to all the channels composing a linear or a rotary touch sensor. It is not recommended to use GPIOs offering a reduced sensitivity.

6.5.4 I/O default-state parameter

For an optimum acquisition noise level, it is recommended to set the `TSLPRM_TSC_IODEF`

or `TSLPRM_IODEF` parameter to output push-pull low.

If the application uses a linear or a rotary touch sensor with channels belonging to different banks, this parameter must be set to input floatin in order to ensure an optimum sensitivity.

7 Revision history

Table 113. Document revision history

Date	Revision	Changes
05-Jan-2016	1	Initial release.
29-Feb-2016	2	Updated *Section 2.4.3: Acquisition and processing layers*. Removed former *Section 2.7: Zone*.
19-May-2016	3	Updated document title. Updated *Section : Introduction*. Updated line TSC_G3_IO4 in *Table 41: Available touch sensing channels for STM32F334x4/x6/x8*.
24-Nov-2016	4	Updated *Section 5.5: Related documents*.
26-Sep-2017	5	Added STM32L083 and STM32L082 lines in *Section 3.7.2*. Added STM32L496 and STM32L4A6 lines in *Section 3.8.2*.
14-Mar-2018	6	Updated *Section 2.9.4: Electrodes placement* including *Figure 8: Electrodes designs*, *Figure 9: Positions 0 and 255*, and *Table 3: Supported linear and rotary touch sensors*.
26-Sep-2018	7	Added *Section 2.18: Sensors acquisition timings* and *Section 2.19: Error management*. Updated: – *Section 2.12: Environment change system (ECS)* – *Section 2.12.4: Usage example* – *Section 5.1.4: The tsl_user file*
23-Oct-2019	8	Updated: – *Introduction* – *Section 2.1.1: Supported microcontrollers* – *Section 2.18.3: Acquisition timing using touchkey, linear and rotary sensors*
05-Oct-2021	9	Updated: – *Introduction* – *Section 1: General information* – *Section 1.2: Related documents* moved – *Section 3.1.2: Development tools* – Note added in *Using up to three touchkey sensors on three groups* – *Section 4.5: Memory footprint* grouping all data – all tables in *Section 4.6* to *Section 4.9: STM32L4 Series microcontrollers* – all tables in *Section 5.6: Available touch sensing channels* Added: – *Section 2: What is STM32Cube?* – *Section 4.10* to *Section 4.13: STM32WB Series microcontrollers* – *Section 6.3: Debug with STM32CubeMonitor*

Table 113. Document revision history (continued)

Date	Revision	Changes
14-Mar-2024	10	Updated the document title and extended the document scope to the STM32U0 series. Updated: – *Introduction* – *Section 2: What is STM32Cube?* – Note added in *Section 3.18.1: Acquisition timing using touchkey sensors* Added: – *Section 4.12: STM32U0 series microcontrollers*
03-Mar-2025	11	Extended the document scope to the STM32U3 series. Updated: – *Introduction* – *Section 2: What is STM32Cube?* Added: – *Section 4.13: STM32U3 series microcontrollers*

IMPORTANT NOTICE – 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 acknowledgment.

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. For additional information about ST trademarks, refer to *www.st.com/trademarks*. 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.

© 2025 STMicroelectronics – All rights reserved