AN2825
Application Note

S-Touch®

STMPE811 resistive touchscreen controller advanced features

Introduction

This application note provides details on the advanced features of the S-Touch® STMPE811 touchscreen controllers.

In particular, the STMPE811 device is a GPIO (general purpose input/output) port expander equipped with an advanced touchscreen controller which helps to reduce the load of the host and simplify data processing.

The document is divided into three sections which respectively describe: touchscreen advanced features, available touchscreen controller settings and basic guidelines on the use of interrupts.

Contents

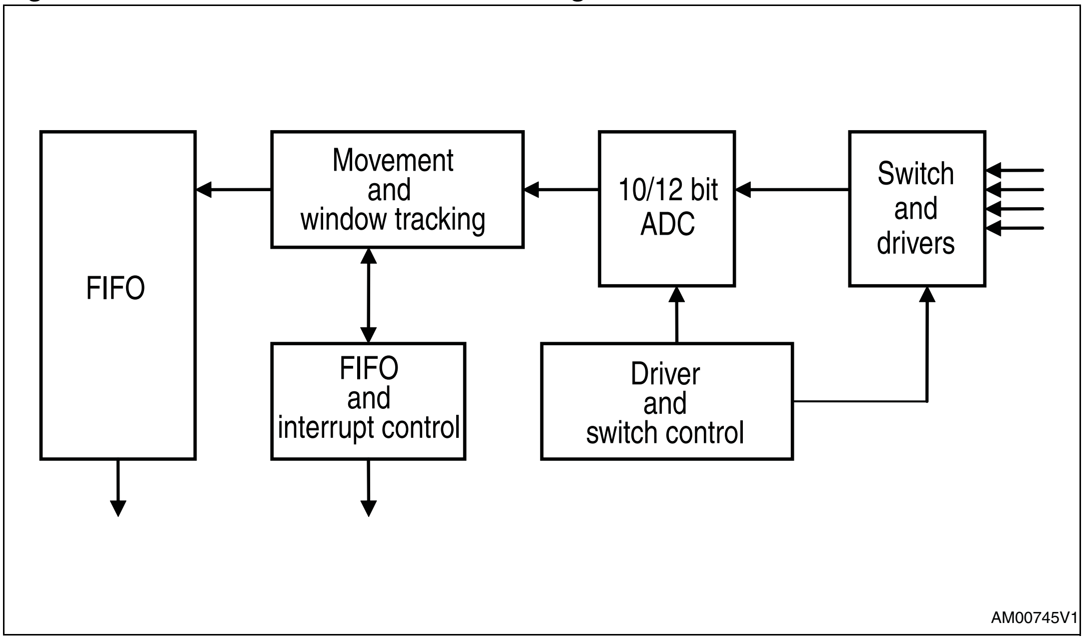
1 STMPE811 advanced features

The touchscreen advanced features such as autonomous operation, movement tracking algorithm to avoid excessive data, a 128-depth FIFO buffer and a programmable active window feature that may be used in applications without increasing the host load.

1.1 Autonomous operation

The STMPE811 device is integrated with a hard-wired touchscreen controller, for a 4-wire resistive type touchscreen, able to operate in an autonomous way, and interrupts the connected CPU only when a pre-defined event occurs.

Figure 1. Touchscreen controller block diagram



The STMPE811 touchscreen controller converts the ADC output into digital location data (coordinates) and stores these in a FIFO for system access. This FIFO can be accessed directly by the system as and when needed. This design reduces the system loading and the I^2C bus utilization.

1.2 FIFO memory

The FIFO memory has a depth of 128 sectors, enough for 128 sets of touch data at maximum resolution (2 x 12 bits). The FIFO can be programmed to generate an interrupt when it is filled to a pre-determined level.

The devices convert the data automatically when a touch is detected. The data is stored in the FIFO memory for the host to read.

The FIFO memory is equipped with a control and status register where the host may refer to it during its operation.

Below is a list of the various status of the FIFO memory reported in the control and status registers (**FIFO__CTRL_STA register, address 0x4B**):

- FIFO overflow:
  reports '1' when FIFO memory is full and new data is coming.
- FIFO full:
  reports '1' when FIFO memory is full.
- FIFO empty:
  reports '1' when there is no data in FIFO memory.
- FIFO threshold trigger:
  reports '1' when the number of data in FIFO memory exceeds the number specified by user in FIFO_SIZE register.

In the same register there is 1 bit for the user to put FIFO memory in reset mode by writing 1 to this bit. The data in the FIFO will be flushed when FIFO is in reset mode.

The host needs to read the data in the FIFO memory consistently. For example, in XYZ mode, a single data consists of 3 bytes where X, Y and Z data is included. Hence, each data read transaction must consist of multiples of 3 bytes in length to avoid confusion and data loss.

Please refer to the STMPE811 datasheet for a detailed description of each register.

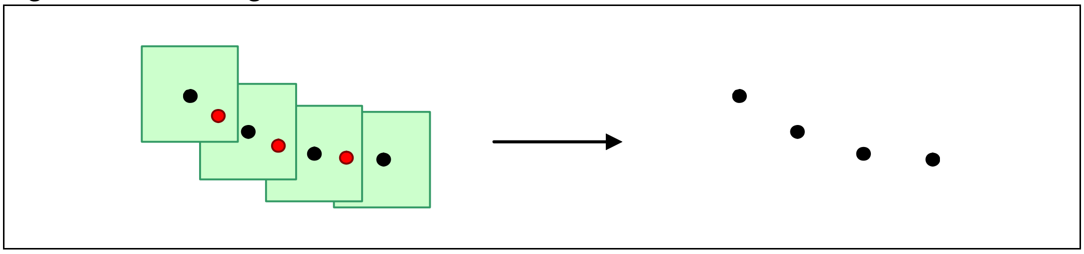
1.3 Movement tracking

The movement tracking feature reduces the number of samples generated by the STMPE811 device. Using this feature, the device will receive only data that satisfies the criteria set by the host.

The system host may select one of the 8 tracking index options: 0, 4, 8, 16, 32, 64, 92 and 127. The value of movement index refers to the native resolution of the ADC, which is 4096 points for full range.

In XY mode, for example, the "127 points" option is selected, valid data form a square that is 127 points in size where the valid data being the center of the square. New data which is inside this box is discarded (not stored in the FIFO memory). If new data is located outside the square, the data is taken and used as the center of new square. This is illustrated in *Figure 2* which shows how the tracking index eliminates the data inside the tracking area which is formed in the surrounding of valid data.

Figure 2. Tracking index



In XYZ mode, there is another factor that affects the tracking index function; the Z value. The tracking index algorithm still considers new data inside the tracking area as valid data if the Z value is lower than the previous valid data (pressure of new data is higher than previous valid data).

1.4 Averaging data to optimize noise cancellation

If the averaging feature is activated, the STMPE811 device samples more than 1 time to produce a single data. The options for averaging are 1, 2, 4 and 8.

For example, if the “4 samples” averaging option is used, the device measures each X-, Y- and Z- data 4 times and calculates the average before storing a data into the FIFO memory.

This feature helps to reduce the host load if the application requires averaging due to random noise captured by the panel.

1.5 Window tracking

Window tracking is another interesting feature which allows the user to define an active area including only the data that are considered to be valid. As a consequence, the data outside the active area are discarded. The host only needs to write in the touchscreen controllers' registers once to use this feature.

It must be ensured that the data written in the STMPE811 window tracking registers is in the form of raw data (ADC data). Hence, the transfer function from raw data to display location data needs to be used.

The transfer function from raw data to display coordinates is shown below:

Equation 1

$$Y_D = 0.5Y_R - 20$$

Where Y_D = Y axis in display coordinates, and Y_R = raw data Y.

For example, if the user wants to set the Top Right Y as 1000 display's Y axis, before writing to the device's register, the data needs to be transformed to an ADC scale using the transfer function:

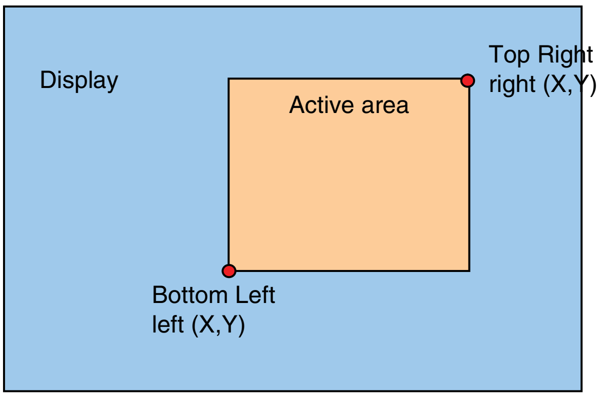
Equation 2

$$Y_R = 2(Y_D + 20)$$

Then the data may be written to the register.

Two coordinates points need to be written to the STMPE811 registers: the top right and the bottom left points. These two points form an active area where the data is considered valid and stored in FIFO memory. The data outside the active area is discarded (*Figure 3*).

Figure 3. Active area definition



2 Settings description

In this section, the various settings available in the STMPE811 touchscreen controller are described.

2.1 ADC sample time

The ADC sample time setting allows you to determine the time required by the ADC to sample a single data in number of clock cycles. The shorter the time, the higher the sampling rate that can be achieved but with less accuracy and vice-versa. The recommended setting is 80 clock cycles as a compromise between speed and accuracy.

2.2 ADC bit mode

The ADC bit mode setting determines the ADC output resolution.

There are two options: 10 and 12 bits. The 10-bit ADC output is enough in most applications using a small screen, such as 3.5” while for bigger screen applications, the 12-bit resolution is recommended to achieve a sufficient touch resolution.

2.3 ADC clock frequency

The recommended setting for the ADC clock frequency is 3.25 MHz to get high a sampling rate and at the same time accurate data. The accuracy of the measurement may be reduced by selecting a higher clock speed.

2.4 TSC operating mode

Five options may be selected determining the type of output of the device’s conversion from the touchscreen panel: XYZ, XY, X, Y and Z. Depending on the application, the user may select different options.

2.5 Tracking index

The tracking index functionality has been explained in *Section 1.5*.

2.6 Averaging

The average setting has been explained in *Section 1.4*.

2.7 Touch detect delay

Touch detect delay is the delay time from the activation of the pull-up resistor in the X+ line to the time the device performs the touch detection. In cases where a filter capacitor is placed in the touchscreen lines, the pull-up resistor needs time to pull the X+ line high. This implies that the user needs to choose enough “touch detect delay” for different values of the filtering capacitor.

The STMPE811 touch detect pull-up resistor is 50 kΩ, hence the minimum touch detect delay can be calculated by using the following formula:

Equation 3

$$t = 1.6094RC + margin = 80470C + margin$$

Where C is the value of the filtering capacitor.

It is good practice to allow around 50% as a margin for safety. If the touch detect delay is not configured correctly, the device may senses false touches and prevent the device from entering auto-sleep mode (more power consumption).

2.8 Driver settling time delay

The same rule described above applies to the conversion/measurement cycle when a touch is detected.

The user needs to set enough settling time in order to get reliable accuracy. If a filtering capacitor is placed, the settling time needs to be set longer. Otherwise, the device starts the measurement when the signal is still in transient mode (not yet settled down), this may lead to inaccuracy of the measurement.

As a general rule, 1-5 nF capacitors require around 500 µs settling time, and 5-10 nF need around 1 ms. The exact value is recommended to be selected during prototype testing since the capacitance of the touchscreen panel may vary.

3 Interrupt user guide

The interrupt sources in the touchscreen controller are:

- Touch detect
- FIFO empty
- FIFO full
- FIFO overflow
- FIFO threshold.

3.1 Touch detect interrupt

A real-time touch detection can be monitored at bit 7 (TSC_STA) of the touchscreen control register (TSC_CTRL). The transition of the status is latched at bit 1 (TOUCH_DET) of the interrupt status register (INT_STA). For example, the transition from 0 to 1 in the TSC_STA-bit of TSC_CTRL register will be captured as '1' in the TOUCH_DET-bit of the INT_STA register. Similarly for transition from 1 to 0. Hence, both pen-down and pen-up events can trigger interrupt signals. The TOUCH_DET-bit needs to be cleared after it is read, otherwise the new status will not be latched.

3.2 FIFO interrupt

Similar to the touch detect, the real-time status of FIFO can be monitored from the FIFO_CTRL_STA register. If any of the status becomes ‘1’, the value is latched to the INT_STA register.

If the status in the interrupt status register is not cleared after it is read, the status is not cleared automatically. This may be misleading for the user when reading the interrupt status register (INT_STA) where FIFO_EMPTY and FIFO_FULL or FIFO_OVERFLOW may occur simultaneously, since they are latched and in some operations it is not cleared.

It is recommended to read only the enabled bit status. For example, FIFO_THRESHOLD and FIFO_OVERFLOW is enabled, when an interrupt is generated, the host needs only to check these two bits, and clear them after doing all the necessary actions. The other bits such as FIFO empty may be in ‘1’ state, but since it is not used, it can be ignored.

4 Revision history

Table 1. Document revision history

Date	Revision	Changes
13-Sep-2011	1	Initial release.