Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!


## Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

# STMPE16M31
# STMPE24M31

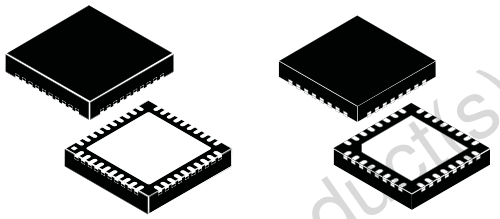
S-Touch® 16/24-channel touchkey controller with PWM and ratio engines

## Features

- Up to 24 capacitive sensor inputs
- Independent and configurable automatic calibration on all channels
- 15 fF resolution, 512 steps with 30 pF auto-tuning
- Up to 30 pF external reference capacitor
- 2 units of 8-input ratiometric engines supporting 256 steps slider/wheel
- PWM and GPIO:
  - Up to 16 general purpose inputs/outputs
  - 8 independent PWM controllers, up to 16 PWM outputs
  - 12 mA sourcing/sinking on GPIO for LED driving (at 3.3 V $V_{IO}$)
  - Maximum source/sink current 120 mA
- Operating voltage:
  - 1.65 - 1.95 V ($V_{CC}$, internally supplied)
  - 2.7- 5.5 V($V_{IO}$)
- Low operating current: 400 µA in active mode, 50 µA in sleep mode and 5 µA in hibernate mode
- $I^2C$ interface (up to 400 kHz). $I^2C$ is 3.3 V tolerant
- 8 kV HBM ESD protection on all sensing pins
- 200V MM ESD protection on all pins

## Applications

- Multimedia bars in notebook computers
- Portable media players and game consoles
- Mobile phones and smartphones



**QFN40 (5 x 5 mm)** **QFN32 (4 x 4 mm)**

## Description

The STMPE16M31 and STMPE24M31 capacitive touchkey controllers offer highly versatile and flexible capacitive sensing capabilities in one single chip.

The devices integrate up to 24 capacitive sensing channels which are highly sensitive and noise tolerant. Two units of hardwired ratiometric engines enable the implementation of a slider/wheel without external computations. Eight independent PWM controllers allow to control up to 16 LEDs with brightness control, ramping and blinking capabilities. The $I^2C$ interface supports up to 400 kHz communication with the system host. A very wide dynamic range allows most applications to work without hardware tuning.

A single STMPE24M31 device can be used to implement a complete notebook multimedia control bar with eight capacitive touchkeys, an 8-channel slider with 256 steps resolution and eight independently controlled LED.

**Table 1. Device summary**

| Order code | Package | Packaging |
|---|---|---|
| STMPE24M31QTR | QFN40 (5 x 5 mm) | Tape and reel |
| STMPE16M31QTR | QFN32 (4 x 4 mm) | Tape and reel |

# Contents

# 1 Pin assignment

**Figure 1. STMPE24M31 pin out**

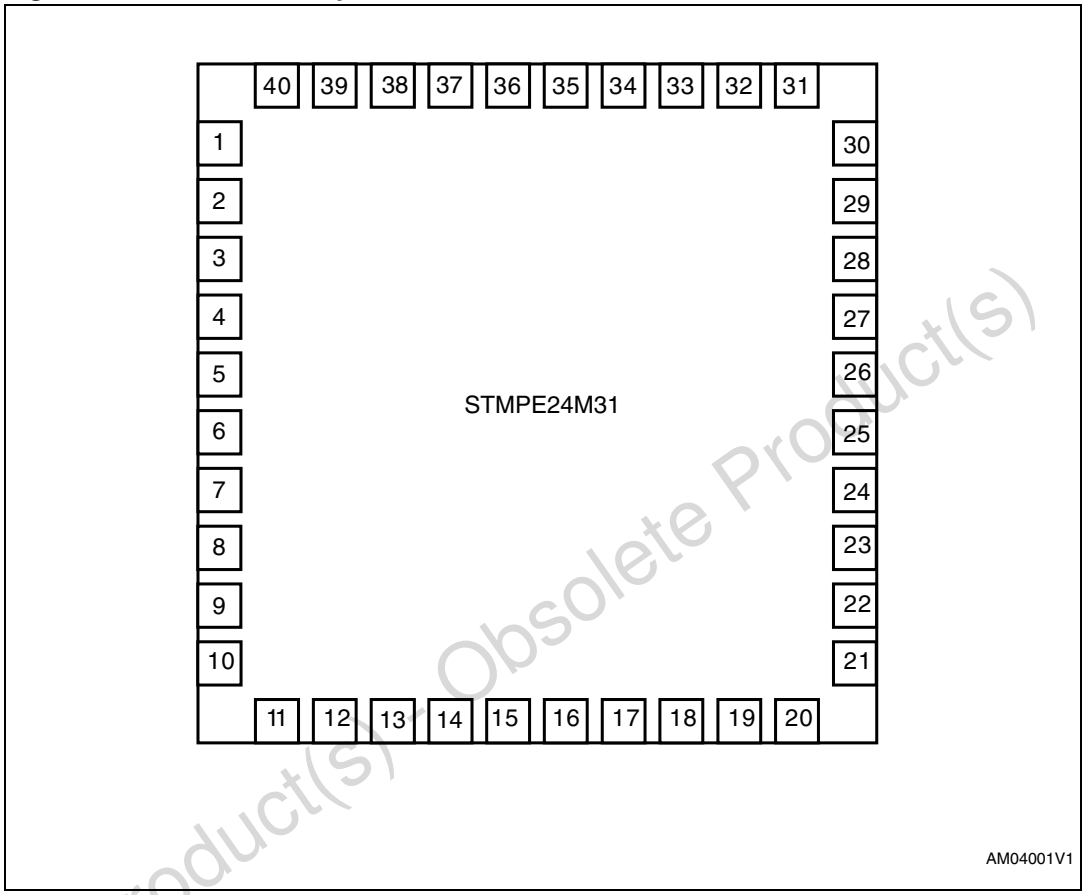


**Figure 2. STMPE16M31 pin out**

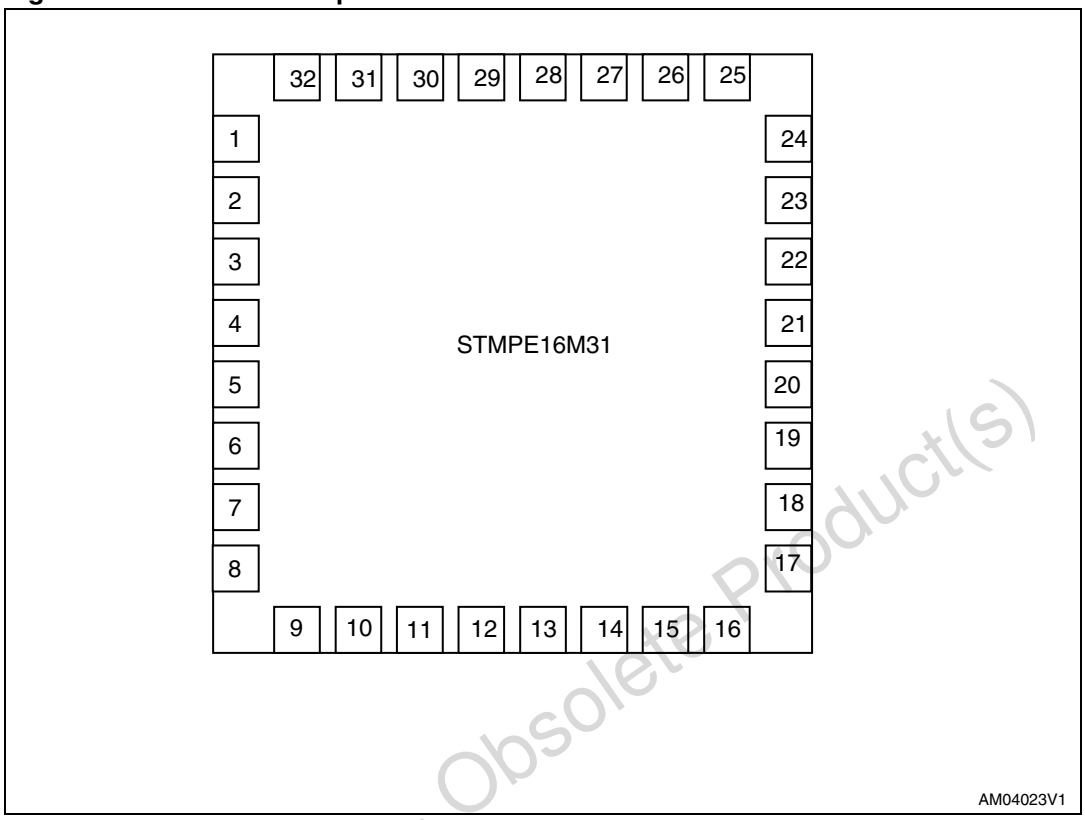



**Table 2. STMPE16M31 and STMPE24M31 pin description**

| STMPE24M31 pin number | STMPE16M31 pin number | Pin name | Voltage domain | Description |
|---|---|---|---|---|
| 1 | 1 | GPIO-0 | VIO | GPIO / capacitive sense |
| 2 | 2 | GPIO-1 | VIO | GPIO / capacitive sense |
| 3 | 3 | GPIO-2 | VIO | GPIO / capacitive sense |
| 4 | 4 | GND | - | Ground |
| 5 | 5 | VIO | - | I/O supply |
| 6 | - | CAP-16 | VCC | Capacitive sense |
| 7 | - | CAP-17 | VCC | Capacitive sense |
| 8 | 6 | GPIO-3 | VIO | GPIO / capacitive sense |
| 9 | 7 | GPIO-4 | VIO | GPIO / capacitive sense |
| 10 | 8 | GPIO-5 | VIO | GPIO / capacitive sense |
| 11 | 9 | GPIO-6 | VIO | GPIO / capacitive sense |
| 12 | 10 | GPIO-7 | VIO | GPIO / capacitive sense |
| 13 | 11 | GND | - | Ground |
| 14 | 12 | VIO | - | I/O supply |

Table 2. STMPE16M31 and STMPE24M31 pin description (continued)

| STMPE24M31 Pin number | STMPE16M31 Pin number | Pin name | Voltage domain | Description |
|---|---|---|---|---|
| 15 | - | CAP-18 | VCC | Capacitive sense |
| 16 | - | CAP-19 | VCC | Capacitive sense |
| 17 | 13 | VCC | - | |
| 18 | 14 | INT | VCC | Open drain interrupt output. This pin should be pulled to VCC or GND, depending on polarity of interrupt used. This pin must not be left floating. |
| 19 | 15 | Address 0 | VCC | $I^2C$ address 0 |
| 20 | 16 | SCL | VCC | $I^2C$ clock |
| 21 | 17 | SDA | VCC | $I^2C$ data |
| 22 | 18 | RESET_N | VCC | Active low reset signal |
| 23 | 19 | Address 1 | VCC | $I^2C$ address 1 |
| 24 | 20 | CRef | VCC | Reference capacitor |
| 25 | - | CAP-20 | VCC | Capacitive sense (minimum 10 pF capacitor is recommended) |
| 26 | - | CAP-21 | VCC | Capacitive sense |
| 27 | 21 | GND | VCC | Ground |
| 28 | 22 | GPIO-8 | VIO | GPIO / capacitive sense |
| 29 | 23 | GPIO-9 | VIO | GPIO / capacitive sense |
| 30 | 24 | VIO | - | I/O supply |
| 31 | 25 | GPIO-10 | VIO | GPIO / capacitive sense |
| 32 | 26 | GPIO-11 | VIO | GPIO / capacitive sense |
| 33 | 27 | GPIO-12 | VIO | GPIO / capacitive sense |
| 34 | 28 | GPIO-13 | VIO | GPIO / capacitive sense |
| 35 | 29 | VIO | - | I/O supply |
| 36 | 30 | GND | - | I/O voltage supply |
| 37 | - | CAP-22 | VCC | Capacitive sense |
| 38 | - | CAP-23 | VCC | Capacitive sense |
| 39 | 31 | GPIO-14 | VIO | GPIO / capacitive sense |
| 40 | 32 | GPIO-15 | VIO | GPIO / capacitive sense |

**Figure 3. STMPE16M31 and STMPE24M31 block diagram**

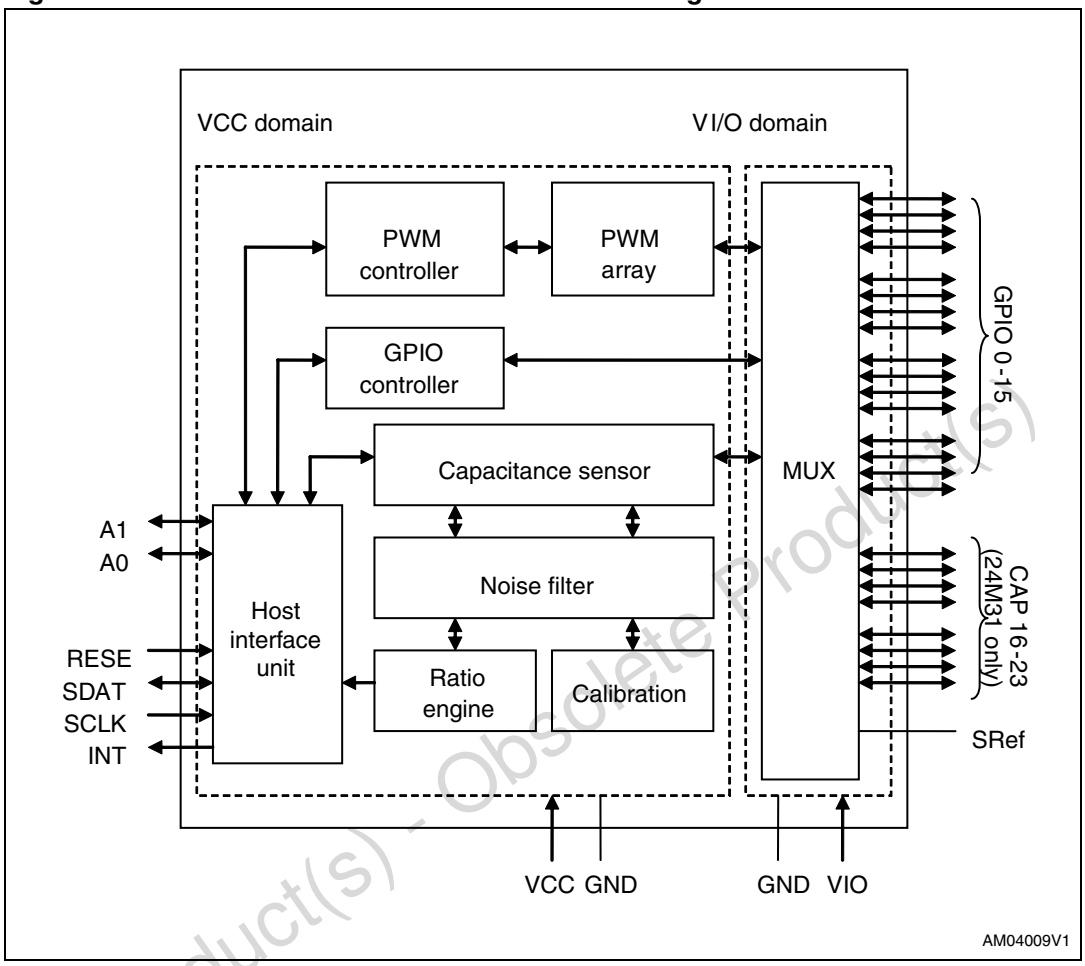


**Figure 4. Sample application - notebook multimedia bar**

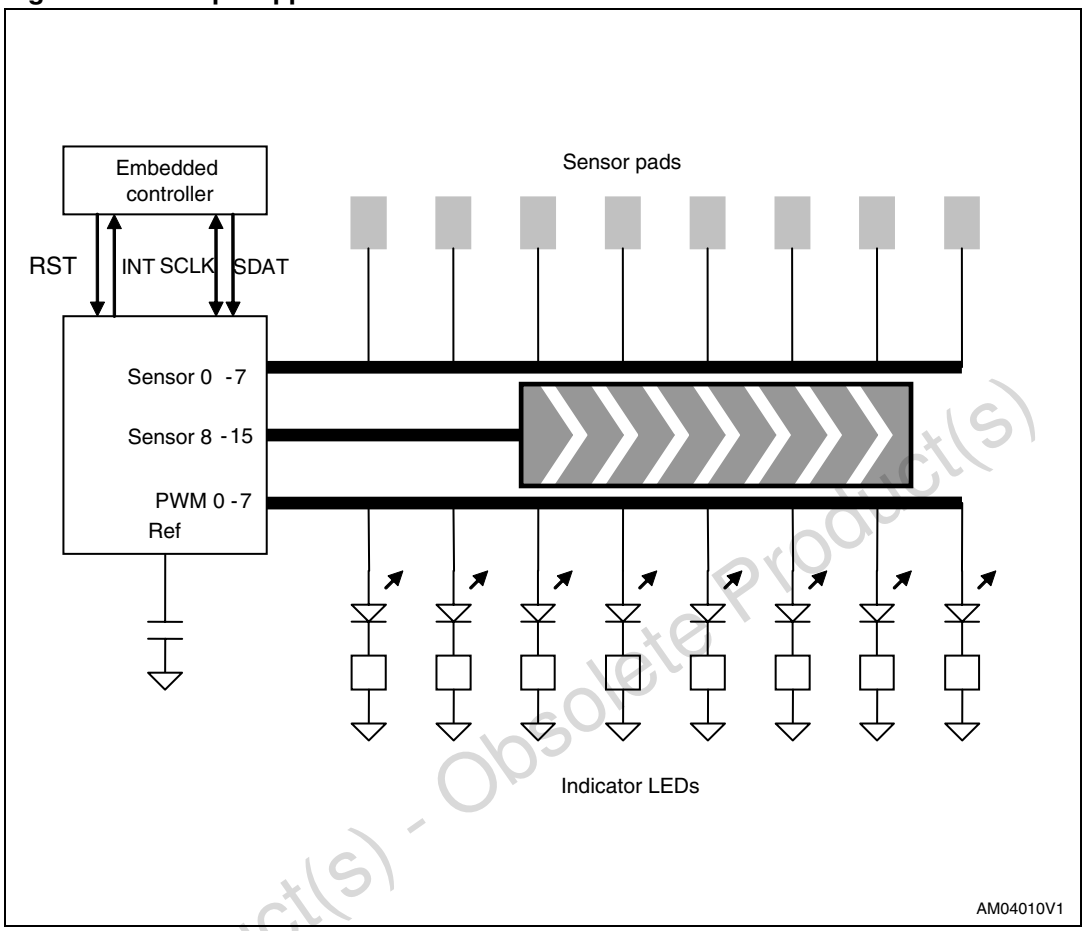



**Table 3. Limitations on intrinsic capacitance on PCB / flexi PCB(1)**

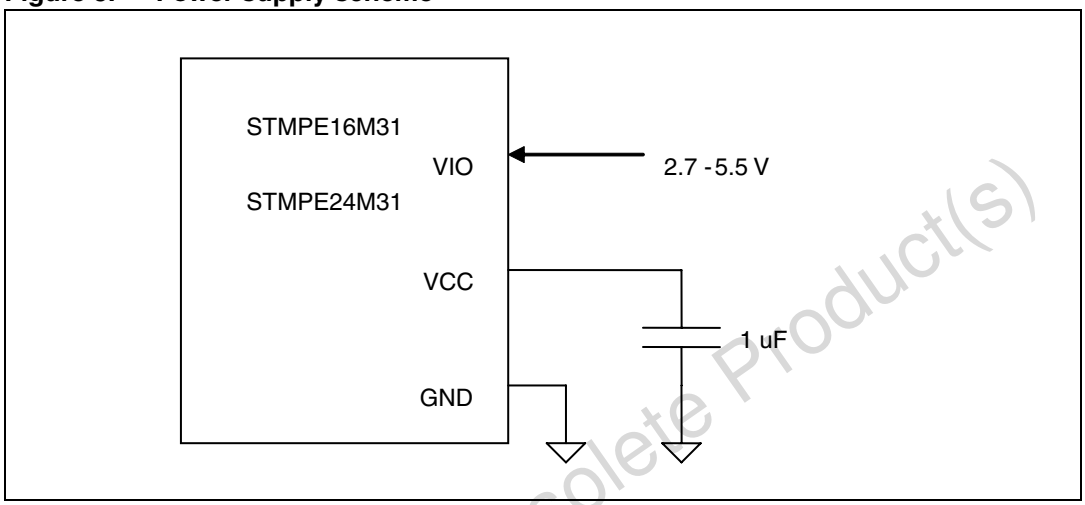
| Cmax-Cmin (Difference between highest and lowest channel capacitance) | Cmax | Matching capacitors |
|---|---|---|
| < 30 pF | < 30 pF | Not required |
| <3 0 pF | > 30 pF, < 60 pF | Cref of up to 30 pF required |
| > 30 pF, < 60 pF | > 30 pF, < 60 pF | Cref of up to 30 pF required<br>Channel matching capacitance of up to 25 pF required |
| > 60 pF | > 60 pF | PCB optimization required |

1. For small PCBs, it is possible to operate the device with CRef left unconnected. However, without a small capacitance at this pin, the capacitive sensing operation tends to be noisier. It is recommended that a capacitor of 10 pF to be connected to this pin.

## 1.1 Power scheme

The STMPE24M31/16M31 is powered by a 2.7- 5.5 V supply. An internal voltage regulator regulates this supply into 1.8 V for core operation. It is recommended to connect a 1 µF capacitor at $V_{CC}$ pin for filtering purpose. The $V_{IO}$ powers all GPIOs directly, if any LED driving is required on the GPIO, the $V_{IO}$ should be at least 3.3 V.

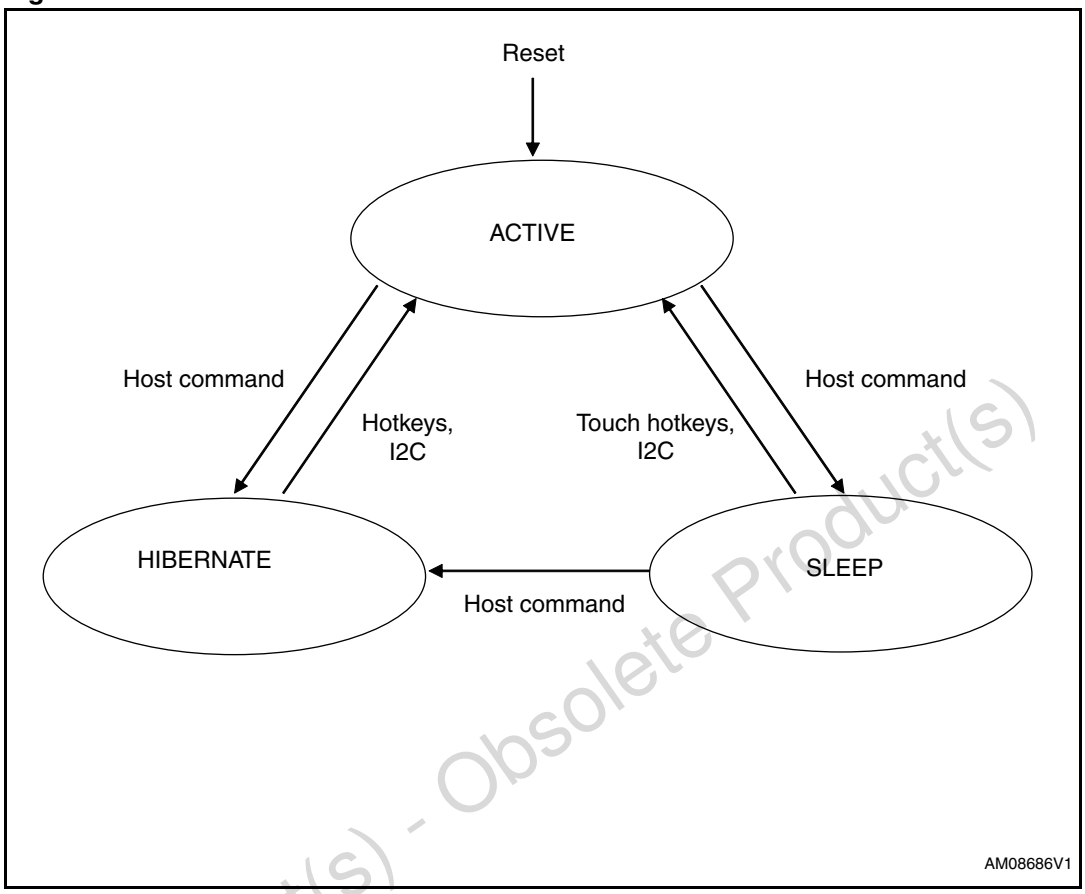
**Figure 5. Power supply scheme**




## 1.2 Power states

The STMPE24M31/16M31 operate in 3 states. *Table 4* illustrates the capability of the device in each of the power states.

**Figure 6. Power states**




1. STMPExxM31 remains in active mode when PWM is running.

**Table 4. Functions available in each power state**

| | Hibernate | Sleep | Active |
|---|---|---|---|
| $I^2C$ | Yes | Yes | Yes |
| GPIO hotkey | Yes | Yes | Yes |
| PWM | No | Yes | Yes |
| Capacitive sensing | No | Slow | Yes |
| Ratio engine | No | No(1) | Yes |

1. When the ratio engine is enabled, device transitions to active state whenever a touch on the slider/wheel is detected, even if it was previously in sleep mode.

# 2 I²C interface module

The STMPE24M31/16M31 has 2 physical I²C address pins, allowing 4 different I²C address settings.

**Table 5. I²C address pins**

| Address 1 | Address 0 | I²C address |
|---|---|---|
| 0 | 0 | 0x58 |
| 0 | 1 | 0x59 |
| 1 | 0 | 0x5A |
| 1 | 1 | 0x5B |

The features that are supported by the I²C interface module are the following ones:

- I²C slave device
- Operates at $V_{CC}$ (tolerant to 3.3 V signaling)
- Compliant to Philips I²C specification version 2.1
- Supports standard (up to 100 kbps) and fast (up to 400 kbps) modes
- 7-bit and 10-bit device addressing modes
- General call
- Start/restart/stop

The features that are not supported are:

- Hardware general call
- CBUS compatibility
- High-speed (3.4 Mbps) mode



## 2.1 Device operation

### Start condition

A Start condition is identified by a falling edge of SDA while SCL is stable at high state. A Start condition must precede any data/command transfer. The device continuously monitors for a Start condition and does not respond to any transaction unless one is encountered.

### Stop condition

A Stop condition is identified by a rising edge of SDA while SCL is stable at high state. A Stop condition terminates the communication between the slave device and bus master. A read command that is followed by NoAck can be followed by a Stop condition to force the slave device into idle mode. When the slave device is in idle mode, it is ready to receive the next I²C transaction. A Stop condition at the end of a write command stops the write operation to registers.

### Acknowledge bit (ACK)

The acknowledge bit is used to indicate a successful byte transfer. The bus transmitter releases the SDA after sending eight bits of data. During the ninth bit, the receiver pulls the SDA low to acknowledge the receipt of the eight bits of data. The receiver may leave the SDA in high state if it would to *not* acknowledge the receipt of the data.

### Data input

The device samples the data input on SDA on the rising edge of the SCL. The SDA signal must be stable during the rising edge of SCL and the SDA signal must change only when SCL is driven low.

### Memory addressing

For the bus master to communicate to the slave device, the bus master must initiate a Start condition and be followed by the slave device address. Accompanying the slave device address, there is a Read/$\overline{W}$ bit (R/$\overline{W}$). The bit is set to 1 for Read and 0 for Write operation.

If a match occurs on the slave device address, the corresponding device gives an acknowledgement on the SDA during the 9$^{th}$ bit time. If there is no match, it deselects itself from the bus by not responding to the transaction. The register memory map of the device is 8-bit address width. Therefore, the maximum number of register is 256 registers of 8-bit width.

*Table 6* illustrates the device operating modes that are supported.

**Table 6. Device operation modes**

| Mode | Bytes | Initial sequence |
|---|---|---|
| Read | ≥1 | START, Device Address, R/$\overline{W}$ =0, Base register Address to be read |
| | | ReSTART, Device Address, R/$\overline{W}$ =1, Data Read, STOP |
| | | If no STOP is issued, the Data Read can be continuously preformed. The address is automatically incremented on subsequent data read. |
| Write | ≥1 | START, Device Address, R/$\overline{W}$ =0, Register Address to be written, Data Write, STOP |
| | | If no STOP is issued, the Data Write can be continuously performed. The address is automatically incremented on subsequent write. |

**Figure 7. Read and write modes (random and sequential)**

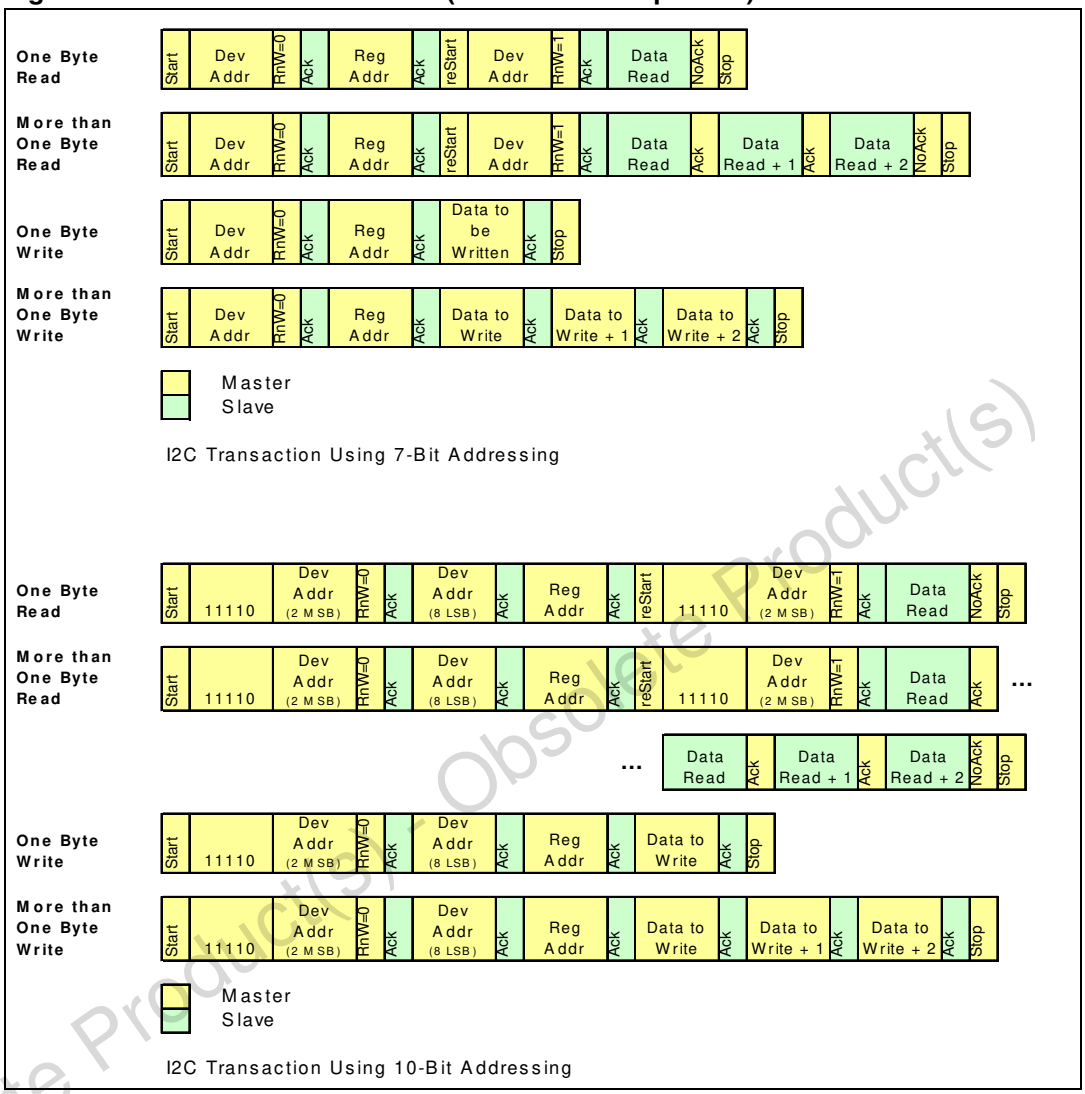
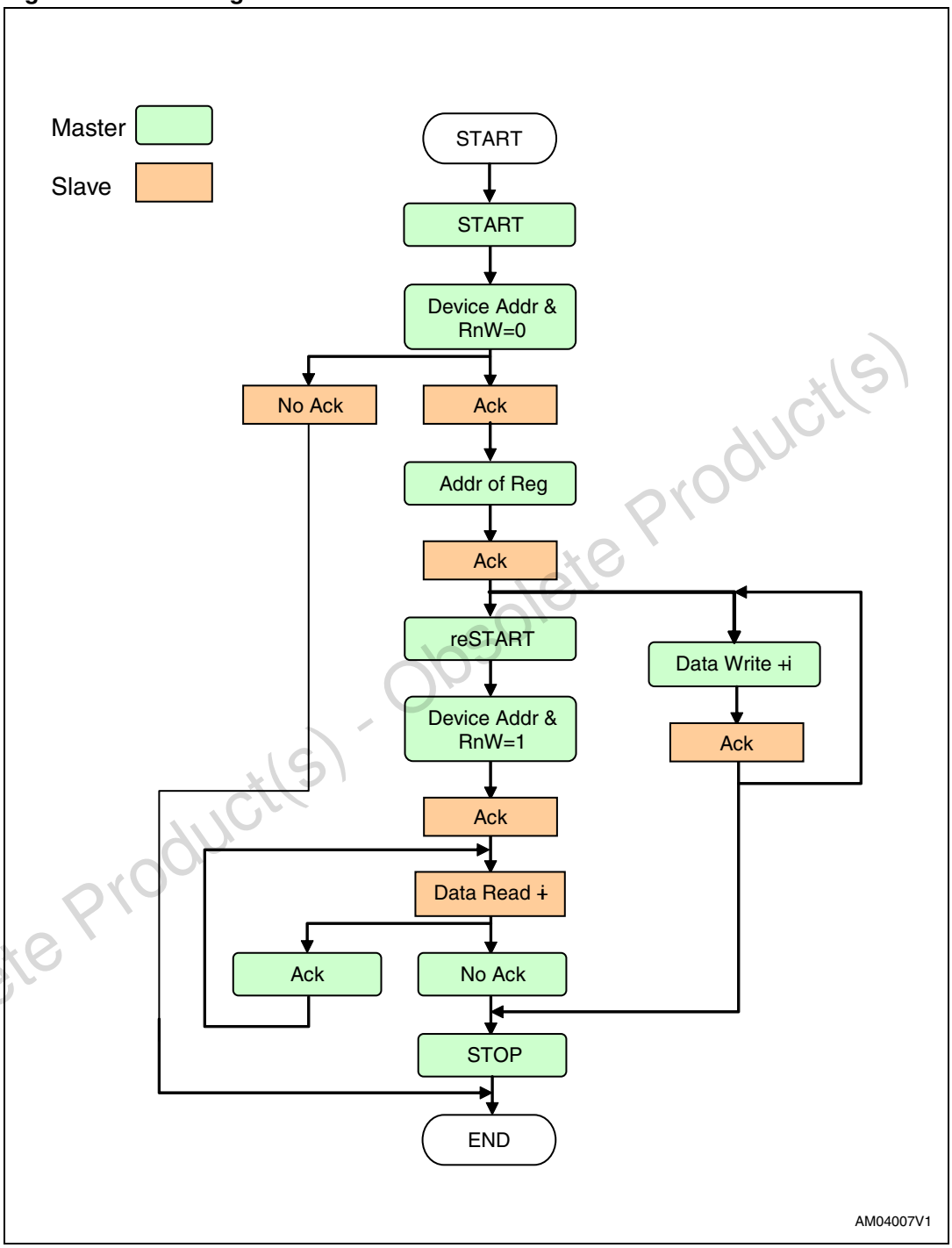


**Figure 8. Flow diagram for read and write modes**



# 3 Read operations

### Read operations for one or more bytes

A write is first performed to load the base register address into the address counter but without sending a Stop condition. Then, the bus master sends a reStart condition and repeats the Device Address with the R/$\overline{W}$ bit set to 1. The slave device acknowledges and outputs the content of the addressed byte. If no more data is to be read, the bus master must not acknowledge the byte and terminates the transfer with a Stop condition.

If the bus master acknowledges the data byte, then it can continue to perform the data reading. To terminate the stream of data byte, the bus master must not acknowledge the last output byte and follow by a Stop condition. The data fetched are from consecutive addresses. After the last memory address, the Address Counter 'rolls-over' and the device continue to output data from the memory address of 0x00.

### Acknowledgement in read operation

For the above read command, the slave device waits, after each byte read, for an acknowledgement during the 9th bit time. If the bus master does not drive the SDA to low state (no acknowledgement by the master), then the slave device terminates and switches back to its idle mode, waiting for the next command.

# 4 Write operations

## 4.1 Write operations for one or more bytes

A write is first performed to load the base register address into the Address Counter without sending a Stop condition. After the bus master receives an acknowledgement from the slave device, it may start to send a data byte to the register (pointed by the Address Counter). The slave device again acknowledges and the bus master terminates the transfer with a Stop condition.

If the bus master would like to continue to write more data, it can just continue write operation without issuing the Stop condition. After the bus master writes the last data byte and the slave device acknowledges the receipt of the last data, the bus master may terminate the write operation by sending a Stop condition. When the Address Counter reaches the last memory address, it 'rolls-over' on the next data byte write.

# 5 General call address

A general call address is a transaction with the slave address of 0x00 and R/$\overline{W}$ = 0. When a general call address is made, the GPIO expander responds to this transaction with an acknowledgement and behaves as a slave-receiver mode. The meaning of a general call address is defined in the second byte sent by the master-transmitter.

**Table 7. Definition of the second byte of the I²C transaction**

| R/$\overline{W}$ | Second byte value | Definition |
|---|---|---|
| 0 | 0x06 | 2-byte transaction in which the second byte tells the slave device to perform a soft reset and write (or latch in) the 2-bit programmable part of the slave address. |
| 0 | 0x04 | 2-byte transaction in which the second byte tells the slave device not to perform a soft reset and write (or latch in) the 2-bit programmable part of the slave address. |
| 0 | 0x00 | Not allowed as second byte. |

*Note: All other second byte values will be ignored.*

*Note: Please allow a gap of approximately 2 µs gap before the next I2C transaction after the General Call of 0x04 or 0x06.*

# 6 Register map and function description

This section lists and describes the registers of the STMPE16M31 and STMPE24M31 devices, starting with a register map and then provides detailed descriptions of register types.

**Table 8. Register map**

| Address | Register name | Reset value | I²C | Register function |
|---|---|---|---|---|
| 0x00 | CHIP_ID | 0x2431 | R | CHIP identification number<br>MSB: 0x24, LSB: 0x31 |
| 0x02 | ID_VER | 0x03 | R | Version of device<br>Engineering samples: 0x01, 0x02<br>Final silicon: 0x03 |
| 0x03 | SYSCON-1 | 0x00 | RW | General system control |
| 0x04 | SYSCON-2 | 0xFE | RW | Sensor and PWM clock divider |
| 0x06 | INT_CTRL | 0x00 | RW | Interrupt control |
| 0x08 | INT_STA | 0x00 | RW | Interrupt status |
| 0x09 | INT_EN | 0x00 | RW | Interrupt enable |
| 0x0A | GPIO_INT_STA | 0x0000 | RW | Interrupt status GPIO |
| 0x0C | GPIO_INT_EN | 0x0000 | RW | Interrupt enable GPIO |
| 0x0E | PWM_INT_STA | 0x00 | RW | Interrupt status PWM |
| 0x0F | PWM_INT_EN | 0x00 | RW | Interrupt enable PWM |
| 0x10 | GPIO_DIR | 0x0000 | RW | GPIO direction setting |
| 0x12 | GPIO_MP_STA | 0x0000 | R | GPIO pin state monitor |
| 0x14 | GPIO_SET_PIN | 0x0000 | RW | GPIO set pin state |
| 0x16 | GPIO_ALT_FUN | 0x0000 | RW | GPIO alternate function |
| 0x20 | GPIO_0_PWM_CFG | 0x00 | RW | Configures PWM output of GPIO-0 |
| 0x21 | GPIO_1_PWM_CFG | 0x00 | RW | Configures PWM output of GPIO-1 |
| 0x22 | GPIO_2_PWM_CFG | 0x00 | RW | Configures PWM output of GPIO-2 |
| 0x23 | GPIO_3_PWM_CFG | 0x00 | RW | Configures PWM output of GPIO-3 |
| 0x24 | GPIO_4_PWM_CFG | 0x00 | RW | Configures PWM output of GPIO-4 |
| 0x25 | GPIO_5_PWM_CFG | 0x00 | RW | Configures PWM output of GPIO-5 |
| 0x26 | GPIO_6_PWM_CFG | 0x00 | RW | Configures PWM output of GPIO-6 |

Table 8. Register map (continued)

| Address | Register name | Reset value | I²C | Register function |
|---|---|---|---|---|
| 0x27 | GPIO_7_PWM_CFG | 0x00 | RW | Configures PWM output of GPIO-7 |
| 0x28 | GPIO_8_PWM_CFG | 0x00 | RW | Configures PWM output of GPIO-8 |
| 0x29 | GPIO_9_PWM_CFG | 0x00 | RW | Configures PWM output of GPIO-9 |
| 0x2A | GPIO_10_PWM_CFG | 0x00 | RW | Configures PWM output of GPIO-10 |
| 0x2B | GPIO_11_PWM_CFG | 0x00 | RW | Configures PWM output of GPIO-11 |
| 0x2C | GPIO_12_PWM_CFG | 0x00 | RW | Configures PWM output of GPIO-12 |
| 0x2D | GPIO_13_PWM_CFG | 0x00 | RW | Configures PWM output of GPIO-13 |
| 0x2E | GPIO_14_PWM_CFG | 0x00 | RW | Configures PWM output of GPIO-14 |
| 0x2F | GPIO_15_PWM_CFG | 0x00 | RW | Configures PWM output of GPIO-15 |
| 0x30 | PWM_MASTER_EN | 0x00 | RW | PWM master enable |
| 0x40 | PWM_0_SET | 0x00 | RW | PWM0 setup |
| 0x41 | PWM_0_CTRL | 0x00 | RW | PWM0 control |
| 0x42 | PWM_0_RAMP_RATE | 0x00 | RW | PWM0 ramp rate |
| 0x43 | PWM_0_TRIG | 0x00 | RW | PWM0 trigger |
| 0x44 | PWM_1_SET | 0x00 | RW | PWM1 setup |
| 0x45 | PWM_1_CTRL | 0x00 | RW | PWM1 control |
| 0x46 | PWM_1_RAMP_RATE | 0x00 | RW | PWM1 ramp rate |
| 0x47 | PWM_1_TRIG | 0x00 | RW | PWM1 trigger |
| 0x48 | PWM_2_SET | 0x00 | RW | PWM2 setup |
| 0x49 | PWM_2_CTRL | 0x00 | RW | PWM2 control |
| 0x4A | PWM_2_RAMP_RATE | 0x00 | RW | PWM2 ramp rate |
| 0x4B | PWM_2_TRIG | 0x00 | RW | PWM2 trigger |
| 0x4C | PWM_3_SET | 0x00 | RW | PWM3 setup |
| 0x4D | PWM_3_CTRL | 0x00 | RW | PWM3 control |
| 0x4E | PWM_3_RAMP_RATE | 0x00 | RW | PWM3 ramp rate |
| 0x4F | PWM_3_TRIG | 0x00 | RW | PWM3 trigger |
| 0x50 | PWM_4_SET | 0x00 | RW | PWM4 setup |
| 0x51 | PWM_4_CTRL | 0x00 | RW | PWM4 control |
| 0x52 | PWM_4_RAMP_RATE | 0x00 | RW | PWM4 ramp rate |

Table 8. Register map (continued)

| Address | Register name | Reset value | I²C | Register function |
|---|---|---|---|---|
| 0x53 | PWM_4_TRIG | 0x00 | RW | PWM4 trigger |
| 0x54 | PWM_5_SET | 0x00 | RW | PWM5 setup |
| 0x55 | PWM_5_CTRL | 0x00 | RW | PWM5 control |
| 0x56 | PWM_5_RAMP_RATE | 0x00 | RW | PWM5 ramp rate |
| 0x57 | PWM_5_TRIG | 0x00 | RW | PWM5 trigger |
| 0x58 | PWM_6_SET | 0x00 | RW | PWM6 setup |
| 0x59 | PWM_6_CTRL | 0x00 | RW | PWM6 control |
| 0x5A | PWM_6_RAMP_RATE | 0x00 | RW | PWM6 ramp rate |
| 0x5B | PWM_6_TRIG | 0x00 | RW | PWM6 trigger |
| 0x5C | PWM_7_SET | 0x00 | RW | PWM7 setup |
| 0x5D | PWM_7_CTRL | 0x00 | RW | PWM7 control |
| 0x5E | PWM_7_RAMP_RATE | 0x00 | RW | PWM7 ramp rate |
| 0x5F | PWM_7_TRIG | 0x00 | RW | PWM7 trigger |
| 0x70 | CAP_SEN_CTRL | 0x00 | RW | Capacitive sensor control |
| 0x71 | RATIO_ENG_REPT_CTRL | 0x00 | RW | Ratio engine report control (only available in final silicon) |
| 0x72 | CH_SEL | 0x00000000 | RW | Selects active capacitive channels |
| 0x76 | CAL_INT | 0x00 | RW | 10ms – 64S calibration interval |
| 0x77 | CAL_MOD | 0x00 | RW | Selects calibration model |
| 0x78 | MAF_SET | 0x00 | RW | Control of median averaging filter |
| 0x7C | DATA_TYPE | 0x00 | RW | Selects type of data available in channel data ports.<br>0x01: TVR<br>0x02: EVR<br>0x03: Channel delay<br>0x04: Impedance (13-bit)<br>0x05:Calibrated Impedance (13-bit)<br>0x06:Locked impedance (13-bit) |
| 0x80 | RATIO_ENG_SET | 0x00 | RW | General setup of ratio engine |
| 0x81 | RATIO_ENG_1_CFG | 0x00 | RW | Configuration of ratio engine 1 |
| 0x82 | RATIO_ENG_2_CFG | 0x00 | RW | Configuration of ratio engine 2 |
| 0x83 | RATIO_ENG_STA | 0x00 | R | Status of ratio engine |
| 0x84 | RATIO_ENG_1_DATA | 0x000000 | R | Output data of ratio engine 1 |
| 0x87 | RATIO_ENG_2_DATA | 0x000000 | R | Output data of ratio engine 2 |
| 0x90 | KEY_FILT_CTRL | 0x00 | | General key filter control |

Table 8. Register map (continued)

| Address | Register name | Reset value | I²C | Register function |
|---|---|---|---|---|
| 0x92 | KEY_FILT_GROUP-1 | 0x00000000 | | Define channels included in key filter group 1 |
| 0x96 | KEY_FILT_GROUP-2 | 0x00000000 | | Define channels included in key filter group 2 |
| 0x9A | KEY_FILT_DATA | 0x00000000 | | Filtered touchkey data |
| 0xB4 | TOUCH_DET | 0x00000000 | R | Touch detection register (real time) |
| 0xC0 | CH_DATA-0 | 0x0000 | | Channel data according to data type setting |
| 0xC2 | CH_DATA-1 | 0x0000 | | |
| 0xC4 | CH_DATA-2 | 0x0000 | | |
| 0xC6 | CH_DATA-3 | 0x0000 | | |
| 0xC8 | CH_DATA-4 | 0x0000 | | |
| 0xCA | CH_DATA-5 | 0x0000 | | |
| 0xCC | CH_DATA-6 | 0x0000 | | |
| 0xCE | CH_DATA-7 | 0x0000 | | |
| 0xD0 | CHDATA-8 | 0x0000 | | |
| 0xD2 | CH_DATA-9 | 0x0000 | | |
| 0xD4 | CH_DATA-10 | 0x0000 | | |
| 0xD6 | CH_DATA-11 | 0x0000 | | |
| 0xD8 | CH_DATA-12 | 0x0000 | | |
| 0xDA | CH_DATA-13 | 0x0000 | | |
| 0xDC | CH_DATA-14 | 0x0000 | | |
| 0xDE | CH_DATA-15 | 0x0000 | | |
| 0xE0 | CH_DATA-16 | 0x0000 | | |
| 0xE2 | CH_DATA-17 | 0x0000 | | |
| 0xE4 | CH_DATA-18 | 0x0000 | | |
| 0xE6 | CH_DATA-19 | 0x0000 | | |
| 0xE8 | CH_DATA-20 | 0x0000 | | |
| 0xEA | CH_DATA-21 | 0x0000 | | |
| 0xEC | CH_DATA-22 | 0x0000 | | |
| 0xEE | CH_DATA-23 | 0x0000 | | |

# 7 System controller

The system controller contains the registers that control the following functions:

- Device identification
- Version identification
- Power state management
- Clock speed management
- Clock gating to various modules

**Table 9. System controller registers**

| Address | Register name | Reset value | R/W | Description |
|---|---|---|---|---|
| 0x00 | CHIP_ID | 0x2431 | R | CHIP identification number MSB: 0x24, LSB: 0x31 |
| 0x02 | ID_VER | 0x03 | R | Version of device |
| 0x03 | SYSCON-1 | 0x00 | RW | General system c control |
| 0x04 | SYSCON-2 | 0xFE | RW | Sensor and PWM clock divider |

## SYSCON-1 General system control

**Address:** 0x03

**Type:** R/W

**Reset:** 0x00

**Description:** The general system control register (SYSCON-1) controls the operation state and clock speed of the device.

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | RESERVED | RESERVED | CLKSPD | SLEEP_EN | Reserved | SOFT_RST | HIBRNT |
| RW | RW | RW | RW | RW | RW | RW | RW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

[7:5] RESERVED: Do not write to these bits. Reads ‘0’. Writing ‘1’ to these bits may result in unpredictable behaviour.

[4] CLKSPD: Selects the macro engine’s speed.
0: 2 MHz
1: RESERVED

[3] SLEEP_EN: Enable or disable the sleep mode. Under all operating conditions, this bit should be set to '0'.
1: Enable the touch sensor’s sleep mode
0: Disable the touch sensor’s sleep mode

[2] RESERVED: Do not write to these bits. Reads ‘0’.

[1] SOFT_RST: Soft reset.
1: To perform soft reset.

[0] HIBRNT: Hibernate.
1: To force the device to hibernate mode.

## SYSCON-2 Sensor and PWM clock divider

**Address:** 0x04

**Type:** R/W

**Reset:** 0xFE

**Description:** Sensor and PWM clock divider. The SYSCON-2 register controls the sensor and PWM clock speed, and the clock gating of various functional modules.

This bit will always read '0'. as the I2C transaction to read this bit will wake up the device from hibernate mode.

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SCLK_DIV | | | PCLK_DIV | | GPIO_CLK | PWM_CLK | CS_CLK |
| RW | | | RW | | RW | | RW |
| 1 | | | 1 | | 1 | | 0 |

[7:5] SCLK_DIV: Sensor clock divider.
000, 001, 010: RESERVED
011: 64
100: 128
101: 256
110: 512
111: 1024
Sensor clock is 2 MHz / ( PRBS_Factor * SCLK_DIV[2:0] )
PRBS factor = 4.5

[4:3] PCLK_DIV: PWM clock divider
00 for 16 kHz
01 for 32 kHz
10 for 64 kHz
11 for 128 kHz

[2] GPIO_CLK: GPIO clock disable
Write “1” to diWrite “1“ to disable the clock to GPIO module.
When clock to GPIO module is disabled, access to GPIO module register will not work correctly.

[1] PMW_CLK: PWM clock disable
Write “1“ to disable the clock to PWM module.
When clock to PWM module is disabled, access to PWM module register will not work correctly.

[0] CS_CLK: Capacitive sensor clock disable
Write “1“ to disable the clock to capactive sensor module
When clock to touch module is disabled, access to touch module registers will not work correclty.

## 7.1 Interrupt system

This module controls the interruption to the host based on the activity of other modules in the system, such as the capacitive sensing, GPIO and PWM modules.

**Figure 9. Interrupt system**

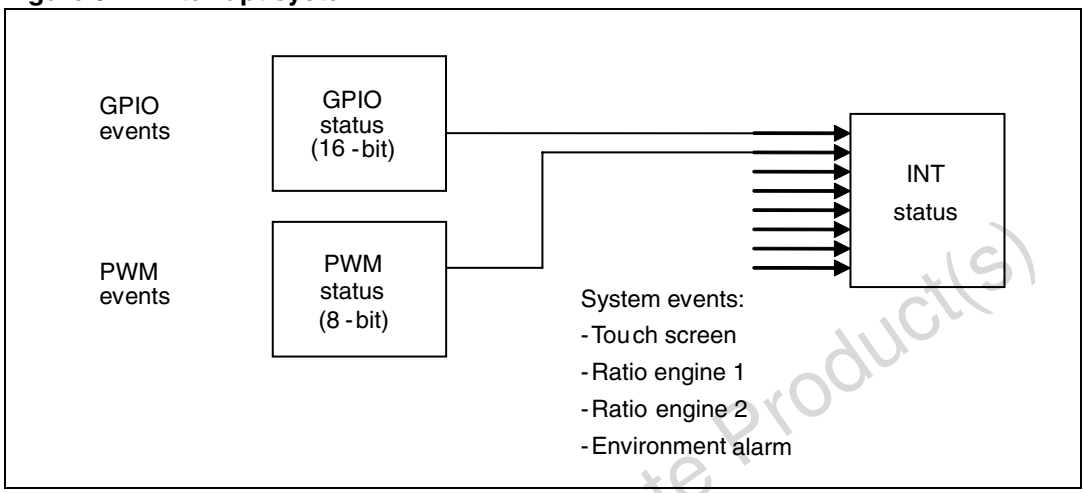



**Table 10. Interrupt system registers**

| Address | Register name | Reset value | R/W | Description |
|---|---|---|---|---|
| 0x06 | INT_CTRL | 0x00 | RW | Interrupt control register |
| 0x08 | INT_STA | 0x00 | RW | Interrupt status register |
| 0x09 | INT_EN | 0x00 | RW | Interrupt enable register |
| 0x0A | GPIO_INT_STA | 0x0000 | RW | Interrupt status GPIO register |
| 0x0C | GPIO_INT_EN | 0x0000 | RW | Interrupt enable GPIO register |
| 0x0E | PWM_INT_STA | 0x00 | RW | Interrupt status PWM register |
| 0x0F | PWM_INT_EN | 0x00 | RW | Interrupt enable PWM register |