AN2466
Application note

STMPE801 - Hardware Interface guide

Introduction

STMPE801 is an eight-bit port expander that can be interfaced to the main digital ASIC or processor via the two-line bidirectional I^2C bus. The digital engines in mobile multimedia platforms usually come with a limited number of I/Os. The port expander ICs can be used to increase the number of I/Os or control signals in such applications.

STMPE801 can be used in advanced digital platforms such as:

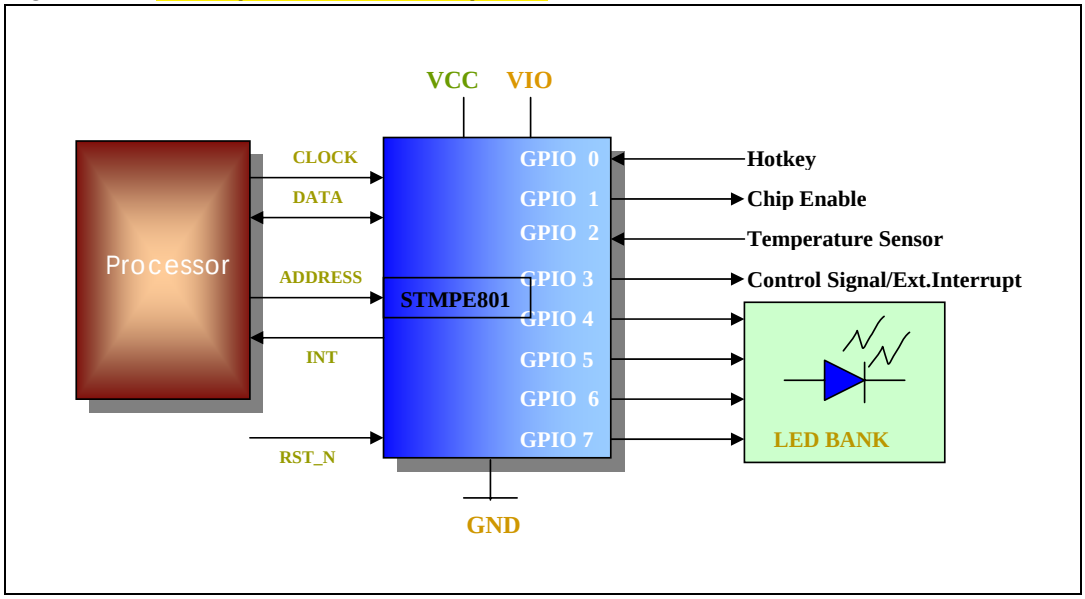
- Portable media players
- Game consoles
- Mobile phones
- Smart phones etc.

This application note explains the setup and hardware interfacing of the device to the main processor.

Contents

1 STMPE801 Device features

Figure 1. Concept of GPIO Port Expander

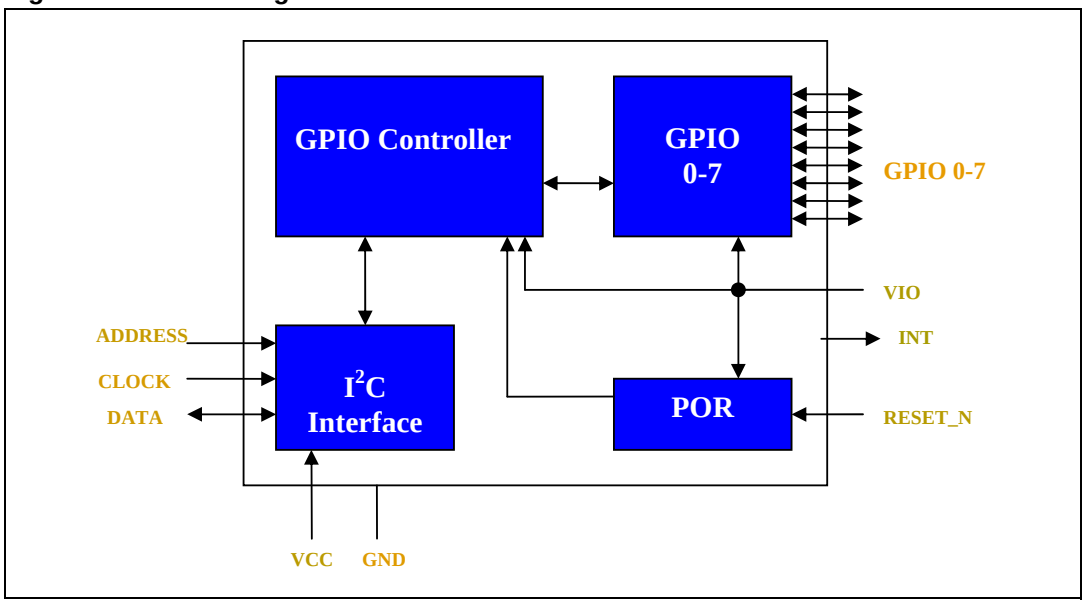


- Low CPU utilization (Interrupts available and so no Polling!)
- Configurable Hotkey Detection on each GPIO
- Flexibility in configuration of each of the eight GPIOs
- Simpler communication with CPU (just two I^2C lines)
- Low power consumption with ultra low standby current (< 1 µA)
- IO and core voltages from 1.65 V up to 3.6 V
- Interrupt output (open drain) pin
- No external clock input required
- Small package QFN16 - 16 pins 1.8 mm x 2.6 mm, making it optimal for use in portable application like mobile phones with critical space constraints.

The STMPE801 offers great flexibility as each I/O can be independently configured as input or output. The eight GPIOs can be connected externally to different modules like LEDs, temperature sensors, chip selects for other devices, or as interrupt inputs from other devices.

This device has been designed with very low quiescent current in standby mode and includes a Hot Key detection for each I/O to optimize the power consumption of the IC.

Figure 2. Block diagram of STMPE801



2 Pin description of STMPE801

The table below gives a list of all the pins on STMPE801.

Table 1. STMPE801 Pin description

Pin	Name	Type	Description
1	INT	O	Interrupt output (open drain)
2	Reset_N	I	External reset input, active LOW
3	CLOCK	A	I^2C Serial clock line
4	ADDRESS	I	Digital Input for I^2C slave address (either High or Low)
5	DATA	A	I^2C Serial data line
6	VCC	-	Supply voltage for I^2C block
7	VIO	-	Supply voltage for GPIO controller (Note: VIO must always be ≥ VCC)
8	GND	-	GND
9	GPIO_0	IO	GPIO
10	GPIO_1	IO	GPIO
11	GPIO_2	IO	GPIO
12	GPIO_3	IO	GPIO
13	GPIO_4	IO	GPIO
14	GPIO_5	IO	GPIO

Table 1. STMPE801 Pin description (continued)

Pin	Name	Type	Description
15	GPIO_6	IO	GPIO
16	GPIO_7	IO	GPIO

2.1 Power Supply - VCC and VIO

The STMPE801 device operates on two different power supplies, VCC and VIO. Both VCC and VIO can function in the range of 1.65 V to 3.6 V. This enables a variety of devices to interface directly to the device without a level translator block.

Proper decoupling capacitors should be used to filter out the Power supply noise. There are some critical points to be noted to ensure proper functioning of the device as listed below:

- VCC pin supplies the I^2C lines and ADDRESS pin.
- All other pins are connected to the VIO supply.
- At any time, to conserve power, the VCC can be turned off if there is no I^2C activity but at no point, should the VIO be turned off while the VCC is ON.
- The VIO should always be ≥VCC

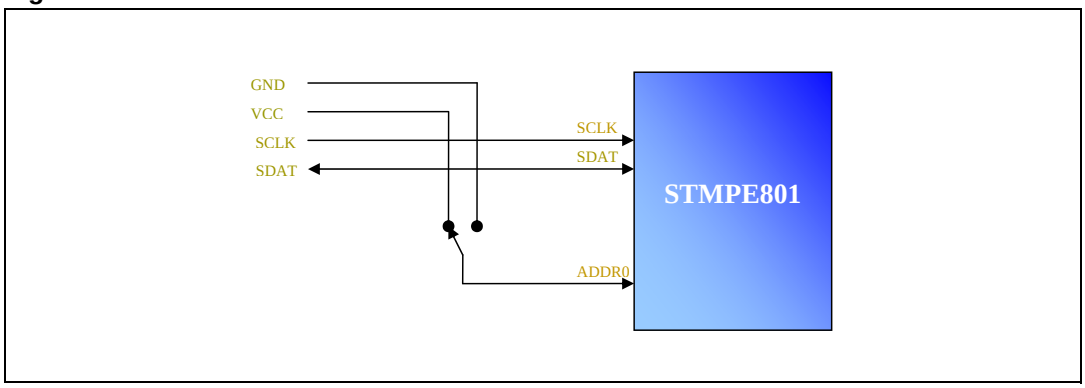
2.2 I^2C Interface

The port expander STMPE801 can be controlled with just the two I^2C lines. All internal registers can be accessed through this I^2C interface. The I^2C interface features are as given below:

- I^2C Slave device
- Operates on the VCC supply (1.65 V to 3.6 V)
- Compliant to Philip I^2C specification version 2.1
- Supports standard (up to 100kbps) and fast (up to 400kbps) modes.
- 7-bit addressing mode supported.
- General call
- Up to two STMPE801 devices can be connected on the same I^2C bus

The slave address is selected by the state of the ADDRESS pin. The address is read every time the I^2C transaction occurs. This implies the slave address of the device can be changed on the fly without any need for latching the address into the device at reset.

Figure 3. STMPE801 Slave address selection

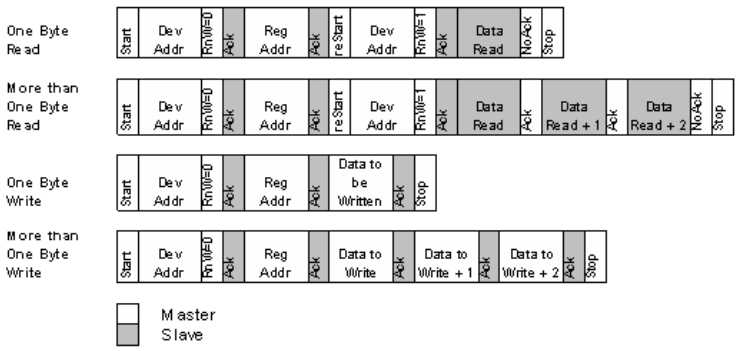


The I²C Read/ $\overline{\text{Write}}$ is done byte by byte. The R/ $\overline{\text{W}}$ bit is added as the LSB to the 7-bit slave address to make up one byte to be sent through the I²C interface from the master.

Table 2. Valid I²C Slave address

ADDRESS pin	7-bit Slave Address	8-bit format to be used (including R/ $\overline{\text{W}}$ bit in LSB)
0	41h (1000001b)	82h
1	44h (1000100b)	88h

Figure 4. I²C Read/write transaction



Once the slave address is configured and responding correctly, the internal registers can be accessed through I²C Interface.

The SCLK and SDATA are open drain pins and should be provided with a pull-up resistor to VCC. The VCC can be selected based on the operating voltage of the I²C host, for example as either 1.8 V or 3.3 V without any need for an intermediate level translator stage.

Figure 5. I²C pin structure

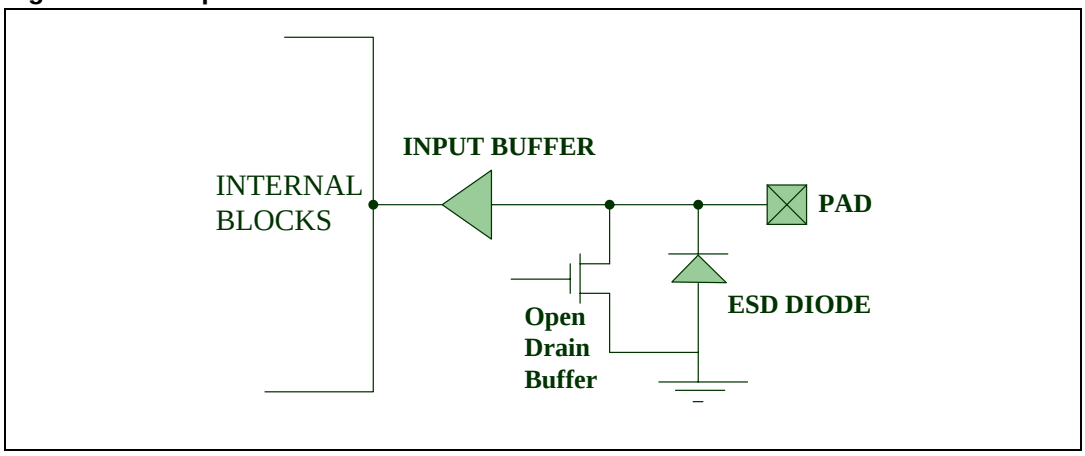
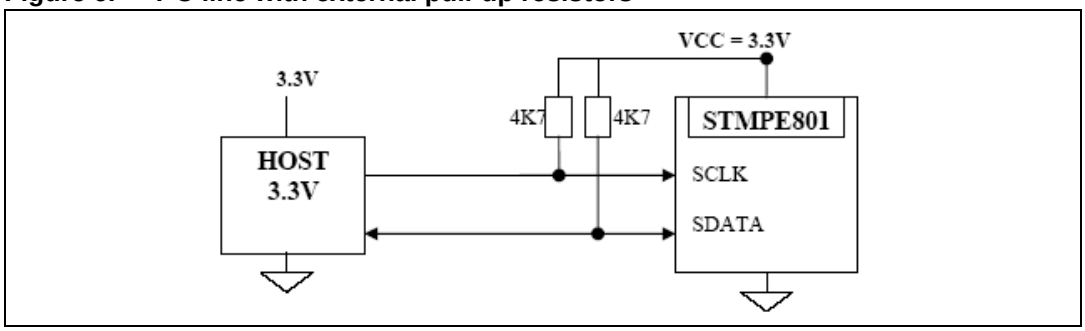


Figure 6. I²C line with external pull-up resistors



2.2.1 I²C General call

STMPE801 supports I²C general call based on the following table. When a general call address of 0x00 with R/ $\overline{W}$ = 0 is sent, the device responds with an acknowledgement and performs the instruction given by the second byte. The device does not give ACK for any second byte other than the valid second bytes listed in *Table 3.*

Table 3. I²C General call

Second byte	Definition
0x06	Slave device should reset and latch in the slave address
0x04	Slave device should latch in the slave address but without reset.
0x0	Not allowed as second byte.

2.3 Reset pin (RESET_N)

The Reset pin is an active low input. It should be tied HIGH to VIO supply in order to start normal operation of the device. Apart from the RESET_N pin, the device can also be reset through the "Soft_Reset" bit in the SYSCON register. Writing a '1' to Bit 7 resets the device and after reset, the bit is cleared to '0' by the Hardware.

2.4 Interrupt pin (INT)

The interrupt pin is an open drain output pin with a structure similar to the SCL,SDA pins. The INT output should be provided with a suitable pull-up resistor to VIO supply. The interrupt output polarity can be configured as active low or active high by setting Bit 0 of the SYSCON register. The interrupt output should be enabled by writing a '1' to Bit 2 in the SYSCON. Even if individual GPIO interrupts are enabled, no interrupt will be generated if the global INT_Enable bit is not set in the SYSCON. The Interrupt is cleared by reading the ISGPIOR register (0x09). At reset, the interrupt output is disabled and the polarity is active LOW.

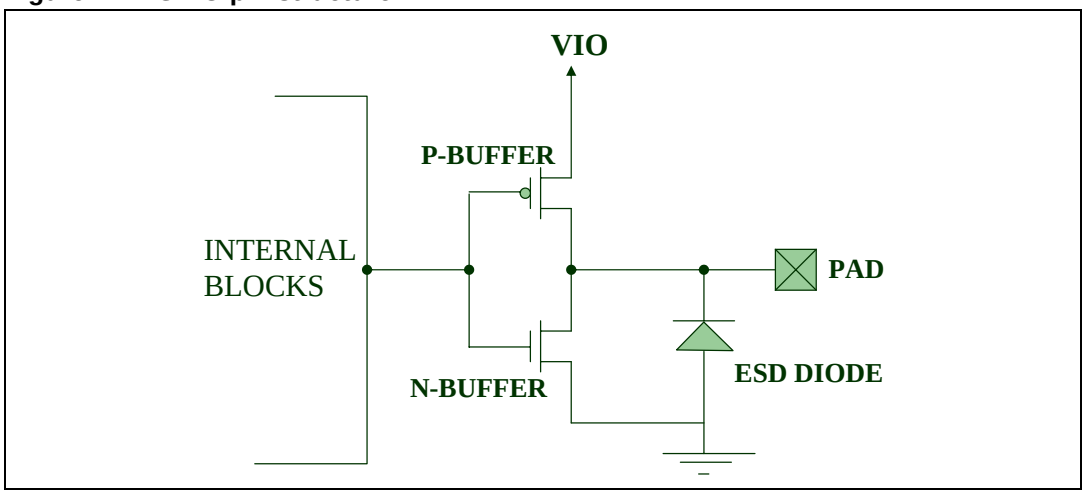
Table 4. SYSCON register

Bit	Reset value	Name	Description
7	0	SoftReset	Writing ‘1’ to this bit causes a soft reset of the device
6	0	I^2C_SHDN	Writing ‘1’ to this bit shuts down the I^2C block on the next valid I^2C clock.
5	0		
4	0		
3	0		
2	0	INT_Enable	‘1’ to enable, ‘0’ to disable INT output
1	0		
0	0	INTPolarity	‘1’ for active HI, ‘0’ for active LOW

2.5 GPIO Pins

All 8 GPIO lines are configured as inputs at power-on/reset and are independent of each other and can be individually programmed as input or output. Unused GPIOs should be configured as outputs to minimize power consumption.

Figure 7. GPIO pin structure



2.5.1 Configuring GPIO registers

The GPIO pin direction can be selected as input or output by writing into the GPDR register (0x12).

Table 5. GPIO Direction register

	GPDR							
Bit	7	6	5	4	3	2	1	0
	IO_7	IO_6	IO_5	IO_4	IO_3	IO_2	IO_1	IO_0
Reset value	0	0	0	0	0	0	0	0
Bits	**Name**	**Description**						
7:0	GPIO[x]	'0': The corresponding GPIO pin is set to Input. '1': The corresponding GPIO pin is set to Output.						

The GPIO output state can be set high or low by writing into the corresponding bit in the GPSR register (0x11).

Table 6. GPIO State register

	GPSR							
Bit	7	6	5	4	3	2	1	0
	IO_7	IO_6	IO_5	IO_4	IO_3	IO_2	IO_1	IO_0
Reset value	0	0	0	0	0	0	0	0
Bits	**Name**	**Description**						
7:0	GPIO[x]	'0': The corresponding GPIO output is set LOW '1': The corresponding GPIO output is set HIGH						

If the GPIO is configured as an input, the pin state is monitored by reading the corresponding bit in the GPMR register (0x10).

Table 7. GPIO Monitor register

	GPMR							
Bit	7	6	5	4	3	2	1	0
	IO_7	IO_6	IO_5	IO_4	IO_3	IO_2	IO_1	IO_0
Reset value	0	0	0	0	0	0	0	0
Bits	**Name**	**Description**						
7:0	GPIO[x]	Reading '0': The corresponding GPIO input state is LOW Reading '1': The corresponding GPIO input state is HIGH						

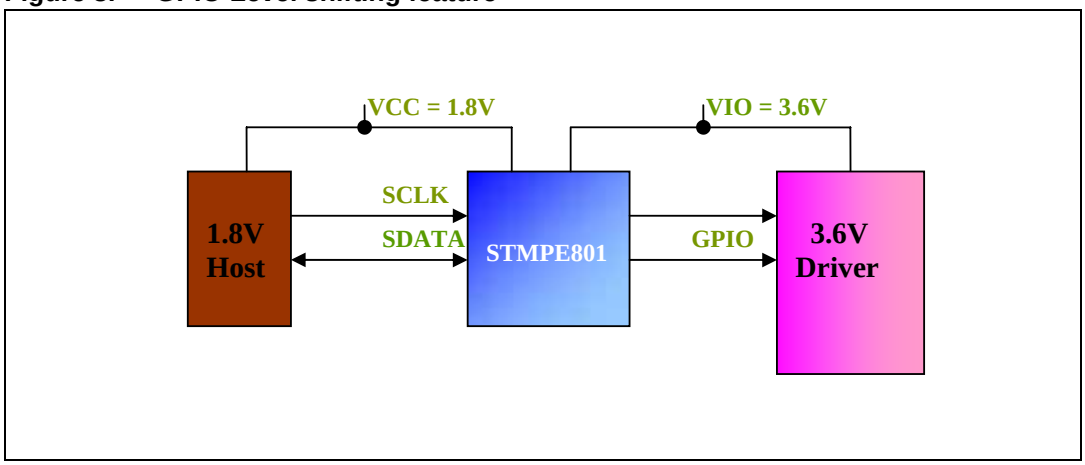
Each GPIO can be individually programmed to generate an interrupt on change of state.

2.5.2 GPIO Level shifting feature

In STMPE801, all GPIO pins are connected to the VIO supply. At reset, all GPIO pins are LOW and if configured as output, the GPIO pins reach HIGH state only if a '1' is written to the corresponding bit in the GPSR register. The HIGH state corresponds to the VIO supply level. This provides a useful level shifting feature without using an explicit level translator device.

For example, low-voltage processors (Example: 1.8V base-band processors) can directly control interfacing modules of much higher operating voltages (example 3.6 V drivers) simply by setting the VIO supply of STMPE801 to the required voltage and setting the GPIO High/Low by writing into the GPIO registers through the I^2C interface.

Figure 8. GPIO Level shifting feature



2.5.3 GPIO-Hot Key feature

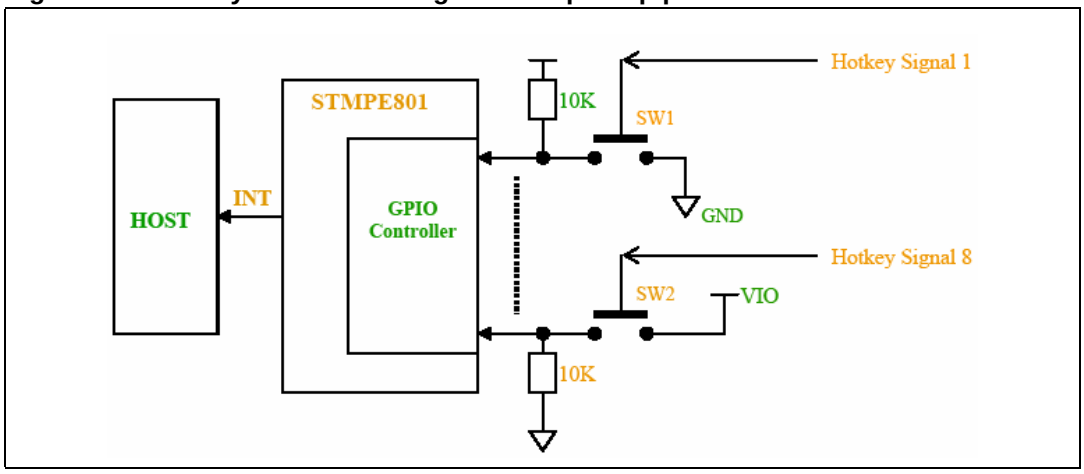
A GPIO is known as 'Hot Key' when it is configured to trigger an interruption to the host whenever the GPIO input changes state from LOW to HIGH or vice versa. This can also be used to Wake-up the host processor from Sleep mode.

The GPIO is normally pulled high or pulled low externally with resistors. Any subsequent change in this logic state triggers an interrupt.

– Programming sequence for Hot Key:

1. The required GPIO pin should be configured as input through the GPDR register.
2. The Global Interrupt (Bit 2) should be enabled by writing '1' and the interrupt polarity should be set (Bit 0) to active low or active high in the SYSCON register.
3. The ISGPIOR register should be read before enabling the GPIO interrupts in order to clear any existing interrupt.
4. The individual GPIO interrupts can be enabled by writing '1' into the corresponding bit in the IEGPIOR register.
5. Now, the port expander is ready to detect the change in logic state on any of the GPIOs and generate an interrupt to the host processor.
6. Each GPIO state change is reported by the corresponding bit in the ISGPIOR register and the source of interrupt can be identified by reading the ISGPIOR register. This permits eight different interrupt sources to be connected to the host through the port expander. It should be noted that the change of GPIO state is recorded in the ISGPIOR even if the GPIO interrupt is not enabled in the IEGPIOR.

Figure 9. Hotkey detection using external pull-up/pull-down resistors on GPIO



2.6 Minimum pulse width

Typically a minimum pulse width of 2 microseconds is required on the GPIO for Hotkey detection. Any pulse width less than the stated value may not be registered in the ISGPIOR.

2.7 Power saving mode of operation

STMPE801 operates entirely on the I^2C clock. When there is no activity on the I^2C bus, the current consumption of the device is extremely low. However, when there is activity on the I^2C bus, current consumption increases, even if the I^2C traffic is not directed to the assigned address.

The host system may choose to shut-down the I^2C block in the STMPE801, if no access to the registers is required. This feature allows the current consumption to drop to the minimum. The host can turn OFF the I^2C block by writing '1' into the I^2C_SHDN bit in the SYSCON register. The I^2C block shuts down on the next valid clock edge of the I^2C clock signal. In this state, the device cannot be accessed by I^2C, as the I^2C block has shut down completely.

Note: The ACK from STMPE801 for a I^2C SHDN command is kept low only for around 250 µs after the SCL goes high. After this duration the device I^2C block shuts down and the SDA line is released. The I^2C host should sample the ACK within this interval.

To turn ON the I^2C block, the system host must reset the STMPE801. This can be accomplished by an active low signal on the RESET_N pin.

Table 8. Typical current rating (VCC = VIO = 3 V)

Operating modes	Typical VIO current consumption at Ta = 25 °C
Normal operation mode with all outputs switching and continuous activity on I^2C bus	~ 10 µA
I^2C Shutdown mode	<10 nA

2.8 Power-Up sequence

1. The GND pin of the device should be grounded first. First ground the GND pin.
2. The Reset_N pin should be connected to VIO through a weak pull-up resistor. Connect the Reset_N pin to VIO through a weak pull-up resistor.
3. The I2C lines, SCL and SDATA should be connected to the host. Connect the I^2C lines, SCL and SDATA to the host. Both lines should have pull-up resistors to VCC.
4. The ADDRESS pin should be connected to GND or VCC as per slave address chosen. Connect the ADRESS pin to GND or VCC according to the slave address chosen. This can also be accomplished by using external weak pull-up and pull-down resistors on those two pins or by driving directly from the host.
5. Both VCC and VIO should be supplied through suitable decoupling capacitors.
6. The INT pin should be pulled high to VIO through a weak pull-up resistor. Pull the INT pin high to VIO through a weak pull-up resistor.
7. With these minimal connections, the device functions in normal mode providing I^2C access to the internal registers. All GPIOs are in default input mode.

- Tips for hardware trouble shooting:

If the device does not function at the end of the power-up sequence provided above, the following tips can be used for troubleshooting.

1. All pins are provided with an internal reverse biased ESD protection diode to GND. The connectivity of the pins vs. GND can be verified.
2. I^2C clock frequency can be reduced or increased to observe the corresponding change in the current consumption of the device. If there is no change, the device connections should be verified again.

3 Conclusion

The STMPE801 is a very versatile device that can be used to reduce the load on the CPU and expand the number of IOs available for the processor. The small size and simple configuration makes it a very attractive proposition for high-end, portable applications like smart phones. This application note provided the setup information to interface this device to a digital engine. The GPIOs in STMPE801 can also be used to perform special functions like key-pad matrix scanning by using suitable software keypad controller routines on the host processor. *(Refer also application note AN2421: Using the STMPE801 as a keypad controller)*

4 Revision history

Table 9. Revision history

Date	Revision	Changes
23-Mar-2007	1	Initial release