AN3299
Application note

A hardware and software guide for the STMPE801 8-bit port expander Xpander Logic™

Introduction

The STMPE801 is a GPIO (general purpose input/output) port expander able to interface a main digital ASIC via the two-line bidirectional bus (I^2C). It offers flexibility as each I/O is configurable as input, output. This device has been designed with very low quiescent current, and includes a wake up feature for each I/O, to optimize the power consumption of the IC.

This document highlights the guidelines and information complementary to the STMPE801; *8-bit port expander Xpander Logic™,* datasheet that is necessary for the successful design of STMPE801 in applications. Please also refer to the AN2421; *Using STMPE801 as keypad controller*, application note for implementation of a matrix keypad controller with the STMPE801.

The first part of the document highlights information on the hardware including external components/connectivity, power, etc.

The second part of the document focuses on information regarding the software, in which programming sample codes are shown.

Contents

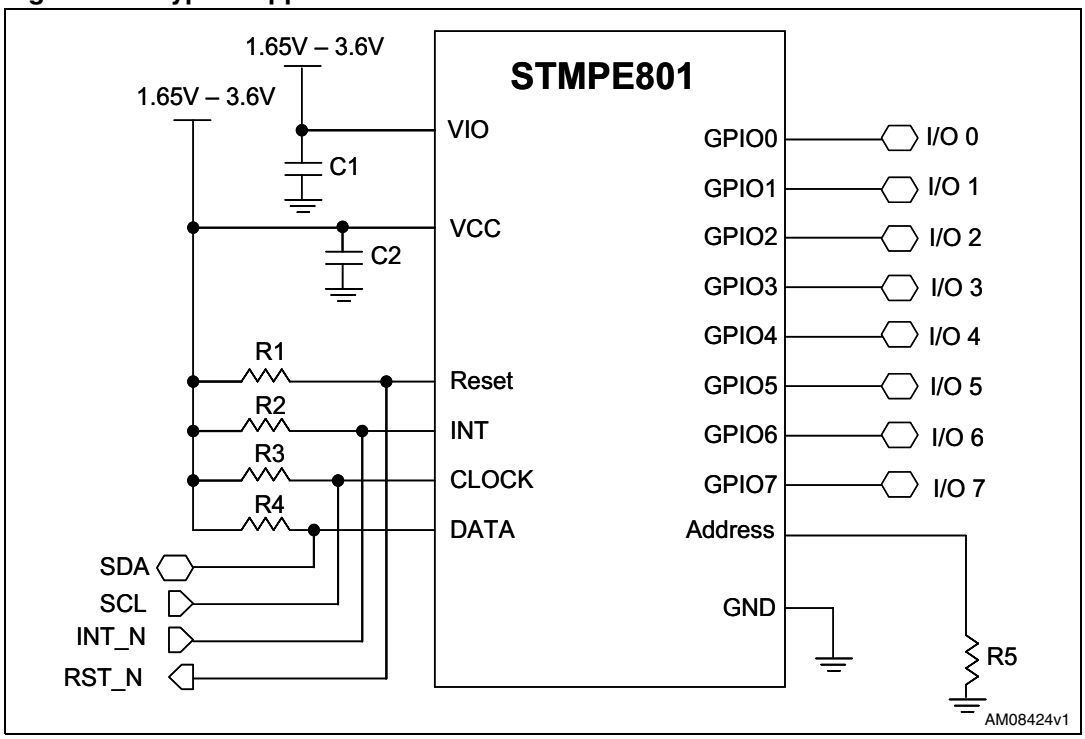
List of figures

1 Hardware

1.1 External components

1.1.1 Typical application circuit

Figure 1. Typical application schematic



Note: Recommended connection at reset is based on the use of baseband/CPU GPIO as control signal. Please refer to Section 1.1.2 for other recommended configurations.

SCLK and SDATA pull up must be to VCC. Pull up to voltage that is less than or greater than VCC may result in excessive leakage current.

In a typical application, the following external components are required:

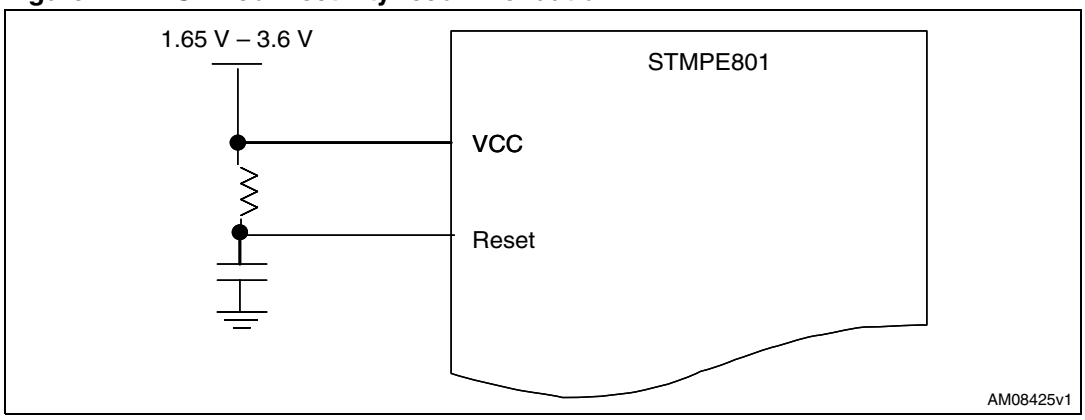
- R1: 10 kΩ pull-up resistor at reset
- R2: 2.2 kΩ-10 kΩ pull-up resistor at INT
- R3: 2.2 kΩ-10 kΩ pull-up resistor at SCLK
- R4: 2.2 kΩ-10 kΩ pull-up resistor at SDATA
- R5: 1 MΩ pull-up/down resistors at address
- C1: 100 nF capacitor at VCC
- C2: 100 nF capacitor at VIO

1.1.2 Reset pin recommendation

The following is a few examples of configurations at the reset pin.

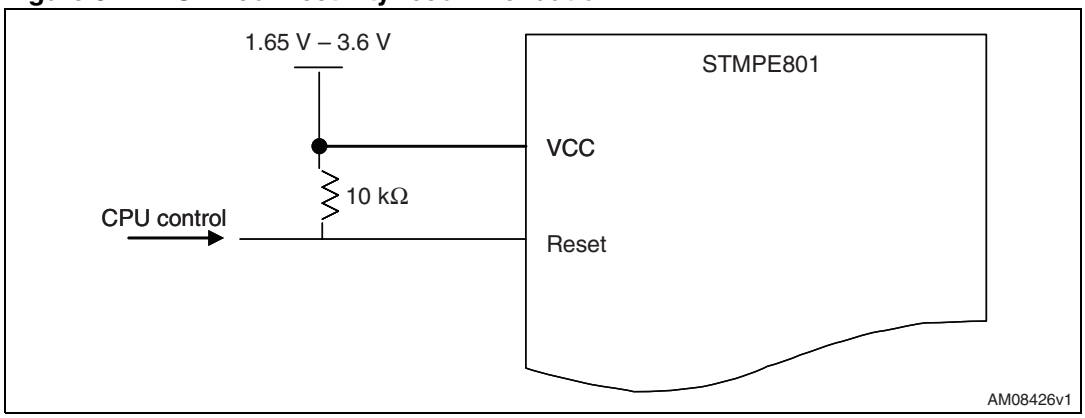
- If a reset delay is desired (recommended) upon power up, an RC delay can be connected to the reset as shown below.

Figure 2. RSTB connectivity recommendation 1



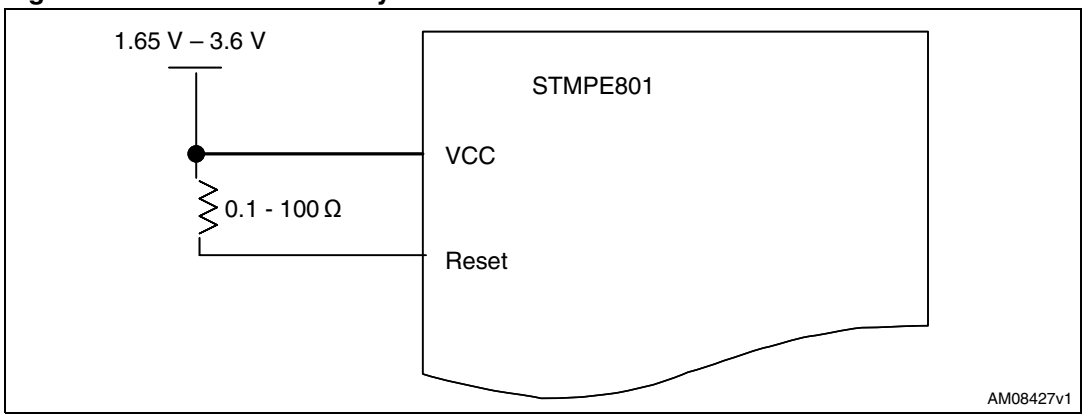
- If external reset assertion is required through CPU/baseband, RSTB can be connected to the GPIO. The diagram below shows the presence of a weak pull-up resistor assuming CPU/baseband control is open drain. The minimum pulse width of the external reset signal is 100 µs.

Figure 3. RSTB connectivity recommendation 2



- If reset delay and external reset assertion, as shown in case1 and case2 above, are not required, it is recommended to short the reset pin to VCC through a low resistor of 0.1 Ω - 100 Ω.

Figure 4. RSTB connectivity recommendation 3



1.1.3 INT pin recommendation

The INT pin is programmable active low or active high. When programmed to active low, a pull-up resistor of 2.2 kΩ -10 kΩ is required. When programmed to active high, a pull-down resistor of 2.2 kΩ - 10 kΩ is required.

If the INT signal is not in use, it is necessary to pull the INT pin to VCC.

1.2 Power supply

There are 2 voltage supplies, VIO and VCC, to the STMPE801.

VIO is the core voltage supply to the internal circuits and powered GPIO_0 to GPIO_7. VCC is the supply for the I^2C blocks.

The operating VCC and VIO voltage range is 1.65 V to 3.6 V. An internal POR circuit monitors the VIO supply and holds STMPE801 in reset state until VIO is valid.

In order to prevent possible leakage, it is necessary to ensure VIO ≥ VCC.

1.2.1 Power sequence (fail safe)

In order to ensure a proper power up of STMPE801, it is necessary to turn on supply to VIO first and then follow with VCC, if VIO and VCC are connected to a different supply voltage.

All GPIO pins of the STMPE801 are in fail safe structure. This means that it is possible to leave GPIO pins driven before STMPE801 power supply on.

It is recommended to connect the SCLK and SDATA pull-up resistors to the same supply as VCC. If a different supply is used, it is recommended to first turn on supply to VCC and then follow by the SCLK and SDATA pull-up supply.

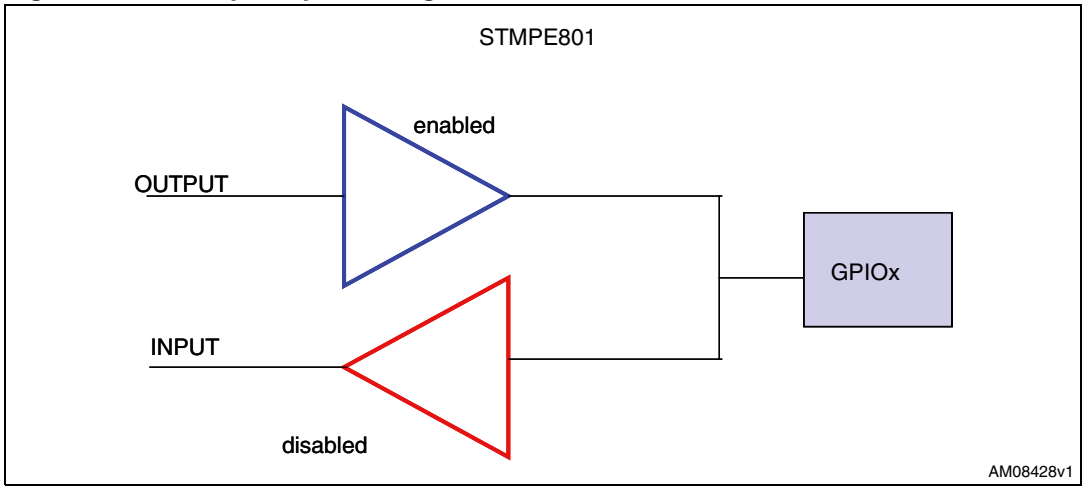
1.3 GPIO output configurations

The STMPE801 provides push-pull type of GPIO output as is. If open drain GPIO outputs are required, it is configurable with a tweak to the software programming routine. See figures *5*, *6*, and *7*.

1.3.1 Push-pull

Set the GPIO to output state through the GPDR (0x12) register. Input path is disabled. Output path is enabled in push-pull configuration.

Figure 5. GPIO push-pull configuration

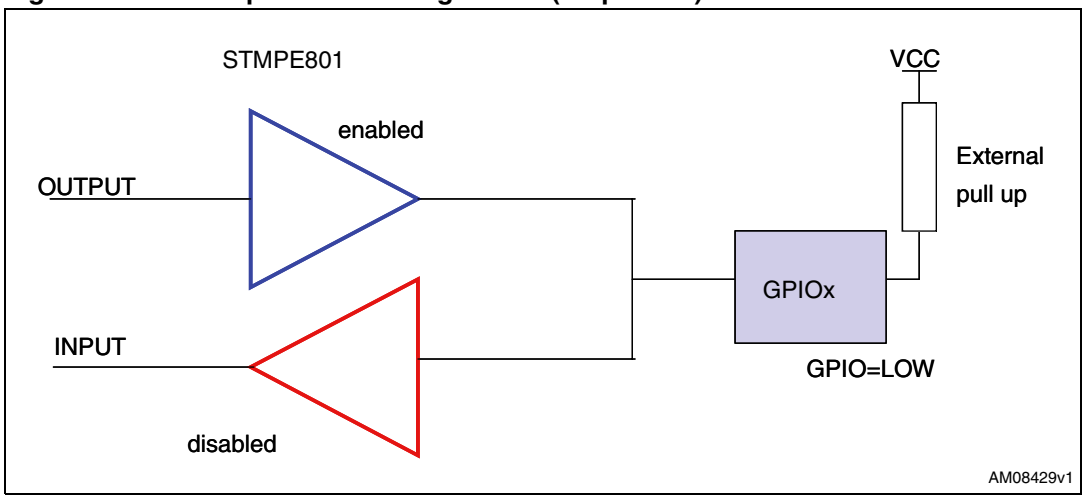


1.3.2 Open drain

GPIO output driven low by STMPE801

Set the GPIO to output state through the GPDR (0x12) register. Input path is disabled. Set the output state to low through the GPSR (0x11) register. Output path is enabled and GPIO pin pulled low.

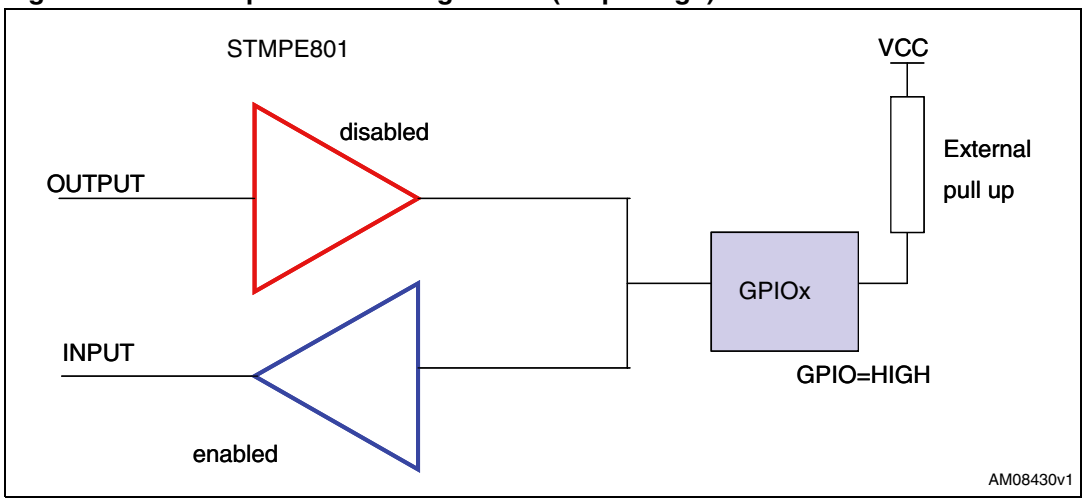
Figure 6. GPIO open drain configuration (output low)



GPIO output pulled high by external pull-up resistor

Set the GPIO to input state through the GPDR (0x12) register. Input path is enabled and output path disabled. GPIO is pulled high by the external pull-up resistor.

Figure 7. GPIO open drain configuration (output high)



1.4 Unused Input pins

If any of the input pins (GPIO, reset, and address pins) are not required or unused in the application, it is necessary to make sure that these pins are either biased to high or low through external pull-up/down resistors. This is to prevent any excessive leakage current.

1.5 Suspend mode

If no access to STMPE801 is required, it is possible for the host to turn off the I^2C block through the SYS_CTRL register (0x04). This shuts down the I^2C block completely and results in minimum current consumption.

A hardware assertion from the host at the reset pin is required to wake up the I^2C block and the device in normal operating mode.

2 Software

2.1 I²C

2.1.1 I²C initialization

It is recommended to insert the software reset as the first command during initialization before starting the I²C transaction.

2.1.2 I²C address

Up to two I²C addresses (0x82 and 0x88) are available through the address pin. In other words, a maximum of 2 pieces of STMPE801 can be used sharing the same I²C bus.

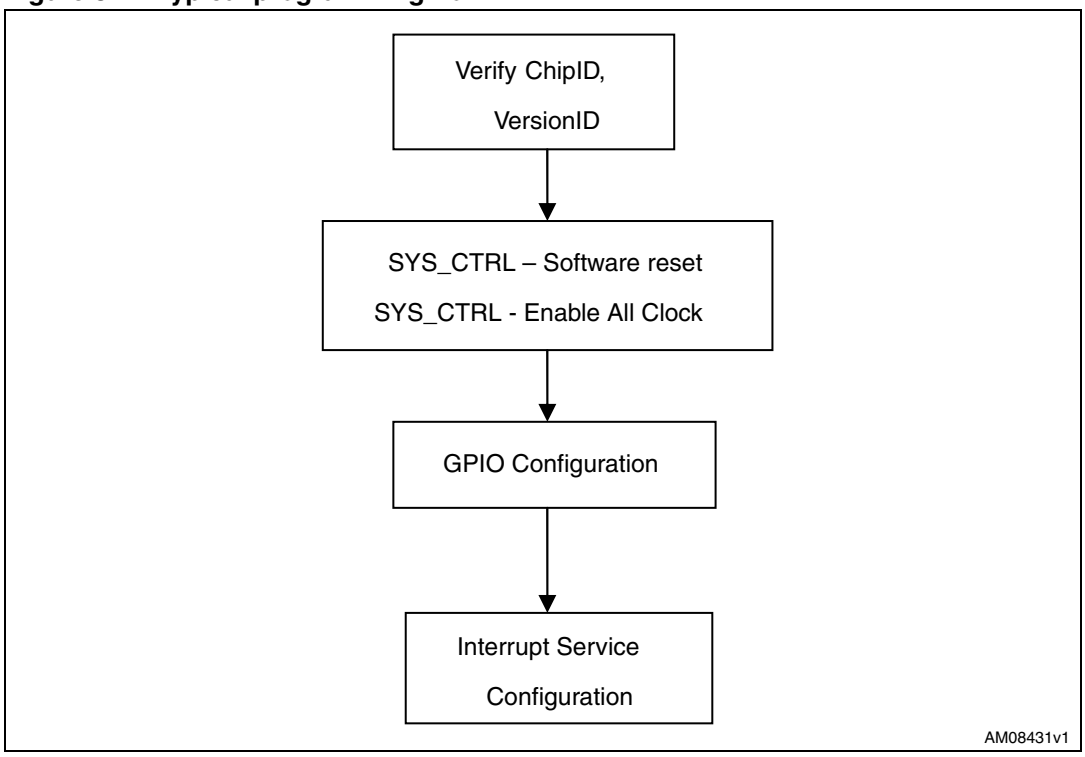
2.1.3 I²C address update/reset

Address bits are reset and updated in the following 3 conditions:

1. Power down/up (POR)
2. Software reset
3. Hardware reset
4. General call reset

2.2 Programming guide

Figure 8. Typical programming flow



2.2.1 Initialization

The following is an example for device initialization based on standard implementation for GPIO.

- GPIO initialization (e.g. GPIO to output)

```
WriteRegister(SYS_CTRL, 0x80);//Issue SW reset
WriteRegister(GPIO_SET_DIR,0x00); //Set port to INPUT
WriteRegister(INT_EN_GPIO_MASK,0xFF); //Enable Input Interrupt
ReadRegister(INT_STA_GPIO);
//Clear all status in GPIO status register
WriteRegister(SYS_CTRL,0x04);
//Active Low, Enable Global Interrupt
```

2.2.2 Interrupt handling

The following is a sample for the device interrupt/hotkey serving routine based on standard implementation.

- Interrupt service routine

```
ON_INTERRUPT

GPIO_INT_STATUS = ReadRegister(INT_STA_GPIO); //Read Interrupt
Status register - Is cleared on read.

If((GPIO_INT_STATUS & 0x80) == 0x80) // Check for interrupt on GPIO7

{

   // Handle GPIO interrupt

}
```

2.3 Hotkey de-bounce

It is possible that a signal from a mechanical connector (e.g. a 3.5 mm earphone jack) is connected to the STMPE801 input as a hotkey input. In such a case, excessive noise is expected from the hotkey input due to the mechanical movement, which the STMPE801 is not equipped to filter. It is necessary that the host implements filtering or de-bouncing by polling the status bit in the Interrupt status register (0x09).

3 Revision history

Table 1. Document revision history

Date	Revision	Changes
04-Nov-2010	1	Initial release.