Smart.IO Hardware Integration Guide

V1.2 October 1st, 2017
richard@imagecraft.com
Richard Man, ImageCraft, https://imagecraft.com/smartio

Physical Characteristics

The Smart.IO module is designed to be easily incorporated into an embedded design with a host MCU. It provides the hardware and software interface to smartphones over the BLE [1] wireless interface. This document describes the hardware interface. Please see the separate document "Smart.IO Software Integration Guide" for the software interface and porting guide.

Physical dimensions:

- 15.5mm x 25.5mm (0.6” x 1”)
- 2x6 0.1in (2.54 x 2.54) male header

Power Requirements

- 3.3V Vdd
- Compatible with 3.3V Arduino with the Smart.IO Arduino shield
- Normal current draw: transmitting (BLE) 15mA @ +8dBm. Active: 2mA
- Sleep mode current draw: 17uA + 0.3uA for I2C EEPROM Standby

Smart.IO Module Placement Recommendations

To obtain the best performance for BLE, the following guidelines should be observed:

- Any host design electronic components should be kept away from the antenna (the white block) of the Smart.IO module
- Likewise, the PCB’s ground planes should be kept away from the same area
- The Smart.IO module should not be covered by a grounded metal case

5V System Compatibility

The Smart.IO module works with a Vdd supply voltage of 1.7V to 3.3V with a maximum of 3.6V. When working with 5V devices like the Atmel AVR and older MCUs, care must be taken to ensure that the Vdd input does not exceed 3.6V. Moreover, the I/O pins must be level-shifted between the two devices; for example, by using a device such as the TXB0108 8-channel

[1] Bluetooth Low Energy, the wireless communication mechanism between the Smart.IO module and smartphones / smart devices

bidirectional channel logic level converter, available here: https://www.adafruit.com/product/395 (if the link is no longer valid, just do a web search with the above descriptions as search terms).

2x6 0.1” Header

The host hardware interfaces with the Smart.IO through the 2x6 0.1” male header. Looking down at the chip module with the header rows on top, pin 1 of the header is located at the lower right corner.

Pin Number (I/O)[2]	**Function**	**Pin Number (I/O)**	**Function**
2 / O	USART Tx	1 / I	USART Rx
4 / I	RESET	3 / I	Vdd[3]
6 / I	SWCLK	5 / I/O	SWDIO
8 / O	SPI MISO	7 / I	SPI MOSI
10 / I	SPI nCS	9 / I	SPI SCK
12 / I	GND	11 / O	Host IRQ / DIO7

Microcontroller Interface

The interface between the host MCU and the Smart.IO consists of:

- SPI - MOSI, MISO, SCK (clock), nCS (chip select)
- Host IRQ - interrupt signal (Smart.IO to MCU), active low. Also used for bootloader firmware update
- RESET - resetting the Smart.IO module, active low
- Vdd and GND

SPI

The MCU is the SPI master in this setup and drives the SPI clock. As multiple SPI slaves may sit on a single SPI bus, the nCS (Chip Select) is used by the SPI master to select the SPI slave which should respond to a particular transaction.

- SPI in 8-bit mode
- Maximum bus frequency is 1 MHz
- CPOL is 0 and CPHA is 1

[2] (I/O) = from the point of view of the Smart.IO module
[3] +3.3V to 3.6V required

- MSBit transmitted first
- nCS is active low

Host IRQ

To inform data availability from Smart.IO to the host MCU, the Host IRQ pin is used. This must be connected to a GPIO pin in the host MCU. On the MCU:

- Configured the connected pin as an input pin
- Input interrupt triggered by transition from low to high
- Signal is pulled high by the Smart.IO module
- Signal is in high impedance state

The pin is held high as long as data is being transmitted from Smart.IO. It's also used for updating firmware using the bootloader. See below.

Smart.IO RESET

The host MCU may use this pin to reset the Smart.IO module. This must be connected to a GPIO pin in the host MCU. On the MCU:

- Configured the connected pin as an output pin
- Normal state is level high
- Must be pulled high by either the MCU internal resistor or an external resistor
- Pull low for one to ten milliseconds to cause a Smart.IO hardware reset

UART Pins

In addition to the SPI and IRQ pins, other pins from the internal BlueNRG1 pins are brought out as well. These pins can be left unconnected in your hardware design, if you do not use their features.

To facilitate advanced debugging, the host firmware can invoke a Smart.IO API to emit debug info on the UART port (at 9600 baud).

UART can also be used in bootload mode. See below.

Bootloader Mode

You may put the BlueNRG1 in bootloader mode by resetting and pulling the DIO7 (Host IRQ) pin high. You will need to use the UART port for bootloader operations. The Smart.IO Arduino Shield available from ImageCraft is a simple to use option for using the bootload mode.

JTAG pins

The JTAG pins (SWCLK and SWDIO) are for flash programming using the JTAG/SWD port.

Arduino Style Shield

WARNING: All official AVR Arduino are 5V only, and will need logic level shifter to be compatible with the Smart.IO.

ImageCraft provides an optional Arduino shield with a dedicated socket for the Smart.IO module. The Smart.IO module and the shield are compatible with 3.3V Arduino-lie systems. The Smart.IO module can draw power from either the shield's Micro-USB connector, or from the Arduino 3.3V pin.

Arduino Shield Header Pinouts

When mounted on the Arduino-style shield, the signals in the 2x6 header are routed to the following pins in the 10-pin Arduino header:

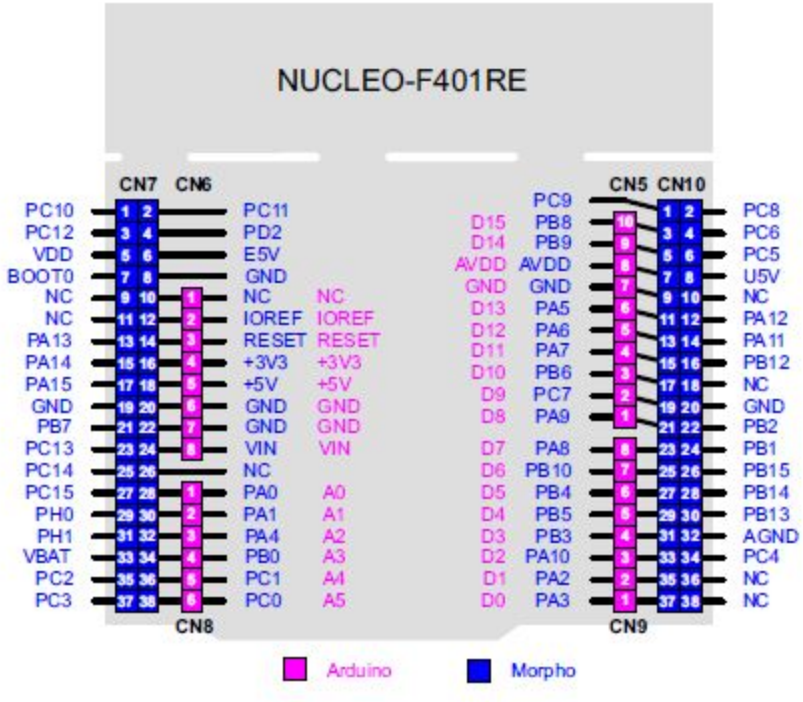
ST Nucleo-401 Pin Number	**Function**	**ST Nucleo-401 Pin Number**	**Function**
1/ PA9	Host IRQ / DIO7	6/ PA5	SPI SCK
2/ PC7	Smart.IO RESET	7/ GND	GND
3/ PB6	SPI CS	8/ Vss	Vss
4/ PA7	SPI MOSI		
5/ PA6	SPI MISO		

The rest of the 2x6 header pins are routed to the JTAG/SWD header and the UART port. See below.

ST-Nucleo Boards with Arduino-style Headers

This is the pinout diagram of the ST Nucleo-411 Arduino compatible board. Other ST Nucleo boards have very similar pinouts:

Figure 16. NUCLEO-F401RE

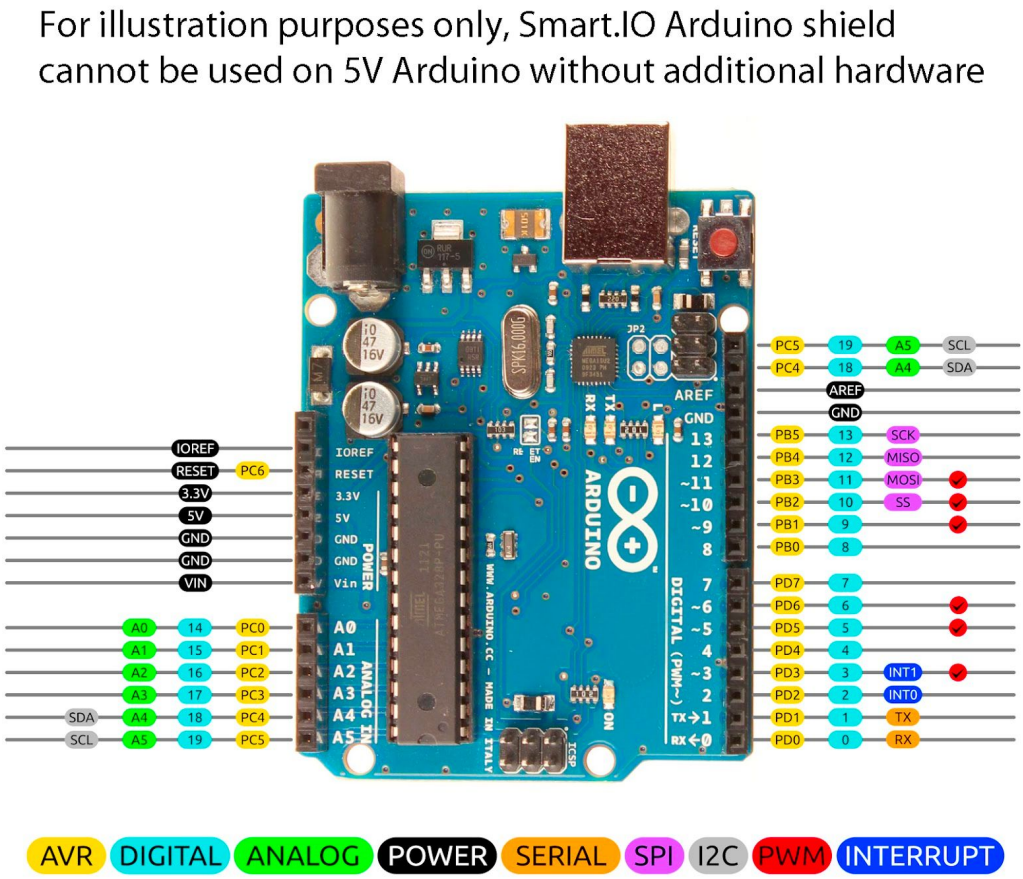


To put Smart.IO in bootload mode for a firmware upgrade with the ST Nucleo-411, you would jumper D8/PA9 and AVDD together while resetting the Smart.IO module.

AVR Arduino

WARNING: All official AVR Arduino are 5V only, and will need logic level shifter to be compatible with the Smart.IO.

This is the pinout diagram of an Arduino board. Note that the diagram uses the Uno R3, as it is considered the “standard” basic Arduino. However, the Uno R3 is 5V, and thus not directly compatible with Smart.IO.



To put Smart.IO in bootload mode for a firmware upgrade with the Arduino, you would jumper PB0 and AREF together while resetting the Smart.IO.

Arduino Shield JTAG Header Pinouts

Additionally, the Smart.IO Arduino shield board comes with a JTAG/SWD header. This is useful for programming the Smart.IO firmware using a JTAG/SWD pod such as the Segger JLINK or the ST ST-LINK. (The pinout details are not presented here since they are standard JTAG header pinouts.) The SWCLK and SWDIO signals from the 2x6 Smart.IO header are routed to this header.

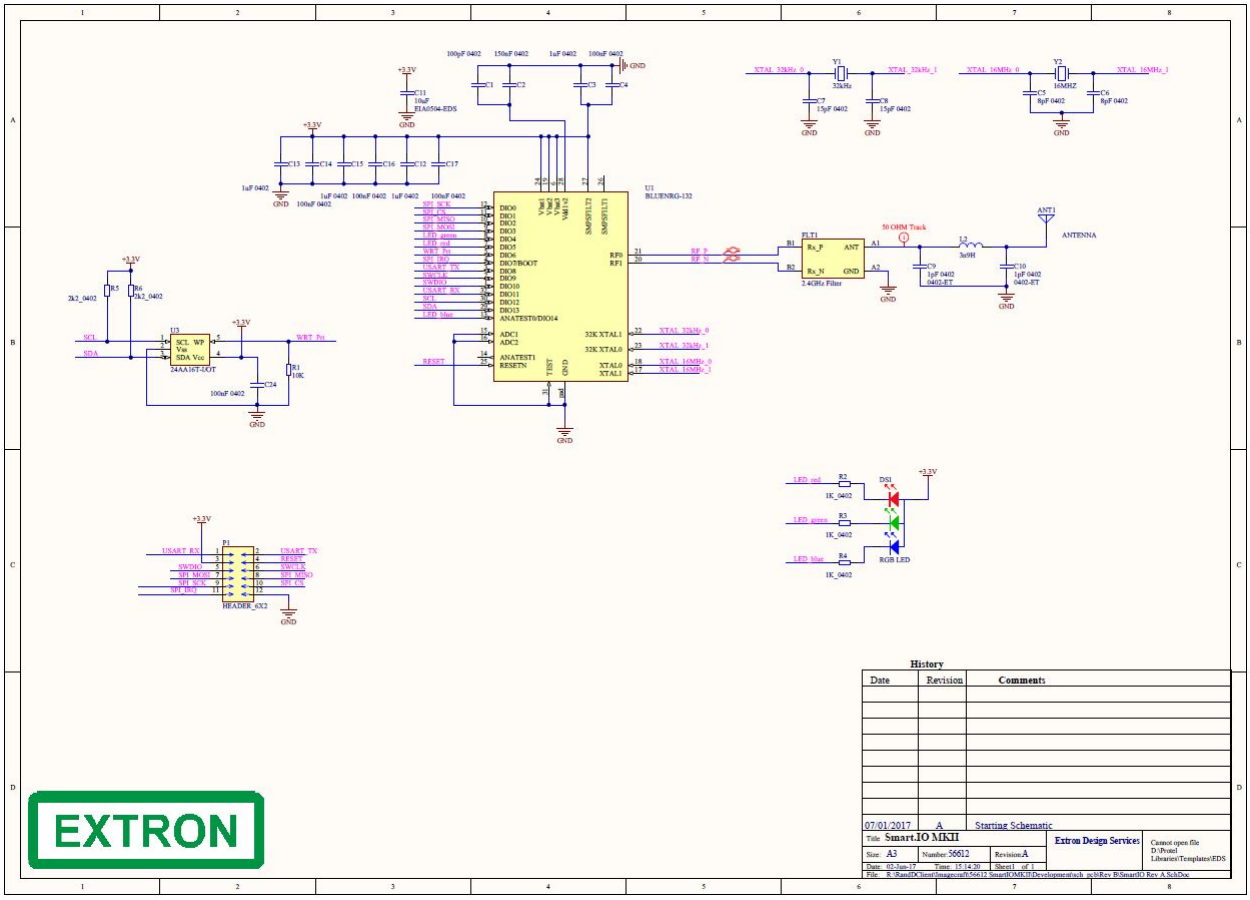
FTDI/USB Micro-USB Connector

FTDI/USB is part of the original V1 release of the Smart.IO Arduino shield and is an optional component in the V2 release. It uses the FTDI industry standard driver. It provides a VCOM port to the Smart.IO module. Smart.IO API functions are provided to use the VCOM for debugging

purposes. It can also be used for bootload programming. The UART Tx and UART Rx from the 2x6 Smart.IO header are routed to the FTDI/USB chip (V1) or the 4-pin UART connector (V2).

Appendix A: Smart.IO Schematic

See here for full size PDF



Appendix B: Smart.IO Arduino Shield Schematic

See here for full size PDF