

# User Guide

## RM1xx Series Development Kit

*Version 1.1*

---

DVK-RM1xx-SM rev. 1

## REVISION HISTORY

Version	Date	Notes	Approver
1.0	24 May 2016	Initial Release	Jonathan Kaye
1.1	08 July 2016	Added section numbers	Sue White

## CONTENTS

1	Laird RM1xx Development Kit Part Numbers.....	4
2	Overview.....	4
3	Introduction.....	4
4	Package Contents .....	4
5	RM1xx Development Kit – Main Development Board .....	5
5.1.	Key Features .....	5
6	Understanding the Development Board .....	7
6.1.	RM1xx Default Configuration and Jumper Settings .....	9
7	Functional Blocks.....	10
7.1.	Power Supply.....	12
7.2.	Reset Button.....	13
7.3.	SWD Interface.....	13
7.4.	4-wire UART Serial Interface .....	13
7.5.	UART Mapping.....	13
7.6.	nAutoRUN Pin and Operating Modes.....	15
7.7.	Mounting Hole for LoRa Dipole Antenna .....	16
8	Software .....	17
9	Breakout Connector Pinouts .....	18
9.1.	J28(J13, J14), J29, J1 (J3, J4, J5, J9), J12(J10), J6 SIO (Special Input / Output Sockets) Breakout Connectors 18	
9.2.	Additional Peripherals / Sensors .....	21
9.3.	Arduino Connector for plugging in an Arduino Shields.....	21
10	Other Features.....	29
10.1.	Current Consumption Measurement .....	29
11	Additional Documentation .....	32

## 1 LAIRD RM1XX DEVELOPMENT KIT PART NUMBERS

Part number: DVK-RM1xx-SM / DVK-RM1xx-SM

Applicable to the following rev. 01 RM1xx module part numbers:

- RM186-SM-01      RM186 LoRa & BLE Module featuring *smartBASIC* (CE)
- RM191-SM-01      RM191 LoRa & BLE Module featuring *smartBASIC* (FCC)

## 2 OVERVIEW

The Laird DVK-RM1xx development kit provides a platform for rapid wireless connectivity prototyping, providing multiple options for the development of LoRa and Bluetooth Low Energy (BLE) applications. This manual is for Rev. 01 and later of the development PCB and relates to RM1xx-SM-01 and later on the PCB itself.

## 3 INTRODUCTION

The Laird LoRa and BLE development kit is designed to support the rapid development of applications and software for the RM1xx series of LoRa and BLE modules featuring Laird’s innovative event driven programming language – *smartBASIC*. More information regarding this product series including a detailed module User’s Manual and *smartBASIC* user guide is available on the Laird’s RM1xx product pages.

## 4 PACKAGE CONTENTS

All kits contain the following items:

<b>Development Board</b>	The development board has the required RM1xx module soldered onto it and exposes all the various hardware interfaces available.
<b>Power Options</b>	<ul style="list-style-type: none"> <li>▪ USB cable – Type A to micro type B. (The cable also provides serial communications via the FTDI USB – RS232 converter chip on the development board.)</li> <li>▪ DC barrel plug with clips for connection to external power supply (7-12V)</li> <li>▪ 3x AAA battery holder fitted on underside of development board</li> </ul>
<b>2pin-Jumpers for Pin Headers x 5</b>	Supplied 2 pin jumpers spares. The jumpers are for 2.54 mm pitch headers used on DVK-RM1xx-V1 development board.
<b>Fly lead x 6</b>	Supplied to allow simple connection of any RM1xxmodule pin (available on Plated Through Holes on J28, J29, J1(J3, J4, J5, J9), J7, J8, J10, J13, J14) to any Arduino pin (available on Plated Through Holes on J15, J23, J16, J17, J20, J24, J21)
<b>External LoRa dipole antenna</b>	External multiband LoRa dipole antenna, 0.9dBi, 863-928MHz, (Laird part # 0600-00060)
<b>Web link Card</b>	Provides links to additional information including the RM1xx user manual, firmware, terminal utilities, schematics, quick start guides, and firmware release notes and much more.
	<p><b>Note:</b>      Sample <i>smartBASIC</i> applications are available to download from the Laird <a href="#">RM1xx applications GitHub webpage</a> or via the <a href="#">Laird global FAE network</a>.</p>

## 5 RM1XX DEVELOPMENT KIT – MAIN DEVELOPMENT BOARD

This section describes the RM1xx development board hardware. The RM1xx development board is delivered with the RM1xx series module loaded with integrated *smartBASIC* runtime engine FW but no onboard *smartBASIC* application; because of this, it starts up in AT command mode by default.

Applications in *smartBASIC* are simple and easy to develop for any LoRa and BLE application. Sample *smartBASIC* applications are available to download from the Laird [RM1xx applications GitHub webpage](#)

The RM1xx development board is a universal development tool to highlight the capabilities of the RM1xx module. The development kit is supplied in a default configuration which should be suitable for multiple experimentation options. It also offers number of header connectors that help isolate on-board sensors and UART from the RM1xx module to create different configurations. This allows you to test different operating scenarios. The development board also has support for plugging in 3<sup>rd</sup> party Arduino Shield boards.

The development board allows the RM1xx series module to physically connect to a PC via the supplied USB cable for development purposes. The development board provides USB-to-Virtual COM port conversion through a FTDI chip – part number FT232R. Any Windows PC (XP or later) should auto-install the necessary drivers; if your PC cannot locate the drivers, you can download them from <http://www.ftdichip.com/Drivers/VCP.htm>

### 5.1. Key Features

The RM1xx development board has the following features:

- RM1xx series module soldered on-development board.
- Power supply options for powering development board from:
  - USB (micro-USB, type B)
  - external DC supply (7-12V)
  - AAA batteries (3xAAA battery holder fitted on underside of development board)
- Regulated 3.3 V for powering the RM1xx module. Optional regulated 1.8 V for powering the RM1xx module via selection switch.
- USB to UART bridge (FTDI chip).
- RM1xx UART can be interfaced to:
  - USB (PC) using the USB-UART bridge
  - External UART source (using IO break-out connectors J1 when development board powered from DC jack)
- Current measuring (for RM1xx module only) options:
  - Pin header (Ammeter)
  - Current shunt monitor IC (volt meter or oscilloscope)
  - Series resistor for differential measurement (oscilloscope)
  - Coulomb Counter
- IO break-out 2.54mm pitch pin header connectors (Plated through Holes) that bring out all interfaces of the RM1xx module [UART, SPI, I2C, SIO (DIO or AIN (ADCs))] and allow for plugging-in external modules/sensors.
- Pin headers jumpers that allow the on-board sensors, LED's (and USB UART FTDI bridge) to be disconnected from RM1xx module (by removing jumpers).
- Three on-board sensors:
  - Analog output Temperature sensor,
  - I2C Coulomb Counter,
  - SPI IO Expander (connects to four LED's and one Button),
- Two Buttons and five LEDs (of which 4 LED's behind SPI IO expander) for user interaction.
- One Analog Buffer (provides a 3.3:1 attenuation) used when Analog source is at 5V into devboard.

- Arduino connectors – allow for plugging of Arduino Shield boards. **DVK-RM1xx-V1 development board is NOT an Arduino Shield**, but is an Arduino base board (similar to the Arduino UNO).
- Arduino connector Test Points – all Arduino connector signals brought out to Plated-through Holes (2.54mm pitch). Allow any Arduino connector signal (D0-D13 or A0-A5) to be connected to any RM1xx module using fly leads for maximum flexibility.
- Arduino connectors orientation at 90 degrees to the dev board long dimension, allowing larger Arduino Shields to hang off side of dev board so not interfering with mounted external antenna or the RM1xx module (the BLE chip antenna).
- External Antenna Mounting Hole – for mounting the RM1xx LoRa external antenna.
- *smartBASIC* runtime engine FW upgrade capability:
  - Via UART (using the FTDI USB-UART)
- *smartBASIC* application upgrade capability:
  - Via UART (using the FTDI USB-UART)

## 6 UNDERSTANDING THE DEVELOPMENT BOARD

Figure 1 illustrates the contents of the DVK-RM1xx development board.

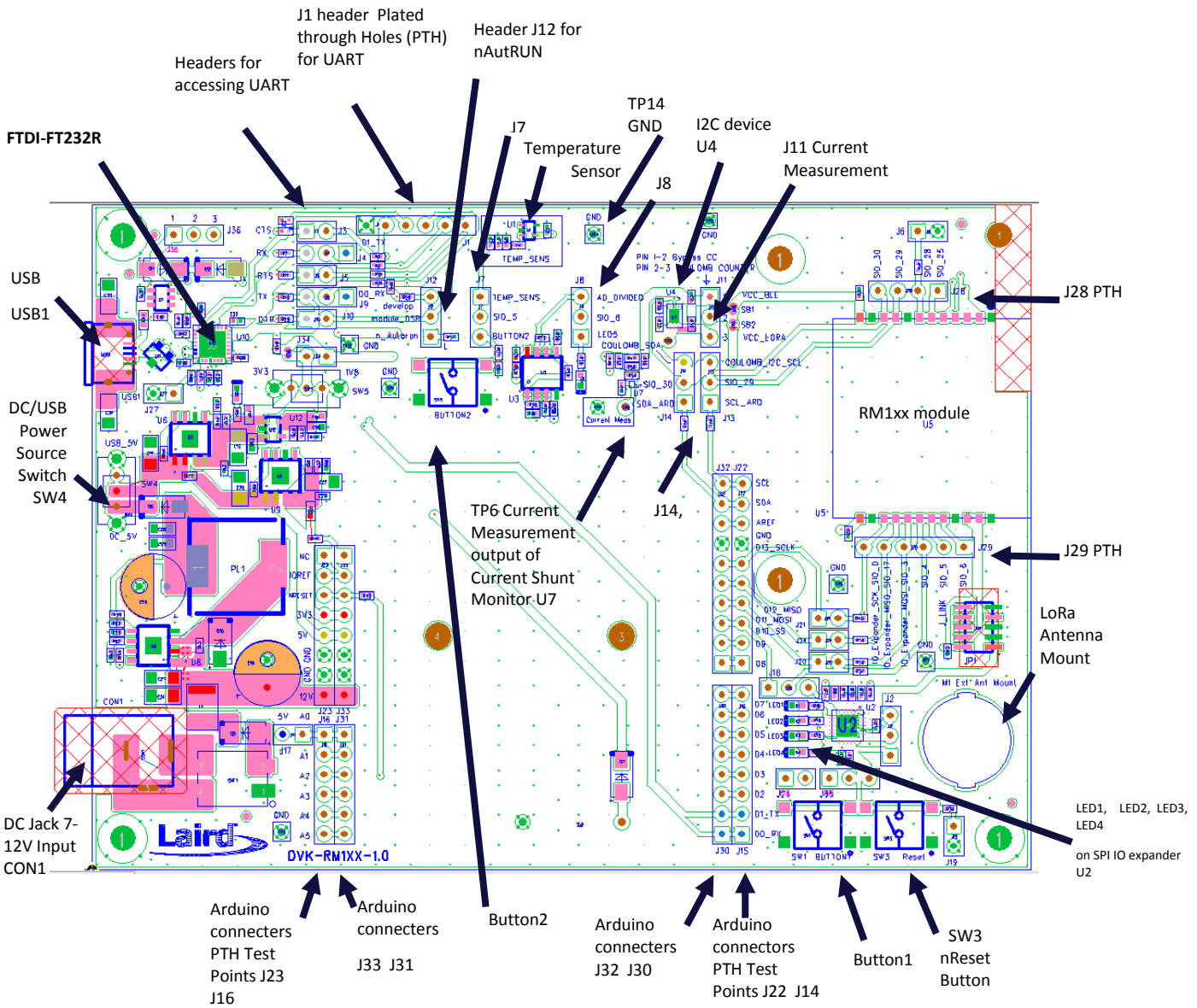


Figure 1: DVK-RM1xx schematic

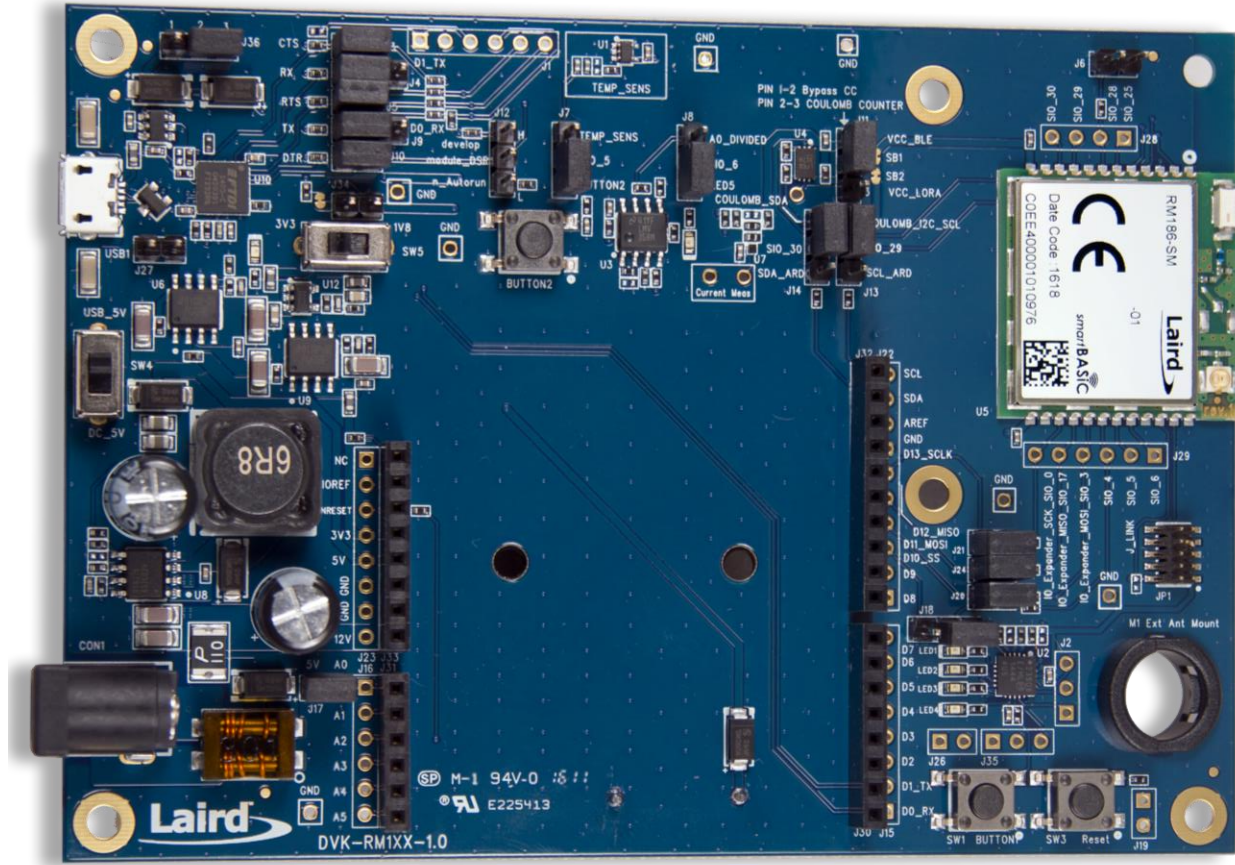
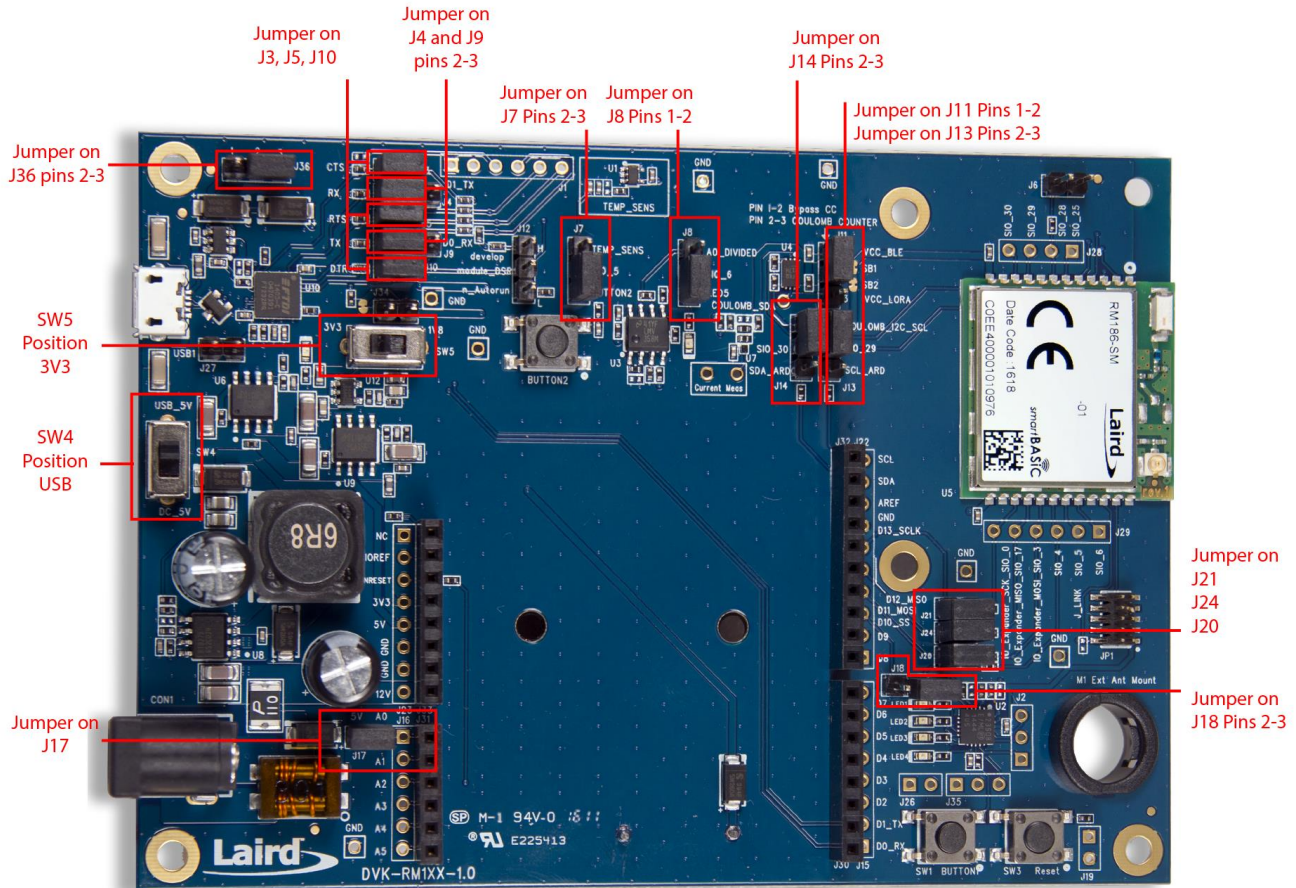


Figure 2: Development board DVK-RM1xx-V1 (fitted with RM186-SM module for example)



## 6.1. RM1xx Default Configuration and Jumper Settings

**Important!** To ensure correct ‘out of the box’ configuration, the **RM1xx** development board must be set according to **Figure 3**.



**Figure 3: Correct development board jumper settings**

## 7 FUNCTIONAL BLOCKS

The RM1xx development board is formed by the major functional blocks shown in Figure 4.

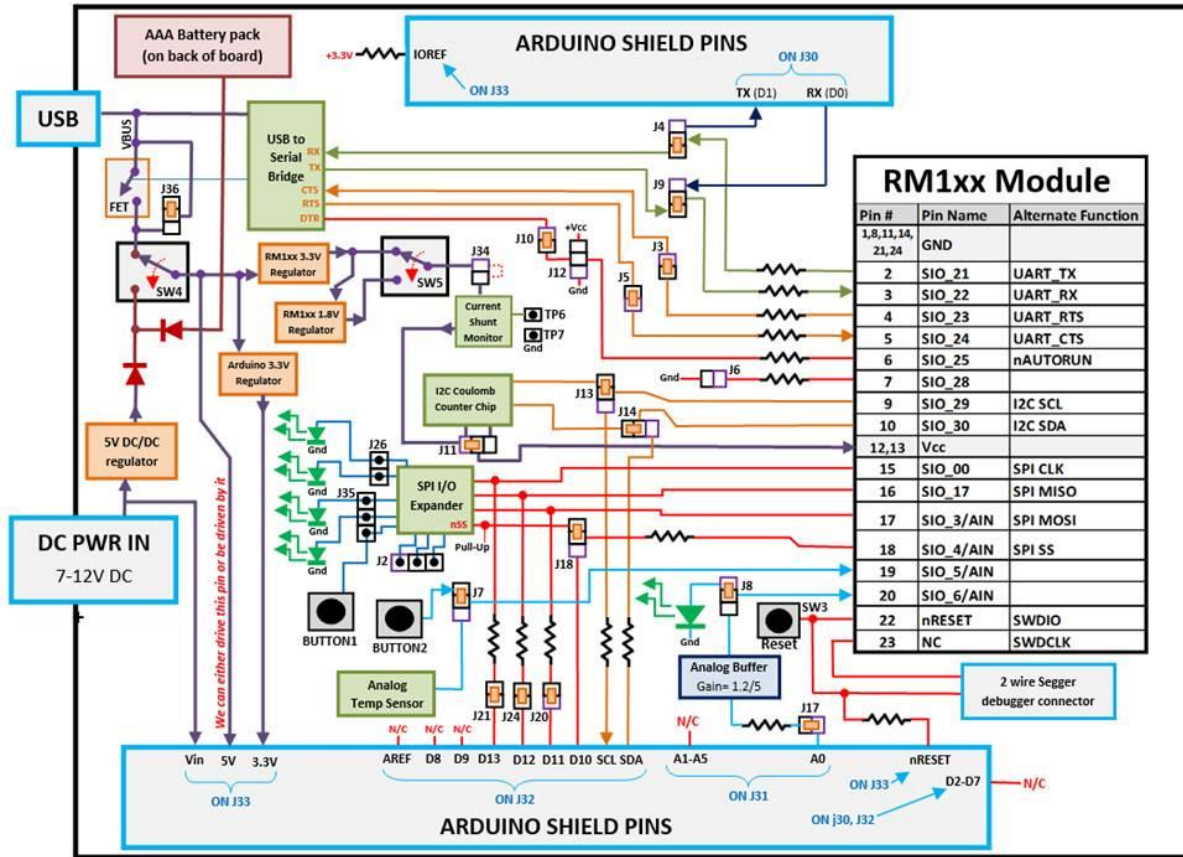


Figure 4: DVK-RM1xx-V1 Block Diagram

Table 1: DVK-RM1xx-V1 Header connectors default jumper positions and – signal mapping on devboard

J#	#pins	Default	On Block Diagram?	Function	Default
J1	6		NO	Serial Port header	Pins not populated
J2	3		YES	I/O expander port pins	Pins not populated
J3	2	Fitted	YES	Routes SIO_23 (RTS) to FTDI CTS or disconnects	Module SIO_23 (RTS) to FTDI CTS
J4	3	Pin 2-3	YES	Routes SIO_21 (TX) to FTDI RX or Arduino TX	Module SIO_21 (TX) to FTDI RX
J5	2	Fitted	YES	Routes SIO_24 (CTS) to FTDI RTS or disconnects	Module SIO_24 (CTS) to FTDI RTS
J6	2	NOT Fitted	YES	Can be used to ground SIO_28	No Jumper- SIO_28 is floating
J7	3	Pin 2-3	YES	Connects BUTTON2 or Temp Sensor to SIO_5	Module SIO_5 to BUTTON2
J8	3	Pin 1-2	YES	Connects SIO_6 to LED5 or Analog Input Buffer	Module SIO_6 to LED5
J9	3	Pin 2-3	YES	Connects SIO_22 (RX) to FTDI TX or Arduino RX	Module SIO_22 (RX) to FTDI TX

J#	#pins	Default	On Block Diagram?	Function	Default
J10	2	Fitted	YES	Connects SIO_25 (nAUTORUN) to FTDI DTR	Module SIO_25 (nAUTORUN) to FTDI DTR
J11	3	Pin 1-2	YES	Inserts/bypass the Coulomb Counter	Coulomb Counter bypassed
J12	3	NOT Fitted	YES	Pulls SIO_25 (nAUTORUN) high or low	NOT Fitted- to use, first remove J10, then pull nAUTORUN high or low via J12
J13	3	Pin 2-3	YES	Routes SIO_29 (I2C SCL) to Coulomb Counter or Arduino SCL pin	SIO_29 routed to Coulomb Counter
J14	3	Pin 2-3	YES	Routes SIO_30 (I2C DAT) to Coulomb Counter or Arduino SDA pin	SIO_30 routed to Coulomb Counter
J15	8		NO	Arduino plated holes for access to D0 thru D7	Pins not populated
J16	6		NO	Arduino plated holes for access to A0 thru A5	Pins not populated
J17	2	Fitted	YES	Connects Arduino A0 pin to Analog Input Buffer	Connects Arduino A0 pin to Analog Input Buffer
J18	3	Pin 2-3	YES	Connects SIO_4 (SPI SS) to I/O Expander or to Arduino D10 (for use as SPI Slave Select)	Module SIO_4 (SPI SS) connected to I/O Expander Slave Select
J19	2		NO	Not populated- just plated through holes	
J20	2	Fitted	YES	Connects SIO_3 (SPI MOSI) to Arduino D11	Module SIO_3 (SPI MOSI) connected to Arduino D11
J21	2	Fitted	YES	Connects SIO_0 (SPI SCK) to Arduino D13	Module SIO_3 (SPI MOSI) connected to Arduino D11
J22	10		NO	Arduino plated holes for access to signals	
J23	8		NO	Arduino plated holes for access to signals	
J24	2	Fitted	YES	Connects SIO_17 (SPI MISO) to Arduino D12	Module SIO_3 (SPI MOSI) connected to Arduino D11
J26				Plated test points for I/O expander pins	
J27	2	NOT Fitted	NO	When installed, it will hold the FTDI chip in reset	FTDI not in reset
J28	4		NO	RM1xx pin plated holes for access	
J29	6		NO	RM1xx pin plated holes for access	
J30	8			Arduino female header	
J31	6			Arduino female header	
J32	10			Arduino female header	
J33	8			Arduino female header	
J34	2	NOT Fitted	YES	Can be used for to make module plus other devboard circuitry current measurement after solder bridge SB3 has been cut	By default, the solder bridge SB3 shorts across this jumper, and the short jumper is not installed
J35				Plated test points for I/O expander pins	

J#	#pins	Default	On Block Diagram?	Function	Default
J36	2	Pin 2-3	YES	When fitted, shorts across USB power switch to allow for a “dumb” USB charger as a power source	By default, the devboard will work properly when connected to a standard USB port on a computer. To use a dumb USB power source, install J36 shorting Pin 1-2

## 7.1. Power Supply

Figure X shows the DVK-RM1xx development board Power Supply block.

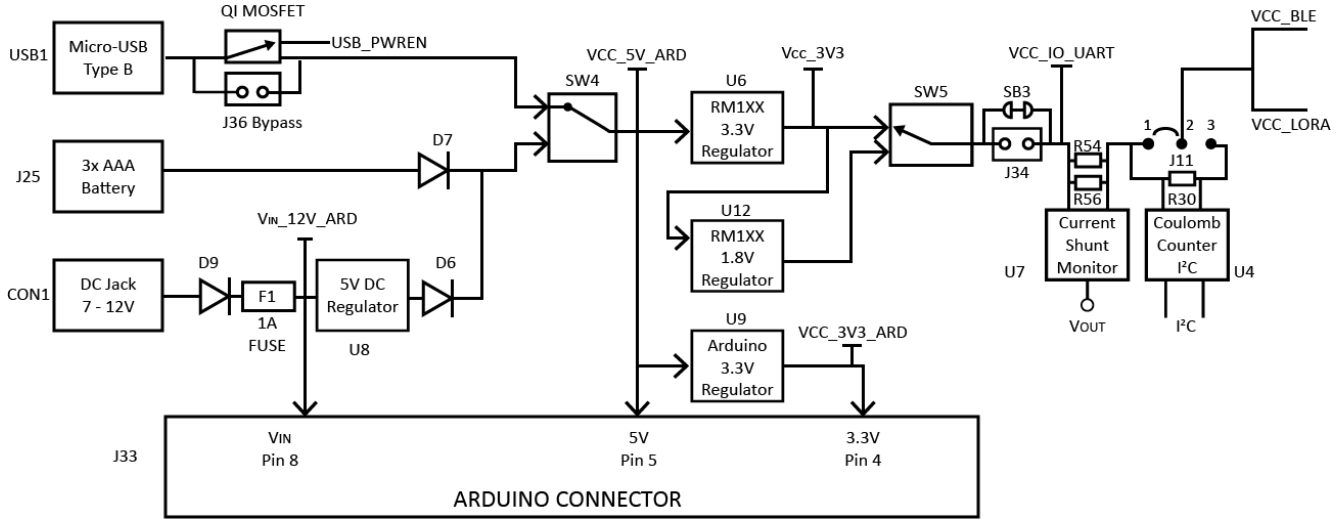


Figure 5: DVK-RM1xx power supply

There are three options for powering the development board:

- USB (type micro-B connector) (USB1)
- external DC supply (7-12V), into DC jack connector (CON1),
- AAA batteries (3xAAA battery holder (J25) fitted on underside of development board)

The power source fed into DC jack (CON1) (which is then regulated by a DCDC to 5V) or 3xAAA batteries (J25) is combined together through diodes (diode-OR) and fed to the SW4 switch. SW4 selects the power source between either USB or the DC jack (5Vregulated)/AAA.

The 5V from the USB or the 5V from DCDC output/AAA batteries is regulated down to 3.3 V with an on-board regulator (U6) on the development board.

The development board also has a 1.8V regulator, allowing for the possibility to power the RM1xx module from a 1.8V rail.

Switch SW5 selects between the regulated 3.3 V and regulated 1.8 V. Default position of SW5 is to select regulated 3.3 V.

Development Board Power Source	SW4	SW5
USB (USB1)	Position “USB”	Position “3V3” or “1V8”
DC jack (CON1) or AAA battery (J25).	Position “DC”	Position “3V3” or “1V8”

The Arduino connector (J33) receives the following:

- 12V from the DC jack (CON1) directly into the Arduino connector J33 pin8 (Vin\_12V\_ARD) via protection diode (D9) and 1A fuse (F1).
- 5V is generated from the on-board DCDC regulator (U8) on the development board into the Arduino connector J33 pin5 (VCC\_5V\_ARD). The U8 12V input is taken from DC jack(CON1).
- 3.3V generated from a separate regulator (U9) is used to supply the Arduino connector J33 pin4, 3.3V domain only (VCC\_3V3\_ARD).

On the development board, the power circuitry is as follows:

- VCC\_3V3 - supplies the FTDI chip power as well as temperature sensor (U1).
- VCC\_IO\_UART - supplies the FTDI chip IO and all other sensors and circuitry.
- VCC\_Radio - supplies the RM1xx series module only plus the Coulomb Counter IC (U4). Current measuring block (the current shunt monitor IC (U7)) on development board only measures the current into power domain VCC\_Radio.  
VCC\_Radio also supplies the VCC\_BLE and VCC\_LORA domains which power the RM1xx series module only.
- VCC\_12V\_ARD - supplies Arduino connector only.
- VCC\_5V\_ARD - supplies Arduino connector only and the Analog buffer IC (U3) attenuator circuit.
- VCC\_3V3\_ARD - supplies Arduino connector only.

---

**Note:** Although the development board allows the RM1xx module to be powered from 3.3V or 1.8V (by selection switch SW5); the Coulomb Counter IC (U4) cannot operate below 2.7V (2.7V-3.6V). When operating RM1xx module on the development board from 1.8V (power selection switch SW5 in position "1V8"), the coulomb counter IC circuit should be bypassed (by fitting jumper back on J11 pins 1 and 2, which is the default).

---

## 7.2. Reset Button

The development board has a reset button (SW3). The Reset is active low (SW3 pushed down). To view its location, refer to [Figure 2](#).

## 7.3. SWD Interface

The development board provides access to the RM1xx module 2-wire interface (called SWD in this document) on JP1. This is NOT required for customer use, since the RM1xx module supports both *smartBASIC* runtime engine firmware and *smartBASIC* application loading over the UART.

## 7.4. 4-wire UART Serial Interface

The development board provides access to the RM1xx module 4-wire UART interface (TX, RX, CTS, RTS) either through USB (via U10 FTDI USB-UART convertor chip) or through a breakout header connector JP5. Refer to [Figure 6](#).

---

**Note:** RM1xx module provides 4-wire UART interface on the HW and the other 4 signals (DTR, DSR, DCD, RI) which are low bandwidth signals can be implemented in a *smartBASIC* application, using any spare digital SIO pins.

---

## 7.5. UART Mapping

UART connection on the RM1xx series module and FTDI IC are shown in table below. [Figure 6](#) explains how the RM1xx series module UART is mapped to the breakout header connectors (J3, J4, J5, J9). These connections are listed in [Table 2](#).

Table 2: SIO / UART connections

RM1xx SIO	RM1xx Default function	FTDI IC UART
SIO.21	UART_TX (output)	USB_RX
SIO.22	UART_RX (input)	USB_TX
SIO.23	UART_RTS (output)	USB_CTS
SIO.24	UART_CTS (input)	USB_RTS

Additionally SIO.25 which is the nAutoRUN input pin on the module can be driven by the USB\_DTR output pin of the FTDI chip. This allows testing the \$autorun\$ application on boot without setting the autorun jumper on the development board. Autorun can be controlled directly from Laird's UWTerminal using the DTR tick box.

### 7.5.1. UART Interface Driven by USB

- **USB Connector.** The development kit provides a USB Type micro-B connector (USB1) which allows connection to any USB host device. The connector optionally supplies power to the development kit and the USB signals are connected to a USB to serial convertor device (FT232R), when SW4 is set to 'USB' position.
- **USB – UART.** The development kit is fitted with a (U10) FTDI FT232R USB to UART converter which provides USB-to-Virtual COM port on any Windows PC (XP or later). Upon connection, Windows auto-installs the required drivers. For more details and driver downloads, visit <http://www.ftdichip.com/Products/FT232R.htm>.
- **UART interface driven by USB FTDI chip.** In normal operation, the RM1xx UART interface is driven by the FTDI FT232R USB to UART converter.

### 7.5.2. UART Interface Driven by External Source

- **UART interface driven by external UART source.** The RM1xx module UART interface (TX, RX, CTS, RTS) is presented at a 2.54 mm (0.1") pitch header (J1). To allow the RM1xx UART interface to be driven from the breakout header connector (J1), the following must be configured:
  - Development board must be powered from DC jack (CON1) or AAA batteries (J25) and switch SW4 is in DC position.
  - The FTDI device must be held in reset. This is achieved automatically by removal of USB cable, placing SW4 in the DC position, or fitting a jumper on J27.
  - Also jumpers on header connectors J3, J4, J5, J9 allows the four RM1xx UART pins to be physically isolated as well from USB-UART FTDI device. By default, the jumpers on J3, J4, J5, J9 are fitted to route UART pins to U10 FTDI FT232R USB –UART convertor.

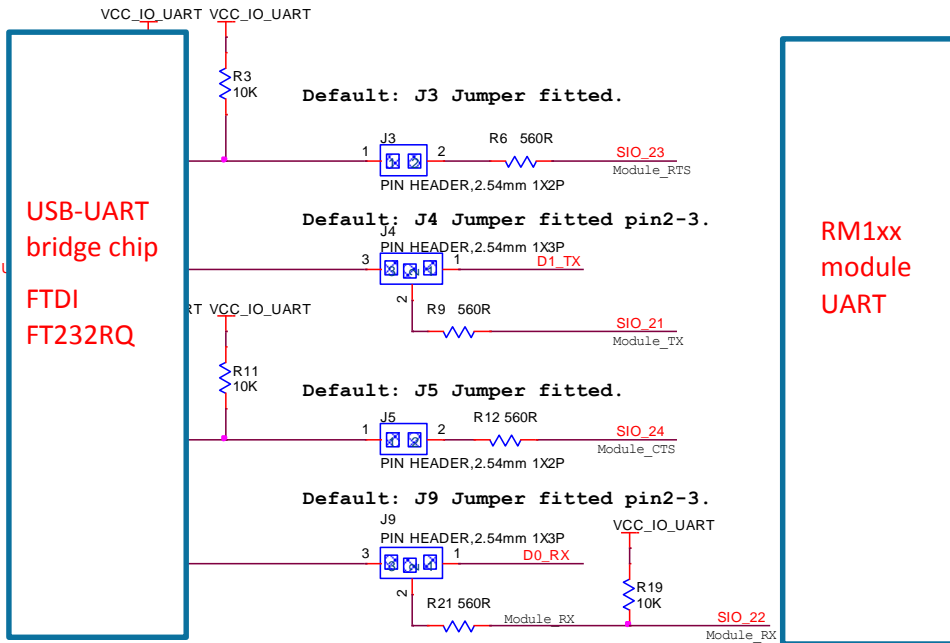


Figure 6: USB to UART Interface and Header to UART interface

J1 pinout is designed to be used with FTDI USB-UART TTL (3.3V) convertor cables (found at <http://www.ftdichip.com/Products/Cables/USBTTLSerial.htm>). One example is FTDI part TTL-232R-3V3.

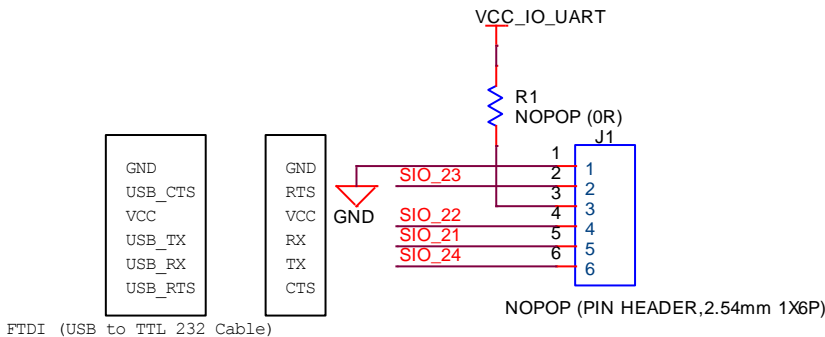


Figure 7: J1 wiring to match FTDI USB-UART cable (TTL-232R-3V3 cable)

Remove jumpers on J3, J4, J5, J9 when connecting an external FTDI USB-UART TTL (3.3V) convertor cable using J1.

## 7.6. nAutoRUN Pin and Operating Modes

On the development board USB\_DTR output (FTDI chip U10) from PC is wired to RM1xx module pin SIO25 (pin6) which is the nAutoRUN pin.

**Note:** *smartBASIC* runtime engine FW checks for the status of nAutoRUN during power-up or reset. The nAutoRUN pin detects if the **RM1xx** module should power up into “Interactive / Development Mode (3.3 V)” or “Self-contained Run Mode (0v)”. The module enters Self-contained Run Mode if the nAutoRUN pin is at 0V and an application called “\$autorun\$” exists in the modules file system, then the *smartBASIC* runtime engine FW will execute the *smartBASIC* application script automatically; hence the name *Self-contained Run mode*.

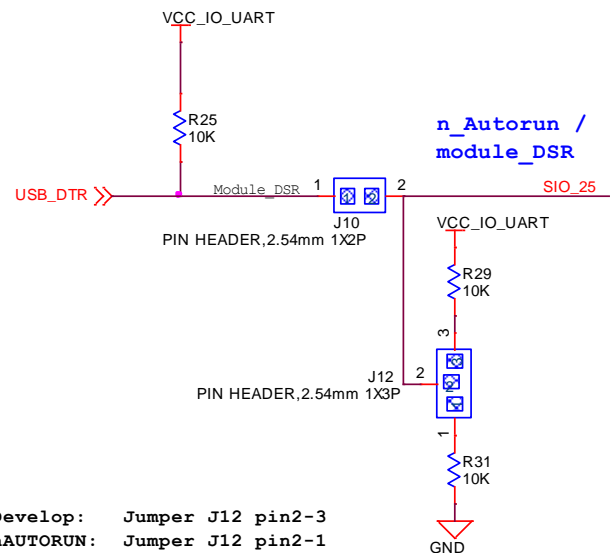
The nAutoRUN pin inhibits the automatic launch \$autorun\$ application on power-up. Tying nAutoRUN to 3.3V inhibits the \$autorun\$ application from running. The J12 3-pin header allows a jumper to be fitted to select between the two operating modes.

**Note:** Header J10 jumper MUST be REMOVED when using J12 3-pin header to select nAutoRUN function.

Table 3: RM1xx nAutorun header

nAutoRUN pin	RM1xx Operating Mode		
	Interactive / Development Mode	Self-contained Run Mode (autorun mode)	Circuit

J12 jumper position



J10 header connector allows USB\_DTR signal from the FTDI chip to be disconnected from reaching the RM1xx.

Since RM1xx nAutoRUN pin6 (SIO.25) is connected to PC FTDI USB\_DTR line, via the J10 header connector:-

- By default a Jumper is fitted into J10 header connector to allow PC (using uWTerminal) to control nAutoRUN pin (SIO.25); with no jumper fitted to J12 (which is the default) (see **Error! Reference source not found.**).

To disconnect RM1xx nAutoRUN pin6 (SIO.25) from the PC FTDI USB\_DTR line:-

- Remove Jumper from J10 header connector. Then nAutoRUN can be controlled by J12 jumper on appropriate pins as per above table.

### 7.7. Mounting Hole for LoRa Dipole Antenna

The mounting hole for the LoRa Dipole Antenna is illustrated in [Figure 1](#) and [Figure 2](#).



## 8 SOFTWARE

The development board connects the RM1xx module to a virtual COM port of a PC or other device. From a PC, you can communicate with the module using Laird's UW Terminal application. (version 6.51 or newer).

UW TerminalX is a terminal emulation application capable of running on Windows, Mac, and Linux. It was developed specifically to aid development and testing of Laird modules. It allows connection to serial devices using any combination of the communications parameters listed in Table 4.

**Table 4: UwTerminalX Communication Parameters for RM1xx**

<b>COM Port:</b>	1 to 255
<b>Baud rate:</b>	300 to 921,600 Note: Baud rate default is 115200 for RM1xx.
<b>Parity:</b>	None, Odd, Even
<b>Data Bits:</b>	8
<b>Stop Bits:</b>	1 or 2
<b>Handshaking:</b>	None or CTS/RTS

**Note:** Baud rates higher than 115200 depend on the COM port capabilities of the host PC and may require an external USB – RS232 adapter or PCMCIA card.

The benefits of using UWTerminalX include:

- Continually displayed status of DSR, CTS, DCD, and RI
- Direct control of DTR on the host PC via a check box
- Direct control of RTS, if CTS / RTS Handshaking is disabled when UWTerminalX is launched
- Sending of BREAK signals
- BASIC tab provides standalone testing and development of *smart*BASIC applications and allows UWTerminalX operation to be automated.
- Additional built-in features (right click in Terminal tab screen) to accelerate development including Automation and various XCompile / Load / Run options for downloading *smart*BASIC applications into the RM1xx.

**Note:** Full details on *smart*BASIC are available in the *smart*BASIC User Manual available for download at the Laird website. This document also includes a basic introduction to UwTerminalX.

**Tip:** If the module returns a four hex digit error code: In UwTerminal, select those four digits, right-click, and select **Lookup Selected ErrorCode**. A description of the error is printed on screen.

## 9 BREAKOUT CONNECTOR PINOUTS

### 9.1. J28(J13, J14), J29, J1 (J3, J4, J5, J9), J12(J10), J6 SIO (Special Input / Output Sockets) Breakout Connectors

Access to all 14 RM1xx series module signal pins (SIO's = signal Input /Output) is available on header connectors J28(J13, J14), J29, J1 (J3, J4, J5, J9), J12(J10), J6 (2.54 mm pitch headers).

**Note:** The RM1xx module signal pins designation SIO (Signal Input /Output).

- DEFAULT type is DIO (Digital Input or Output) or UART (on fixed pins)
- ALTERNATE type is either AIN (Analog Input ADC), I2C, SPI, DIO (on fixed pins)
- Alternate function is selectable in *smartBASIC* application
- DIO or AIN functionality is selected using the GpioSetFunc() function in *smartBASIC*
- AIN configuration selected using GpioSetFunc() function
- I2C, UART, SPI controlled by xxxOPEN() functions in *smartBASIC*
- SIO\_21 to SIO\_24 are DIO by default when \$autorun\$ app runs on power up

These breakout connectors can interface to a wide array of sensors with the RM1xx function user configurable by *smartBASIC* application script from the default function (DIO, UART) to alternate functions (AIN (ADC), I2C, SPI, DIO). The RM1xx development kit incorporates additional fly-lead cables inside the box, to enable simple, hassle-free testing of the multiple interfaces.

Table 5 shows the RM1xx module pins that are brought out to plated through Holes (suitable for 2.54 mm pitch headers).

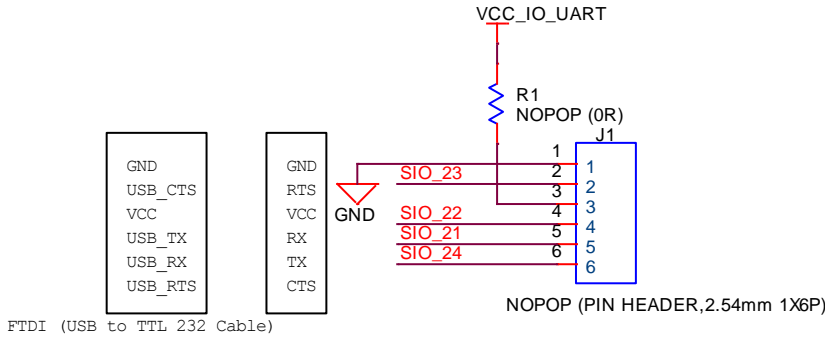
Table 5: Module pins exposed by plated through holes

Plated Through Holes or Header Connector		RM1xx module signals exposed
<p><b>J28</b> RM1xx pin plated holes for access</p> <p>NOPOP (PIN HEADER,2.54mm 1X4P)</p> <p>J28</p>	<p>RM1xx pin plated holes for access</p> <p>SIO_25, SIO_28, SIO_29, SIO_30,</p>	
<p><b>J29</b></p> <p>NOPOP (PIN HEADER,2.54mm 1X6P)</p>	<p>RM1xx pin plated holes for access</p> <p>SIO_6, SIO_5, SIO_4, SIO_3, SIO_17, SIO_0,</p>	

**Plated Through Holes or Header Connector** **RM1xx module signals exposed**

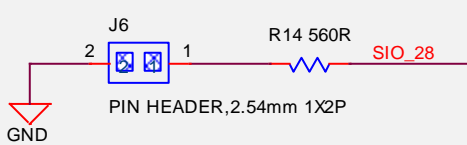
**J1**

Serial Port plated holes for access



**J6**

SIO\_28  
Can be used to ground SIO\_28  
No Jumper on SIO\_28

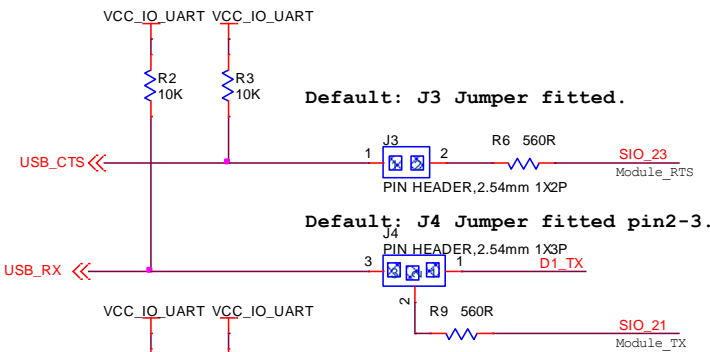


**J3**

J3 Routes SIO\_23 (RTS) to FTDI CTS or disconnects.

**J4**

J4 Routes SIO\_21 (TX) to FTDI RX or Arduino TX

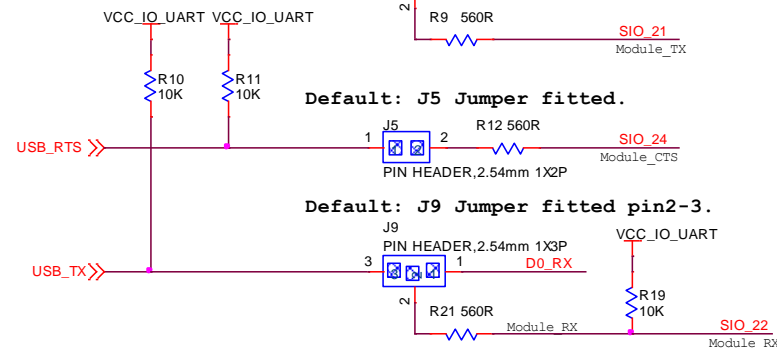


**J5**

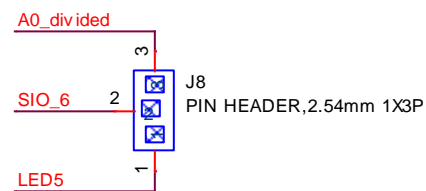
J5 Routes SIO\_24 (CTS) to FTDI RTS or disconnects

**J9**

J9 Connects SIO\_22 (RX) to FTDI TX or Arduino RX



Plated Through Holes or Header Connector	RM1xx module signals exposed
<p><b>J10</b></p> <p>Develop: Jumper J12 pin2-3 nAUTORUN: Jumper J12 pin2-1 Line driven by USB_DTR: No jumper in J12 (Default) Line driven by USB_DTR: J10 jumper fitted (Default)</p>	<p>Connects SIO_25 (nAUTORUN) to FTDI DTR</p>
<p><b>J13</b> , <b>J14</b></p> <p>I2C_SCL: J13 jumper fitted pin2-3 (Default) I2C_SDA: J14 jumper fitted pin2-3 (Default)</p>	<p>J13 Routes SIO_29 (I2C_SCL) to Coulomb Counter or Arduino SCL pin.  J14 Routes SIO_30 (I2C_SDA) to Coulomb Counter or Arduino SDA pin.</p>
<p><b>J7</b></p> <p>Button2: J7 jumper fitted pin2-3 (Default)</p>	<p>J7 routes SIO_5 out . Connects BUTTON2 or Temp Sensor to SIO_5</p>

Plated Through Holes or Header Connector	RM1xx module signals exposed
<p><b>J8</b></p>  <p>J8 PIN HEADER, 2.54mm 1X3P</p> <p>LED5: J8 jumper fitted pin2-1 (Default)</p>	<p>J8 Connects SIO_6 to LED5 or Analog Input Buffer</p>

## 9.2. Additional Peripherals / Sensors

The RM1xx development board provides for simple and hassle free connectivity to a wide range of sensors, but also includes several on-board sensors and options to enable a developer to test functionality straight out of the box.

In the *smartBASIC* application code written to use sensors on the development board (including the Temperature sensor (U1) – analog output, LED5(D5), Button1(SW1), SPI IO expander (U2), I2C Coulomb Counter (U4), the SIO pins direction and type must be set in the *smartBASIC* application to override the defaults in the RM1xx firmware.

For more information on these sample applications, see the Sample Applications for the RM1xx application note, available on the documentation tab of the [RM1xx product page](#).

## 9.3. Arduino Connector for plugging in an Arduino Shields

The DVK-RM1xx-V1 development board is NOT an Arduino Shield, but is an Arduino base board (like the Arduino UNO).

The four Arduino connectors (J30, J31, J32 and J33) on the development board allow Arduino Shields to be plugged in.

- All Arduino connector signals are brought out to Plated-through Holes (2.54mm pitch) J15, J16, J22, J23. This allows any Arduino connector signal (D0-D13 or A0-A5) to be connected to any RM1xx module using fly leads for maximum flexibility.
- Arduino connectors orientation are at 90 degrees perpendicular to the long dimension, allowing larger Arduino Shields to hang off side of the board without interfering with a mounted external antenna or the RM1xx module (the BLE chip antenna).

There are Arduino pins that are not used on the development board:-

- Arduino pin IOREF** on development board (on J33pin2), is connected to 3.3V domain (VCC\_ARD\_ARD via 0R resistor (R44). Arduino IOREF allows Arduino shields to adapt to the voltage provided from the board, Since we are sending 3.3V up (from the development board) to the IOREF , the Arduino documentation states that a properly configured Shield should respect our logic levels as a function of this pin. Therefore we do not need level translators, just series resistors in all Shield IO lines to help protect against an inappropriate logic level (something greater than 3.3V). This series of resistors (R37, R38, R39, R40, R42, R41, R14, R6, R9, R12, R21), provide the voltage drop as current flows through, activating the ESD protection diode in the RM1xx module.  
**RM1xx module PINS DO NOT SUPPORT 5V IO.** Do not connect greater than 3.3V IO from Arduino Shields others.
- Arduino RESET pin** on development board (on J33pin3), is connected to RM1xx nRESET pin (U5pin22 via 0R resistor (R43).

- **Arduino pins A1 through to A5** are simply left open but are wired out from Arduino connector J31 to Plated-through Holes on J16. These Arduino pins (A2-A5) can be fly-leaded to any RM1xx analog input pin.
- **Arduino D2-D9** are simply left open but are wired out from Arduino connector J30 and J32 to plated through holes on J15 and J22 respectively next to the Arduino shield connectors. These Arduino pins (D2-D9) can be accessed by soldering a jumper wire to plated through holes on J15, to any RM1xx digital pin.
- **Arduino AREF** is simply left open but is wired out from Arduino connector J32 to plated through holes on J22 which is next to the Arduino shield connector (J32). AREF is supplied by a Shield board and is an input to the Arduino base board to indicate the maximum expected value of the analog signal. The RM1xx module does not support this function.

Table 6: Arduino connectors and signals

J#	Arduino Connectors and Plated Through Holes (Test points)	Arduino signals
J30	<p>RM1xx pin plated holes for access</p> <p>HEADER, FEMALE, 2.54mm, 1X8P      NOPOP (PIN HEADER, 2.54mm 1X8P)</p>	<p>Arduino female header J30. J15 is plated through holes for access signals on J30</p>
J29	<p>HEADER, FEMALE, 2.54mm, 1X6P</p> <p>PIN HEADER, 2.54mm 1X2P      NOPOP (PIN HEADER, 2.54mm 1X6P)</p> <p>J17 jumper fitted (Default)</p>	<p>Arduino female header J31. J16 is plated through holes for access signals on J30.</p> <p>J17 Connects Arduino A0 pin to Analog Input Buffer (U3).</p>
J32	<p>Enable IO expander: J18 Jumper in pin2-3 (default)</p> <p>J20 jumper fitted (Default)      IO_Expander_CS_SIO_4</p> <p>J21 jumper fitted (Default)</p> <p>J24 jumper fitted (Default)</p> <p>PIN HEADER, 2.54mm 1X2P      PIN HEADER, 2.54mm 1X2P      PIN HEADER, 2.54mm 1X2P      HEADER, FEMALE, 2.54mm, 1X10P      NOPOP (PIN HEADER, 2.54mm 1X10P)</p>	<p>Arduino female header J32. J22 is plated through holes for access signals on J30. Connects SIO_4 (SPI SS) to I/O Expander (U2) or to Arduino D10 (for use as SPI Slave Select). J20 Connects SIO_3 (SPI MOSI) to Arduino D11. J20 Jumper fitted. J24 Connects SIO_17 (SPI MISO) to Arduino D12. J24 Jumper fitted. J21 Connects SIO_0 (SPI SCK) to Arduino D13. J21 Jumper fitted.</p>
J33	<p>HEADER, FEMALE, 2.54mm, 1X8P      NOPOP (PIN HEADER, 2.54mm 1X8P)</p>	<p>Arduino female header J33. J23 is plated through holes for access signals on J33.</p>

### 9.3.1. Analog input Buffer and attenuator circuit (U3)

Figure 8 shows the Analog Buffer circuit that accepts a 0V to 5V analog input signal from Arduino shield pin and scale it down to an acceptable range of 0V to 1.2V set by potential divider R26 (3.3kOhms) and R28 (1kOhms) with a gain of 0.23(=1/ (1+3.3)).

Max Input (Volts) (A0_5V_IN)	R26 (kOhms)	R28 (kOhms)	Output (Volts) (A0_divided)
5	3.3	1	1.16

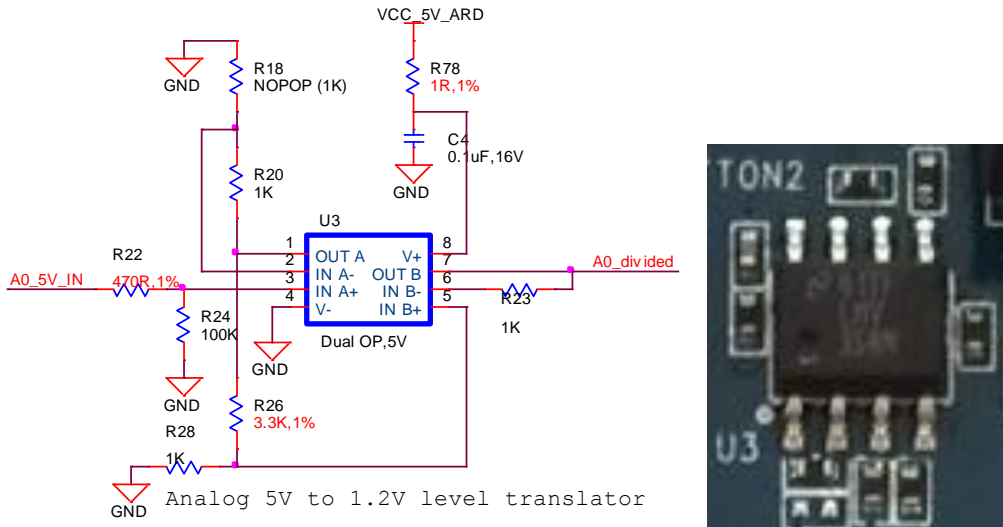
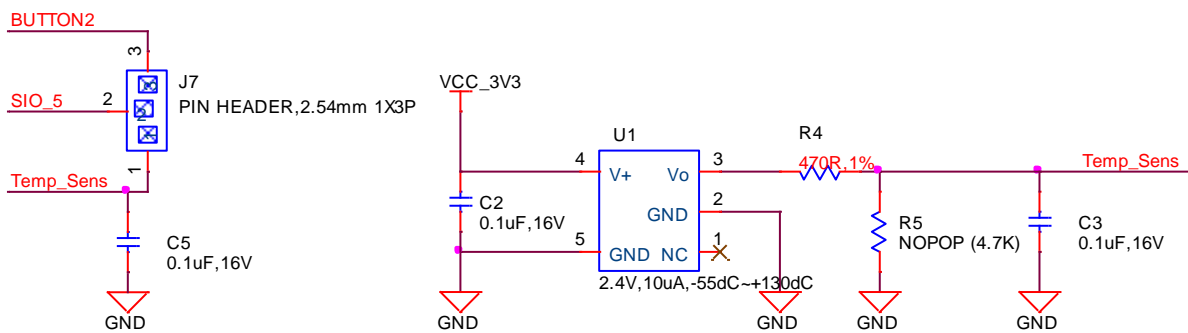


Figure 8: Analog Buffer

### 9.3.2. Temperature Sensor

The temperature sensor (U1) by default is not connected to the RM1xx module, as jumper is fitted to J7 pins 2-3. The temperature sensor (U1) can be connected by moving jumper from J7 pins 2-3 to J7 pins 1-2, bridging TEMP\_SENS and SIO\_5.



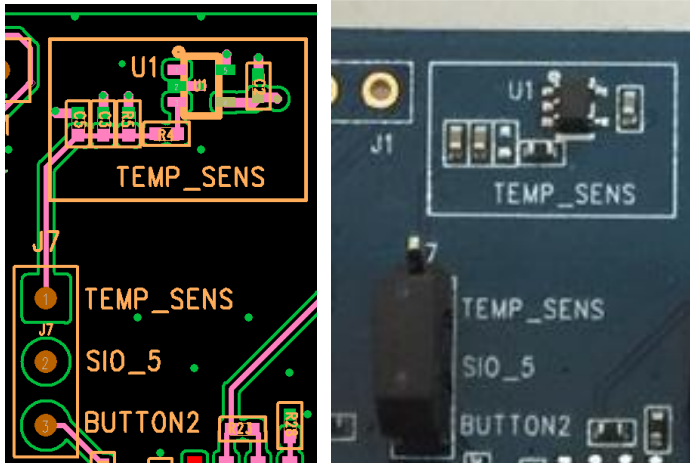


Figure 9: Temperature Sensor

The on-board temperature sensor (TI LM20BIM7 - [www.ti.com/lit/ds/symlink/lm20.pdf](http://www.ti.com/lit/ds/symlink/lm20.pdf)) has an Analogue output that can be connected to RM1xx module pin SIO\_5; but since the LM20BIM7 has an analogue output, the RM1xx module SIO\_5 digital pin (DIO) must be configured as AIN analogue input (ADC). To configure the SIO\_5 pin from DIO pin to Alternate function AIN, see the example file “tempsens.rm1xx.sb” in the RM1xx sample applications library: <https://github.com/LairdCP/RM1xx-Applications>.

Key specifications of the LM20BIM7 are as follows in Table 7.

Table 7: LM20BIM7 Specifications

<b>Output type</b>	Analogue output
<b>Accuracy at 30°C</b>	±1.5°C ±4°C (max)
<b>Accuracy at 40°C to +85°C</b>	approx. ±2.5°C ±5°C (max)
<b>Power supply voltage range</b>	+2.4 V to 5.5 V
<b>Current Drain</b>	10 uA (max)
<b>Output impedance</b>	160 Ohms (max)

The LM20BIM7 datasheet states the relationship of Temperature (T) to Voltage output (Vo) can be approximated as a linear equation (for temperature range of -40°C to +85°C):

$$V_o(\text{mV}) = -11.79\text{mV}/^{\circ}\text{C} \times T + 1858.3$$

gives below calculated Vo versus temperature:

Table 8: LM20BIM7 Temperature to Voltage Output relationship

Temperature (T)	Typical Vo
+80°C	+924.7mV
+70°C	+1041.4mV
+60°C	+1158.1mV
+50°C	+1274.8mV
+40°C	+1391.5mV
+30°C	+1508.2mV
+20°C	+1624.9mV
+10°C	+1741.6mV



Temperature (T)	Typical Vo
+0°C	+1858.2mV
-10°C	+1975.0mV
-20°C	+2091.7mV
-30°C	+2208.4mV

### 9.3.3. I2C sensor (Coulomb Counter)

The I2C Coulomb Counter (U4) senses the current drawn by RM1xx VCC pins via sense resistor R30. By default, it is bypassed with jumper fitted in J11 pins 1-2. To connect the coulomb counter to RM1xx VCC pins (namely VCC\_BLE pin 12 and VCC\_LORA pin 13) you must fit a jumper to J11 pins 2-3. The output of the coulomb counter is on the I2C bus and is by default connected to the RM1xx module via jumpers on J13 pins 2-3 and J14 pins 2-3.

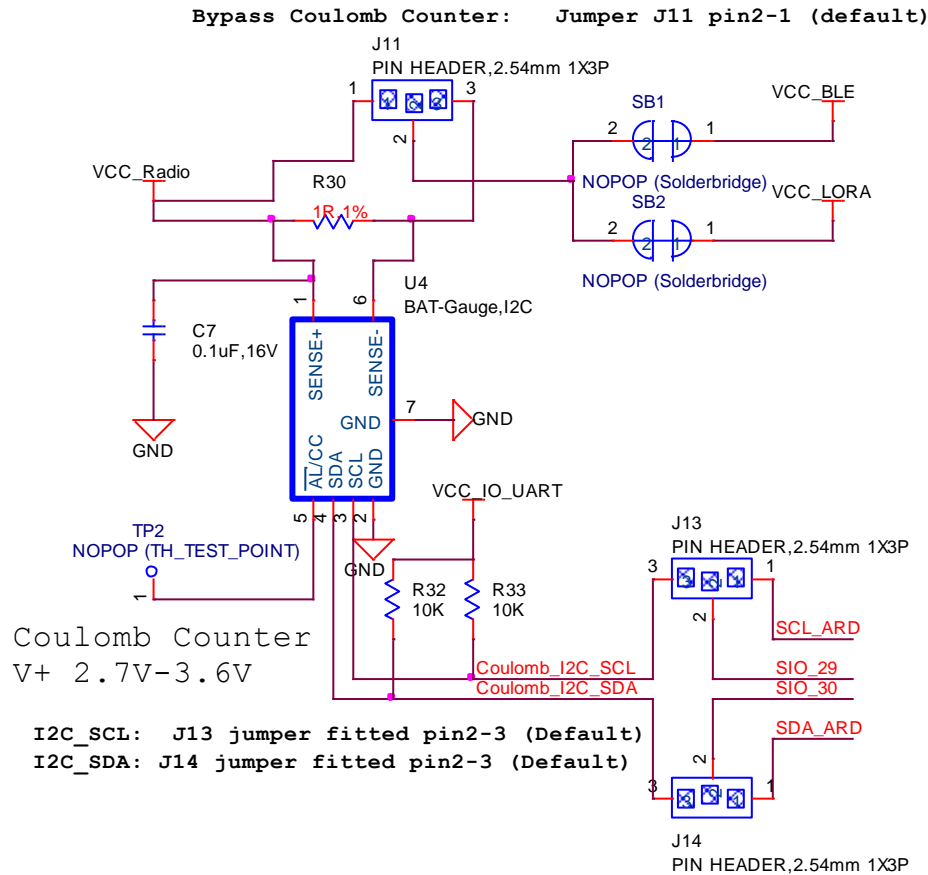


Figure 10: Coulomb counter schematic



Figure 11: I2C sensor - Coulomb Counter

The Coulomb Counter measures current taken over time and outputs this as I2C data. For a working example, see the file *ltc2941.sb* in the *smartBASIC* sample application library: <https://github.com/LairdCP/RM1xx-Applications>.

### 9.3.4. SPI IO Expander and connected Push Button and LEDs on IO expander

The SPI IO Expander (U2) is connected to the RM1xx SPI pins directly. The 3-pin header J18 connects SIO\_4 (SPI SS) to I/O Expander or to Arduino D10 (for use as SPI Slave Select). By default, the RM1xx Module SIO\_4 (used as the SPI SS) is connected to I/O Expander (U2) slave select line via J8 with a jumper fitted on J18 pins 2-3. Table 9 lists signal mappings for all four LEDs and the push button of IO Expander.

Table 9: IO Expander signal mappings

Part	SIO
LED1 (D1)	SPI IO expander (U2)pin9 GP0
LED2 (D2)	SPI IO expander (U2)pin10 GP1
LED3 (D31)	SPI IO expander (U2)pin11 GP2
LED4 (D4)	SPI IO expander (U2)pin12 GP3
Button 1 (SW1)	SPI IO expander (U2)pin13 GP4

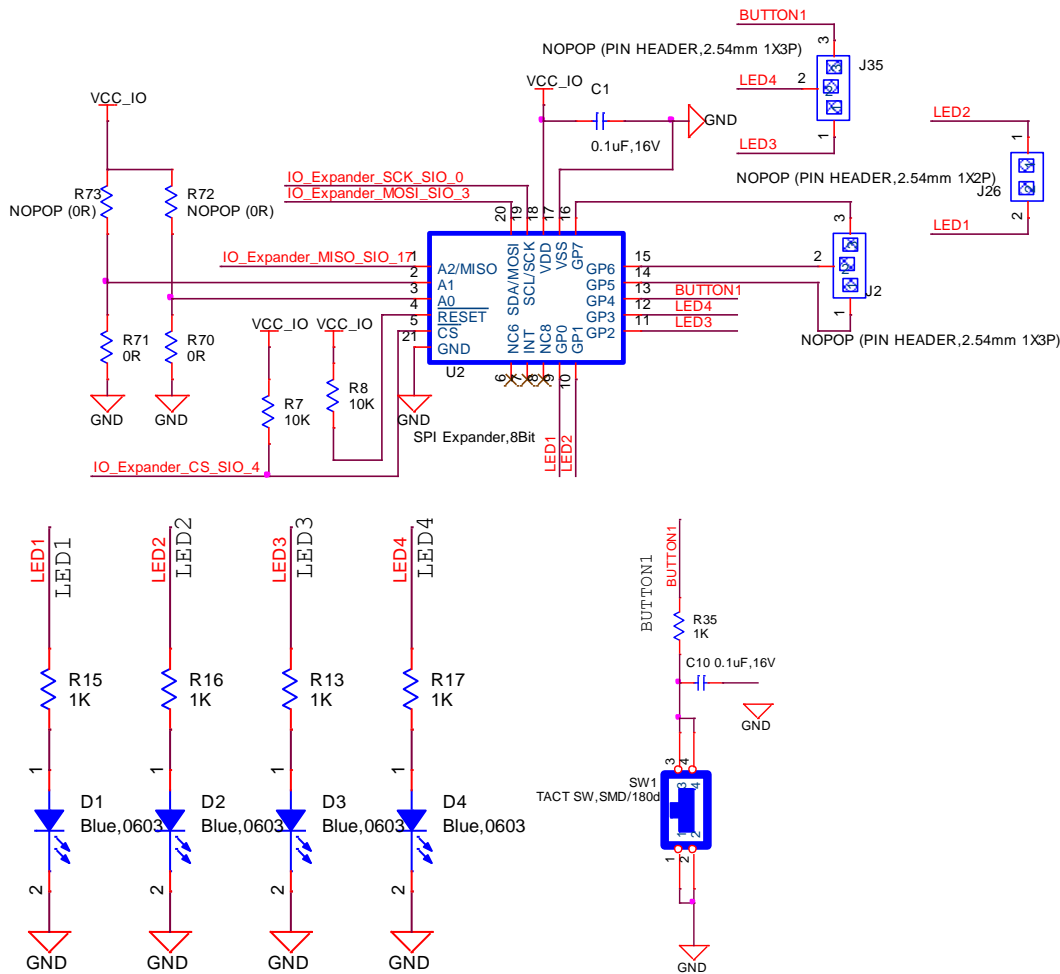


Figure 12: IO Expander schematic



Figure 13: SPI IO Expander and connected Push button and LEDs

For a working example of the IO Expander in use, see the file `ioexpander.rm1xx.sb` in the `smartBASIC` sample applications library: <https://github.com/LairdCP/RM1xx-Applications>

### 9.3.5. Push Button and LED connected to RM1xx

The selection jumper on J7 connects either BUTTON2 or the analog Temp Sensor to SIO\_5. Fit the jumper on J7 to short pins 2-3 to connect BUTTON2 to SIO\_5.

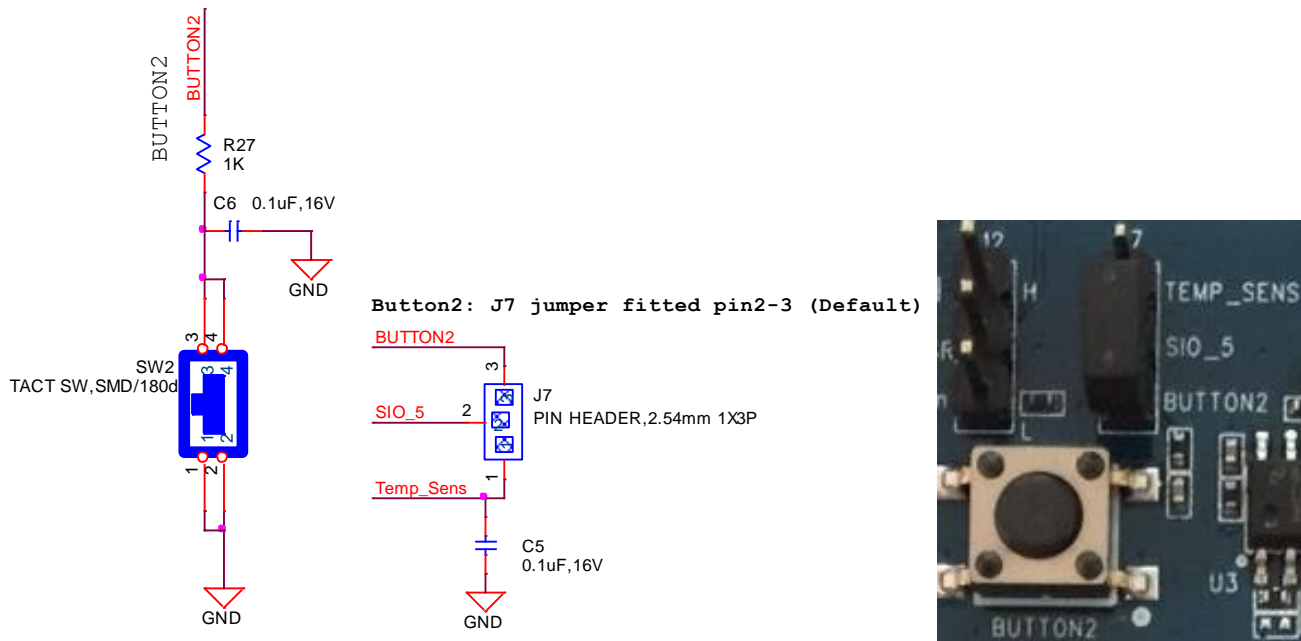


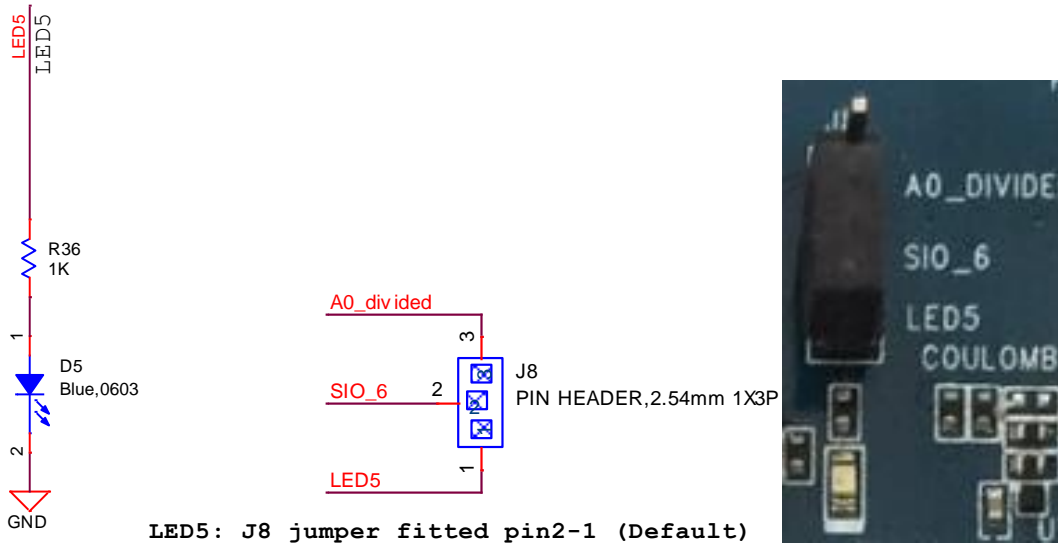
Figure 14: J7 Jumper

The button (BUTTON2) has no external pull-up resistor, so to use the button, the SIO\_5 pin must be configured as an input with internal pull-up resistor, such as a *smartBASIC* line that configures the pull-up:

```
rc = GPIOSETFUNC(5,1,4):          '//sets sio5 (Button2) as a digital in,
strong pull up
```

### 9.3.6. LED and Analog Buffer Connected to RM1xx

The selection jumper on J8 connects either LED5 or the output of the Arduino analog input buffer to SIO\_6. By default, LED5 is selected. To select the Arduino analog input buffer to SIO\_6, fit the jumper on J7 to short pins 2-3.



LED5: J8 jumper fitted pin2-1 (Default)

Figure 15: J8 Jumper

## 10 OTHER FEATURES

### 10.1. Current Consumption Measurement

A removable jumper (on J11) is provided to break the power supply line directly to the module, allowing you to measure current consumption. For normal operation, jumper on J11 between pin1 and pin2 must be fitted (and is fitted by default).

---

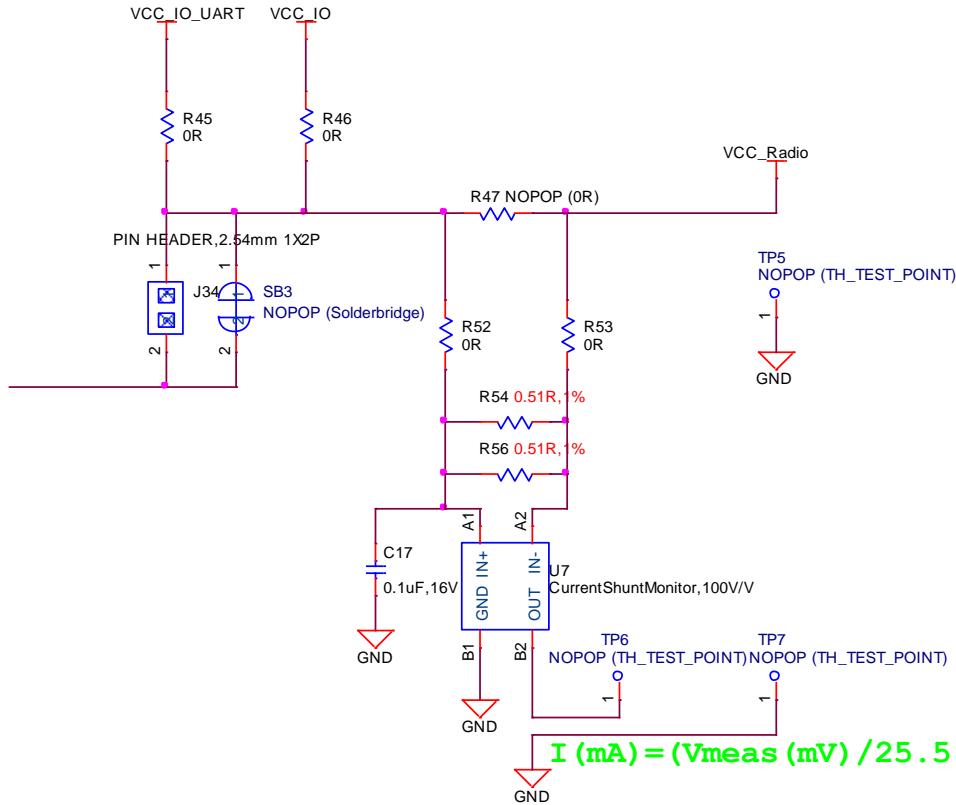
**IMPORTANT:** To achieve the optimal power consumption of the RM1xx series module on the development board, see the “LowPower.SB” file in the *smartBASIC* sample application library on RM1xx product pages at <https://github.com/LairdCP/RM1xx-Applications>.

---

**Note:** This measures the current consumption of the **RM1xx** series module ONLY.

---

The current drawn by the RM1xx series module can be monitored on the development board by bypassing the coulomb counter circuitry. [Figure 16](#) and [Figure 17](#) show the schematic (and location of measuring points on PCB) related to current measurements.



Bypass Coulomb Counter: Jumper J11 pin2-1 (default)

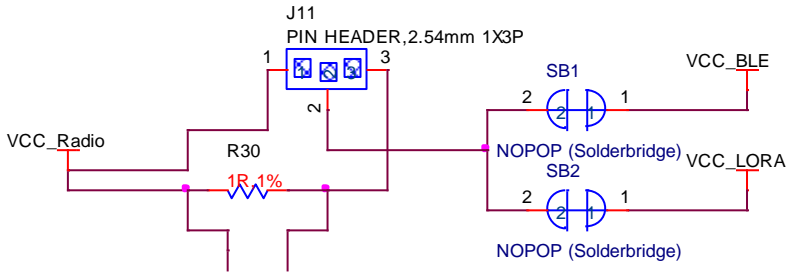


Figure 16: Current measurement schematic

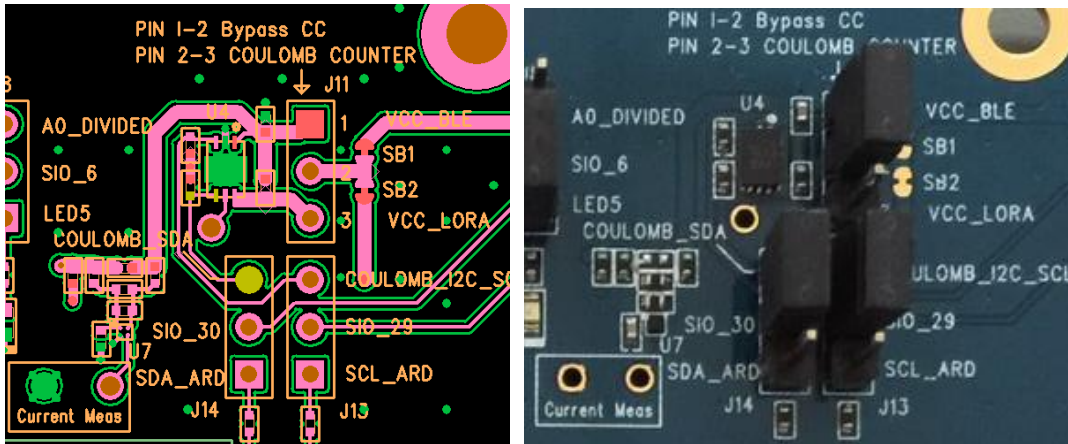


Figure 17: Current measurement design and component location

There are two primary ways to measure the current consumption:

- **Using Ammeter** – Connect an ampere meter between the two pins of J11 pins 1-2. This monitors the current directly.
- **Using Oscilloscope** – Mount a resistor across J11 pins 1-2. The resistor should not be larger than 10 Ohm. Connect an oscilloscope or similar with two probes on the pins on the J11 connector and measure the voltage drop. The voltage drop is proportional with current consumption. If a 1 Ohm resistor is chosen, 1 mV equals 1mA.

There is also a third way to measure current:

- **Using Current Shunt Monitor** – The current drawn by the RM1xx module can be monitored using the Current Shunt Monitor (CSM), INA216 (U7). The gain of INA216 is 100 V/V for lowest possible drop voltage.

---

**Note:** Using the current shunt monitor method allows the dynamic current consumption waveforms on oscilloscope as the RM1xx radio operates. This can provide insight into power optimization.

---

Current consumed by the RM1xx series module is measured as a voltage (that is proportional to the current) using the current shunt monitor (U7). This is performed by connecting a voltmeter or oscilloscope to TP6 and also the ground to TP7. Current in milliamps can be determined from the following equation:

$$I(\text{mA}) = V_{\text{meas\_TP6}}(\text{mV}) / 25.5$$

---

**CAUTION:** Take care not to short TP7 (the Current Shunt Monitor IC (U7)) output to GND, as that will permanently damage the IC U7.

---

## 11 ADDITIONAL DOCUMENTATION

Laird offers a variety of documentation and ancillary information to support our customers through the initial evaluation process and ultimately into mass production. Additional documentation can be accessed from the Documentation tab of the [Laird RM1xx Product Page](#).

- User Guide – *smart*BASIC Core Functionality
- User Guide – *smart*BASIC RM1xx Extensions
- RM1xx – *smart*BASIC sample applications library
- Application Note – Firmware Upgrade Application note
- Application Note – Sample Applications for the RM1xx
- Application Note – Using BLE and LoRa on the RM1xx
- Application Note – Interfacing with the RM186
- Application Note – Interfacing with the RM191
- Application Note – Connecting to a Kerlink Gateway
- Application Note – Connecting to a Multitech Conduit Gateway
- Application Note – Using an I2C GPS Sensor with the RM1xx
- Application Note – Using a LCD Keypad Shield with the RM1xx
- Application Note – Using UwTerminal with the RM1xx
- Application Note – Connecting to the Semtech Website

For any additional questions or queries, or to receive technical support for this Development Kit or for the RM1xx module series, please contact the Embedded Wireless Solutions Support Center: <http://ews-support.lairdtech.com>.

© Copyright 2016 Laird. All Rights Reserved. Patent pending. Any information furnished by Laird and its agents is believed to be accurate and reliable. All specifications are subject to change without notice. Responsibility for the use and application of Laird materials or products rests with the end user since Laird and its agents cannot be aware of all potential uses. Laird makes no warranties as to non-infringement nor as to the fitness, merchantability, or sustainability of any Laird materials or products for any specific or general uses. Laird, Laird Technologies, Inc., or any of its affiliates or agents shall not be liable for incidental or consequential damages of any kind. All Laird products are sold pursuant to the Laird Terms and Conditions of Sale in effect from time to time, a copy of which will be furnished upon request. When used as a tradename herein, *Laird* means Laird PLC or one or more subsidiaries of Laird PLC. Laird™, Laird Technologies™, corresponding logos, and other marks are trademarks or registered trademarks of Laird. Other marks may be the property of third parties. Nothing herein provides a license under any Laird or any third party intellectual property right.