

# User Guide

## BL654 Development Kit

Part # 455-00001 & 455-00002

*Version 0.1*

---

PRELIMINARY

## REVISION HISTORY

Version	Date	Notes	Contributor(s)	Approver
0.1	19 Feb 2018	Initial PRELIMINARY version	Raj Khatri	Jonathan Kaye

PRELIMINARY

## CONTENTS

Overview.....	4
Laird BL654 Development Kit Part Numbers.....	4
Package Contents .....	4
BL654 Development Kit – Main Development Board .....	5
Understanding the Development Board .....	7
Functional Blocks.....	8
Power Supply.....	9
Reset Button.....	12
SWD (JTAG) Interface .....	13
Four-wire UART Serial Interface.....	16
UART Mapping.....	16
nAutoRUN Pin and Operating Modes .....	18
vSP (Virtual Serial Port) Modes and OTA (Over the Air) smart BASIC App Download .....	19
Software .....	21
Breakout Connector Pinouts .....	22
SIO (Special Input/Output Sockets) Breakout Connectors.....	22
Additional Peripherals/Sensors.....	26
Temperature Sensor .....	27
I2C Sensor (RTC Chip) .....	28
SPI Device EEPROM .....	29
Push Button and LED Connected to BL654.....	30
NFC External Antenna Connector and NFC Antenna RF Matching Circuit .....	32
Optional External Serial QSPI Flash IC .....	33
Optional 32.76 kHz Crystal .....	34
Other Features.....	35
Current Consumption Measurement.....	35
Appendix.....	37
Coin Cell Insertion .....	37
Coin Cell Removal.....	38
Additional Documentation .....	38

## OVERVIEW

The Laird BL654 Development Kit provides a platform for rapid wireless connectivity prototyping, providing multiple options for the development of Bluetooth Low Energy (BLE) plus Near Field Communication (NFC) applications.

The Laird BLE development kit is designed to support the rapid development of applications and software for the BL654 series of BLE modules featuring Laird’s innovative event driven programming language – *smartBASIC* or for use with the Nordic SDK for C development. More information regarding this product series including a detailed module user’s guide and *smartBASIC* user guides are available on Laird’s BL654 product page: <http://www.lairdtech.com/products/bl654-ble-thread-nfc-modules>

## LAIRD BL654 DEVELOPMENT KIT PART NUMBERS

Part Number	Product Description
455-00001	Development Kit for 451-00001 module – Integrated antenna
455-00002	Development Kit for the 451-00002 module – External antenna

Applicable to the following BL654 module part numbers:

Part Number	Product Description
451-00001	Bluetooth v5/802.15.4/NFC module – Integrated antenna
451-00002	Bluetooth v5 / 802.15.4 / NFC module – External antenna

## PACKAGE CONTENTS

All kits contain the following items:

<b>Development Board</b>	The development board has the required BL654 module soldered onto it and exposes all available hardware interfaces.
<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 board.</li> <li>▪ DC barrel plug for connection to external power supply (5.5VDC max)</li> <li>▪ 3x AAA battery holder fitted on underside of development board</li> <li>▪ Coin-cell holder (powers module only) fitted on underside of development board</li> </ul>
<b>Two-pin jumpers for pin headers (5)</b>	Five jumpers for 2.54 mm pitch headers used on BL654 development board.
<b>Fly leads (6)</b>	Supplied to allow simple connection of any BL654 module pin (available on Plated Through Holes or headers on J44, J47, J48, J41, J29, J1, J12, J1, J5, J17, J21, J6 and J36).
<b>External BLE dipole antenna</b>	Supplied with Development Kit part # 455-00002 only. External antenna, 2 dBi, FlexPIFA (Laird part #001-0022) with integral RF coaxial cable with 100 mm length and IPEX-4 compatible RF connector.
<b>External NFC antenna</b>	Supplied with Development Kit part # 455-00002 only. Laird NFC flexi-PCB antenna – Part # 0600-00061
<b>Web link card</b>	Provides links to additional information including the BL654 user guide, firmware, terminal utilities, schematics, quick start guides, firmware release notes and more. <b>Note:</b> Sample <i>smartBASIC</i> applications are available to download from the Laird <a href="https://github.com/LairdCP/BL654-Applications">https://github.com/LairdCP/BL654-Applications</a>

## BL654 DEVELOPMENT KIT – MAIN DEVELOPMENT BOARD

This section describes the BL654 development board hardware. The BL654 development board is delivered with the BL654 series module loaded with integrated *smartBASIC* runtime engine firmware but no onboard *smartBASIC*; because of this it starts up in AT command mode by default.

Applications in *smartBASIC* are simple and easy to develop for any BLE application. Sample *smartBASIC* applications scripts are available to download from the Laird GitHub repository on the BL654 product page at <https://github.com/LairdCP/BL654-Applications>. The development board also can be used with Nordic SDK.

The BL654 development board is a universal development tool that highlights the capabilities of the BL654 module. The development kit is supplied in a default configuration which should be suitable for multiple experimentation options. It also offers several header connectors that help isolate on-board sensors and UART from the BL654 module to create different configurations. This allows you to test different operating scenarios.

The board allows the BL654 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>

### Key Features

The BL654 development board has the following features:

- BL654 series module soldered onto the development board
- The following power supply options for powering the development board:
  - USB (micro-USB, type B)
  - External DC supply (2.5-5.5V)
  - AAA batteries (three AAA battery holder fitted on underside of development board)
  - USB (micro-USB, type B) –for direct use of BL654 USB interface as well
- Powering the BL654 module in Normal Voltage mode (OPTION1) via selection switch (SW7). Regulated 3.3V or Regulated 1.8V via selection switch.
- Powering the BL654 module in High Voltage mode (OPTION2) via selection switch (SW7). Regulated 2.5V or 4.5V (from 3xAAA battery (4.5V)) via selection switch. Option to inject external voltage anywhere between 2.5V to 5.5V for the High Voltage mode (via J28).
- Power supply option for coin-cell (CR2032) operation of the BL654 module ONLY (not development board)
- USB to UART bridge (FTDI chip)
- BL654 UART can be interfaced to:
  - USB (PC) using the USB-UART bridge (FTDI chip)
  - External UART source (using IO break-out connectors J1 when the development board is powered from a DC jack or AAA batteries)
  - Atmel MCU by use of an analog switch to route the BL654 UART (for those customers working with Nordic SDK)
- Current measuring options (BL654 module only):
  - Pin header (Ammeter)
  - 10R Series resistor for differential measurement (oscilloscope)
- IO break-out 2.54 mm pitch pin header connectors (plated through-holes) that bring out all interfaces of the BL654 module – UART, SPI, QSPI, I2C, SIO [DIO or AIN (ADCs)], PWM, FREQ, NFC – and allow for plugging in external modules/sensors.
- Pin headers jumpers that allow the on-board sensors (I2C sensor, LEDs, SPI interface, etc.; and the USB UART FTDI bridge) to be disconnected from BL654 module (by removing jumpers).

- Four on-board sensors:
  - Analog output temperature sensor
  - I2C device (RTC chip)
  - SPI device (EEPROM)
  - QSPI device (Flash IC)
- Four buttons and Four LEDs for user interaction
- One reset button (via an analog switch)
- NFC antenna connector on-board development board for use with supplied flexi-PCB NFC antenna
- Optional external 32.768 kHz crystal oscillator and associated load capacitors. **Not required** for operation of the BL654; is disconnected by open solder bridges by default.
- **Optional** external serial (SPI) flash IC. Not required for operation of the BL654; is disconnected by open solder bridges by default.
- **Optional** external serial (QSPI) flash IC. Not required for operation of the BL654; is disconnected by open solder bridges by default.
- Access to BL654 JTAG – also known as Serial Wire Debug (SWD) Interface
- On-board SWD (JTAG) programmer circuitry
- *smartBASIC* runtime engine FW upgrade capability:
  - Via UART (using the FTDI USB-UART)
  - Via SWD (JTAG) using on-board JTAG programmer circuitry on the BL654 Development Kit
- *smartBASIC* application upgrade capability:
  - Via UART (using the FTDI USB-UART)
  - Via OTA (Over-the-Air)

## UNDERSTANDING THE DEVELOPMENT BOARD

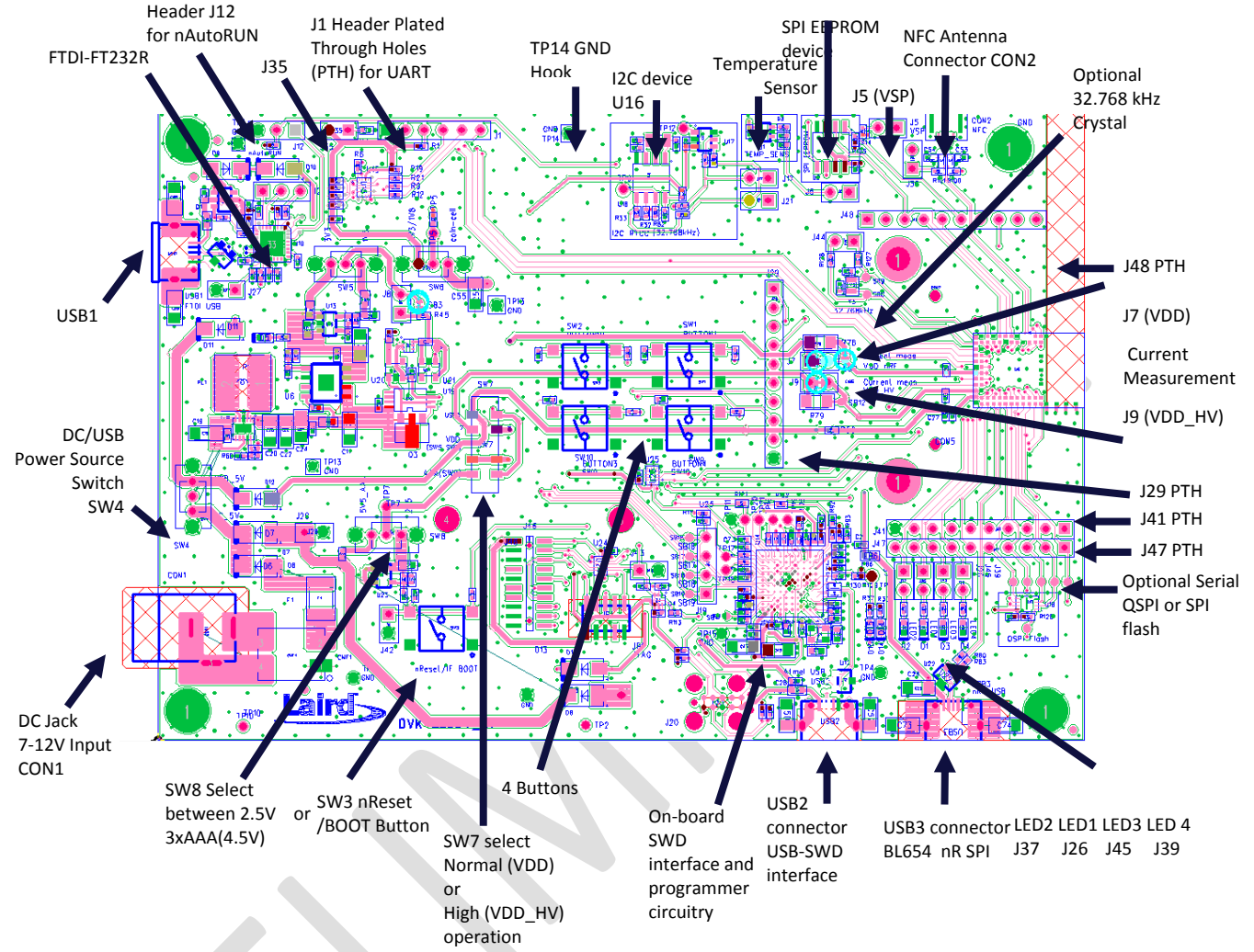


Figure 1: Dev board contents and locations

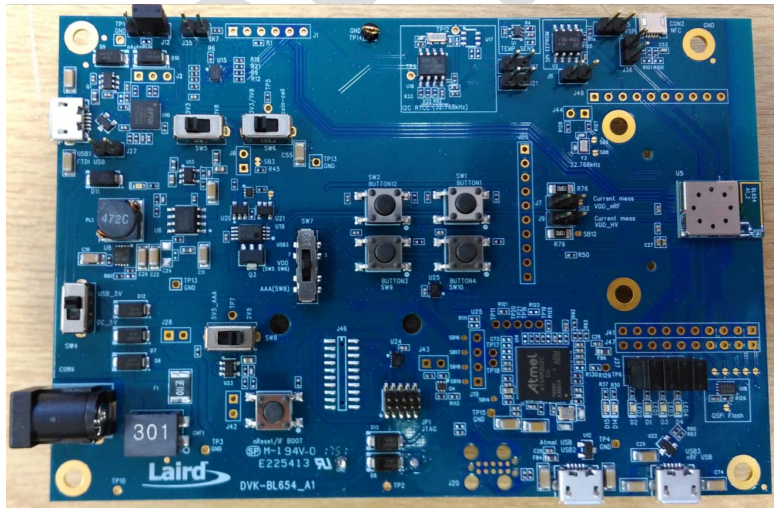


Figure 2: Development board for BL654 (fitted with 451-00001 Integrated Antenna module for example)

## BL654 Default Configuration and Jumper Settings

**Important!** To ensure correct out-of-the-box configuration, the **BL654** development board must be set according to

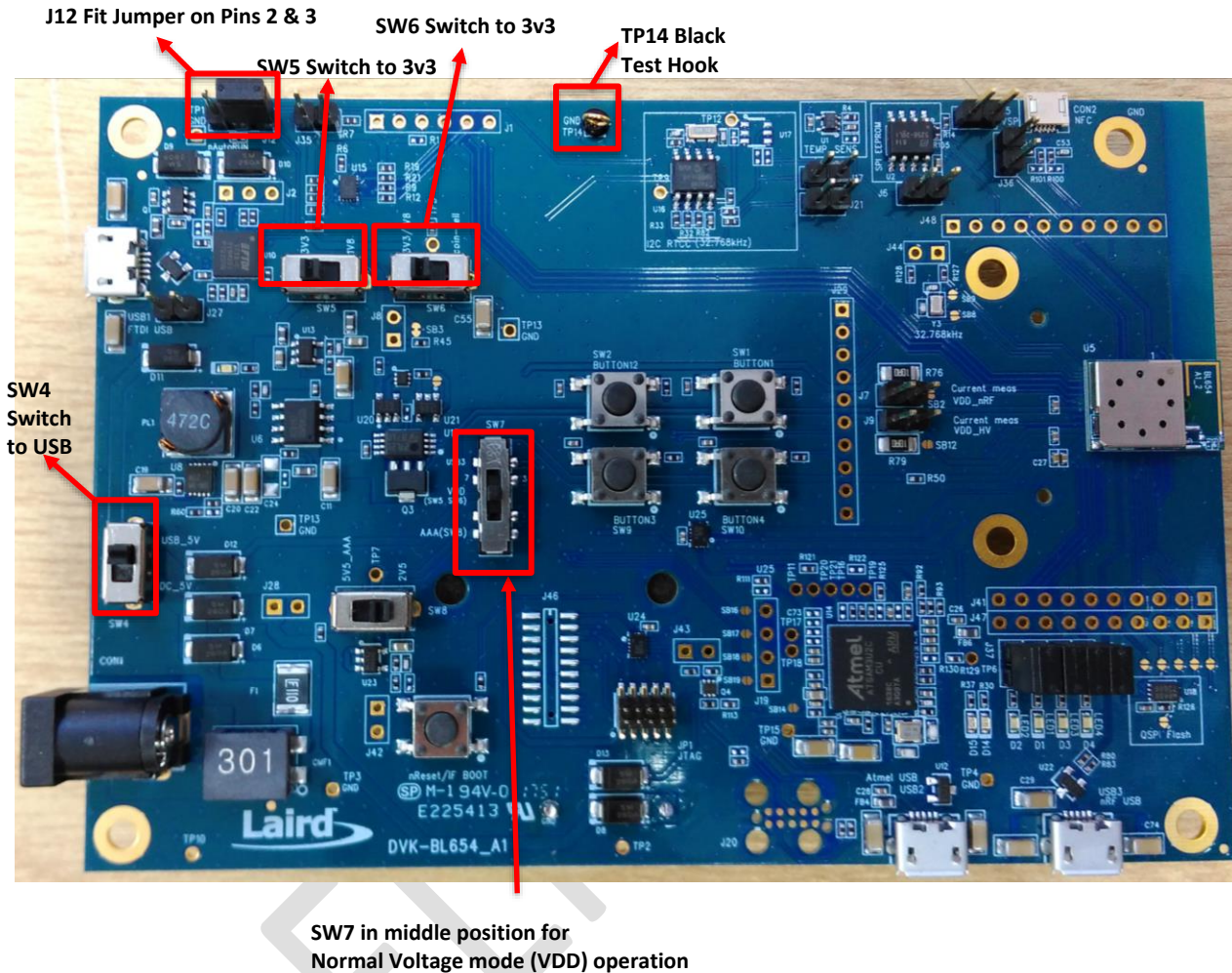


Figure 3: Correct BL654 development board jumper and switch settings

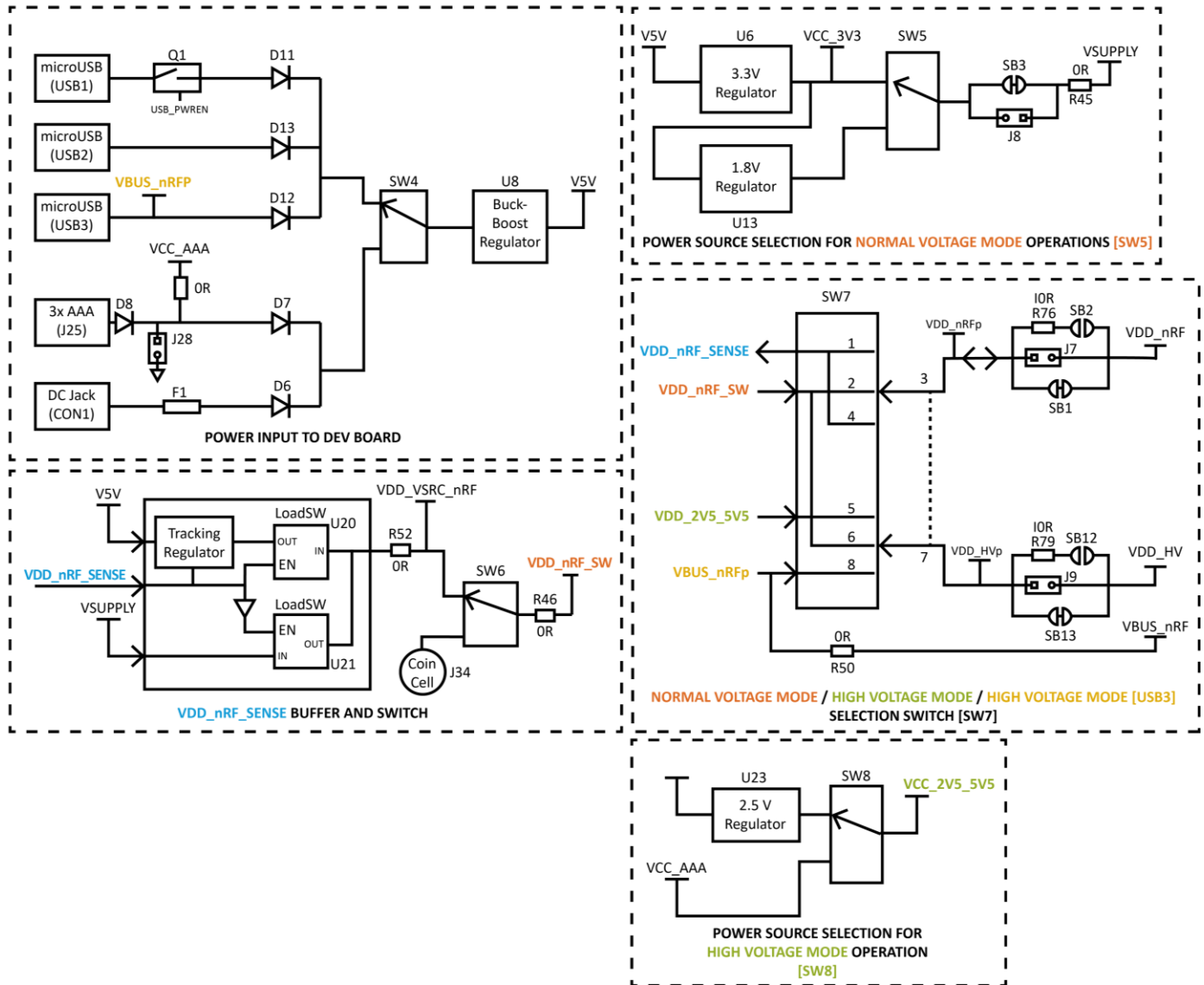
## FUNCTIONAL BLOCKS

TBC



## Power Supply

Figure 4 shows the BL654 development board Power Supply block.



**Figure 4: BL654 development Kit power supply**

There are five options for powering the development board:

- USB1 USB type micro-B connector – To be used if require the FTDI USB1-UART (BL654) path
- USB2 USB type micro-B connector – To be used if require the Atmel USB2-SWD (BL654) path
- USB3 USB type micro-B connector – To be used if require the USB3 to USB (BL654) path
- External DC supply (2.5V-5.5V), into DC jack connector (CON1),
- AAA batteries – Three AAA (4.5V) battery holder (J25) fitted on underside of development board

The external power sources are fed into selection switch SW4 which allows a selection between either USB sources or the DC jack/AAA.

All the external power sources listed above are buck-boost regulated to a fixed 5V on the development board.

The BL654 module has the following power supply pins:

- VDD pin (operating range of 1.7V to 3.6V) – Used for Normal Voltage mode
- VDD\_HV pin (operating range of 2.5V to 5.5V) – Used for High Voltage mode
- VBUS pin (operating range of 4.35V to 5.5V) – Used for BL654 USB mode

It can be powered in the following ways:

- **Normal Voltage mode operation**

**Option 1** – Normal voltage power supply mode entered when the external supply voltage is connected to both the VDD and VDD\_HV pins (so that VDD equals VDD\_HV). Connect external supply within a 1.7V to 3.6V range to the BL654 VDD and VDD\_HV pins.

For Normal Voltage mode operation, the BL654 Development Board power supply section generates the following:

- Regulated 3.3V
- Regulated 1.8V

and then via selection switch SW5 (default is 3.3V position), you can select whether to use 3.3V or 1.8V.

- **High Voltage mode operation**

**Option 2** – High voltage mode power supply mode (using BL654 VDD\_HV pin) entered when the external supply voltage is ONLY connected to the VDD\_HV pin and the VDD pin is not connected to any external voltage supply. Connect external supply within a 2.5V to 5.5V range to the BL654 VDD\_HV pin. Leave the BL654 VDD pin unconnected.

For High Voltage mode operation, the BL654 Development Board power supply section generates the following:

- Regulated 2.5V
- 3 x AAA generated 4.5V (or inject external voltage into J28pin1 up to 5.5V)

and then via selection switch SW8 (default is 2.5V position), you can select whether to use 2.5V or 4.5V.

**Option 3** – High voltage mode with voltage via USB3

- For either option, if you use the BL654 USB interface, the BL654 VBUS pin must be connected to and external supply within the range of 4.35V to 5.5V.

The BL654 development board power supply section is designed to cater to the above and you should follow these two steps:

1. Set SW7 – Select one of the following three positions:
  - High Voltage mode operation and BL654 USB (connect USB cable to USB3 connector) – Top position. Source from USB3.
  - Normal Voltage mode operation – Middle position (default). Source from SW5.
  - High Voltage mode operation – Bottom position (default). Source from SW8.
2. Depending on chosen SW7 position, select one of the following three positions:
  - Plug in USB cable into USB3 – If SW7 is set to Top position.
  - SW5 (either 3.3V or 1.8V) – Default SW5 on 3.3V position. If SW7 set to Middle position.
  - SW8 (either 2.5V or 4.5V (3xAAA)) – Default SW5 on 2.5V position. If SW7 set to Bottom position.

Table 1 summarises the dev-board on-board power sources and switch positions.

Table 1: Dev board power sources and switch positions

Selection Switch SW7 positions (silk screen) Source Voltage Operating mode	Dev Board Power Supply Switch Positions		
	SW5 Selects between 1.8V or 3.3V	SW8 Selects between 2.5V or 4.5V (3xAAA)	Connect USB cable into USB3
Present selected voltage to the BL654 pin			
	BL654 VDD pin	BL654 VDD_HV pin	BL654 VBUS pin
<b>SW7 Top position</b> – Silkscreen: USB3 (Source from USB3) High Voltage Mode with BL654 USB used (USB3)	Note 1	USB3 voltage	USB3 voltage
<b>SW7 Middle position</b> – Silkscreen: SW5 SW6 (Source from SW5, Note 2) Normal Voltage Mode	Decided by SW5 (default SW5 on 3.3V position)	N/A	N/A
<b>SW7 Bottom position</b> – Silkscreen: AAA (SW8) (Source from SW8) High Voltage Mode	Note 1	Decided by SW8 (default SW8 on 2.5V position)	N/A

**Power Source and Switch Location Notes:**

**Note 1:** No voltage is presented to VDD pin, as in High voltage mode, the VDD pin becomes an output voltage pin. It can be used to supply external circuitry from the VDD pin. Before any current can be taken from the BL654 VDD pin, this feature must be enabled in the BL654. Additionally, the VDD output voltage is configurable from 1.8V to 3.3V with possible settings of 1.8V, 2.1V, 2.4V, 2.7V, 3.0V, and 3.3V. The default voltage is 1.8V.

**Note 2:** When SW6 is set to position “coin-cell,” then the voltage selected with SW5 (default position 3V3) does not get presented to the BL654. The CR2032 coin cell (in J34) voltage is not regulated but is fed directly to the BL654 module supply pin. Switch SW6 selects between the regulated 3V3V/1V8 and coin cell. The coin cell powers only the BL654 module directly (on the development board); this is power domain VDD\_nRF\_SW through R46 OR.

**Note:** The development board for BL654 has on-board circuitry to allow access to BL654 SWD interface (via USB connector USB2). Use USB2 only to power the development board when BL654 SWD interface is needed. Refer to [SWD Interface](#). When USB2 is used, USB1 does not need to be used for DC power.

The development board power supply circuitry special feature is it resolves whether the BL654 VDD pin is an input supply pin (in Normal Voltage mode) or becomes an output supply voltage pin (in High Voltage mode).

On the development board, the power circuitry net names are as follows:

- VCC\_3V3 – Supplies regulated 3.3V power to the FTDI chip as well as temperature sensor (U1).
- VSUPPLY – Supplies regulated 3.3V or 1.8V via selection switch SW5 to net VSUPPLY which is connected to input of Load switch U21.
- VCC\_2V5\_5V5 – Selection switch SW8 supplies either regulated 2.5V or 3x AAA battery voltage (4.5V) can be used for when BL654 is powered in High Voltage mode (using the VDD\_HV pin).
- V5V – The main development board power supply’s buck-boosted output (that is 5V) supplies a discrete regulator made up of Q3 and U19. U19 OpAmp drives Q3 to generate regulated voltage (that then is connected to input of load switch U20) that tracks control signal VDD\_nRF\_SENSE.

- VDD\_VSRC\_nRF – Supplies the FTDI chip IO and all other sensors and circuitry. VDD\_VSRC\_nRF is generated from load switches U20 or U19.
- VDD\_nRF\_SENSE – Used as control signal to drive control pin of load switches U20 and U19. The source of VDD\_nRF\_SENSE is the BL654 VDD pin. When BL654 is powered in High Voltage mode (using the VDD\_HV pin), the BL654 VDD pin becomes an output.
- VDD\_nRF\_SW – Selection switch SW6 supplies either VDD\_SRC\_nRF or coin-cell (J34). When the BL654 operated in Normal Voltage mode (SW7 in middle position and voltage source is either 1.8V or 3.3V selected by SW5). Also supplies the I2C RTC chip (U16). The use case for powering this is that the RTC chip can be configured so that, after the pre-determined time, the RTC chip outputs (via RTC\_ALARM pin) a transition level that can be used to wake up the BL654 module up from deep sleep.
- VDD\_nRFp – Supplies the BL654 series module only. Current measuring block on the development board only measures the current into power domain VDD\_nRFp (that is current going into header J7 pin1).
- VDD\_nRF – Supplies the BL654 series module only and is to the current that has come out of the current measuring block on the development board on header connector J7pin2.
- VDD\_HVp – Supplies the BL654 series module only. Current measuring block on the development board only measures the current into power domain VDD\_nRFp (that is current going into header J9 pin1).
- VDD\_HV – Supplies the BL654 series module only and is to the current that has come out of the current measuring block on the development board on header connector J9pin2.
- VBUS\_nRFp – This voltage from USB cable plugged into connector USB3, that is directly fed to BL654 VBUS pin (via 0R resistor R50) on net VBUS\_nRF.

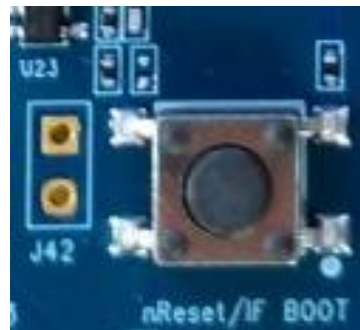
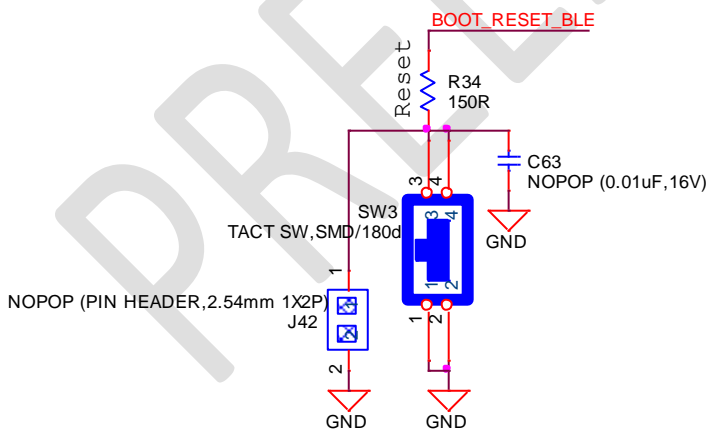
**Additional Power Option (for BL654 module only –not development board) – Coin Cell**

The coin cell powers **only** the BL654 module directly via SW6 (on the development board – power domain VDD\_nRF\_SW) and through R46 provides power to the I2C RTC chip (U16).

Refer to the [Appendix](#) for the correct method of coin cell insertion and removal.

**Reset Button**

The development board has a reset button (SW3). The nReset is active low (SW3 pushed down) that is routed to the BL654 module nReset pin via an analog switch U25. The placement of the Reset button is shown in [Figure 5](#).



**Figure 5: Reset button placement**

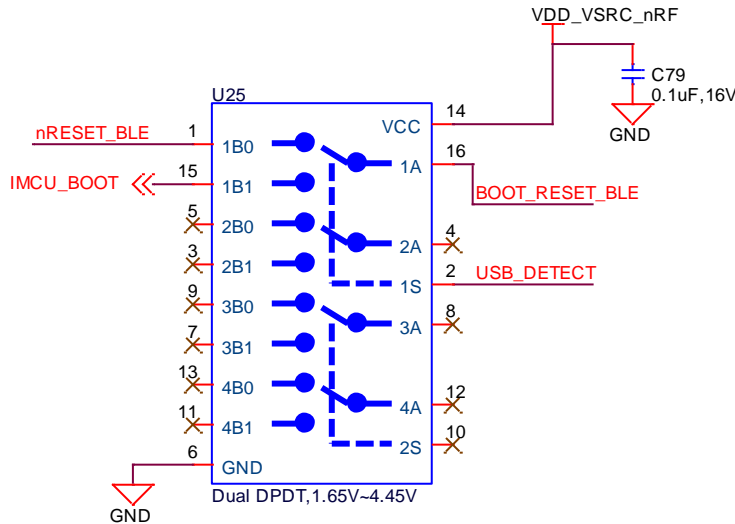


Figure 6: Reset button schematic and routing through analog switch and location diagram

By default, the analog switch (U25) control line (USB\_DETECT) is low and therefore nReset button (SW3) is routed via the analog switch (U25) to BL654 nReset pin (nRESET\_BLE).

The development board has USB2 cable detection circuit that generates the USB\_DETECT signal on its output, so if a USB cable is plugged into connector USB2, then USB2 cable detection circuit generates a HIGH on USB\_DETECT and routes nReset button (SW3) to the Atmel MCU. This is useful for those customers working with the Nordic SDK.

Table 3 displays the routing of SW3 reset button via analog switch U25 to either the BL654 reset pin or Atmel MCU depending on state of the USB\_DETECT control line (which is HIGH on USB cable being plugged into USB2).

Table 2: USB U4 USB-SWD to BL654 SWD signal routing connections

SW3 reset button into Analog switch U25 (net name)	USB cable plugged into USB2 (USB_DETECT HIGH) Route SW3 reset button to Atmel MCU (U4) via analog switch U25 Net name	No USB cable plugged into USB2 (USB_DETECT LOW) Route SW3 reset button to BL654 reset pin to JP1 via analog switch U25	Comments
BOOT_RESET_BLE	IMCU_BOOTBLE (pin 6)	nRESET_BLE	

## SWD (JTAG) Interface

The development board provides access to the BL654 module two-wire SWD interface on JP1 via analog switch U24. This is REQUIRED for customer use, since the BL654 module supports *smartBASIC* runtime engine firmware over JTAG (as well as over UART).

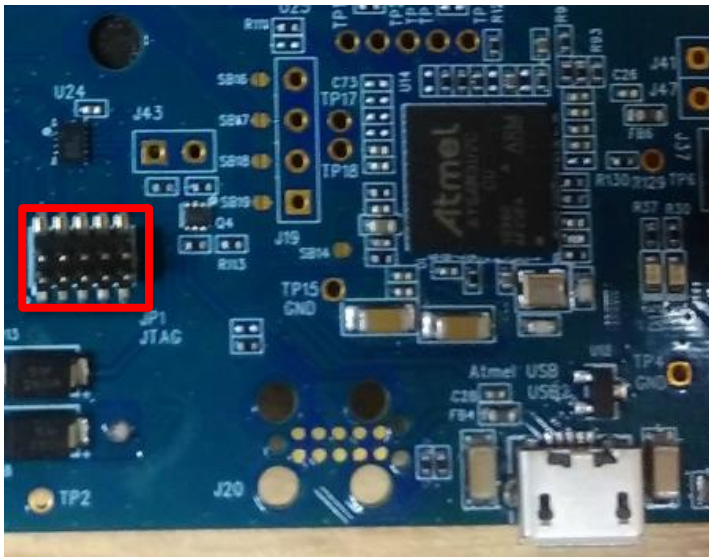
Laird recommends you use JTAG (2-wire interface) to handle future BL654 module firmware upgrades. You MUST wire out the JTAG (2-wire interface) on your host design (four lines should be wired out, namely SWDIO, SWDCLK, GND and VCC). Firmware upgrades can still be performed over the BL654 UART interface, but this is slower (60 seconds using UART vs. 10 seconds when using JTAG) than using the BL654 JTAG (2-wire interface).

Upgrading *smartBASIC* runtime engine firmware or loading *smartBASIC* applications also can be done using the UART interface.

For those customers (using Nordic SDK) that require access to BL654 SWD (JTAG) interface, the BL654 development board (see [Figure 1](#)) has on-board circuitry to allow access to BL654 module SWD interface (via USB connector USB2).

[Figure 8](#) shows the SWD on-board circuitry routing via analog switch U24. When USB cable is plugged into connector USB2 (then the USB cable detection output generates a HIGH for USB\_DETECT) and Atmel MCU SWD (JTAG) signals are routed to the BL654 SWD interface. This is required to connect the two-wire SWD (JTAG) interface from U14 to the BL654 SWD (JTAG) interface.

When no USB cable is plugged into connector USB2, the USB cable detection output generates a LOW for USB\_DETECT and the Atmel MCU SWD (JTAG) signals are routed to connector JP1 so an external BL654 module can be programmed over the SWD interface.



**Figure 7: JP1 on development board**

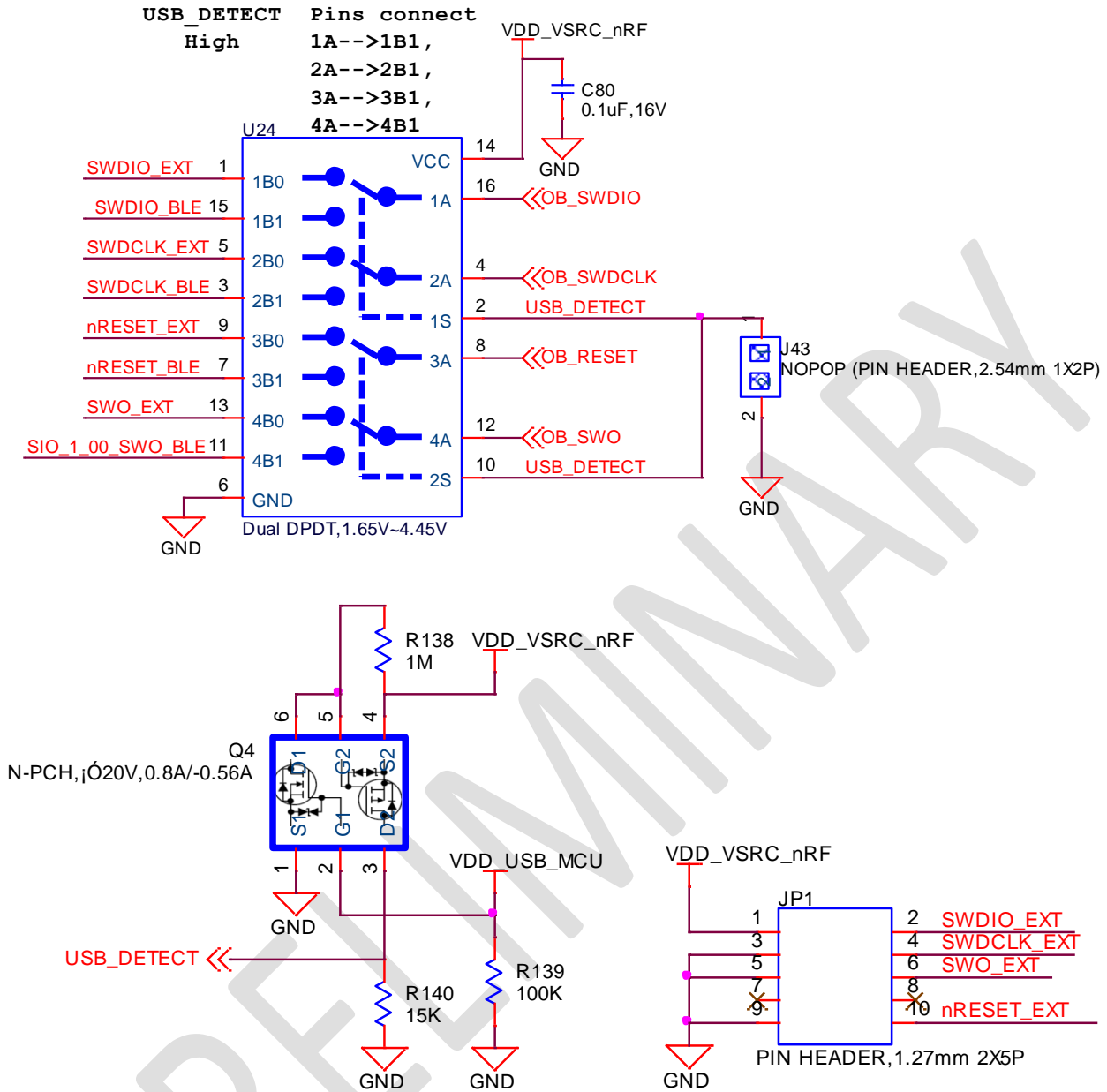


Figure 8: USB to SWD onboard circuitry routing via analog switch (U24)

Table 3 displays the four signals running from Atmel MCU U14 (SWD interface plus SIO\_32 and nReset\_BLE) to the BL654 module (SWD interface plus SIO\_32 and nReset\_BLE).

Table 3: USB U4 USB-SWD to BL654 SWD signal routing connections

U4 (Atmel MCU) Net SWD Interface into Analog switch U24	USB cable plugged into USB2 (USB_DETECT HIGH) Route SWD Interface from Atmel MCU (U4) to BL654 Module Net Name	No USB cable plugged into USB2 (USB_DETECT LOW) Route SWD Interface from Atmel MCU (U4) to JP1	Comments
OB_SWDCCLK	SWDCCLK_BLE (pin 3)	SWDCCLK_EXT (JP1 pin4)	
OB_SWDIO	SWDIO_BLE (pin 1)	SWDIO_EXT (JP1 pin2)	
OB_RESET	nRESET_BLE (pin 19)	nRESET_EXT (JP1 pin10)	
OB_SWO	SIO_32 (pin 7)	SWO_EXT (JP1 pin6)	

SIO\_32 is a Trace output (called SWO, Serial Wire Output) and is not necessary for programming BL654 over the SWD interface.

nReset\_BLE is not necessary for programming BL654 over the SWD interface.

### Four-wire UART Serial Interface

The development board provides access to the BL654 module four-wire UART interface (TX, RX, CTS, RTS) either through USB (via UT10 FTDI USB-UART convertor chip) or through a breakout header connector J1.

**Note:** The BL654 module provides four-wire UART interface on the HW and the other four signals (DTR, DSR, DCD, RI), which are low bandwidth signals, can be implemented in a *smartBASIC* application using any spare digital SIO pins.

### UART Mapping

The UART connection on the BL654 series module and the FTDI IC are shown in Table 4. Figure 9 explains how the BL654 series module UART is mapped to the breakout header connector J1. These connections are listed in Table 4.

Table 4: SIO/UART connections

BL654 (U5) SIO	BL654 Default Function	FTDI IC UART
SIO_06 (U5 pin35)	UART_TX (output)	USB_RX
SIO_08 (U5 pin29)	UART_RX (input)	USB_TX
SIO_05 (U5 pin39)	UART_RTS (output)	USB_CTS
SIO_07 (U5 pin37)	UART_CTS (input)	USB_RTS

**Note:** Additionally, SIO\_35 (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. nAutorun can be controlled directly from Laird’s UWTerminalX using the DTR tick box.



### 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 converter device (FT232R) when SW4 is set to the 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 the following website: <http://www.ftdichip.com/Products/FT232R.htm>.
- **UART Interface Driven by USB FTDI Chip:** In normal operation, the BL654 UART interface is driven by the FTDI FT232R USB-to-UART converter.

### UART Interface Driven by External Source

- **UART Interface Driven by External UART Source:** The BL654 module UART interface (TX, RX, CTS, RTS) is presented at a 2.54 mm (0.1”) pitch header (J1). To allow the BL654 UART interface to be driven from the breakout header connector (J1), the following must be configured:
  - The development board must be powered from a DC jack (CON1) or AAA batteries (J25) and with switch SW4 in DC position.
  - The FTDI device must be held in reset. This is achieved automatically by removal of the USB cable (from connector USB1), placing SW4 in the DC position or fitting a jumper on J27.
  - Fit a jumper on J35 (to switch the Analog switch U15 and route BL654 UART to J1) when connecting an external UART source (for example FTDI USB-UART TTL (3.3V) converter cable) using J1. This isolates the BL654 UART from the on-board USB-UART FTDI device. By default, the jumper on J35 is not fitted, so by default BL654 UART is routed to U10 FTDI FT232R USB –UART converter.

**Note:** The BL654 UART signal levels always need to match the supply voltage net VDD\_VSRC\_nRF, of the BL654.

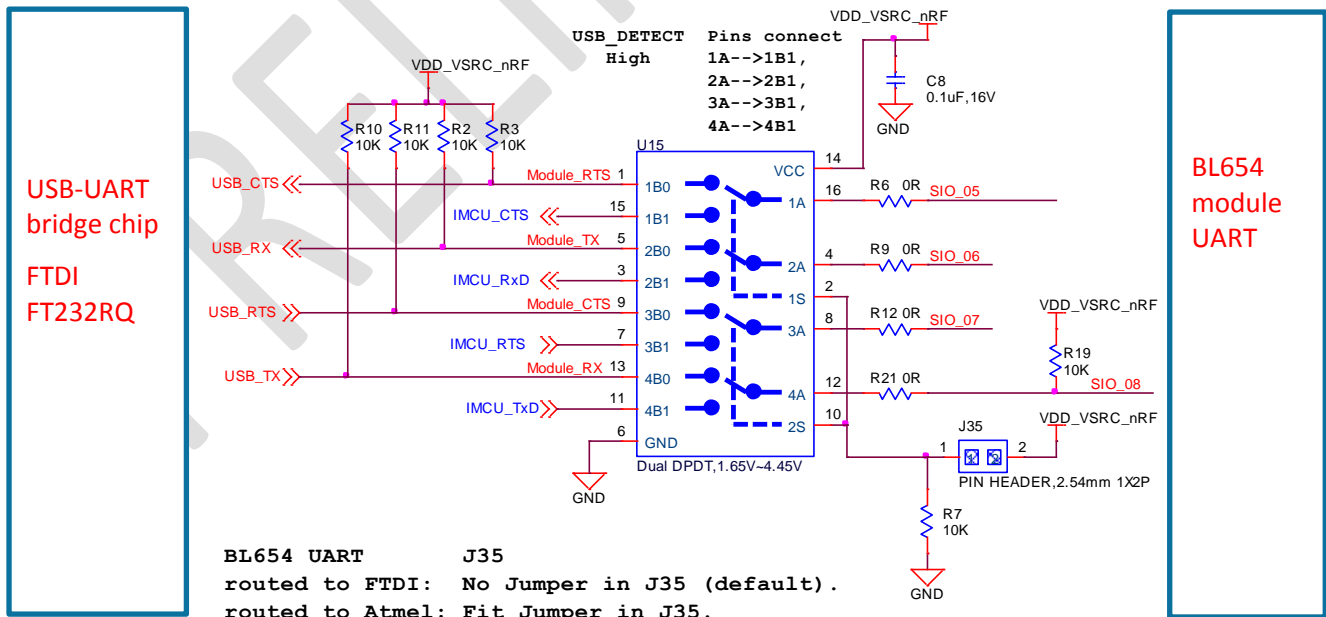
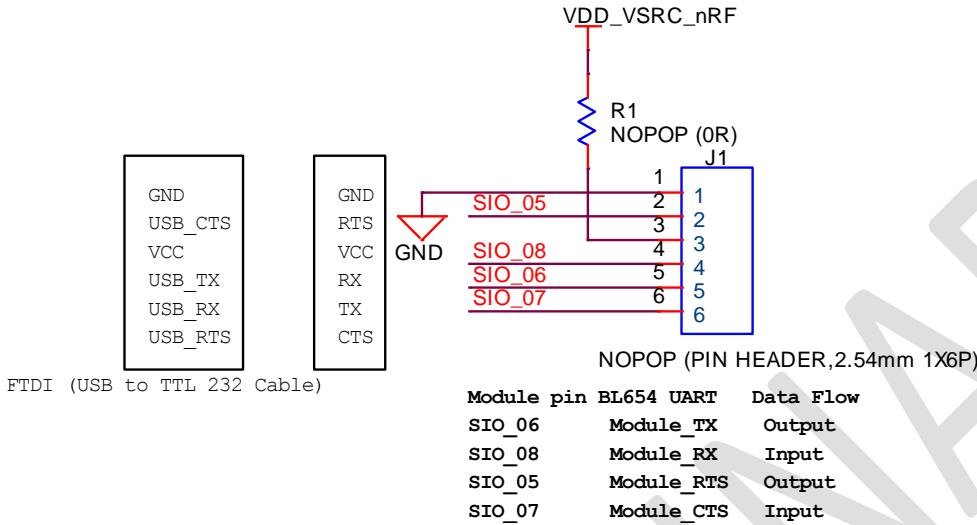


Figure 9: USB to UART (via FTDI chip on devboard) interface via analog switch U15

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

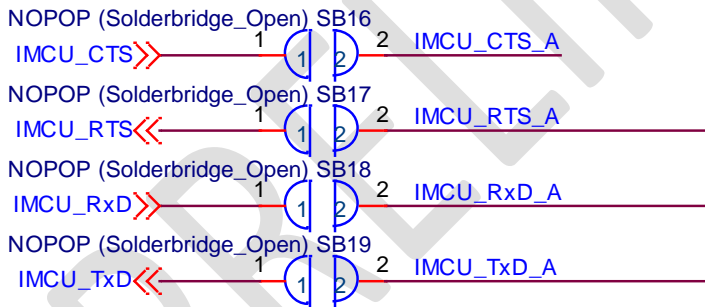
If the BL654 on the development board is powered from 1.8V supply, then you must use the 1.8V version of the FTDI USB-UART cable. UART signal levels always need to match the supply voltage net VDD\_nRF\_SW of the BL654 Development Board.



**Figure 10: J1 wiring to match FTDI USB-UART cable (TTL-232R-3V3 cable)**

Fit a jumper in J35 (to switch the Analog switch U15 and route BL654 UART to J1) when connecting an external FTDI USB-UART TTL (3.3V) converter cable using J1.

Fitting a jumper in J35 also allows the BL654 UART to be routed to Atmel MCU UART (signal also on J19 and net names beginning with IMCU\_) via open solder bridges SB16 to SB19 shown in Figure 11. These bridges must be connected with solder. This may be useful for those customers wanting to work with the Nordic SDK.



**Figure 11: Open solder-bridges on the UART interface running from Atmel MCU (U4) to analog switch U15 (to BL654 ultimately)**

## nAutoRUN Pin and Operating Modes


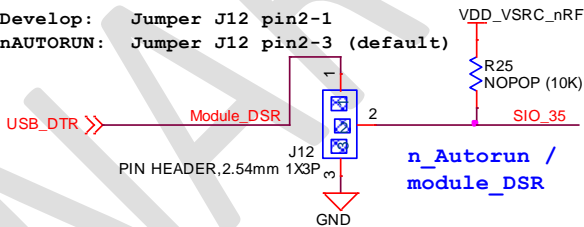
On the development board, the USB\_DTR output (FTDI chip U10) from the PC is wired to BL654 module pin SIO\_35 (pin 5) which is the nAutoRUN pin.

**Note:** *smartBASIC* runtime engine FW checks for the status of nAutoRUN during power-up or reset. The nAutoRUN pin detects if the BL654 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 module’s file system, then the *smartBASIC*

runtime engine FW executes the *smartBASIC* application script automatically; hence the name *Self-contained Run mode*.

Tying nAutoRUN HIGH (to net name on devboard VDD\_VSRC\_nRF) inhibits the \$autorun\$ application from running. As an alternative to using USB\_DTR, the J12 three-pin header allows a jumper to be fitted to select between the two operating modes.

**Table 5: BL654 nAutoRUN header**

nAutoRUN Pin	BL654 Operating Mode (pin28, nAutoRUN Mode/SIO_35)		
	Interactive/ Development Mode (SIO_35 set High Externally)	Self-contained Run Mode (nAutoRUN mode) (SIO_35 Low Internally)	Circuit
<b>J12 Jumper Position</b> 	<b>Develop</b> Jumper on J12 pins 2-1	<b>nAutoRUN (default)</b> Jumper on J12 pins 2-3 BL654 has internal pull-down enabled, jumper in J12 in 2-3 can also be left off	

The J12 header connector allows the USB\_DTR signal from the FTDI chip to be disconnected from the BL654.

To connect the BL654 nAutoRUN pin SIO\_35 (pin 5) to PC FTDI USB\_DTR line via the J12 header connector, do the following:

- Fit the jumper into the J12 (pin 2-1) header connector to allow the PC (using UwTerminal) to control nAutoRUN pin (SIO\_35).

To disconnect the BL654 nAutoRUN SIO\_35 (pin 5) from the PC FTDI USB\_DTR line, do the following:

- Remove the jumper on header connector J12 pin 2-1. Then nAutoRUN can be controlled by inserting the jumper onto J12 (pin 2-3) as shown in Table 5 (this is the default). The BL654 by default has pull-down enabled on the SIO\_35 (nAutoRUN) pin, so the jumper into J12 (pin 2-3) is optional.

### vSP (Virtual Serial Port) Modes and OTA (Over the Air) *smartBASIC* App Download

The OTA feature makes it possible to download *smartBASIC* applications over the air to the BL654. To enable this feature, SIO\_02 must be pulled high externally.

On the development board, header connector J5-pin1 brings out the BL654 SIO\_02; J5-pin 2 brings out VCC\_nRF\_SW. To pull BL654 SIO\_02 high (to net name VCC\_nRF\_SW on devboard), fit jumper into header J5.

**Note:** When SIO\_02 is high, ensure that SIO\_35 (nAutoRun) is NOT high at same time, otherwise you cannot load the *smartBASIC* application script.

This section discusses VSP Command mode through pulling SIO\_02 high and nAutoRUN (SIO\_35) low. Refer to the documentation tab of the **BL654 product page**:

<http://www.lairdtech.com/products/bl654-ble-thread-nfc-modules>.

Figure 12 shows the difference between VSP Bridge to UART mode and VSP Command mode and how SIO\_02 and nAutoRUN (SIO\_35) must be configured to select between these two modes.

- **VSP Bridge to UART mode** takes data sent from phone or tablet (over BLE) and sends to BL654 to be sent out of the BL654 UART (therefore data not stored on BL654).
- **VSP Command mode** takes data sent from phone or tablet and sends to BL654 which will interpret as an AT command and response will be sent back. The OTA Android or iOS application can be used to download any *smartBASIC* application script over the air to the BL654, since a *smartBASIC* application is downloaded using AT commands.

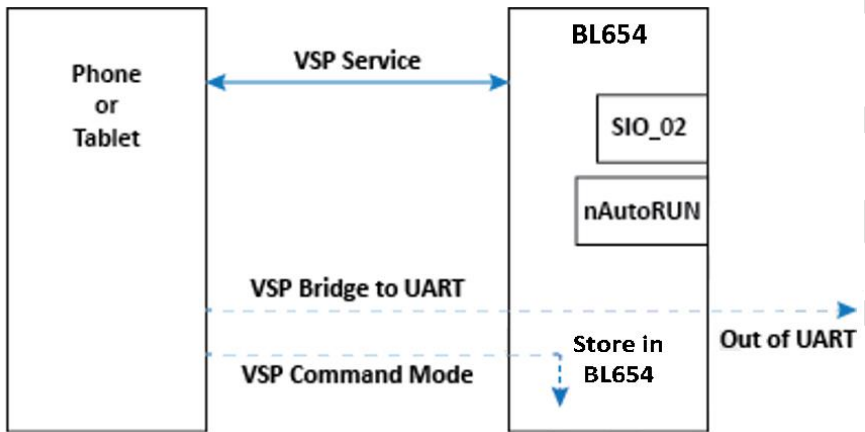


Figure 12: Differences between VSP bridge to UART mode and VSP Command mode

Table 6: vSP modes

Mode	SIO_02 and Jumper position J5	nAutoRUN (SIO_35) and Jumper position J12
VSP Bridge to UART mode	High by fitting jumper in J5	High by fitting jumper in J12 pin 2-1 and untick DTR box in UwTerminalX (the DTR box is ticked by default in UwTerminalX)
VSP Command mode	High by fitting jumper in J5	Low by fitting jumper in J12 pin 2-3

SIO\_02 High (externally) selects the VSP service. When SIO\_02 is High and nAutoRUN (SIO\_35) is Low (externally), this selects VSP Command mode. When SIO\_02 is High and nAutoRUN is High (externally), this selects VSP Bridge to UART mode.

When SIO\_02 on module is set HIGH (externally), VSP is enabled and auto-bridged to UART when connected. However, for VSP Command mode, auto-bridge to UART is not required. With SIO\_02 set to High and nAutoRUN (SIO\_35) set to Low, the device enters VSP Command mode and you can then download the *smartBASIC* application onto the module over the air from the phone (or tablet).

## SOFTWARE

The development board connects the BL654 module to a virtual COM port of a PC or other device. From a PC, you can communicate with the module using [Laird's UwTerminalX](#) (cross platform software available for Windows, Mac, and Linux). This utility allows connections to serial devices using any combination of the communications parameters listed in [Table 7](#).

**Table 7: UwTerminalX communication parameters for BL654**

<b>Port (Windows)</b>	1 to 255
<b>Port (Mac/Linux)</b>	Any/dev/tty device
<b>Baud Rate</b>	1200 to 1000000 <b>Note:</b> Baud rate default is 115200 for BL654.
<b>Parity</b>	None
<b>Data Bits</b>	8
<b>Stop Bits</b>	1
<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 ExpressCard – RS232 card.

The benefits of using UwTerminalX include the following:

- 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 UART BREAK signals. Following provides explanation UART Break.  
([https://en.wikipedia.org/wiki/Universal\\_asynchronous\\_receiver/transmitter#Break\\_condition](https://en.wikipedia.org/wiki/Universal_asynchronous_receiver/transmitter#Break_condition))
- Additional built-in features (right click in Terminal tab screen) to accelerate development including Automation and various XCompile/Load/Run options for downloading *smartBASIC* applications into the BL654.

**Note:** Full details on *smartBASIC* are available in the *smartBASIC* User Guide available at the Laird product page for BL654, along with a document giving a basic introduction to UwTerminalX. A help file is included with UwTerminalX that gives an overview of the program. Visit the BL654 product page at <http://www.lairdtech.com/products/bl654-ble-thread-nfc-modules>.

**Tip:** If the module returns a four-hex digit error code: In UwTerminalX, select those four digits, right-click, and select **Lookup Selected Error-Code (Hex)**. A description of the error is then printed on screen.

## BREAKOUT CONNECTOR PINOUTS

### SIO (Special Input/Output Sockets) Breakout Connectors

Access to all 48 BL654 series module signal pins (SIO's = Signal Input /Output) is available on Plated-Through Holes (for 2.54 mm pitch header connectors) on J44, J47, J48, J41, J29, J1, J12, J1, J5, J17, J21, J6 and J36.

**Note:** The BL654 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, QSPI, DIO (on fixed pins), PWM, FREQ, and NFC
- DIO or AIN functionality is selected using the GpioSetFunc() function in *smartBASIC*
- I2C, UART, SPI, QSPI controlled by xxxOPEN() functions in *smartBASIC*
- SIO\_05 to SIO\_08 are DIO by default when \$autorun\$ app runs on power up
- SIO\_09 and SIO\_10 are NFC pins by default; they can be set to alternative function SIO using the GpioSetFunc() function in *smartBASIC*

These breakout connectors can interface to a wide array of sensors, the BL654 is user configurable through the *smartBASIC* application script to change each SIO pin from the default function (DIO, UART) to alternate functions (AIN (ADC), I2C, SPI, QSPI, DIO), PWM, FREQ, and NFC. The BL654 development kit incorporates additional fly-lead cables inside the box to enable simple, hassle-free testing of these multiple interfaces.

Table 8 shows the BL654 module pins that are brought out to plated through Holes (suitable for 2.54 mm pitch headers).

**Table 8: Module pins exposed by plated through holes**

Plated Through Holes or Header Connector	BL654 Module Signals Exposed
<p><b>J44</b></p> <p>NOPOP (PIN HEADER, 2.54mm 1X2P) J44</p> <p>The diagram shows two signal paths from the module to headers. The top path goes from pin 1 of SB9 (SIO_00) through resistor R127 to SIO_00_Header. The bottom path goes from pin 1 of SB8 (SIO_01) through resistor R128 to SIO_01_Header. A crystal Y3 (32.768KHz, 20ppm, 9pF) is connected to pins 2 of SB8 and SB9. Capacitors C70 (10pF, 50V) and C71 (12pF, 50V) are connected to ground and the crystal pins.</p>	<p>BL654 pin plated holes for access SIO_00 SIO_01</p> <p>J44 connects to SIO_00 and SIO_01 via OR resistors R127 and R128. By default, the optional external 32.768 kHz crystal circuit is not connected to BL654 as SB8 and SB9 are open.</p>

Plated Through Holes or Header Connector		BL654 Module Signals Exposed
<b>J47</b>	<p>NOPOP (PIN HEADER, 2.54mm 1X10P)</p>	<p>BL654 pin plated holes for access:</p> <p>SIO_23 SIO_21 SIO_22 SIO_19 SIO_20 SIO_17 SIO_16 SIO_15 SIO_13 SIO_14</p>
<b>J26</b>		<p>J26 Connects SIO_35 to LED1 J26 jumper fitted (default).</p>
<b>J37</b>	<p>PIN HEADER, 2.54mm 1X2P</p>	<p>J37 Connects SIO_14 to LED2 J37 jumper fitted (default).</p>
<b>J45</b>	<p>PIN HEADER, 2.54mm 1X2P</p>	<p>J45 Connects SIO_15 to LED3 J45 jumper fitted (default).</p>
<b>J39</b>	<p>PIN HEADER, 2.54mm 1X2P</p>	<p>J39 Connects SIO_16 to LED4 J39 jumper fitted (default).</p>
<b>J48</b>	<p>NOPOP (PIN HEADER, 2.54mm 1X10P)</p>	<p>BL654 pin plated holes for access</p> <p>SIO_28 SIO_29 SIO_46 (connects to Eeprom_MISO) SIO_03 SIO_47 (connects to Eeprom_SCK) SIO_44 (connects to Eeprom_CS) SIO_43 SIO_45 (connects to Eeprom_MOSI) SIO_42 SIO_39</p>

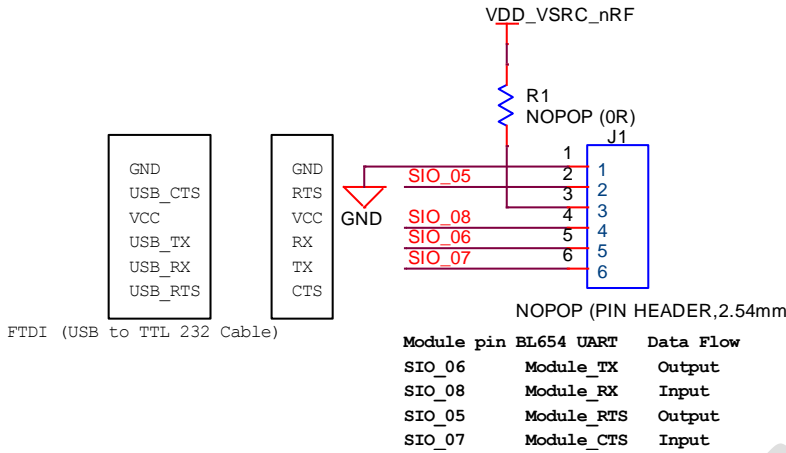
Plated Through Holes or Header Connector	BL654 Module Signals Exposed
<p><b>J41</b> NOPOP (PIN HEADER, 2.54mm 1X10P)</p> <p>Diagram showing 10 pins of connector J41. Pin 1 is SIO_38, Pin 2 is SIO_36, Pin 3 is SIO_37, Pin 4 is SIO_35, Pin 5 is SIO_34, Pin 6 is SIO_33, Pin 7 is SIO_25, Pin 8 is SIO_32_SWO_BLE, Pin 9 is SIO_24, and Pin 10 is GND.</p>	<p>BL654 pin plated holes for access</p> <p>SIO_38 SIO_36 SIO_37 SIO_35 SIO_34 SIO_33 SIO_25 SIO_33 SIO_25 SIO_32 SIO_24 GND</p>
<p><b>J29</b> NOPOP (PIN HEADER, 2.54mm 1X10P)</p> <p>Diagram showing 10 pins of connector J29. Pin 1 is SIO_27, Pin 2 is SIO_31, Pin 3 is SIO_26, Pin 4 is SIO_30, Pin 5 is SIO_04, Pin 6 is SIO_40, Pin 7 is SIO_41, Pin 8 is SIO_12, Pin 9 is SIO_11, and Pin 10 is GND.</p>	<p>BL654 pin plated holes for access</p> <p>SIO_27 SIO_31 SIO_26 SIO_30 SIO_04 SIO_40 SIO_41 SIO_40 SIO_41 SIO_12 SIO_11 GND</p>
<p><b>J46</b></p> <p>Diagram showing 20 pins of connector J46. Pin 1 is VDD_VSRC_nRF, Pin 2 is SWDIO_BLE, Pin 3 is SWDCLK_BLE, Pin 4 is SWDCLK_BLE, Pin 5 is SIO_32_SWO_BLE, Pin 6 is SIO_32_SWO_BLE, Pin 7 is nRESET_BLE, Pin 8 is nRESET_BLE, Pin 9 is SIO_07, Pin 10 is SIO_07, Pin 11 is SIO_32_SWO_BLE, Pin 12 is SIO_32_SWO_BLE, Pin 13 is SIO_12, Pin 14 is SIO_12, Pin 15 is SIO_11, Pin 16 is SIO_11, Pin 17 is SIO_41, Pin 18 is SIO_41, Pin 19 is SIO_41, and Pin 20 is GND.</p>	<p>J46 is NOPOP but is compatible to same connector on Nordic development board and brings out same signals.</p> <p>TRACECLK TRACEDATA0 TRACEDATA1 TRACEDATA2 TRACEDATA3</p>



**Plated Through Holes or Header Connector** | **BL654 Module Signals Exposed**

J1

Serial Port plated holes for access



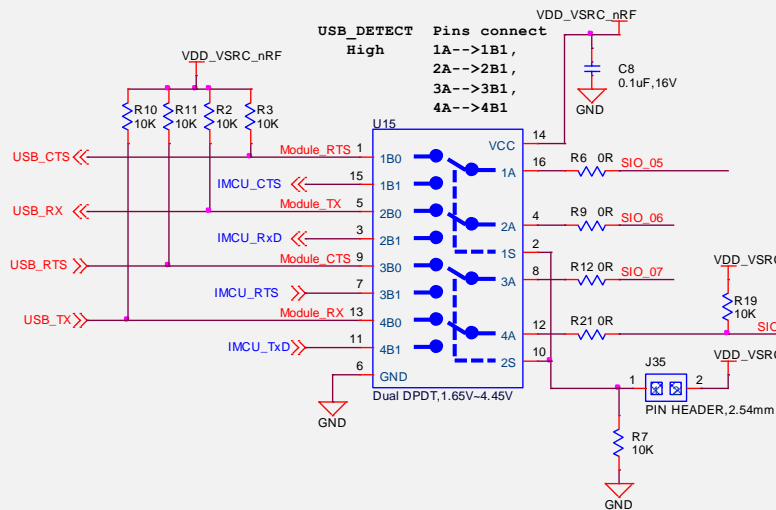
J35

**BL654 UART** routed to FTDI: **No Jumper in J35 (default).**  
**J35** routed to Atmel: **Fit Jumper in J35.**

Jumper in J35 selects between BL654 UART routed to FTDI Atmel MCU:

**No Jumper on J35 (default)**  
 Routes SIO\_05 (RTS) to FTDI CTS  
 Routes SIO\_06 (TX) to FTDI RX  
 Routes SIO\_07 (CTS) to FTDI RTS  
 Routes SIO\_08 (RX) to FTDI TX

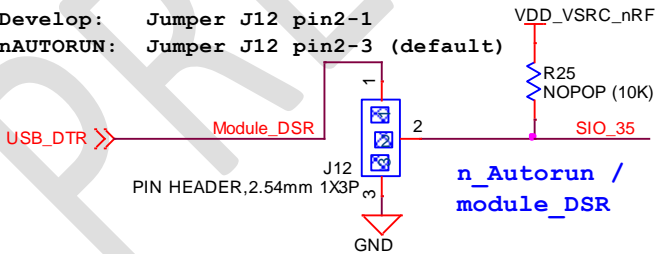
**Jumper on J35 (Route to Atmel)**  
 Routes SIO\_05 (RTS) to IMCU\_CTS  
 Routes SIO\_06 (TX) to IMCU\_RxD  
 Routes SIO\_07 (CTS) to IMCU\_RTS  
 Routes SIO\_08 (RX) to IMCU\_TxD

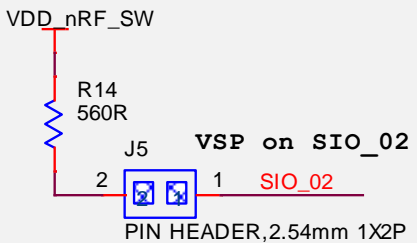
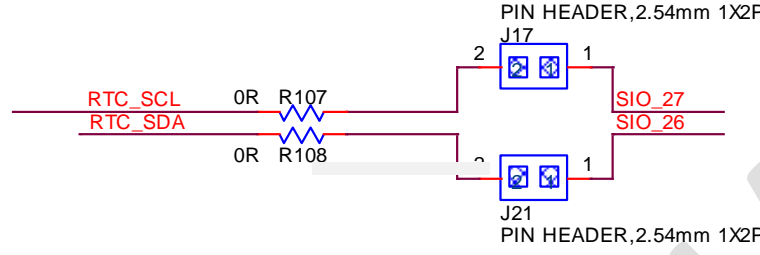
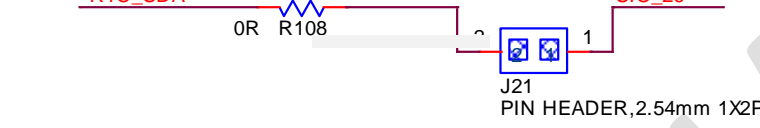
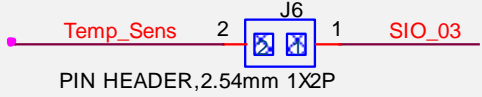
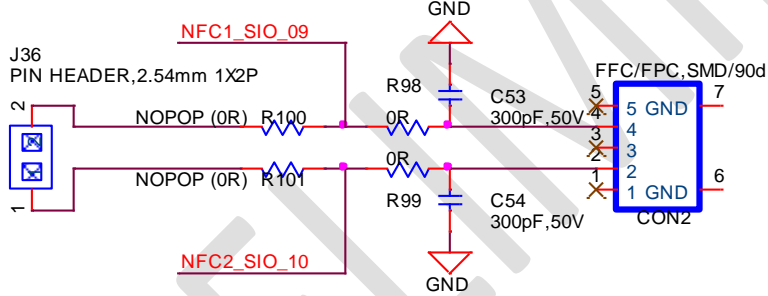


J12

**Develop:** Jumper J12 pin2-1  
**nAUTORUN:** Jumper J12 pin2-3 (default)

Connects SIO\_35 (nAutoRUN) to FTDI DTR  
 Default jumper fitted in J12 pin 2-3.



Plated Through Holes or Header Connector	BL654 Module Signals Exposed
<p><b>J5</b></p> 	<p>SIO_02 (for VSP capability) Can be used to pull-up SIO_02 to VDD_nRF_SW</p> <p>Default No Jumper fitted on J5 SIO_02</p>
<p><b>J17</b></p>  <p><b>J21</b></p> 	<p>J17 Routes SIO_27 (I2C_SCL) to RTC_SCL device J17 pin 2-1 jumper NOT fitted (default).</p> <p>J21 Routes SIO_26 (I2C_SDA) to RTC_SDA device J21 pin 2-1 jumper NOT fitted (default).</p>
<p><b>J6</b></p> 	<p>J6 routes SIO_03 to Temp Sensor J6 pin 2-1 jumper NOT fitted (default)</p>
<p><b>J36</b></p> 	<p>BL654 pin plated holes for access SIO_09 via R100 0R (default NOPOP) SIO_10 via R101 0R (default NOPOP)</p> <p>To use NFC1 pin as SIO_09 fit R100 0R and NOPOP R98. To use NFC1 pin as SIO_09 fit R101 0R and NOPOP R99.</p>

## Additional Peripherals/Sensors

The BL654 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, SPI EEPROM (U2), I2C RTC chip (U16), LED1(D1), LED2(D2), LED3(D3), LED4(D4) Button1(SW1), Button2(SW2), Button3(SW9) and Button4(SW10) the SIO pins direction and type must be set in the *smartBASIC* application to override the defaults in the BL654 firmware.

For more information on these sample applications, see GitHub *smartBASIC* sample applications repository on the BL654 product page at <https://github.com/LairdCP/BL654-Applications>.

### Temperature Sensor

The temperature sensor (U1) by default is connected to the BL654 module as jumper on J6 pin bridges TEMP\_Sens and SIO\_03.

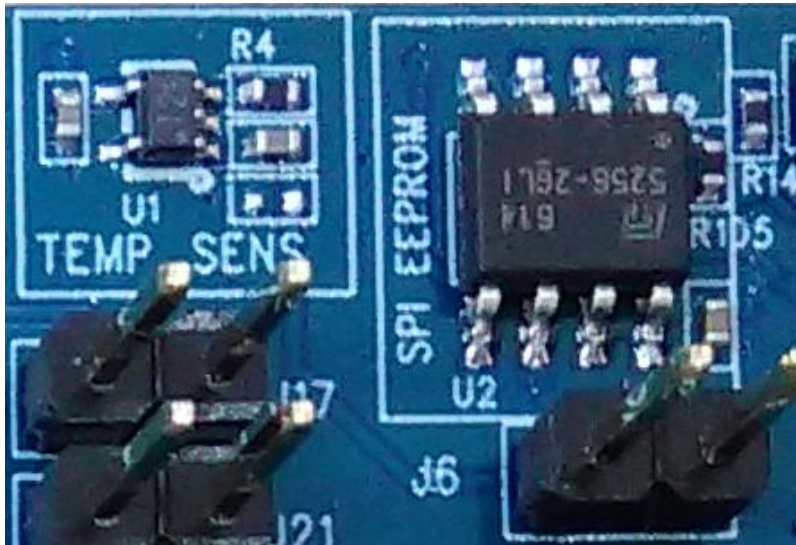
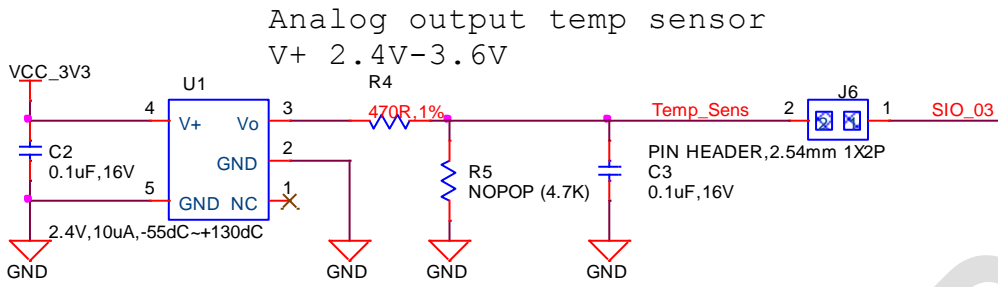


Figure 13: Temperature sensor schematic and PCB

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 BL654 module pin SIO\_03; but since the LM20BIM7 has an analogue output, the BL654 module SIO\_03 digital pin (DIO) must be configured as AIN analogue input (ADC). To configure the SIO\_03 pin from DIO pin to Alternate function AIN, see the example file “*ts.temperature.sensor.sb*” in the GitHub *smartBASIC* sample applications repository on the BL654 product page at <https://github.com/LairdCP/BL654-Applications>

Key specifications of the LM20BIM7 are as follows in Table 9.

Table 9: 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):

$$V_o(\text{mV}) = -11.67\text{mV}/^{\circ}\text{C} \times T + 1858.3$$

gives the following calculated Vo versus temperature:

**Table 10: LM20BIM7 Temperature to Voltage Output relationship**

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

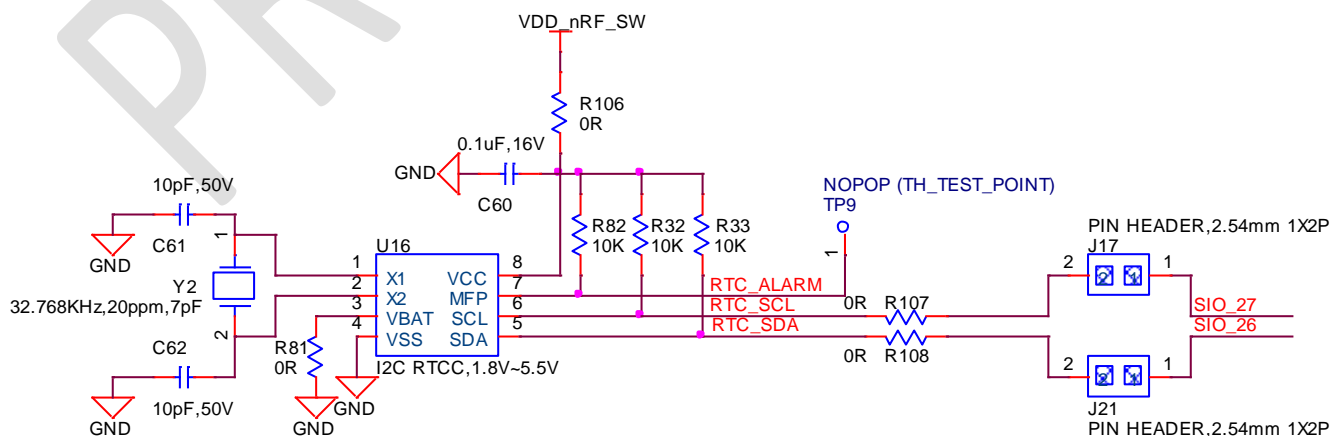
### I2C Sensor (RTC Chip)

The I2C RTC chip (U16) allows the BL654 I2C interface to be tested. The output of the RTC chip (U16) is on the I2C bus and is by default connected to the BL654 module via jumpers on J13 pins 2-3 and J14 pins 2-3.

**Table 11: I2C RTC chip BL654 I2C signal mappings**

I2C RTC EEPROM (U16)	BL654 module (U5) SIO	Comments
(U16 pin6) RTC_SCL	(U5 pin38) SIO_27	Fit jumper on J17 to route
(U16 pin5) RTC_SDA	(U5 pin36) SIO_26	Fit jumper on J21 to route

Fitting a jumper on J17 routes the RTC\_SCL signal to BL654 SIO\_27 and fitting a jumper on J21 routes the RTC\_SDA to BL654 SIO\_26.



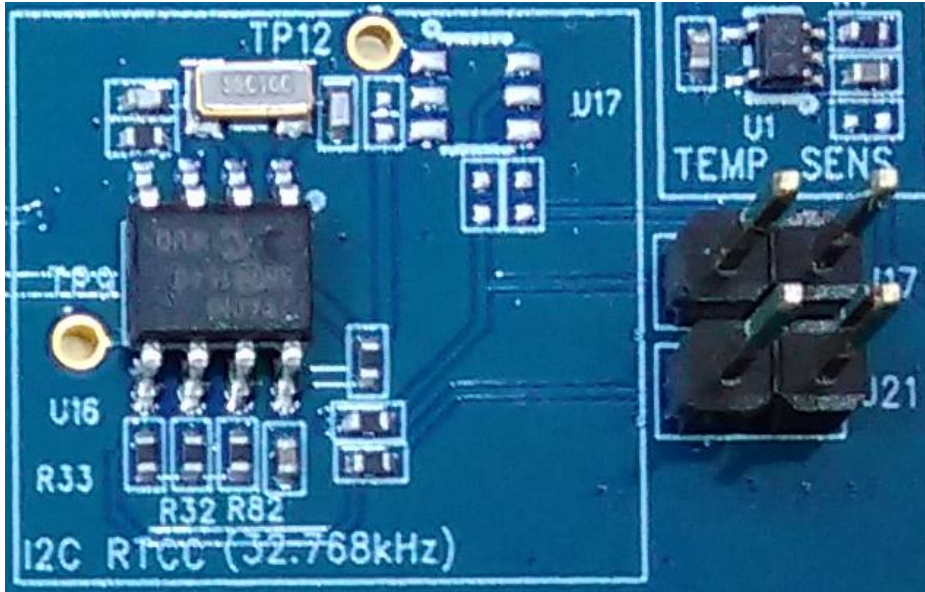


Figure 14: I2C device RTC chip schematic and PCB

To test the BL654 I2C interface, use *smartBASIC* application *rtcs.erver.sb* in the GitHub *smartBASIC* sample application repository on the BL654 product page at <https://github.com/LairdCP/BL654-Applications>. This application runs on the BL654 and can be used with an Android phone (requires an app such as nRF connect, [https://play.google.com/store/apps/details?id=no.nordicsemi.android.mcp&hl=en\\_GB](https://play.google.com/store/apps/details?id=no.nordicsemi.android.mcp&hl=en_GB)) or another BT900/BL620/BL652/BL654 loaded with "*rtcc.lient.sb*".

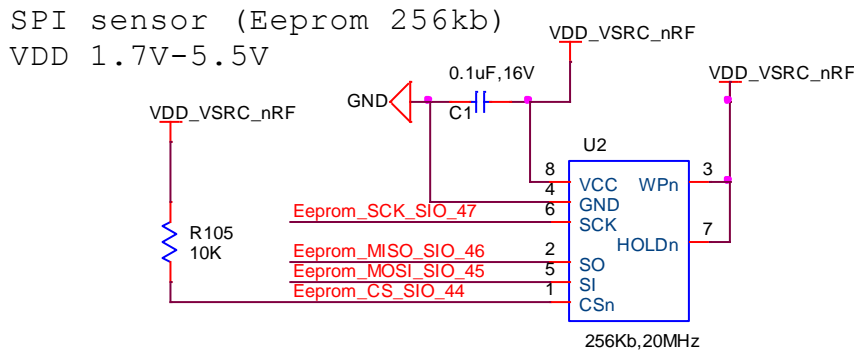
The *smartBASIC* application "*rtcs.erver.sb*" is a BLE RTC server, and it advertises the current time (which it gets from the I2C RTC chip (U4)).

### SPI Device EEPROM

The SPI EEPROM device (U2) is connected to the BL654 SPI pins **directly**. By default, the BL654 Module SIO\_44 (used as the SPI\_CS) is connected to EEPROM (U2) slave select line. Table 12 lists signal mappings of how the SPI EEPROM (U2) is wired to BL654 SIO pins.

Table 12: SPI EEPROM to BL654 SPI signal mappings

SPI EEPROM (U2)	BL654 (U5) SIO	Comments
(U2pin6) Eeprom_SCK_SIO_47	(U5pin53) SIO_47	
(U2pin2) Eeprom_MISO_SIO_46	(U5pin51) SIO_46	
(U2pin5) Eeprom_MOSI_SIO_45	(U5pin56) SIO_45	
(U2pin1) Eeprom_CS_SIO_44	(U5pin54) SIO_44	Configure SIO_44 as an output and drive output low in <i>smartBASIC</i> application to select SPI slave (SPI EEPROM U2).



**Figure 15: SPI EEPROM schematic and PCB**

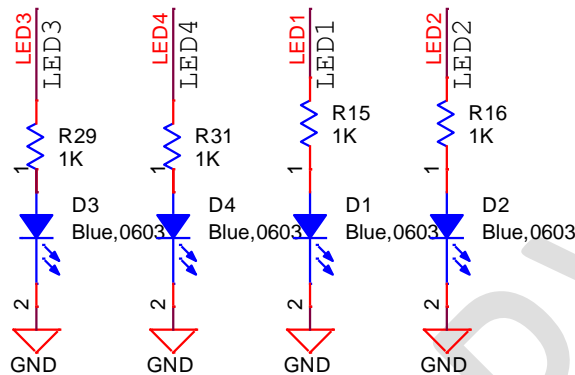
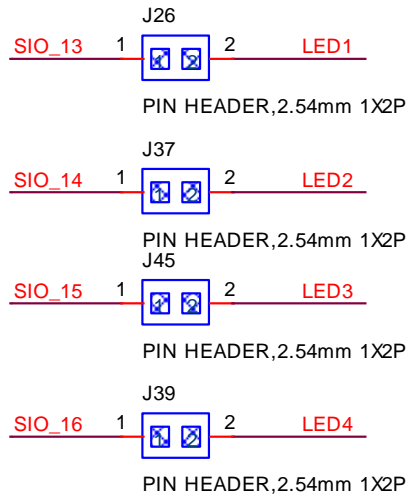
For a working example of the BL654 SPI interface using the SPI EEPROM (U2), a *smartBASIC* application for this will be available in the future in the GitHub *smartBASIC* sample application repository on the BL654 product page at <https://github.com/LairdCP/BL654-Applications>.

### Push Button and LED Connected to BL654

The two push buttons and two LEDs on the BL654 are connected to dedicated SIOs of the BL654 module.

**Table 13: LED's and Buttons to BL654 SIO signal mappings**

Part	BL654 (U5) SIO	Comments
LED1 (D1)	Pin 20 SIO_13 (via header J26)	To connect LED1 to SIO_13, fit jumper in J26
LED2 (D2)	Pin 22 SIO_14 (via header J37)	To connect LED1 to SIO_14, fit jumper in J37
LED1 (D3)	Pin 18 SIO_15 (via header J26)	To connect LED1 to SIO_15, fit jumper in J45
LED2 (D4)	Pin 21 SIO_16 (via header J37)	To connect LED1 to SIO_16, fit jumper in J39
Button 1 (SW1)	Pin 27 SIO_11	
Button 2 (SW2)	Pin 28 SIO_12	
Button 3 (SW1)	Pin 10 SIO_24	
Button 4 (SW2)	Pin 8 SIO_25	



SIO_11	BUTTON1
SIO_12	BUTTON2
SIO_24	BUTTON3
SIO_25	BUTTON4

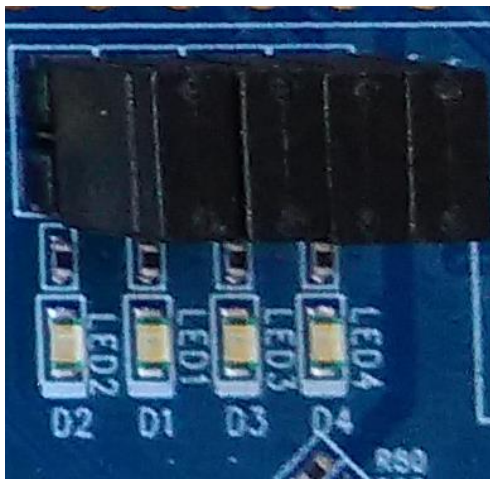


Figure 16: LEDs and Buttons schematic and PCB

The buttons (BUTTON1 and BUTTON2) have no external pull-up resistor, so to use the buttons, the SIO\_11 and SIO\_12 pins must be configured as inputs with internal pull-up resistors (which is the default). The following *smartBASIC* lines configure the pull-ups:

```
rc = GPIOSETFUNC(11,1,4)           '//sets SIO_11 (Button1) as a digital in,
                                   strong pull up
rc = GPIOSETFUNC(12,1,4)           '//sets SIO_12 (Button2) as a digital in,
                                   strong pull up
```

Refer to the *smartBASIC* application script example “*btn.button.led.test.sb*” in the GitHub *smartBASIC* sample application repository on the BL654 product page at <https://github.com/LairdCP/BL654-Applications>

The LEDs are active high, meaning that writing a logical one (“1”) to the output pin illuminates the LED.

One example of when push buttons can be used is when a *smartBASIC* application is written to simulate a generic data profile. Push buttons can then be pressed to increment and decrement, such as a heart rate.

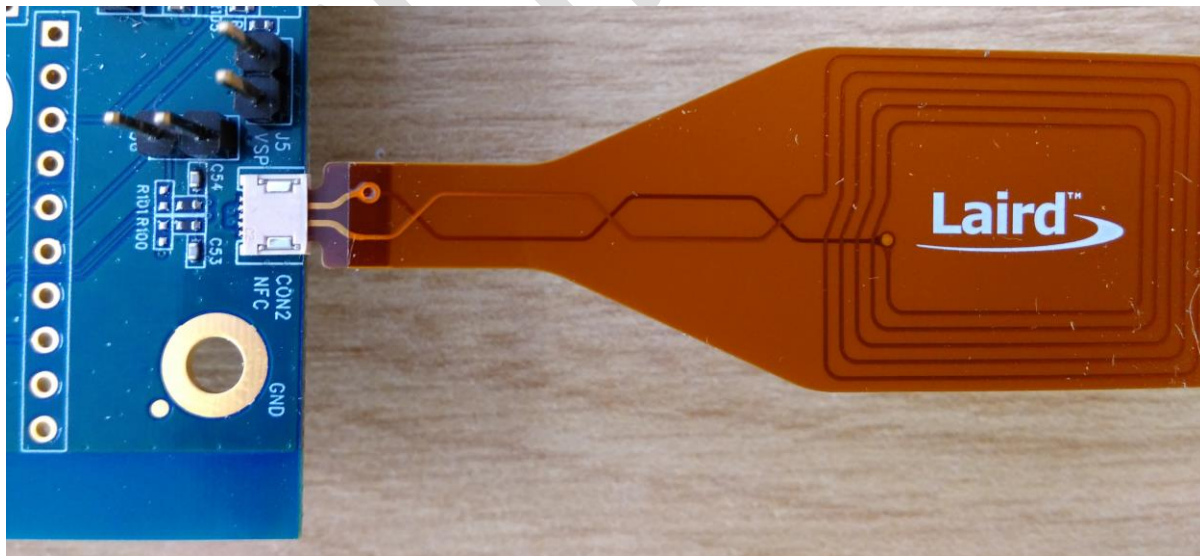
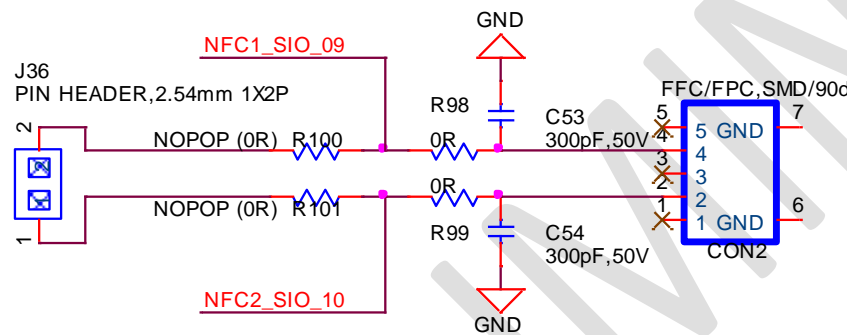
### NFC External Antenna Connector and NFC Antenna RF Matching Circuit

The NFC antenna input connector (CON2) allows the Laird supplied flex-PCB NFC antenna to be plugged in. The BL654 module NFC circuit uses two pins, pin 59 (**NFC1/SIO\_9**) and pin 57 (**NFC2/SIO\_10**) to connect the antenna. These pins are shared with GPIOs (**SIO.09** and **SIO.10**). BL654 NFC pins are enabled by default. NFC can be disabled via *smartBASIC* application. Pin 59 (**NFC1/SIO\_9**) and pin 57 (**NFC2/SIO\_10**) are configured by default on the development board schematic to use NFC antenna, but if pin 59 (**NFC1/SIO\_9**) and pin 57 (**NFC2/SIO\_10**) are needed as normal GPIOs, R98 and R99 must be removed and R100 and R101 must be shorted by 0R.

C53 (300pF) and C54 (300pF) are RF tuning elements for the flexi-PCB NFC antenna.

**Table 14: NFC input BL654 SIO signal mappings**

BL654 (U5) SIO	Bring out SIO_9 and SIO_10 to NFC antenna connector (CON2)	Bring out SIO_9 and SIO_10 to Header connector (J36)
pin 59 ( <b>NFC1/SIO_9</b> )	Fit R98 0R (default) Remove R100 0R (default)	Remove R98 0R Fit R100 0R
pin 57 ( <b>NFC2/SIO_10</b> )	Fit R99 0R (default) Remove R101 0R (default)	Remove R99 0R Fit R101 with 0R



**Figure 17: NFC antenna RF matching circuit, NFC antenna connector schematic and NFC plugged in to connector CON2**



The *smartBASIC* application *nfc.all.launch.sb* in the GitHub *smartBASIC* sample application repository on the BL654 product page at <https://github.com/LairdCP/BL654-Applications> exercises the following over the BL654 NFC:

- On Android NFC enabled devices – Opens the Laird toolkit application or shows it in the Google Play store if it's not installed
- On Windows NFC enabled devices – Opens the calculator
- On other NFC enabled devices – Shows the Laird website or text saying **this is a BL654**

### Optional External Serial QSPI Flash IC

There is an optional external serial QSPI flash IC (U18) that may be used, for example, for data logging purposes. U18 can also be capable of SPI mode.

Solder bridges SB4, SB5, SB6, SB7, SB10 and SB11 must individually be shorted to connect this optional external serial (QSPI) flash (U18 to the BL654 module).

By default, these BL654 pins are GPIO pins.

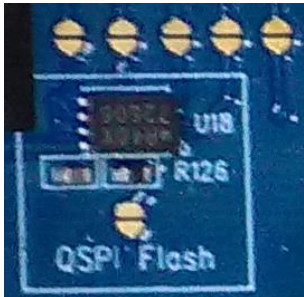
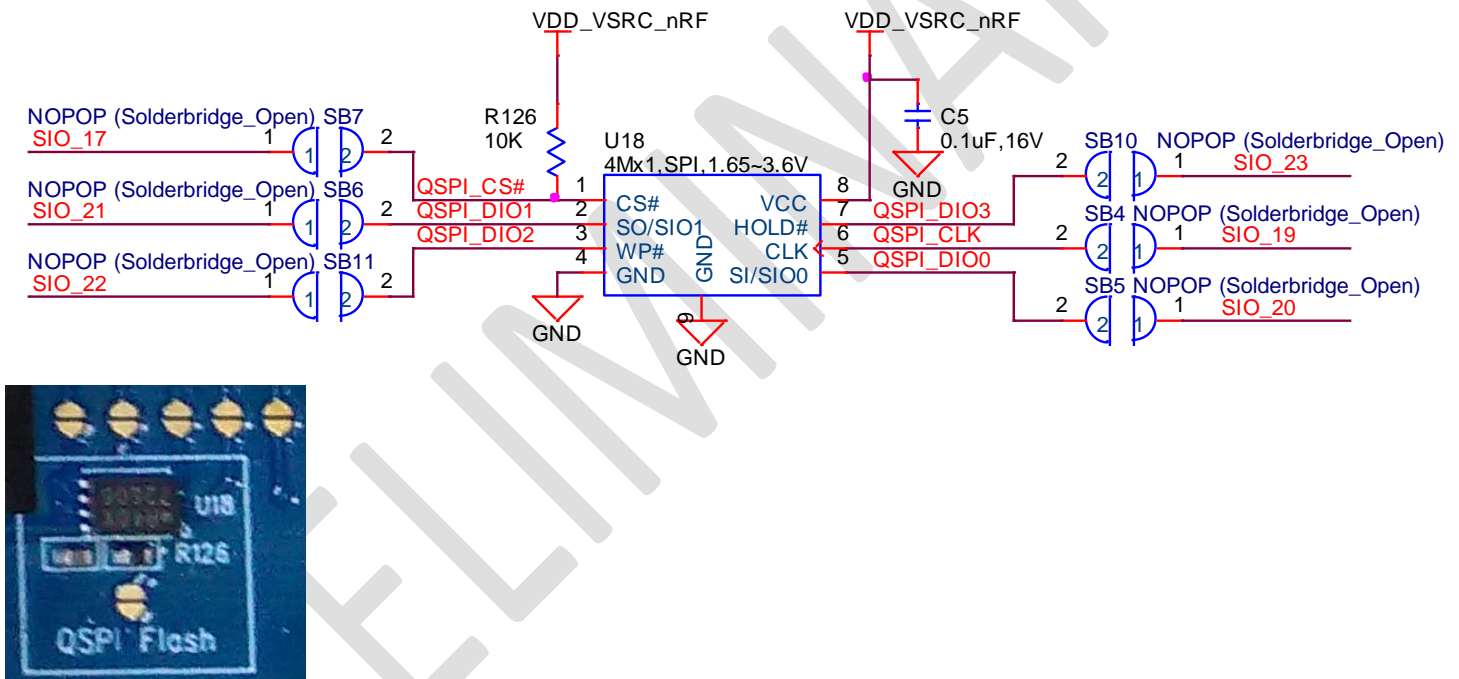


Figure 18: Optional external serial SPI flash IC (U18) schematic and PCB

Table 15 shows the U18 pin mapping to BL654 SIO pin mapping (via open solder bridges).

Table 15: U18 SPI or QSPI flash IC pin mapping to BL654 GPIO mappings

BL654 Pin (Pin Name)	U18 Pin (QSPI flash IC) and Pin Name	Via Open Solder bridge
Pin 16 (SIO_17/QSPI_CS)	Pin 1 CS#	SB7 BL654 Dev board R126 10K pull-up on, therefore by default, device not selected. Drive from BL654 line low to select.
Pin 14 (SIO_19/QSPI_CLK)	Pin 6 SCLK	SB4
Pin 13 (SIO_20/QSPI_DIO0)	Pin 5 SIO_0/SI	SB5

BL654 Pin (Pin Name)	U18 Pin (QSPI flash IC) and Pin Name	Via Open Solder bridge
Pin 12 (SIO_21/QSPI_DIO1)	Pin 2 SIO_1/SO	SB6
Pin 11 (SIO_22/QSPI_DIO2)	Pin 3 SIO_2/WP#	SB11
Pin 9 (SIO_23/QSPI_DIO3)	Pin 7 SIO_3/HOLD#	SB10

The *smartBASIC* application for this external optional serial QSPI flash IC will be available in the future in the GitHub *smartBASIC* sample application repository on the BL654 product page at <https://github.com/LairdCP/BL654-Applications>

### Optional 32.76 kHz Crystal

The BL654 on-chip 32.768kHz RC oscillator provides the standard accuracy of  $\pm 250$  ppm, with calibration required every 8 seconds (default) to stay within  $\pm 250$  ppm.

The BL654 also allows, as an option, to connect an external higher accuracy ( $\pm 20$  ppm) 32.768 kHz crystal to the BL654-SX-xx pins SIO\_01/XL2 (pin 41) and SIO\_00/XL1 (pin 42). This provides improved protocol timing and helps with radio power consumption in the system standby doze/deep sleep modes by reducing the time that the Rx window must be open.

To connect the optional external 32.76kHz crystal oscillator circuit to the BL654 module, remove R127 and R128 and short SB8 and short SB9.

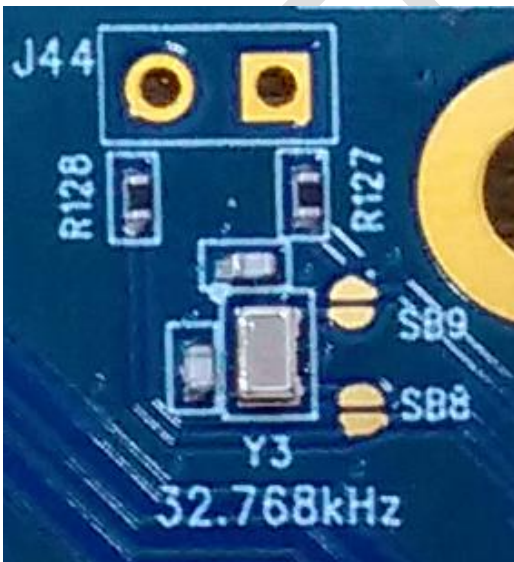
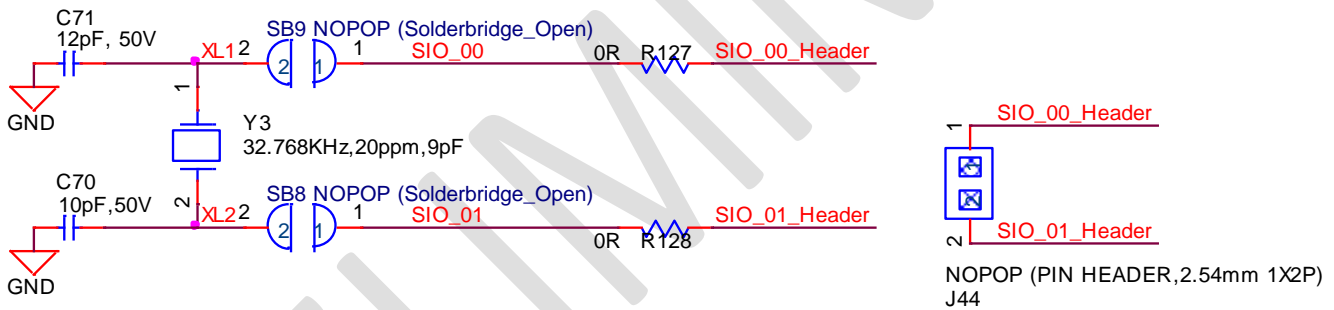


Figure 19: Optional external 32.768kHz crystal circuit schematic and PCB

A *smartBASIC* application will be available in the future in the GitHub *smartBASIC* sample application repository on the BL654 product page at <http://www.lairdtech.com/products/bl654-ble-thread-nfc-modules>.

## OTHER FEATURES

### Current Consumption Measurement

A removable jumper (on J7 and J9) is provided to break the power supply line directly to the module, allowing you to measure current consumption. For normal operation, the jumper on J7 (and J9) between pin1 and pin2 must be fitted (and is fitted by default).

**IMPORTANT:** To achieve the optimal power consumption of the BL654 series module on the development board, see the "*lp.low.power.deep.sleep.sb*" file in the GitHub *smartBASIC* sample application repository on the BL654 product page at <https://github.com/LairdCP/BL654-Applications>

**Note:** This measures the current consumption of the **BL654** series module ONLY.

The current drawn by the BL654 series module can be monitored on the development board. Figure 20 shows the schematic and location of measuring points on the PCB related to current measurements.

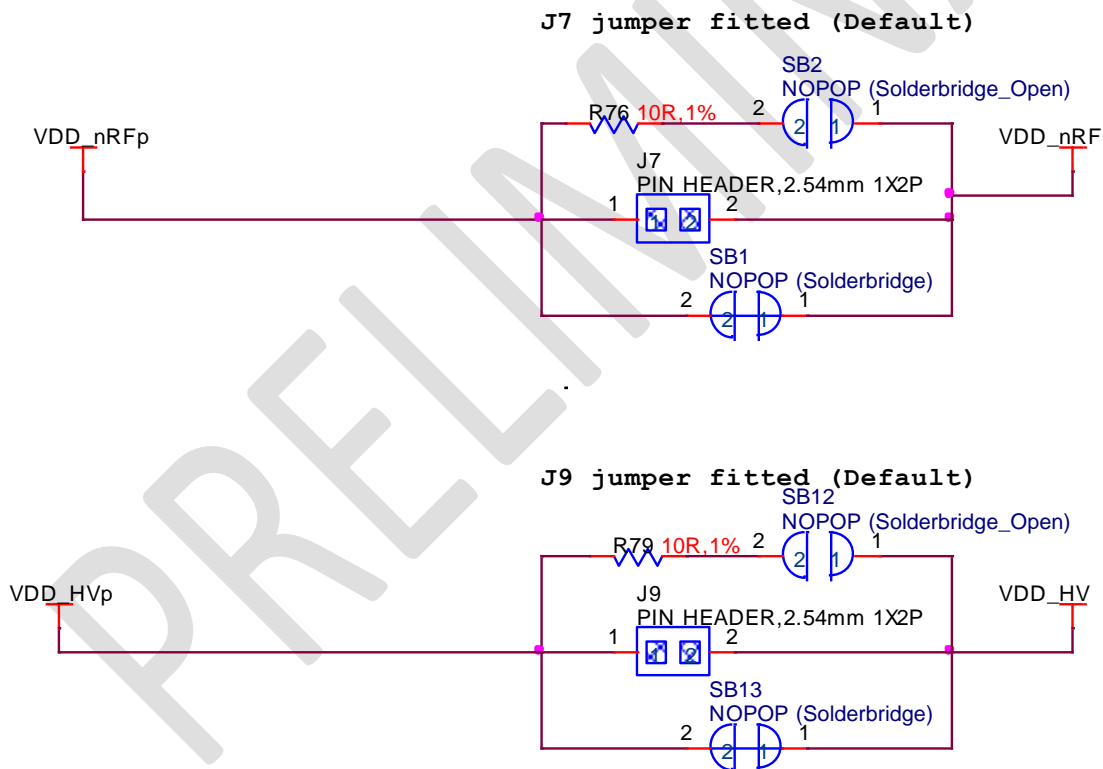




Figure 20: Current measurement schematic and PCB

There are two primary ways to measure the current consumption:

- **Using Ammeter** – Cut solder bridge SB1 and connect an ampere meter between the two pins of J7 pins 1-2. This monitors the current directly. This is when BL654 is powered using the normal voltage Mode (BL654 operated VDD pin). If the BL654 is powered using the high voltage Mode (BL654 operated VDD\_HV pin), then only cut solder bridge SB13 and connect an ampere meter between the two pins of J9 (pins 1-2).
- **Using Oscilloscope** – The open solder bridge SB2 first needs to be shorted with solder, then the on-board 10 Ohm resistor R76 which is mounted across J7 pins 1-2 can be used as current sense resistor. Connect an oscilloscope or similar with two probes on the pins on the J7 connector and measure the voltage drop. The voltage drop is proportional with current consumption. If the 10 Ohm resistor is chosen, 10 mV equals 10mA.  
This method allows the dynamic current consumption waveforms to be shown on an oscilloscope as the BL654 radio operates. This can provide insight into power optimization.

## APPENDIX

### Coin Cell Insertion

To insert the coin cell, follow these steps:

1. Push the coin cell against positive contact spring of holder J34 (on the back side of the devboard).

---

**Note:** The coin cell sits below the positive contact spring (as shown with arrow).

---



Figure 21: Inserting the coin cell (step 1)

2. Push the coin cell down into the holder (J34).



Figure 22: Inserting the coin cell (step 2)

## Coin Cell Removal

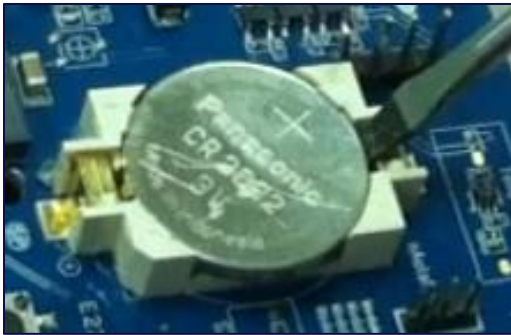
To remove the coin cell, follow these steps:

1. Hold down the coin cell holder (J34) at the corners.
2. Use a screwdriver in the position shown in picture below, to gently remove the coin cell from the coin cell holder (J34). This is the correct method to remove coin-cell from holder (J34).

---

**Note:** Due to tight fit of coin cell in the coin-cell holder (J34), care should be taken prevent damage to the J34 land pads.

---



**Figure 23: Removing the coin cell (step 2)**

## 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 BL654 Product Page](#).

For any additional questions or queries, or to receive technical support for this Development Kit or for the BL654 module series, please contact Embedded Wireless Solutions Support: <http://ews-support.lairdtech.com>

© Copyright 2018 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.