Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

User Guide

BL654 Development Kit

Part # 455-00001 & 455-00002

Version 0.1

Revision History

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

CONTENTS

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 – *smart*BASIC or for use with the Nordic SDK for C development. More information regarding this product series including a detailed module user's guide and *smart*BASIC 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:

Development Board	The development board has the required BL654 module soldered onto it and exposes all available hardware interfaces.
Power Options	▪ USB cable – Type A to micro type B. The cable also provides serial communications via the FTDI USB – RS232 converter chip on the board. ▪ DC barrel plug for connection to external power supply (5.5VDC max) ▪ 3x AAA battery holder fitted on underside of development board ▪ Coin-cell holder (powers module only) fitted on underside of development board
Two-pin jumpers for pin headers (5)	Five jumpers for 2.54 mm pitch headers used on BL654 development board.
Fly leads (6)	Supplied to allow simple connection of any BL654 module pin (available on Plated Though Holes or headers on J44, J47, J48, J41, J29, J1, J12, J1, J5, J17, J21, J6 and J36.
External BLE dipole antenna	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.
External NFC antenna	Supplied with Development Kit part # 455-00002 only Laird NFC flexi-PCB antenna – Part # 0600-00061
Web link card	Provides links to additional information including the BL654 user guide, firmware, terminal utilities, schematics, quick start guides, firmware release notes and more. **Note:** Sample *smart*BASIC applications are available to download from the Laird https://github.com/LairdCP/BL654-Applications

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 *smart*BASIC runtime engine firmware but no onboard *smart*BASIC; because of this it starts up in AT command mode by default.

Applications in *smart*BASIC are simple and easy to develop for any BLE application. Sample *smart*BASIC 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
- *smart*BASIC 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
- *smart*BASIC application upgrade capability:
  - Via UART (using the FTDI USB-UART)
  - Via OTA (Over-the-Air)

Understanding the Development Board

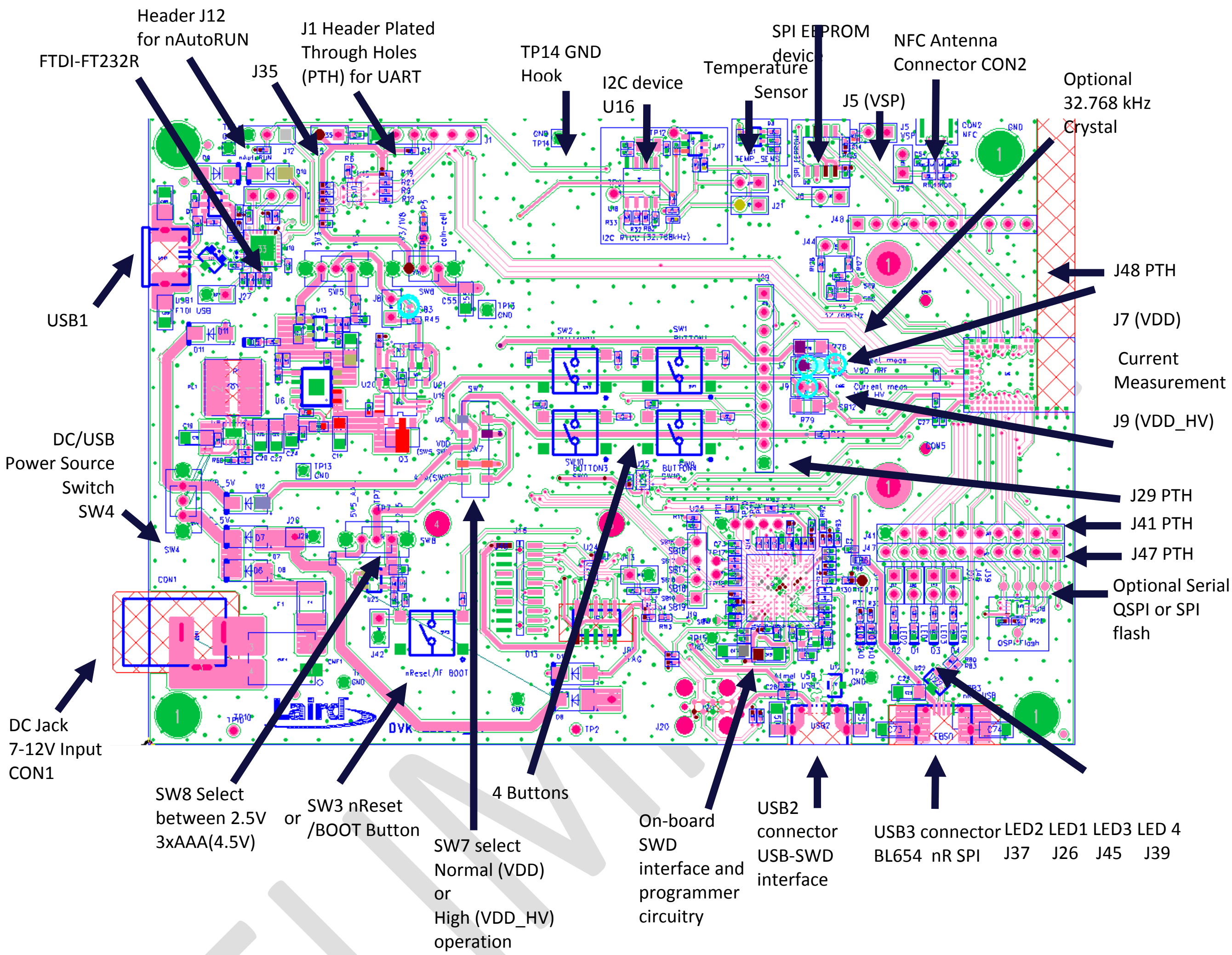


Figure 1: Dev board contents and locations

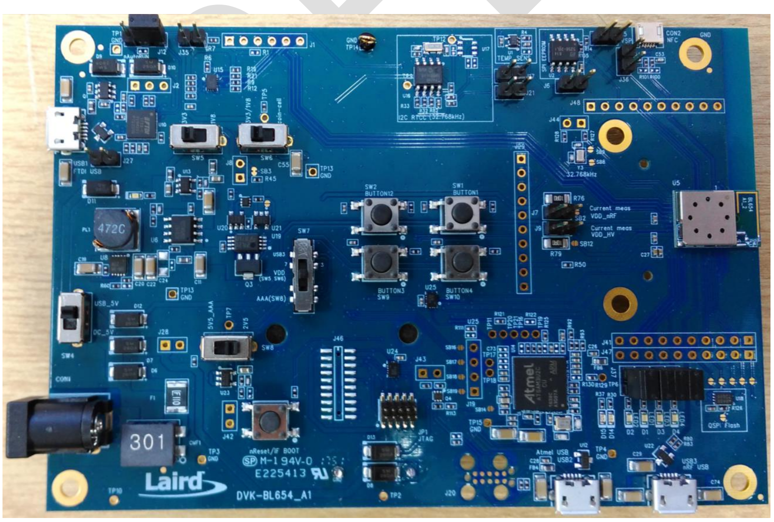


Figure 2: Development board for BL654 (fitted with 451-00001 Integrated Antenna module for example)

BL654 Default Configuration and Jumper Settings

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

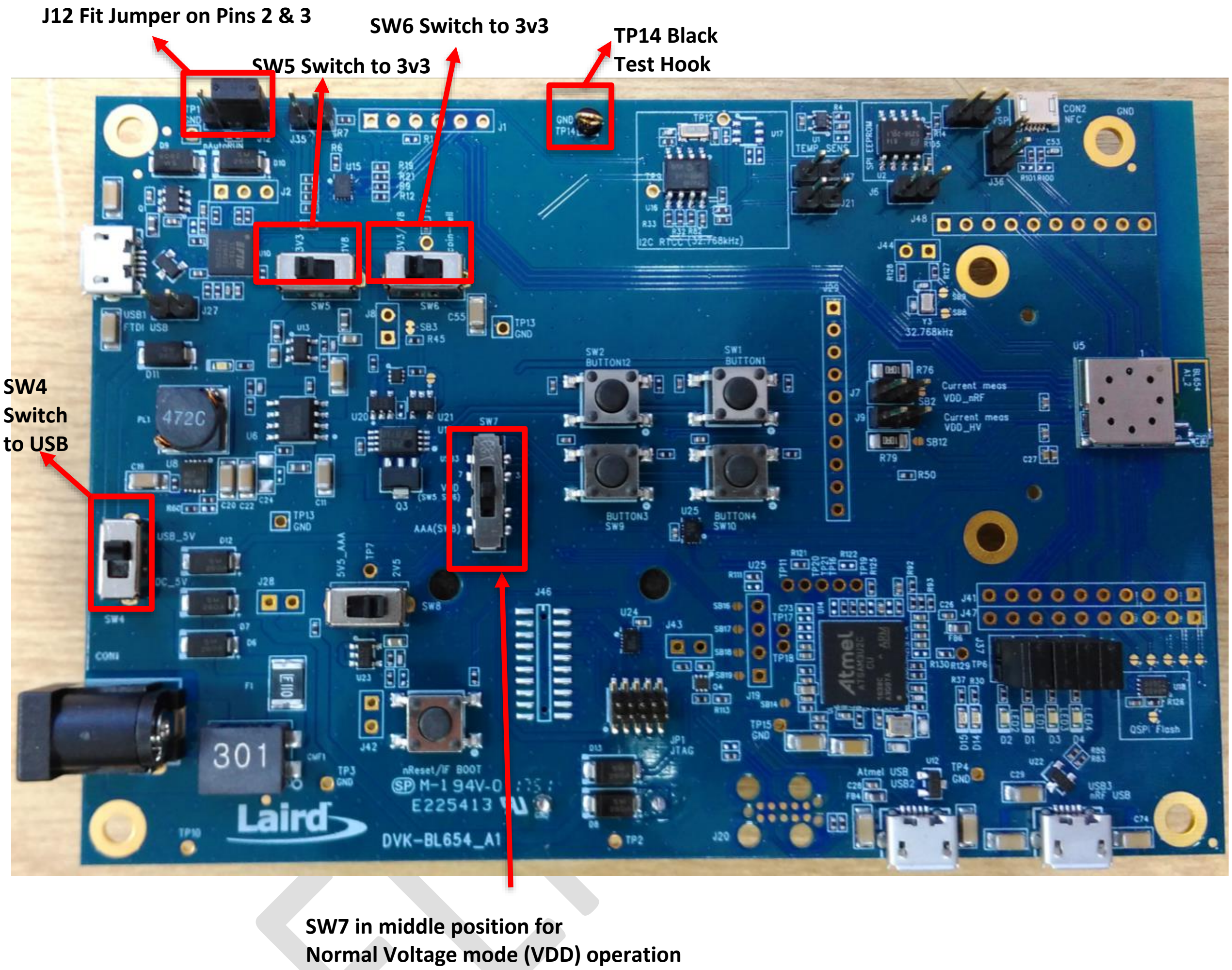


Figure 3: Correct BL654 development board jumper and switch settings

Functional Blocks

TBC

Power Supply

Figure 4 shows the BL654 development board Power Supply block.

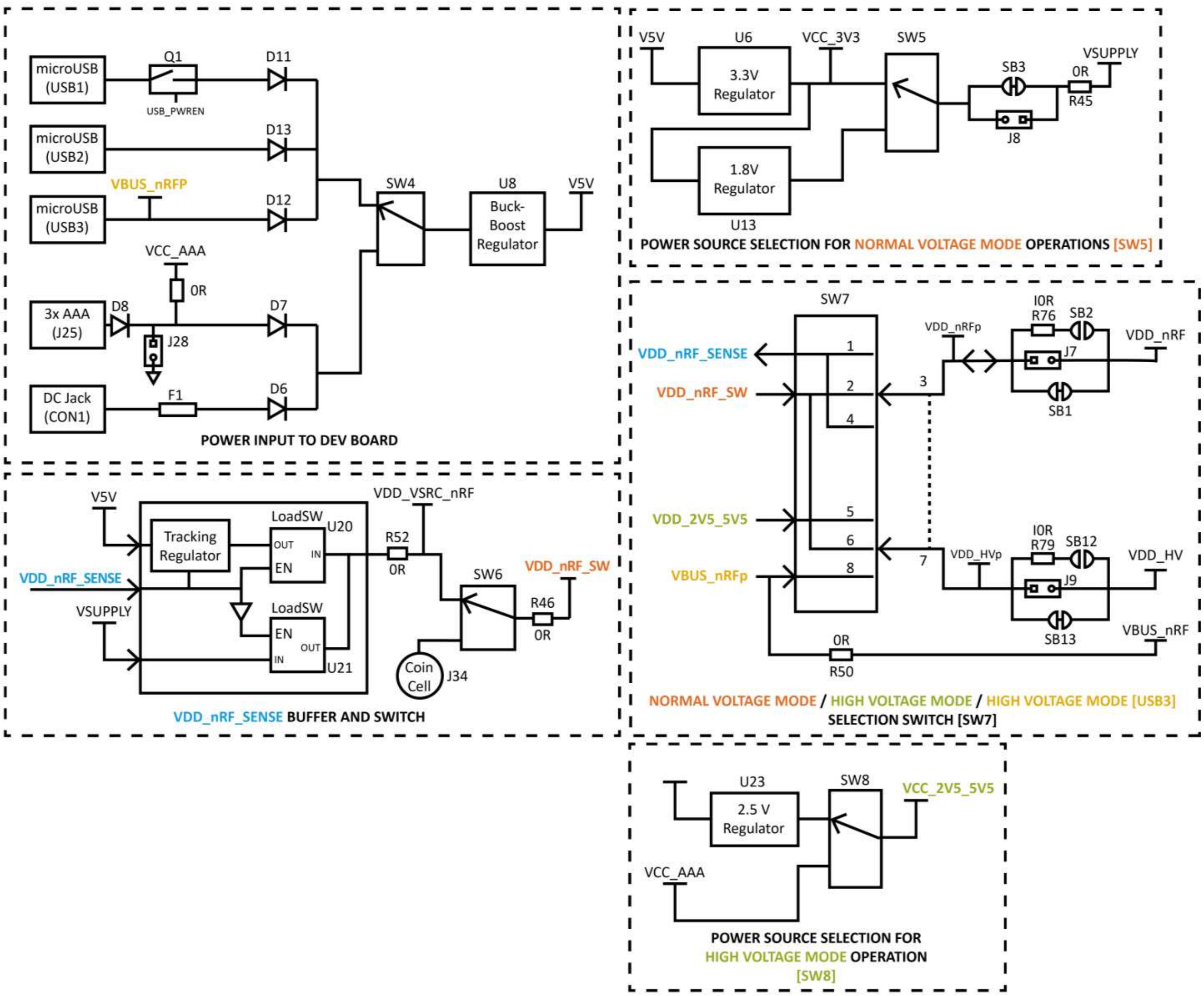


Figure 4: BL654 development Kit power supply

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-5.5VV), 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 in 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**
SW7 Top position – Silkscreen: USB3 (Source from USB3) High Voltage Mode with BL654 USB used (USB3)	Note 1	USB3 voltage	USB3 voltage
SW7 Middle position – Silkscreen: SW5 SW6 (Source from SW5, Note 2) Normal Voltage Mode	Decided by SW5 (default SW5 on 3.3V position)	N/A	N/A
SW7 Bottom position – 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 0R.

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 circuity 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.

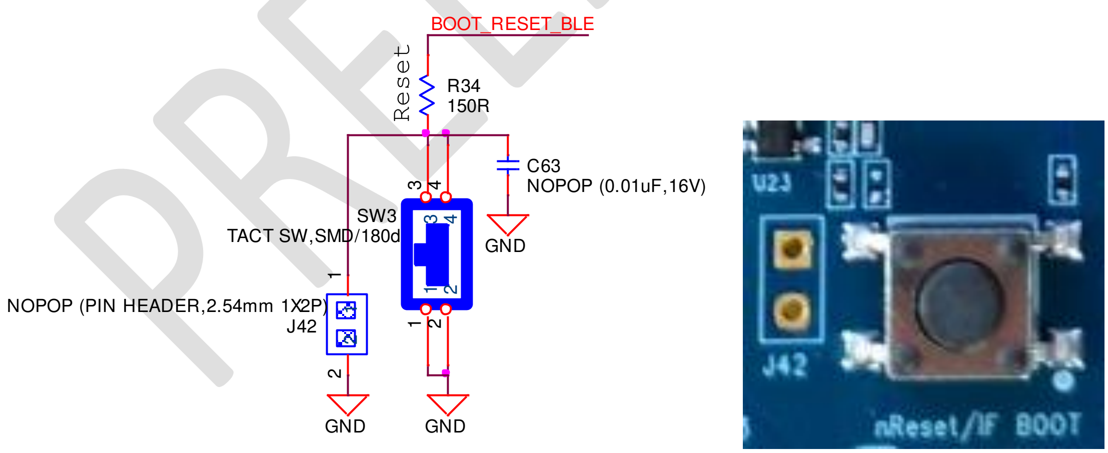


Figure 5: Reset button placement

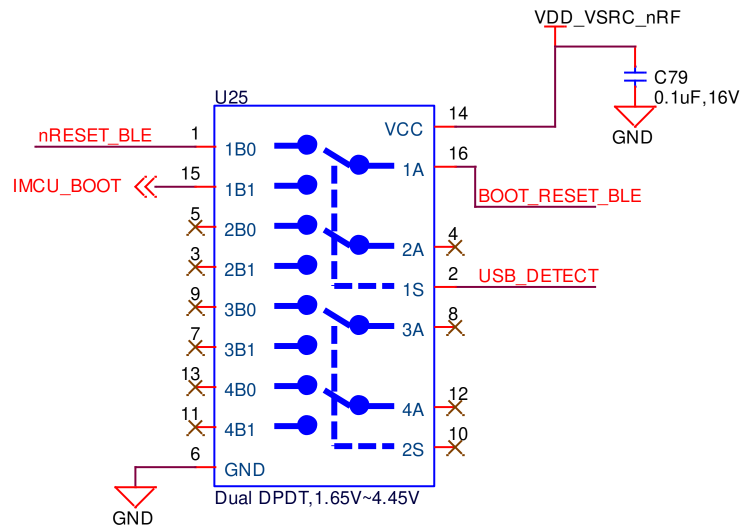


Figure 6: Reset button schematic and routing through analog switch and location diagram

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 *smart*BASIC 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 *smart*BASIC runtime engine firmware or loading *smart*BASIC 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.

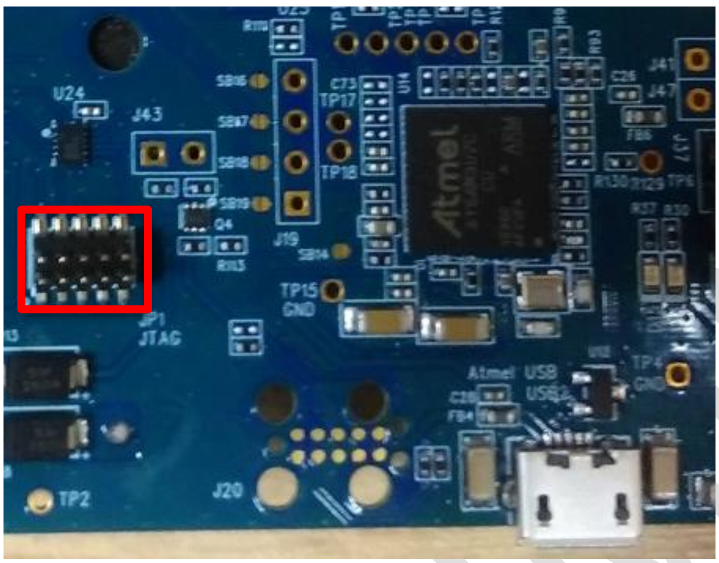


Figure 7: JP1 on development board

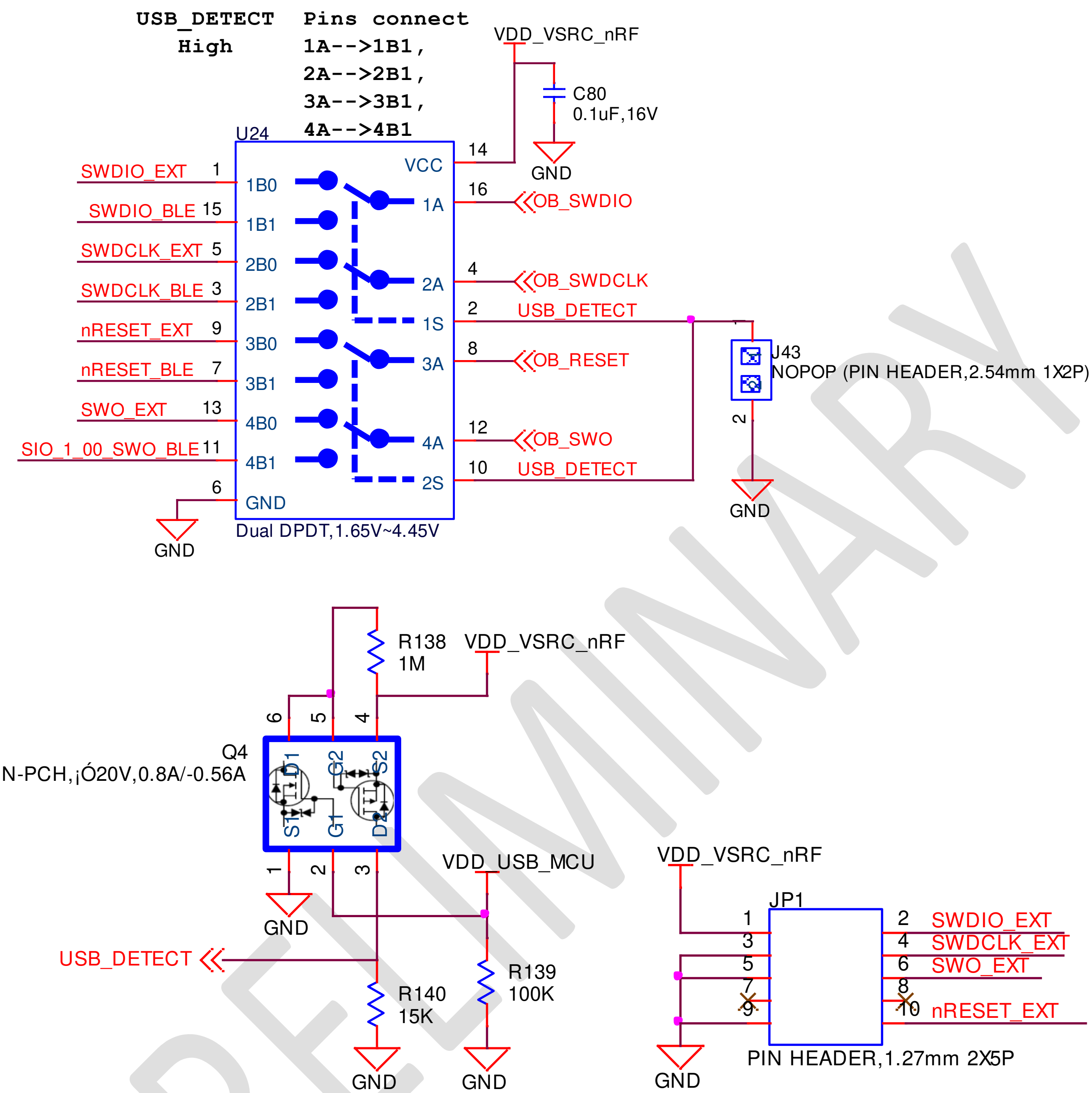


Figure 8: USB to SWD onboard circuitry routing via analog switch (U24)

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_SWDCLK	SWDCLK_BLE (pin 3)	SWDCLK_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 *smart*BASIC 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.

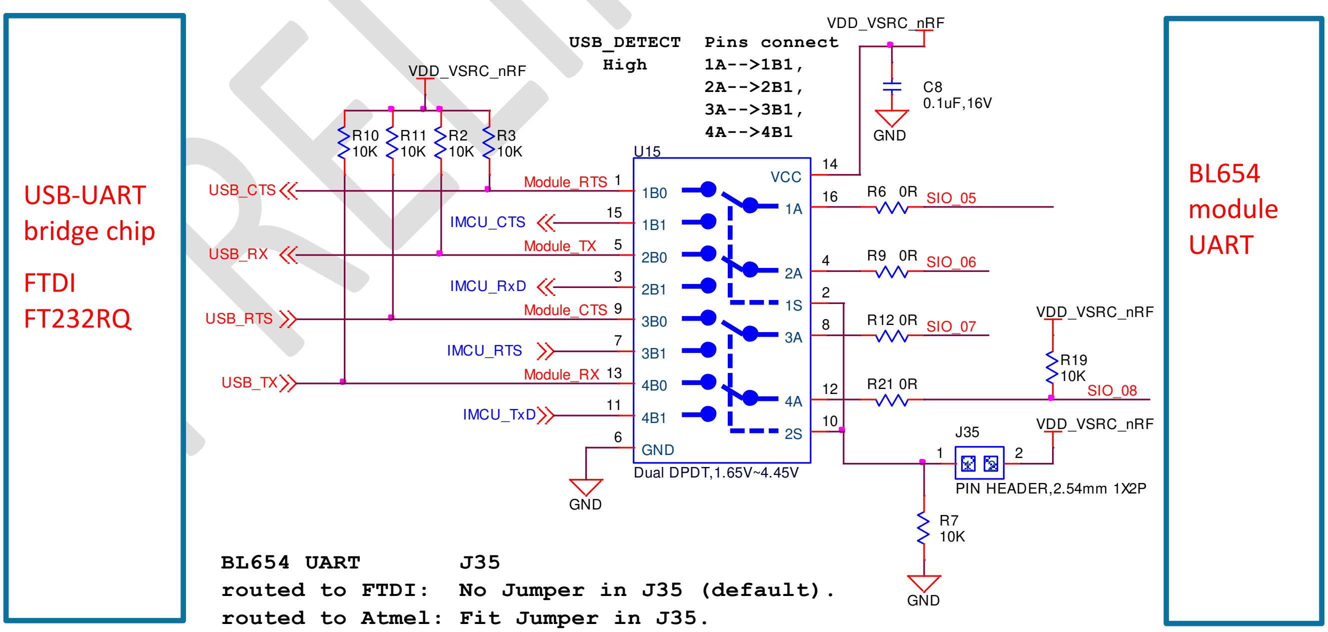
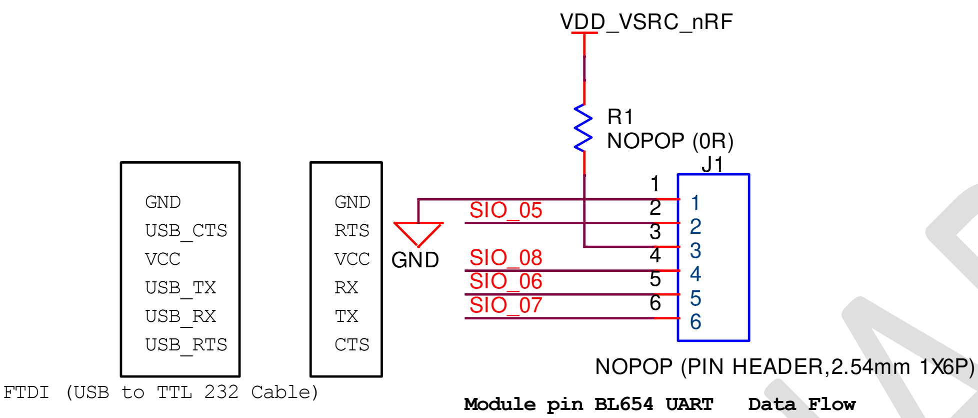


Figure 9: USB to UART (via FTDI chip on devboard) interface via analog switch U15

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.



Module pin	BL654 UART	Data Flow
SIO_06	Module_TX	Output
SIO_08	Module_RX	Input
SIO_05	Module_RTS	Output
SIO_07	Module_CTS	Input

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.

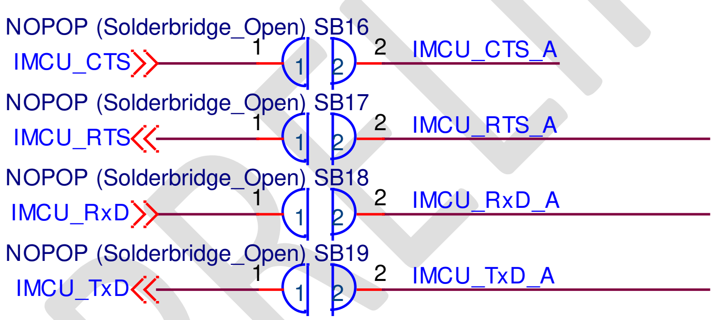


Figure 11: Open solder-bridges on the UART interface running from Atmel MCU (U4) to analog switch U15 (to BL654ultimately)

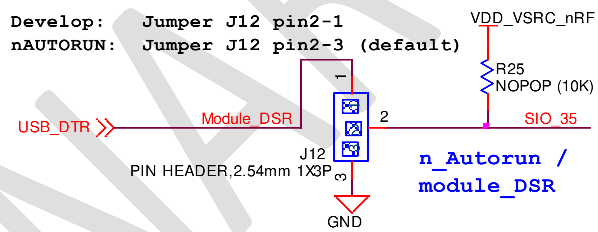
nAutoRUN Pin and Operating Modes

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: *smart*BASIC 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 *smart*BASIC runtime engine FW executes the *smart*BASIC 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
J12 Jumper Position	 **Develop** Jumper on J12 pins 2-1	 **nAutoRUN (default)** 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) *smart*BASIC App Download

The OTA feature makes it possible to download *smart*BASIC 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 *smart*BASIC 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 *smart*BASIC application script over the air to the BL654, since a *smart*BASIC application is downloaded using AT commands.

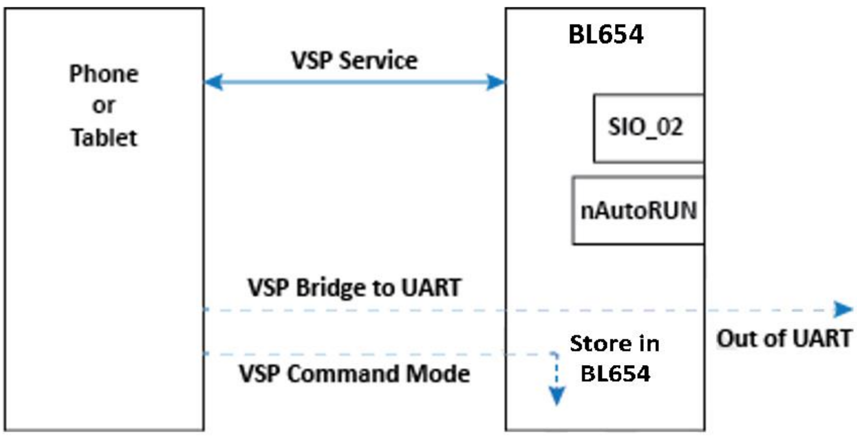


Figure 12: Differences between VSP bridge to UART mode and VSP Command mode

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 *smart*BASIC 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

Port (Windows)	1 to 255
Port (Mac/Linux)	Any/dev/tty device
Baud Rate	1200 to 1000000 **Note:** Baud rate default is 115200 for BL654.
Parity	None
Data Bits	8
Stop Bits	1
Handshaking	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)
- 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 BL654.

Note: Full details on *smart*BASIC are available in the *smart*BASIC 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.

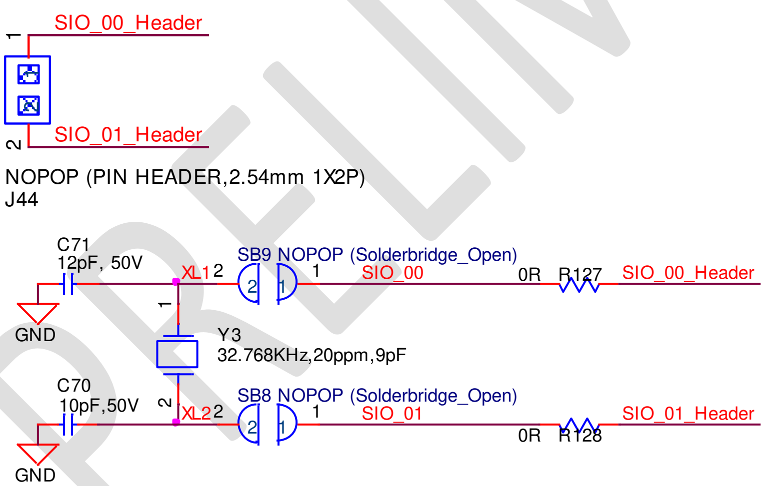
Note: The BL654 module signal pins designation SIO (Signal Input /Output).

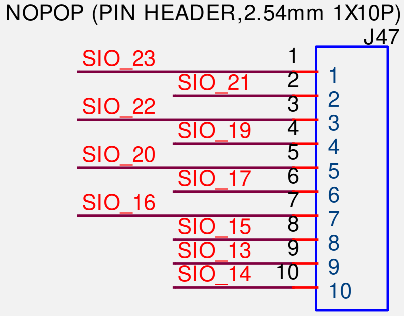
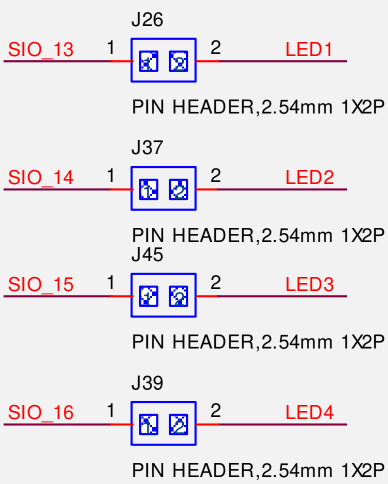
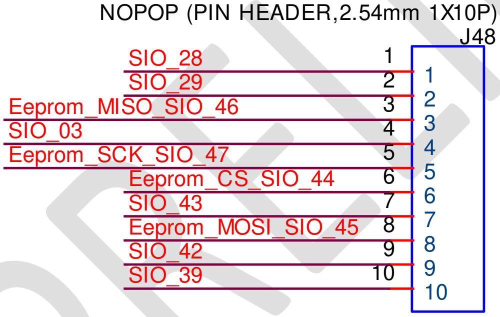
- DEFAULT type is DIO (Digital Input or Output) or UART (on fixed pins)
- ALTERNATE type is either AIN (Analog Input ADC), I2C, SPI, QSPI, DIO (on fixed pins), PWM, FREQ, and NFC
- DIO or AIN functionality is selected using the GpioSetFunc() function in *smart*BASIC
- I2C, UART, SPI, QSPI controlled by xxxOPEN() functions in *smart*BASIC
- 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 *smart*BASIC

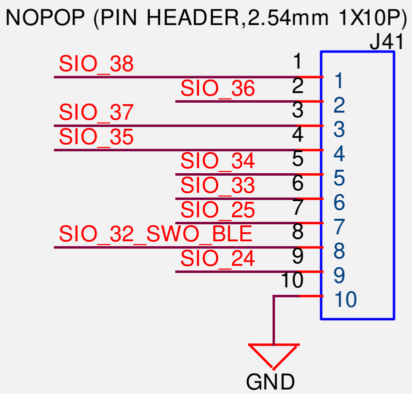
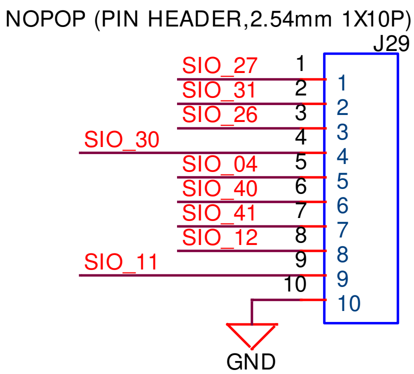
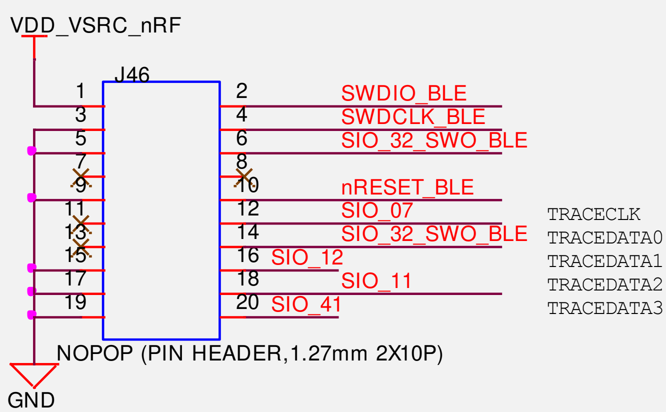
These breakout connectors can interface to a wide array of sensors, the BL654 is user configurable through the *smart*BASIC application script to change each SIO pin from the default function (DIO, UART) to alternate functions (AIN (ADC), I2C, SPI, QSPI, DIO), PWM, FREQ, and NFC. The BL654 development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

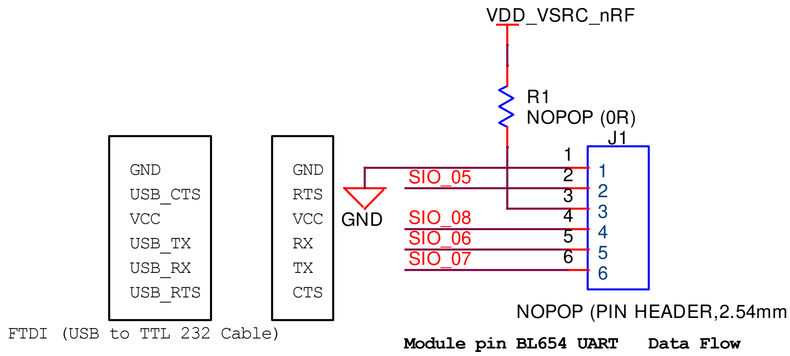
Table 8 shows the BL654 module pins that are brought out to plated through Holes (suitable for 2.54 mm pitch headers).

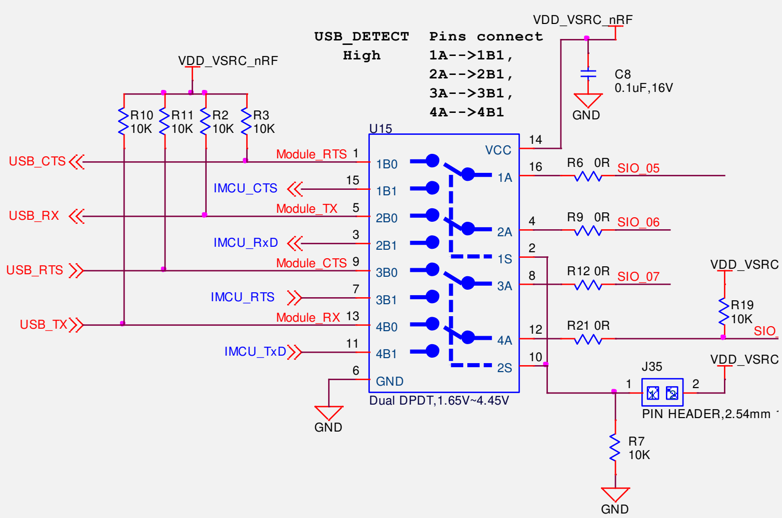
Table 8: Module pins exposed by plated through holes

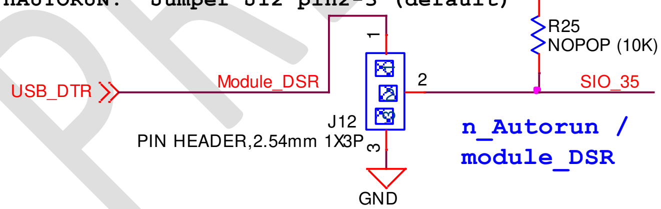
	Plated Through Holes or Header Connector	BL654 Module Signals Exposed
J44		BL654 pin plated holes for access SI0_ 00 SI0_ 01 J44 connects to SIO_00 and SIO_01 via 0R resistors R127 and R128. By default, the optional external 32.768 kHz crystal circuit is not connected to BL654 as SB8 and SB9 are open.

	Plated Through Holes or Header Connector	BL654 Module Signals Exposed
J47		BL654 pin plated holes for access: SI0_ 23 SI0_ 21 SI0_ 22 SI0_ 19 SI0_ 20 SI0_ 17 SI0_ 16 SI0_ 15 SI0_ 13 SI0_ 14
J26 **J37** **J45** **J39**		J26 Connects SIO_35 to LED1 J26 jumper fitted (default). J37 Connects SIO_14 to LED2 J37 jumper fitted (default). J45 Connects SIO_15 to LED3 J45 jumper fitted (default). J39 Connects SIO_16 to LED4 J39 jumper fitted (default).
J48		BL654 pin plated holes for access SI0_ 28 SI0_ 29 SI0_ 46 (connects to Eeprom_MISO) SI0_ 03 SI0_ 47 (connects to Eeprom_SCK) SI0_ 44 (connects to Eeprom_CS) SI0_ 43 SI0_ 45 (connects to Eeprom_MOSI) SI0_ 42 SI0_ 39

<table>
<tr><th></th><th>Plated Through Holes or Header Connector</th><th>BL654 Module Signals Exposed</th></tr>
<tr><td>J41</td><td>
</td><td>BL654 pin plated holes for access
SI0_ 38
SI0_ 36
SI0_ 37
SI0_ 35
SI0_ 34
SI0_ 33
SI0_ 25
SI0_ 32
SI0_ 24
GND</td></tr>
<tr><td>J29</td><td>
</td><td>BL654 pin plated holes for access
SI0_ 27
SI0_ 31
SI0_ 26
SI0_ 30
SI0_ 04
SI0_ 40
SI0_ 41
SI0_ 12
SI0_ 11
GND</td></tr>
<tr><td>J46</td><td>
</td><td>J46 is NOPOP but is compatible to same connector on Nordic development board and brings out same signals.</td></tr>
</table>

<table>
<tr><th></th><th>Plated Through Holes or Header Connector</th><th>BL654 Module Signals Exposed</th></tr>
<tr><td>J1</td><td>


<pre>
Module pin BL654 UART Data Flow
SIO_06   Module_TX   Output
SIO_08   Module_RX   Input
SIO_05   Module_RTS Output
SIO_07   Module_CTS Input
</pre>
</td><td>Serial Port plated holes for access</td></tr>
<tr><td>J35</td><td>
<pre>
BL654 UART   J35
routed to FTDI: No Jumper in J35 (default).
routed to Atmel: Fit Jumper in J35.
</pre>


</td><td>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</td></tr>
<tr><td>J12</td><td>
<pre>
Develop: Jumper J12 pin2-1
nAUTORUN: Jumper J12 pin2-3 (default)
</pre>


</td><td>Connects SIO_35 (nAutoRUN) to FTDI DTR
Default jumper fitted in J12 pin 2-3.</td></tr>
</table>