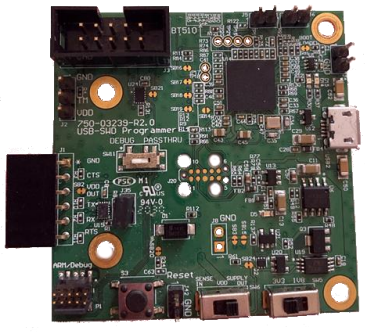
Using the USB-SWD Programmer

USB-SWD Programmer (Kit part # 453-00062-K1)

Application Note *v1.0*

1 INTRODUCTION



The USB-SWD Programmer board is a multifunction debugging/programming/passthrough/UART board which can be used to program Laird Connectivity's range of wireless modules and sensors including BL600, BL620, BL651, BL652, BL653, BL653µ, BL654, BL654PA, RM1xx, BT510, BT6x0, BT710, Pinnacle 100, and MG100 devices. Features include the following:

- USB connection to host PC exposes CMSIS-compatible debug interface
- Mass storage device with drag-and-drop hex/bin file programming support
- Programmer firmware upgrade support
- ARM 9-pin debug connector for connecting to modules
- 10-pin IDC connector for use with Tag-Connect TC2050-IDC cable for connecting to BT510 sensors (this port is not for connecting to any other devices)
- UART (with optional hardware flow control on the module side only) supporting the following baud rates: 1200, 2400, 4800, 9600, 14400, 19200, 38400, 57600, 115200, 230400, 460800, 921600, 1000000
- 3.3v or 1.8v output voltage to target the host module (up to 100 mA can be provided) or target can supply a voltage to programming board between 1.8v-3.6v as the programming reference voltage
- Reset button for resetting target device
- Unrestricted royalty-free programming of any nRF51/nRF52 module
- Passthrough mode for allowing a connected BT510 to be used with an external programmer such as a JLink

An annotated picture of the USB-SWD programmer board is shown below:

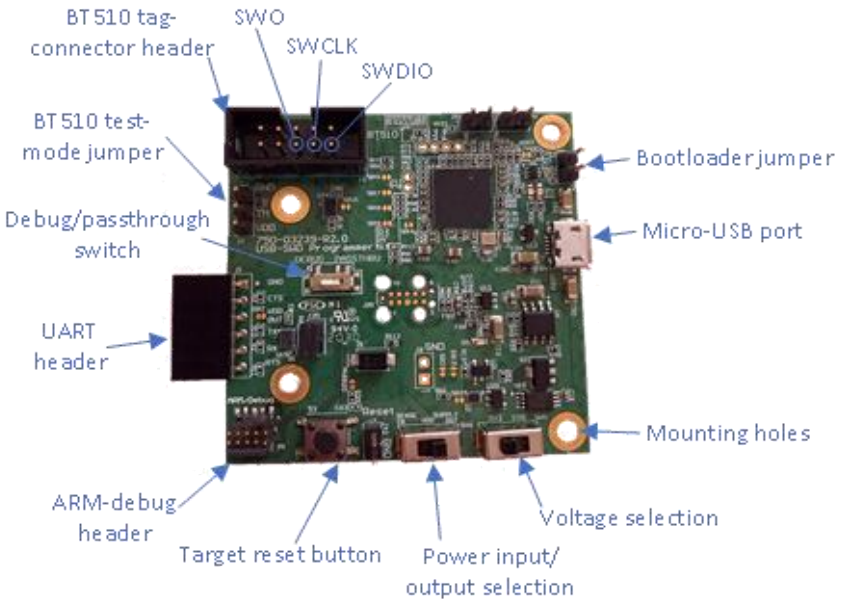


Figure 1: USB-SWD programmer board

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

This document describes how to use the USB-SWD programmer to program and debug modules, change configuration options, and upgrade the programmer firmware.

2 REQUIREMENTS

To use the SWD programmer, you need the following:

- SWD Programmer board (including micro USB cable and debug cables)
- PC (Windows/Linux/Mac supported; this guide is based on Windows 7)
- Target module, board, or sensor to program
- UwTerminalX (available at https://github.com/LairdCP/UwTerminalX/releases)
- pyOCD (instructions for install available on https://github.com/pyocd/pyOCD)
- MBED serial port driver (for Windows 7 only, newer versions of Windows do not need this driver, available at https://os.mbed.com/docs/mbed-os/v6.8/program-setup/windows-serial-driver.html)

3 SETUP

3.1 MBED Serial Port Driver (Windows 7 only)

If using Mac, Linux, or a version of Windows newer than Windows 7, please skip this step and start at the Hardware Setup section. Windows 7 requires an external driver to be installed on the system to communicate with the USB-SWD programmer board. This can be downloaded from the MBED website: https://os.mbed.com/docs/mbed-os/v6.8/program-setup/windows-serial-driver.html

Once downloaded, launch the installer and follow the instructions on screen to install the driver on your system. Administrator access is required for the install to complete. Once installed, the USB-SWD programmer is usable and a serial port should be visible along with a CMSIS debug interface.

3.2 Hardware Setup

The USB-SWD programmer can be configured to provide remote power to a device/module at 1.8v or 3.3v, or can use the reference voltage of the target device (1.8v-3.6v supported).

Note: The USB-SWD programmer board generates its own power internally and, if set to follow the voltage of the target, the target voltage is used as a reference only and does not power the USB-SWD programmer board.

3.2.1 Supplying Power to External Module/Device

To supply power to an external module or device follow these steps:

1. Set the SW6 switch to the *Supply out* position.
2. Set the SW5 switch into the 3V3 or 1V8 position, depending upon what voltage the target should be powered at.

   For nRF51 based devices, the voltage must be 3.3v for flash writing/erasing to occur. For nRF52 devices, the flash can be written/erased at either 3.3v or 1.8v.

   Refer to your module/device datasheet for information on voltage operating ranges as some modules (such as the Pinnacle 100) cannot be operated at these voltages.

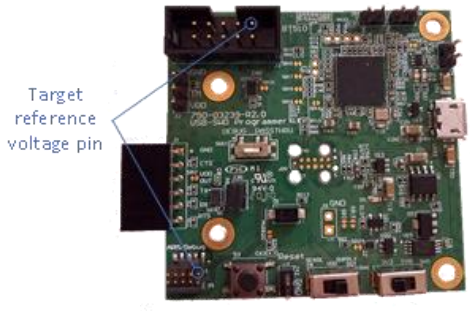
Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

Note: The target device should draw less than 100 mA of current using a USB port that can supply at least 250 mA of power (including the power usage of the USB-SWD programming board and target device). Going beyond this limit may cause damage to the target module/device, USB-SWD programming board, or PC.

When programming BT510 modules, this setup must be used. The voltage should be set to 3.3v to closely match the coin cell which operates at 3v (the battery cannot be connected whilst the debugger is connected, therefore there is no issue of a voltage mismatch or back-powering of the coin cell).

3.2.2 Referencing External Module/Device Voltage

To use the target's voltage source as a reference voltage, set the SW6 switch to the *Supply in* position and set the SW5 switch into the 1V8 position. The reference voltage of the target must be supplied on pin 1 of the ARM-debug connector (P1) or pin 1 of the BT510 IDC connector (J3) with a suitable ground connection to the ground of the USB-SWD programmer board.



The USB-SWD programmer board operates at the voltage set on SW5 until the voltage of an external device is sensed. We recommend that you keep this switch on 1.8v if devices operating at less than 3.3v are connected. This prevents any power passing from the USB-SWD programmer board to the target device before the USB-SWD programmer board's reference voltage circuitry has updated with the voltage being synchronised to all power-systems on the board.

4 Setting Options

The USB-SWD programmer has various volatile and non-volatile options that can be set by copying empty files with specific filenames to the USB mass storage device. There is additional information about this feature on the MBED DAPLINK github site: https://github.com/ARMmbed/DAPLink/blob/master/docs/MSD_COMMANDS.md

4.1 Volatile Settings

Table 1 displays the supported actions (volatile, these are reset to defaults when the USB-SWD programmer is powered up) for the USB-SWD programmer.

Table 1: Supported actions

Filename	Function	Checking
start_bl.act	Enters bootloader mode on the USB-SWD programmer so that the firmware can be upgraded	If the red LED and green LED are on constantly (and the mass storage USB drive is named LC_SWDMAINT) then the device is in bootloader mode.
start_if.act	Enters normal mode on the USB-SWD programmer (if device is in bootloader mode and BOOTLOADER header pin is not fitted)	If the red LED and green LED are both off or just one is on (and the mass storage USB drive is not named LC_SWDMAINT) then the device is in normal mode.
refresh.act	Closes the USB mass storage device and re-opens it	

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

Filename	Function	Checking
erase.act	Erases the flash of the connected module (if QSPI/SPI flash is connected to target device and option is enabled in firmware, this also erases the flash memory contents of said flash chip)	
page_off.act	Performs a full-chip erase before programming a file via drag and drop programming to the connected module (non-persistent configuration option – reverts to persistent configuration when power of programmer is cycled. See non-volatile configuration options for details)	After USB mass storage device refresh, open DETAILS.TXT, check the *Page erasing* line (should be 0 for active)
page_on.act	Performs sector erases before programming a file via drag and drop programming to the connected module (non-persistent configuration option – reverts to persistent configuration when power of programmer is cycled. See non-volatile configuration options for details)	After USB mass storage device refresh, open DETAILS.TXT, check the *Page erasing* line (should be 1 for active)
device.act	Queries information from the connected device and puts this information into the LASTDEV.TXT file	Open LASTDEV.TXT after USB drive refreshes. It is populated with information from the attached module. ***Note:*** *If readback protection is enabled then this information cannot be retrieved.*

4.2 Non-Volatile Settings

Table 2 displays the supported configuration options (non-volatile, these are persistent even when the USB-SWD programmer is powered off) for the USB-SWD programmer:

Table 2: Supported configuration options

Filename	Function	Checking
nrfauto.cfg	Sets drop-and-drag programming mode to automatic nRF51/nRF52 detection	After USB mass storage device refresh, open DETAILS.TXT, check the Device line (should state automatic detection)
nrf51.cfg	Sets drop-and-drag programming mode to nRF51 device mode (BL600/BL620/RM1xx)	After USB mass storage device refresh, open DETAILS.TXT, check the Device line (should state nRF51)
nrf52.cfg	Sets drop-and-drag programming mode to nRF52 device mode (BL65x/BT510/Pinnacle 100/MG100)	After USB mass storage device refresh, open DETAILS.TXT, check the Device line (should state nRF52)
flow_off.cfg	Disables hardware flow control on the USB-SWD UART interface ***Note:*** *Hardware flow control cannot be enabled on the USB serial port as the CDC driver does not support flow control, and so is always disabled.*	After USB mass storage device refresh, open DETAILS.TXT, check the UART HW Flow option (should be 0)

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

Filename	Function	Checking
flow_on.cfg	Enables hardware flow control on the USB-SWD UART interface ***Note:*** *Hardware flow control cannot be enabled on the USB serial port as the CDC driver does not support flow control, and so is always disabled. The state of the CTS pin is reflected in the DSR signal.*	After USB mass storage device refresh, open DETAILS.TXT, check the UART HW Flow option (should be 1)
ovfl_off.cfg	Disables serial port data overflow reporting on the serial CDC port If there is an overflow on the UART, there will be no response on the PC.	After USB mass storage device refresh, open DETAILS.TXT, check the UART Overflow option (should be 0)
ovfl_on.cfg	Enables serial port data overflow reporting on the serial CDC port If there is an overflow on the UART, a message is sent to the CDC port on the PC.	After USB mass storage device refresh, open DETAILS.TXT, check the UART Overflow option (should be 1)
page_off.cfg	Turns page/sector erasing off and performs a full-chip erase before programming via drag and drop programming as the default when the programming board is powered up (does not change active configuration. See volatile configuration for details)	After USB mass storage device refresh, open DETAILS.TXT, check the Page erasing line (should be 0 for non-vol)
page_on.cfg	Turns page/sector erasing on and only erases sectors that are specified for writing before programming via drag and drop programming as the default when the programming board is powered up (does not change active configuration. See volatile configuration for details)	After USB mass storage device refresh, open DETAILS.TXT, check the Page erasing line (should be 1 for non-vol)
last_off.cfg	Disables listing the details of the last device programmed/erased/queried in the LASTDEV.TXT file	After USB mass storage device refresh, open DETAILS.TXT, check the Last device info option (should be 0)
last_on.cfg	Enables listing the details of the last device programmed/erased/queried in the LASTDEV.TXT file	After USB mass storage device refresh, open DETAILS.TXT, check the Last device info option (should be 1)
default.cfg	Restores the volatile and non-volatile configuration values to their defaults (see the Restoring To Default Configuration/Settings section for details on the default settings)	After USB mass storage device refresh, open DETAILS.TXT, check all the settings are back to their defaults.
qspi_on.cfg	Enables QSPI and SPI flash access functionality	After USB mass storage device refresh, open DETAILS.TXT, check the QSPI/SPI flash access line (should be 1)
qspi_off.cfg	Disables QSPI and SPI flash access functionality	After USB mass storage device refresh, open DETAILS.TXT, check the QSPI/SPI flash access line (should be 0)
spi_on.cfg	Enables QSPI and SPI flash access functionality	After USB mass storage device refresh, open DETAILS.TXT, check the QSPI/SPI flash access line (should be 1)
spi_off.cfg	Disables QSPI and SPI flash access functionality	After USB mass storage device refresh, open DETAILS.TXT, check the QSPI/SPI flash access line (should be 0)

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

Filename	Function	Checking
swd1mhz.cfg	Sets the SWD maximum clock speed to 1 MHz	After USB mass storage device refresh, open DETAILS.TXT, check the SWD line (should be 1 MHz)
swd2mhz.cfg	Sets the SWD maximum clock speed to 2 MHz	After USB mass storage device refresh, open DETAILS.TXT, check the SWD line (should be 2 MHz)
swd3mhz.cfg	Sets the SWD maximum clock speed to 3 MHz	After USB mass storage device refresh, open DETAILS.TXT, check the SWD line (should be 3 MHz)
swd4mhz.cfg	Sets the SWD maximum clock speed to 4 MHz	After USB mass storage device refresh, open DETAILS.TXT, check the SWD line (should be 4 MHz)
swd5mhz.cfg	Sets the SWD maximum clock speed to 5 MHz	After USB mass storage device refresh, open DETAILS.TXT, check the SWD line (should be 5 MHz)
swd6mhz.cfg	Sets the SWD maximum clock speed to 6 MHz	After USB mass storage device refresh, open DETAILS.TXT, check the SWD line (should be 6 MHz)

By default, the USB-SWD programmer is set to automatically detect if an nRF51/nRF52 device is detected and program it, with hardware flow control being disabled.

Note: The nRF51/nRF52 selection is for the USB mass storage programming only. If using pyOCD then this option is ignored and pyOCD sends the commands to select the correct device as supplied on the command line.

5 Reading Module Information

The USB-SWD programmer supports querying the attached device to retrieve information on the device which includes silicon revision, flash/RAM size, and other details. If readback protection is enabled, then this information cannot be retrieved. To read module information, follow these steps:

1. Attach the target device to the USB-SWD programmer.
2. Plug the SWD programmer into the PC with a micro USB cable.
3. Ensure the programmer is set to the correct target device, open DETAILS.TXT, and check if it is set for nRF51 or nRF52 or automatic detection. If it is set to the wrong device then create an empty file on the host computer named *nrf51.cfg, nrf52.cfg*, or *nrfauto.cfg* (depending on target device). Copy this file and paste it in the USB-SWD programmer mass storage device drive
4. Create an empty file on the host computer named *device.act*. Copy this file and paste it in the USB-SWD programmer USB mass storage device.
5. The USB-SWD programmer queries the attached device and refresh the USB mass storage device.
6. Once complete, open LASTDEV.TXT from the USB mass storage device.
7. If the module information could be read correctly then the information about the device is displayed here.

The Variant field displays the revision of silicon in the module (nRF52 only). This information can be used to check SDK support details and errata listings. Refer to the Nordic documentation for your part for details on this information.

Note: If this feature is enabled with the last_on.cfg option then any flash erase/program on the target module also generates the LASTDEV.TXT file information. The information returned during this time is the configuration values prior to programming and may have changed during the erase/program process.

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

6 Programming Via Mass Storage Device (Drag and Drop)

The USB-SWD programmer allows connected modules to be programmed by copying a .hex file to a mass storage USB device. Follow the steps below to program a hex file.

6.1 nRF52-Based Modules (BL65x/BT510/Pinnacle 100)

1. Ensure the target voltage and input/output mode is set correctly.
2. Connect the target module to the programming board.
3. Plug the USB-SWD programmer board into the PC with a micro USB cable.
4. If the USB-SWD programmer was switched to nRF51 mode, create an empty file somewhere on the host computer named *nrf52.cfg* or *nrfauto.cfg* (depending on if automatic detection or forced-nRF52 support is required), copy this file and paste it into the USB-SWD programmer USB mass storage device. The device resets, goes back into the USB drive, opens DETAILS.TXT, and ensures that the Device line is updated.
5. Copy the .hex file with the new firmware on to the USB mass storage device. It programs it to the module. If there are multiple hex files (such as one for a softdevice and one for an application), then these must be copied one at a time and the current file copy must complete before the next one can begin.
6. The target module should reset and begin running the new application. At this point the USB-SWD programmer can be disconnected.

Note: QSPI flash programming on the nRF52840 is supported and verified for the Pinnacle 100 only. Compatibility with different QSPI flash chips is not guaranteed. If the QSPI chip has a register/command for enabling QSPI mode, then this must be set/enabled by code on the nRF52840 prior to downloading a hex file. QSPI flash programming is only available on nRF52840-based modules and begins at the memory-mapped address for QSPI access on the nRF52840, 0x12000000.

6.2 nRF51-Based Modules (BL600/BL620/RM1xx)

1. If the USB-SWD programmer board needs to supply power to the target device, ensure that the voltage output is set to output and on 3.3v, as nRF51 devices do not support 1.8v flash programming.
2. Connect the target module to the programming board.
3. Plug the USB-SWD programmer board into the PC with a micro USB cable.
4. If the USB-SWD programmer was not switched to nRF51 mode, create an empty file somewhere on the host computer named *nrf51.cfg* or *nrfauto.cfg* (depending on if automatic detection or forced-nRF51 support is required). Copy this file and paste it into the USB-SWD programmer USB mass storage device. The device resets, goes back into the USB drive, opens DETAILS.TXT, and ensures that the Details line is updated.
5. Copy the .hex file with the new firmware on to the USB mass storage device. This programs it to the module. If there are multiple hex files (such as one for a softdevice and one for an application), then these must be copied one at a time and the current file copy must complete before the next one can begin.
6. The target module should reset and begin running the new application. At this point the USB-SWD programmer can be disconnected.

Note: SPI flash programming on the nRF51822 is supported and verified for the RM1xx only, compatibility with different SPI flash chips is not guaranteed. SPI flash programming is only available on nRF51822-based modules and begins at address 0x12000000.

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

7 Programming Via pyOCD

pyOCD is a python-based command line utility which allows for the programming and debugging of code using the USB-SWD programmer board. To program a hex file to a module, follow these steps:

1. Set the correct voltage and power options for your module on the SWD programmer board.

Note: For target nRF51 devices, the power must be set to output and on 3.3v, as nRF51 devices do no support 1.8v flash programming).

Note: Programming using pyOCD does not require that the programmer be set in the correct nRF51/nRF52 mode for the target device as pyOCD sends specific commands for the attached module.

2. Connect the target module to the programming board.
3. Plug the SWD programmer board into the PC with a micro USB cable.
4. Open a command prompt or terminal.
5. Use pyOCD to program the module. The command line options are: pyocd flash -t <TARGET> -e <ERASETYPE> <FILENAME>
   where <TARGET> can be nrf5, nrf52 or nrf52840
   where <ERASETYPE> can be chip (erases the whole chip before programming) or sector (only erases sectors where data is to be written which is the default)
   where <FILENAME> is a .hex file for the target module
6. If there is an issue during programming, it could be related to the flash programming speed, the speed can be reduced by using the -f <FREQUENCY> option. Try starting at 5M and decrementing by 1. If programming still fails at 1 MHz, then the setup of the programming board and target module should be investigated.

Note: When using pyOCD, SPI/QSPI flash devices cannot be programmed.

8 Debugging via pyOCD

pyOCD can be used as a debugging tool by acting as a GDB server. It can then be used to step through code, set breakpoints, and use other debugging functionality.

This process assumes that you are already familiar with using the ARM GCC utilities and currently have them set up on your system. The latest versions can be downloaded from https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/gnu-rm/downloads).

To begin debugging using pyOCD, follow these steps:

1. Set the correct voltage and power options for your module on the SWD programmer board

Note: For target nRF51 devices, the power must be set to output and on 3.3v because nRF51 devices do no support 1.8v flash programming.

Note: Debugging using pyOCD does not require that the programmer be set in the correct nRF51/nRF52 mode for the target device as pyOCD sends specific commands for the attached module.

2. Connect the target module to the programming board.
3. Plug the SWD programmer board into the PC with a micro USB cable.
4. Open a command prompt or terminal.

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

5. Use pyOCD to start a GDB server with the module. The command line options are: pyocd gdbserver -t <TARGET> where <TARGET> can be nrf51, nrf52 or nrf52840.
6. If there is an issue during debugging, it could be related to the flash programming speed.

   The speed can be increased or reduced by using the -f <FREQUENCY> option. Try starting at 5M and decrementing by 1. If programming still fails at 1 MHz then the setup of the programming board and target module should be investigated. Speeds can also be increased up to 8M for faster debugging.
7. Launch ARM GCC GDB from where your files to debug are located.
8. Connect from GDB to the server by sending the following command: target remote localhost:3333.

   You are now able to load/debug code on the target device. It is possible that external toolchains can be set up to allow debugging through a GUI. This is outside the scope of this document.

9 Programming a BT510 From Zephyr RTOS

The BT510 module can be programmed directly from a Zephyr RTOS build tree. For details and an example of how to do this, follow the instructions on the Zephyr documentation site: https://docs.zephyrproject.org/latest/boards/arm/bt510/doc/bt510.html

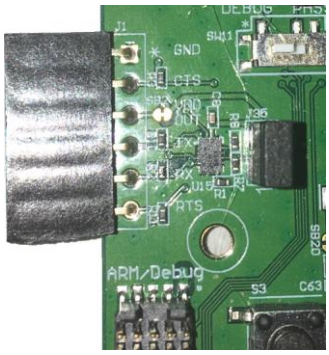
A boards file is included with Zephyr which maps the sensors/peripherals and GPIOs out for software development. We recommend that a complete application should have the application firmware and the mcuboot bootloader programmed so that the software on the unit can be upgraded in the field (for example, using Bluetooth). The design of such an upgrade system is outside the scope of this document. Details and information on this can be found on the Zephyr project website.

10 Using The SWD Programmer UART

The USB-SWD Programmer features a UART (J1) which can be used to connect to modules and other devices. A jumper must be fitted on J35 to route the pins to the UART interface, otherwise they are disconnected. Note that this is a UART interface which is exposed to the host PC, no external USB/PC UART cable needs to be connected to this PC for access of the UART by the PC.

Note: Power is supplied via this interface (and subject to the same constraints as listed in the Supplying Power to External Module/Device section) on the V_{DD} pin at the level set by the switches. There is a solder bridge (SB2) which can be cut to prevent this if it is not desired.

The UART operates over the voltage range of 0v to V_{DD} (set by the switches) as a TTL UART and is not compatible with an RS232 UART. Connecting RS232 lines directly to the UART interface is likely to cause irreparable damage to the programming board and devices to which it is connected. A voltage level translation circuit is required for this operation which is outside the scope of this document.



The pinout of this connector is displayed on the board, and is as follows:

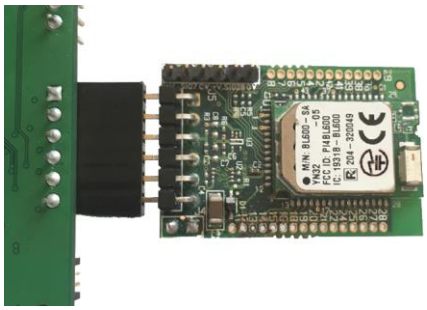
Pin	Function	Perspective (USB-SWD Programmer Board)
1	Ground	

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

Pin	Function	Perspective (USB-SWD Programmer Board)
2	CTS	Input
3	V_{DD}	Input/Output (SW6)
4	TX	Output
5	RX	Input
6	RTS	Output

10.1 Bx600

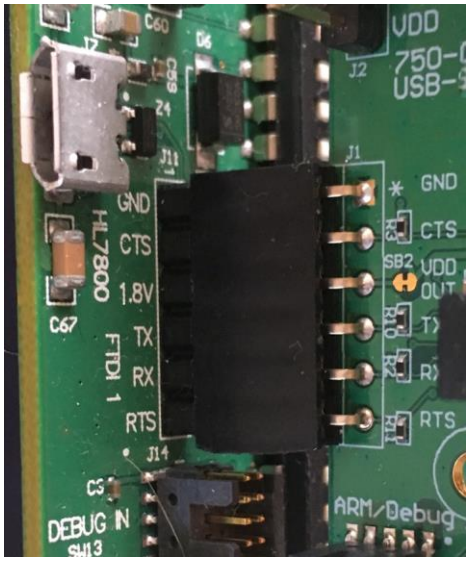
The Bx600 modules can be directly connected to the UART header by plugging it in. Note that the top of the Bx600 module (with the BL600 facing upwards) is inserted facing downwards into the USB-SWD programmer board (with the components of the USB-SWD programming facing upwards):



The Bx600 expects a baud rate of 9600 with flow control enabled by default. Due to silicon constraints of the nRF51, the SWD programmer should be operating at 3.3v, otherwise the flash of the nRF51 silicon in the connected BL600 module cannot be written to.

10.2 Pinnacle 100

The Pinnacle 100 development board can be directly connected to the UART header by plugging it in. The GND pin names should align. For example, to the HL7800 debug UART port. Note that the operating voltage must be set to 1.8v or damage may be caused to the equipment.



Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

This debug UART can then be used to update firmware of the modem. For details, please see the documentation on the Pinnacle 100 website.

10.3BT510

The BT510 sensor can be debugged and controlled via the UART, when running the official firmware. To set the BT510 up for debugging, place a jumper on J2 from TM to GND with the BT510 connected as shown in Figure 2. Open a terminal utility such as UwTerminalX with baud rate set to 115200, flow control set to none, parity set to none and open the port, then press the reset button on the USB-SWD programmer to reboot the BT510, debug data will be shown on the terminal utility's screen.

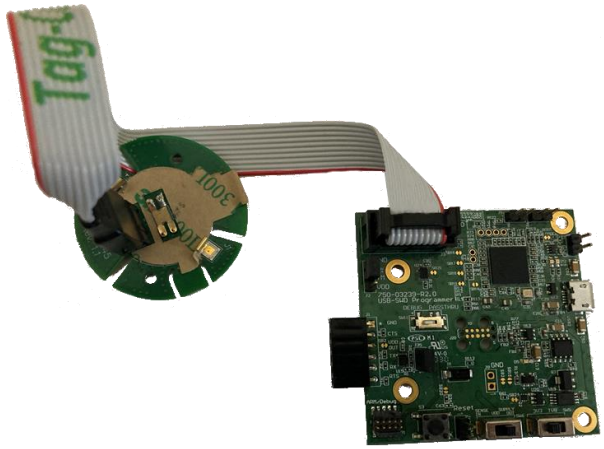


Figure 2: BT510 connected to USB-SWD Programmer Board with test mode enabled

11 BT510 PASSTHROUGH MODE (FOR JLINK)

The USB-SWD programmer board can be used with a BT510 to provide a debugging connection to the target board via a Segger JLink (or other external programmers/debuggers). In this mode the USB-SWD programmer board acts as a passthrough device for the signals and supplies power to the BT510. To use the USB-SWD programmer board in passthrough mode:

1. Set the USB-SWD programmer board to supply power to the target module at 3.3v.
2. Connect the BT510 up to the programmer board by using the TagConnect TC-2050 IDC cable. Remove the back plate from the BT510, remove the battery, insert the TagConnect into the BT510, and insert the IDC connector into the SWD programmer board.
3. Change SW1 to be in the PASSTHROUGH position.
4. Connect the target JLink (or other device) to the USB-SWD programmer board using the ARM/Debug header.
5. Connect the USB-SWD programmer up to the computer via a micro USB cable.
6. The target device can now be debugged/programmed using the external programmer.

Note: The two UART pins are still connected to the USB-SWD programmer board and can be used as normal from the CDC serial port. If this feature is not desired, then the J35 header can be removed. This disconnects the UART pins from the target BT510 device. In this mode, the J1 UART pins can be connected to another device and will work.

To use the USB-SWD programming board in non-passthrough mode, SW1 needs to be moved to the DEBUG position.

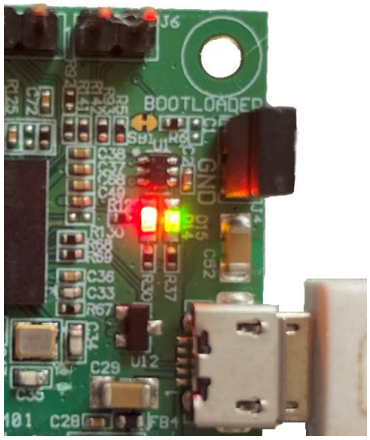
Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

12 Upgrading Programmer Firmware

The USB-SWD programmer board has an inbuilt bootloader which allows the firmware to be upgraded. Details of the current firmware can be viewed by opening the DETAILS.TXT file on mass storage device when the programmer is plugged in. If there is a new firmware, then it is listed and available for download on the Laird Connectivity USB-SWD programmer website.

To upgrade the firmware on the USB-SWD programmer, remove all debug connectors and the USB cable from the board. Place a jumper on the BOOTLOADER header near the USB port.

Plug in the micro USB cable. Both the red and green LEDs illuminate at this point to indicate that the unit is in bootloader mode. Remove the jumper from the BOOTLOADER header pin.



The USB drive should be called LC_SWDMAINT. Copy the .bin upgrade file from the firmware upgrade package to this drive. The bootloader flashes the new firmware onto the device and reboots. Once complete, the USB-SWD programmer LEDs turns off and the programmer appears as normal.

13 Restoring To Default Configuration/Settings

To restore the USB-SWD programmer to default configuration and state, follow these steps:

1. Move SW11 into the DEBUG position.
2. Move SW6 into the SUPPLY OUT position.
3. Move SW5 into 3V3 position.
4. Remove any jumpers from J2.
5. Remove any jumpers from J4.

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

6. Remove any jumpers from J5.
7. Remove any jumpers from J6.
8. Ensure a jumper is fitted on J35.
9. Plug the USB-SWD programmer into a computer using a micro USB cable.
10. Create a new empty file on the host computer named DEFAULT.CFG.
11. Copy the DEFAULT.CFG file and paste it into the USB-SWD programmer mass storage device drive.

The programmer remounts itself and all settings are restored to default.

The settings and configuration are now restored to factory defaults. Default settings of the USB-SWD programmer are as follows:

- Automatic reset – enabled
- Page erasing (non-volatile) – enabled
- UART hardware flow control – disabled
- UART overflow detection – enabled
- Device – nRF51/nRF52 automatic detection
- QSPI/SPI flash access – enabled
- Last device information – disabled
- Maximum SWD speed – 6 MHz

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

14 TROUBLESHOOTING

This section contains some common issues that may be encountered when using the USB-SWD programmer board and possible solutions for these problems.

14.1 Red LED remains on

Problem	The unit is connected to the computer and the red LED is on constantly, the green LED is not on.
Possible Resolution	Check if there is a NOLIC.TXT file in the virtual flash drive. If there is, please contact Laird Connectivity support with the details of this file.

14.2 Programming module fails

Problem	Attempting to program or erase a module fails and does not work.
Possible Resolution	Check that the target device is correctly connected to the USB-SWD programmer board and that the voltage level is set correctly for the target (nRF51-based modules (BL600 and RM1xx) do not support writing/erasing flash at 1.8v and must be programmed at 3.3v). Check that the programmer is set to the correct device type. If automatic detection fails or has issues, set the programmer to force a specific module type by creating an empty file *nrf51.cfg* for nRF51-based modules or *nrf52.cfg* for nRF52-based modules. Copy it to the virtual flash drive, then retry the write/erase process. If this still fails, try reducing the maximum SWD clock speed by creating an empty file *swd1mhz.cfg* and copying it to the virtual flash drive then retrying the write/erase process again.

14.3 MSD drive is not visible from computer or shows up as empty

Problem	The MSD drive does not appear on the computer or is visible but is empty as if there is no disk inserted into it.
Possible Resolution	Try power cycling the USB-SWD programmer board and ensure the voltage level is set up correctly for your module. If you are using Windows 7, ensure that the MBED serial port driver is installed as described in the MBED Serial Port Driver (Windows 7 only) section.

14.4 Erasing a Pinnacle 100/MG100 or device with a QSPI is very slow and/or times out

Problem	Writing to or erasing nRF52 devices with a QSPI memory chip can be very slow and/or time out.
Possible Resolution	The QSPI chip fitted in the Pinnacle 100/MG100 can take between 1-3 minutes to perform a full erase. We do not recommend that you perform a full erase of the QSPI chip. Try disabling QSPI/SPI flash access by creating an empty file *qspi_off.cfg* and copying it to the virtual flash drive. This hould prevent use of the QSPI erase function. **Note:** This entirely disables QSPI support.

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

14.5 Transferring firmware file to module is very slow

Problem	Transferring firmware files via the drag-and-drop interface takes a long time (much longer than expected)
Possible Resolution	This can be caused if the file is trying to write to QSPI/SPI flash and there is no flash attached, or there may be other hardware on these pins and the module is causing a voltage conflict. Try disabling QSPI/SPI flash access by creating an empty file *qspi_off.cfg* and copying it to the virtual flash drive.

14.6 SWD or resetting device does not work

Problem	After using pyOCD, SWD access or other functionality is not working.
Possible Resolution	At time of writing with pyOCD version 0.29.0, there appears to be a bug in that SWD is not configured but is dis-configured in some circumstances. There is currently no known workaround for this issue other than by disconnecting and reconnecting the USB-SWD programmer to the computer.

14.7 UART is not receiving data

Problem	Data is being transmitted or received on the UART but is not visible on the host PC.
Possible Resolution	Check that the J35 header has a jumper on it. If there is no jumper, the UART lines are disconnected from the target device/header. Check that the voltage source and level is correctly set for the module to which it is connected. If one side is operating at 1.8v and the other at 3.3v then UART communication is not possible.

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

15 ACRONYMS

Acronym	Description
CMSIS	Cortex-M Software Interface Standard
CDC	Communications Device Class
USB	Universal Serial Bus

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

16 Additional Information

Additional information on products referred to in this document can be located at the following URLs:

- **BL600 –** https://www.lairdconnect.com/wireless-modules/bluetooth-modules/bluetooth-42-and-40-modules/bl600-series-bluetooth-module
- **BL620 –** https://www.lairdconnect.com/wireless-modules/bluetooth-modules/bluetooth-42-and-40-modules/bl620-bluetooth-module
- **BL600 breakout boards –** https://www.lairdconnect.com/wireless-modules/bluetooth-modules/bluetooth-42-and-40-modules/bl600-bluetooth-breakout-boards
- **RM1xx –** https://www.lairdconnect.com/wireless-modules/lorawan-solutions/sentrius-rm1xx-lora-ble-module
- **BL651 –** https://www.lairdconnect.com/wireless-modules/bluetooth-modules/bluetooth-5-modules/bl651-series-bluetooth-module
- **BL652 –** https://www.lairdconnect.com/wireless-modules/bluetooth-modules/bluetooth-5-modules/bl652-series-bluetooth-v5-nfc-module
- **BL653 –** https://www.lairdconnect.com/wireless-modules/bluetooth-modules/bluetooth-5-modules/bl653-series-bluetooth-51-802154-nfc-module
- **BL653µ –** https://www.lairdconnect.com/wireless-modules/bluetooth-modules/bluetooth-5-modules/bl653-micro-series-bluetooth-51-802154-nfc-modules
- **BL654 –** https://www.lairdconnect.com/wireless-modules/bluetooth-modules/bluetooth-5-modules/bl654-series-bluetooth-module-nfc
- **BL654PA –** https://www.lairdconnect.com/wireless-modules/bluetooth-modules/bluetooth-5-modules/bl654pa-series-long-range-bluetooth-module
- **Pinnacle 100 –** https://www.lairdconnect.com/wireless-modules/cellular-solutions/pinnacle-100-cellular-modem
- **MG100 –** https://www.lairdconnect.com/iot-devices/iot-gateways/sentrius-mg100-gateway-lte-mnb-iot-and-bluetooth-5
- **BT510 –** https://www.lairdconnect.com/iot-devices/iot-sensors/bt510-bluetooth-5-long-range-ip67-temperature-sensor
- **BT6x0 –** https://www.lairdconnect.com/bt6x0-series
- **USB-SWD Programmer –** https://www.lairdconnect.com/usb-swd-programmer

17 Revision History

Version	Date	Notes	Contributor(s)	Approver
1.0	28 Apr 2021	Initial Release	Jamie Mccrae	Jonathan Kaye

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610