Implementing a Bootloader on the SAM D21 MCU

Table of Contents

Prerequisites

Hardware Prerequisites

- SAM D21 Xplained Pro Evaluation Kit
  - Part Number: ATSAMD21-XPRO
- One Micro USB cable (type A/Micro B)

Software Prerequisites

- Atmel® Studio 7
  - Version: 7.0.1645 or higher
- SAM-BA® Host (PC utility) and SAM-BA Monitor (MCU application)
  - Version: 2.18
- Atmel START
  - Version: 2018 – May (v1.2) or higher
- Internet connection

Audience: Beginner

Estimated Completion Time: 45 minutes

Introduction:

These labs show you how to use a bootloader on the SAM D21 microcontroller.

A bootloader gives you the ability to update the firmware in your microcontroller, without the need of a programmer/debugger (e.g., SAM-ICE™ JTAG Emulator). This is particularly useful for field updates. Just connect to the microcontroller's USB or UART interface and hold a specific I/O pin low during a reset to upload new firmware.

In lab 1, you will use Atmel START to create two new projects:

- LEDflasher0 blinks LED0 once every second
- LEDflasher1 blinks LED0 five times every second

You will upload these projects in later labs with the SAM-BA bootloader.

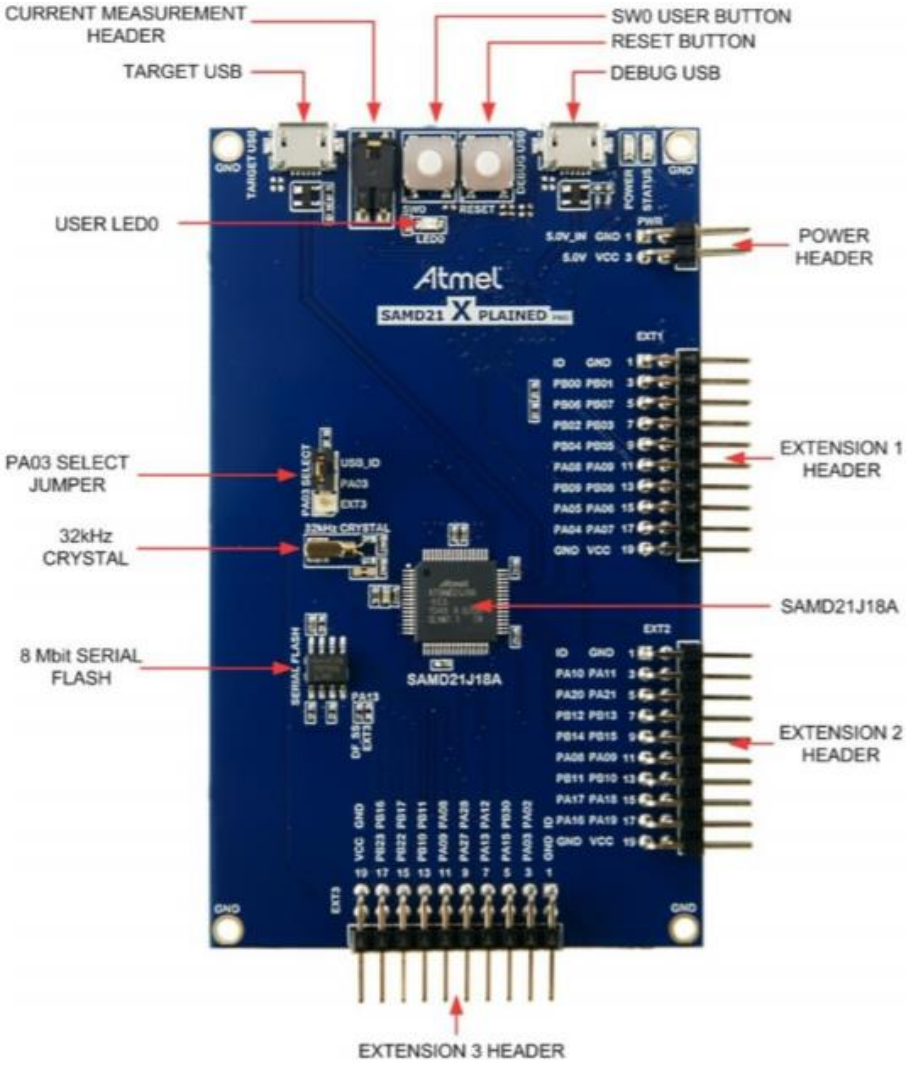
In lab 2, you will program the SAM D21 MCU with the SAM-BA Monitor (BA stands for boot assistance) application. The SAM-BA Monitor is used to communicate with the SAM-BA host PC utility. On reset, the SAM-BA Monitor will look for a Start condition (bootloader mode) on the UART and/or USB interfaces. When the Start condition is detected, the SAM-BA Monitor enters an infinite loop waiting for commands from the SAM-BA PC utility. The most essential command is the "upload this file" command.

In lab 3, you will use the SAM-BA Monitor to upload one of the LEDflasher applications to the MCU. You now have both the SAM-BA Monitor application and LEDflasher application loaded into the MCU.

In lab 4, you will use the bootloader to change one LEDflasher application with the other (i.e., update the firmware). You will use the SAM-BA Monitor to establish a connection with the PC application and upload the second LEDflasher application to the MCU.

The SAM D21 Xplained Pro Evaluation Kit is the hardware platform for these labs. The MCU on this board is an ATSAMD21J18A. It includes a user push button and LED, two USB micro-B connectors (one for program/debug), a current measurement header, and three extension headers for connecting Xplained Pro extension boards.

This board has an on-board programmer/debugger. No external programmer/debugger (e.g., SAM-ICE™ emulator) is required.



LAB 1:
Create two Application Projects using Atmel START

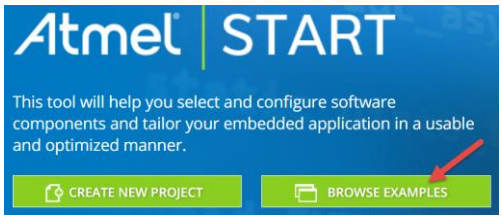
Overview:

In this first assignment, you will use Atmel START to create two simple new projects. One project blinks the LED once a second. The other blinks the LED five times a second.

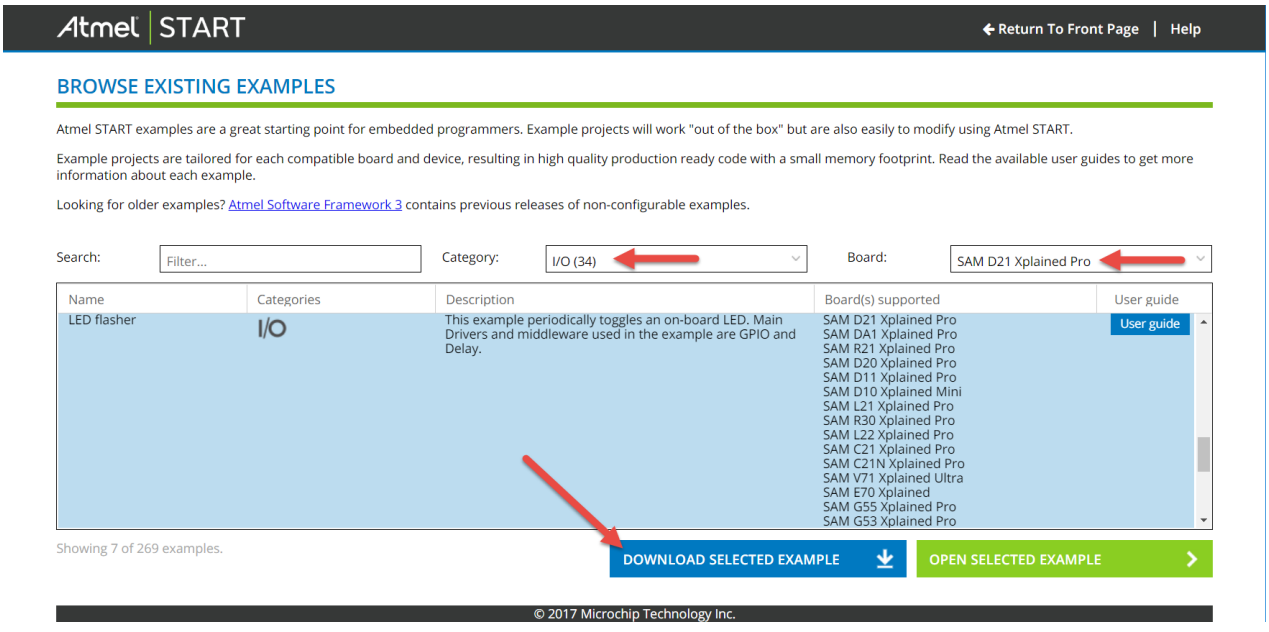
Atmel START (http://start.atmel.com) is an online project generation tool that will help you to select and configure software components, drivers, middleware and example projects. It performs all the MCU initialization and configuration tasks, allowing you to focus on your application code.

1.1: Download and build an example Atmel START project

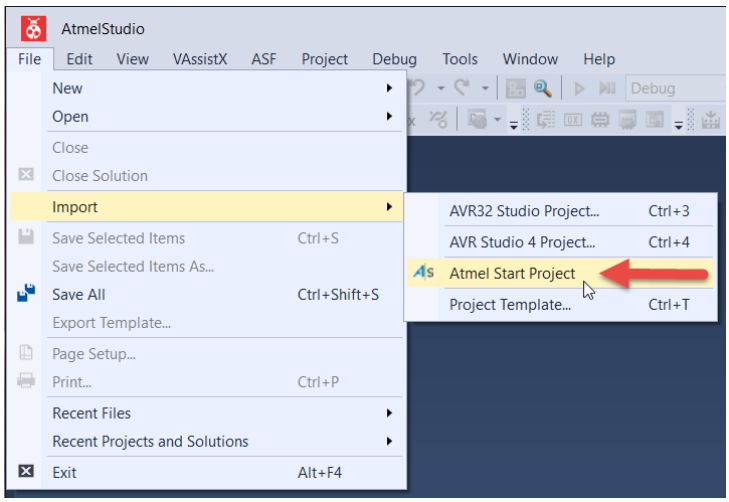
- Open a browser and go to http://start.atmel.com
- Select "Browse Examples"



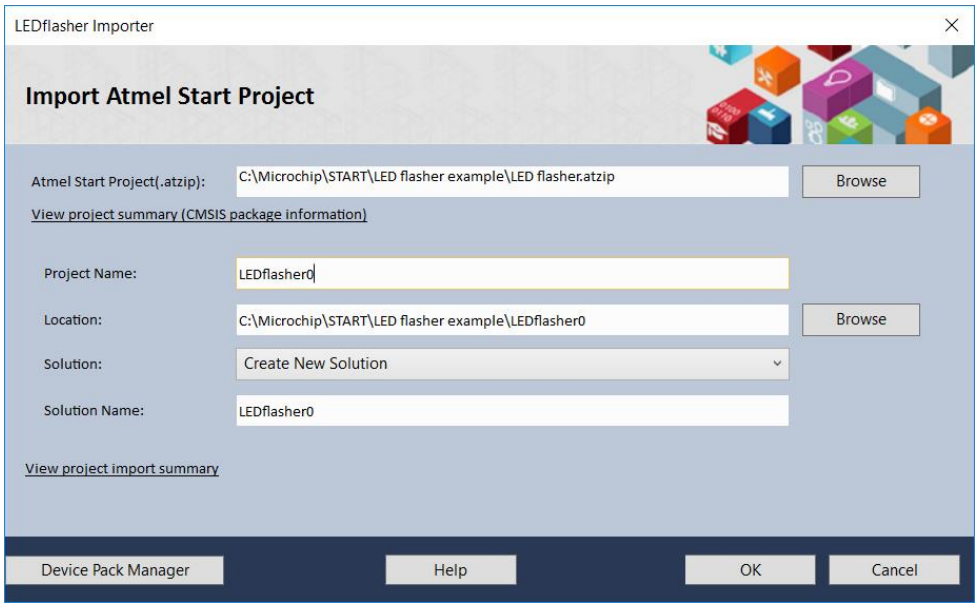
- In the Board field select "SAM D21 Xplained Pro"
- In the Category field select I/O
- Scroll down to the "LED flasher" example and click on it
- Click "DOWNLOAD SELECTED Example"
- Select "SAM D21 Xplained Pro" then "Select board"
- Browse to the folder you want to save the project in and click Save



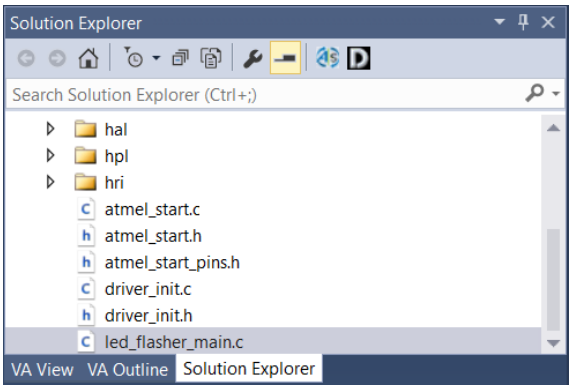
- Open the Studio 7 IDE
- Click on File > Import > Atmel Start Project



- Browse to the folder you saved the project in and select the atzip file as the Atmel START project.
- Provide a location and name for the project, then click OK



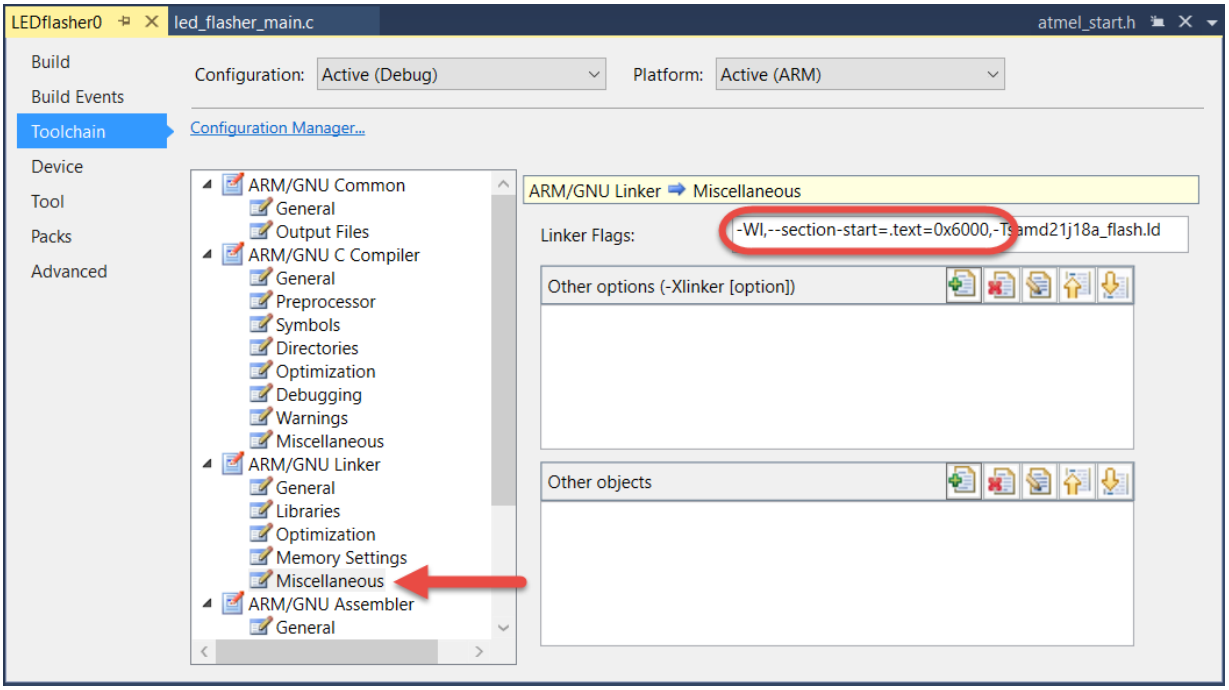
- Find the “led_flasher_main.c” file at the bottom of the Solution Explorer window. Double-click to open it.



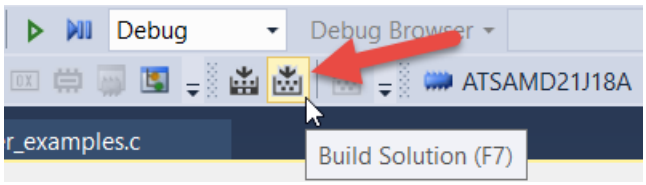
- Find the main function at the bottom of the file. Note this example project toggles LED0 once every 500 ms.

```
40
41  int main(void)
42  {
43      atmel_start_init();
44
45      while (true) {
46          delay_ms(500);
47          gpio_toggle_pin_level(LED0);
48      }
49  }
50
```

- Change the project build properties to locate the application start address at 0x6000
  - The SAM-BA Monitor needs to know where your application code lives, so it can start it if no bootloader event is found. This application start address is hardcoded in one of the SAM-BA Monitor header files (device_config.h).
  - Right-click on the project name (LEDflasher0 in this case) in the 'Solution Explorer' window, then select 'Properties'.
  - Select Toolchain > ARM/GNU Linker > Miscellaneous
  - Add these flags to the 'Linker Flags' window:
    - **-Wl,--section-start=.text=0x6000,**



- Generate the binary file (<project name>.bin) for this project by clicking on the 'Build Solution' icon.
    - This file will be found in the '<project name>/Debug' folder.



1.2: Create and build the second project

Create a duplicate project based on the one you just built.

- Import the same Atmel Start project
    - Click on File > Import > Atmel Start Project
- Provide a different name and location for the project and click OK

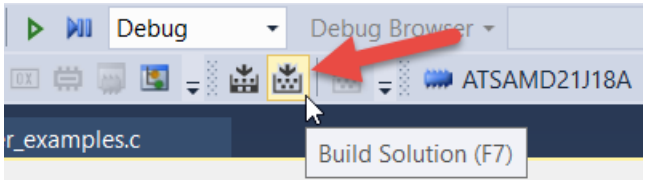
You should now have two open projects in the Solution Explorer window.

- Find the new 'led_flasher_main.c' file in the project you just created and open it.
- Change the parameter for the 'delay_ms' function from 500 to 100
    - This delay toggles LED0 every 100ms.
- Save the file (type Ctrl + s)

```
40
41  int main(void)
42  {
43      atmel_start_init();
44
45      while (true) {
46          delay_ms(100);
47          gpio_toggle_pin_level(LED0);
48      }
49  }
50
```

- As you did in the previous project, change the project build properties to locate the application start address at 0x6000
    - Right-click on the project name (LEDflasher1 in this case) in the 'Solution Explorer' window, then select 'Properties'.
    - Select Toolchain > ARM/GNU Linker > Miscellaneous
    - Add these flags to the 'Linker Flags' window:
        - **-Wl,--section-start=.text=0x6000,**

- Generate the binary file for this second project by clicking on the 'Build Solution' icon.



You now have two projects; each one with its own binary file. These binary files will be uploaded to the SAM D21 (via bootloader) in labs 3 and 4.

Lab 2: Program the SAM D21 XPRO with the SAM-BA Monitor

2.1: Download the SAM-BA Host Utility and Monitor Project

Go to Microchip's SAM-BA In-system Programmer website and download the SAM-BA Monitor (SAM project) and the SAM-BA Host PC utility.

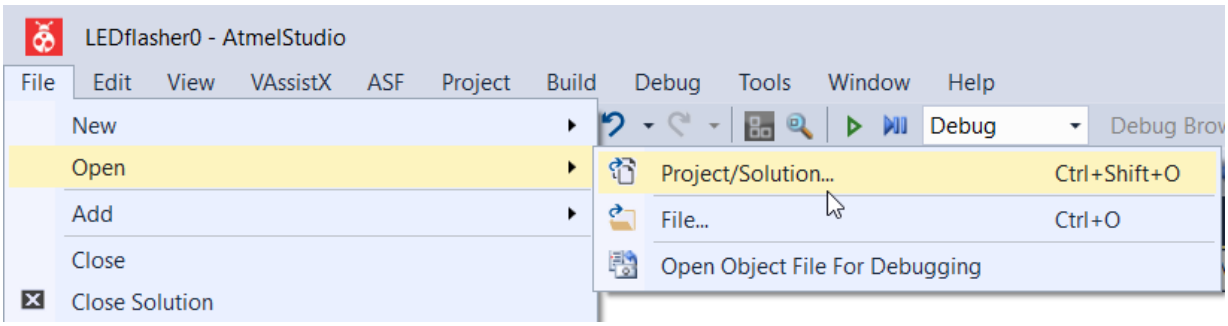
- Website >
  - SAM-BA MONITOR ROMLESS v2.18
  - SAM-BA 2.18 for Windows

- Install the SAM-BA Host PC utility and extract the SAM-BA Monitor zip file.

2.2: Open the SAM-BA Monitor Project in the Studio 7 IDE

Open the SAM-BA_MONITOR_ROMLESS project in the Studio 7 IDE:

- File > Open > Project/Solution…
- Browse to "SAM-BA_MONITOR_ROMLESS.atsln" file and click Open.



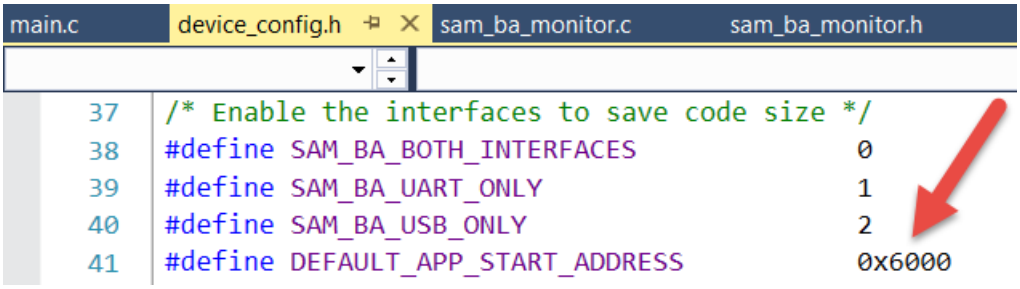
2.3: Change the Application Start Address in the SAM-BA Monitor

If the SAM-BA Monitor does not detect a bootloader event, it needs to know the starting address of your application so it can execute it.

The default application start address is 0x2000. You need to change this to 0x6000 to match both the SAM-BA Host utility, and your application project configuration (you configured this in the previous lab).

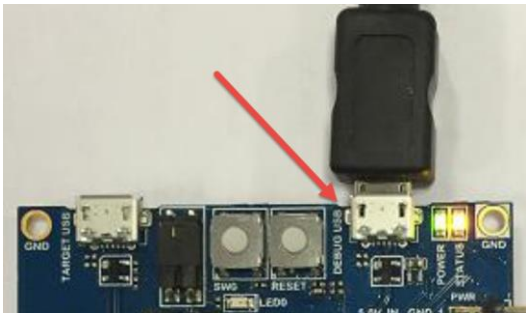
Note: The SAM-BA Host utility protects the bootloader area from being overwritten accidentally. It does this by specifying the size of the SAM-BA Monitor application. If you want to choose a different Monitor size, you can customize and regenerate applets for the SAM-BA Host. See application note AN2565 for details.

- Open **device_config.h**
  - Found in the 'Solution Explorer' window in this folder: 'SAM-BA_MONITOR_ROMLESS/device_config'
- Change the DEFAULT_APP_START_ADDRESS to 0x6000



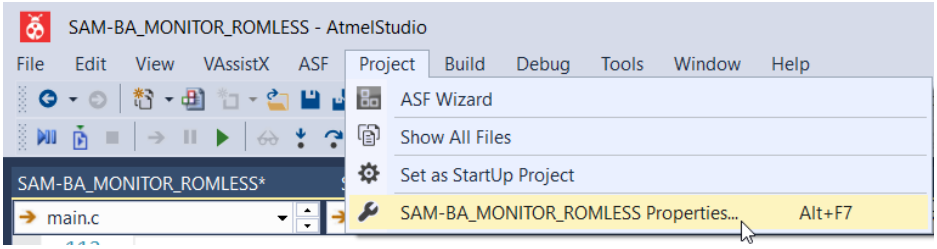
2.4: Program the SAM-BA Monitor onto the SAM D21 board

- Connect the SAM D21 Xplained Pro board to your computer using the DEBUG USB Connector
  - Please be patient as the driver installs (it may take a minute or so)

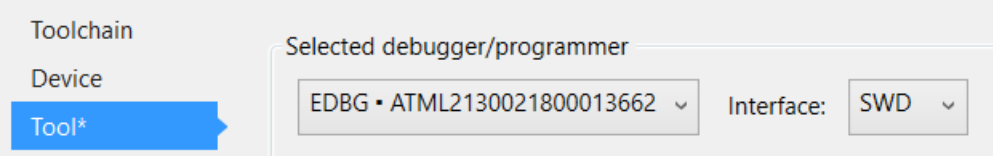


2.5: Select the Debugger/Programmer, Compile and Program the Project

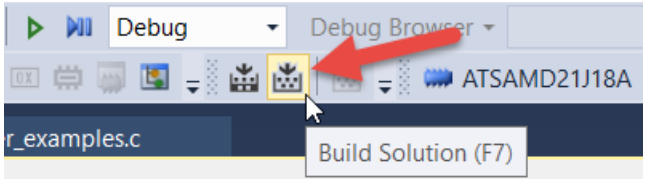
- Select the embedded (on-board) debugger found on the SAM D21 XPRO board as the Debugger/Programmer for the project:
  - Click on Project > Properties



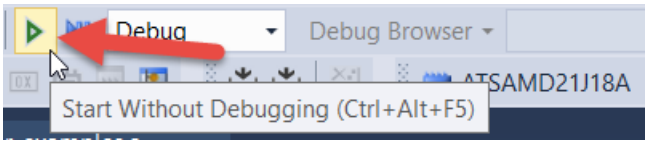
- Select Tool > EDBG as debugger/programmer and SWD as Interface:



- Compile the project by clicking on the Build Solution icon or by typing ‘F7’



- Program the application by clicking on the Start Without Debugging icon



Note: If the firmware on the evaluation board is out of date, a “Firmware Upgrade” window will appear asking you if you want to upgrade the firmware. Select “Upgrade” and allow the process to complete.

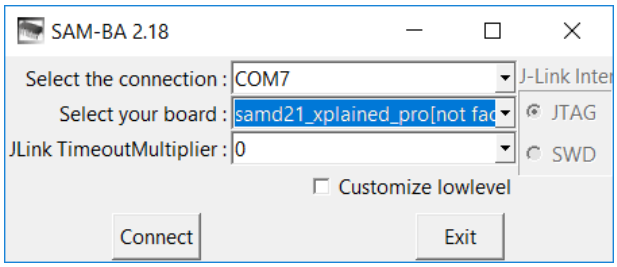
The SAMBA Monitor is now running on the evaluation board.

Lab 3:
Use SAM-BA Host (PC) to Upload Application to Board

3.1: Connect to SAM D21 XPRO Using the SAM-BA Host Utility

- Make sure your PC is still connected to the 'DEBUG USB' connector on the SAM D21 XPRO board.
- Open the SAM-BA Host utility
- Use the drop-down window to select the Window's COM port
- Use the drop-down window to select the board you are connecting to
    - samd21_xplained_pro[not factory programmed]

Note: Use Windows' Device Manager to find the COM port number assigned to the SAM D21 XPRO board (EDGB Virtual COM Port).

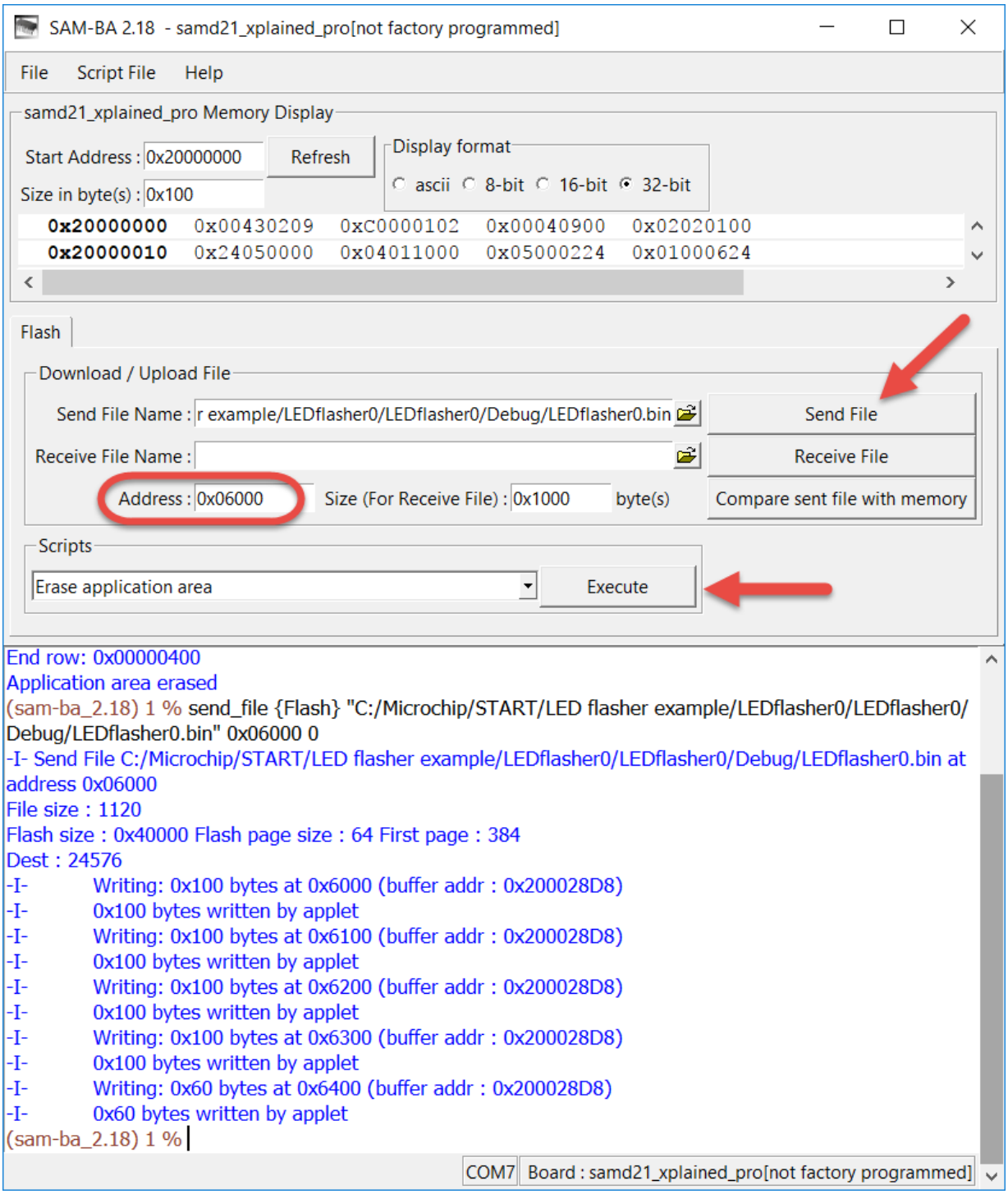


After clicking 'Connect', the yellow 'STATUS' LED on the SAM D21 XPRO will turn on. The SAM-BA Host and Monitor are now connected.

3.2: Upload an application to the board

- Before uploading your application to the board, **erase** the flash memory used to store the application.
    - Click the 'Execute' button in the Scripts field (see following image).
    - Note: The 'application' area in flash does not include the flash addresses containing the bootloader.

- Upload your project's binary file to the SAM D21 XPRO board
    - Specify the start address for your application as **0x6000.**
        - This is the same location you configured both application projects for in lab 1, and is the same location specified in the SAM-BA Monitor.
        - Note: If you want to choose a different start address, you can customize and regenerate applets for the SAM-BA Host. See app note AN2565 for details.
    - Click on the folder icon to browse to the folder containing the binary file for one of the 'LED flasher' projects you created in lab 1.
        - This file will be found in the '<project name>/Debug' folder.
    - Click the 'Send File' button.



3.3: Reset the board to verify you've successfully uploaded your project

- Push the RESET button (located beside the DEBUG USB connector) to exit the SAM-BA Monitor mode.
- LED0 should blink every second or five times a second.

Lab 4: Use Bootloader to Replace one Application with Another

Both the SAM-BA Monitor and your application are running on the SAM D21 XPRO board. In this lab, you will replace one LED flasher application (created in lab 1) with another.

4.1: How the SAM-BA Monitor Works

When the SAM D21 is reset, the SAM-BA Monitor checks if the starting application address is erased (i.e., no application has been uploaded). If it is erased, it stays in bootloader mode, allowing you to upload your application.

If the starting application is not erased (i.e., an application exists), the SAM-BA Monitor will check the status of one of its I/O pins to determine if it should stay in bootloader mode or execute the application.

If the I/O pin is low, the SAM-BA Monitor will keep the SAM D21 in bootloader mode until the next reset.

If the I/O pin is high, the SAM-BA Monitor will execute the application as if no bootloader exists.

4.2: Place the SAM D21 in bootloader mode

The SAM-BA Monitor can use the SAM D21's USB and/or UART interface(s) to upload an application's binary file. You configure the SAM-BA Monitor project for the interface you want to use (default configuration is both interfaces).

The SAM D21's USB interface is connected to the 'TARGET USB' connector.
The SAM D21's UART interface is connected (virtually) to the 'DEBUG USB' connector.

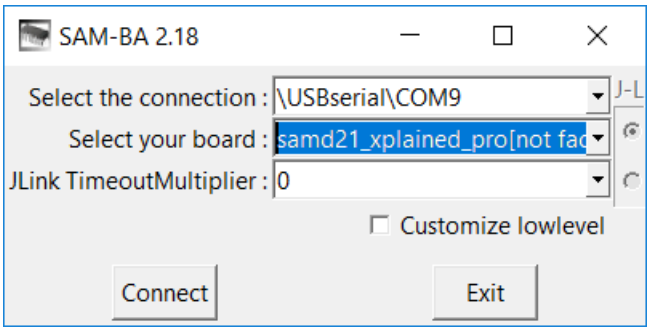
In the previous lab, you connected to the SAM D21 XPRO board via the 'DEBUG USB' connector, so you had a virtual connection to the UART on the SAM D21. In this lab, you will connect to the 'TARGET USB' connector to verify the SAM-BA Monitor's USB bootloader functionality.

- Move the USB cable from the 'DEBUG USB' connector to the 'TARGET USB' connector.
    - You will note the application you loaded in the previous lab starts to run automatically.
- Put the SAM D21 in bootloader mode
    - Press and hold the SW0 push button
    - Press and release the RESET push button
    - Release the SW0 push button

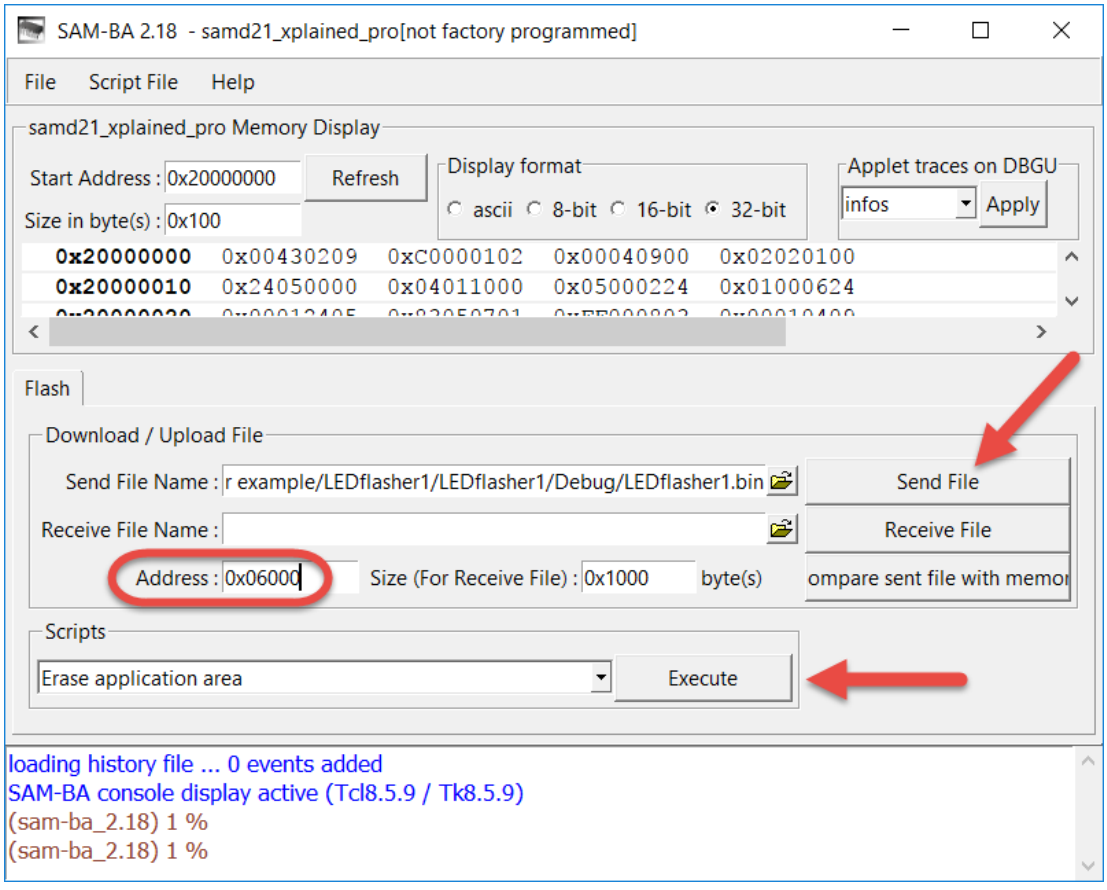
- Your LED flasher application will stop running (LED0 stops flashing) indicating the SAM-BA monitor is now running. Now that the SAM-BA monitor is running, it can connect to the SAM-BA Host utility.

4.3: Upload your second project to the SAM D21

- Start the SAM-BA Host utility on the PC
- Select the '\USBserial\COMx' connection and choose the samd21_xplained_pro board
- Click the Connect button



- As you did in the previous lab, click the Execute button in the Scripts field to erase the flash application area
- Change the start address from 0x0000 to 0x6000
- Browse to the folder containing your second application and select the binary file
- Click the 'Send File' button to upload the application



4.4: Reset the board to verify you've successfully uploaded your project

- Push the RESET button (located beside the DEBUG USB connector) to exit the SAM-BA Monitor mode.
- Verify LED0 blinks as expected (either once or five times per second).