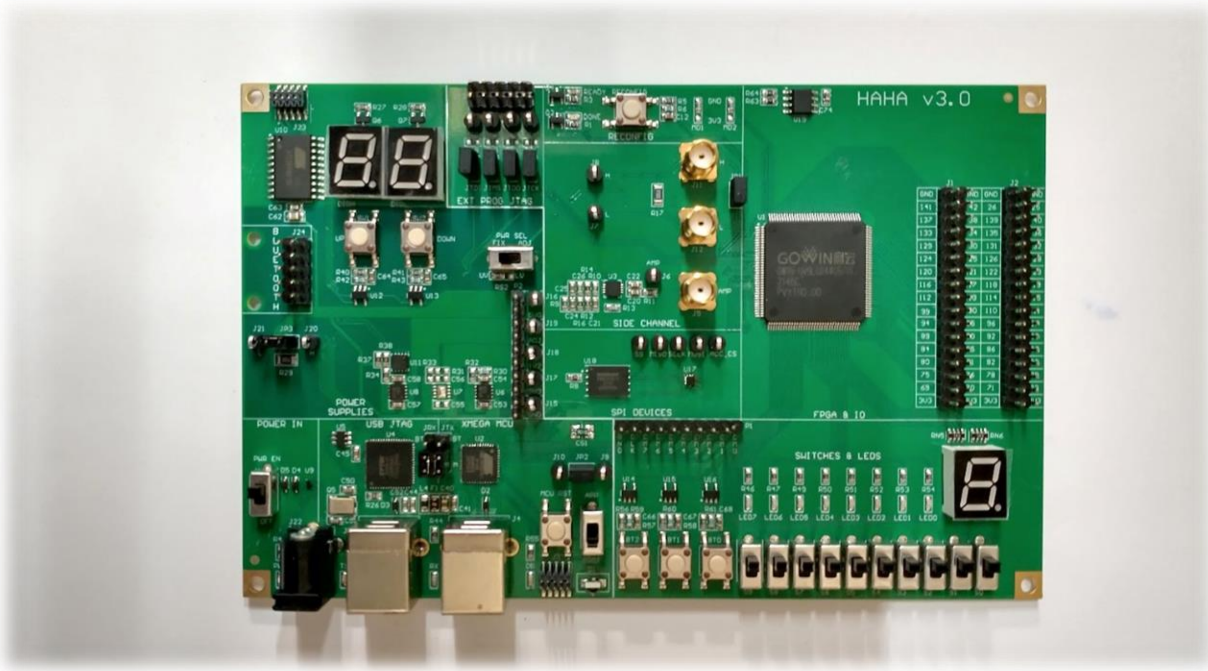
HAHA 3 Hardware Overview

Table of Contents


HAHA v3.0
RECONFIG
EXT PROG JTAG
PWR SEL
SIDE CHANNEL
SPI DEVICES
POWER SUPPLIES
POWER IN
USB JTAG
XMEGA MCU
SWITCHES & LEDS
FPGA & IO
GOWIN

Overview

The HAHA 3 is a hardware security platform featuring a Gowin GW1N-9C FPGA and a ATxmega16A4U microcontroller (MCU). The board is capable of several attacks to the FPGA.

Required Software and Drivers

The HAHA 3 requires the use of the following software and drivers.

- Microchip Studio 7 https://www.microchip.com/en-us/tools-resources/develop/microchip-studio
- Gowin EDA Standard Edition
    a. Apply for a free license at https://www.gowinsemi.com/en/support/license/ then download Gowin EDA.
- FTDI Drivers
    a. https://ftdichip.com/drivers/
- FLIP 3.4.7(optional)
    a. https://www.microchip.com/en-us/development-tool/flip select the version with the java runtime already included.

Board Layout

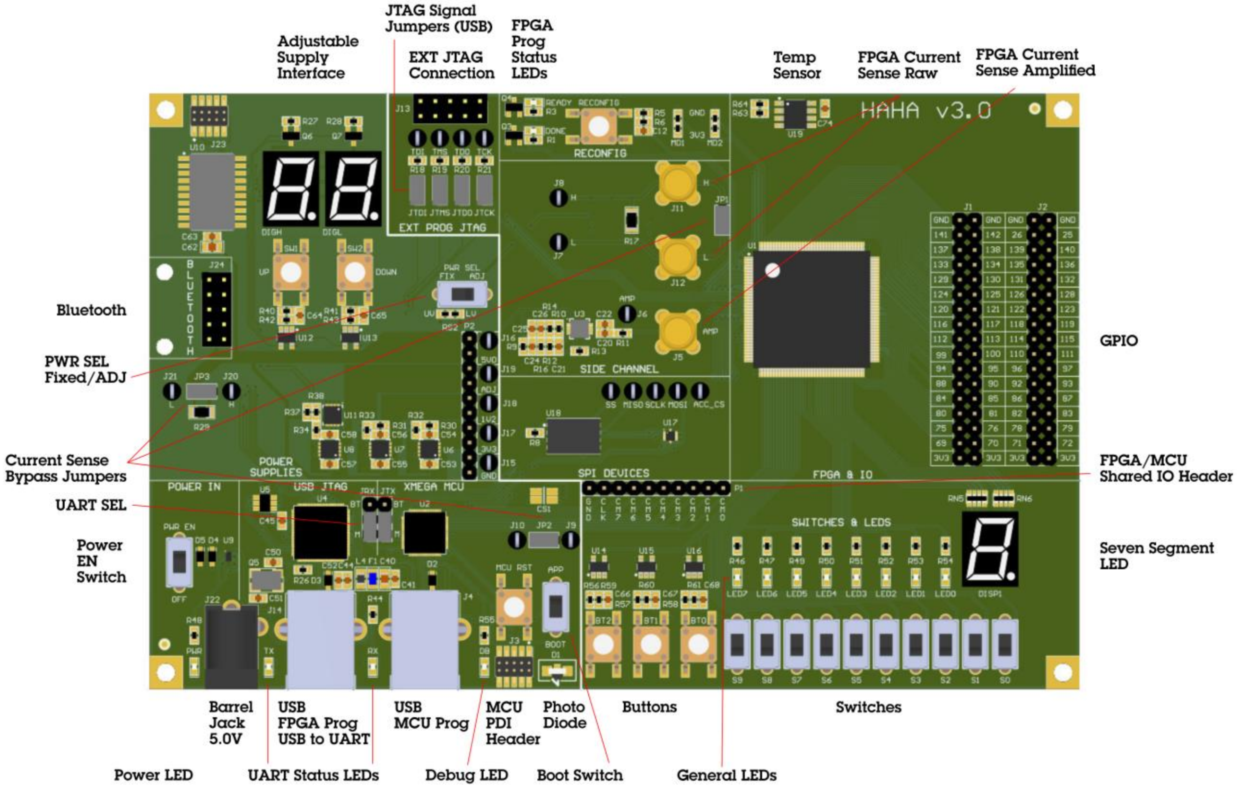


Figure 1: HAHA 3

Features

The following is an overview of features of the HAHA 3 for connection details reference the Connections Appendix. For locating interfaces or connectors refer to Figure 1.

Power Management

The HAHA 3 can be powered by USB (either port) or through the barrel power connection (2x5.5mm center positive 5V). Due to the current requirements of some applications USB power alone may be limiting.

Never power the HAHA 3 with a supply rated for more than 5V.

Power switch and Source Switching

The HAHA 3 uses a FPF1320UCX (U9) to switch between either USB or barrel power. Barrel power takes precedence over USB due to higher current capability. To turn on the HAHA 3 use the PWR EN switch at the bottom left of the board.

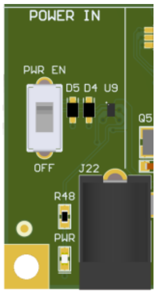


Figure 2: Power EN Switch

Power supplies

The HAHA 3 is designed to support both 1.2V (LV) and 3.3V (UV) core voltage GW1N-9 FPGAs there are 3 voltage regulators 3.3V, 1.2V, and ADJ. For boards featuring UV chips the 1.2V rail is unpopulated. The ADJ (adjustable) supply is used for attacking the FPGA. The adjustable supply is preconfigured for 3.3V (UV)or 1.2V (LV) max voltage.

Adjustable Power supply Interface

The adjustable power supply uses a up/down button interface to change the voltage of the ADJ supply. The voltage is displayed on two seven segment displays at two decimal places of precision. This is achieved by scrolling. Example for 1.25V: 1.2 will be displayed followed by 25.

SW1 is the Up button, and SW2 is the Down button. As visual feedback, the decimal points of the above seven segment display will indicate when the buttons are pressed, and the scroll effect of the display will be reset upon any interaction with Up or Down. The ADJ supply stores the current state in nonvolatile memory.

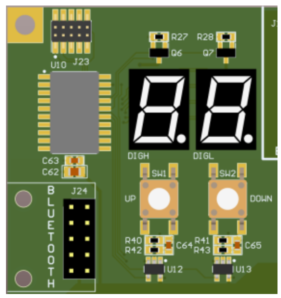


Figure 3: Adjustable Supply Interface

FPGA Core Voltage Select

The core voltage of the FPGA can be connected to the adjustable (ADJ) supply or the fixed (FIX) voltage supply using the PWR SEL switch located in the Power Supplies section of the HAHA 3 board.

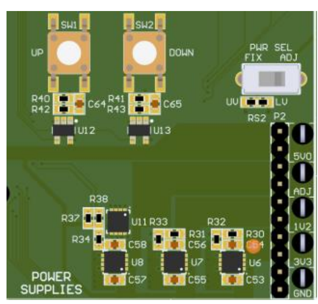


Figure 4:PWR SEL Switch

USB

There are two USB connectors to the HAHA 3. (J14) is a dual-purpose port for UART to USB communications and FPGA programming.

FPGA Programmer (Connector J14)

This connection can be used to program the GW1N-9C FPGA. Note the four jumpers in the EXT PROG JTAG section of the HAHA 3 board (Figure 5) must be present for programming. Otherwise, the USB programmer will not be connected to the FPGA's JTAG interface.

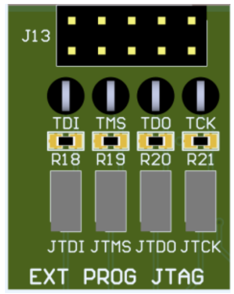


Figure 5: JTAG Signal Jumpers

JTAG and USB to UART connections can be used simultaneously.

USB to UART Bridge (Connector J14)

The second function of the J14 USB connector is USB to UART communications. The UART connection can either be the MCU or a Bluetooth dongle (not included). The selection jumpers JRX and JTX select either MCU (M) or Bluetooth (BT) as the UART target. Consult MCU USB UART Considerations for this interface.

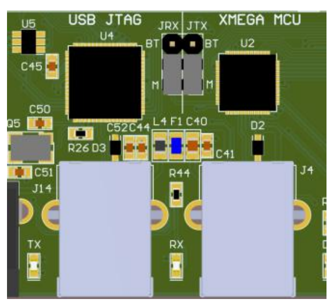


Figure 6: UART Connection Jumpers JRX and JTX

The MCU target's UART is USARTD0 pins PD3 (Xmega Tx), and PD2 (Xmega Rx)

MCU Programming Port (J4)

USB connector J4 connects directly to the MCU's (ATxmega16A4U) USB interface (pins PD7 and PD6). The MCU is programmed through the Flip program over the USB interface. Consult the Programming section for details.

Buttons

There are three push buttons connected to the FPGA. The button circuits contain a RC filter and an inverting Schmitt Trigger. When depressed their output is HIGH (3.3V).

Switches

There are 10 SPDT switches connected to the FPGA. The two throws are connected to 3.3V and GND.

General LEDs

The FPGA is connected to eight discrete LEDs These LEDs are HIGH true.

Seven Segment LEDs

The MCU has its own debug LED it is HIGH true.

Indicator LEDs

The HAHA 3 has several status LEDs. There are the UART status LEDs RX and TX that indicate when data is sent and received over the USB to UART bridge connected to J14.

There is a power LED PWR that indicates when the board is receiving power.

Seven segment LED displays DIGH and DIGL are the display outputs of the adjustable supply.

There are two FPGA programming status LEDs, DONE and READY, that display the value of the FPGA signals of the same name. When the FPGA is ready to be programmed READY will be lit. If the FPGA is programmed DONE will indicate this.

GPIO

There are 60 GPIO pins available through two banks J1, and J2. These connections also conveniently provide 3.3V and GND.

FPGA/MCU Shared IO

The FPGA and MCU share SPI connections for flash, a bus of 8 data pins and a clock pin.

Current Sensing

The HAHA 3 allows for current sensing of the board, the FPGA core voltage supply, and the MCU. Shunt resistors are placed in line with the circuits, and optionally jumpers can be used to bypass these resistors if current sensing is not desired.

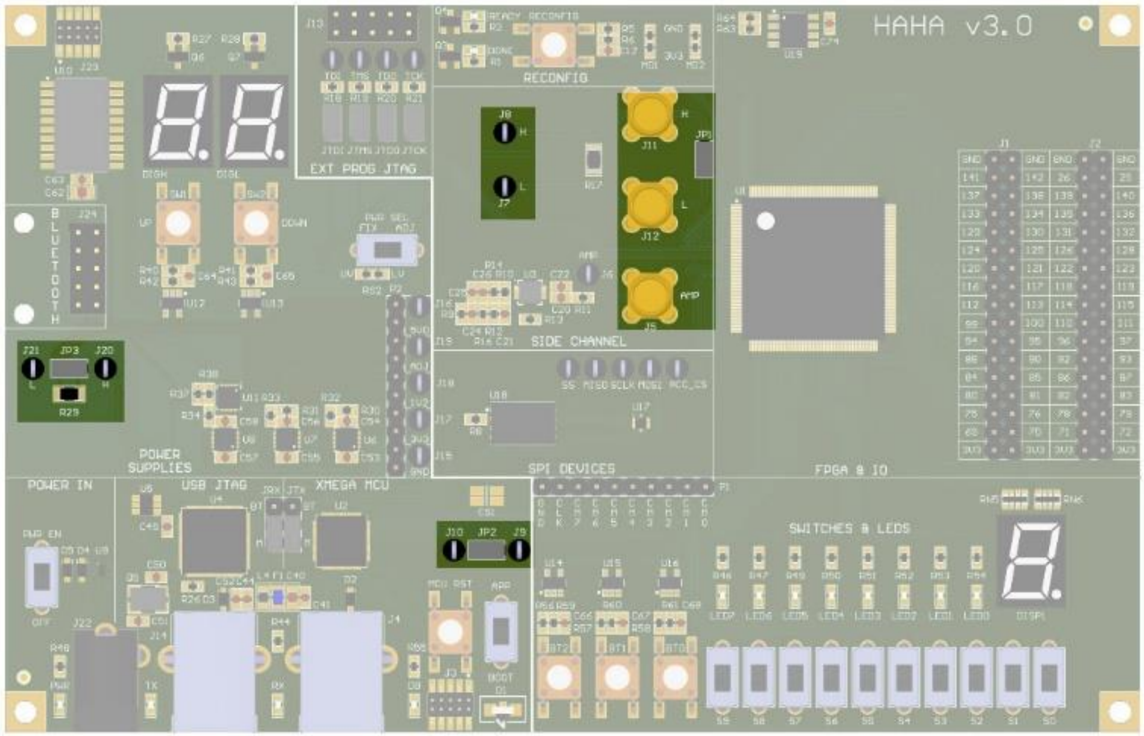


Figure 7: Current Sense Locations

Table 1: Current Sense Terminals

Current sense target	High Side Terminal(s)	Low Side Terminal(s)
Board	J20	J21
FPGA	J8, J11(SMA)	J7, J12(SMA)
FPGA Amplified	J6, J5(SMA)	N/A
MCU	J10	J9

Board Current Sensing

This current sense location connects to either the 5V rail coming off the two USB inputs or the external power jack if it is powered. The external power jack takes precedence if connected. The only device that receives power before the current sense terminals is the main board power LED. The sense resistance of the board current sense resistor is 100-milliohms.

FPGA Current Sensing and Side Channel Analysis

The FPGA VCC pins are monitored when performing side channel analysis. The terminals listed in the above table under "FPGA" are raw signal current sense terminals, where the terminals measure across an in line 100-milliohm resistor. The "FPGA Amplified" terminals are a single ended amplified version of the raw signal with some filtering designed to accentuate the spikes of side channel attacks for easier detection with an oscilloscope.

MCU Current Sensing

The MCU has a 10-milliohm resistor in line with its 3.3V power rail.

Photo Diode

The MCU has an IR photo diode connected to analog input pin (PB3).

Flash

Onboard SPI flash is connected between the FPGA and the MCU. The MISO pin has an inline 1K resistor for spying on communication to the device. For device connections reference SPI Device Connections in Appendix I.

Accelerometer

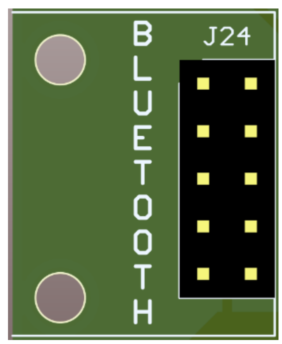
The MCU is connected to an SPI 3-axis accelerometer (MC3635). For device connections reference SPI Device Connections in Appendix I.

Bluetooth (Dongle Not Included)

The HAHA 3 has the option of future expansion to connect to a Bluetooth dongle to header J24. There is no radio attached to the board by default. The board connections are arranged in the following table as the pins are arranged in the accompanying snippet image.

Table 2: Dongle connections

PIN 1 +5V	PIN 2 3.3V
PIN 3 UART TX	PIN 4 DTR
PIN 5 UART RX	PIN 6 DSR
PIN 7 RTS	PIN 7 DCD
PIN 9 CTS	PIN 8 GND2

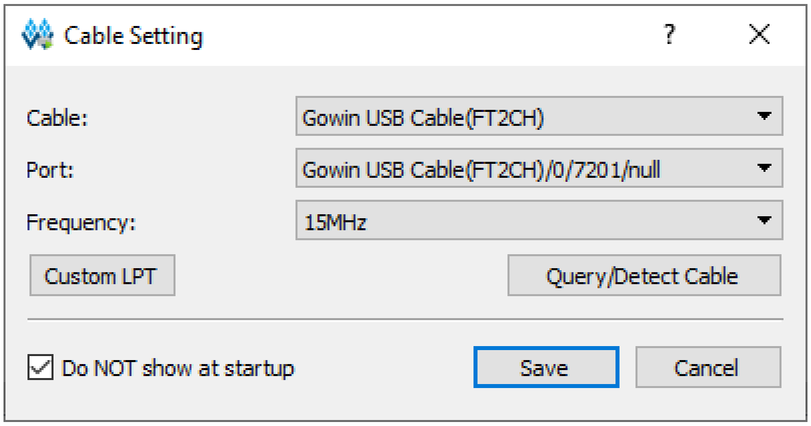


Programming

FPGA Programming

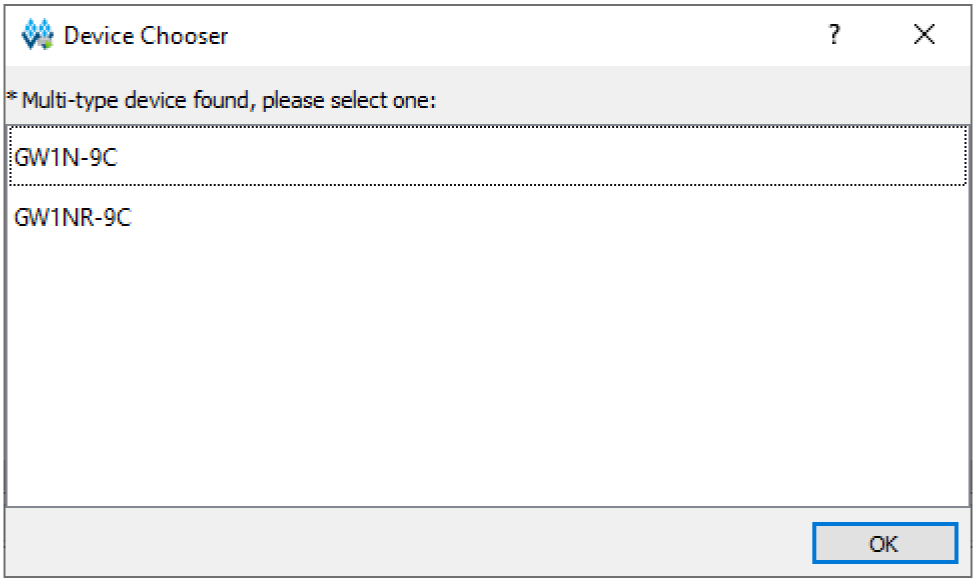
1. Connect USB (J14) of the HAHA 3 to the computer.
2. From within GOWIN FPGA Designer press to open the Programmer window.

   Select “USB Cable Setting” to configure the on-board programmer, and setup accordingly. The most important setting is the port. The USB device has two ports 0 and 1. Port 0 is used for JTAG programming. The other is reserved for UART communications.

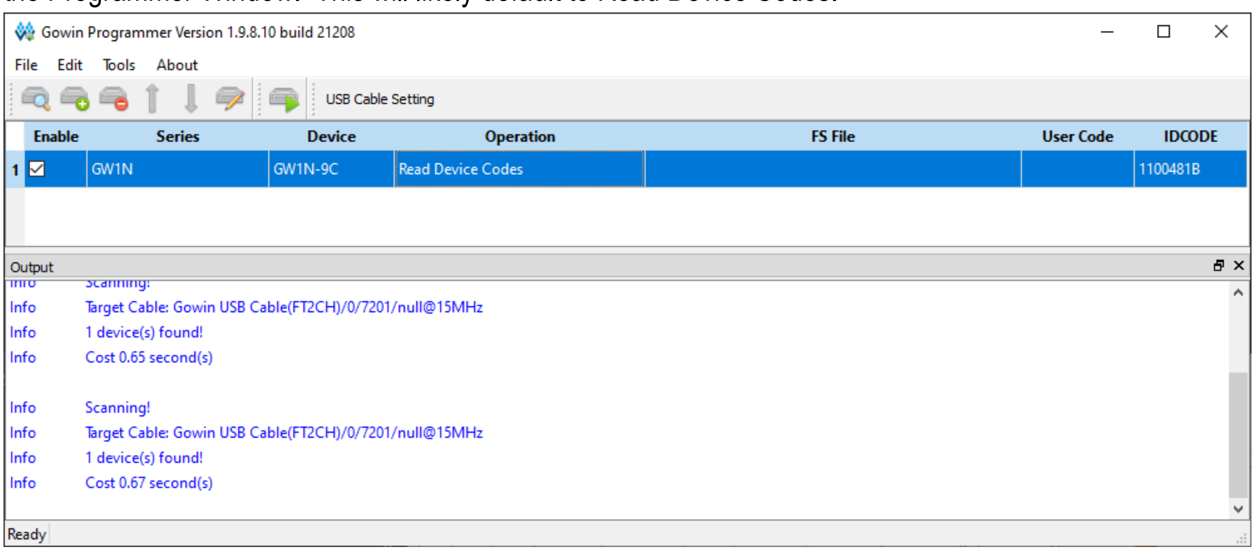
   Once configured save the Cable Setting.

3. To confirm the programmer is setup use select Scan Device . The Device Chooser window should appear.

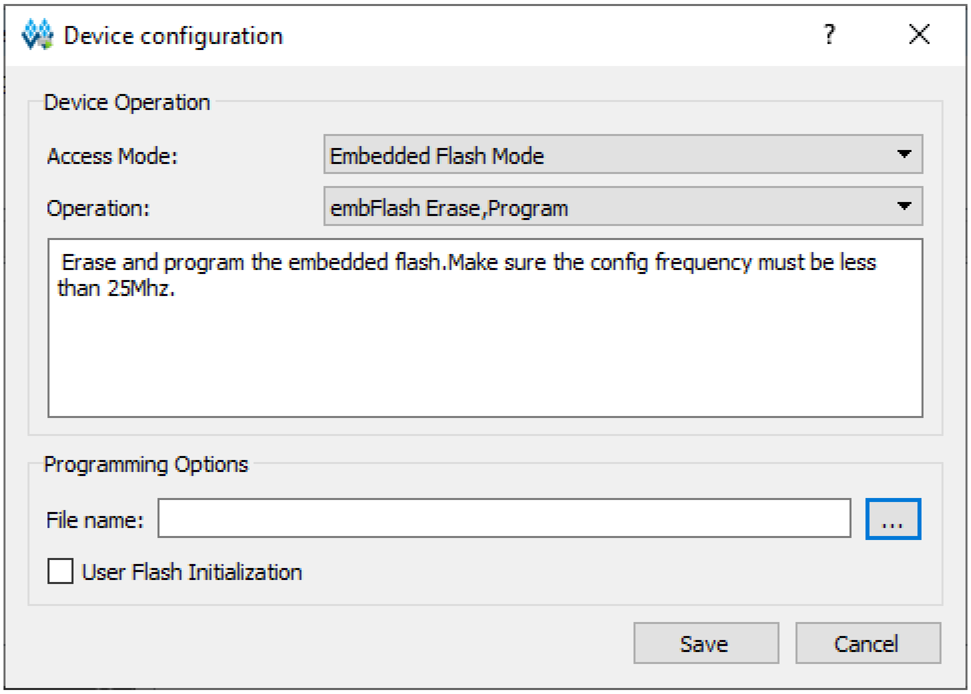


Select GW1N-9C.

4. The next step is to set the Operation of the programmer. Double click the Operation now shown in the Programmer Window. This will likely default to Read Device Codes.



The device configuration window should now appear. Select the access mode of either SRAM Mode (volatile) or Embedded Flash Mode. Then select the Operaton. If selecting either SRAM Program or embFlash Erase,Program and or verify a file name dialog will appear.



select a desired .fs file with the ... button and then press Save.

5. To program the device then press  in the Programmer window.

MCU Programming

Microchip Studio (recommended method)

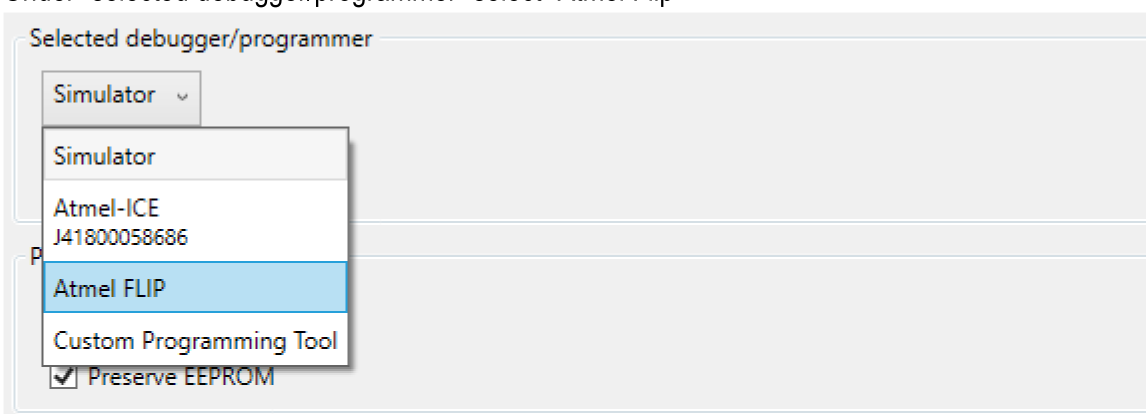
Microchip Studio supported direct programming using Atmel Flip within the Microchip Studio IDE. This is the most convenient method.

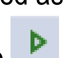
1. Connect to the USB connector J4.

2. When creating a project in Microchip Studio there is an option for selecting "Atmel Flip" as the programming tool. This will allow for programming of the Microcontroller (MCU) directly within the software development environment.

   a. To Select Atmel Flip as the tool either navigate to the menu
   Project => "Your Project Name" Properties

   b. Under "selected debugger/programmer" select "Atmel Flip"



3. Once Atmel Flip has been selected as the tool then programming can be performed directly from within Microchip Studio using the  button or by pressing (CTRL+ALT+F5).

1. When programming the MCU the Boot switch in the XMEGA MCU section of the HAHA 3 must be set to "BOOT", and the MCU RST button must be pressed to reset the MCU to accept programming. Refer to Figure 1 to locate the Boot switch and "MCU RST" button.

4. To run software written to the MCU switch the Boot switch to "App" and reset the MCU using the MCU RST button. To Program again go back to step 4.

Flip GUI

Microchip/Atmel has a separate Flip GUI for programming USB AVR devices. Below are instructions for using this software

2. Connect to the USB connector J4.

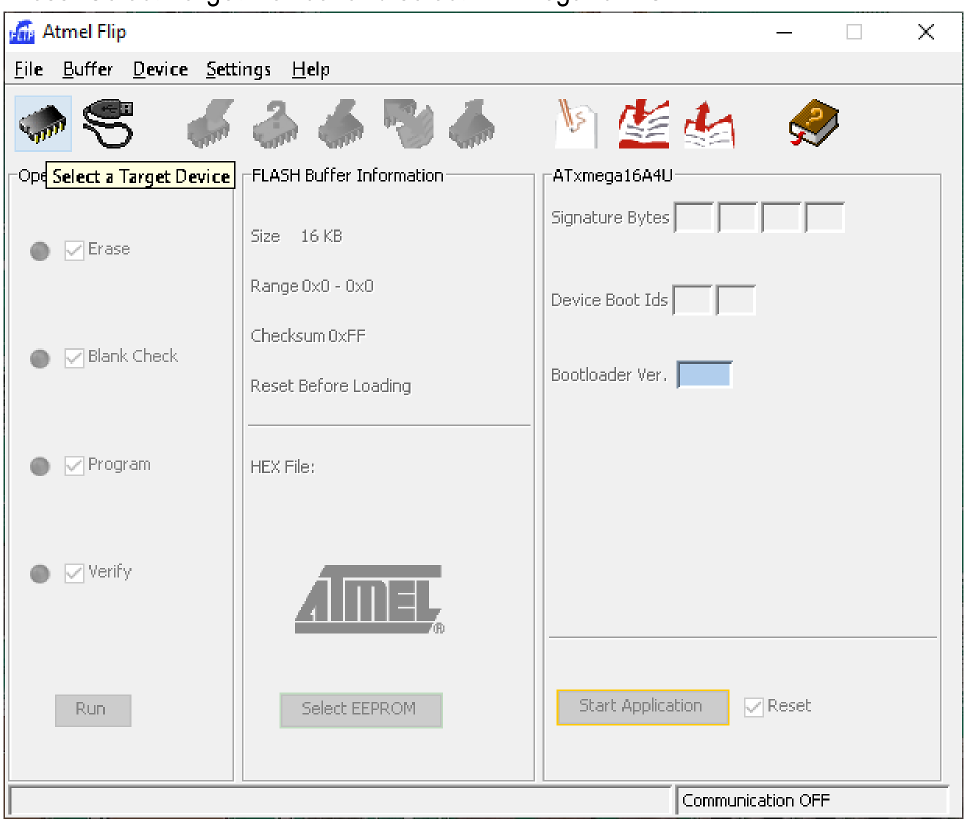
3. The Boot switch in the XMEGA MCU section of the HAHA 3 must be set to "BOOT". Refer to Figure 1 to locate the Boot switch.

4. While the board is powered on press MCU RST to reset the MCU into the bootloader program.

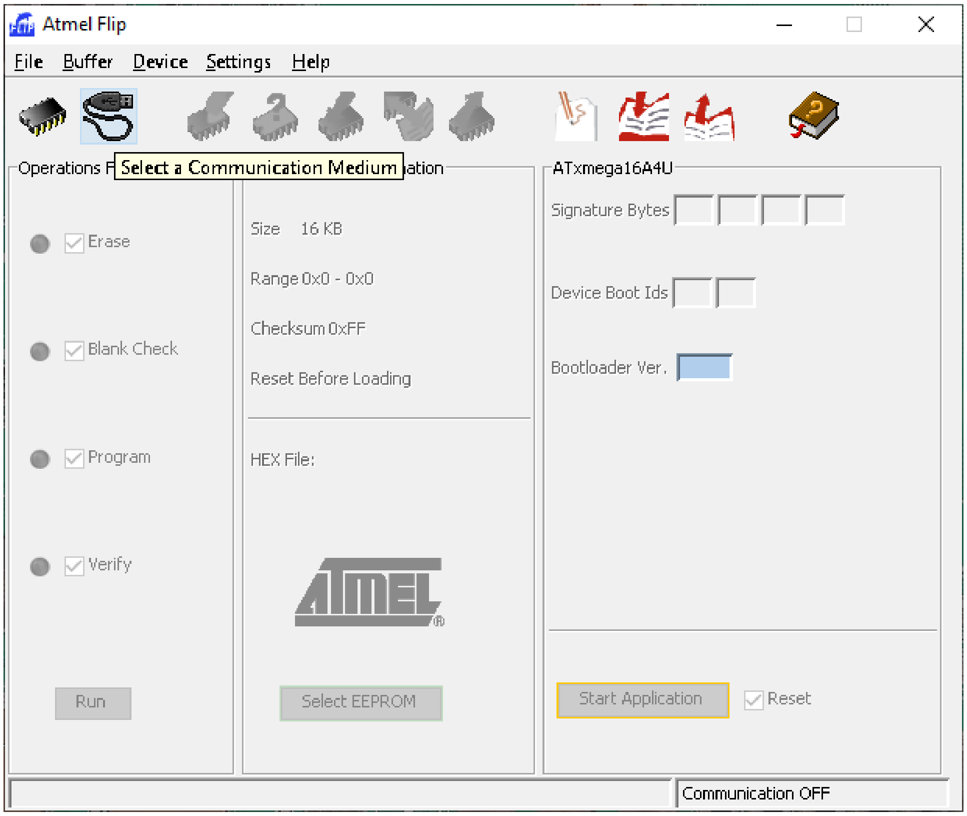
    a. A connection chime in Windows should be heard.

5. Open Atmel Flip

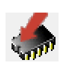
6. Press "Select Target Device" and select ATxmega16A4U



7. Press "Select a Communication Medium" and select USB then Open



8. To load a hex file press 

9. To program press 

10. To run the uploaded program move the Boot switch on the HAHA 3 to APP and pressing the MCU RST to boot into the uploaded application firmware.

Command Line Programming of MCU

For advanced users there is an option for programming using the Microchip Studio Command Prompt (included with the Microchip Studio installation).

The following example is the contents of a batch file contained in the same directory as a project hex file "HAHA_ledTest.hex". The example first performs a chip erase of the device and then the program is written to the MCU.

Atprogram calls the command line program (old naming scheme that still exists prior to microchip buying Atmel)

-t is the tool. in this case the Flip interface

-i is the interface USB

-d is the device atxmega16a4u

chiperase is the command for erasing the chip

program is a command for programming

> **–verify** is an option of program
>
> **-f** is the file. In this case it is HAHA_ledTest.hex. Since the file in this case is at the same directory level there is no need to specify a file name including a path.

Example:

atprogram -t FLIP -i usb -d atxmega16a4u chiperase

atprogram -t FLIP -i usb -d atxmega16a4u program --verify -f HAHA_ledTest.hex

Appendix I: Connections

FPGA Connections

Clocks

Table 3: FGPA Clock Sources

Signal	FPGA Pin
50MHz	98
(Optional) 2nd OSC	106

MCU/FPGA

Table 4: MCU/FPGA Shared Connections

Signal	**MCU Pin**	**FPGA Pin**
CM0	PA0	15
CM1	PA1	12
CM2	PA2	10
CM3	PA3	9
CM4	PA4	8
CM5	PA5	7
CM6	PA6	6
CM7	PA7	7
CLK	PC0	23
SS (Flash)	PC4	29
MISO	PC6	30
SCLK	PC7	32
MOSI	PC15	34
WP_n	PE2	27
HOLD_n	PE1	28

UI

Table 5: FPGA LED Connections

Signal	FPGA Pin
LED0	51
LED1	52
LED2	54
LED3	56
LED4	57
LED5	58
LED6	59
LED7	60
SEG A	68
SEG B	67
SEG C	66
SEG D	64
SEG E	63
SEG F	62
SEG G	61
SEG DP	65

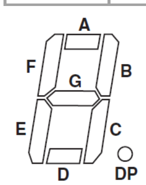


Figure 8: Seven Segment Labels

Table 6: FPGA Button and Switch Connections

Signal	FPGA Pin
BT1	38
BT2	39
BT3	40
S0	41
S1	42
S2	43
S3	44
S4	45
S5	46
S6	47
S7	48
S8	49
S9	50

Table 7: GPIO Connections

Signal(J1)	Signal(J1)		Signal(J2)	Signal(J2)
GND	GND		GND	GND
141	142		26	25
137	138		139	140
133	134		135	136
129	130		131	132
124	125		126	128
120	121		122	123
116	117		118	119
112	113		114	115
99	100		110	111
94	95		96	97
88	90		92	93
84	85		86	87
80	81		82	83
75	76		78	79
69	70		71	72
3.3V	3.3V		3.3V	3.3V

MCU Connections

For connections between the FPGA and the MCU refer to Table 3.

Table 8: MCU Specific Connections

Signal	MCU Pin
ACC CS	PE3
DB LED	PE0
USARTD0 TX (USB J14)	PD3
USARTD0 RX (USB J14)	PD2
BOOT n (BOOT switch)	PC3
Photodiode	PB3

SPI Device Connections

Though many of these signals are provided above here is a list of only the SPI connections

Table 9: SPI Device Connections

Signal	MCU Pin	FPGA Pin
ACC CS	PE3	N/A
SS (Flash)	PC4	29
MISO	PC6	30
SCLK	PC7	32
MOSI	PC15	34
WP n	PE2	27
HOLD n	PE1	28

Appendix II: Application Notes

MCU USB UART Considerations

The FT2232H IC used with J14 is a dual purpose USB device. The windows system will display two comports for this device. The lower number port is the FPGA programmer. The higher value port is the USB to UART bridge used for communicating with either the MCU or a Bluetooth dongle. Figure 11 shows the Device Manager window in Windows 10. Note that there are two sequential ports 10 and 11 of type "USB Serial Port". COM11 is the USB to UART port and COM10 is the JTAG programmer.

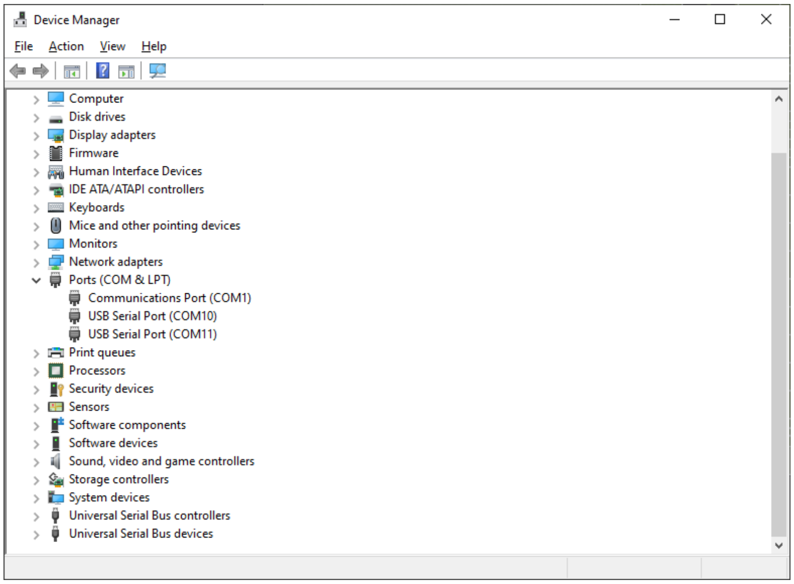


Figure 9: Device Manager Comport Display

FPGA Special Pin Configurations

When creating a new FPGA project there are dual purpose pins that should be converted to regular IO. To locate go to Project>Configuration and then select "Dual Purpose" under "Place and Route". Check both "USE SSPI as regular IO" and "USE MSPI as regular IO".

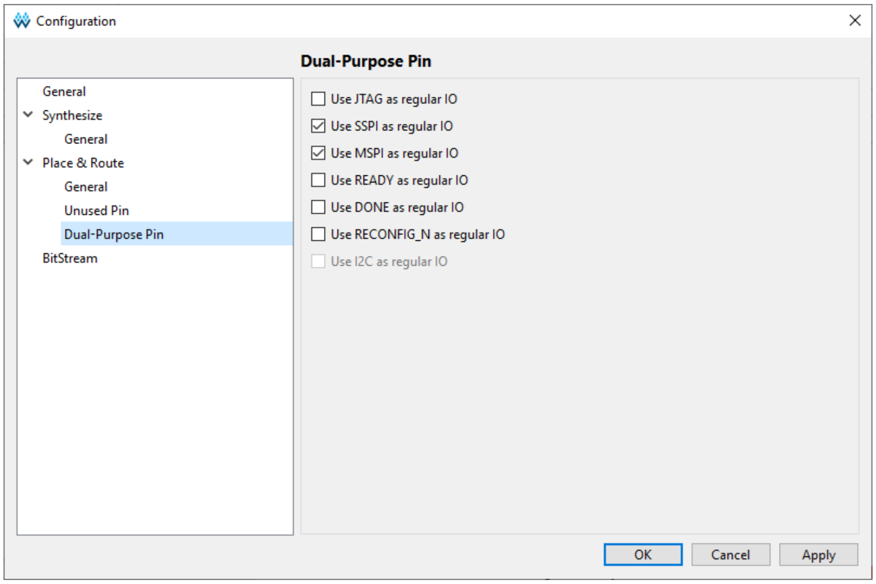


Figure 10: Dual Purpose Pin Configuration