

XMega Microcontroller Guide for HaHa v3.0 Board

Contents

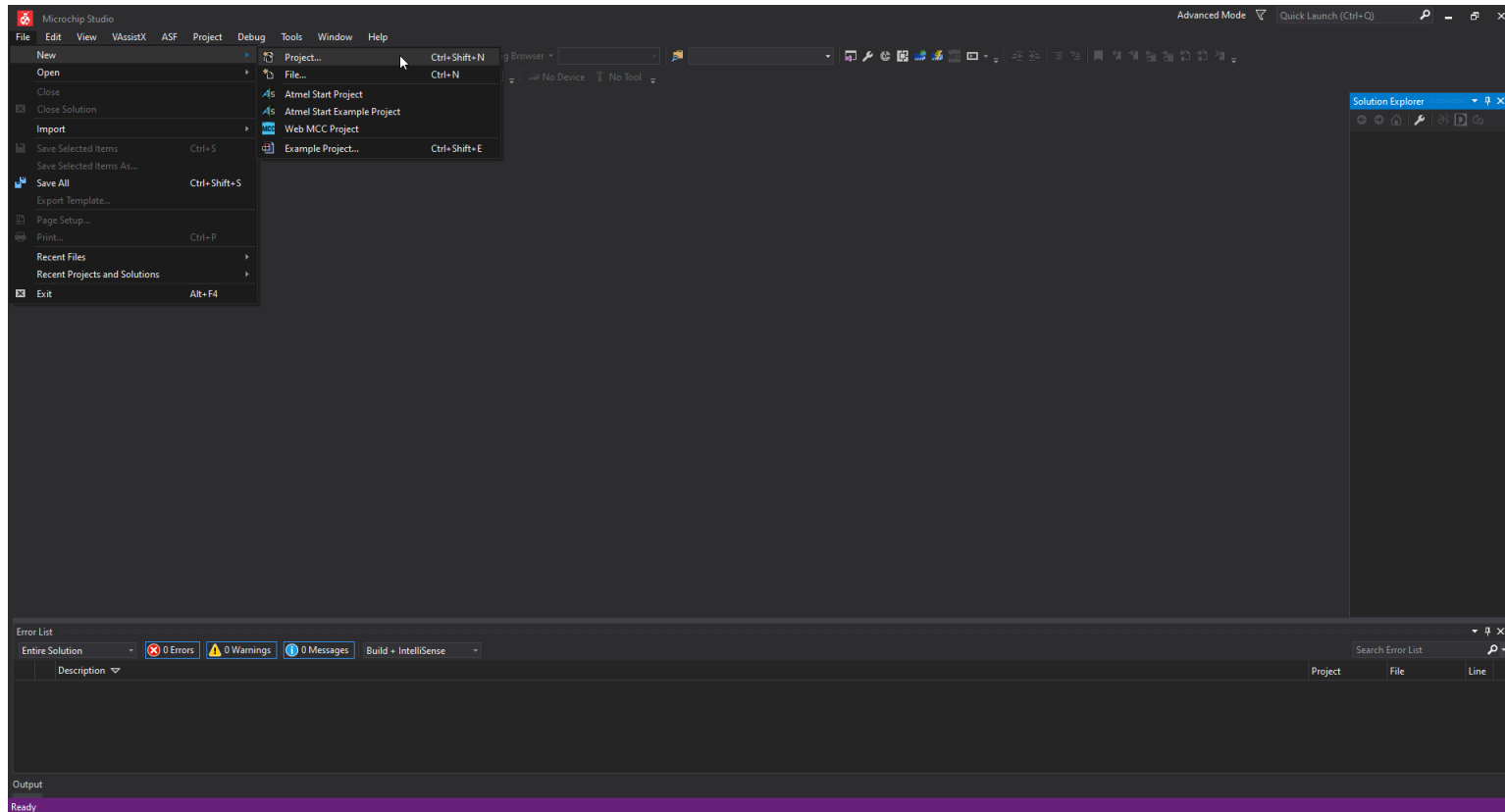
Starting a Microchip Studio Project	1
Programming the Xmega microcontroller	12
Using the SPI Libraries within HaHa v3 Header Files	15

Instructor: Dr. Swarup Bhunia

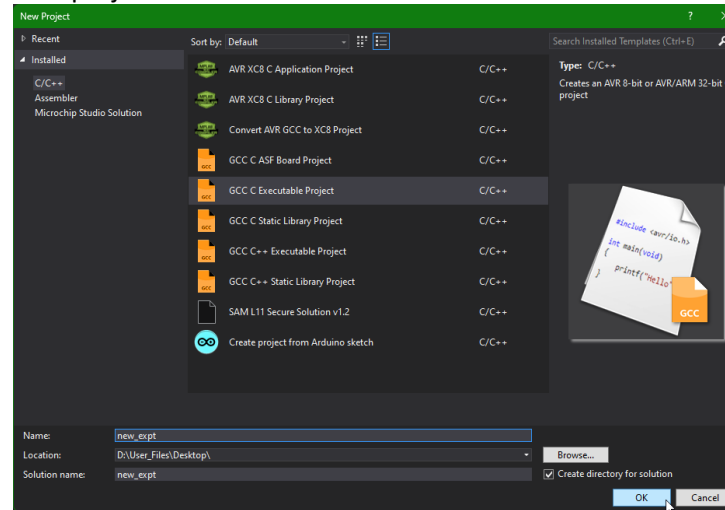
Co-Instructor: Reiner Dizon-Paradis

Starting a Microchip Studio Project

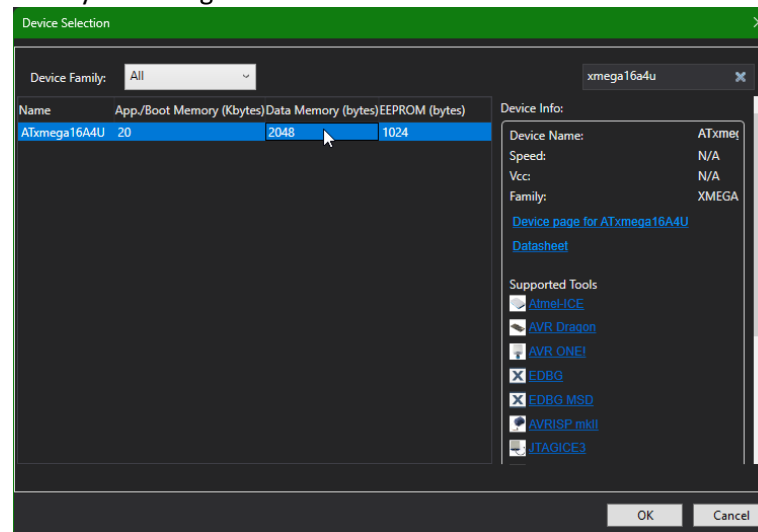
1. Open Microchip Studio. Go to **File > New > Project**.



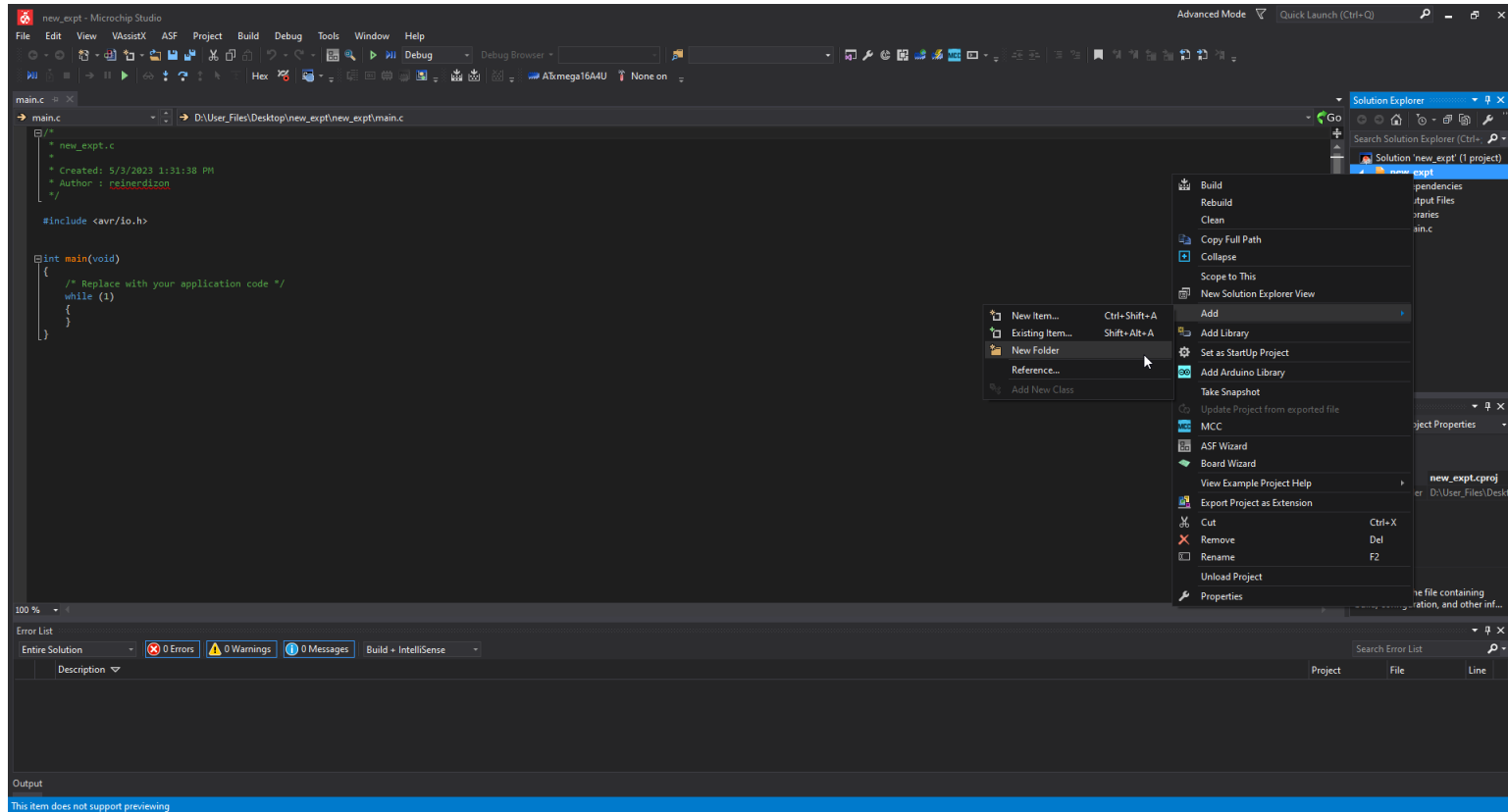
2. Select **GCC C Executable Project**. Type the project name and its location. Press **OK**.



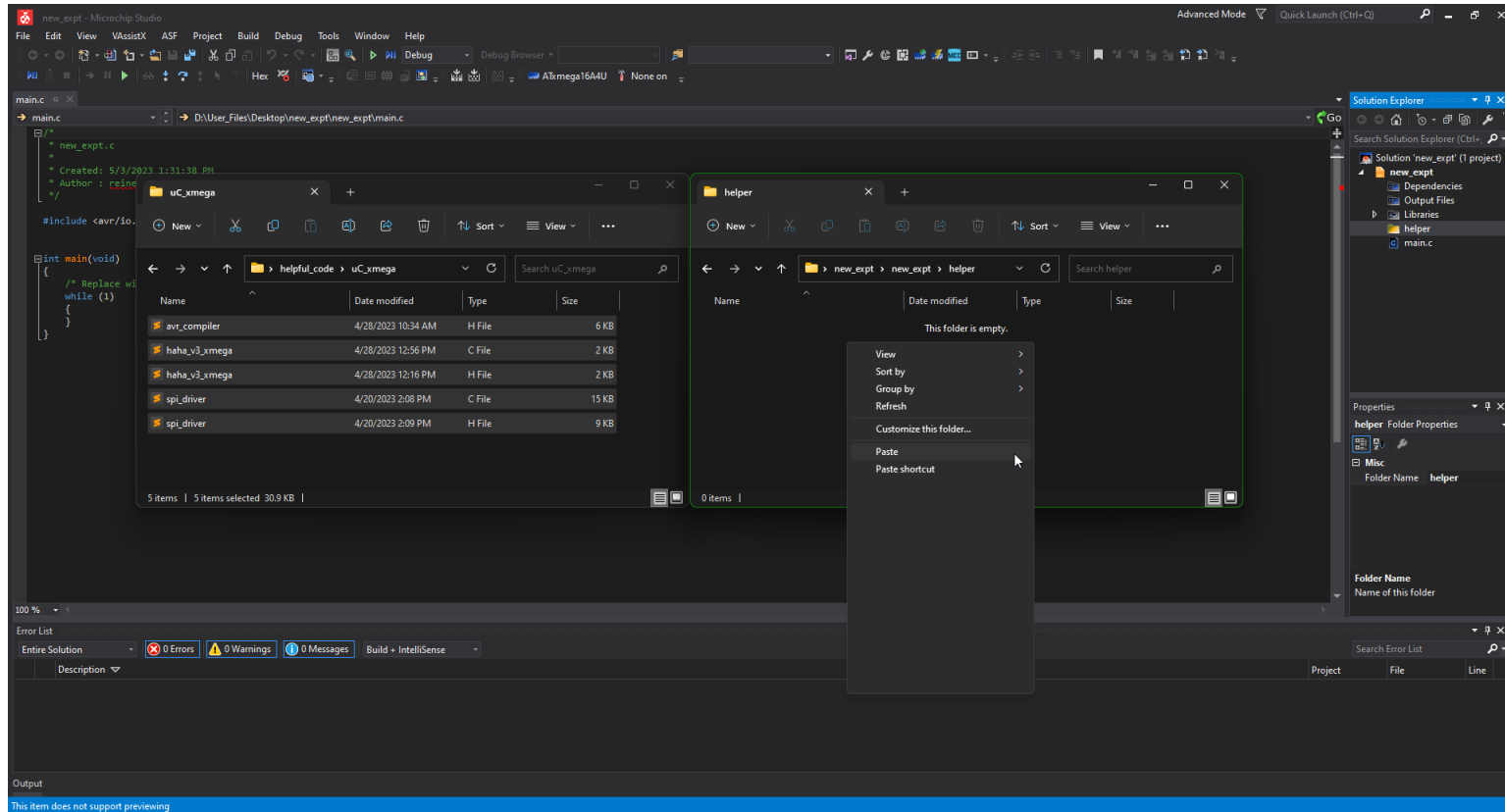
3. Select the device called **ATxmega16A4U** by searching “16a4u” in the search box. Press **OK**.



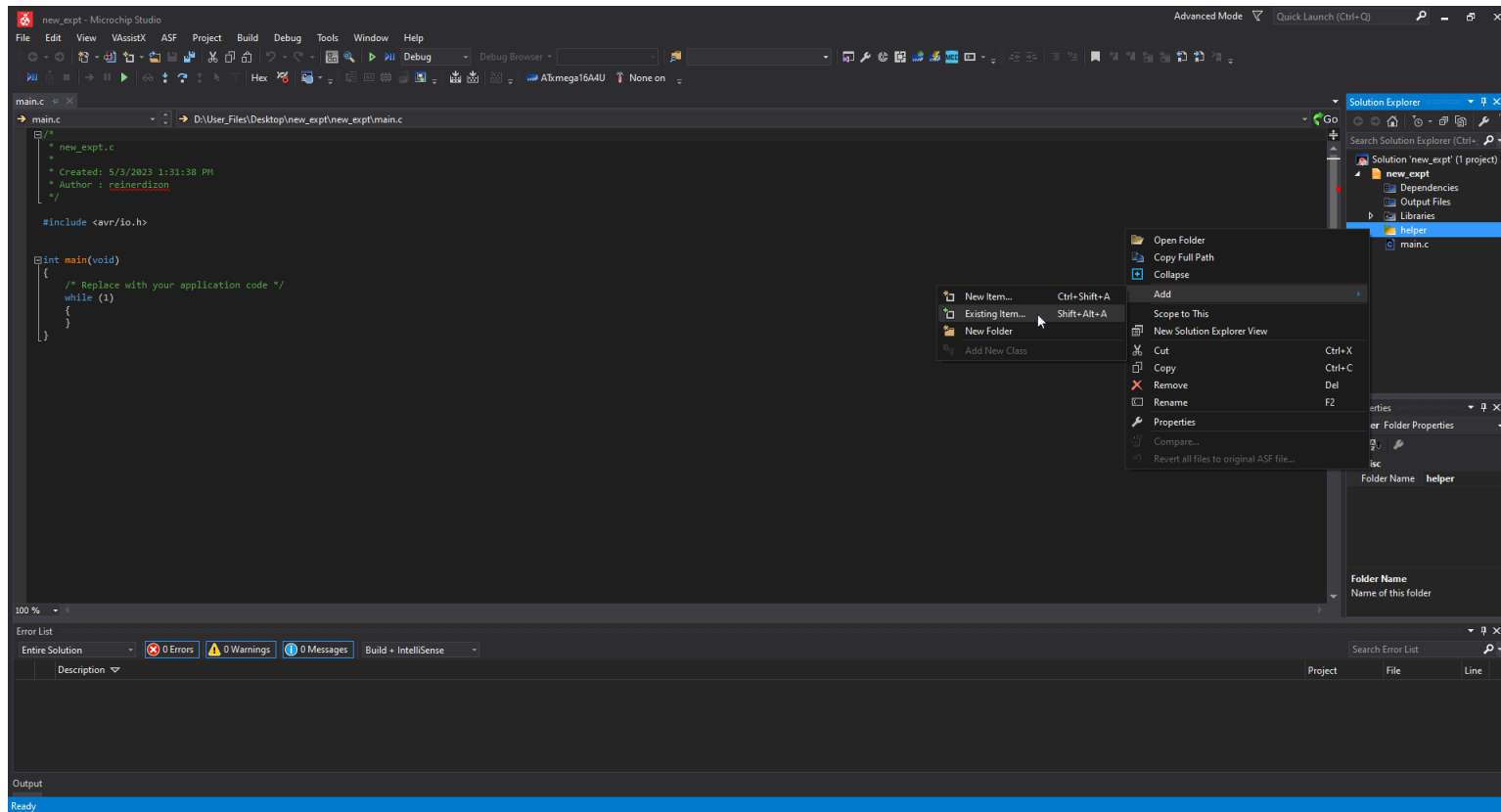
4. Right click on **<project_name>** folder on the right pane. Go to **Add > New Folder**. Call the folder *helper*.



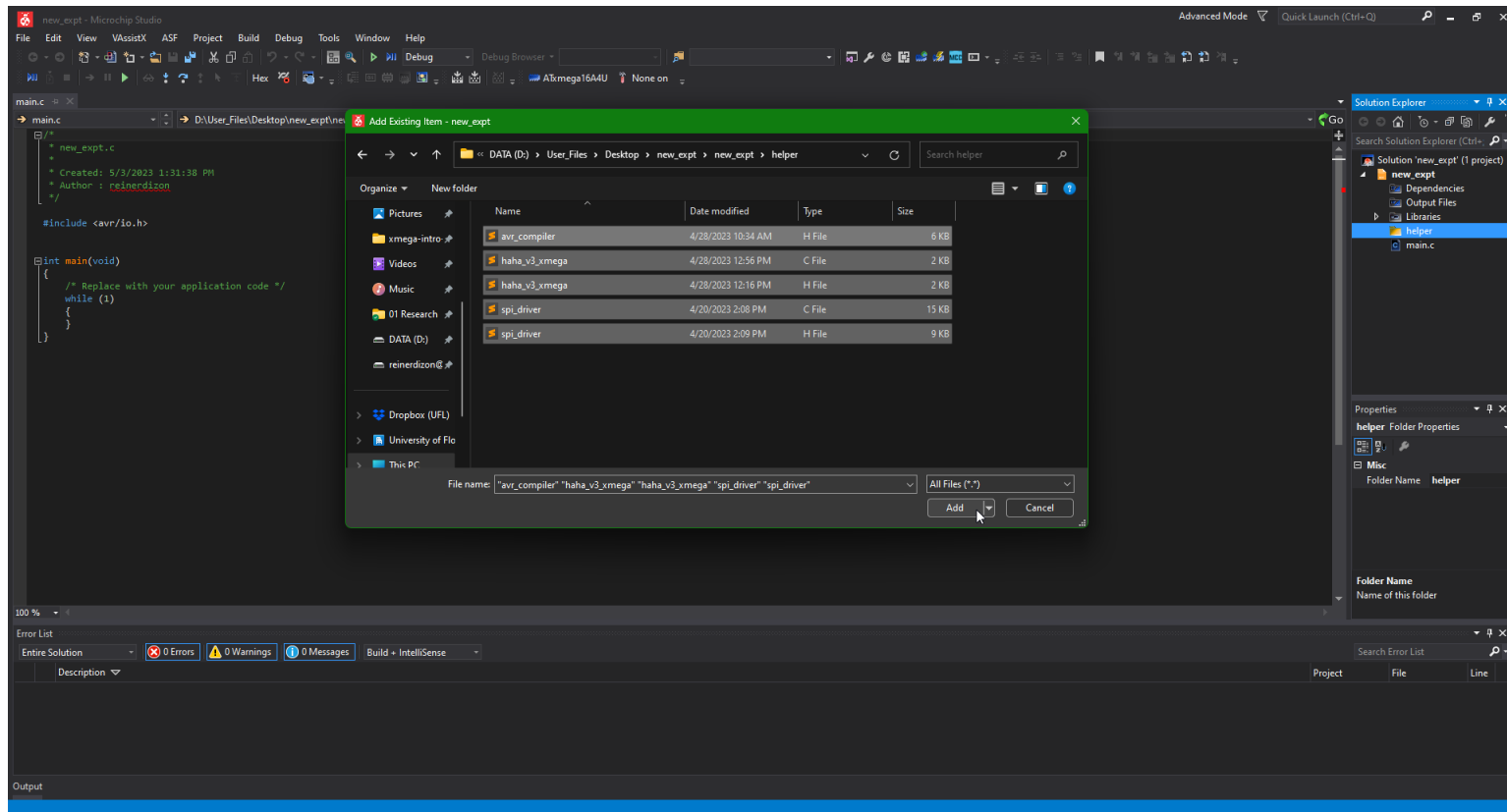
- Copy the helper codes (from *helpful_code/uC_xmega/*) to the *<project_name>/<project_name>/helper* folder.



6. Right click on *helper* folder on the right pane. Go to **Add > Existing Item**.

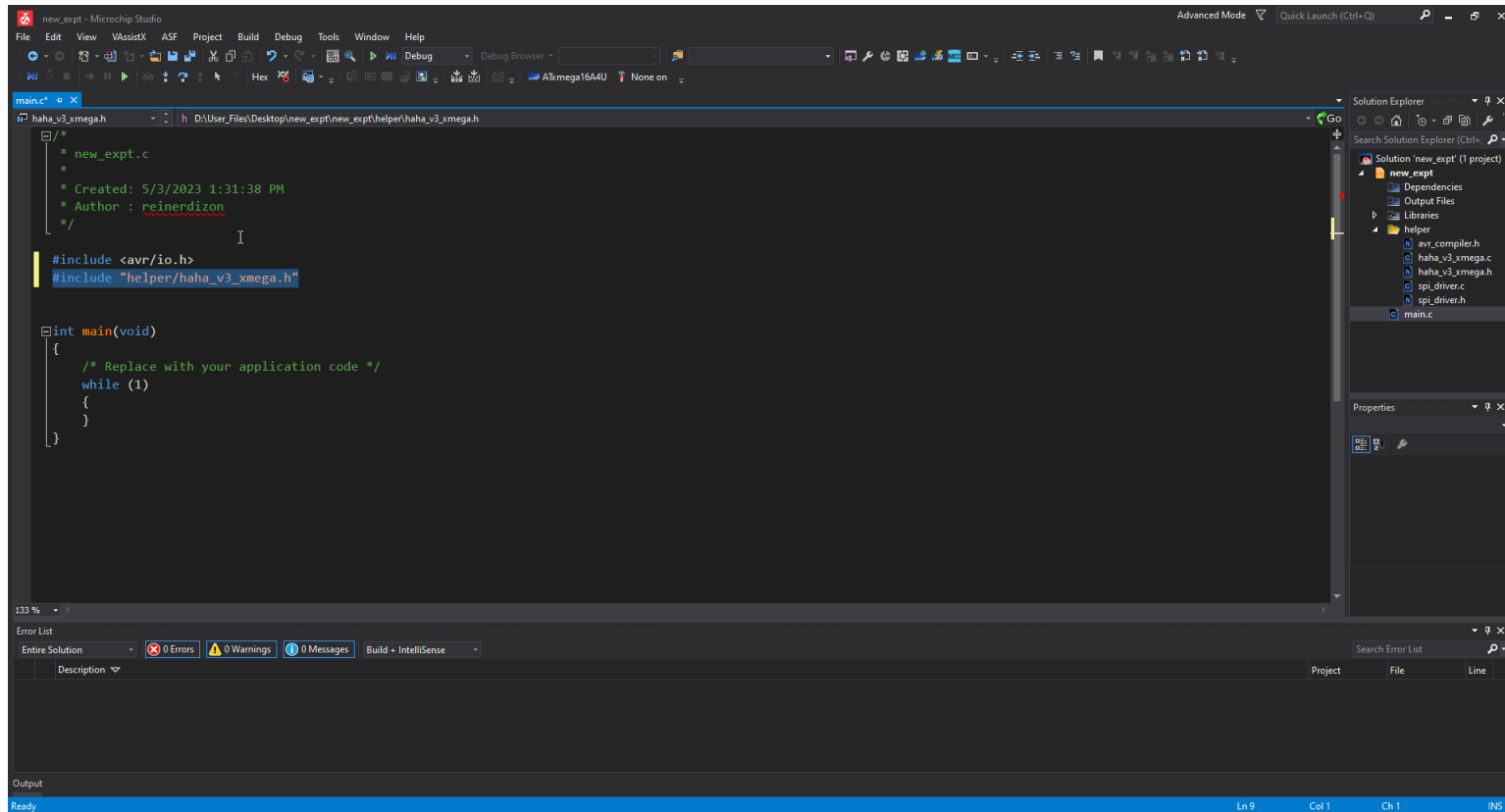


7. Add all the helper codes inside <project_name>/<project_name>/helper folder.



8. In your *main.c* file, add the following include statement:

```
#include "helper/haha_v3_xmega.h"
```



9. Examine **haha_v3_xmega.h** header file. This file includes helper constants, variables, and function prototypes that defines the chip interconnection ports and Serial Peripheral Interface (SPI) ports. In order to access SPI libraries, you need to uncomment the following line in the code below. Otherwise, let the file as it is to save space in the microcontroller.

```

/*
 * haha_v3_xmega.h
 *
 * HaHa v3.0 Board
 * Author: Reiner Dizon-Paradis
 */

#ifndef HAHA_V3_XMEGA_H
#define HAHA_V3_XMEGA_H

#include "avr_compiler.h"
// #include "spi_driver.h" // Uncomment if using SPI
#include <avr/io.h>

/* Pin Definitions for Chip Interconnection - Refer to the HaHa Manual */
#define HAHA_V3_CM_PORT PORTA
#define HAHA_V3_CLK_INTER_PORT PORTC
#define HAHA_V3_CLK_INTER_PIN PIN0_bm
#define HAHA_V3_CLK_INTER_DELAY 25

/* Function Prototypes */
void haha_v3_interBegin(void);
void haha_v3_clkInterPos(void);
void haha_v3_clkInterNeg(void);
void haha_v3_sendDataToFPGA(uint8_t data);

```

The screenshot shows the Microchip Studio IDE with the `haha_v3_xmega.h` file open. The code defines constants for pin configurations and provides function prototypes for interconnection and SPI operations. The line `// #include "spi_driver.h" // Uncomment if using SPI` is highlighted, indicating where a user might need to uncomment it to use SPI libraries.

10. Examine **haha_v3_xmega.c** implementation file. This file includes helper function definitions that set up the chip interconnection ports and interfaces with Serial Peripheral Interface (SPI).

```

new_exp1 - Microchip Studio
File Edit View VAssistX ASF Project Build Debug Tools Window Help
Debug Browser
haha_v3_xmega.c haha_v3_xmega.h main.c
D:\User_Files\Desktop\new_exp1\new_exp1\helper\haha_v3_xmega.c
Go
/*
 * haha_v3_xmega.c
 *
 * HaHa v3.0 Board
 * Author: Reiner Dizon-Paradis
 */

#include "haha_v3_xmega.h"

/* Set up chip interconnection as outputs */
void haha_v3_interBegin(void)
{
    HAHA_V3_CM_PORT.DIRSET = 0xFF; // set all PORTA as output
    HAHA_V3_CM_PORT.OUT = 0x00; // set PORTA to all 0's
    HAHA_V3_CLK_INTER_PORT.DIRSET = HAHA_V3_CLK_INTER_PIN; // set PC0 as output
    haha_v3_clkInterNeg(); // set to falling edge of clock
}

/* Send a rising edge for the Chip interconnection clock */
void haha_v3_clkInterPos(void)
{
    HAHA_V3_CLK_INTER_PORT.OUTSET = HAHA_V3_CLK_INTER_PIN;
    _delay_ms(HAHA_V3_CLK_INTER_DELAY);
}

/* Send a falling edge for the Chip interconnection clock */
void haha_v3_clkInterNeg(void)

```

133 %

Error List

Entire Solution 0 Errors 0 Warnings 0 Messages Build - IntelliSense

Description Project File Line

Output

Build succeeded Ln 1 Col 1 Ch 1 INS

11. Scroll down the **haha_v3_xmega.c** implementation file. The SPI functionalities do not include setting up the SS pin for device(s) used for a particular experiment. Refer to the experiment instructions and its references for more details.

```

new_exp1 - Microchip Studio
File Edit View VAssistX ASF Project Build Debug Tools Window Help
Debug Browser
haha_v3_xmega.c haha_v3_xmega.h main.c
D:\User\File\Desktop\new_exp1\new_exp1\helper\haha_v3_xmega.c

/* SPI Functions */
#ifdef SPI_DRIVER_H
void haha_v3_SPIBegin(void)
{
    /* Initialize SPI master on port C */
    SPI_MasterInit(&spiMasterC,
        &SPIC,
        &PORTC,
        false,
        SPI_MODE_0_gc,
        SPI_INTLVL_OFF_gc,
        false,
        SPI_PRESCALER_DIV4_gc);
}

/* Add your own function that initializes SS (HOLD_N, WP_N) ports as needed for selected SPI device. It should called before SPI operations - Example below: */

/* Init SS pin as output with wired AND and pull-up. */
// MC3635_SS_PORT.DIRSET = MC3635_SS_PIN;
// MC3635_SS_PORT.PINACTRL = PORT_OPC_WIREDANDPULL_gc;

/* Set SS output to high. (No slave addressed). */
// MC3635_SS_PORT.OUTSET = MC3635_SS_PIN;

#endif

```

133 %

Error List

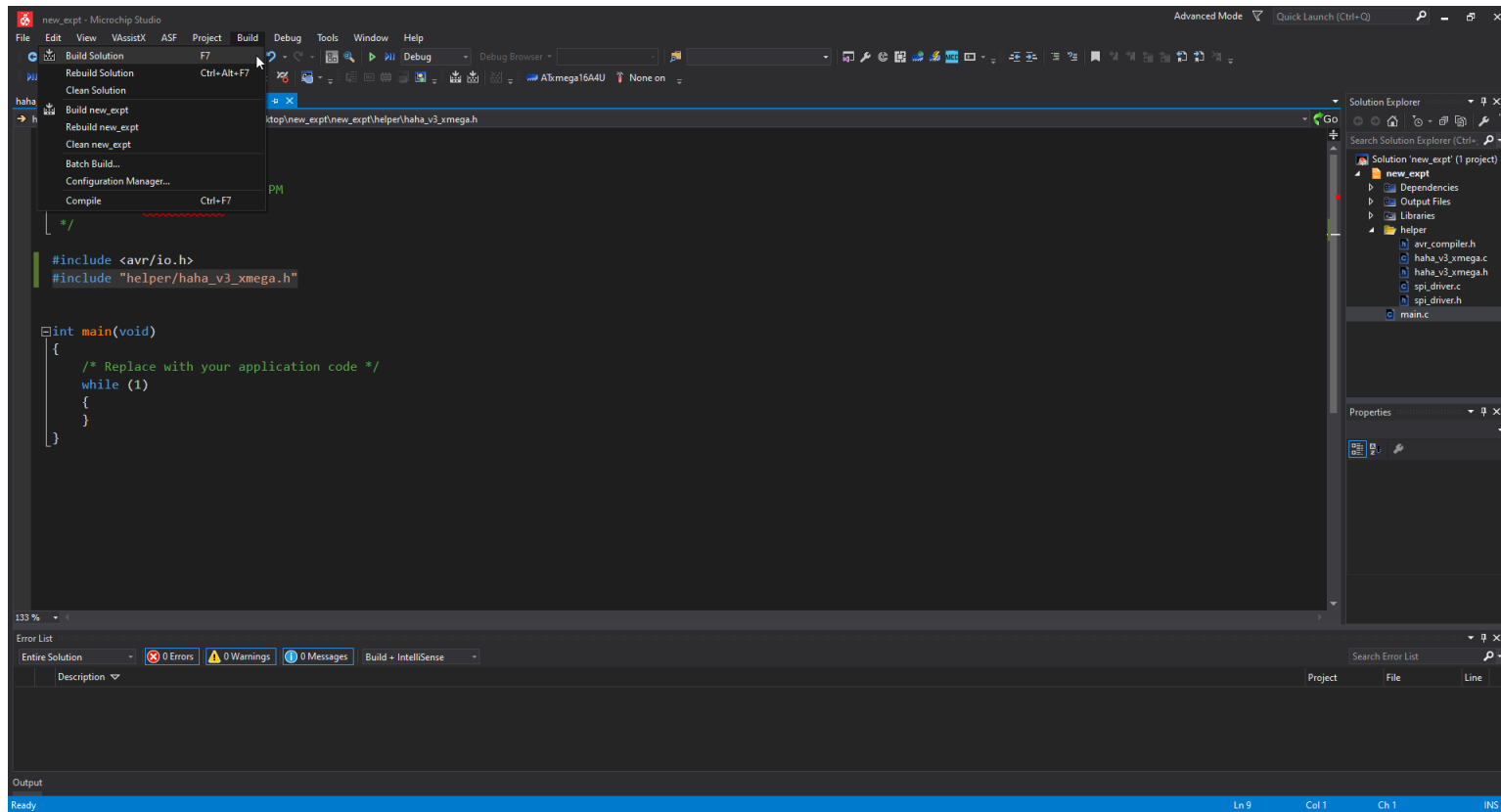
Entire Solution 0 Errors 0 Warnings 0 Messages Build + IntelliSense

Output

Build succeeded

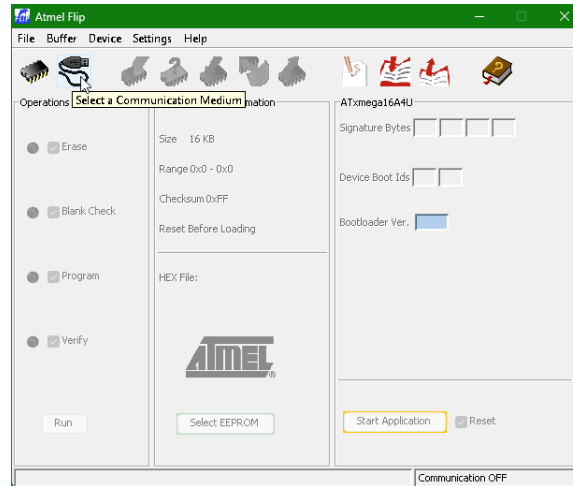
Ln 44 Col 20 Ch 20 INS

12. In theory, programming binaries can be built for this project, but nothing will happen on the HaHa board. Once you make the changes appropriate for the experiment instructions, you can build your project by going to **Build > Build Solution**. Fix any issues that arise. After that, you can proceed to programming the microcontroller.

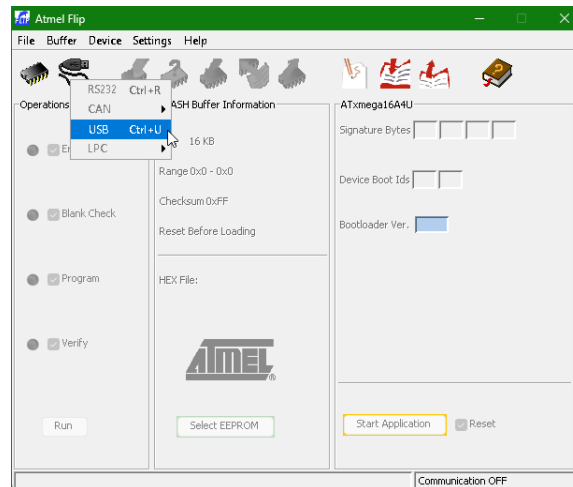


Programming the Xmega microcontroller

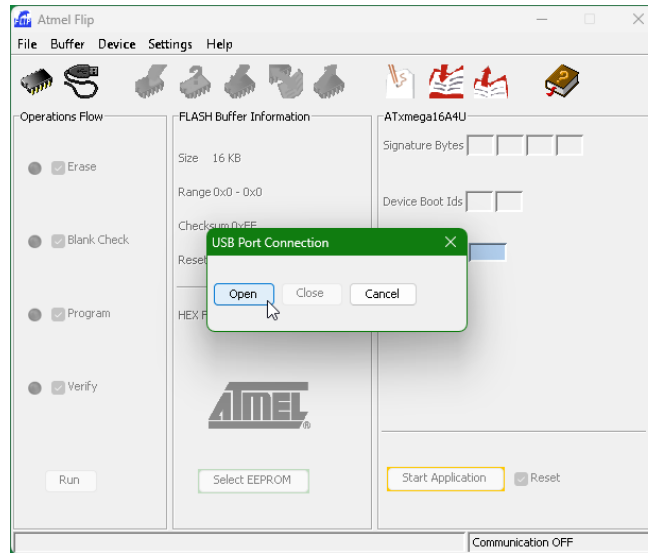
1. Make sure the switch next to the *MCU RST* button is switched down to *BOOT*. Double tap the *MCU RST* button.
2. Open Atmel FLIP. Click on the button (**Select a Communication Medium**) shown below.



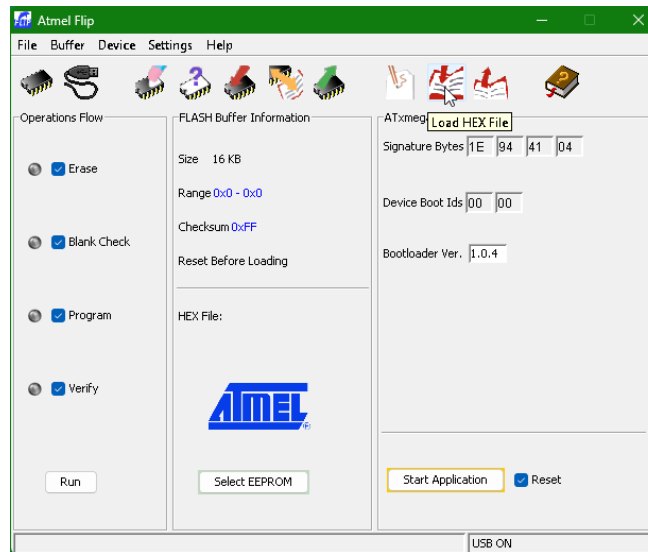
3. Select the **USB** option.



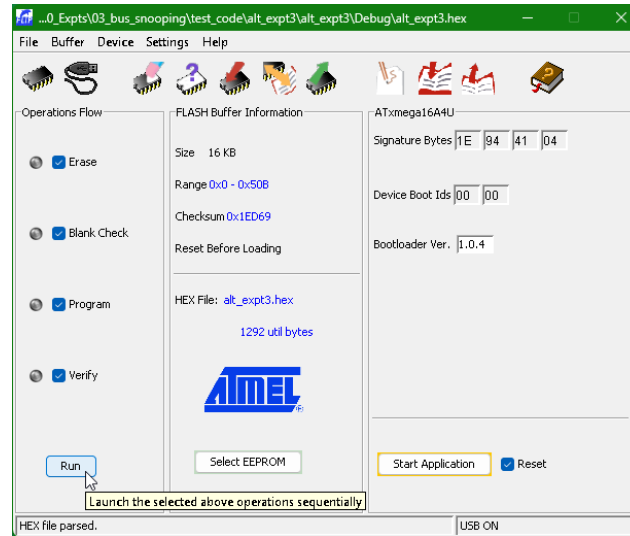
4. Click **Open**.



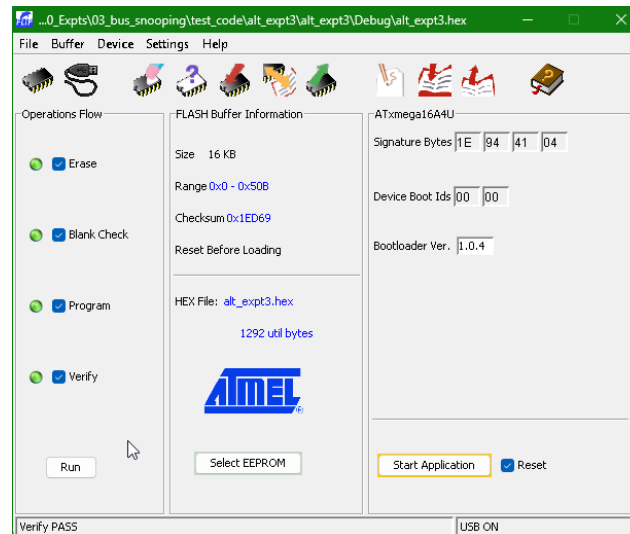
5. Click on the button (**Load HEX File**) below. Locate the HEX file and Click **OK**.



6. Click **Run**.



7. Your microcontroller program should be loaded. Then, the switch next to the *MCU RST* button should be switched up to *APP*. Double tap the *MCU RST* button.



Using the SPI Libraries within HaHa v3 Header Files

- For your reference, here are the pin names for FLASH pins:
 - SS (Slave Select): PORTC4
 - HOLD_N: PORTE1
 - WP_N: PORTE2
- For your reference, here are the pin names for Accelerometer pins:
 - SS (Slave Select): PORTE3
- Set those pins appropriately for your application.
- Make sure to add the following include statement:

```
#include "haha_v3_xmega.h"
```

- Make sure to uncomment the following line inside haha_v3_xmega.h file:

```
#include "spi_driver.h"
```

- Initialize your SPI master using the following:

```
haha_v3_SPIBegin();
```

- Make sure to set SS low before sending/reading data from SPI and then high after the intended operation.
- Send 1 byte of data to SPI slave using the following:

```
SPI_MasterTransceiveByte(&spiMasterC, <send_data>);
```

- Read 1 byte of data to SPI slave using the following:

```
read_data = SPI_MasterTransceiveByte(&spiMasterC, <send_data>);
```