
Software Manual

for eYFi-Mega Development Board

Produced by: e-Yantra Team

<https://e-yantra.org/productsdtl>

Version: 0.1.1

Dated: December 16, 2022

About this Document

This document is created for all developers wanting to work with the **eYFi-Mega** development board. It provides the detailed description and specification of the board.

Revision History

For revision history of this document, please refer to the Appendix [D](#).

Disclaimer and Copyright Notice

Information in this document, including URL references, is subject to change without notice. This document is provided as is with no warranties whatsoever, including any warranty or merchantability, non-infringement, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification or sample. All liability, including liability for infringement of any proprietary rights, relating to use of information in this document is disclaimed. No licenses express or implied to any intellectual property rights are granted herein. The e-Yantra Products logo is a registered trademark.

Copyright ©2022-23 e-Yantra, ERTS Lab, CSE Dept., IIT Bombay, India.

Table of Contents

1	Default State of eYFi-Mega	4
1.1	Default SSID and Password of eYFi-Mega Wireless Access Point . . .	4
2	eYFi-Mega ESP32 Partition Table	5
3	eYFi-Mega OTA Application	6
3.1	eYFi-Mega File Server	6
3.2	Set Wi-Fi SSID and Password	7
3.3	Wi-Fi Status LED Indication	7
4	Programming eYFi-Mega	9
4.1	Installing Arduino IDE packages for eYFi-Mega	9
4.2	Uploading program on ATmega2560 / ESP32 using Arduino IDE . . .	11
5	eYFi-Mega Wireless Serial Terminal	12
	Appendices	14
A	Know-hows for Linux systems	14
A.1	Kill Any Command	14
A.2	Linux and Blank Spaces	14
B	Flashing eYFi-Mega ESP32 files	15
B.1	Flashing eYFi-Mega ESP32 Bootloader	15
B.2	Flashing eYFi-Mega ESP32 Partition Table	15
B.3	Flashing eYFi-Mega ESP32 OTA Application	15
B.4	Flashing eYFi-Mega ESP32 User Application and Wireless Serial Terminal Application	16
C	Flashing Boot-loader on AVR	17
D	Revision History	20

List of Figures

1	eYFi-Mega File Server	6
2	Installing Arduino IDE 1.8.19	9
3	Adding eYFi-Mega package URLs	10
4	Installing board packages	10
5	Flashing started for ATmega2560	11
6	Flashing completed for ATmega2560	11
7	eYFi-Mega Wireless Serial Terminal Output	13
8	Device Programming dialog box in Atmel Studio 7	18
9	Fuse settings to be done	18
10	All OK message after programming the Fuses	19

List of Tables

1	eYFi-Mega ESP32 Partitions	5
2	WiFi-Status LED indications for different states	8
3	Revision History of the document	20

Default State of eYFi-Mega

The eYFi-Mega board comes out of the box with pre-flashed **bootloaders, partition table, OTA (Over-the-air) application** and LED blink **user application** on both AVR and ESP32 microcontrollers.

On powering up the board with the micro-USB connector, the AVR application should blink the RGB LED onboard and the ESP32 application should blink the Wi-Fi status LED.

1.1 Default SSID and Password of eYFi-Mega Wireless Access Point

Wi-Fi SSID: **eYFi-MEGA**

Wi-Fi Password: **eyantra123**



eYFi-Mega ESP32 Partition Table

We have partitioned the 4 MB Flash of onboard ESP32 to accommodate **1 MB OTA App**, **2 MB for User App** and **700 KB for File Storage**. This partition table can be modified by you as per your requirement.

For example, if you want to increase or decrease the size of the 700 KB File Storage you can update the partition table. If you decide to modify the partition table for your project, you might not be able to use the OTA App because our OTA App is compatible with the partition table described below in Table 1.

In case you accidentally erase the partition you can always download our Partition Table and our OTA App from our website.

Table 1: eYFi-Mega ESP32 Partitions

Partition Name	Start Address	Size
reserved area (nvs, otadata, phy_init)	-	300 KB
factory (ota-app)	0x10000	1 MB
app1 (user-app)	0x110000	2 MB
storage (spiffs)	0x310000	700 KB

eYFi-Mega OTA Application

The eYFi-Mega OTA Application which resides in the factory partition of ESP32, allows user to flash the firmware of ATmega2560 or ESP32 wirelessly and gives access to the file storage of eYFi-Mega.

3.1 eYFi-Mega File Server

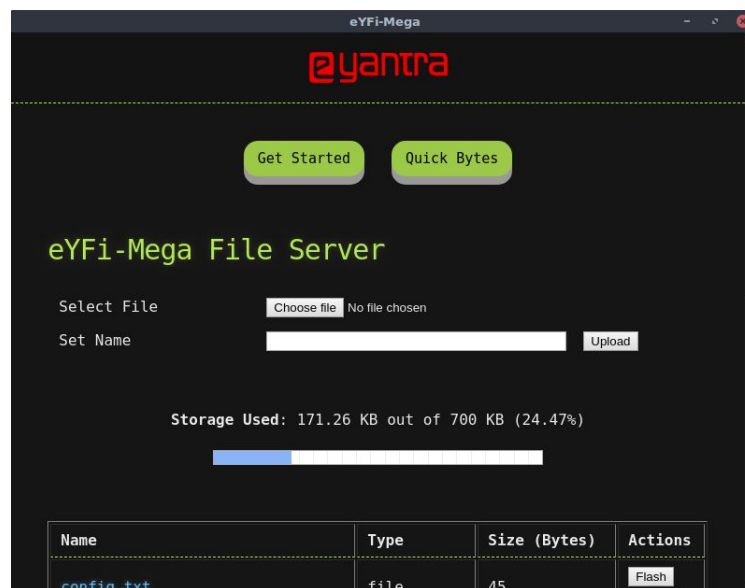


Figure 1: eYFi-Mega File Server

eYFi-Mega File Server is like a file explorer for the 700 KB onboard storage. To access this file server you need to first connect to eYFi-Mega's Wireless Access Point. After connecting to the Access Point you can use any web-browser to open the file server window at **192.168.4.1** IP address.

You can use this storage to store any files like. You can use this storage to configure your WiFi-SSID and password (explained in the next section), log data from sensors, you can even store **.bin** or **.hex** files of your firmware and flash it in any of the two controllers through this File Server.

NOTE: When you are setting the name of your file which is to be uploaded, keep the name of the file including the name of the extension less than 31 characters in length.

3.2 Set Wi-Fi SSID and Password

To set your own Wi-Fi SSID and Password of the board, upload a **config.txt** file to your eYFi-Mega File Server with your SSID and Password in the following format in the file. After the upload is done, reset the ESP32 by pressing the **ESP_RESET** button, and you are all set!

Filename: **config.txt**

WiFi-SSID [your-wifi-ssid]

WiFi-Pass [your-wifi-pass]

For example,

WiFi-SSID [eYFi-MEGA]

WiFi-Pass [eyantra123]

will set the SSID to "eYFi-MEGA" and Password to "eyantra123".

In case the **config.txt** file is missing from your file server or the format of the file is not proper then, the *default SSID and Password* will be set for your Wireless Access Point.

3.3 Wi-Fi Status LED Indication

When you are connected to your eYFi-Mega over a Wi-Fi link, it's important to know the different states in which your board is. These states are indicated by the Blue **Wi-Fi Status** LED, which is right next to Red **ESP_ON** LED. Table 2 is the description of different LED Patterns associated with different states of the board.

Table 2: WiFi-Status LED indications for different states

State	LED Pattern
Wi-Fi Client is connected to ESP32	ON
Wi-Fi Client is disconnected from ESP32	OFF
File Upload Start	OFF
File Upload End	Blink Fast for 100 ms
Firmware Flash Start	Blink Fast
Firmware Flash End for AVR	Blink Slow for 5s
Firmware Flash End for ESP32	Blink Slow till application Switch #1(S1) is toggled

Programming eYFi-Mega

The eYFi-Mega supports **Arduino IDE** as a programming environment. After installing packages as instructed in upcoming sections, both ATmega2560 and ESP32 of eYFi-Mega can be programmed and flashed directly using Arduino IDE.

4.1 Installing Arduino IDE packages for eYFi-Mega

- Install Arduino IDE version 1.8.19.

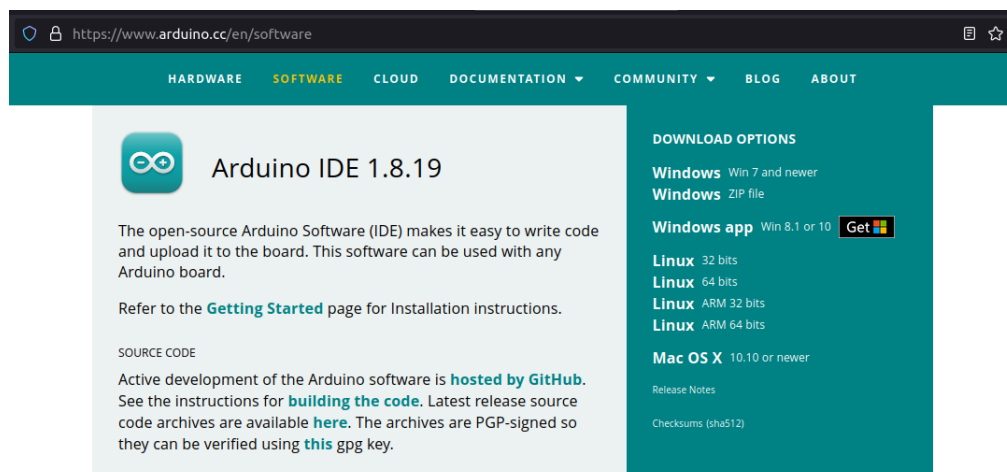


Figure 2: Installing Arduino IDE 1.8.19

- Add package URLs in Arduino IDE, go to **Files** → **Preferences**. Add the below two URLs in the text box labeled, *Additional Board Manager URLs*, one per line

For ATmega2560 : `package-url`

For ESP32 : `package-url`

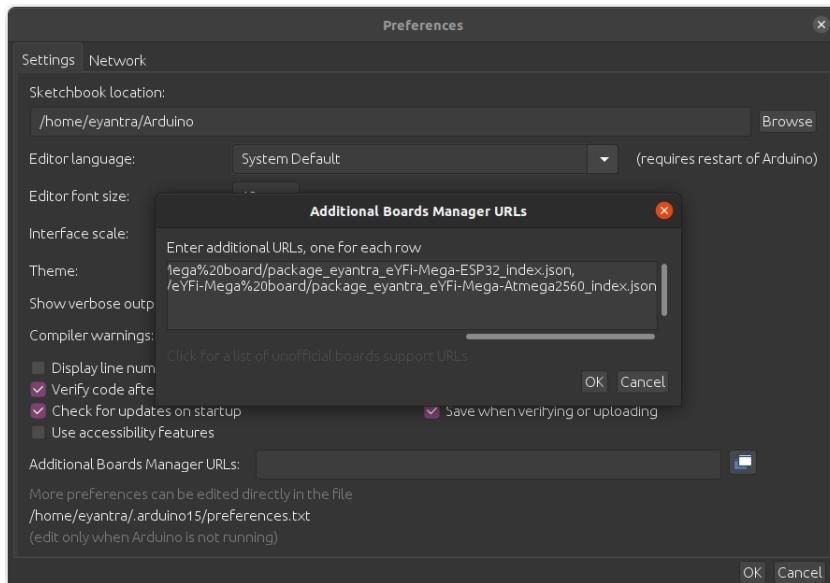


Figure 3: Adding eYFi-Mega package URLs

- Install board packages in **Tools** → **Boards Manager**. Search "eyfi" in the Boards Manager.

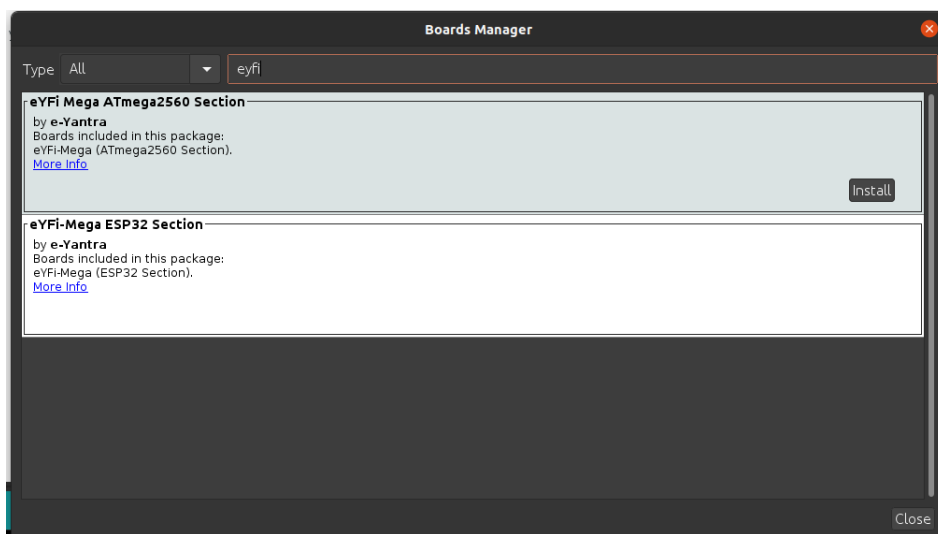


Figure 4: Installing board packages

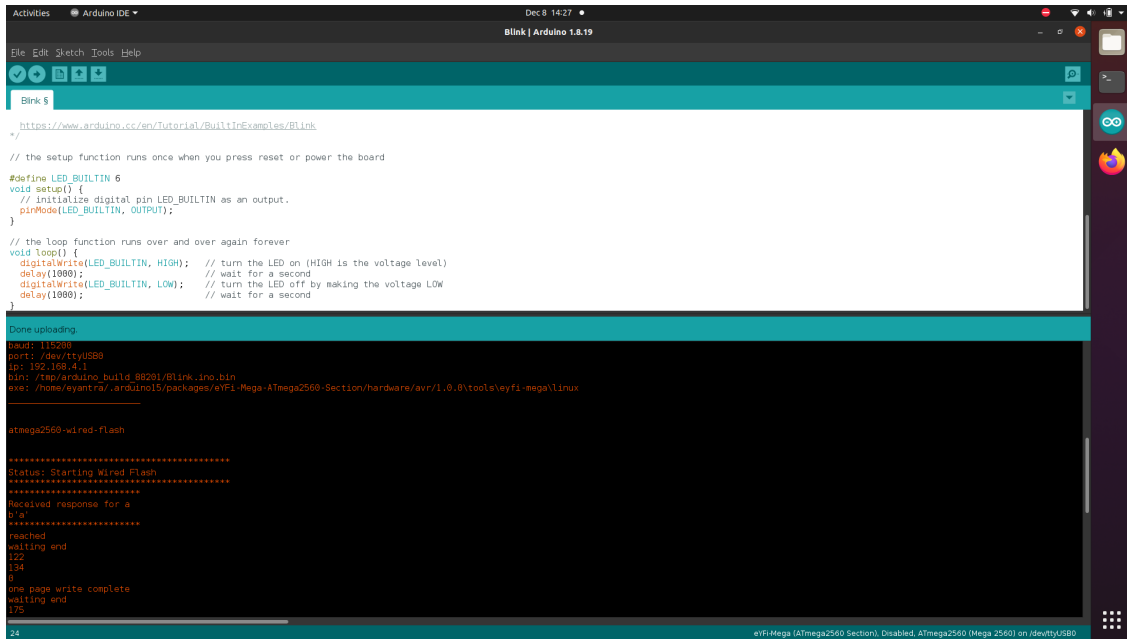
- eYFi-Mega board sections should be visible in **Tools** → **Board** selector dropdown options. Install both the packages.
- On Linux: change AVR & ESP flasher permissions: Find **.arduino15** folder in your \$HOME directory

```
chmod u+x ~/.arduino15/packages/eYFi-Mega-ATmega2560-Section/hardware/avr/1.0.0/tools/eyfi-mega/linux/eyfi*
```

You may require sudo access while running the above command.

4.2 Uploading program on ATmega2560 / ESP32 using Arduino IDE

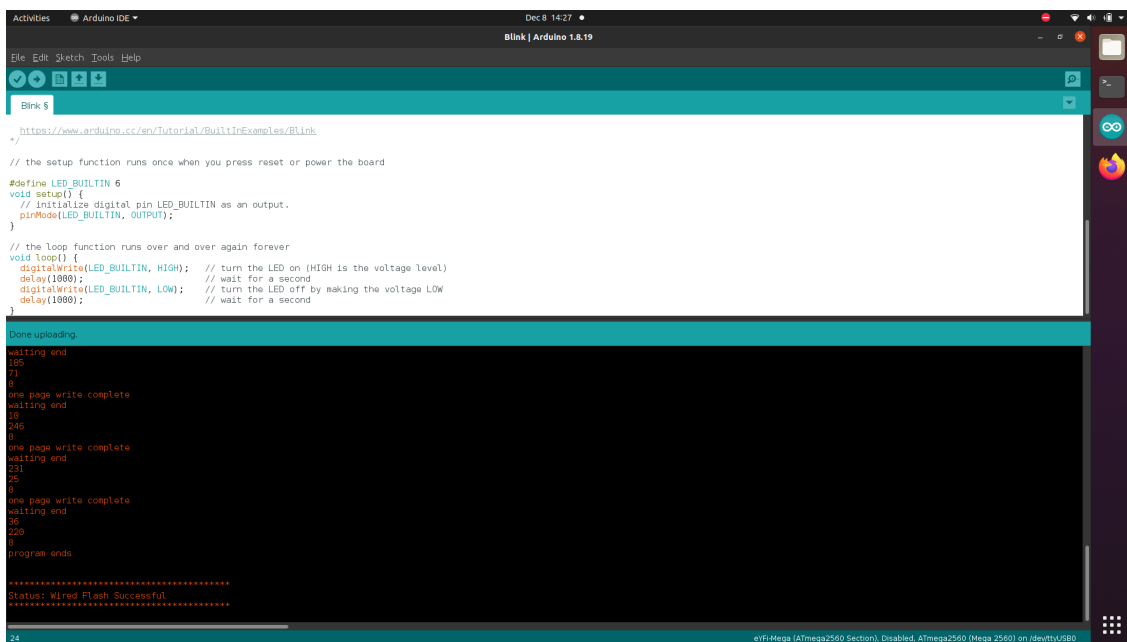
- Select board from **Tools** → **Board** selector dropdown options.
- Select USB port connected to eYFi-Mega board from **Tools** → **Port** selector dropdown
- **Compile & Upload**



The screenshot shows the Arduino IDE interface with the Blink sketch loaded. The Serial Monitor is open, displaying the progress of the upload. The status bar at the bottom indicates 'eYFi-Mega (ATmega2560 Section), Disabled, ATmega2560 (Mega 2560) on /dev/ttyUSB0'.

```
Done uploading.
Data: 115000
Port: /dev/ttyUSB0
IP: 192.168.4.1
Bin: /tmp/arduino-build-88281/Blink-ino-bin
Exec: /home/ryantav/.arduino15/packages/eYFi-Mega-ATmega2560-Section/hardware/avr/1.0.0/tools/avrdude-mega/linux
-----
atmega2560-wired-Flash
-----
Status: Starting Wired Flash
-----
Received response for a
b'a'
-----
reached
waiting and
122
134
8
one page write complete
waiting and
175
-----
Status: Wired Flash Successful
-----
```

Figure 5: Flashing started for ATmega2560



The screenshot shows the same Arduino IDE interface as Figure 5, but the upload process is now complete. The Serial Monitor displays the final status of the flash operation.

```
Done uploading.
waiting and
185
71
8
one page write complete
waiting and
13
206
8
one page write complete
waiting and
231
25
8
one page write complete
waiting and
36
229
8
program ends
-----
Status: Wired Flash Successful
-----
```

Figure 6: Flashing completed for ATmega2560

eYFi-Mega Wireless Serial Terminal

We have developed an ESP32 application which allows ATmega2560 to transmit its data on UART #0 over Wi-Fi. So with the help of this application, you can receive serial data over Wi-Fi at your side.

- To use this first, you would have to flash this application which you can download from eYFi-Mega's website in the user-app partition of ESP32.
- After that make sure the S2 switch is towards the Wi-Fi symbol. This will connect UART #0 of ATmega2560 with UART #1 of ESP32.
- To start this application switch S1 away from the Wi-Fi symbol and then press **ESP_RESET** button, to start the application.
- In your ATmega2560 application make sure that you are sending data to UART #0 at a baud rate of 115200.
- To receive data on your Linux system you can use **netcat** utility. Run the following command in the Terminal to receive data from the UART #0 of ATmega2560 over Wi-Fi,

```
nc 192.168.4.1 3333
```

You can also run this command on your Android Device. To run terminal commands on your Android Device use **Termux App** which you can download from Google Play Store. [Termux App Download URL](#)

Before you run this command, make sure that you are connected to the Wireless Access Point of eYFi-Mega. For this application the Wi-Fi SSID is **eYFi-Wireless-Serial** and there is **no** password.

In the Quick Bytes page of eYFi-Mega which you can find at eYFi-Mega's website, you can find the example code for using this Wireless Serial Terminal.

```
nc 192.168.4.1 3333
nc 192.168.4.1 3333 80x24
> nc 192.168.4.1 3333
Hello from ATmega 2560!!
Hello from ATmega 2560!!
Hello from ATmega 2560!!
Hello from ATmega 2560!!
Hello from ATmega 2560!!
```

Figure 7: eYFi-Mega Wireless Serial Terminal Output



Know-hows for Linux systems

A.1 Kill Any Command

While any command is being executed either in your terminal and you want to kill it, just press **CTRL+C** in the terminal.

A.2 Linux and Blank Spaces

Linux systems are not good in handling blank spaces so, we recommend not to have any blank spaces in your folder name or filename. Use **camelCase** or **snake_case** instead of having a blank space between two words.



Flashing eYFi-Mega ESP32 files

In case you accidentally erase any of the important firmware which is required for eYFi-Mega to function properly, you can always download them from our website and flash them again. In this section, we will discuss how you can do that.

To flash firmware in ESP32 you would need **esptool**. The **esptool** and the firmware described in the following sections can be downloaded from eYFi-Mega's website which can be found at [e-Yantra's products page](#).

B.1 Flashing eYFi-Mega ESP32 Bootloader

Copy the bootloader file in the esptool folder and execute the following command to flash the bootloader.

```
$ python esptool.py -b 921600 --port /dev/ttyUSB0 write_flash  
0x1000 eyfi-mega_esp32_bootloader.bin
```

B.2 Flashing eYFi-Mega ESP32 Partition Table

Copy the partition table file in the esptool folder and execute the following command to flash the partition table.

```
$ python esptool.py -b 921600 --port /dev/ttyUSB0 write_flash  
0x8000 eyfi-mega_esp32_partitions.bin
```

B.3 Flashing eYFi-Mega ESP32 OTA Application

Copy the OTA Application file in the esptool folder and execute the following command to flash the OTA Application.


```
$ python esptool.py -b 921600 --port /dev/ttyUSB0 write_flash  
0x10000 eyfi-mega_esp32_ota_app.bin
```

B.4 Flashing eYFi-Mega ESP32 User Application and Wireless Serial Terminal Application

Copy the Wireless Serial Terminal Application or User Application file in the esptool folder and execute the following command to flash the file.

```
$ python esptool.py -b 921600 --port /dev/ttyUSB0 write_flash  
0x110000 eyfi-mega_esp32_user_app.bin
```



Flashing Boot-loader on AVR

Note: Refer to this section only if you have accidentally overwritten the Boot-loader firmware of ATmega2560 on eYFi-Mega board.

Follow each of the steps given below carefully:

1. Components and Software required to get started:
 - (a) eYFi-Mega board
 - (b) USB micro-B plug to USB-A plug cable
 - (c) USB A-Male to B-Male cable
 - (d) Atmel AVRISP mkII (for more details, refer to [link](#))
 - (e) Atmel Studio 7 software (only on Windows platform)
 - (f) Boot-loader firmware of ATmega2560 micro-controller on eYFi-Mega board (can be downloaded from **Downloads** section on **eYFi-Mega** page, link: <https://e-yantra.org/productsdtl>).
2. Connect the USB A-Male to B-Male cable to the PC and Atmel AVRISP mkII.
3. Connect the 6-pin AVR ISP FRC cable of Atmel AVRISP mkII to the ICSP header on eYFi-Mega board.

Note: The notch on the FRC cable should match with the notch provided around the ICSP header.
4. Connect the USB micro-B plug to USB-A plug cable to the PC and Micro-USB connector on eYFi-Mega board.
5. Once the above connections are done, check the LED status on Atmel AVRISP mkII. If the LED color is **Green**, this means that everything is working fine and you can proceed with the next step. But, if the LED color is **Orange** or **Red**, this means that the connections between the Atmel AVRISP mkII and eYFi-Mega board is weak.

6. Open the Atmel Studio 7 software. Select the option **Device Programming** or use the shortcut **Ctrl + Shift + P**. A dialog box will open as shown in Figure 8.
7. Under **Tool** option, select **AVRISP mkII**.
8. Under **Device** option, select **ATmega2560**.
9. Under **Interface** option, select **ISP**.
10. Click on **Apply**.
11. Click on **Read** to read the Device Signature.
12. Check whether the **Device Signature** value is **0x1E9801** and the **Target Voltage** is around **5V**.
13. Go to **Fuses** and select the Fuse settings as shown in the Figure 8.

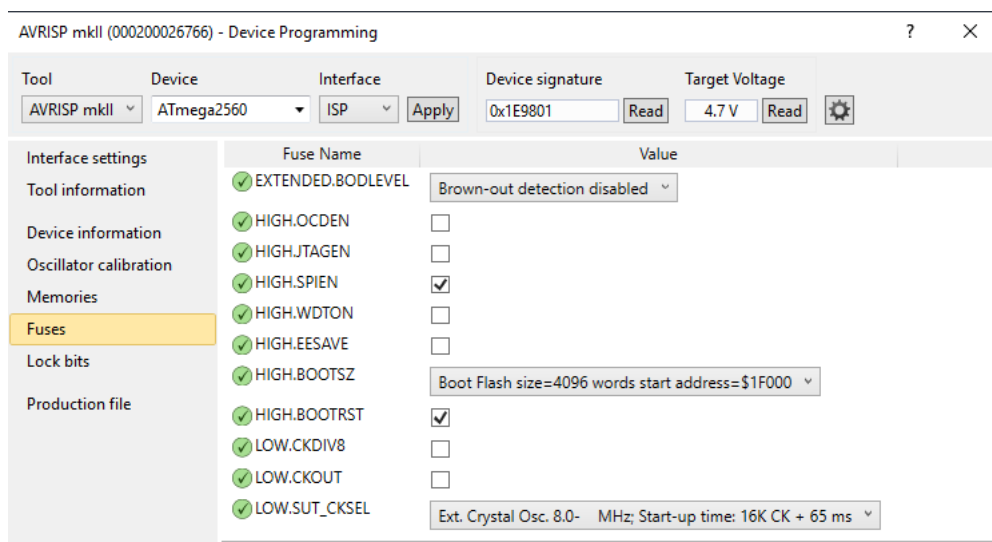


Figure 8: Device Programming dialog box in Atmel Studio 7

14. Make sure that the option of **Auto read** and **Verify after programming** are selected as shown in Figure 8.

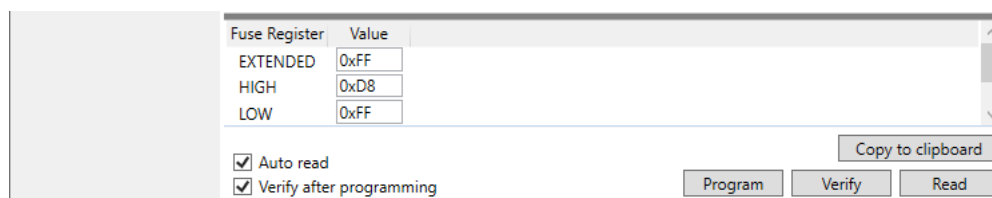


Figure 9: Fuse settings to be done

15. Once you are done with the above settings, click on **Program**.

16. After the programming is complete, you will see the message as shown in the below Figure 8.

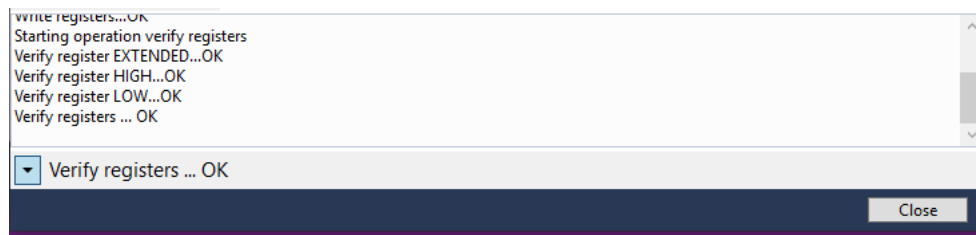


Figure 10: All OK message after programming the Fuses

17. Go to **Memories**, select **Erase Now** to erase the ATmega2560 chip before flashing the Boot-loader firmware on to it.
18. Make sure the options **Erase device before programming** and **Verify flash device after programming** are selected.
19. Select the Boot-loader firmware downloaded under the **Flash (256 KB)** section and press **Program**.
20. Once the programming is complete, you will get the message: **Verifying Flash....OK**. The LED color on Atmel AVRISP mkII will turn to **Orange** and will blink continuously.
21. Congrats! Your eYFi-Mega board is updated with the Boot-loader firmware of ATmega2560.

Revision History

Table 3: Revision History of the document

Date	Version	Release notes
December 14, 2019	v0.1	First release of the document
December 15, 2022	v0.1.1	Added Arduino IDE support Added default state of eYFi-Mega board Discarded VScode section