Bionics Kit reflash instructions

To reflash the Bionics programming to the Arduino-compatible microcontroller, complete the following steps. The first five sections of this document walk through the intial Arduino IDE, Git, and library/plugin installations; these only need to be done once on each PC you're using. If you want to reflash the microcontroller using a PC that has already gone through those installation steps, skip to section 6: Reflashing the microcontroller.

Hardware needed:

Computer with internet access
ESP32 microcontroller included in Bionics Kit
Micro-USB cable

1 Installing Arduino IDE

1. First, the Arduino IDE needs to be downloaded from the official Arduino site in order to program the microcontroller. The files and instructions are located here: https://www.arduino.cc/en/Main/Software

2. Ensure the Arduino IDE has been successfully installed before proceeding to the next section.

2 Installing Git

1. Additional information and files must be downloaded so the Arduino software can communicate with the ESP32 microcontroller. Make sure the Arduino IDE is closed before proceeding with these steps. Download the Git software program from the following location: https://git-scm.com/download

2. Perform the Git installation with the default settings.

3 Additional tools and libraries (ESP32 files)

1. The Windows Start menu should now have a program called Git GUI. Open the program and click on "Clone Existing Repository".

2. In the Clone Existing Repository menu, enter the following:

   Source location:

   https://github.com/espressif/arduino-esp32.git

   Target directory:

   C:/Users/[username]/Documents/Arduino/hardware/espressif/esp32

   Note: You need to put your username in the section highlighted in yellow.

3. Once you enter the information, click "Clone". After the files have been successfully downloaded, a new window appears that can be closed.

4. In Windows Explorer, navigate to the Target Directory path that you entered:

   C:/Users/[username]/Documents/Arduino/hardware/espressif/esp32

   Open the "tools" folder. There should be a file called "get.exe"; open and run it. This will download additional software related to the ESP32.

4 Additional tools and libraries (AsyncWebServer)

1. Open Git GUI; click on "Clone Existing Repository".

2. In the Clone Existing Repository menu, enter the following:

   Source location:

   https://github.com/me-no-dev/ESPAsyncWebServer.git

   Target directory:

   C:/Users/[username]/Documents/Arduino/libraries/AsyncWebServer

   Note: You need to put your username in the section highlighted in yellow.

3. Once you enter the information, click "Clone". After the files have been successfully downloaded, a new window appears that can be closed.

4. Open Git GUI again; click on "Clone Existing Repository".

5. In the Clone Existing Repository menu, enter the following:

   Source location:

   https://github.com/me-no-dev/AsyncTCP.git

   Target directory:

   C:/Users/[username]/Documents/Arduino/libraries/AsyncTCP

   Note: You need to put your username in the section highlighted in yellow.

6. Once you enter the information, click "Clone". Do not close the window that opens up once the cloning process is finished.

7. In the window that appears, click on Branch --> Checkout. In the Checkout Branch Window, choose the bullet point labeled Tracking Branch, then select origin/idf-update. Then click the Checkout button. (If prompted with an additional window, click ok.)

5 Additional tools and libraries (ESP32)

1. Open Git GUI; click on "Clone Existing Repository".

2. In the Clone Existing Repository menu, enter the following:

   Source location:

   https://github.com/jkb-git/ESP32Servo.git

   Target directory:

   C:/Users/[username]/Documents/Arduino/libraries/ESP32Servo

   Note: You need to put your username in the section highlighted in yellow.

3. The ESP32 Chip needs a special upload tool to load the website to the internal file system; make sure the Arduino IDE is closed for these steps. To install this tool, first go to the following URL:

   https://github.com/me-no-dev/arduino-esp32fs-plugin/releases/

4. Download the zip file of the sketch data upload plugin. The zip file should be named "ESP32FS-v0.1.zip".

5. Navigate to the Arduino folder and create a folder named "tools" if it doesn’t exist yet. The path to the Arduino folder should be similar to the following:

   C:\Program Files (x86)\Arduino\tools

   Note: This directory differs from the one used to install the libraries in the previous sections.

6. Unpack the previously downloaded zip archive (ESP32FS-v0.1.zip) into the tools directory. The path will look like:

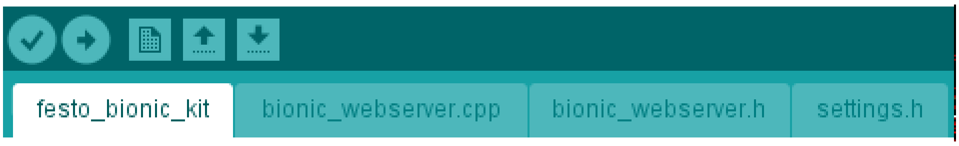
   /Arduino/tools/ESP32FS/tool/esp32fs.jar

6 Reflashing the microcontroller

1. On the Bionics4Education website, the Downloads section contains a zip file with the program for upload to the microcontroller. Download the zip file named "festo_bionic_kit.zip"; extract the folder "festo_bionic_kit" somewhere on your hard drive.

2. Open the extracted folder. Double-click to open the "festo_bionic_kit.ino" file. A dialogue box may pop up asking for a password: use "education" and click OK. The program should open in the Arduino IDE.

3. Ensure that the "settings.h" and "bionic_webserver.h" header files and the "bionic_webserver.cpp" file also opened with the .ino file. The files should show in additional tabs and look like this:



4. Ensure the correct board is selected in the Arduino IDE. Select Tools --> Board --> ESP32 Dev Module

5. Connect the microcontroller to the PC via a micro-USB cord. A green light on the board should indicate that it is connected and powered.

6. Click Tools --> Port and select the correct port for the connected microcontroller. If the microcontroller is not being detected by the Arduino IDE (either in the port section or while uploading the program), you may need to upload an additional driver; see the Troubleshooting section for information on solving this issue.

7. You should now be ready to upload the program to the microcontroller. Make sure the "festo_bionic_kit" program tab is actively selected and click either Sketch --> Upload or the right-pointing arrow icon ➜. The upload may take a few minutes.

8. After the program is finished uploading, the files for the Bionic Kit control website need to be uploaded. Click on Tools --> ESP32 Sketch Data Upload.

9. When that upload is complete, reset the ESP32 by pressing down on the gray button on the microcontroller.

10. Once the program is done compiling and is finished uploading, you can disconnect the microcontroller from the PC and close out the Arduino IDE program.

Troubleshooting

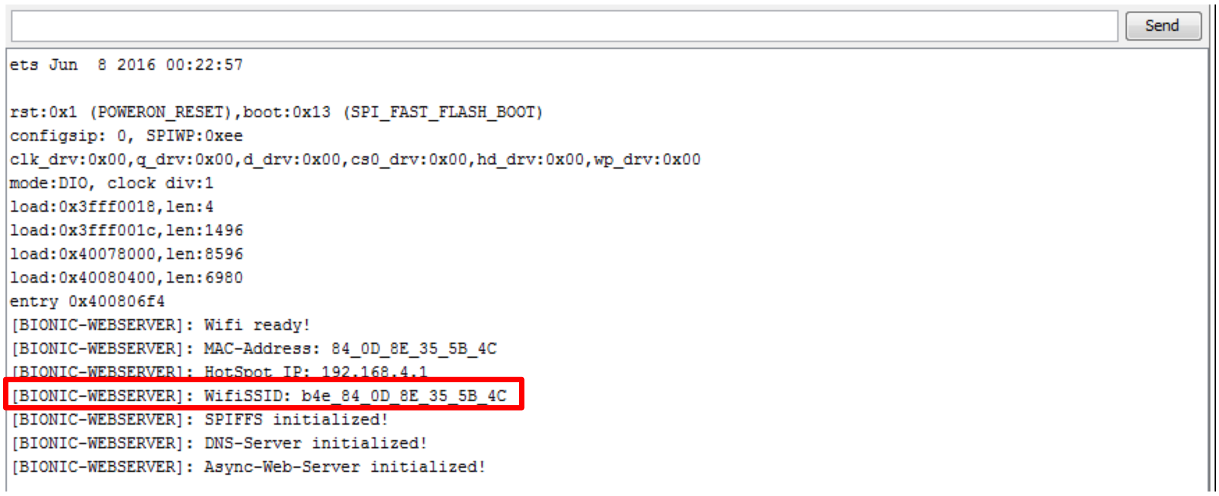
WiFi connection:

The WiFi network name is unique to each microcontroller. The WifiSSID for a particular microcontroller can be identified by completing the following steps:

1. Connect the microcontroller to a PC with a micro-USB cable.

2. Open the Arduino IDE. Select the Serial Monitor by either clicking Tools --> Serial Monitor or the magnifying glass icon 🔍.

3. Reset the microcontroller by pressing down on the reset button. The output window should identify the WifiSSID:



```
ets Jun  8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1496
load:0x40078000,len:8596
load:0x40080400,len:6980
entry 0x400806f4
[BIONIC-WEBSERVER]: Wifi ready!
[BIONIC-WEBSERVER]: MAC-Address: 84_0D_8E_35_5B_4C
[BIONIC-WEBSERVER]: HotSpot IP: 192.168.4.1
[BIONIC-WEBSERVER]: WifiSSID: b4e_84_0D_8E_35_5B_4C
[BIONIC-WEBSERVER]: SPIFFS initialized!
[BIONIC-WEBSERVER]: DNS-Server initialized!
[BIONIC-WEBSERVER]: Async-Web-Server initialized!
```

4. The WifiSSID should be available when searching for available WiFi networks on a PC or mobile device. You can connect to the WiFi network with the password "education"; the control interface can then be accessed by opening a web browser and navigating to www.b4e.de or 192.168.4.1

Using multiple microcontrollers:

If using multiple Bionic Kits/microcontrollers or multiple devices to control the microcontrollers, make sure any autoconnect options are turned off when connecting to a microcontroller network via PC or mobile device. Only one device can be connected to the WiFi network at a time, so disabling autoconnect prevents devices from automatically connecting to microcontrollers when they are first powered on, and therefore prevents any potential device conflicts.

When switching between multiple devices to connect to a microcontroller, make sure the device that is currently connected to the network is manually disconnected before trying to connect the new device.

Driver installation needed:

When the microcontroller is connected via PC, it may require a driver to be installed in order to communicate correctly and be properly detected by the Arduino IDE. The following link has information regarding the installation of device drivers if you encounter this type of problem with the microcontroller:

https://www.ftdichip.com/Support/Documents/InstallGuides.htm