Installation Guide Bionic-Kit

In this guide we provide you with instruction on how to write and deploy code on the Bionics-Kit. We provide guides for Ubuntu Linux and Windows 10.

Ubuntu

Arduino IDE

This guide is tested on Ubuntu 16.04. It should work with subsequent versions.

To develop code for the Bionic-Kit the Arduino IDE is required.

In case you encounter an error, visit the Arduino Troubleshooting Page

1. The current version can be retrieved here.

4. Now execute the installer using

3. Navigate to the extracted folder and grant the installer execution rights:

2. Extract the archive to a location of your choice, for example in your /home folder.

- sudo chmod +x install.sh
- ./install.sh
- To start the Arduino IDE run the binary inside the extracted Arduino folder using the terminal:
- ./arduino

Additional Packages

To compile the source code of the Bionic-Kit, additional libraries need to be installed. **Espressif ESP32** The Bionic-Kit is based on the ESP32 SoC from Espressif. To use the ESP32 we first install the

asccording Arduino Core module. Follow the instructions on the official Espressif Arduino Github

1. For the purpose of displaying the control website of the Bionic-Kit a special web server for the

Asynchron web server for ESP32

ESP32 is needed. Therefore run:

cd ~/Arduino/libraries 2. Then clone the repository of the web server to the present directory:

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

3. The web server needs an additional package. Therefore stay in the libraries folder and clone the AsyncTCP repository as well:

git checkout idf-update

might not be necessary.

- git clone https://github.com/me-no-dev/AsyncTCP.git AsyncTCP
- 4. A special branch of the AsyncTCP has to be used, because of the current development status of the AsyncWebServer. For this, enter the directory and checkout the idf-update branch with: cd AsyncTCP
- **ESP32 Servo Library** The last library which is needed to get the Bionic-Kit running is the ESP32Servo library. 1. For this go back to the libraries folder and clone the following repository:

This extra step is necessary for the Asyncwebserver Version dated 26.09.2018. In future releases this step

cd ~/Arduino/libraries git clone https://github.com/jkb-git/ESP32Servo.git ESP32Servo 4. Sketch Data Upload Plugin

To load the website to the internal file system of the ESP32 Chip a special upload tool is needed.

download the ZIP file of the sketch data upload plugin.

1. For this purpose go to https://github.com/me-no-dev/arduino-esp32fs-plugin/releases/ and

2. Then navigate to the Arduino folder and create a folder named "tools" if it doesn't exist yet.

<home_dir>/Arduino/tools/ESP32FS/tool/esp32fs.jar). 4. Then restart the Arduino IDE if its open.

This installation guide is tested on Windows 10.

cd ~/Arduino

mkdir tools

Arduino IDE

here.

Step-by-step:

run

needed.

ESP32 Servo Library

Sketch Data Upload Plugin

4. Restart the Arduino IDE

esptool not found please try the follwing:

Locate the file esptool.exe at

Additional Packages

Windows 10

3. Unpack the previous downloaded ZIP archive into tools directory (the path will look like

For the usage of the Bionic-Kit the Arduino IDE is needed. Install it from the Microsoft Store. You can also download the latest version from here.

For the additional Packages next to the Esperessif ESP32 Package you can click on the github link,

Then open the Project-Sketch (the festo_bionic_kit file) and inside the Ardunio IDE select

The Bionic-Kit is based on the ESP32 SoC from Espressif, therefore the Arduino Core for the SoC is

2. Open a Git Bash session in [ARDUINO_SKETCHBOOK_DIR]/hardware/espressif/esp32 and

3. Navigate to [ARDUINO_SKETCHBOOK_DIR]/hardware/espressif/esp32/tools and run the

For the purpose of displaying the control website of the Bionic-Kit a special web server for the ESP32 is

get exe. This might take up to an hour. Make sure your PC doesn't go to standby or shuts down.

mandatory. To install all needed files follow the instructions on the official Espressif Arduino GIT

To compile the source code of the Bionic-Kit, additional libraries are needed.

click on Clone or download and then on Download ZIP.

Sketch > Include Library > Add ZIP Library .

Espressif ESP32

In the open file dialog select downloaded .zip-file

git submodule update --init --recursive

1. Clone https://github.com/espressif/arduino-esp32.git to

[ARDUINO_SKETCHBOOK_DIR] is usually located at

C:/Users/[YOUR_USER_NAME]/Documents/Arduino

[ARDUINO_SKETCHBOOK_DIR]/hardware/espressif/esp32.Your

4. Open the Arduino IDE. Under Tools > Board you should see your ESP32 after connecting it.

The last library which is needed to get the Bionic-Kit running is the ESP32Servo library.

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

Clone the following Libraries and add them to the Arduino IDE:

<home_dir>/Arduino/tools/ESP32FS/tool/esp32fs.jar).

Compiling the Source Code

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

Clone the following Libraries and add them to the Arduino IDE:

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

Asynchron Web server for ESP32

To load the website to the internal file system of the ESP32 Chip a special upload tool is needed. 1. Go to https://github.com/me-no-dev/arduino-esp32fs-plugin/releases and download the ZIP file of the sketch data upload plugin. 2. Navigate to the Arduino folder and create a folder named tools if it doesn't exist yet.

3. Unpack the previous downloaded ZIP archive into tools directory (the path will look like

If the Tools -> ESP32 Sketch Data Upload fails due to the following error SPIFFS Error:

 Copy it to the parent folder [ARDUINO_SKETCHBOOK_DIR]\hardware\espressif\esp32\tools\ Restart the Arduino IDE

[ARDUINO_SKETCHBOOK_DIR]\hardware\espressif\esp32\tools\esptool\esptool.exe

- Auto Format Ctrl+T Archive Sketch bionic_edu_32 Fix Encoding & Reload 10 Serial Monitor Ctrl+Shift+M 11
- Port: "/dev/ttyUSB0" Arduino Industrial 101 27 Tick Get Board Info Linino One 28 29 Serv Programmer: "AVRISP mkII (serial)" Arduino Uno WiFi 30 Serv Burn Bootloader

○ Widora AIR

Nano32

O LOLIN D32

Port is named COM X, if you use Ubuntu your serial port is called /dev/ttyUSBX

Serial ports

36 int16_t g_actual_web_values_a[3]; //!< Storage for the WebServer values

40 int16_t g_servo_amplitude_max_i16 = 150; //!< Calculated max value of the fin amplitude 41 int16_t g_servo_amplitude_min_i16 = 40; //!< Calculated min value of the fin amplitude

7. Click on the upload button (Arrow) on the top left corner of the Arduino IDE to compile the code

44 const String PRINT_PREFIX = "[SKETCH]: "; //!< Prefix for the serial output of the BionicWebserv

O LOLIN D32 PRO

○ WEMOS LOLIN32

O Dongsen Tech Pocket 32

○ "WeMos" WiFi&Bluetooth Battery

6. Make sure that the right port is selected. For this select Tools -> Port . If you use Windows the

>\se; //!< Flag if there are new values for the servos

//!< Servo object to control the servo 1</pre>

//!< Servo object to control the servo 2</pre>

//!< Servo object to control the servo 3</pre>

//!< Servo object to control the servo 4</pre>

//!< Variable for the actual servo position</pre>

//!< Variable for the actual servo speed</pre>

//!< Instance of the BionicWebserver

>lse; //!< Flag if there are new values from the Webserver

/dev/ttyUSB0 < Timer object to control the speed of the bionic_fish</p>

//!< Calculated offset from the fin center point</pre>

Electronic SweetPeas - ESP320

Calculated max value of the fin ampl:

Calculated min value of the fin ampl:

Calculated offset from the fin center

Prefix for the serial output of the E

The output can be changed

- 1. Download or clone the Bionic-Kit source code 2. Save it to a location of your choice. 3. Open the *.ino file with the Arduino IDE 4. Connect the Bionic-Kit to the PC with an Micro-USB cable 5. In the Arduino IDE select Tools -> Board: -> ESP32 Dev Module <u>File Edit Sketch</u> <u>Tools</u> <u>H</u>elp Serial Plotter Ctrl+Shift+L 12 */ 13 WiFi101 Firmware Updater 14// I 15 #inc ESP Exception Decoder 16 #inc ESP32 Sketch Data Upload 17 #inc 18 #inc Arduino Pro or Pro Mini Flash Mode: "QIO" 19 #inc Arduino NG or older Flash Frequency: "80MHz" 20 #inc Arduino Robot Control Flash Size: "4MB (32Mb)" 21 22 #def PSRAM: "Disabled" Arduino Robot Motor Partition Scheme: "Default" Arduino Gemma 24 bool Upload Speed: "921600" there are new values for the servos Adafruit Circuit Playground 25 bool Core Debug Level: "None" there are new values from the Webserv Arduino Yún Mini 26 ject to control the speed of the bior ject to control the servo 1 ject to control the servo 2 31 Servo g_servo_03; ject to control the servo 3 ArbotiX Std 32 Servo g servo 04; ject to control the servo 4 of the BionicWebserver 34 BionicWebserver myServer(80) ○ ESP32 Wrover Module 36 int16_t g_actual_web_values_a for the WebServer values ESP32 Pico Kit ○ TTGO LoRa32-OLED V1 38 int16_t g_servo_pos_i16 = 90 for the actual servo position 39 int16_t g_servo_speed_i16 = (\cap XinaBox CW02 for the actual servo speed
- 14// I 15 #inc ESP Exception Decoder 16 #inc ESP32 Sketch Data Upload 17 **#inc** Board: "ESP32 Dev Module" 19 #inc Flash Mode: "QIO" 20 #inc Flash Frequency: "80MHz" Flash Size: "4MB (32Mb)"

Ctrl+Shift+M

Ctrl+Shift+L

40 int16_t g_servo_amplitude_ma; O sparkFun ESP32 Thing

42 int16_t g_servo_offset_i16 =

44 const String PRINT_PREFIX =

48 * @brief Blink with onboard

50 * EN: This function let the

45

46 47 / **

49 *

<u>F</u>ile <u>E</u>dit <u>S</u>ketch <u>Tools</u> <u>H</u>elp Auto Format

> Fix Encoding & Reload Serial Monitor

WiFi101 Firmware Updater

Partition Scheme: "Default"

29 Serv Programmer: "AVRISP mkII (serial)"

34 BionicWebserver myServer(80);

38 int16_t g_servo_pos_i16 = 90;

39 int16_t g_servo_speed_i16 = 0;

42 int16_t g_servo_offset_i16 = 0;

48 * @brief Blink with onboard LED

Gesichert

Gesichert

Gesichert

Gesichert

(c.

WLAN

b4e_84_0D_8E_35_5A_EC

Netzwerk- und Interneteinstellungen

Flugzeug-

Min

Direction:

Left

Swing:

Small

8. Where to go from here

control the servos using the Bionics-Kit website.

modus

Dient zum Ändern von Einstellungen. Beispielsweise kann eine Verbindung in eine getaktete Verbindung geändert werden.

(q)

5. Connect to b4e* wifi network with password education

Fish

Mobiler

Hotspot

□ 口 口 (*)

6. Once connected, open your web browser and open http://192.168.4.1

Serial Plotter

22 #def PSRAM: "Disabled"

Port: "/dev/ttyUSP0"

27 Tick Get Board Info

30 Serv Burn Bootloader

45 46 47 / **

24 bool Upload Speed: "921600"

25 bool Core Debug Level: "None"

31 Servo g_servo_03;

32 Servo g_servo_04;

bionic_edu_32

11 *

12 */

41 int16_t g_servo_amplitude_mir _ u-blox NINA-W10 series (ESP32)

- and upload it to the Bionic-Kit. This may take a while. 8. The last step is to upload the files for the Bionic-Kit control website. For this click on Tools -> ESP32 Sketch Data Upload Now reset the ESP32 by pressing down the reset button on the PCB **Connection test** 1. Connect the Bionic-Kit to the PC with a USB cable 2. Open the Serial Monitor of the Arduino IDE (magnifying glass symbol in the top right corner) 3. Now reset the ESP32 by pressing down the reset button. The console output will show: 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_5A_EC [BIONIC-WEBSERVER]: HotSpot IP: 192.168.4.1 [BIONIC-WEBSERVER]: WifiSSID: b4e_84_0D_8E_35_5A_EC [BIONIC-WEBSERVER]: SPIFFS initialized! [BIONIC-WEBSERVER]: DNS-Server initialized! [BIONIC-WEBSERVER]: Async-Web-Server initialized! 4. Open your wifi settings menu of your PC and search for the shown WifiSSID for example "b4e_84_0D_8E_35_5A_EC" Verbunden Gesichert
- **FESTO** BIONIC FISH Control your Bionic Fish. Speed:

Elephant

Middle

7. If the batteries are attached on the Bionic-Kit and the switch is turned on, you should now be able to

Chameleon

Max

Right

Big

14:06

25.10.2018

Useful development tools **EspExceptionDecoder** To decode the ESP32 stacktraces you can use the EspExceptionDecoder https://github.com/me-nodev/EspExceptionDecoder. For installation instructions follow the README in the repository. More on the ESP32 SoC If you are interested in more details regarding the ESP32 for Arduino explore the official GIT repository.