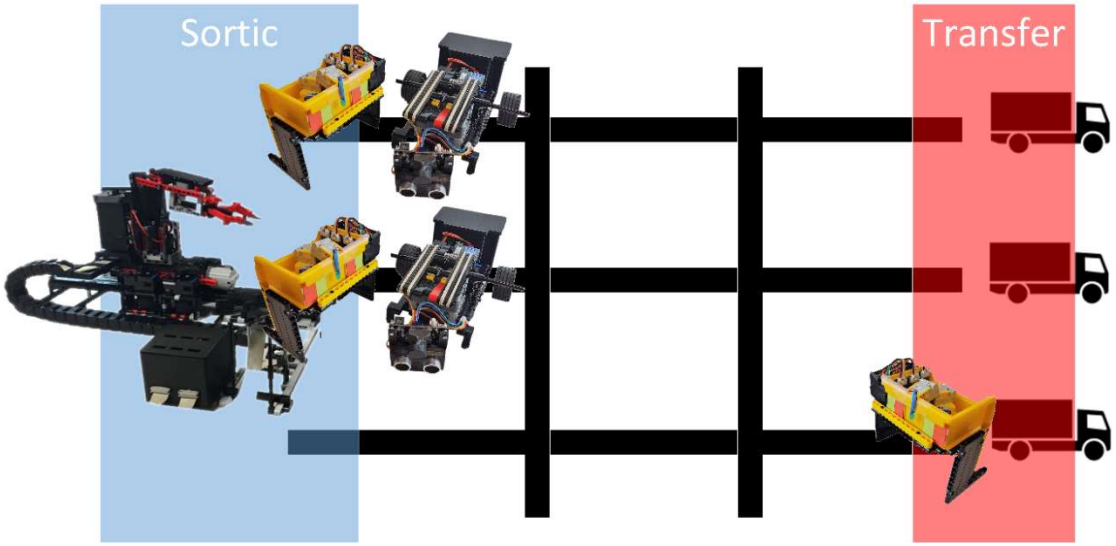
Installation Guide SmartFactory

Version: 1.1



Autor: Philip Zellweger

Erstellt am: 22. Mai 2018

Letzte Änderung am: 07. Oktober 2019

Purpose and aim of this document

Purpose and aim	This installation guide will help you get started and develop the SmartFactory project. It explains all the necessary software installations to work on the project.
System architecture	Which software of the SmartFactory project can be found on which hardware is shown in the system architecture.

Änderungsnachweis

Version	Datum	Autor	Änderungsgrund / Bemerkungen
1.0	01.10.19	Philip Zellweger	Ersterstellung
1.1	07.10.19	Philip Zellweger	Formatierung

Inhalt

1 Raspberry Pi

1.1 Hardware

1.1.1 Generally

GPIO **https://www.raspberrypi.org/documentation/usage/gpio/**

1.1.2 Raspberry Pi 3 Model B+

Specifications

The Raspberry Pi 3 Model B+ is the latest product in the Raspberry Pi 3 range.

- Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- 1GB LPDDR2 SDRAM
- 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
- Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
- Extended 40-pin GPIO header
- Full-size HDMI
- 4 USB 2.0 ports
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- 4-pole stereo output and composite video port
- Micro SD port for loading your operating system and storing data
- 5V/2.5A DC power input
- Power-over-Ethernet (PoE) support (requires separate PoE HAT)

Datasheet

Raspberry-Pi-Model-Bplus-Product-Brief

Mechanical Drawing

rpi_MECH_3bplus.pdf

Schematic diagramms

rpi_SCH_3bplus_1p0_reduced.pdf

1.2 Software

Operating system

To set up the Raspberry pi, you need to download the Raspbian software.
If there is enough memory, it is recommended to download the Raspbian Buster with desktop and recommended software, otherwise the Raspbian Buster with desktop can be downloaded as well.
Next, the image must be written to the SD card. Here is a guide for it.
It is recommended to use a 16GB SD card.

MQTT-Broker Mosquitto

A broker is used for the communication. For this, the broker have be installed on raspberry pi, which wants to be used as a broker. The broker used is "Mosquitto". Here's a guide to install "Mosquitto" on raspberry pi.

Node red

Node red can be used for communication via MQQT. In addition, a graphical user interface on raspberry pi can be realized. To learn more about Node red and download it on raspberry pi click here.
To install the graphical user interface, run “cd ~/.node-red” in terminal, after that run “npm install node-red-dashboard”. The graphical user interface will now be installed.

1.3 Network / Operation

1.3.1 SSH Connection via Laptop

Purpose

The SSH connection allows direct access to the raspberry pi terminal via the WLAN.

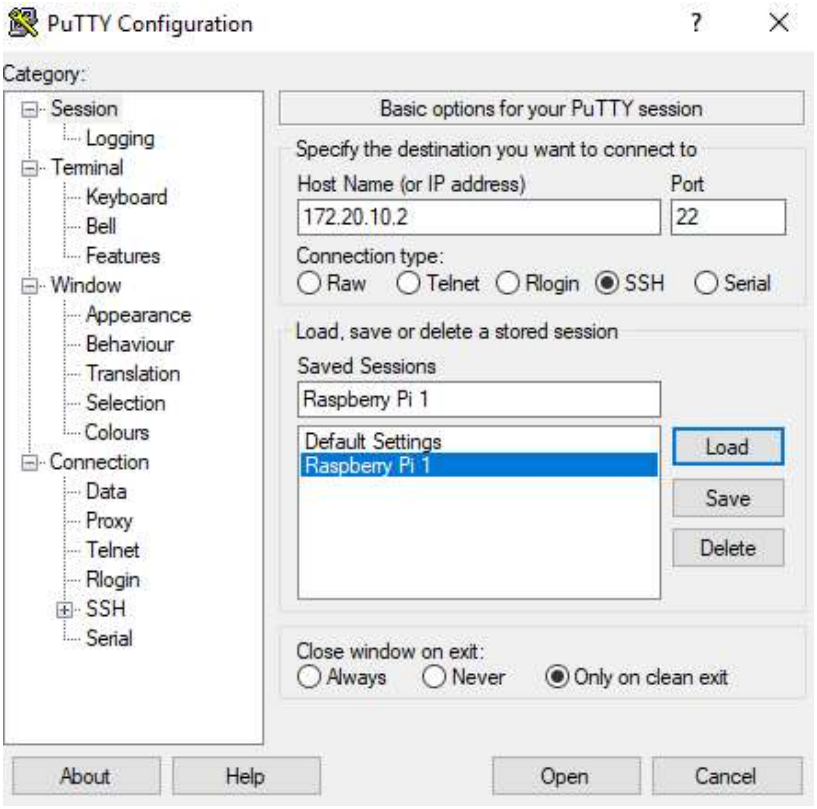
Raspberry pi

The SSH connection must be activated on the raspberry pi. This can be done with existing screen and keyboard via the terminal. If there is no screen and keyboard available, the settings have to be made differently.

SSH via terminal
SSH without screen and keyboard → Step 3: Enabling SSH Without Monitor

Laptop

On the laptop the software Putty must be installed, which enables the SSH connection to the laptop. Instructions for installing Putty.
Host Name: "IP Address"
Port: 22
Connection type: SSH
If you want to save the connection, you can save it at saved sessions with a name.



Userinterface Putty

1.3.2 VNC Remote Connection

Purpose

The VNC connection allows direct access to the raspberry pi desktop (complete external operation) via WLAN.

Raspberry pi

On the raspberry pi the VNC connection must be activated. This can be done via terminal. Run «sudo raspi-config» on raspberry pi, navigate to “Interfacing Options” → “VNC”, select “Yes”.

Laptop

The VNC Viewer is installed by REALVNC on the laptop. The software is the user interface for the operation of the raspberry pi via remote. You can download VNC here.

To create a new connection:

- File -> new connection
- Enter IP address
- define name for raspberry pi (to use for multiple use)

2 Adafruit Feather M0 WiFi - ATSAMD21 + ATWINC1500

Specifications

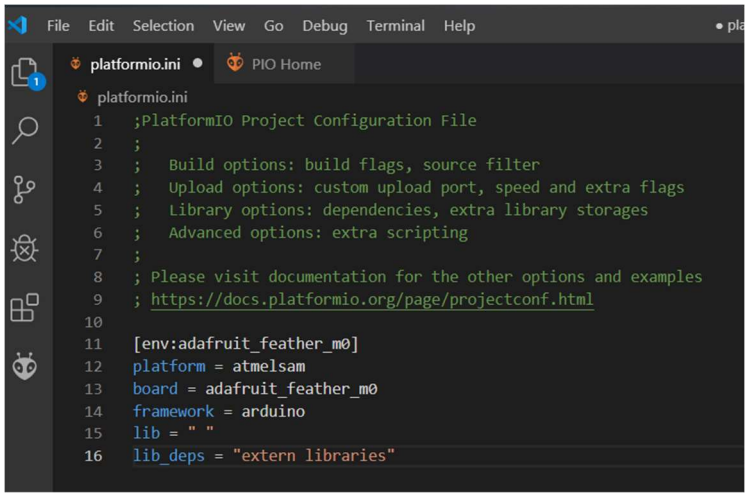
- Measures 2.1" x 0.9" x 0.3" (53.65mm x 23mm x 8mm) without headers soldered in. Note it is 0.1" longer than most Feathers
- Light as a (large?) feather - 6.1 grams
- ATSAMD21G18 @ 48MHz with 3.3V logic/power
- 256KB FLASH, 32KB SRAM, No EEPROM
- 3.3V regulator (AP2112K-3.3) with 600mA peak current output, WiFi can draw 300mA peak during xmit
- USB native support, comes with USB bootloader and serial port debugging
- You also get tons of pins - 20 GPIO pins
- Hardware Serial, hardware I2C, hardware SPI support
- 8 x PWM pins
- 10 x analog inputs
- 1 x analog output
- Built in 200mA lipoly charger with charging status indicator LED
- Pin #13 red LED for general purpose blinking
- Power/enable pin
- Mounting holes
- Reset button

Datasheet
Mechanical Drawing
Schematic Drawing

- https://learn.adafruit.com/adafruit-feather-m0-wifi-atwinc1500/downloads

platformio.ini

To use the Adafruit Feather, you have to write the text on the picture in your platformio.ini file.



platformio.ini

3 Arduino Uno

Specifications

- Microcontroller: ATmeag328P
- Operating Voltage: 5V
- Input Voltage: 7-12V
- Input Voltage (limit) : 6-20V
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- PWM Digital I/O Pins: 6
- Analog Input Pins: 6
- Flash Memory: 32 KB (ATmega328P) of which 0.5 KB used by bootloader
- SRAM: 2 KB
- EEPROM: 1 KB
- Clock Speed: 16 Mhz
- Length: 68.6mm
- Width: 53.4mm
- Weight: 25g

Datasheet
Mechanical Drawing
Schematic Drawing

- https://docs.espressif.com/projects/esp-idf/en/latest/hw-reference/modules-and-boards.html#esp-modules-and-boards-esp32-wroom-32

platformio.ini

To use the Arduino Uno, you have to write the text on the picture in your platformio.ini file.

```
[env:uno]
platform = atmelavr
framework = arduino
board = uno
```

platformio.ini

4 Esp32 DevKit C

Specifications

- Model: NodeMCU ESP32
- Article No.: SBC-NodeMCU-ESP32
- Type: ESP32
- Processor: Tensilica LX6 Dual-Core
- Clock Frequency: 240 MHz
- SRAM: 512 kB
- Memory: 4 MB
- Wireless Standard: 802.11 b/g/n
- Frequency: 2.4 GHz
- Bluetooth: Classic / LE
- Data Interfaces: UART / I2C / SPI / DAC / ADC
- Operating Voltage: 3,3V (operable via 5V-microUSB)
- Operating Temperature - 40°C - 125°C
- Dimensions (W x D x H) : 48 x 26 x 11.5 mm

Datasheet
Mechanical Drawing
Schematic Drawing

- https://docs.espressif.com/projects/esp-idf/en/latest/hw-reference/modules-and-boards.html#esp-modules-and-boards-esp32-wroom-32

platformio.ini

To use the Esp32, you have to write the text on the picture in your platformio.ini file.

```
[env:esp32doit-devkit-v1]
platform = espressif32
board = esp32doit-devkit-v1
framework = arduino
```

platformio.ini

5 Visual Studio Code with PlatformIO

IDE

The IDE used is Visual Code Studio with the PlatformIO plugin. The PlatformIO plugin allows connection to many different microcontrollers. Visual Studio Code is used to organize and develop the software.

Visual Studio Code

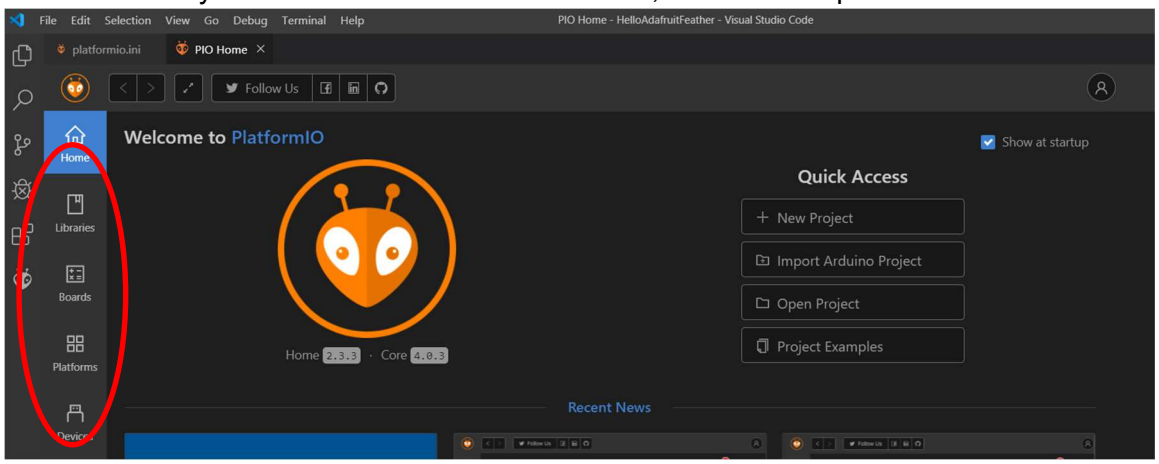
Visual Studio Code can be easily downloaded here.

PlatformIO plugin

To install and learn more about the PlatformIO plugin you can click here.

Boards
Libraries
Platforms

On PIO Home you can search for different boards, libraries and platforms.



User interface Visual Studio Code with PlatformIO plugin

Platformio.ini

Every project needs a platformio.ini file! In the platformio.ini file the specifications are stored to work with the microcontrollers and to add external and internal libraries to compile the code.

Doxygen

To install and learn more about the Doxygen Documentation Creator click here.