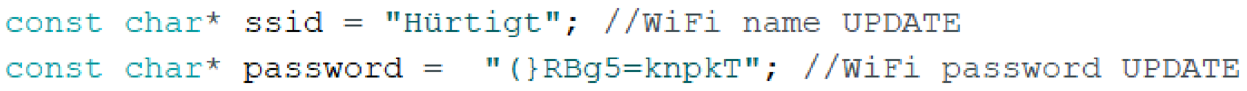
ESP32 setup:

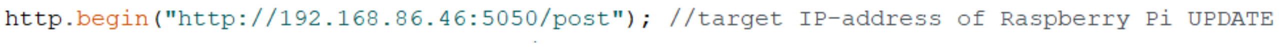
The code for the ESP32 is located in the ESP32_Sensor_system.ino file. After opening this file in the ArduinoIDE, three values need to be updated.

The wifi settings must be updated to the local wifi:



```
const char* ssid = "Hürtigt"; //WiFi name UPDATE
const char* password =  "(}RBg5=knpkT"; //WiFi password UPDATE
```

The final variable to change is the target IP address of the Raspberry Pi on which the server is running:



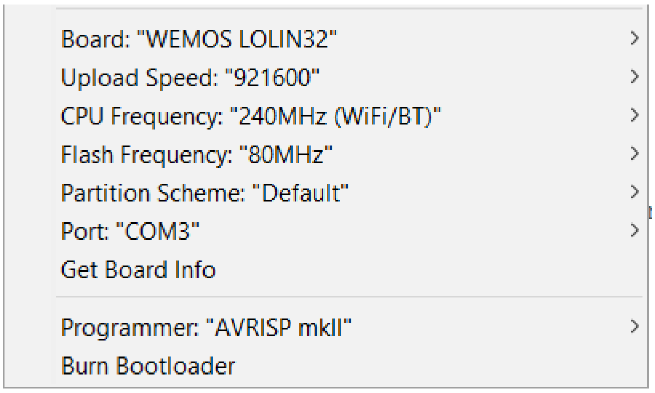
```
http.begin("http://192.168.86.46:5050/post"); //target IP-address of Raspberry Pi UPDATE
```

The target IP address can be found using the set up for the Raspberry PI guide.

To compile and run the program it is needed to download an extra library called NewPing. This is done by following the instructions from:

https://playground.arduino.cc/Code/NewPing/ (Look for the Download section)

Finally, before uploading the code, make sure the correct port is chosen. Furthermore, theses are the settings used in the prototype (although others may work):



Board: "WEMOS LOLIN32"
Upload Speed: "921600"
CPU Frequency: "240MHz (WiFi/BT)"
Flash Frequency: "80MHz"
Partition Scheme: "Default"
Port: "COM3"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader

NOTE:

Following warning is prompted and should be ignored:

“WARNING: library NewPing claims to run on avr, arm architecture(s) and may be incompatible with your current board which runs on esp32 architecture(s).”

Raspberry PI setup:

The raspberry Pi needs to be hooked up to a screen via a micro-HDMI connection, and a keyboard and mouse must also be connected.

After booting up the system, it is needed to change the WIFI connection to be on the local WIFI that the ESP32 is also connected to. The settings are found in the upper right corner at the WIFI-logo.

Next step for the Raspberry Pi is to find the local IP address. This is done via the terminal by writing “ifconfig”. This will then display the network settings of the device, and the IP we need is located under wlan0: and was in the demonstration “192.168.86.46”.

After finding the IP and updating this info in the ESP32 setup, running the server is next step. This is done again via the terminal. Start off by making sure you are in the dir “Desktop/smartpark”. Afterwards you should be able to run the python server using the syntax: “python3 SMART_Parking_V0.1.py”. The server should now be accessible via other devices via a webbrowser and the IP-address with “:5050” at the end.

Hardware

The prototype system is placed on three breadboards. The system should be ready so that only the power supply for the 5V and the connector for the ESP32 should be inserted. If the system however does not run, it is possible that there are faulty or unstable connections as we are using breadboards. Had the situation been normal and we were allowed on campus, we would have made the system on a PCB. So, if it does not work, please check for wires that have been disconnected and other obvious faults.