Exercise 1

Assembled the raspberry pi 3 and the case.

Gabriel: Reconfigured the ulnoiot.conf file with the new devicename and a new password.

Setup the the rapsi iot gateway with the provided SD-card image.
Connect to the gateway and start the clone process.
Change the directory to cd /src/rpi-clone and start clone process with the command:
sudo ./rpi-clone -v sda

Use the rootfs file system.

After the cloning process we used the port forwarding command:
ssh pi@192.168.12.1 -L 5901:localhost:5901
to connect with vncviewer.

Command for vncviewer:
vncviewer localhost:1

Soldering pin connectors onto the Wemos DI mini board.

Blinking LED:

Change acces to the arduino apps via:
chmod -R u+x /home/pi/apps/arduino/*

And flash the blink program to the board.

Failures:
Wrong board, Arduino Uno, also the wrong Port, COM1, were selected.

After changing the development board to Wemos DI mini and loading the correct blink program the board started blinking.

Remote control for blinking program:

We used an example program from the esp8266wifi folder to generate a webserver.
To control the LED, we used the URL.

http://server_ip/gpio/0 will set the GPIO2 low and the led is turned off,
http://server_ip/gpio/1 will set the GPIO2 high and the led is turned on.

Failures:
Before we used the WIFIServer example code we used two external libraries.
aREST for the rest webservice and aREST_UI for the web UI which are two simple buttons (On/Off).
Via Serial.print we logged the IP address used from the Wemos board to connect

from a computer to the provided webservice.
If we click the “ON” button the led blinks and a click on the “OFF” button turns off the LED by setting it to “LOW”.
The button could not control directly the blinking LED because it only can be set to 0 or 1. We needed an extra pin which is set to 0 or 1 and to read the actual pin state to trigger the LED blinking or LED off. This method using aREST_UI library does not work.