Smart Wearables for the Internet of Things
Block I – Sheet 1

The goal of this session is to replace the standard remote control of the quadcopter with an Arduino Due and a 2.4GHz transmitter module. The protocol for communication is already implemented on the Arduino\(^1\). This sheet guides you through the steps how to connect the module to the Arduino via SPI\(^2\). As a second task, the server necessary to communicate with the quadcopter is setup and a first connection to the quadcopter is made.

**Connect A7105 Module to Arduino via SPI (See last page for pinout)**

1.) Collect a breadboard and Arduino Due for your team along with some jumper wires.

2.) Provide power to the module:
   a. Connect VDD (3.3V) of Arduino to the „+“ rail of the board.
   b. Connect GND of Arduino to the „-“ rail of the board.
   c. Wire jumper cables from the GND rail to both GND pins of the A7105 module.
   d. Wire a cable to the VDD rail of the board.

\(^1\) [https://github.com/andihit/coptermanager-arduino](https://github.com/andihit/coptermanager-arduino)

3.) Connect SCK and MISO pins of SPI
   a. Connect MISO pin of Arduino to GPIO1 of the module.
   b. Connect SCK pin of Arduino to SCK pin of the module.

4.) Wire MOSI over the resistor and the SCS pin
   a. Connect MOSI pin to one side of the resistor
   b. Connect other side of resistor to "sdio" pin of module
   c. Wire Pin 10 of Arduino to "scs" of module.
Setup NodeJS server
Now, with the “transmitter” in place, we need a way to communicate with the laptop. Therefore, we will setup a NodeJS server, which will be used to send commands to the quadcopter and to receive commands from a smartwatch.

1.) Download NodeJS from https://nodejs.org/en/ (version 6.x.x) and install.
2.) Extract the provided sample code and navigate to the folder
3.) Install required packages: npm install express serialport
4.) Start server by typing: node test_server.js
5.) Open a browser and type: localhost:3000
6.) You should see an image of the quadcopter and can check the documentation of the serial communication library.
7.) Congrats, your server is running!

Run test script
Finally, let’s test if you have wired the breadboard correctly and we are able to connect to the quadcopter. Once, the copter is bound, the LEDs do not flash anymore and the motors will spin for 1 second.

1.) Connect the battery of the quadcopter. It should start to flash the leds.
2.) Connect the Arduino board via USB.
3.) Run the test script by typing: node test_copter.js
   2.1.) If the serial port is not found, open the library serial.js with an editor\(^3\) and edit the used port.
3.) If the copter is bound and the rotors moved, congratulations!

The communication between quadcopter and server is now established. In the next sessions, we will write a smartwatch app that sends controls to the server using SocketIO. The controls are then sent to the quadcopter via a NodeJS module. You can check out the available functions in the documentation available on your server.

\(^3\) Example: https://atom.io/
Pinout

1 Pinout A7105

2 Pinout SPI Arduino Due