Co-funded by the
Erasmus+ Programme
of the European Union

Remote Control – the second part

Meeting no. 8 Ibi(Spain)

Prepared by Tomasz Kądziołka
ZS10 Poland

Our team

-Aleksandra Pawłowska

sources, media documentation

-Tomasz Kądziołka

Software development (android, arduino), idea, coordination

-Paweł Modzelewski

Documentation, electronic consultation, prsentation

Remote control part two

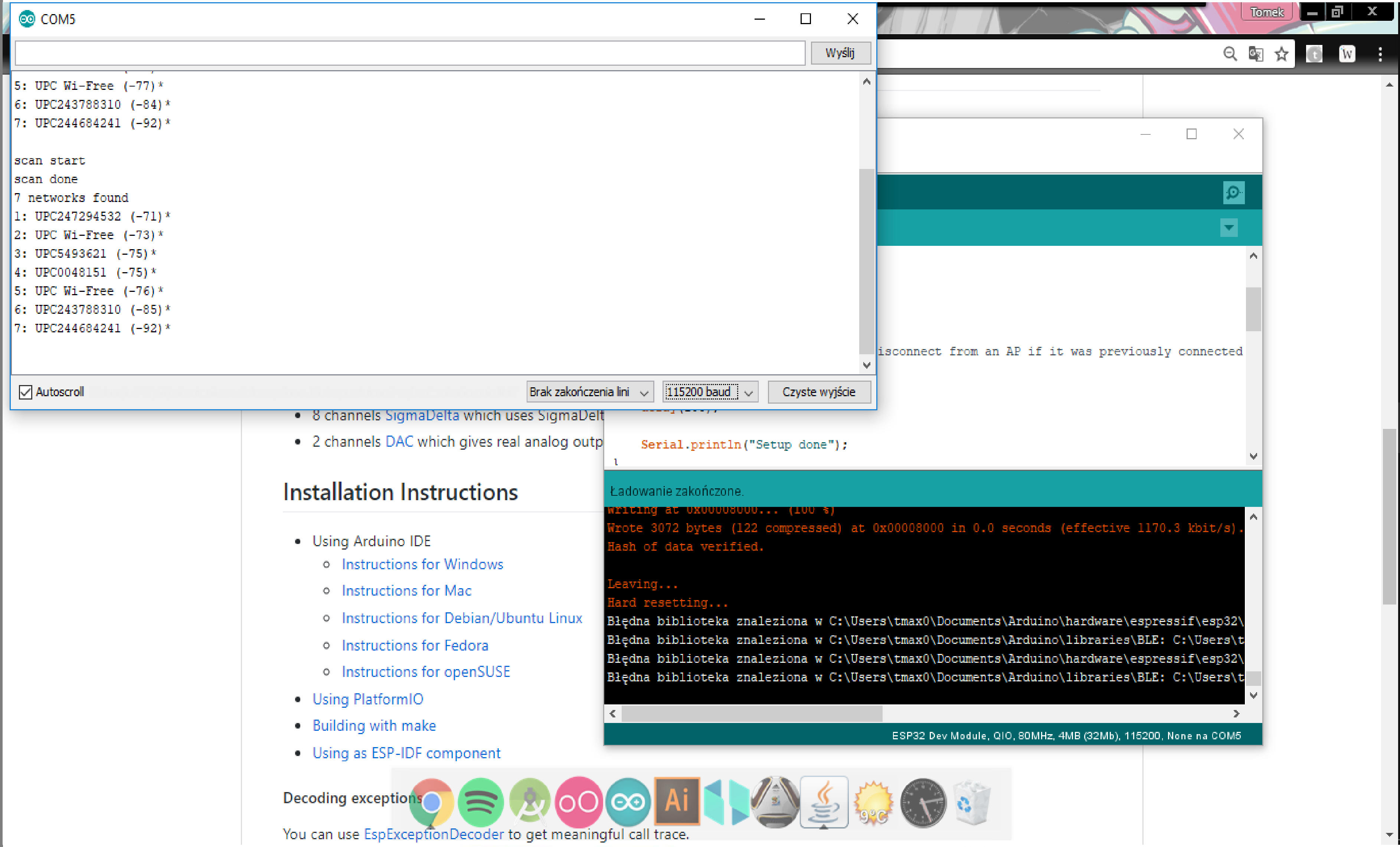
In march 2018 we announced a new extra activity with our first drone.

During second meeting in the Poland you could see elements and a software needed to achieve that connection.

ESP32

To create a connection between a drone and a phone we had to establish a wireless connection via wi-fi.

The first step was to turn an esp32 into access point mode. This mode is available from the public esp32 library.



COM5

Wyślij

```
5: UPC Wi-Free (-77)*
6: UPC243788310 (-84)*
7: UPC244684241 (-92)*

scan start
scan done
7 networks found
1: UPC247294532 (-71)*
2: UPC Wi-Free (-73)*
3: UPC5493621 (-75)*
4: UPC0048151 (-75)*
5: UPC Wi-Free (-76)*
6: UPC243788310 (-85)*
7: UPC244684241 (-92)*
```

☑ Autoscroll | Brak zakończenia lini | 115200 baud | Czyste wyjście

```
isconnect from an AP if it was previously connected

  Serial.println("Setup done");
}
```

Ładowanie zakończone.

```
writing at 0x00008000... (100 %)
Wrote 3072 bytes (122 compressed) at 0x00008000 in 0.0 seconds (effective 1170.3 kbit/s).
Hash of data verified.

Leaving...
Hard resetting...
Błędna biblioteka znaleziona w C:\Users\tmax0\Documents\Arduino\hardware\espressif\esp32\
Błędna biblioteka znaleziona w C:\Users\tmax0\Documents\Arduino\libraries\BLE: C:\Users\t
Błędna biblioteka znaleziona w C:\Users\tmax0\Documents\Arduino\hardware\espressif\esp32\
Błędna biblioteka znaleziona w C:\Users\tmax0\Documents\Arduino\libraries\BLE: C:\Users\t
```

ESP32 Dev Module, QIO, 80MHz, 4MB (32Mb), 115200, None na COM5

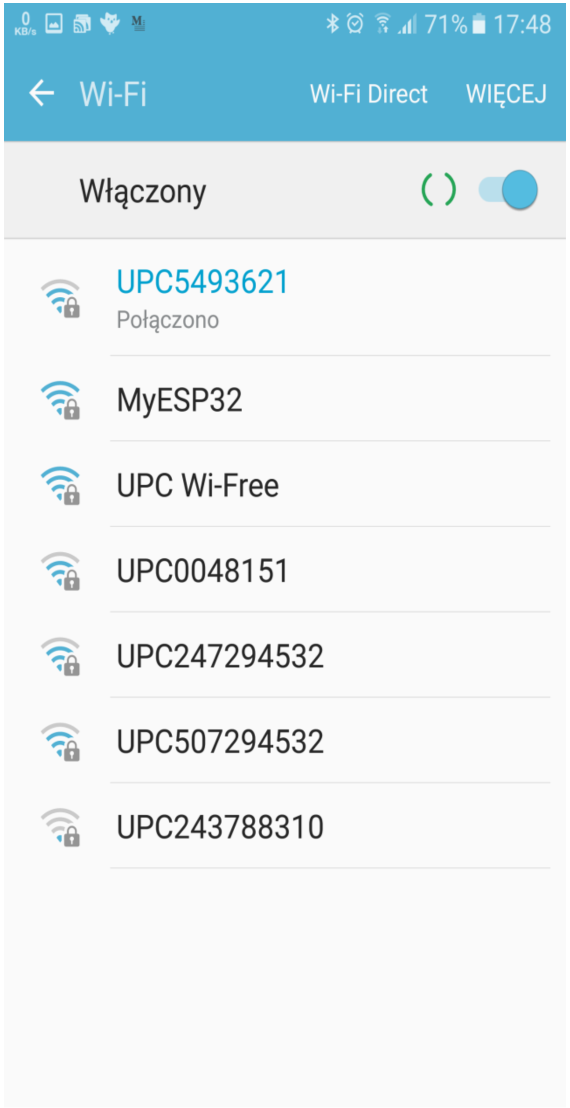
- 8 channels SigmaDelta which uses SigmaDelt
- 2 channels DAC which gives real analog outp

Installation Instructions

- Using Arduino IDE
  - Instructions for Windows
  - Instructions for Mac
  - Instructions for Debian/Ubuntu Linux
  - Instructions for Fedora
  - Instructions for openSUSE
- Using PlatformIO
- Building with make
- Using as ESP-IDF component

Decoding exceptions

You can use EspExceptionDecoder to get meaningful call trace.



71% 17:48

Wi-Fi

Wi-Fi Direct

WIĘCEJ

Włączony

UPC5493621
Połączono

MyESP32

UPC Wi-Free

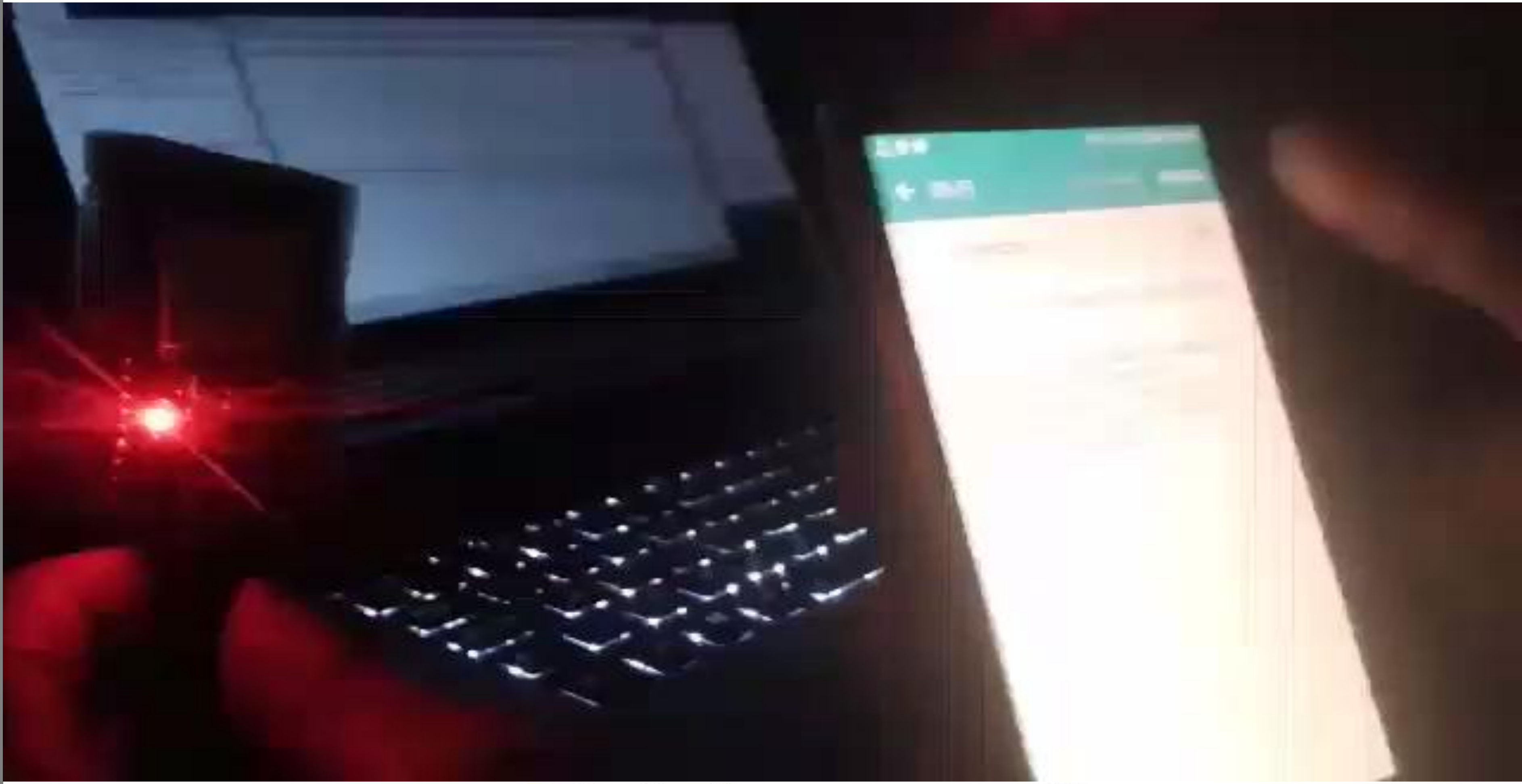
UPC0048151

UPC247294532

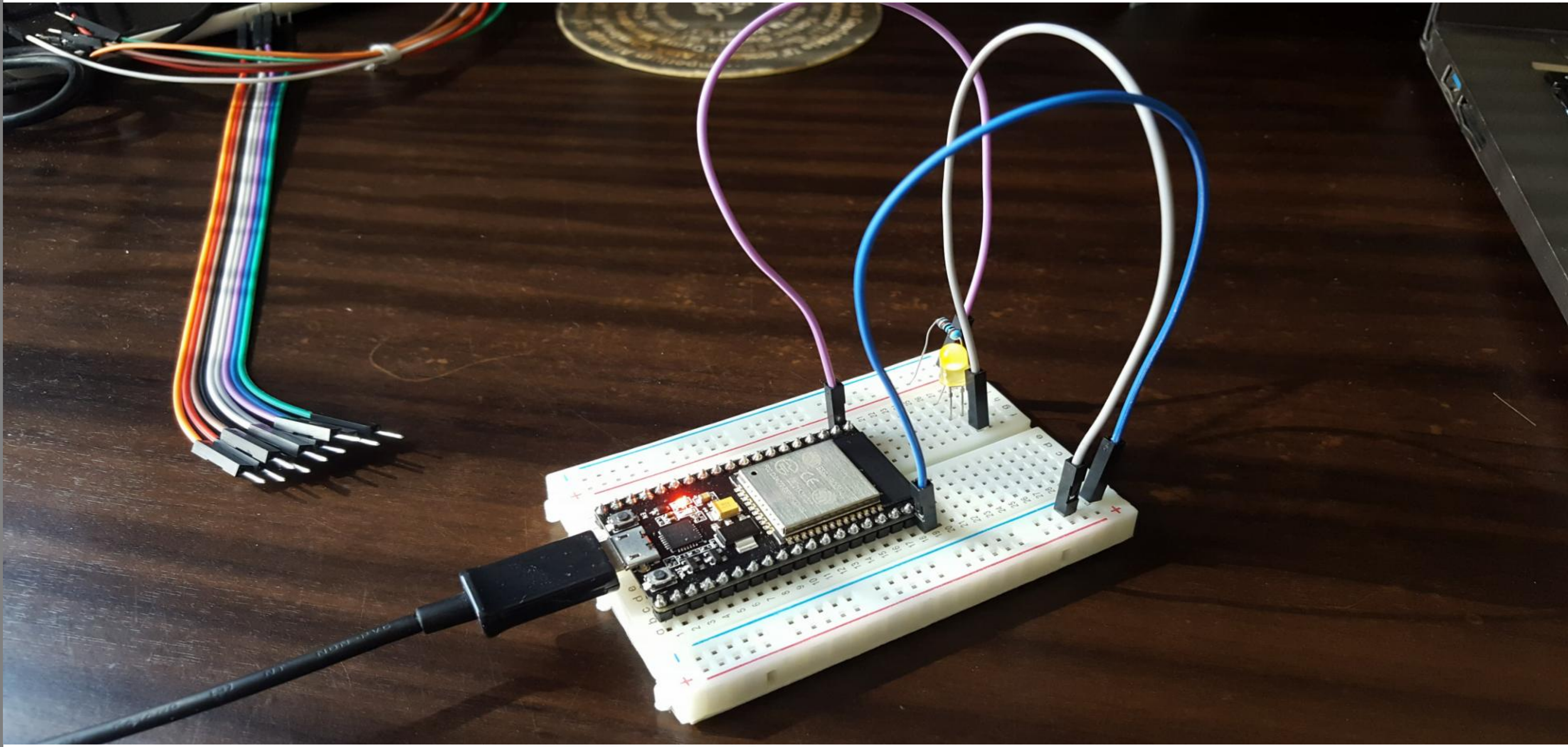
UPC507294532

UPC243788310

Indication of a wireless connection



Basis for the soft AP server



Explanation

We have put an esp32 on a Arduino breadboard and connected one resistor and yellow LED to indicate a connection to the AP server.

AP server

The soft AP server is a wireless serial alternative for native Arduino (UART).

This communication has a support for the multiple (RX,TX) pins.

It also contains packed of the common commands like begin(), available(), read().

HardwareSerial

Implicity Arduino uses **softwareSerial** library to estabilish an UART connection.

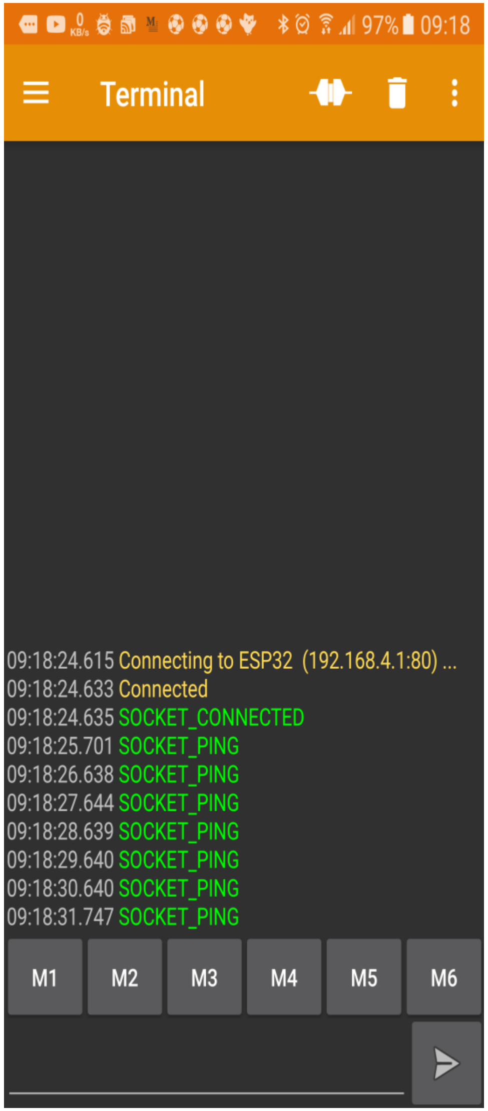
Esp constructors decided to implement faster and more flexible **hardwareSerial.**

Serial data transfer

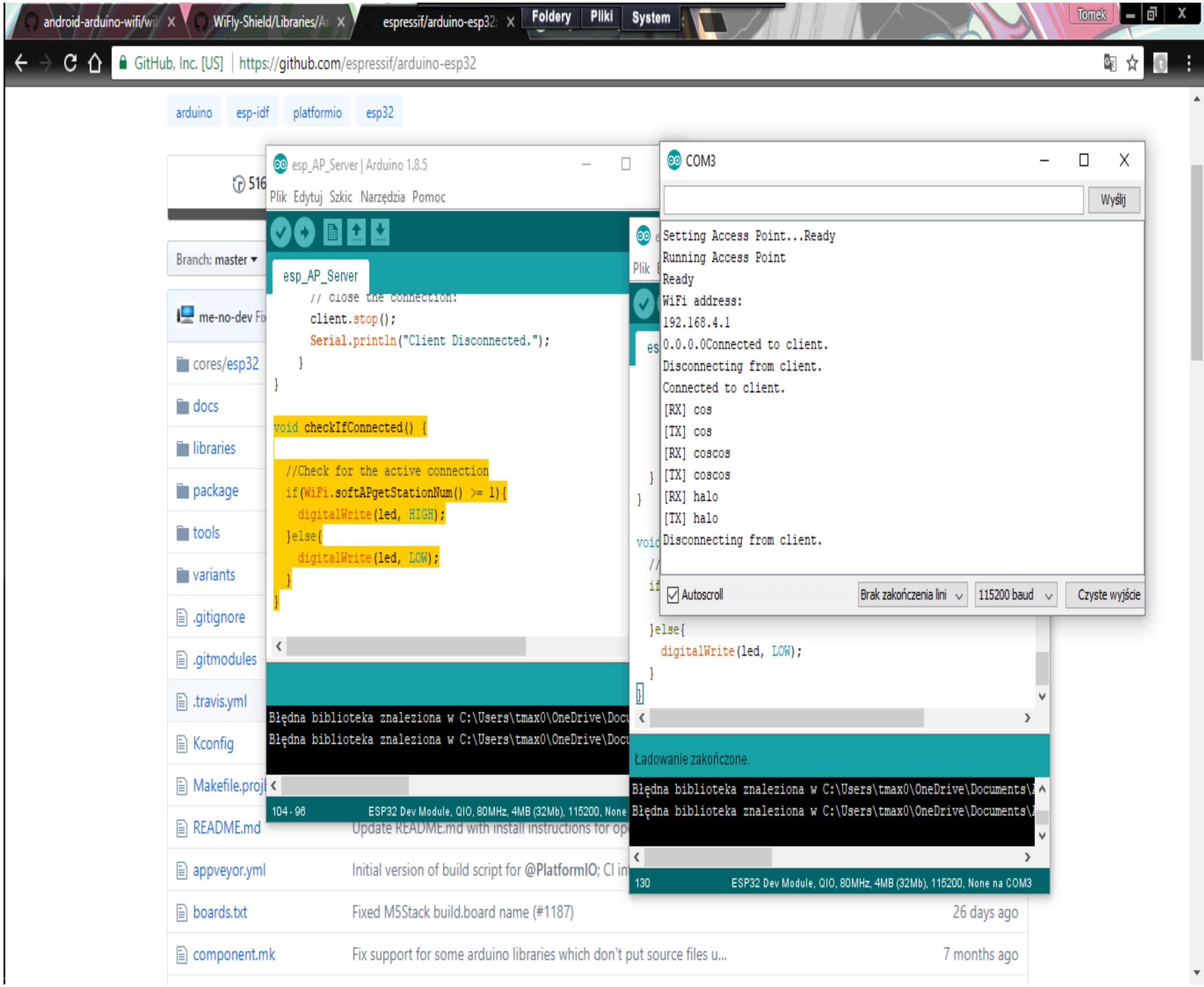
Every action that is passed through serial ports is a simple text command.

As a first step we had to check our connection and return results.

A good way to do that was to write a simple ping interval.



```
09:18:24.615 Connecting to ESP32  (192.168.4.1:80) ...
09:18:24.633 Connected
09:18:24.635 SOCKET_CONNECTED
09:18:25.701 SOCKET_PING
09:18:26.638 SOCKET_PING
09:18:27.644 SOCKET_PING
09:18:28.639 SOCKET_PING
09:18:29.640 SOCKET_PING
09:18:30.640 SOCKET_PING
09:18:31.747 SOCKET_PING
```



```
    // close the connection:
    client.stop();
    Serial.println("Client Disconnected.");
  }
}

void checkIfConnected() {

  //Check for the active connection
  if(WiFi.softAPgetStationNum() >= 1){
    digitalWrite(led, HIGH);
  }else{
    digitalWrite(led, LOW);
  }
}
```

```
Błędna biblioteka znaleziona w C:\Users\tmax0\OneDrive\Docu
Błędna biblioteka znaleziona w C:\Users\tmax0\OneDrive\Docu
```

104 - 96 ESP32 Dev Module, QIO, 80MHz, 4MB (32Mb), 115200, None

```
Setting Access Point...Ready
Running Access Point
Ready
WiFi address:
192.168.4.1
0.0.0.0Connected to client.
Disconnecting from client.
Connected to client.
[RX] cos
[TX] cos
[RX] coscos
[TX] coscos
[RX] halo
[TX] halo
Disconnecting from client.
```

```
  }else{
    digitalWrite(led, LOW);
  }
}
```

Ładowanie zakończone.

```
Błędna biblioteka znaleziona w C:\Users\tmax0\OneDrive\Documents\
Błędna biblioteka znaleziona w C:\Users\tmax0\OneDrive\Documents\
```

130 ESP32 Dev Module, QIO, 80MHz, 4MB (32Mb), 115200, None na COM3

me-no-dev Fi		
cores/esp32		
docs		
libraries		
package		
tools		
variants		
.gitignore		
.gitmodules		
.travis.yml		
Kconfig		
Makefile.proj		
README.md	Update README.md with install instructions for op	
appveyor.yml	Initial version of build script for @PlatformIO; CI in	
boards.txt	Fixed M5Stack build.board name (#1187)	26 days ago
component.mk	Fix support for some arduino libraries which don't put source files u...	7 months ago

Implementation in a drone app

We had to rewrite all the interfaces to the app’s side code.

In the first approach aplication has automatically found esp server and connected to it.

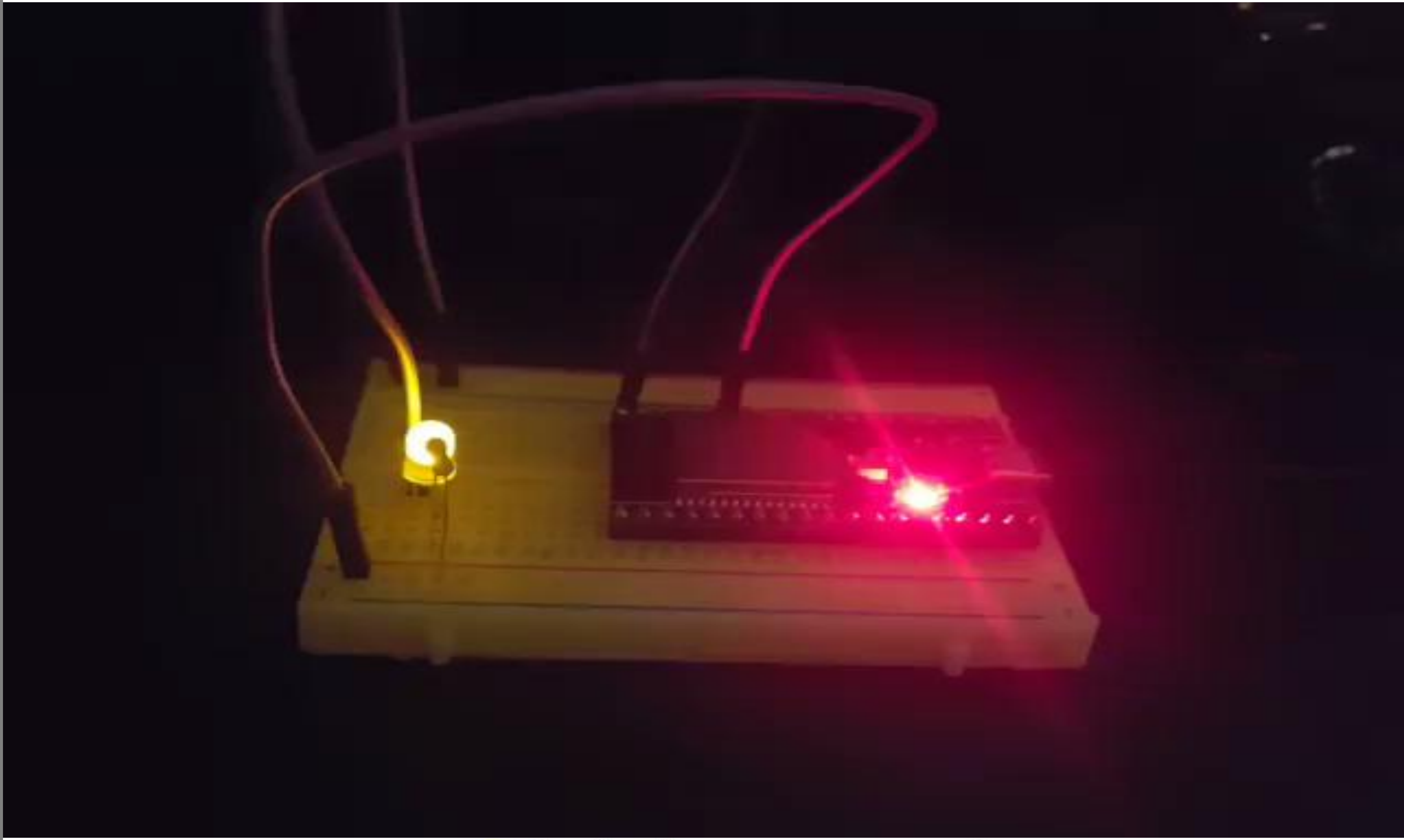
Drone
Home
Maps
Control
Checklist
Extras
Settings
Open Source
Contact
About
Countries
Croatia
Poland

Testing a PWM ports

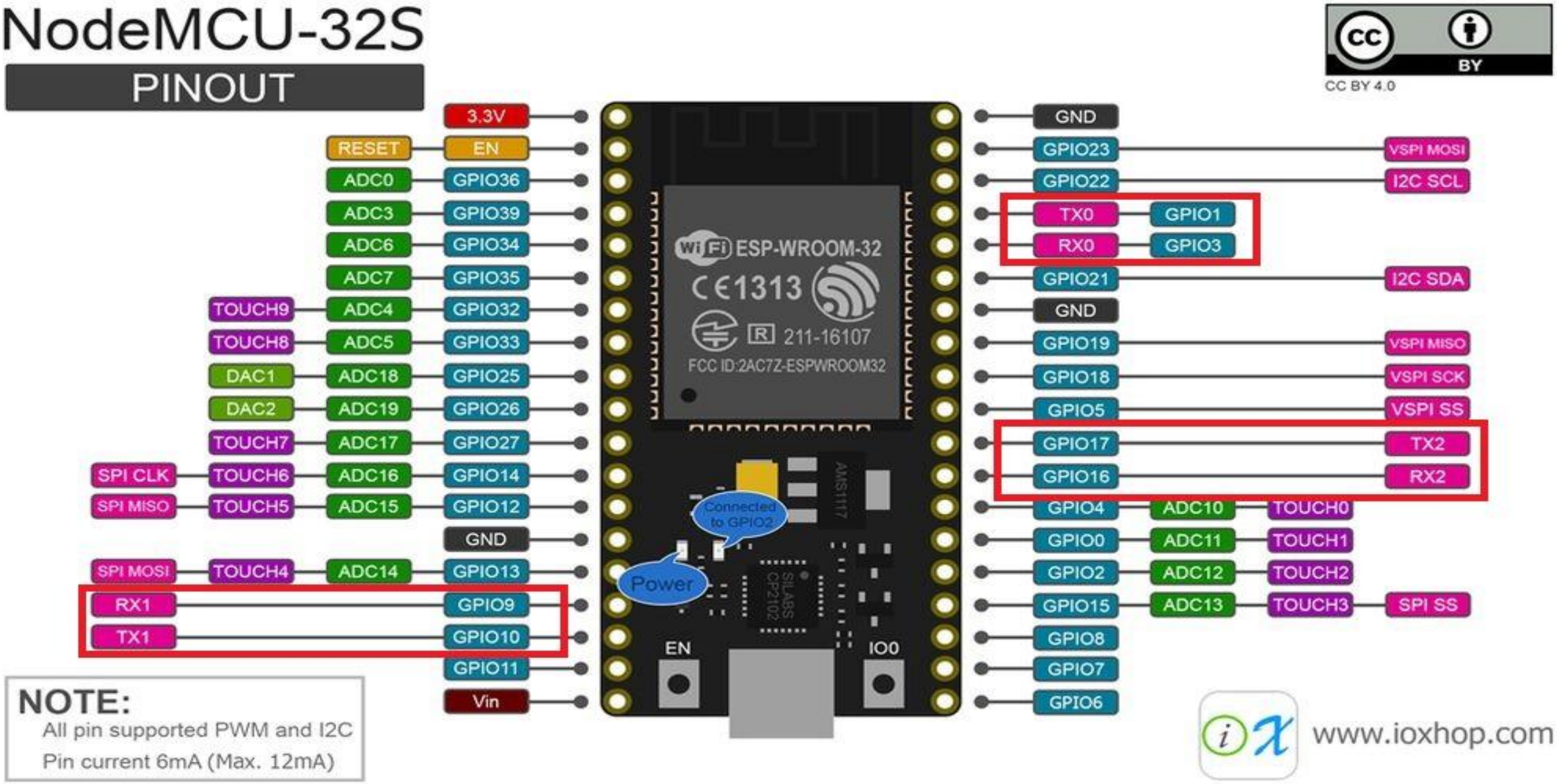
PWM (Pulse Width Modulation) allows to move out modulated signal of value between 0-255.

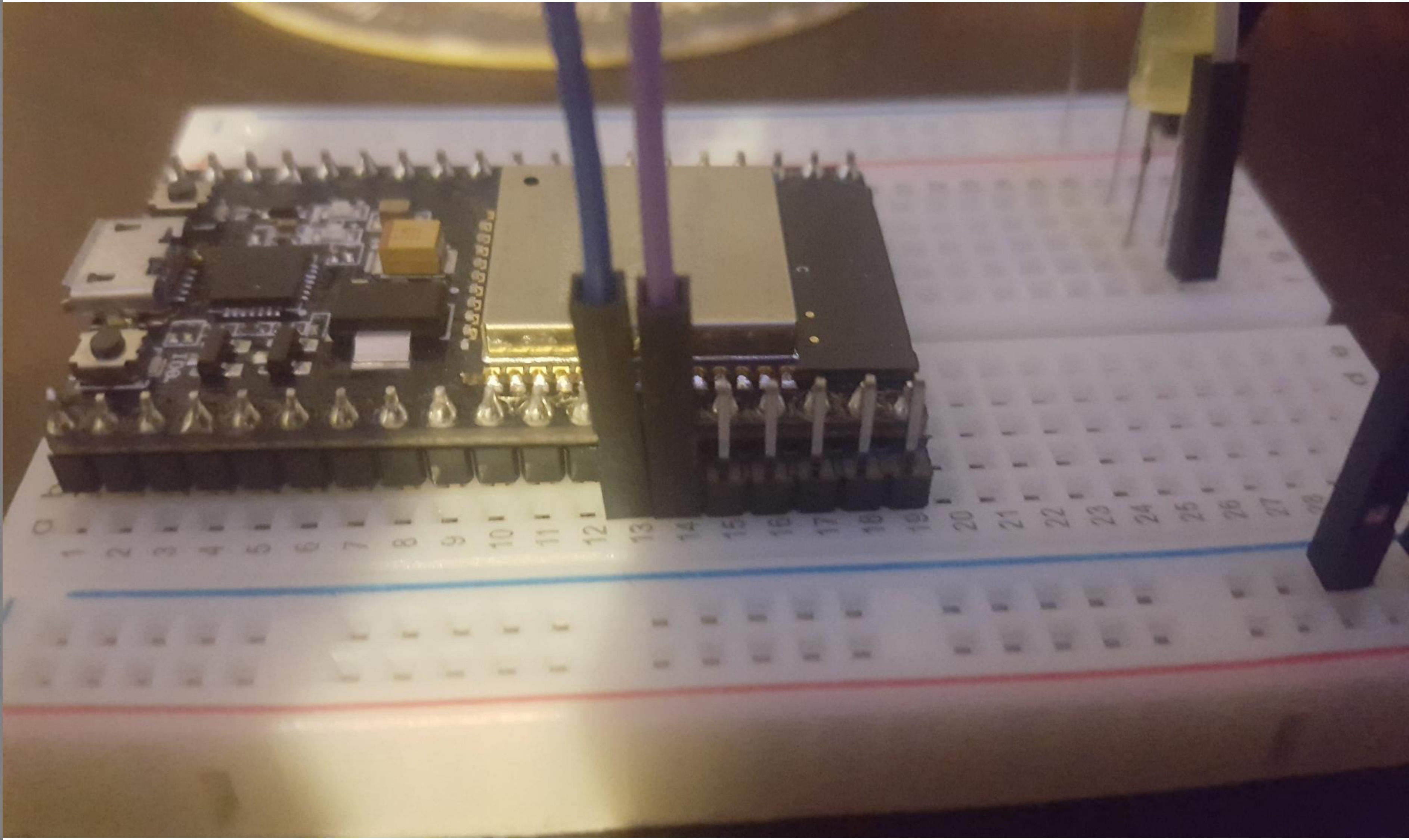
Good way to learn how the PWM works, was to make a common LED example.

PWM modulation is widely used in motor regulation.



Locate RX and TX pins




1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28

Different GPIO's

GPIO1, GPIO3 support serial connection but block other serial ports like usb so it's quite problematic using them.

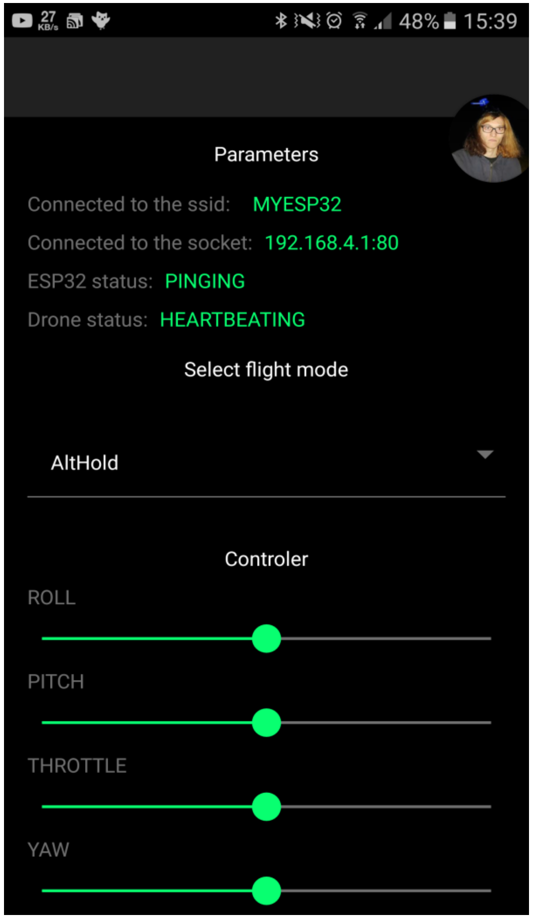
In a next step we decided to change ports to the GPIO16, GPIO17.

Esp32 serial tricks

HardwareSerial supports baudrate from 9600 to 115200, where 115200 is the fault value.

It is possible to run synchronized 57600 **hardwareSerial** for apm telemetry and a Serial 115200 for a app commands.

Creating an GUI in app



27 KB/s 48% 15:39

Parameters

Connected to the ssid: MYESP32

Connected to the socket: 192.168.4.1:80

ESP32 status: PINGING

Drone status: HEARTBEATING

Select flight mode

AltHold

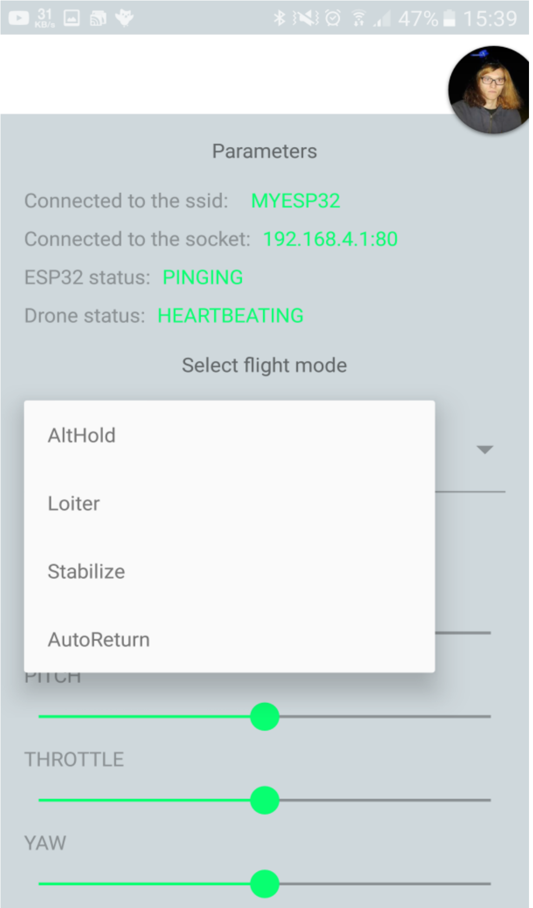
Controler

ROLL

PITCH

THROTTLE

YAW



31 KB/s 47% 15:39

Parameters

Connected to the ssid: MYESP32

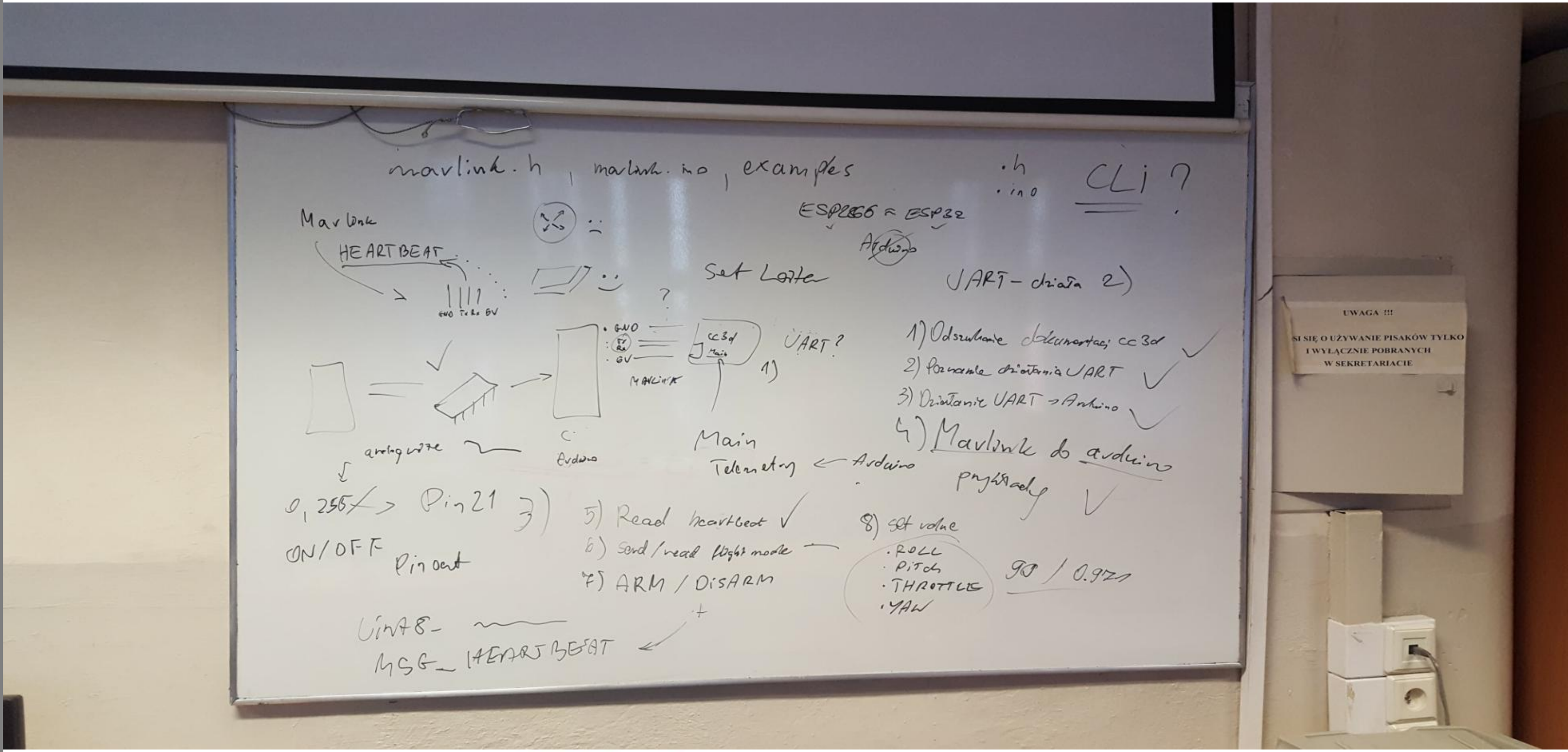
Connected to the socket: 192.168.4.1:80

ESP32 status: PINGING

Drone status: HEARTBEATING

Select flight mode

Discussing a proper development



1) Odszukanie dokumentacji cc3d ✓
2) Poznanie działania UART ✓
3) Działanie UART -> Arduino ✓
4) Mavlink do arduino przykłady ✓
5) Read heartbeat ✓
6) Send / read flight mode
7) ARM / DISARM
8) Set value
- ROLL
- PITCH
- THROTTLE
- YAW

Looking for a good way

Ola and Paweł have spent many, many hours looking for a mavlink documentation and Arduino code.

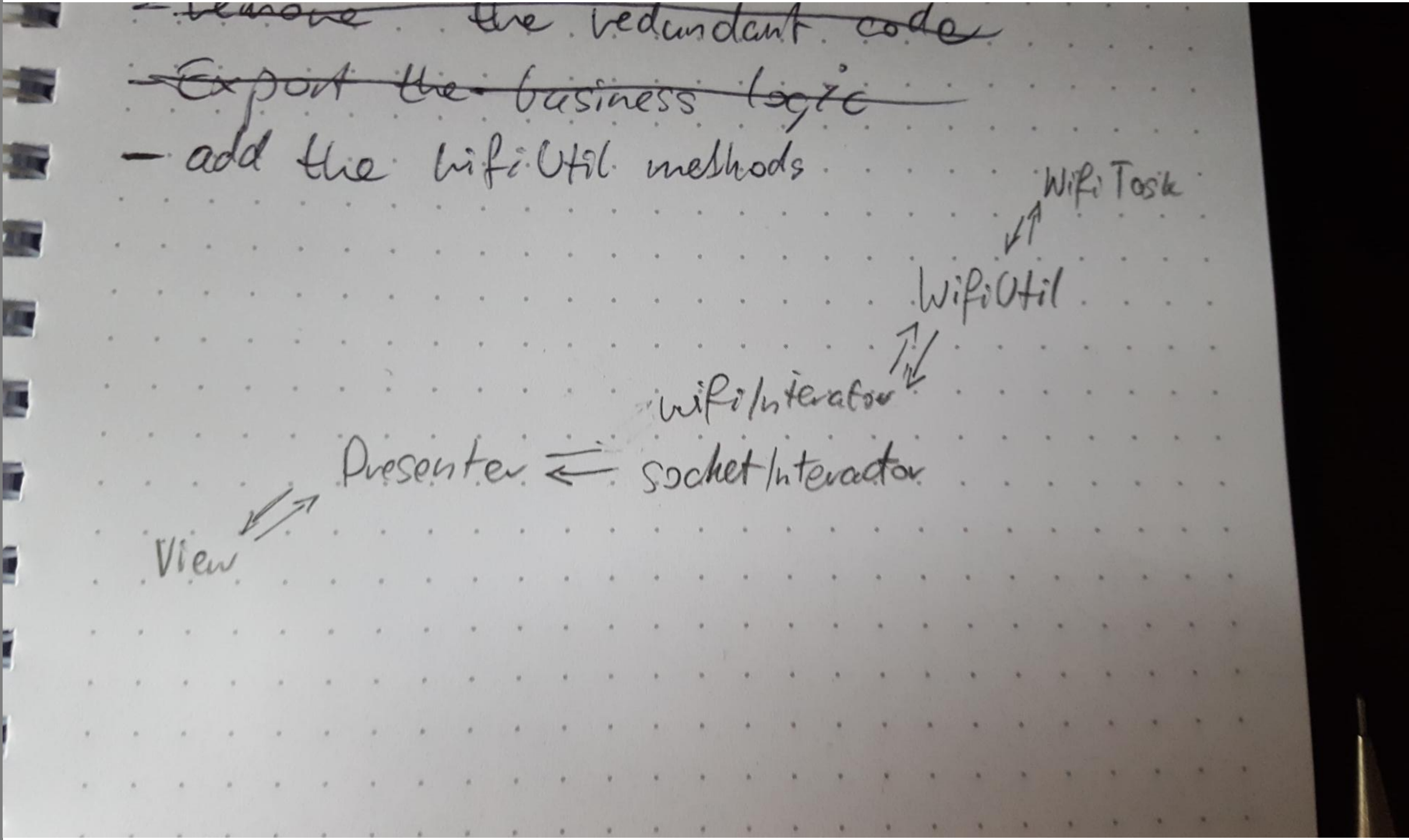
These sources are crucial in a code development because mavlink is well documented but for experts.

Changing app architecture

I spent a lot of time changing the previous code layout on the M (model) V (view) P (presenter) approuch.

The big amount of code and high complication forced me to separated code sections.

Thus now app is more flexible and easier to testing.



- ~~remove the redundant code~~
- ~~Export the business logic~~
- add the WifiUtil methods

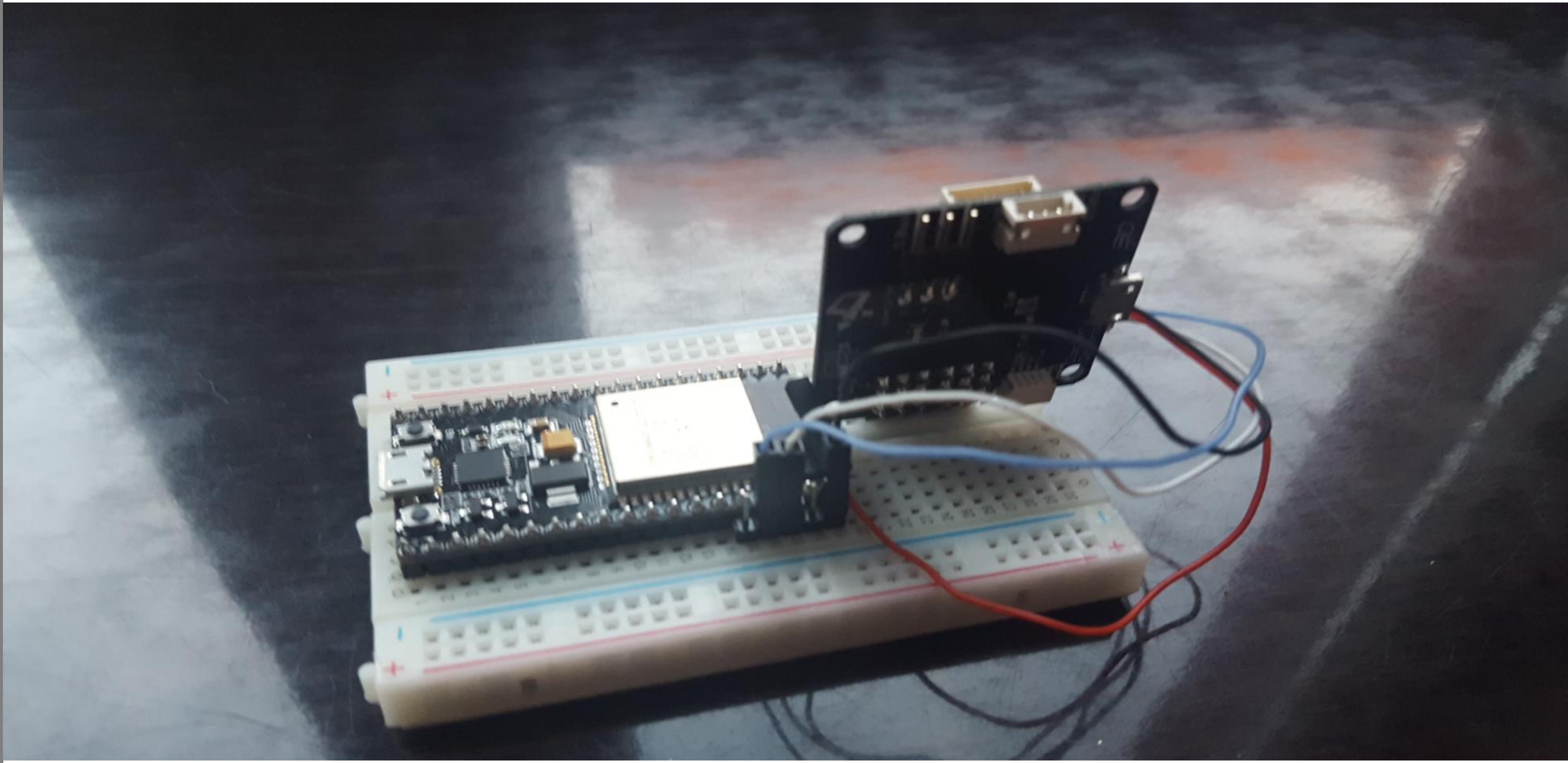
WifiTask
WifiUtil
wifiInteractor
Presenter
socketInteractor
View

CC3D

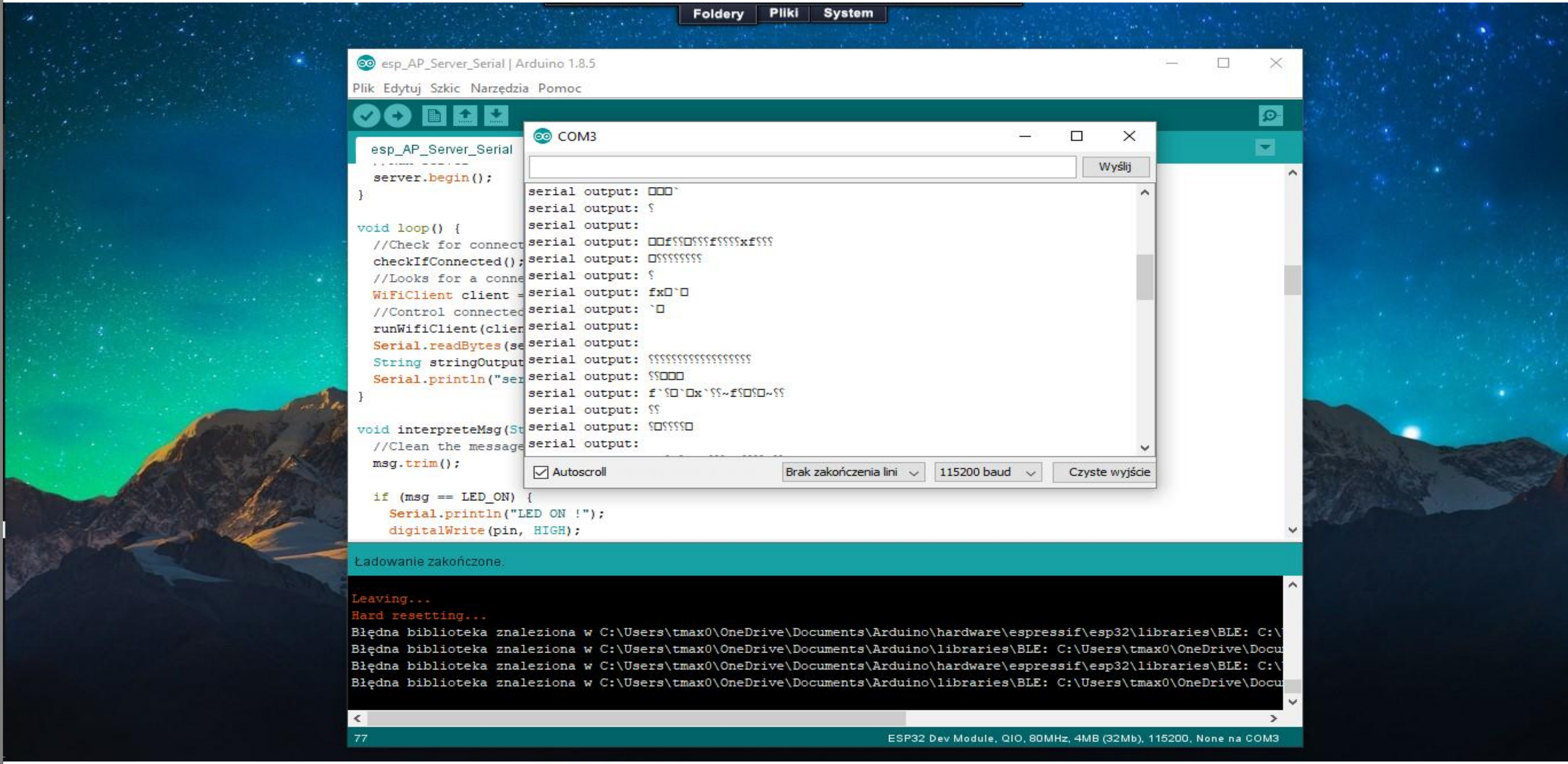
In our class we had a free cc3d board so we decided to start work with it.

The next slides will show you first steps with drone telemetry and mavlink.

Connection



Initiating serial outputs


Foldery
Pliki
System
esp_AP_Server_Serial | Arduino 1.8.5
Plik Edytuj Szkic Narzędzia Pomoc
esp_AP_Server_Serial
COM3
Wyślij

```
  server.begin();
}

void loop() {
  //Check for connect
  checkIfConnected();
  //Looks for a conne
  WiFiClient client =
  //Control connected
  runWifiClient(clien
  Serial.readBytes(se
  String stringOutput
  Serial.println("ser
}

void interpreteMsg(St
  //Clean the message
  msg.trim();

  if (msg == LED_ON) {
    Serial.println("LED ON !");
    digitalWrite(pin, HIGH);
```

```
serial output: □□□`
serial output: ⸮
serial output:
serial output: □□f⸮⸮□⸮⸮⸮f⸮⸮⸮⸮xf⸮⸮⸮
serial output: □⸮⸮⸮⸮⸮⸮⸮⸮
serial output: ⸮
serial output: fx□`□
serial output: `□
serial output:
serial output:
serial output: ⸮⸮⸮⸮⸮⸮⸮⸮⸮⸮⸮⸮⸮⸮⸮⸮⸮⸮⸮
serial output: ⸮⸮□□□
serial output: f`⸮□`□x`⸮⸮~f⸮□⸮□~⸮⸮
serial output: ⸮⸮
serial output: ⸮□⸮⸮⸮⸮□
serial output:
```

☑ Autoscroll   Brak zakończenia lini   115200 baud   Czyste wyjście

Ładowanie zakończone.

```
Leaving...
Hard resetting...
Błędna biblioteka znaleziona w C:\Users\tmax0\OneDrive\Documents\Arduino\hardware\espressif\esp32\libraries\BLE: C:\
Błędna biblioteka znaleziona w C:\Users\tmax0\OneDrive\Documents\Arduino\libraries\BLE: C:\Users\tmax0\OneDrive\Docu
Błędna biblioteka znaleziona w C:\Users\tmax0\OneDrive\Documents\Arduino\hardware\espressif\esp32\libraries\BLE: C:\
Błędna biblioteka znaleziona w C:\Users\tmax0\OneDrive\Documents\Arduino\libraries\BLE: C:\Users\tmax0\OneDrive\Docu
```

77   ESP32 Dev Module, QIO, 80MHz, 4MB (32Mb), 115200, None na COM3

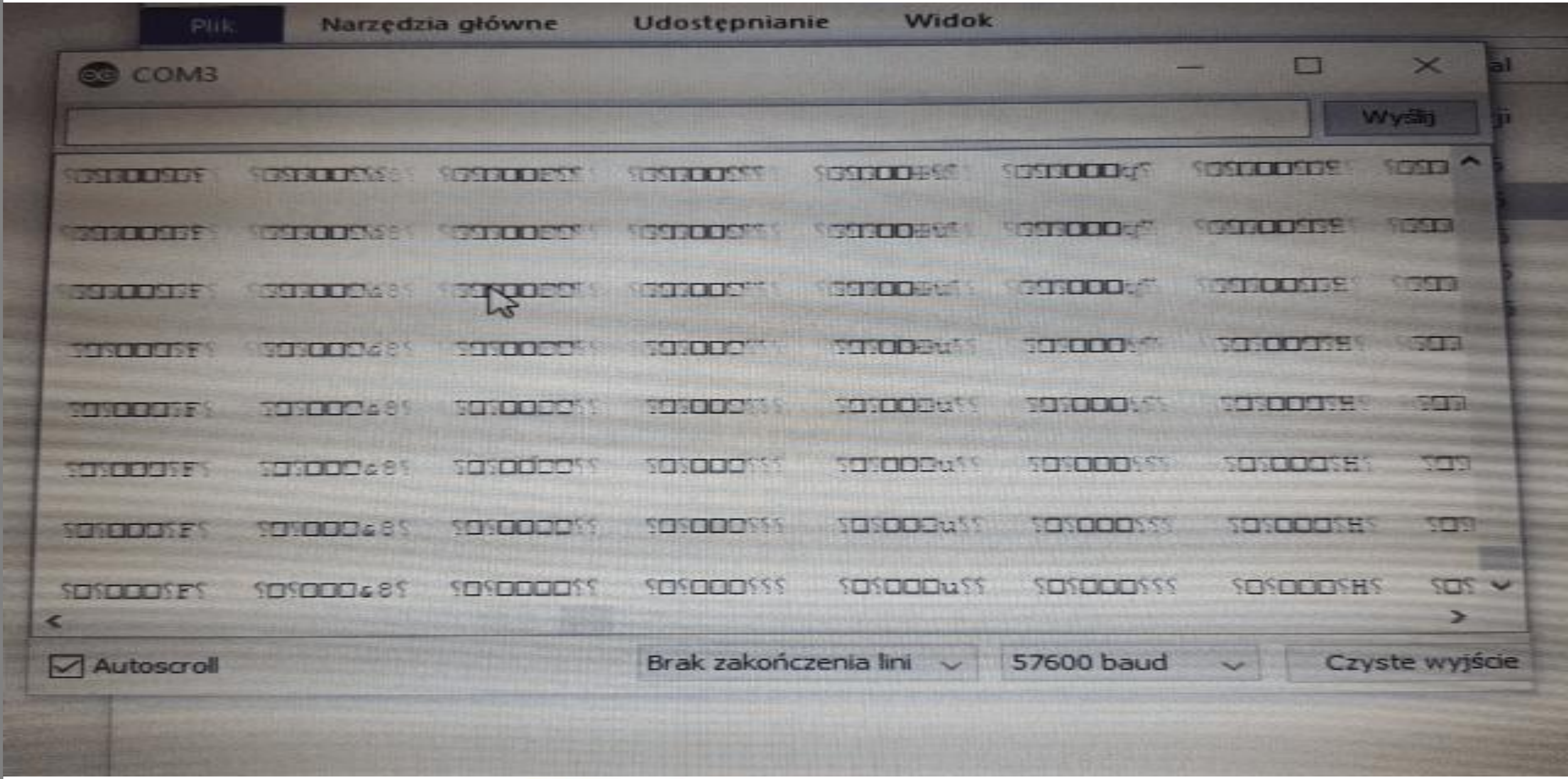
Problems with CC3D

After many, many tries and receiving raw bit frames.

It turned out that cc3d doesen’t use mavlink interface and requires software serial.

These obstacles stopped our work. This was something that we couldn’t bypass.

Raw bitframes

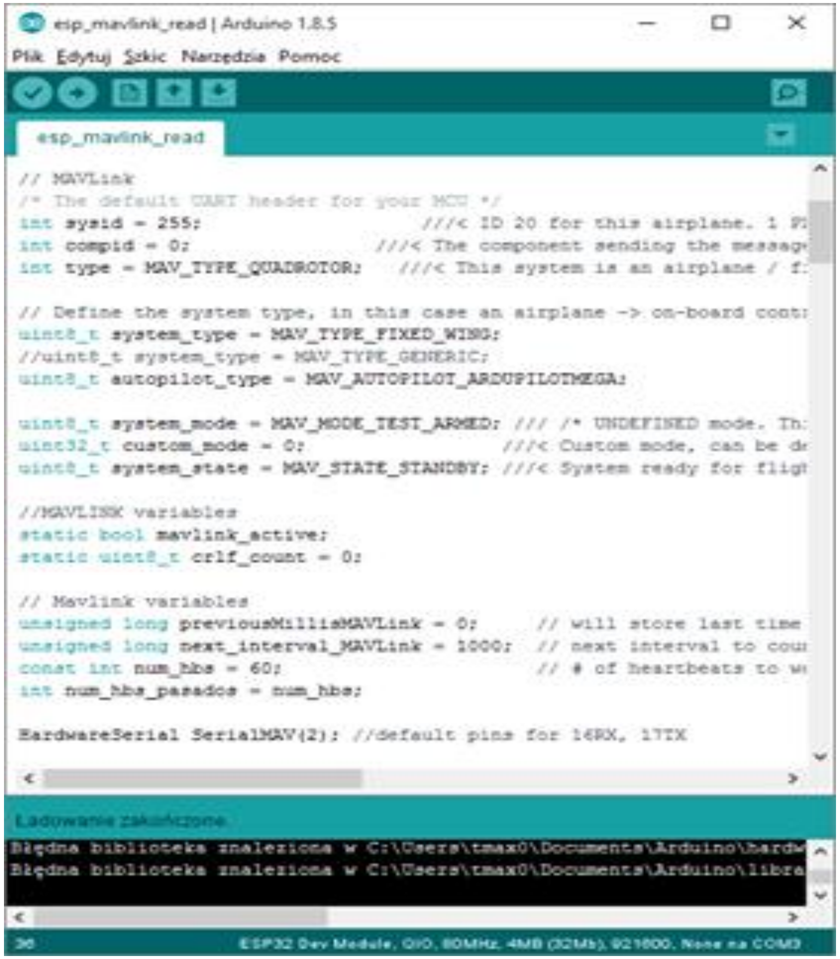


☑ Autoscroll

Brak zakończenia lini

57600 baud

Mavlink code

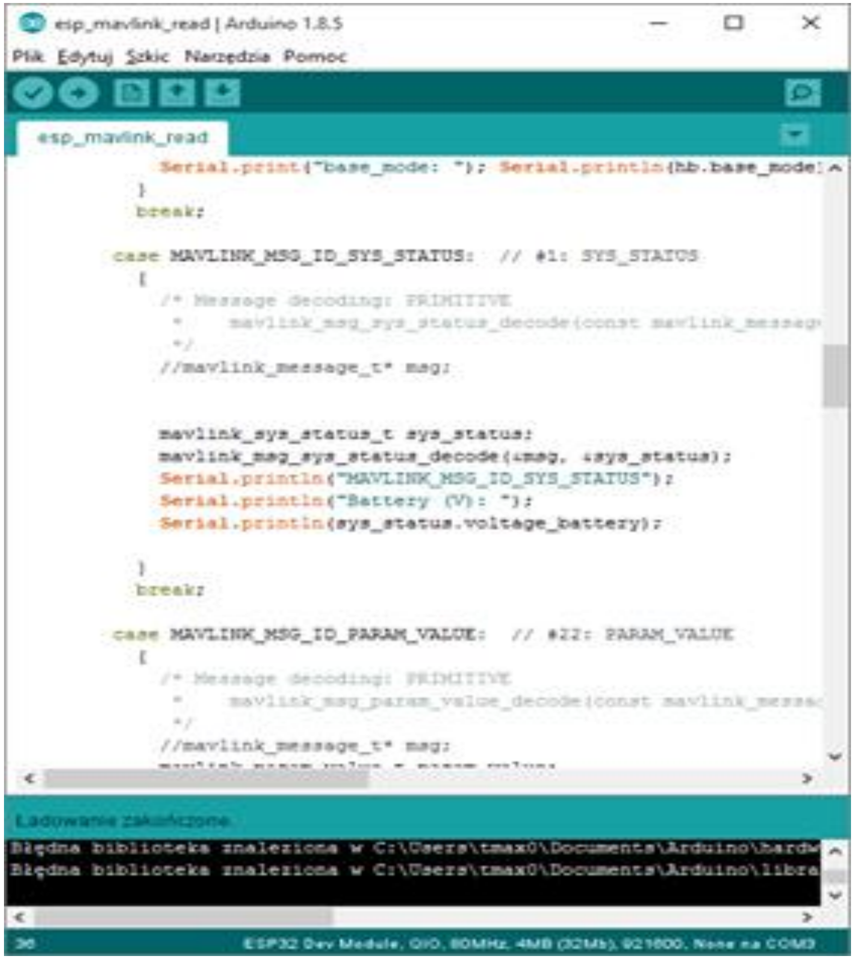

esp_mavlink_read | Arduino 1.8.5
Plik Edytuj Szkic Narzędzia Pomoc
esp_mavlink_read

```
// MAVLink
/* The default UART header for your MCU */
int sysid = 255;                    ///< ID 20 for this airplane. 1 P
int compid = 0;                 ///< The component sending the messag
int type = MAV_TYPE_QUADROTOR;    ///< This system is an airplane / f

// Define the system type, in this case an airplane -> on-board cont
uint8_t system_type = MAV_TYPE_FIXED_WING;
//uint8_t system_type = MAV_TYPE_GENERIC;
uint8_t autopilot_type = MAV_AUTOPILOT_ARDUPILOTMEGA;

uint8_t system_mode = MAV_MODE_TEST_ARMED; /// /* UNDEFINED mode. Th
uint32_t custom_mode = 0;                 ///< Custom mode, can be de
uint8_t system_state = MAV_STATE_STANDBY; ///< System ready for fligh

//MAVLINK variables
static bool mavlink_active;
static uint8_t crlf_count = 0;

// Mavlink variables
unsigned long previousMillisMAVLink = 0;     // will store last time
unsigned long next_interval_MAVLink = 1000;  // next interval to cou
const int num_hbs = 60;                      // # of heartbeats to wa
int num_hbs_pasados = num_hbs;

HardwareSerial SerialMAV(2); //default pins for 16RX, 17TX
```

Ladowanie zakończone.

```
Błędna biblioteka znaleziona w C:\Users\tmax0\Documents\Arduino\hardw
Błędna biblioteka znaleziona w C:\Users\tmax0\Documents\Arduino\libra
```

36 ESP32 Dev Module, QIO, 80MHz, 4MB (32Mb), 921600, None na COM3


esp_mavlink_read | Arduino 1.8.5
Plik Edytuj Szkic Narzędzia Pomoc
esp_mavlink_read

```
        Serial.print("base_mode: "); Serial.println(hb.base_mode
      }
      break;

    case MAVLINK_MSG_ID_SYS_STATUS:  // #1: SYS_STATUS
      {
        /* Message decoding: PRIMITIVE
         *    mavlink_msg_sys_status_decode(const mavlink_messag
         */
        //mavlink_message_t* msg;


        mavlink_sys_status_t sys_status;
        mavlink_msg_sys_status_decode(&msg, &sys_status);
        Serial.println("MAVLINK_MSG_ID_SYS_STATUS");
        Serial.println("Battery (V): ");
        Serial.println(sys_status.voltage_battery);

      }
      break;

    case MAVLINK_MSG_ID_PARAM_VALUE:  // #22: PARAM_VALUE
      {
        /* Message decoding: PRIMITIVE
         *    mavlink_msg_param_value_decode(const mavlink_messa
         */
        //mavlink_message_t* msg;
        mavlink_param_value_t param_value;
```

Ladowanie zakończone.

```
Błędna biblioteka znaleziona w C:\Users\tmax0\Documents\Arduino\hardw
Błędna biblioteka znaleziona w C:\Users\tmax0\Documents\Arduino\libra
```

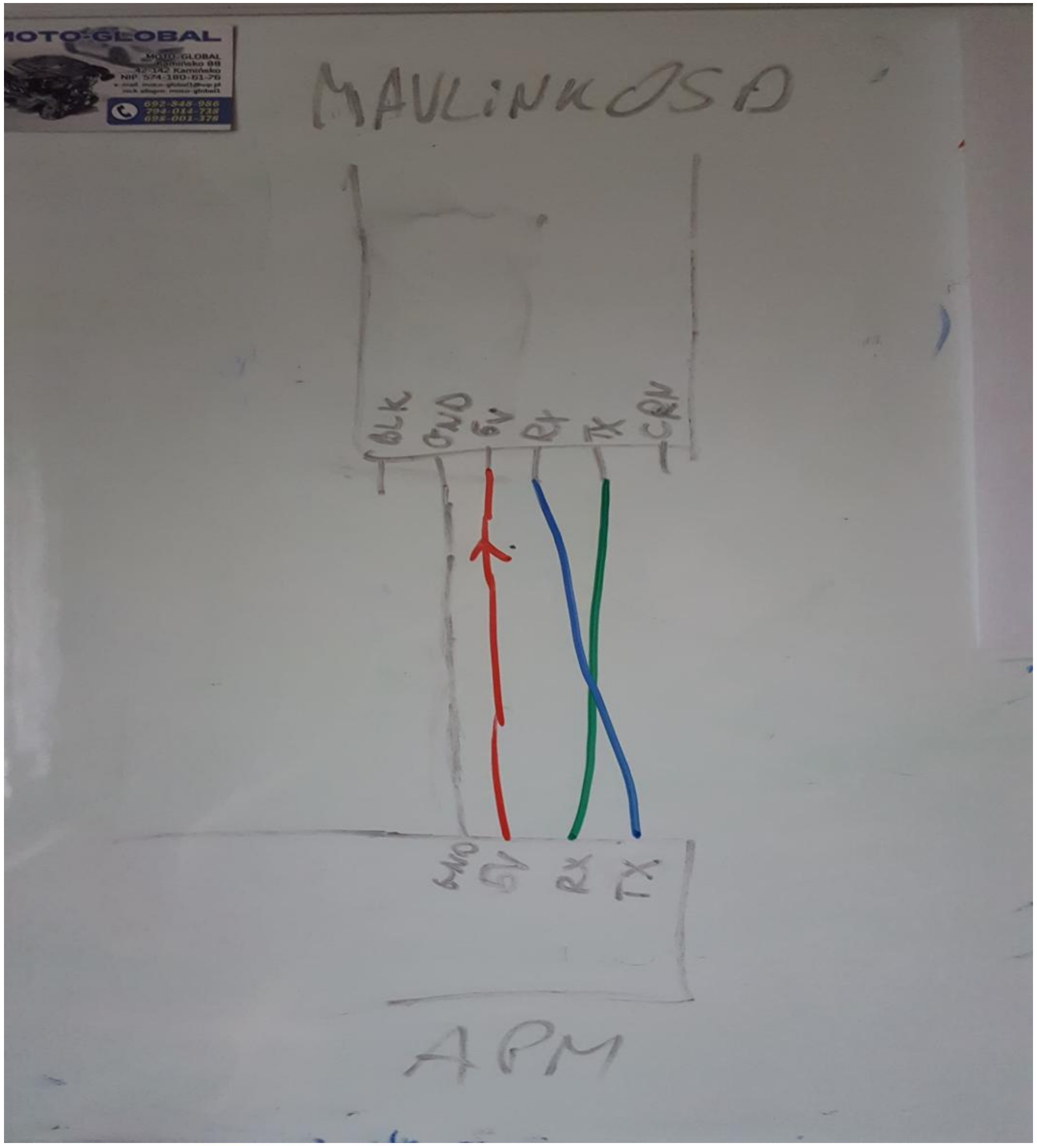
36 ESP32 Dev Module, QIO, 80MHz, 4MB (32Mb), 921600, None na COM3

Exchange from CC3D to APM

Dissasembly propellers

For safety reasons I had to dissasembly the propellers in case of arming.

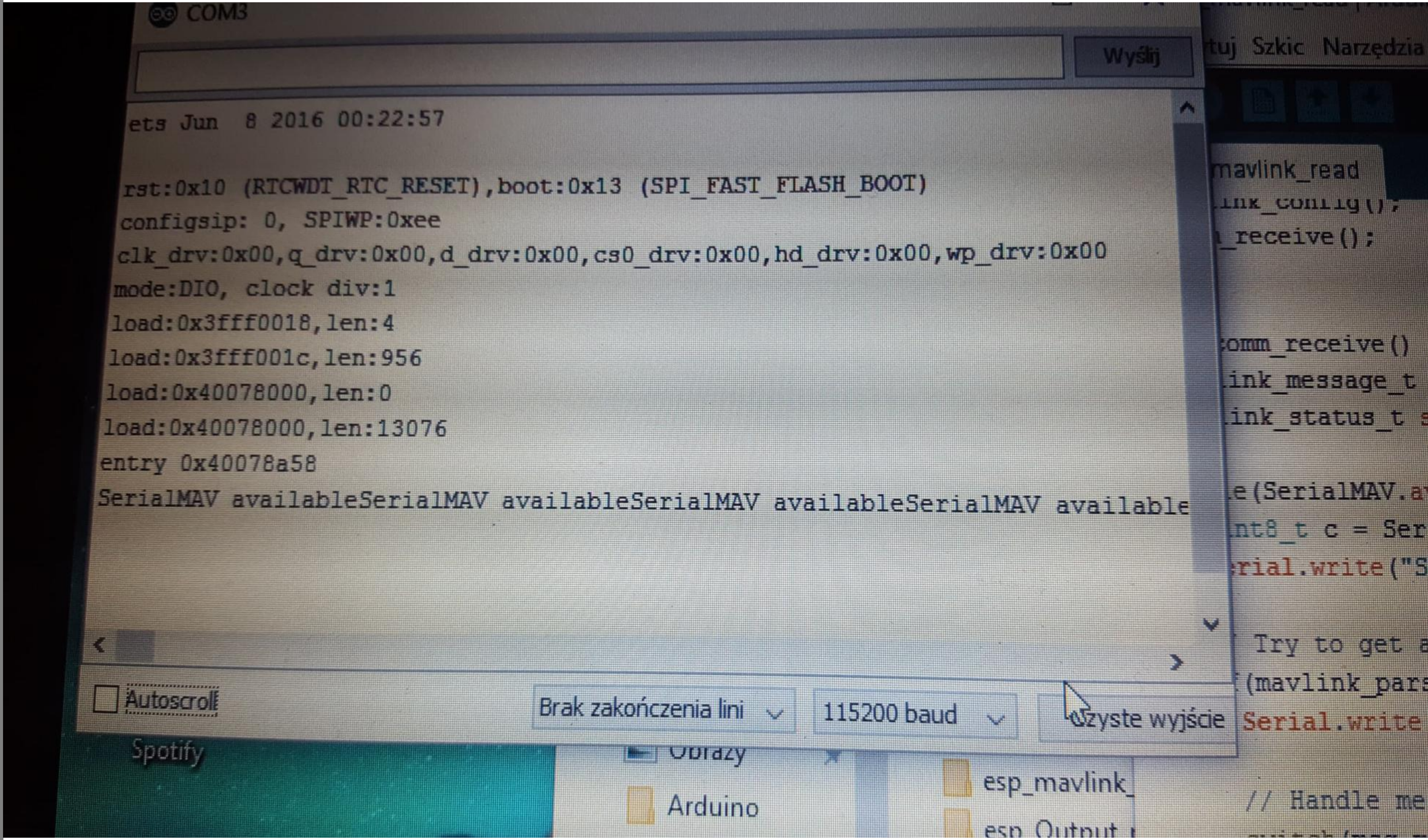
Another things to do was reconnect telemetry cables from OSD (On Screen Display) to esp32.


MAVLINKOSD
BLK
GND
5V
RX
TX
GRN
GND
5V
RX
TX
APM

Mavlink and APM

With apm the things had gone match better. Apm supports **hardwareSerial** and mavlink commands.

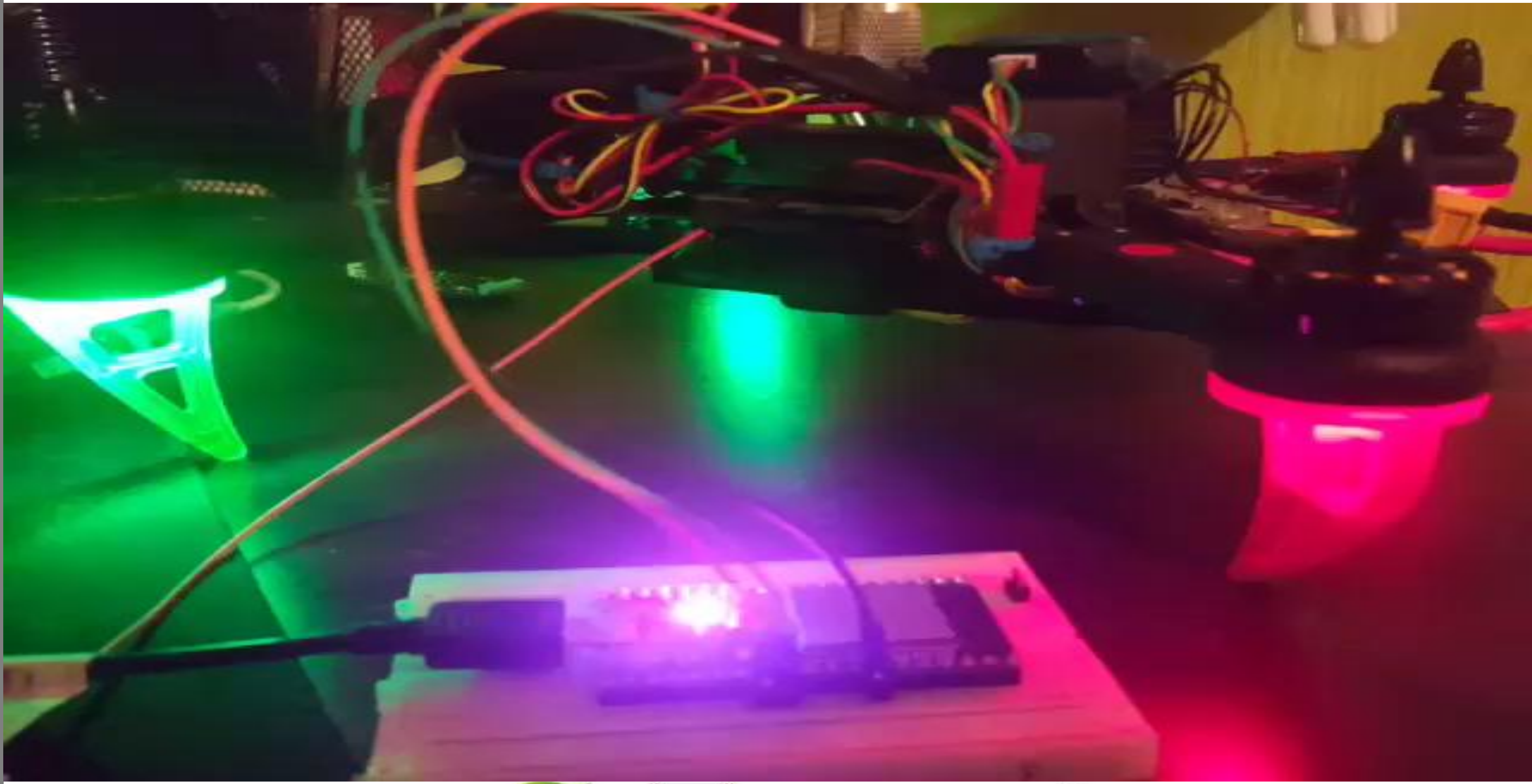
We were able to use already written mavlink code and check bits transfer.



```
ets Jun  8 2016 00:22:57

rst:0x10 (RTCWDT_RTC_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:956
load:0x40078000,len:0
load:0x40078000,len:13076
entry 0x40078a58
SerialMAV availableSerialMAV availableSerialMAV availableSerialMAV available
```

```
ink_config();
_receive();

omm_receive()
ink_message_t
ink_status_t s
e(SerialMAV.a
nt8_t c = Ser
rial.write("S

 Try to get a
(mavlink_pars
Serial.write

// Handle me
```

Indicating serial read

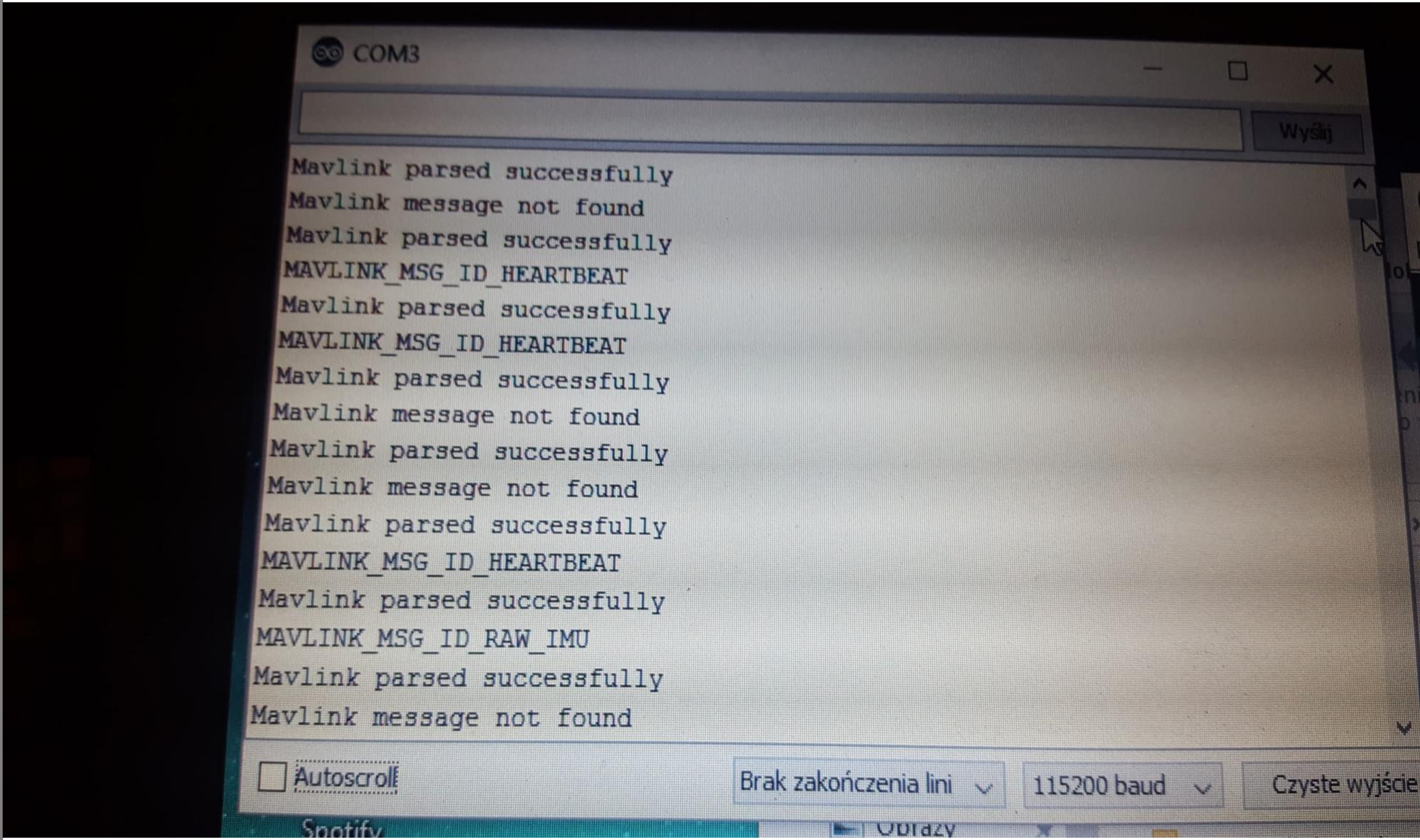


Mavlink explanation

Drones are using heartbeat to confirm a connection.

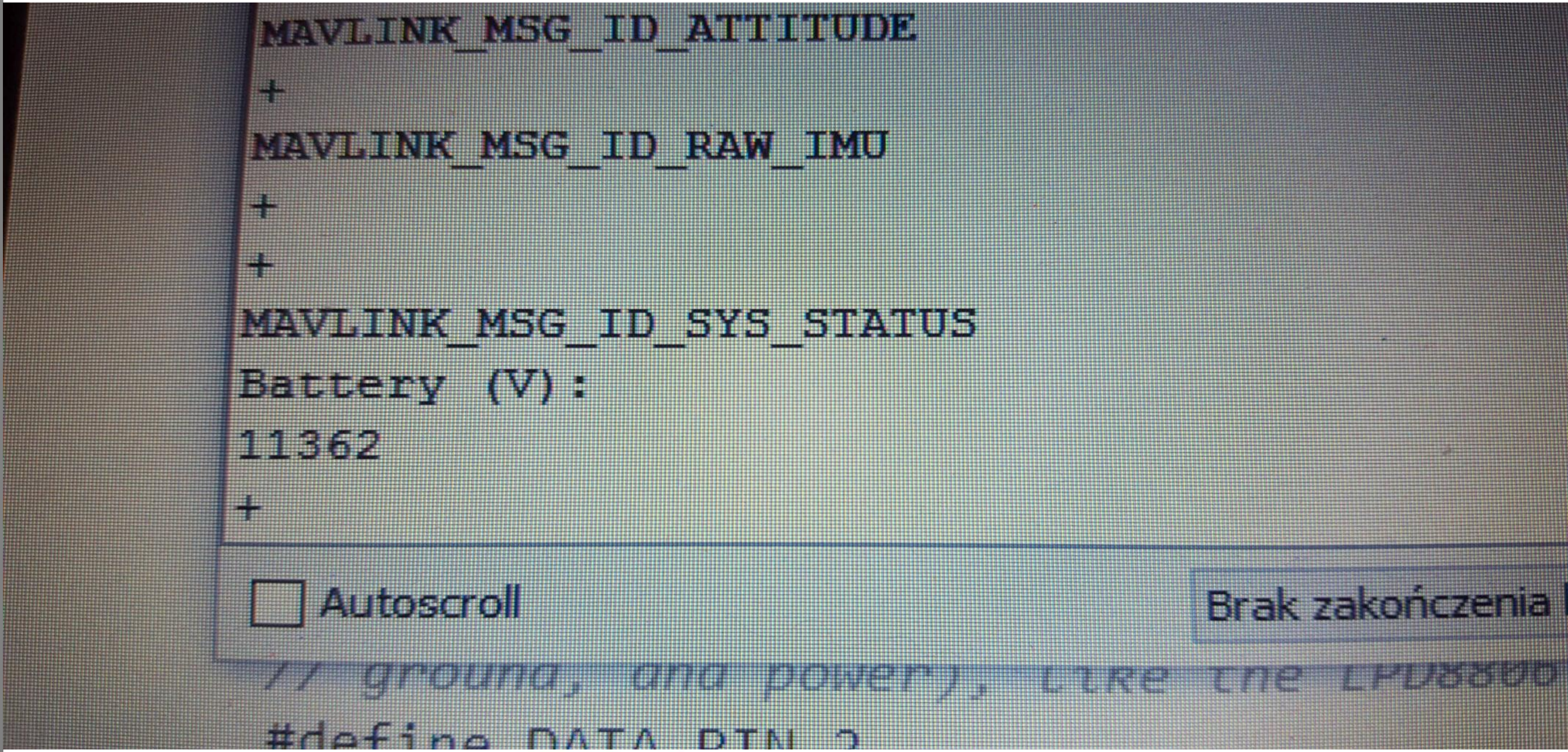
Receiving heartbeat packets is first and most important thing.

After that we can decode other drone’s parameters.



```
Mavlink parsed successfully
Mavlink message not found
Mavlink parsed successfully
MAVLINK_MSG_ID_HEARTBEAT
Mavlink parsed successfully
MAVLINK_MSG_ID_HEARTBEAT
Mavlink parsed successfully
Mavlink message not found
Mavlink parsed successfully
Mavlink message not found
Mavlink parsed successfully
MAVLINK_MSG_ID_HEARTBEAT
Mavlink parsed successfully
MAVLINK_MSG_ID_RAW_IMU
Mavlink parsed successfully
Mavlink message not found
```

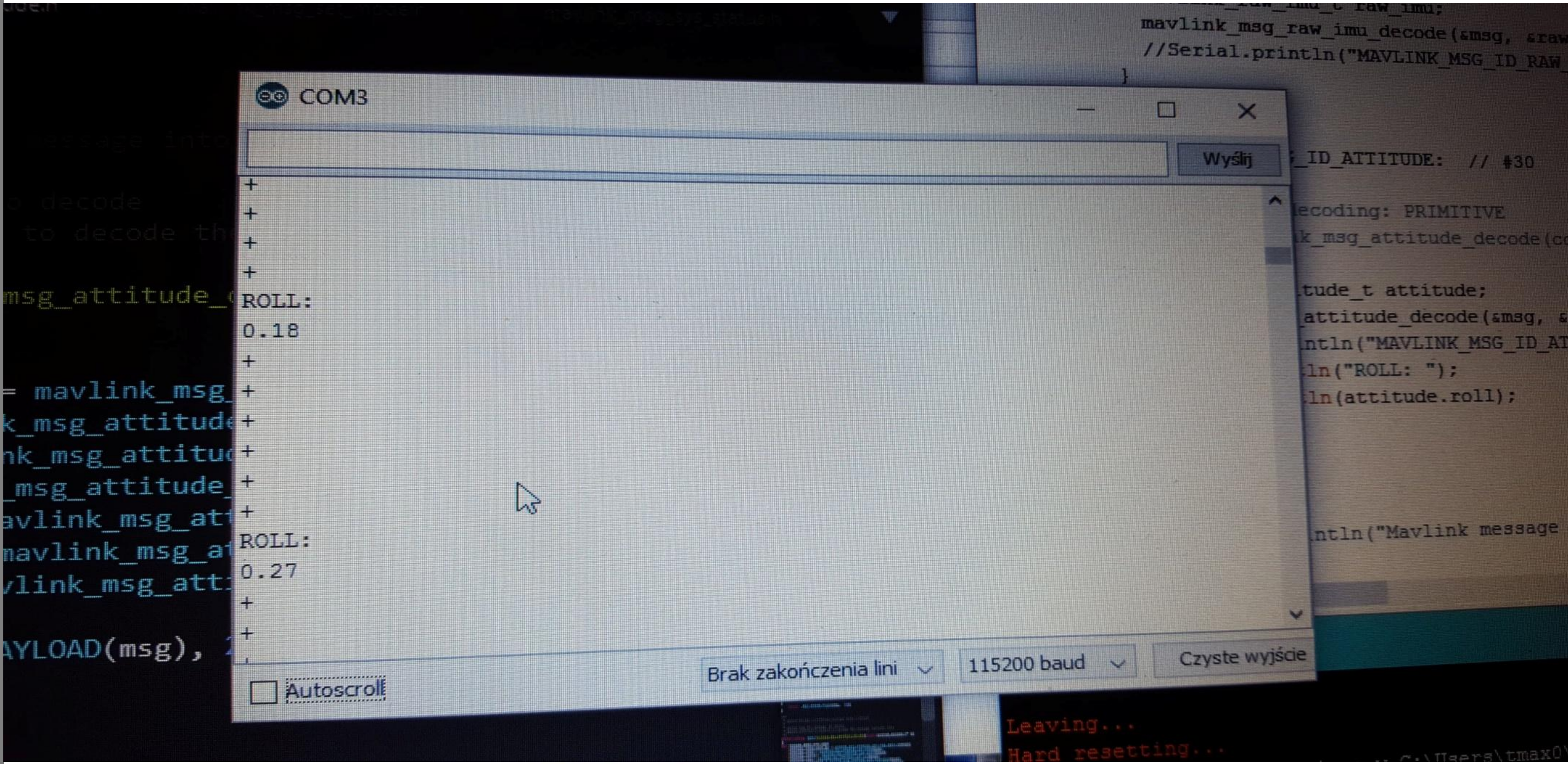
Battery status



```
MAVLINK_MSG_ID_ATTITUDE
+
MAVLINK_MSG_ID_RAW_IMU
+
+
MAVLINK_MSG_ID_SYS_STATUS
Battery (V):
11362
+
```

```
// ground, and power), tie the LPD8806
#define DATA_PIN 2
```

Position values



```
+
+
+
+
ROLL:
0.18
+
+
+
+
+
+
+
ROLL:
0.27
+
+
```

```
mavlink_msg_raw_imu_decode(&msg, &raw
//Serial.println("MAVLINK_MSG_ID_RAW_
}
_ID_ATTITUDE:  // #30
ecoding: PRIMITIVE
k_msg_attitude_decode(c
tude_t attitude;
attitude_decode(&msg, &
ntln("MAVLINK_MSG_ID_AT
ln("ROLL: ");
ln(attitude.roll);
ntln("Mavlink message
```

```
Leaving...
Hard resetting...
```

Sending mavlink requests

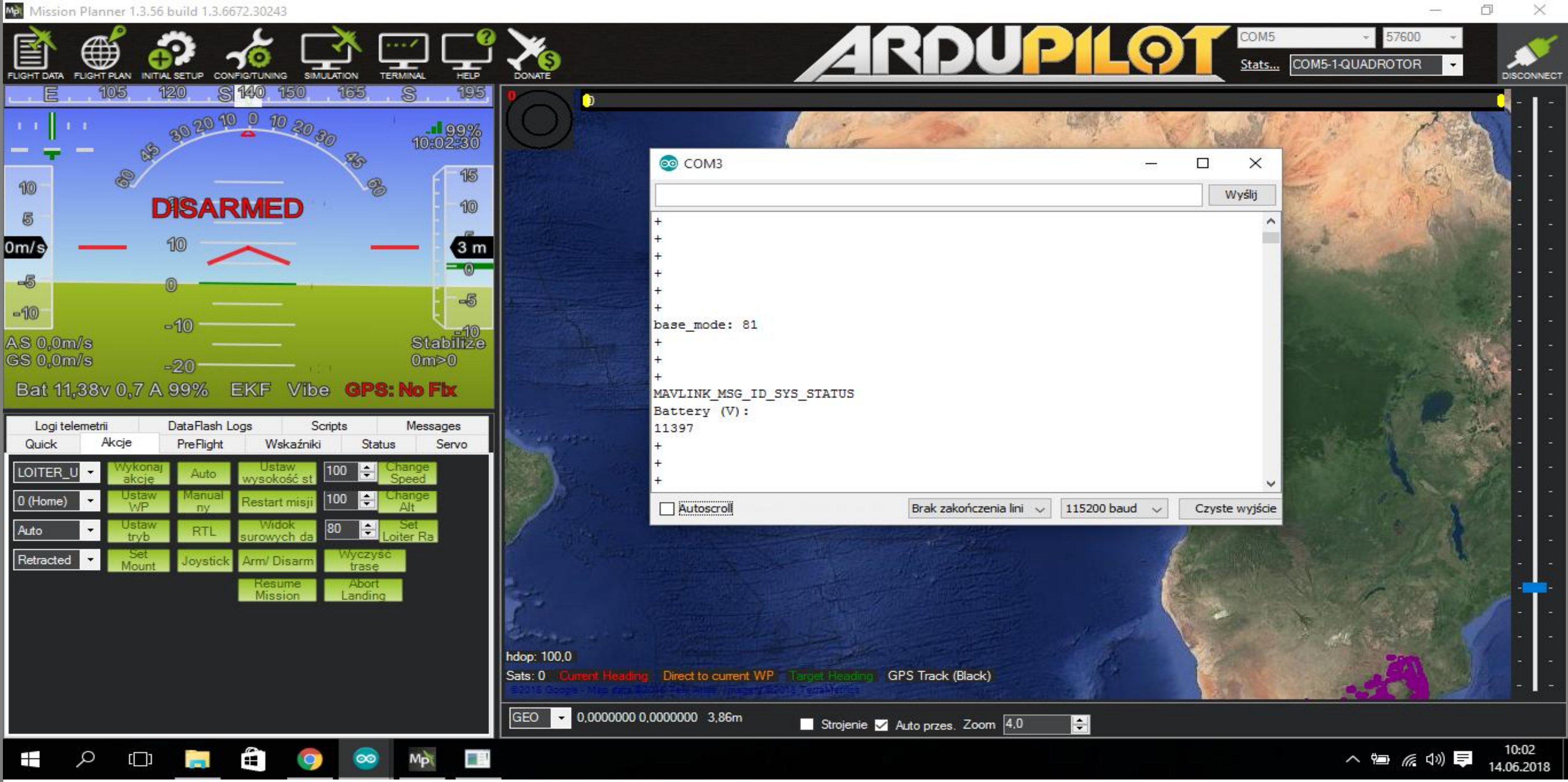
After we have learned how to receive mavlink's data we are able to send our commands.

As always the initial thing is to arm the drone.

We have discovered that:

- Base mode= 81 (drone disarmed)
- Base mode=209 (drone armed)

Drone disarmed



COM5 | 57600 | COM5-1-QUADROTOR

LOITER_U | 0 (Home) | Auto | Retracted | 100 | 100 | 80

```
+
+
+
+
+
+
base_mode: 81
+
+
+
MAVLINK_MSG_ID_SYS_STATUS
Battery (V):
11397
+
+
+
```

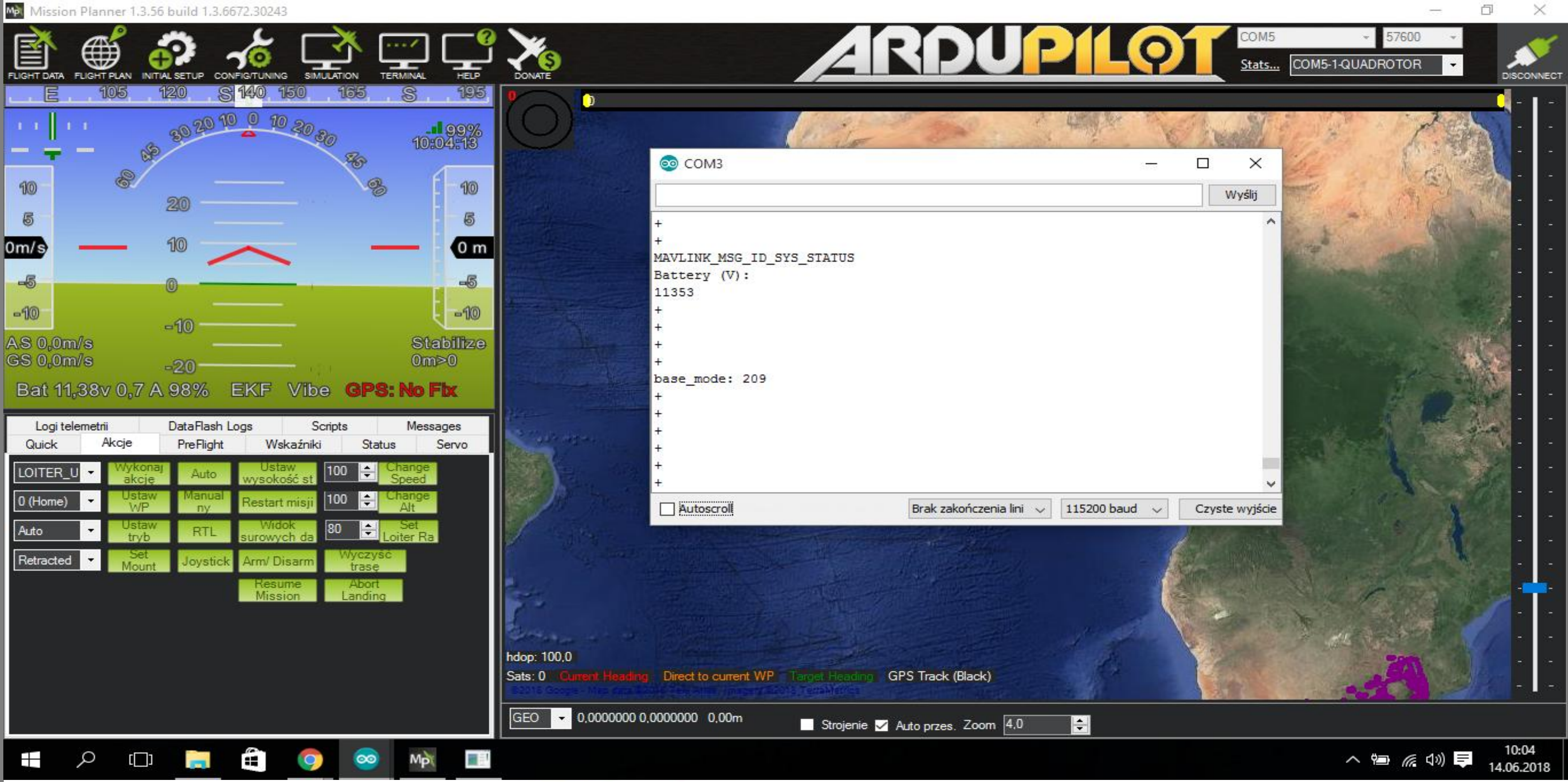
□ Autoscroll | Brak zakończenia lini | 115200 baud

hdop: 100,0
Sats: 0

GEO 0,0000000 0,0000000 3,86m

□ Strojenie ☑ Auto przes. Zoom 4,0

Drone armed



```
+
+
MAVLINK_MSG_ID_SYS_STATUS
Battery (V):
11353
+
+
+
+
base_mode: 209
+
+
+
+
+
+
```

□ Autoscroll   Brak zakończenia lini   115200 baud   Czyste wyjście

Recognition RC channels

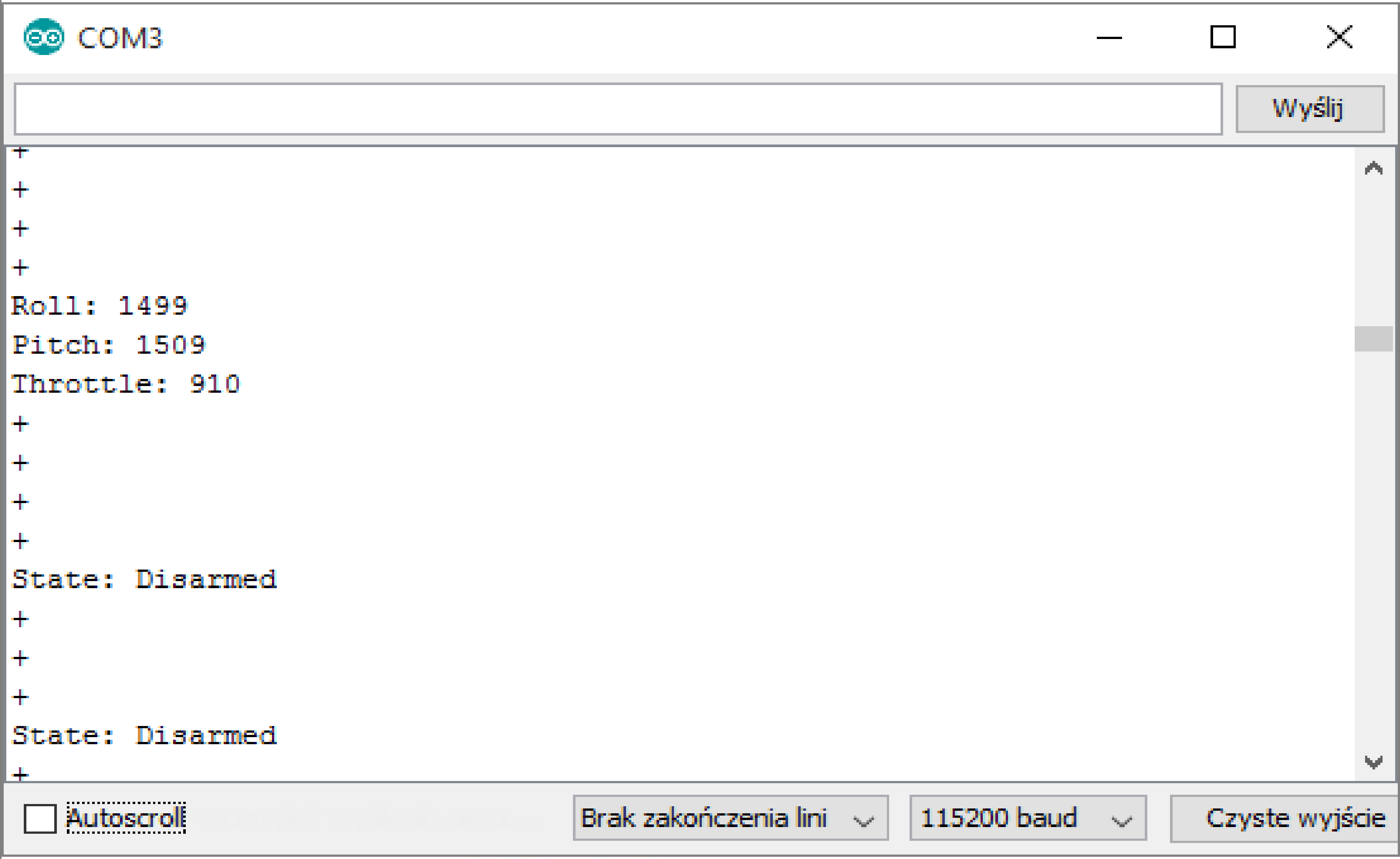
RC (Radio Controler- every equivalent of telemetry).

RC channels correspond to actions that are called on the telemetry e.g flight mode, pitch, rtl.

We had to check witch channels were for pitch, roll, throttle, yaw.

Channels

- 1- roll
- 2- pitch
- 3- throttle
- 4- yaw
- CH5- free
- -
- CH8 - free



```
+
+
+
+
Roll: 1499
Pitch: 1509
Throttle: 910
+
+
+
+
State: Disarmed
+
+
+
State: Disarmed
+
```

What command is for motors?

Command DO_SET_SERVO is only used for extra servos like triggers arms.

For setting speed of a motor we have to use

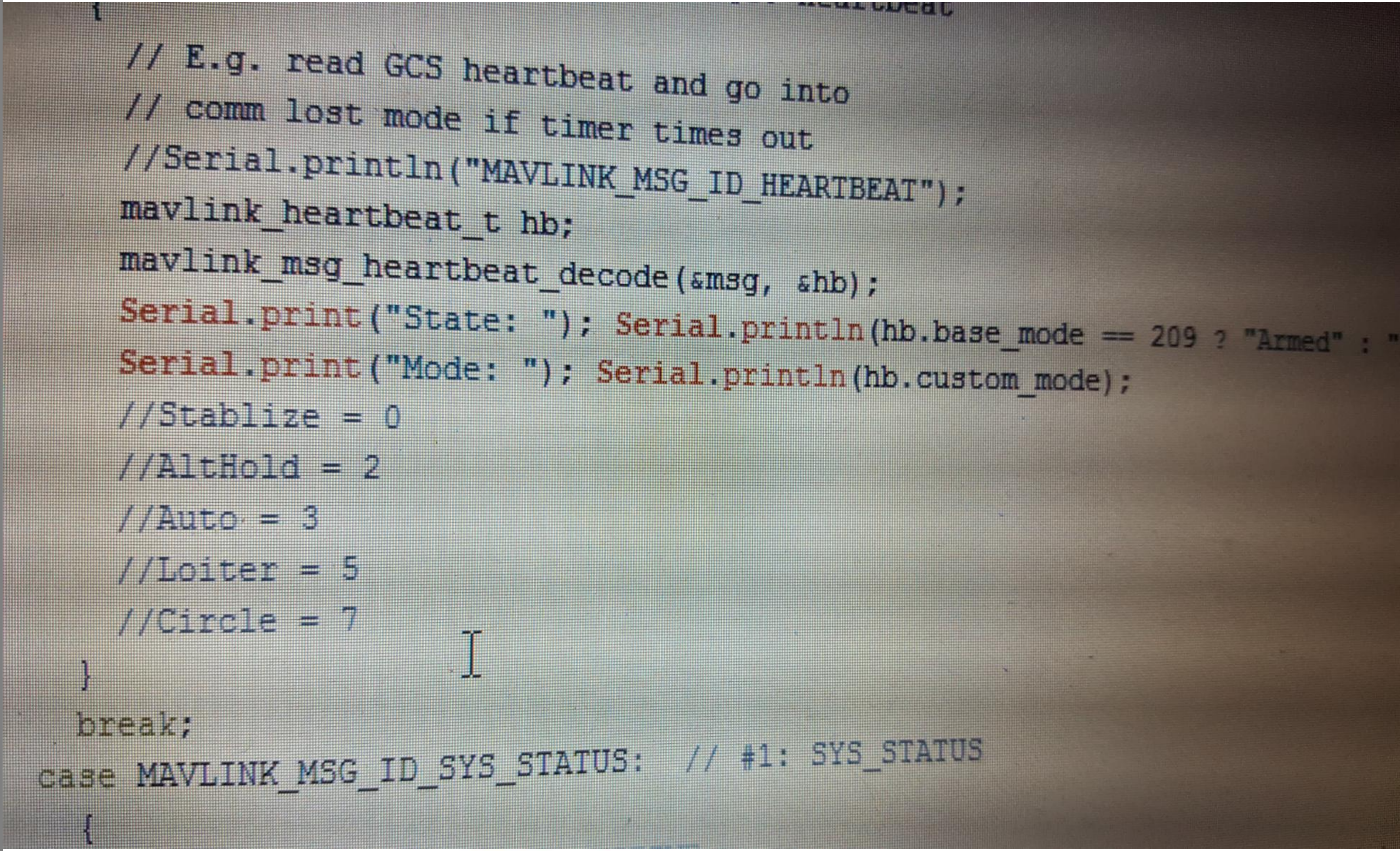
Mavlink_msg_rc_channels_override.

Reading flight modes

In order to check and change current flight mode in Arduino I read custom_mode parameter.

Custom_mode returns number representation of current mode.

A value that stands for a number can be checked in file common.xml



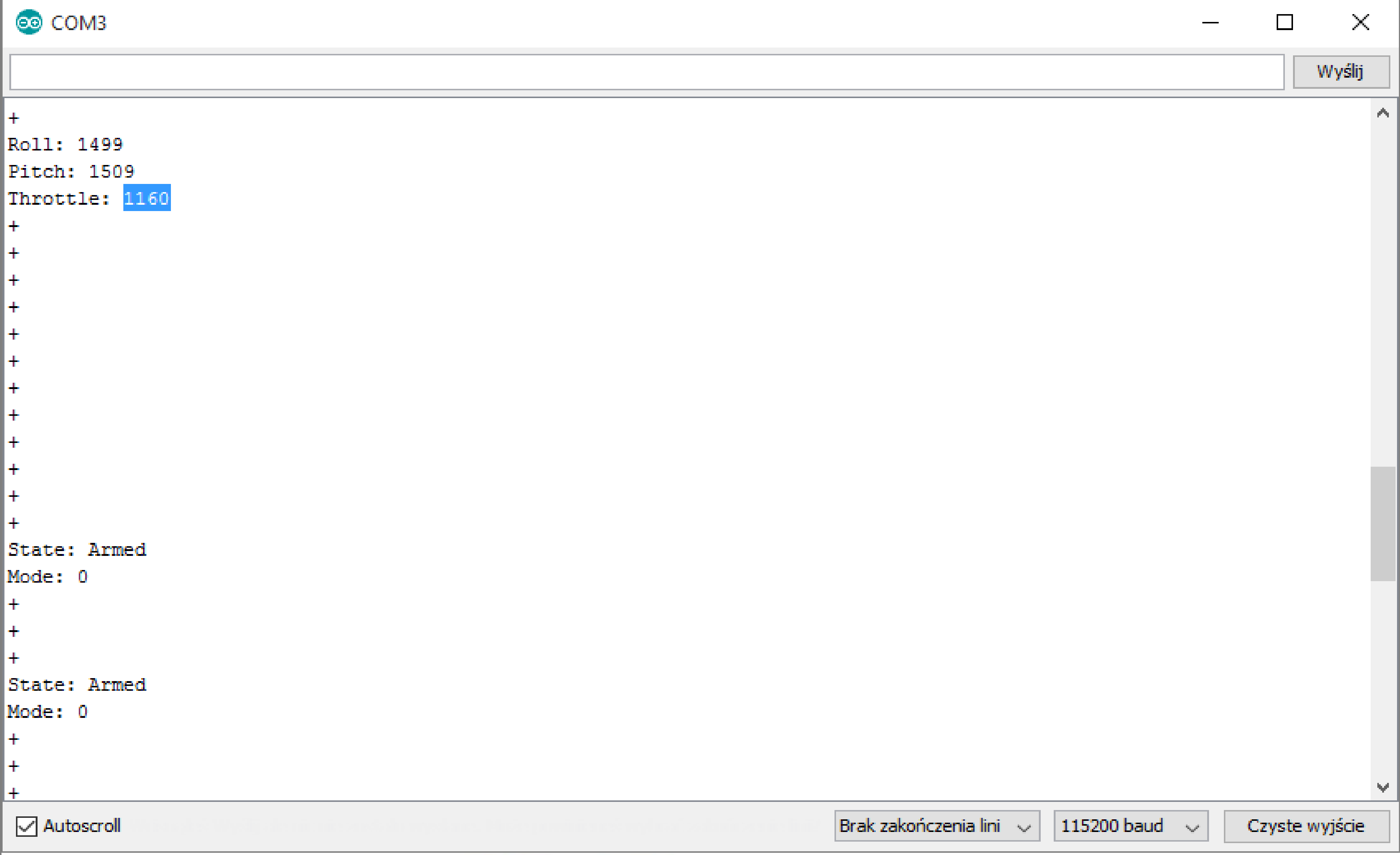
```
  {
    // E.g. read GCS heartbeat and go into
    // comm lost mode if timer times out
    //Serial.println("MAVLINK_MSG_ID_HEARTBEAT");
    mavlink_heartbeat_t hb;
    mavlink_msg_heartbeat_decode(&msg, &hb);
    Serial.print("State: "); Serial.println(hb.base_mode == 209 ? "Armed" : "
    Serial.print("Mode: "); Serial.println(hb.custom_mode);
    //Stablize = 0
    //AltHold = 2
    //Auto = 3
    //Loiter = 5
    //Circle = 7
  }
  break;
case MAVLINK_MSG_ID_SYS_STATUS:  // #1: SYS_STATUS
  {
```

The biggest breakpoint

After many, many weeks I have run the propellers without using mission planer or telemetry.

Our drone starts running above 1150 value over throttle.

Before that the drone had to be armed.

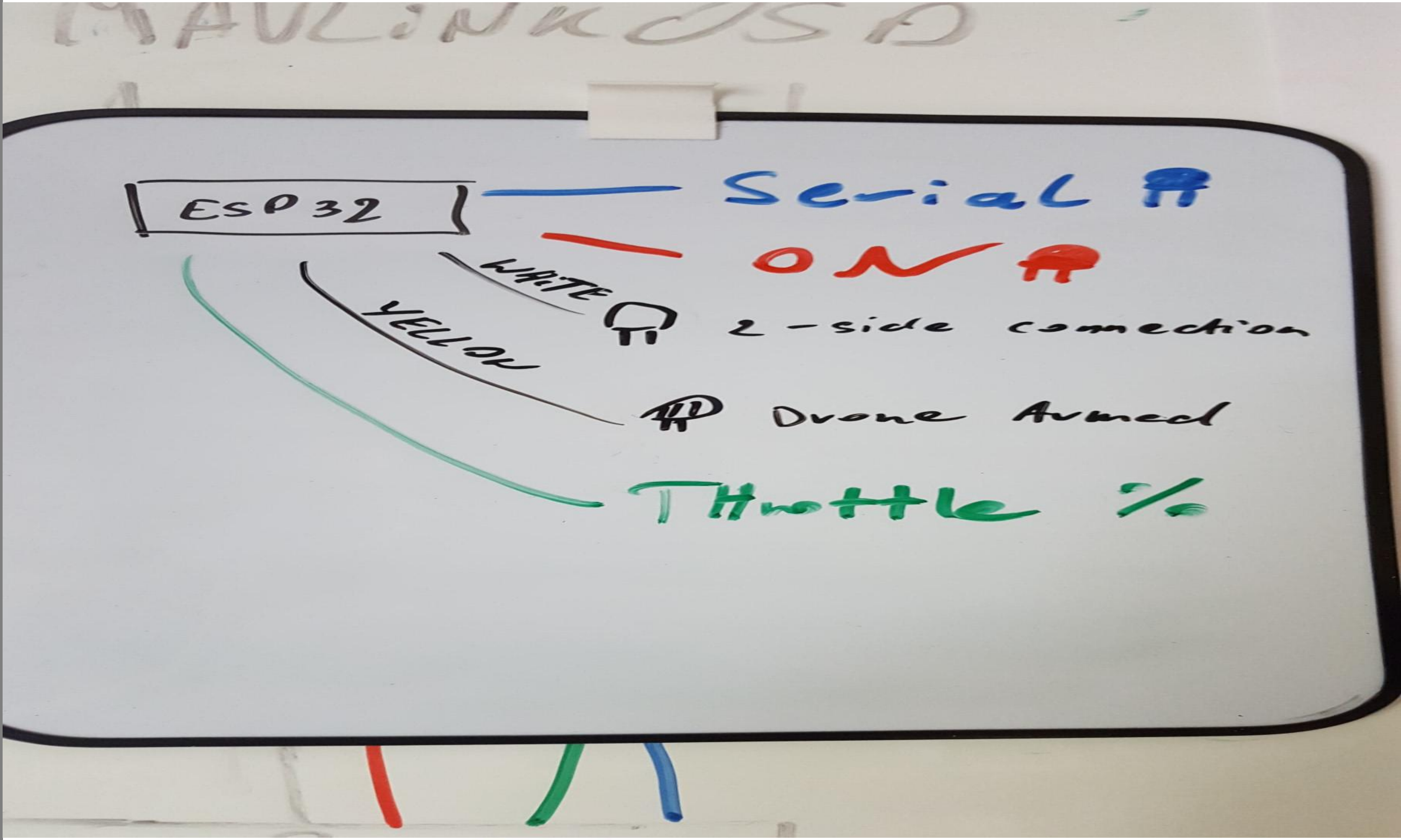


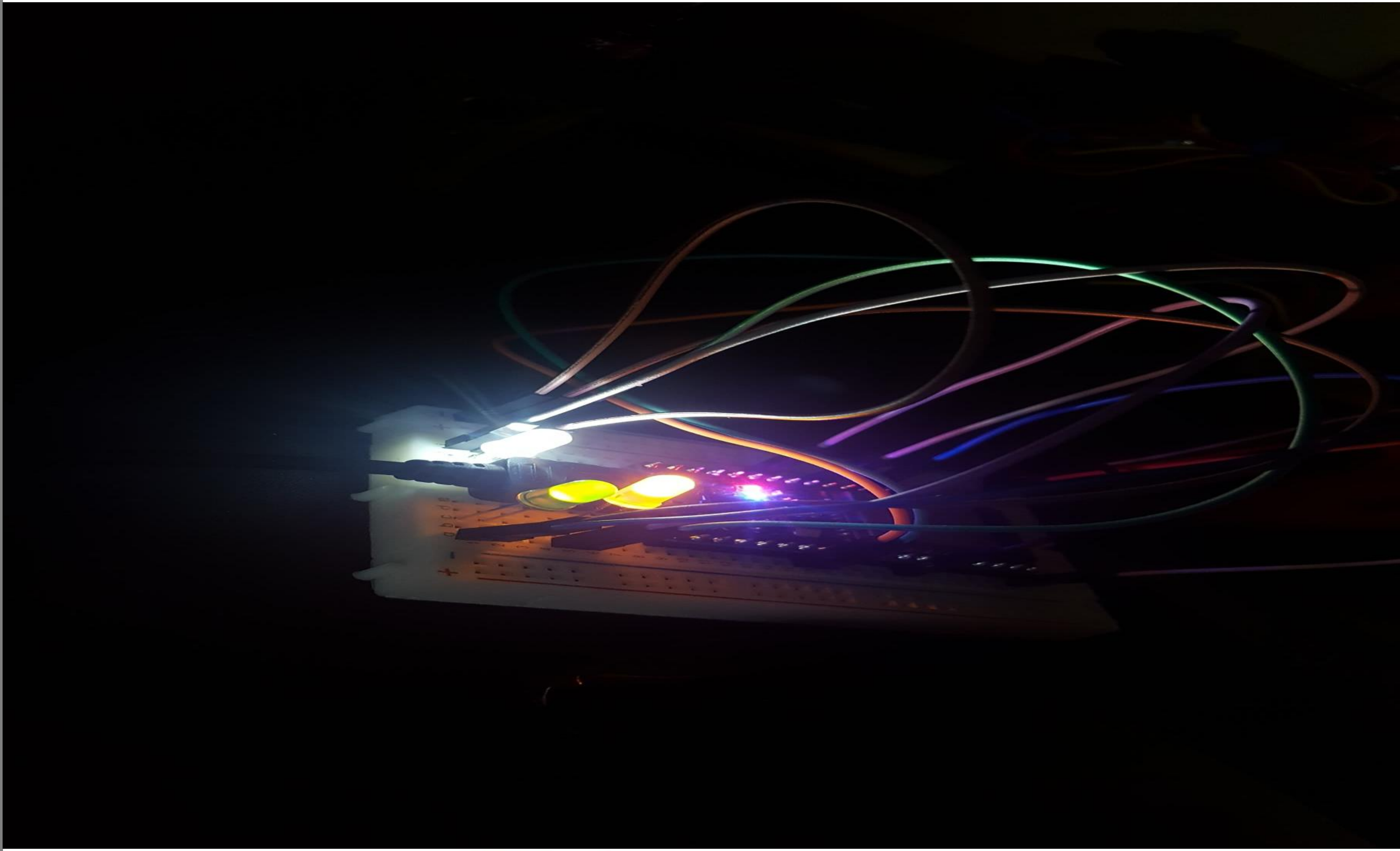
```
+
Roll: 1499
Pitch: 1509
Throttle: 1160
+
+
+
+
+
+
+
+
+
+
+
+
State: Armed
Mode: 0
+
+
+
State: Armed
Mode: 0
+
+
+
```

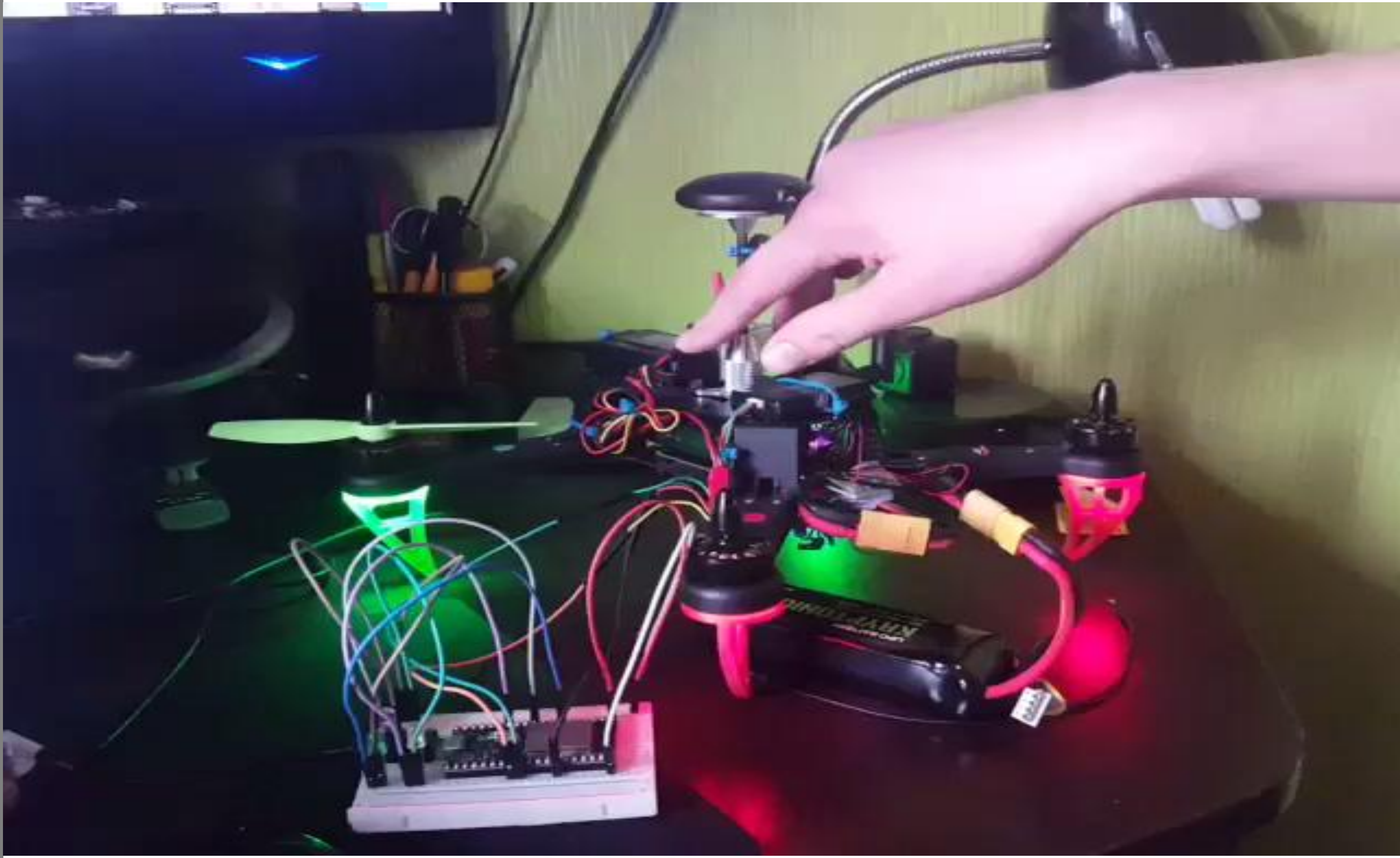
Designing indication system

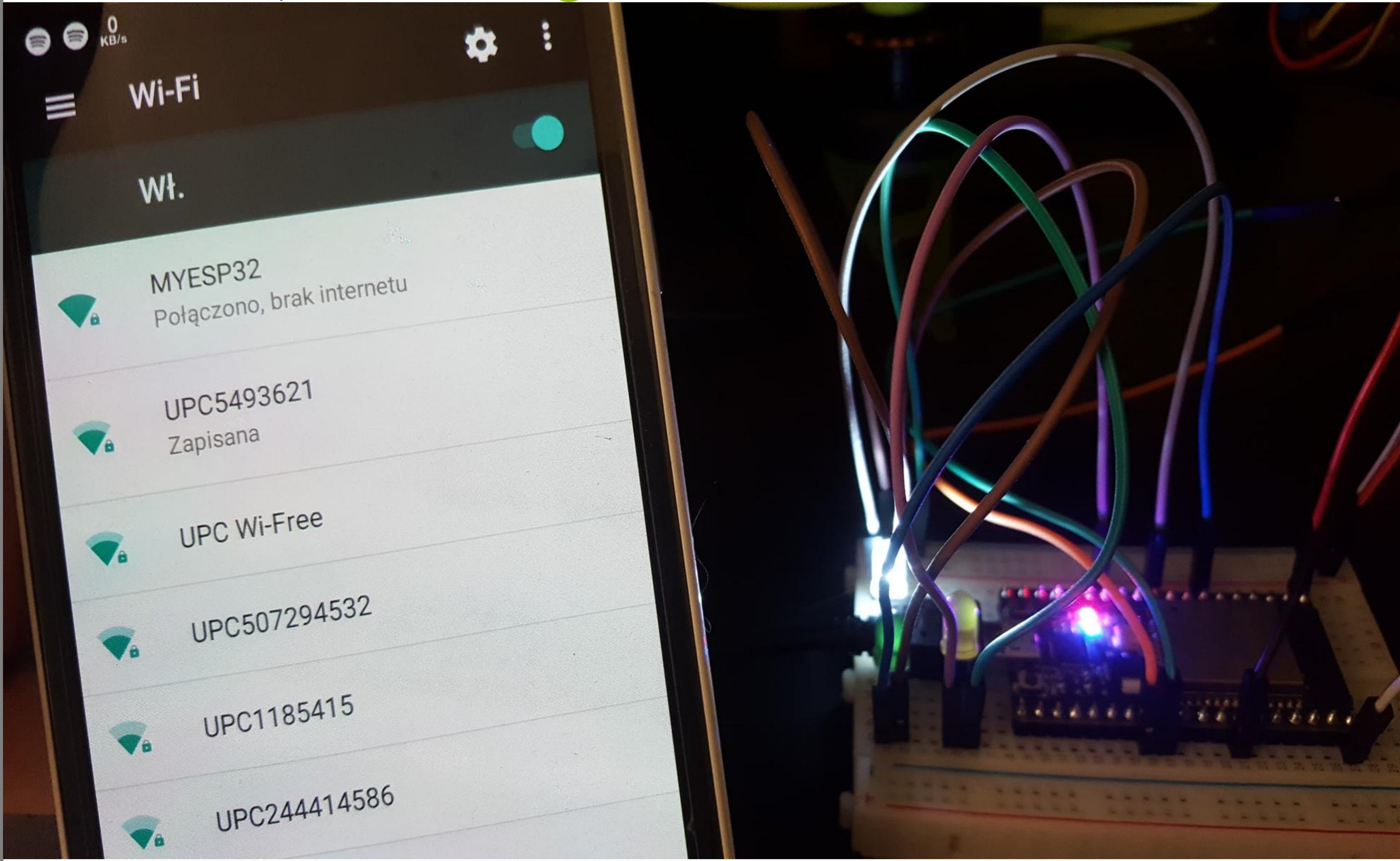
For pointing present state (arm, stabilize, throttle) we use builtin and outers leds.

- Blue- serial data read
- Red- state of esp32
- White- connection with a phone
- Yellow- arming state
- Green- throttle in percent


ESP32
Serial
ON
WHITE
YELLOW
2-side comection
Drone Armed
Throttle %







Wi-Fi

Wł.

MYESP32
Połączono, brak internetu

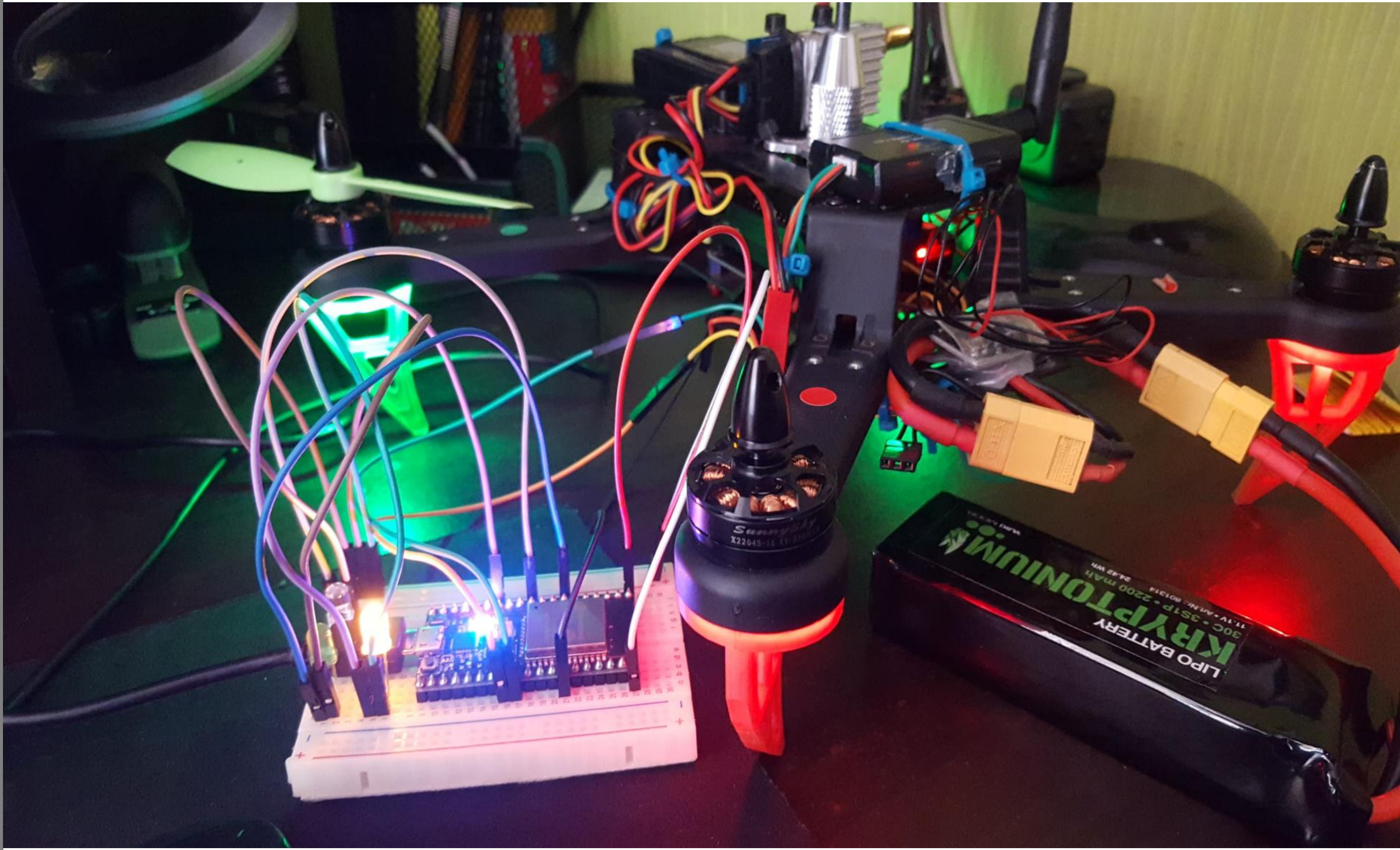
UPC5493621
Zapisana

UPC Wi-Free

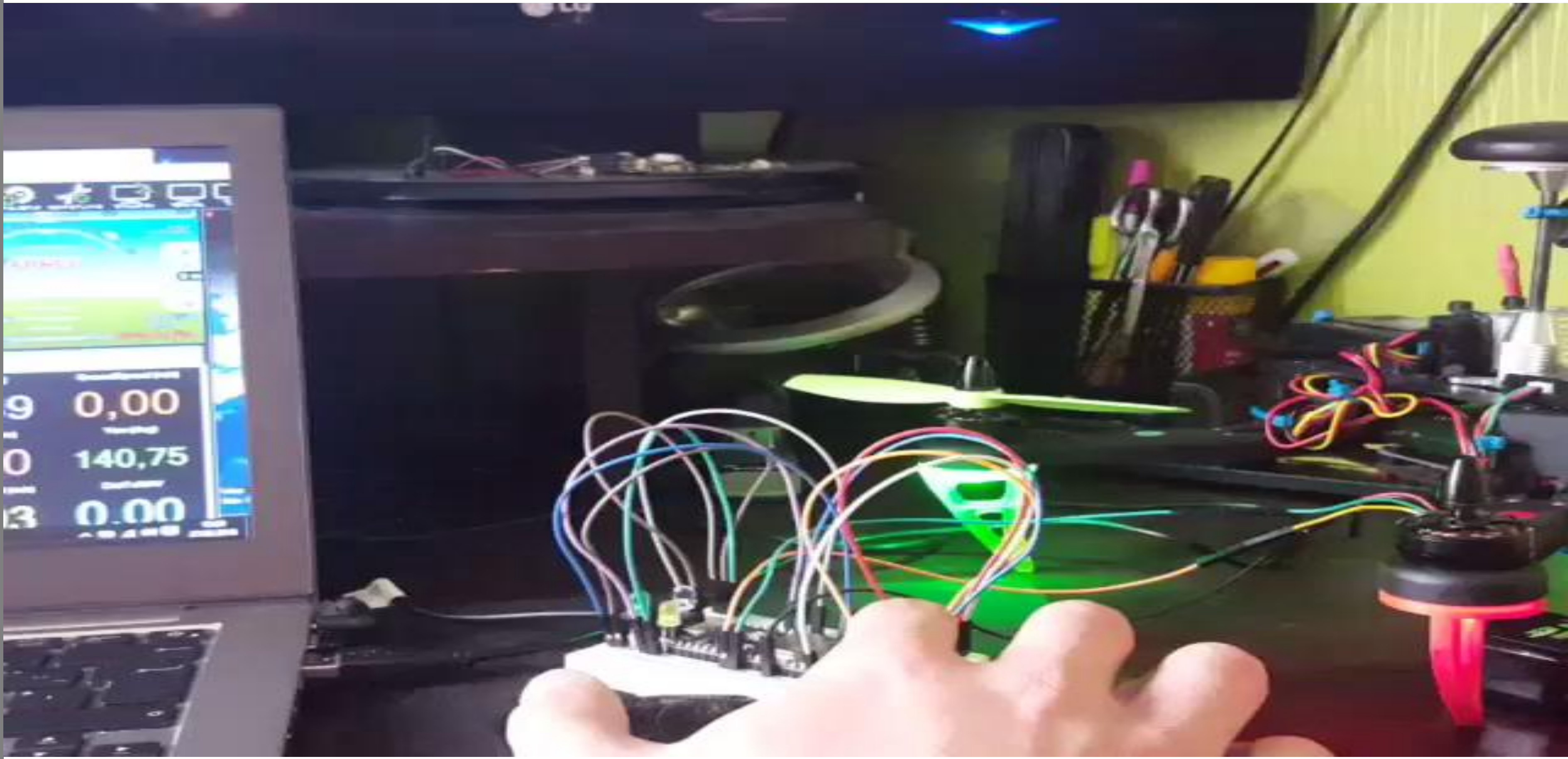
UPC507294532

UPC1185415

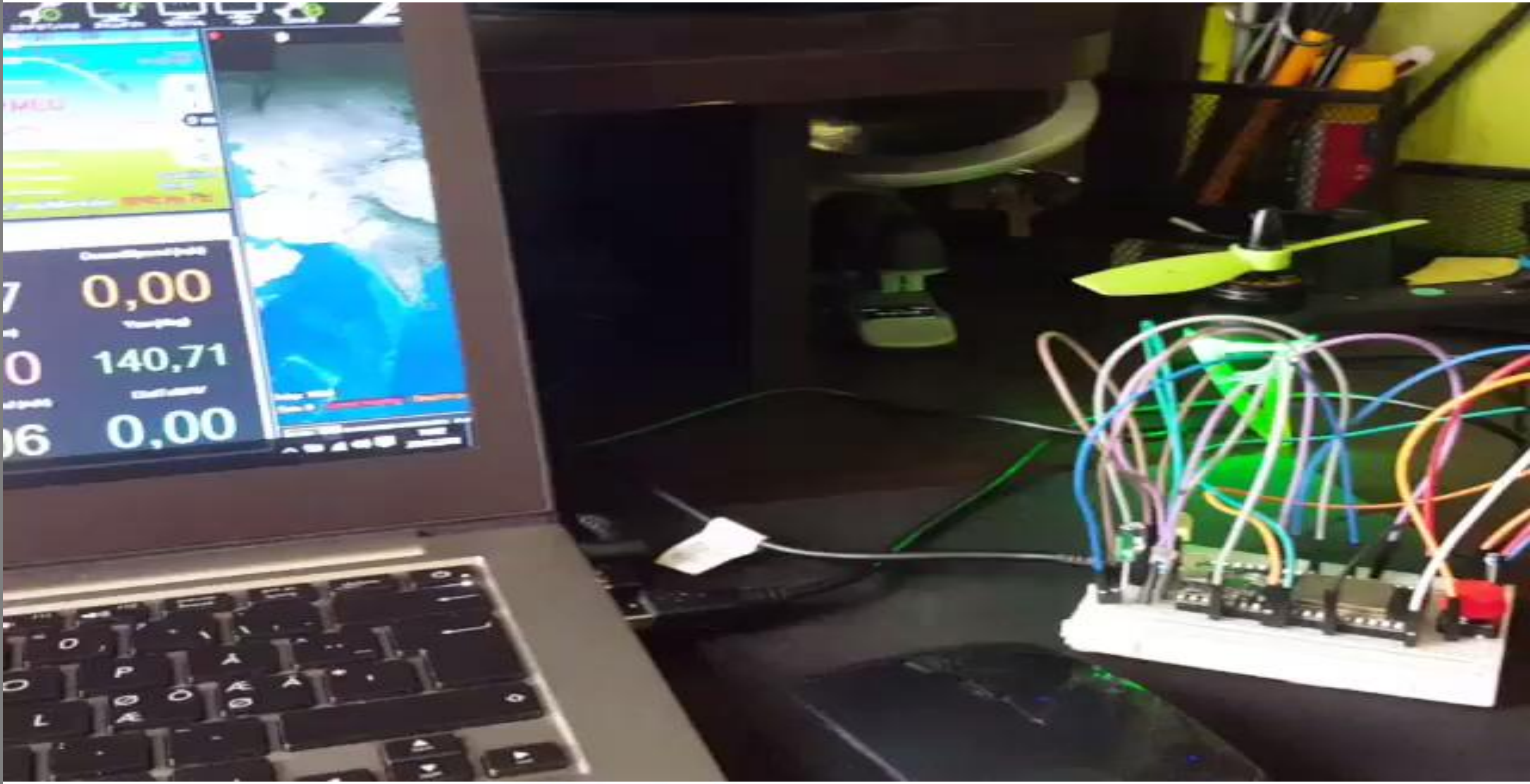
UPC244414586


SunnySky
X2204S
LIPO BATTERY
KRYPTONIUM
30C • 3S1P • 2200 mah

Arming with a button



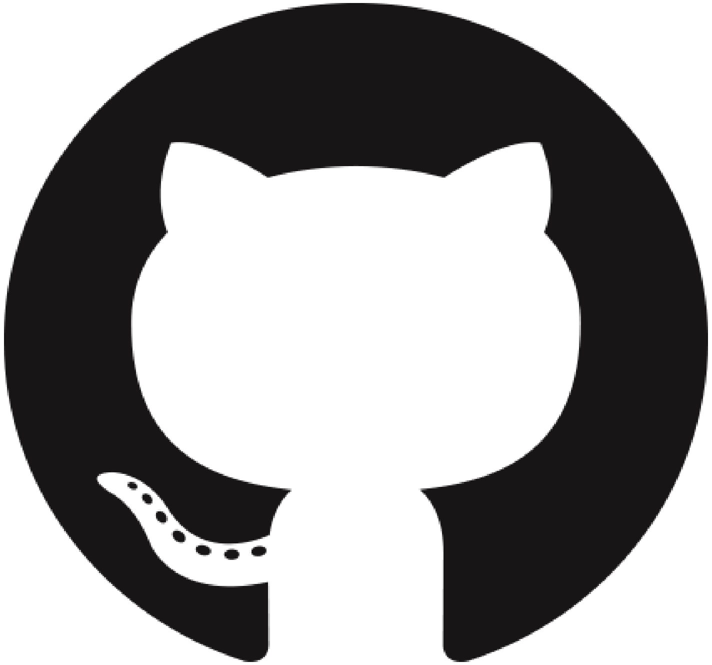
Final esp32 control



Code summary

Arduino	Android
•Created files: 15 •Libraries: 34 •Lines of code: 645	•Created files: 43 •Libraries: 26 •Lines of code: 19 087 (with libraries)

Full code uploaded



Tutorials on my site

After this meeting I’m going to publish code and tutorials on github and my blog: **tkadziolka.pl/blog.html**

Thank YOU!