YOLO and CV Tracking

Synerduino Hardware setup

VERSIONS: F405, F411, H743 ,2560

For more Information:
www.synerflight.com

INTRO

- Computer Vision Serial Interface application

In todays Era of AI its apparent that AI be able to interact with its environment as what we describe as physical AI manifesting its self in Physical machines

- This works in conjunction with the Synerduino Board ,INAV PLC , Arduino ADC
- Requires FPV camera and OTC UVC Receiver
- Wifi Camera that outputs RTSP on a server

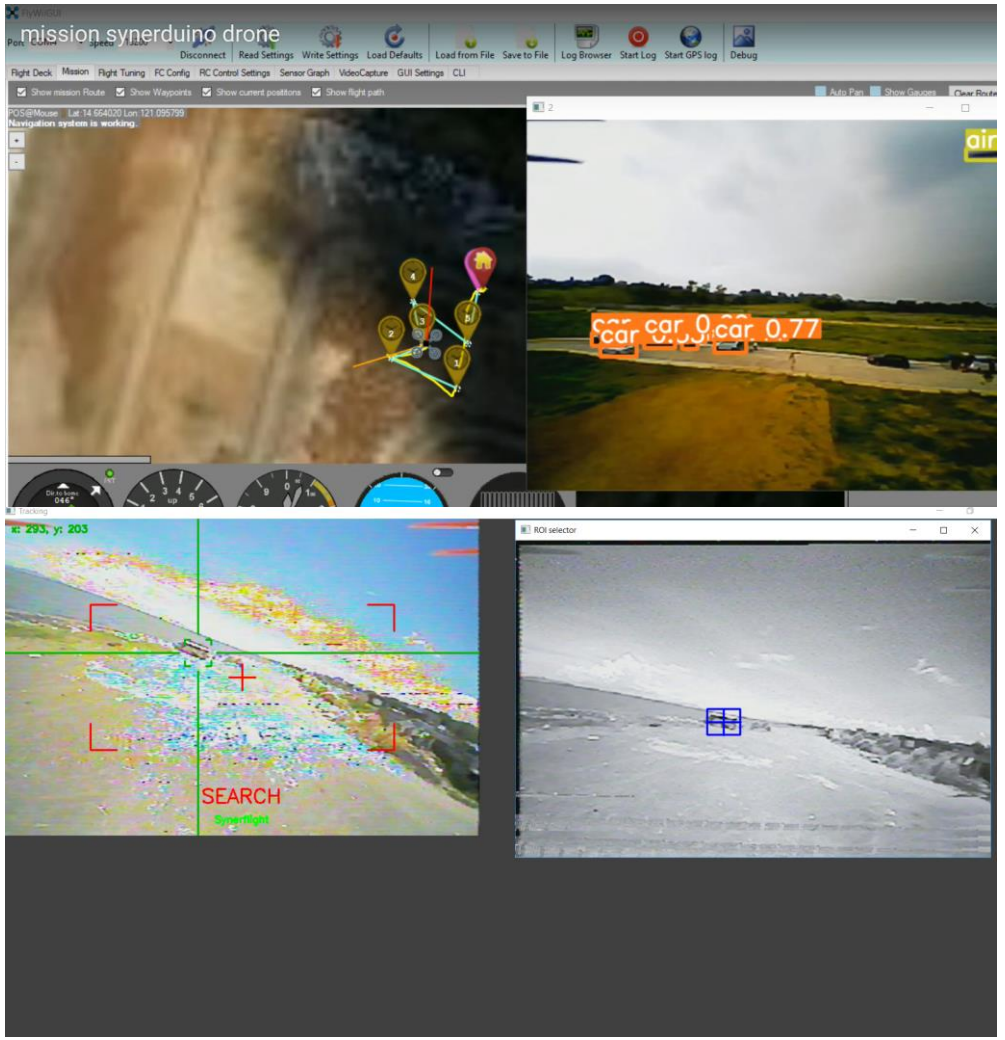
Of the Following Board

- Synerduino Arduino
- Synerduino STM

INTRO

Computer Vision Serial Interface application

Teaching the Machine what its Looking For and the Condition of the subject its looking at

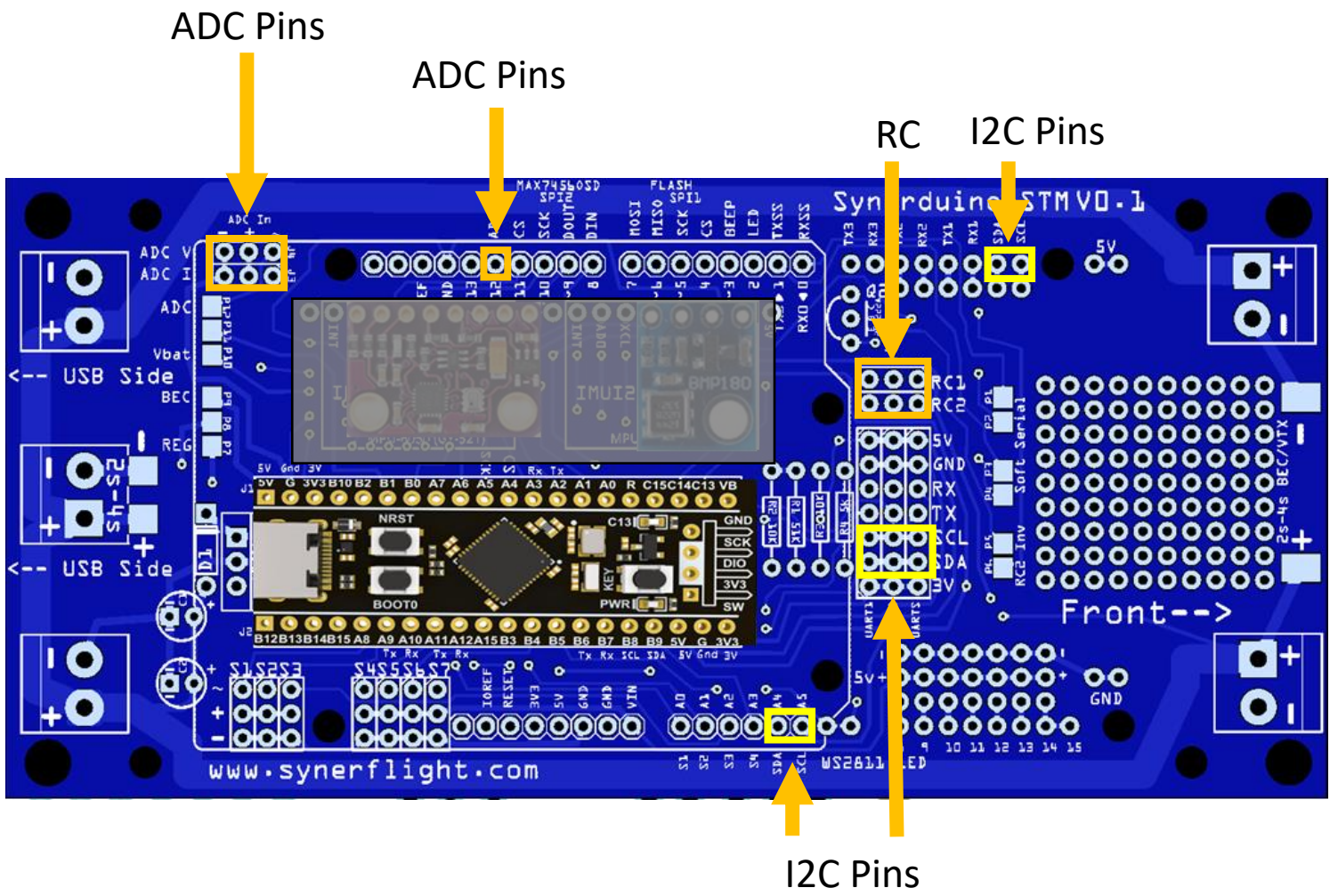


SYNERDUINO STM F411 SHIELD

Access interface Pins

Analog ADC method

- A1-Voltage
- A2-Current
- A3-RSSI
- A4-Airspeed

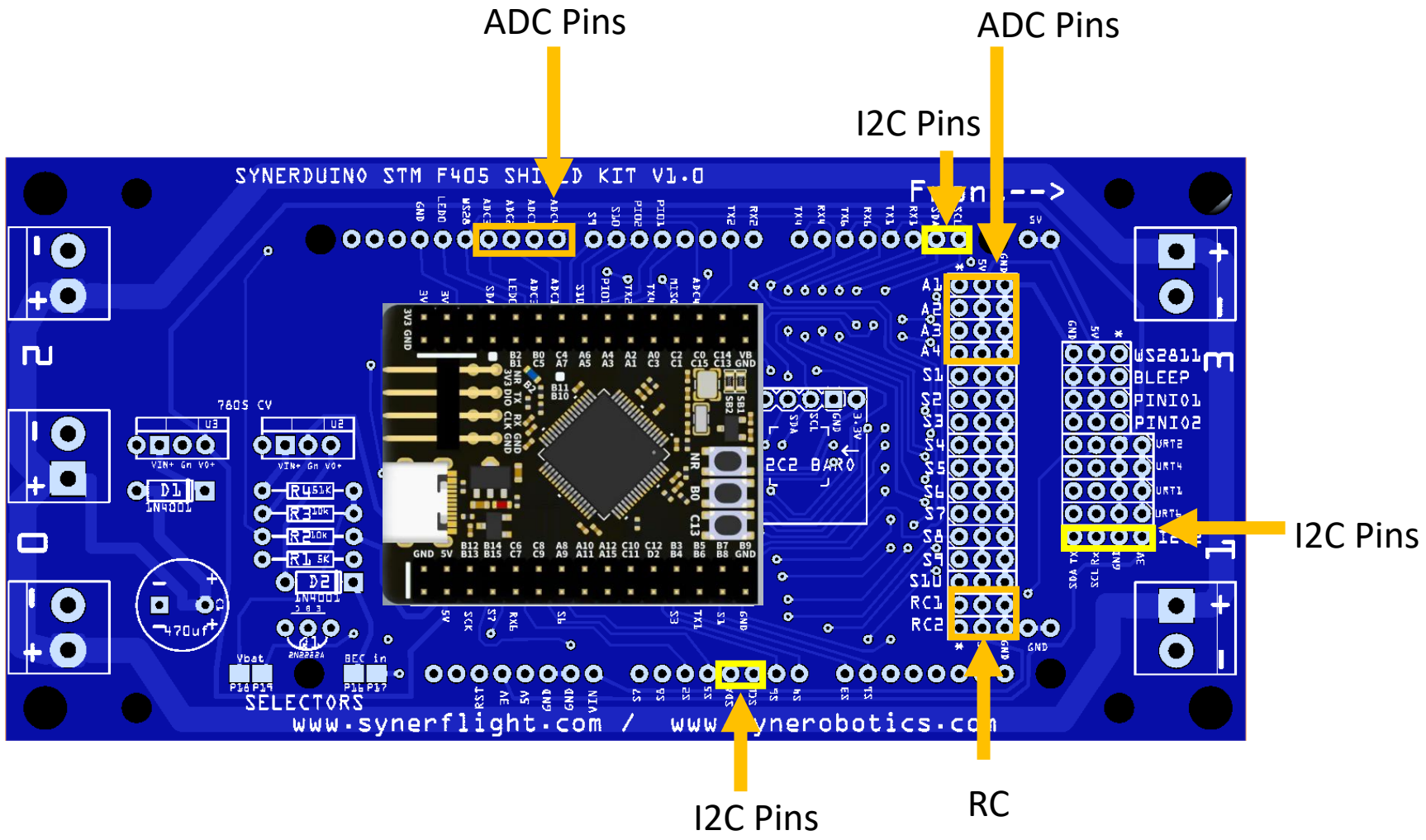


SYNERDUINO STM F405 SHIELD

Access interface Pins

Analog ADC method

- A1-Voltage
- A2-Current
- A3-RSSI
- A4-Airspeed

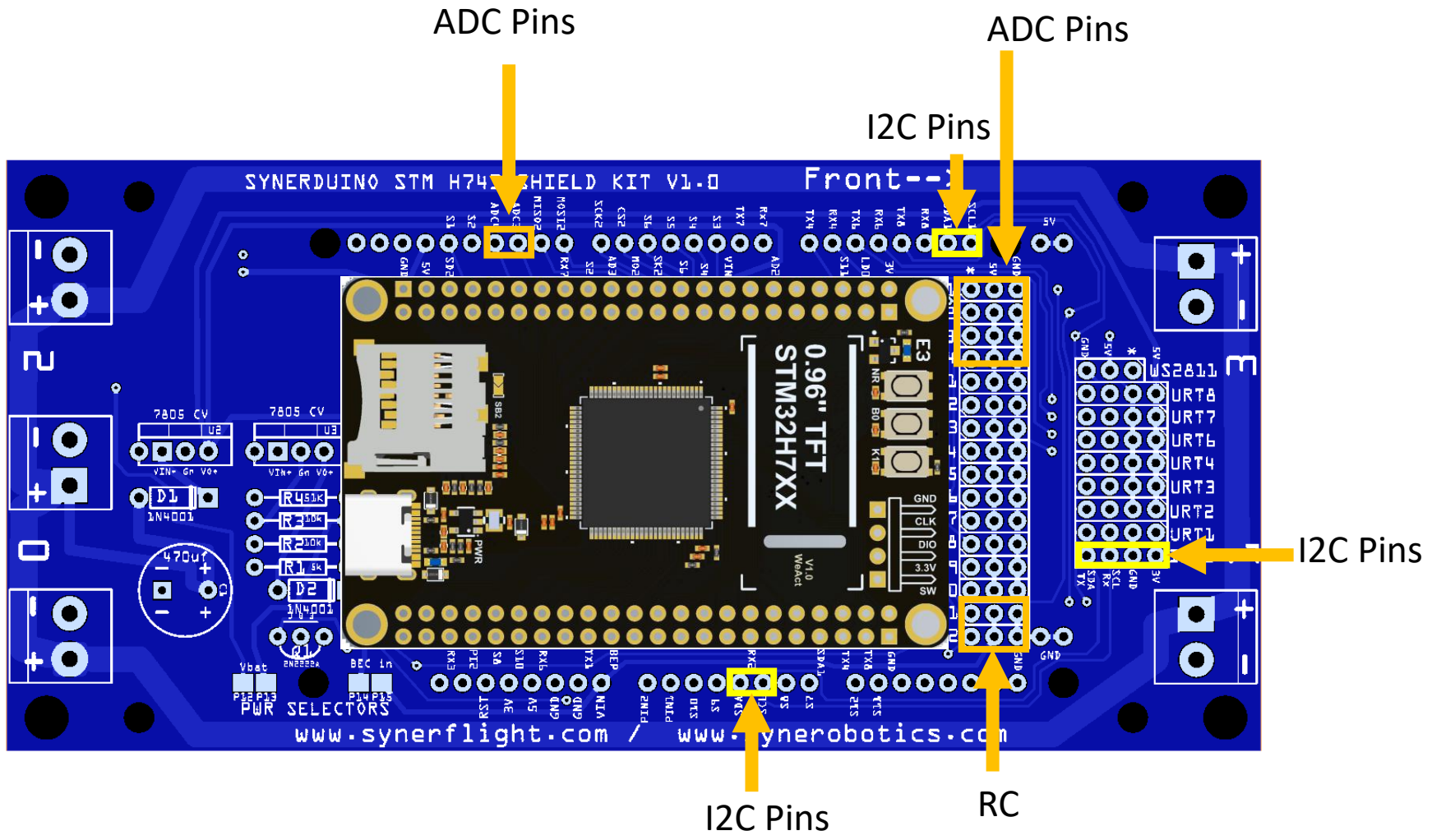


SYNERDUINO STM H743 SHIELD

Access interface Pins

Analog ADC method

- A1-Voltage
- A2-Current
- A3-RSSI
- A4-Airspeed



SYNERDUINO ARDUINO

Aux ADC in

Description: Auxiliary input for connecting additional sensors or components that output analog signals, allowing the board to read and process external analog data.

Note: *Input Voltage: 3.3-5V*

ESC / Servo PWM Out

Description: These are 36 PWM (Pulse Width Modulation) output pins used to control the motors through ESCs (Electronic Speed Controllers) or servos.

Serial Pins

Description: These are 12 serial pins for communication with external devices or modules like GPS or telemetry systems using a UART interface.

GPS Serial Pins

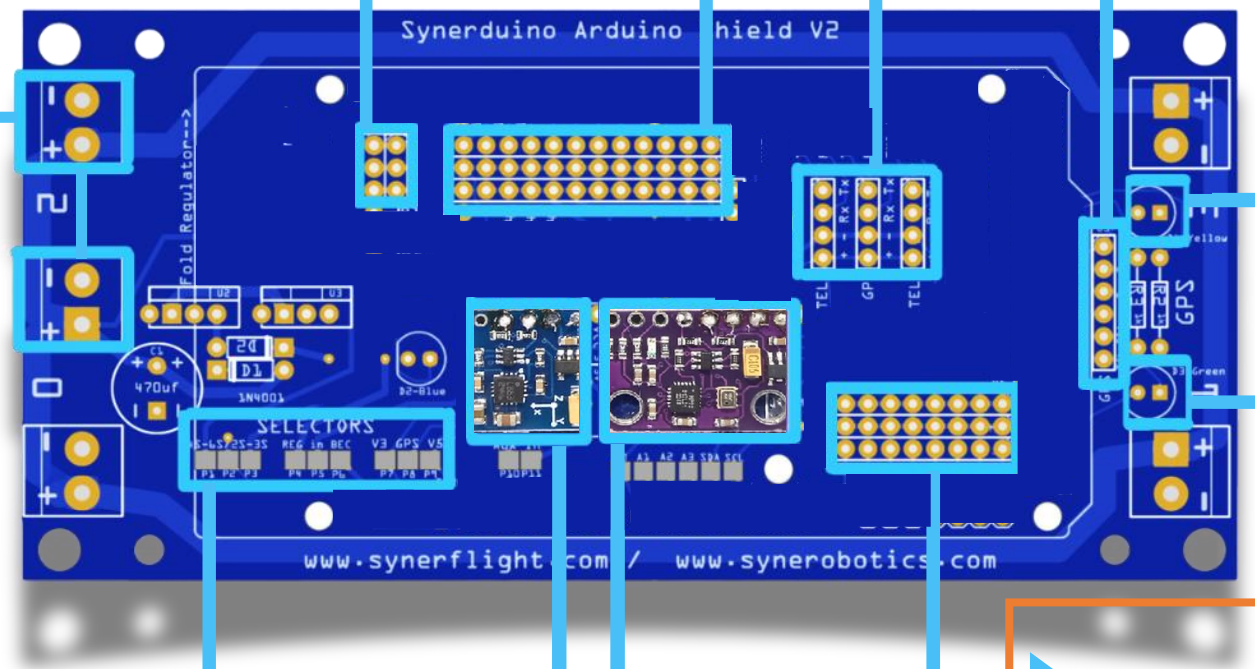
Description: These are 6 dedicated pins for connecting a GPS module's TX and RX (Transmit and Receive) lines for serial communication.

GPS LED

Description: This LED blinks or stays lit depending on whether the GPS is locked (has found satellites) or is searching for a signal.

Status LED

Description: A general-purpose status indicator for the board. It could be used to indicate power, initialization, or operational status.



Power Input

Description: This is the main power input for the board, designed for a 3-cell (3S) - 4-cell (4S) LiPo battery - 11.1V and 14.8V respectively. It powers the ESCs, servos, and other components on the board.

Soldering Note: ESCs should only be soldered on the top side of the board, ensuring the solder joints do not penetrate through to the bottom.

Jumper Pads Selector Zone

Description: These are PWM (Pulse Width Modulation) output pins used to control the motors through ESCs (Electronic Speed Controllers) or servos.

5883 Mag

IMU MPU-9250

Note: *These modules may vary depending on the manufacturer or version. Some versions might use different sensor combinations that could exclude certain components, like the magnetometer.*

RC PWM In

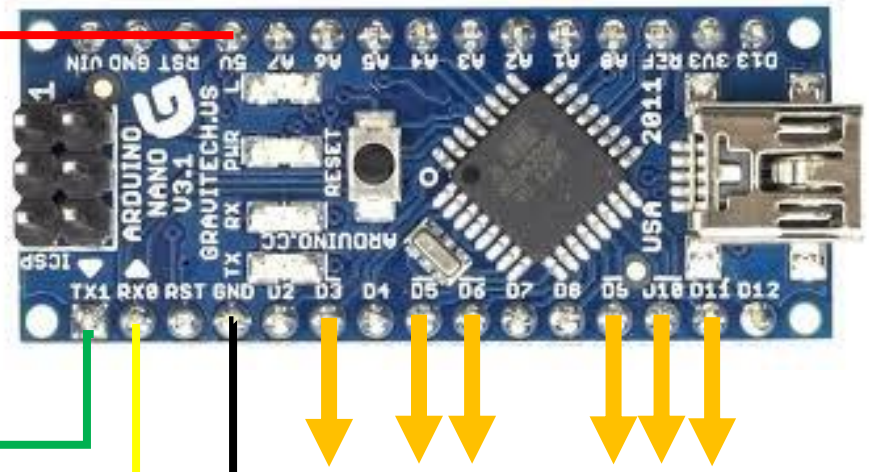
Description: These are 24 pins which accept PWM signals from an RC (radio control) receiver, allowing manual control via an RC transmitter.

Companion Arduino ADC Mediator

Access interface Pins

Analog ADC method

- D3 – PWM0
- D5 – PWM1
- D6 – PWM2
- D9 – PWM3
- D10 – PWM4
- D11– PWM5



This PWM output can be translated to ADC input

Connecting to Synerduino A1-A4 Pins

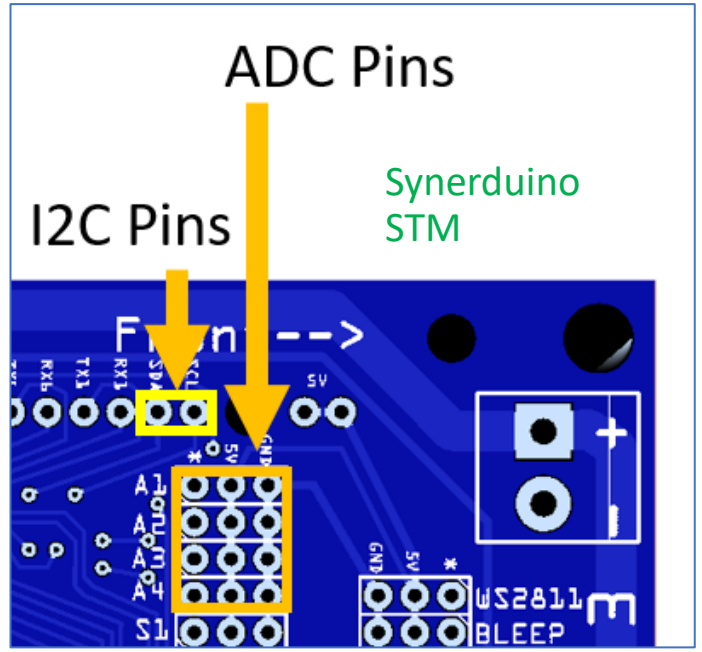
D3 - A1

D4 - A2

D6 - A3

D9 - A4

Analog Output PWM of
0-225 8-bit resolution



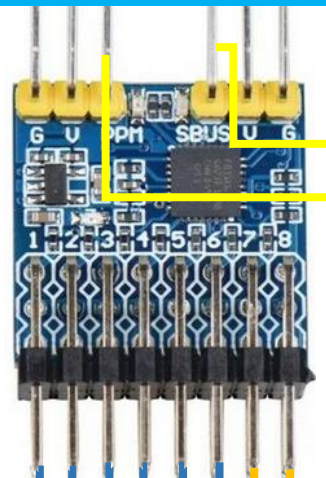
Synerduino
Arduino

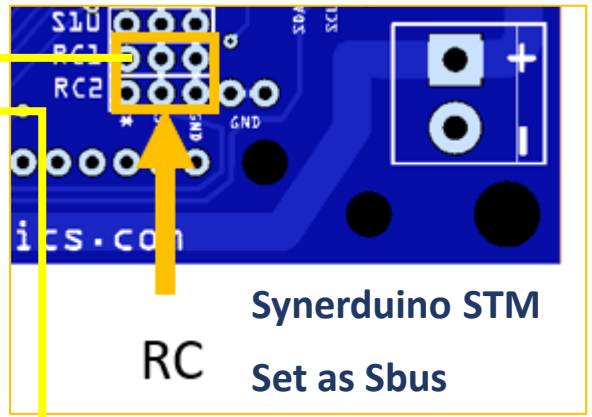
A0 – Analog input

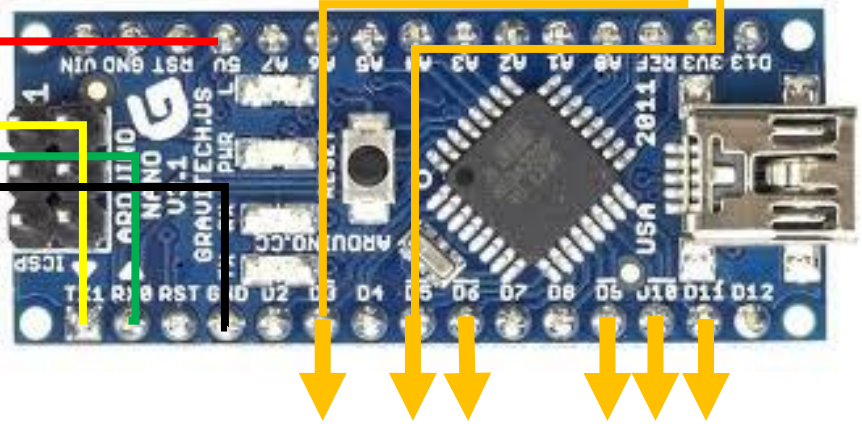
Companion Arduino RC PWM Mediator

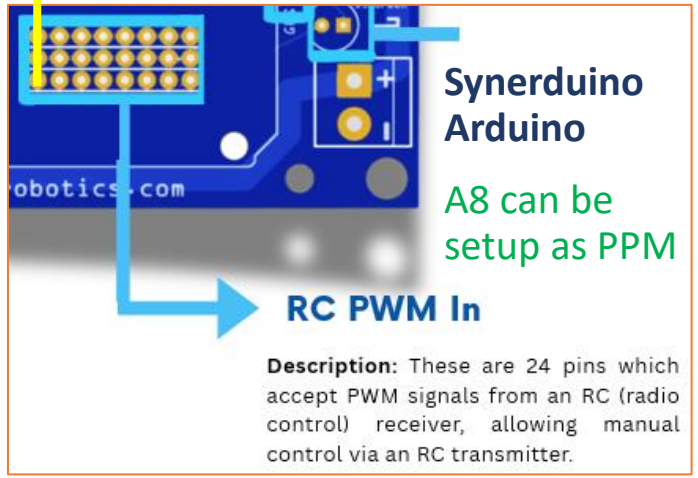
The PWM interface requires a PPM PWM SBUS converter to accomplish this setup which interface as PWM on one of the free channels no used by the main drone controls

Its best to use PWM receiver for this application as translating PWM and Sbus multiple times would increase Latency









This PWM output can be translated to Servo PWM

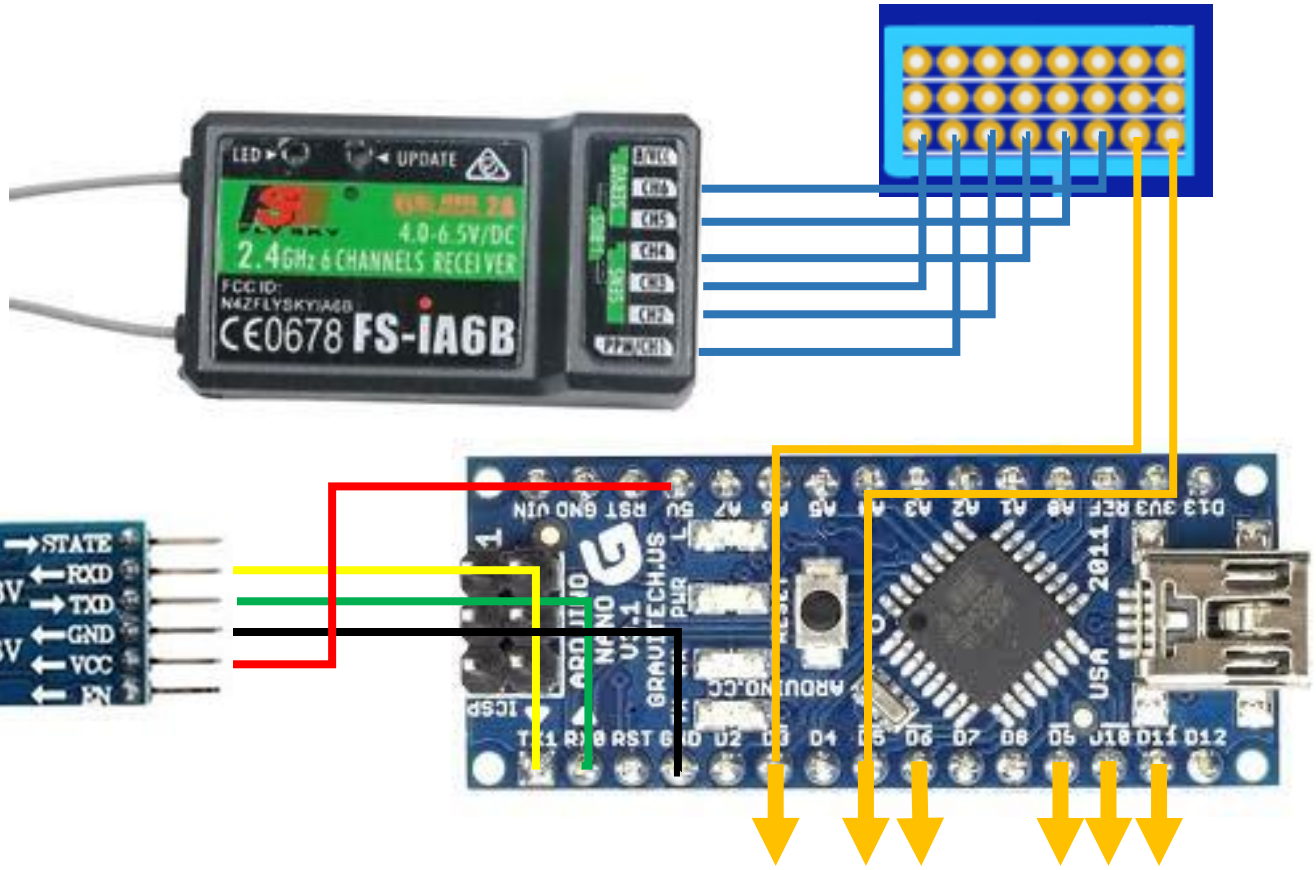
Note: For Arduino Mega all Digital pins can be set as PWM

Companion Arduino RC PWM Mediator

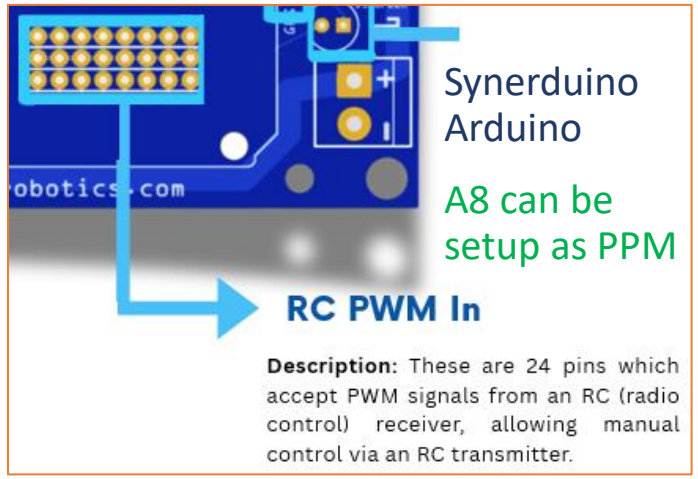
The PWM Alternative is to use A8 to A15 PWM with 2 of the free channels for the PWM mediator

Its best to use PWM receiver for this application as translating PWM and Sbus multiple times would increase Latency

Arduino PWM setup



This PWM output can be translated to Servo PWM



Note: For Arduino Mega all Digital pins can be set as PWM

Hardware Setup / Synerduino Arduino

The is useful for analog base inputs
PWM, or pulse width modulation is a technique which allows us to adjust the average value of the voltage that's going to the electronic device by turning on and off the power at a fast rate.

This method is the most simplest interface that one can incorporate even the most basic of logic circuits for simple tasks

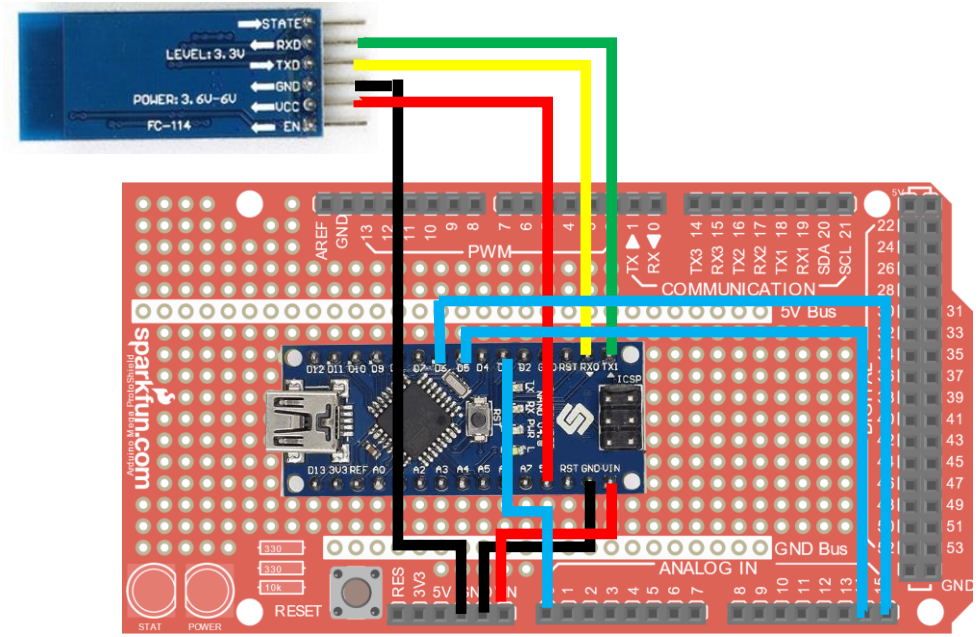
Arduino Nano

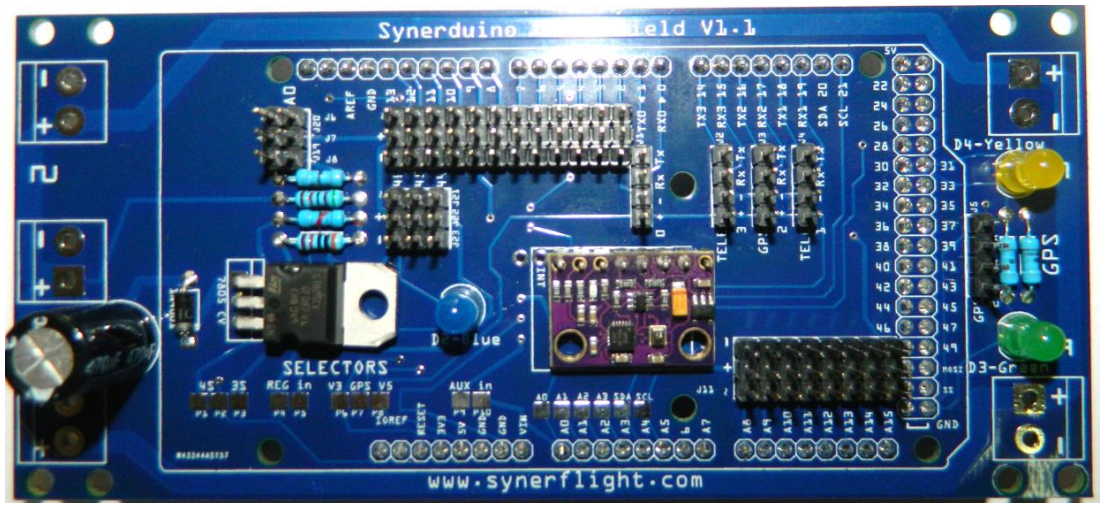
The Arduino nano on top of a Prototyping shield serves 2 purpose

- Provide a bypass from the Arduino mega to the Synerduino shield
- Allows PWM from the D3 of the Nano to interface with the A0 of the Mega
- Purchase an Arduino NANO without the headers installed (Omit the Top SPI headers) and directly solder the Nano to the Development shieid via wire to ensure it still fits under the Synerduino shield

An 2nd Bluetooth or Serial device required to interface with a computer running Yolov5

Serial 0 of the Arduino Board





Note: this development shield is the 2nd layer under the Synerduino board
And the Arduino mega on the 3rd Layer

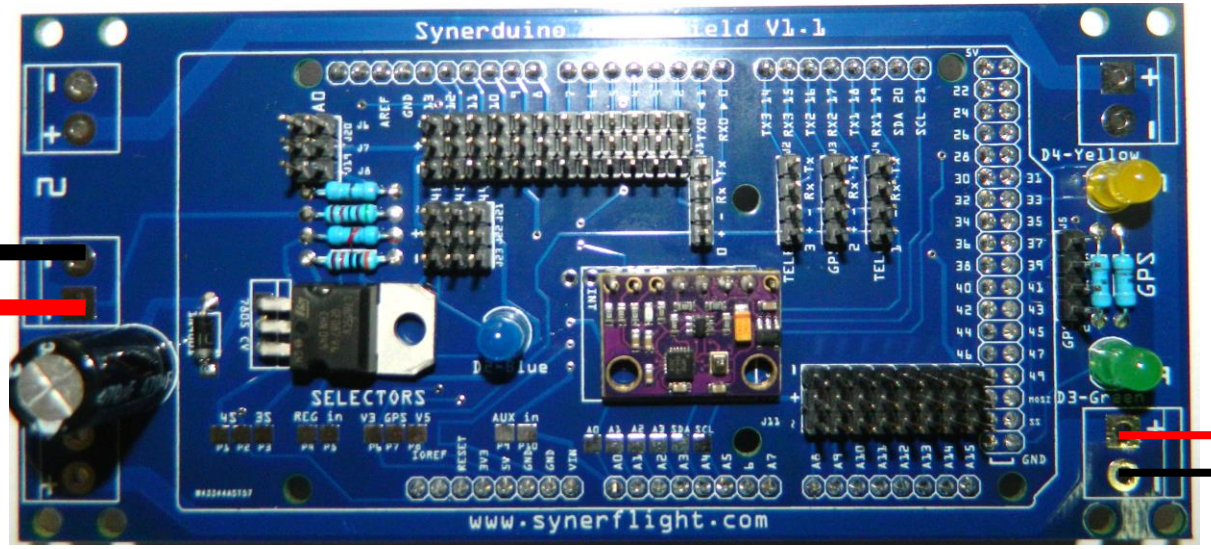
FPV Standalone / Synerduino Arduino

This requires no introduction as it uses a BEC to supply a standalone FPV25mw camera with integrated VTX

This also can be apply to split camera and VTX sets as well (some Standalone VTX can support 2s to 6s meaning they can directly hook up to the main batter Pads with requiring a BEC supplement)

FPV camera 25mw Standalone

Battery 3s or 4s



BEC or Buck converter supplying extra power

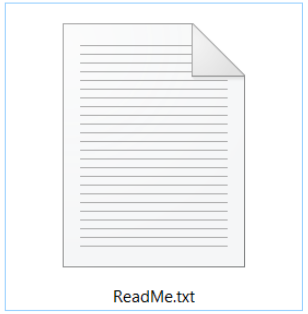
Setup your Environment

The application Runs on Python 3.9

Preparing your Environment including

Zip file Samples

- CV Tracking
- YOLO



Setup YOLO

Readme.txt contains all the information needed for setting up your Yolov5 synerduin

Install requirement .bat Batch install all your Yolov5 requirements

Edit : pip install -r SynerCV2/requirements.txt

This Runs the application Yolov5 with Synerduino Detection

python SynerCV2/detectardu20.py --source 0 --weights yolov5s.pt --img 640 --class 0

See: Yolov5 tutorial on custom Train data and image Annotators

https://www.makesense.ai/

Setup YOLO

Datas - contains all the Parameters for configuring your CV Serial Coms and instructions to send to the Arduino

datas folder contains the readme file for settings configuration and documentation
datas/baud.txt - change baud
datas/com.txt - change Serial com
datas/ipt.txt - object location on screen
datas/noct.txt - names or classes
datas/soct.txt - number of objects count

Setup YOLO

Syner CV2 - contains all the Yolov5 applications

detectardu20.py – Contains the Python codes and the PT Train image files

Setup YOLO

ArduinoCV - this contains the Arduino sketches and adp files of Ardublock (Arduino 1.8.18 compatible)

Tool folder containing Ardublock must be extracted in

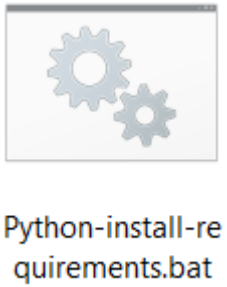
this PC/Documents/Arduino/Tools

This sketch is used to convert the Serial communication coming from the YOLOV5 and Converting it into PWM or ADC output for the Arduino companion board to read and send appropriate input to Synerduino board to registered

Setup CV Tracking

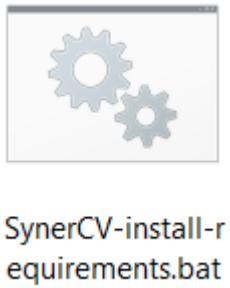
Readme.txt contains all the information needed for setting up your CV Tracking synerduino

This operates without serial Arduino connection



Python-install-requirements.bat

Allows you to create ROI Box on your target on start



SynerCV-install-requirements

Auto create a target lock on press E

Setup CV Tracking

Datas - contains all the Parameters for configuring your CV Serial Coms and instructions to send to the Arduino

datas folder contains the readme file for settings configuration and documentation
datas/baud.txt - change baud
datas/com.txt - change Serial com
datas/ipt.txt - object location on screen
datas/ipsource.txt – for mp4 video

Setup CV Tracking

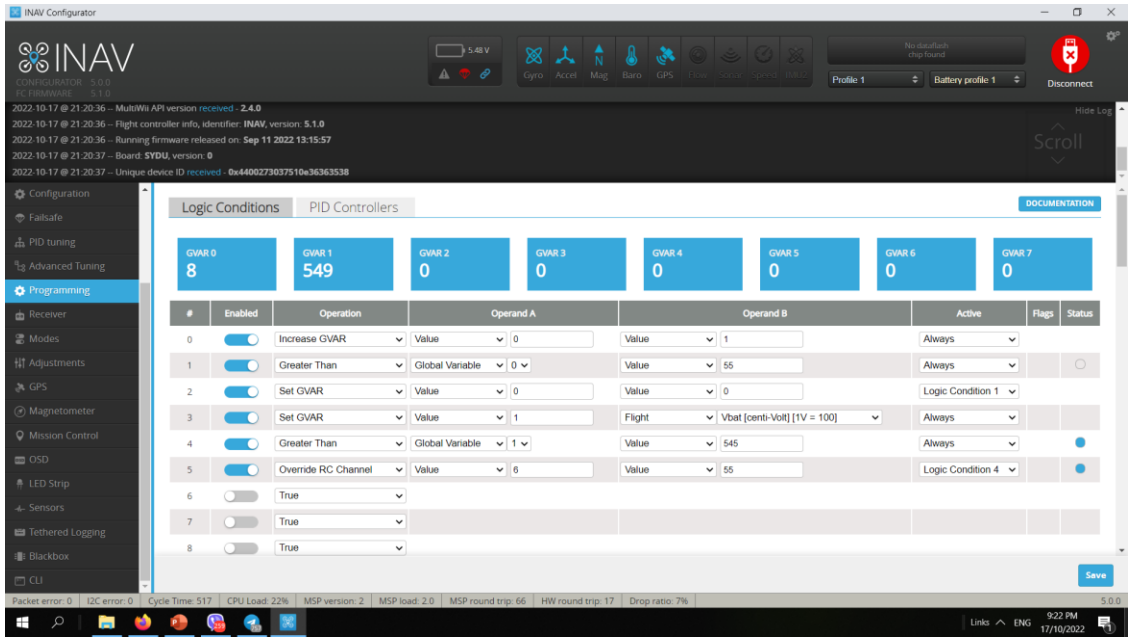
ArduinoCV - this contains the Arduino sketches and adp files of Ardublock (Arduino 1.8.18 compatible)

Tool folder containing Ardublock must be extracted in

this PC/Documents/Arduino/Tools

This sketch is used to convert the Serial communication coming from the CV Tracking and Converting it into PWM or ADC output for the Arduino companion board to read and send appropriate input to Synerduino board to registered

Set up your GCS

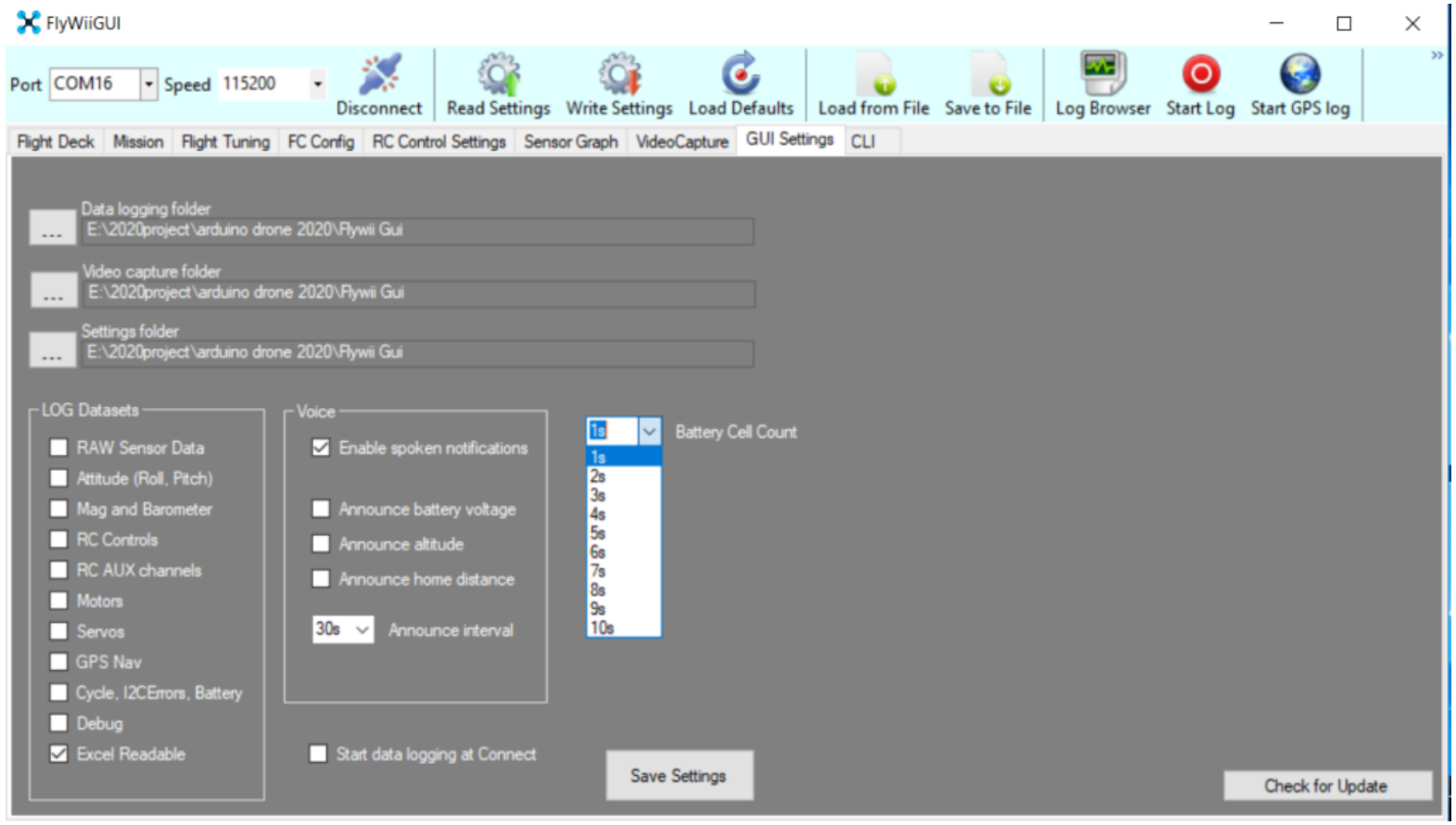


Flywii GUI – Synerduino Arduino INAV Synerduino STM

Synerduino Arduino

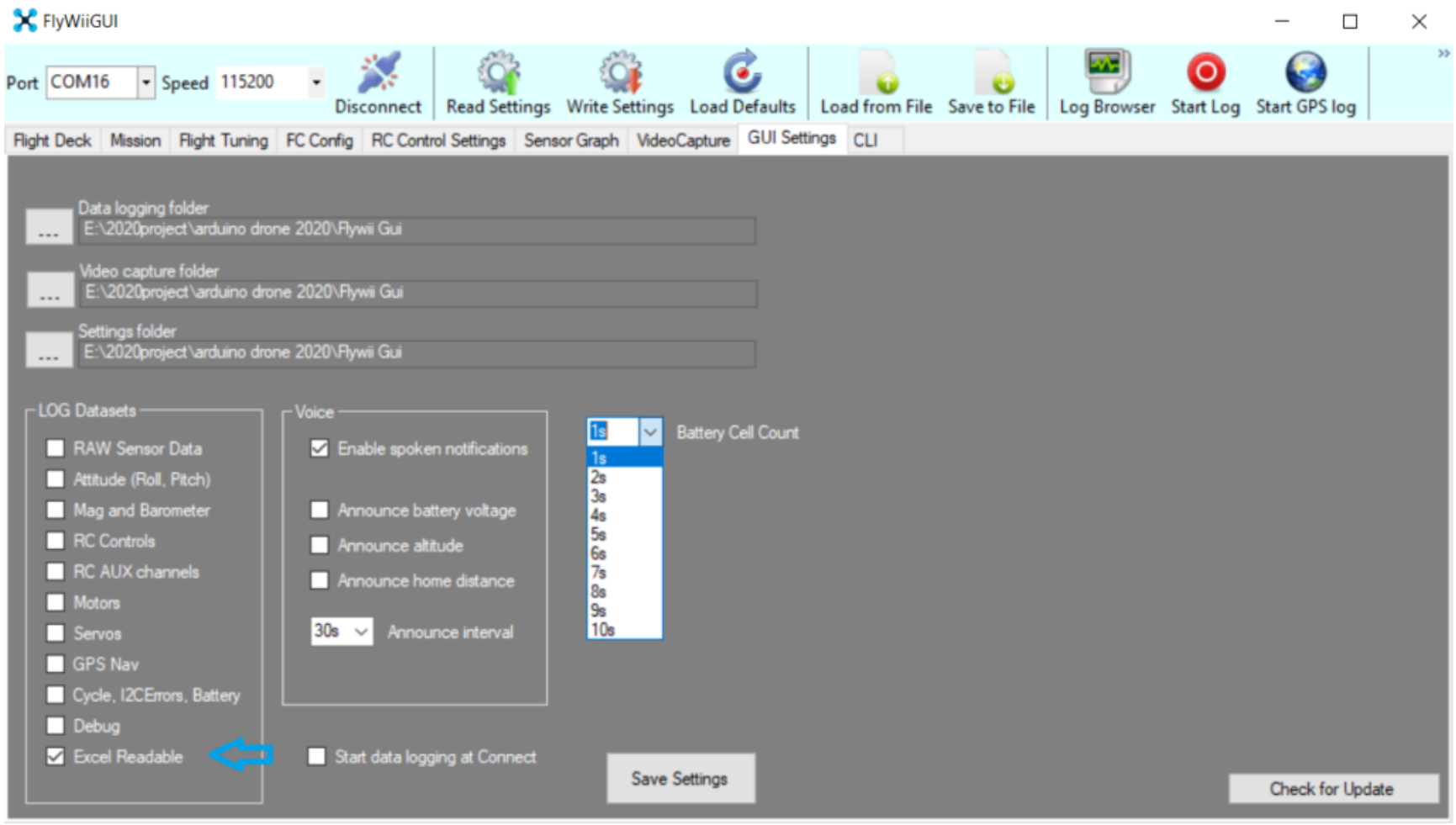
FlyWii GUI Setup

First connect the Drone and go to GUI settings and Change the Battery Cell Count to 1s as we are using it for sensor signal mode.



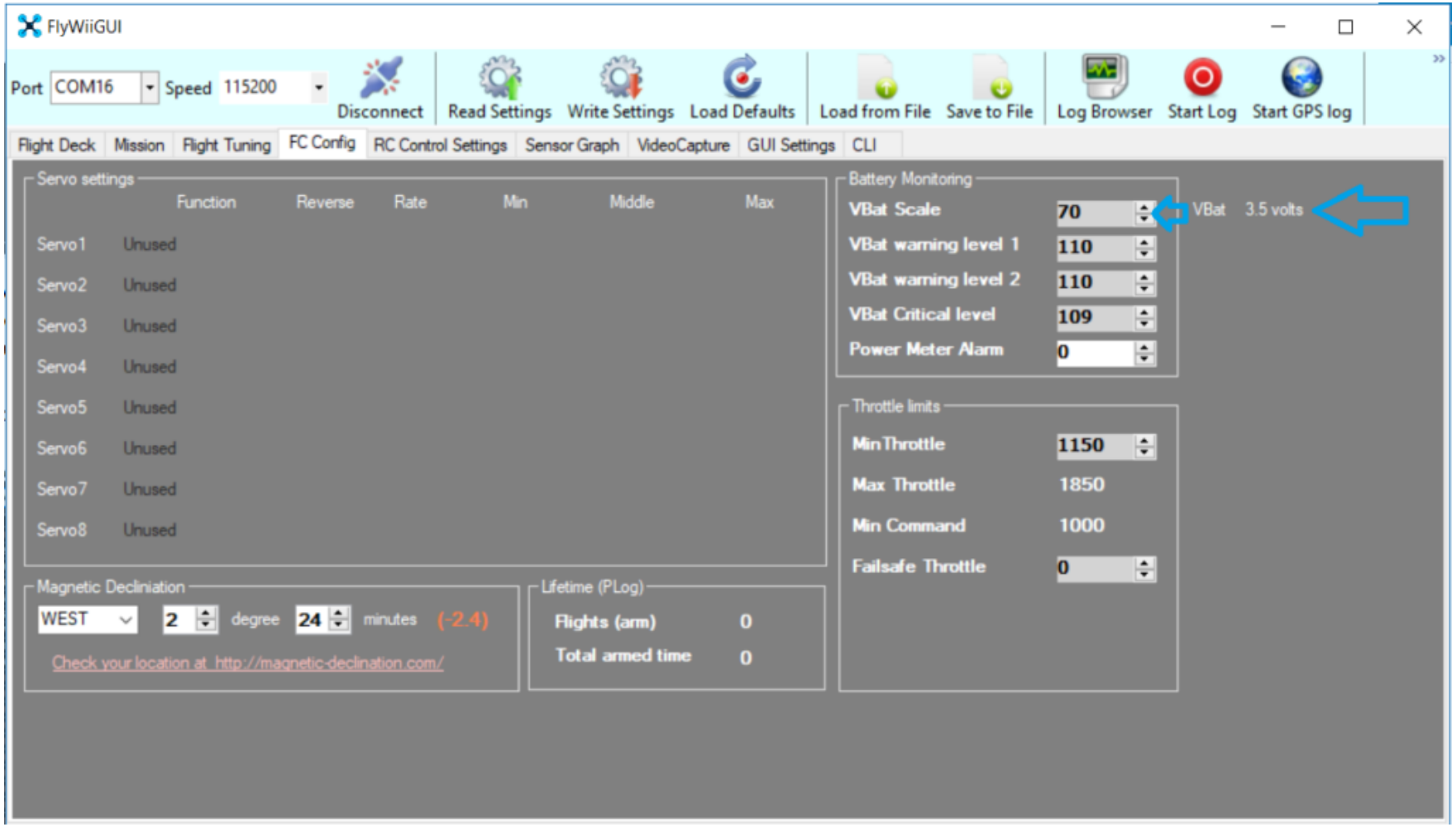
Synerduino Arduino

Available in FlywiiGUI18 select Excel Readable and Save settings it will run when you hit the Start Log button on top or when you setup start logging on connect(warning it will start recording the moment serial connection is on and would incur useless idle data)



Synerduino Arduino

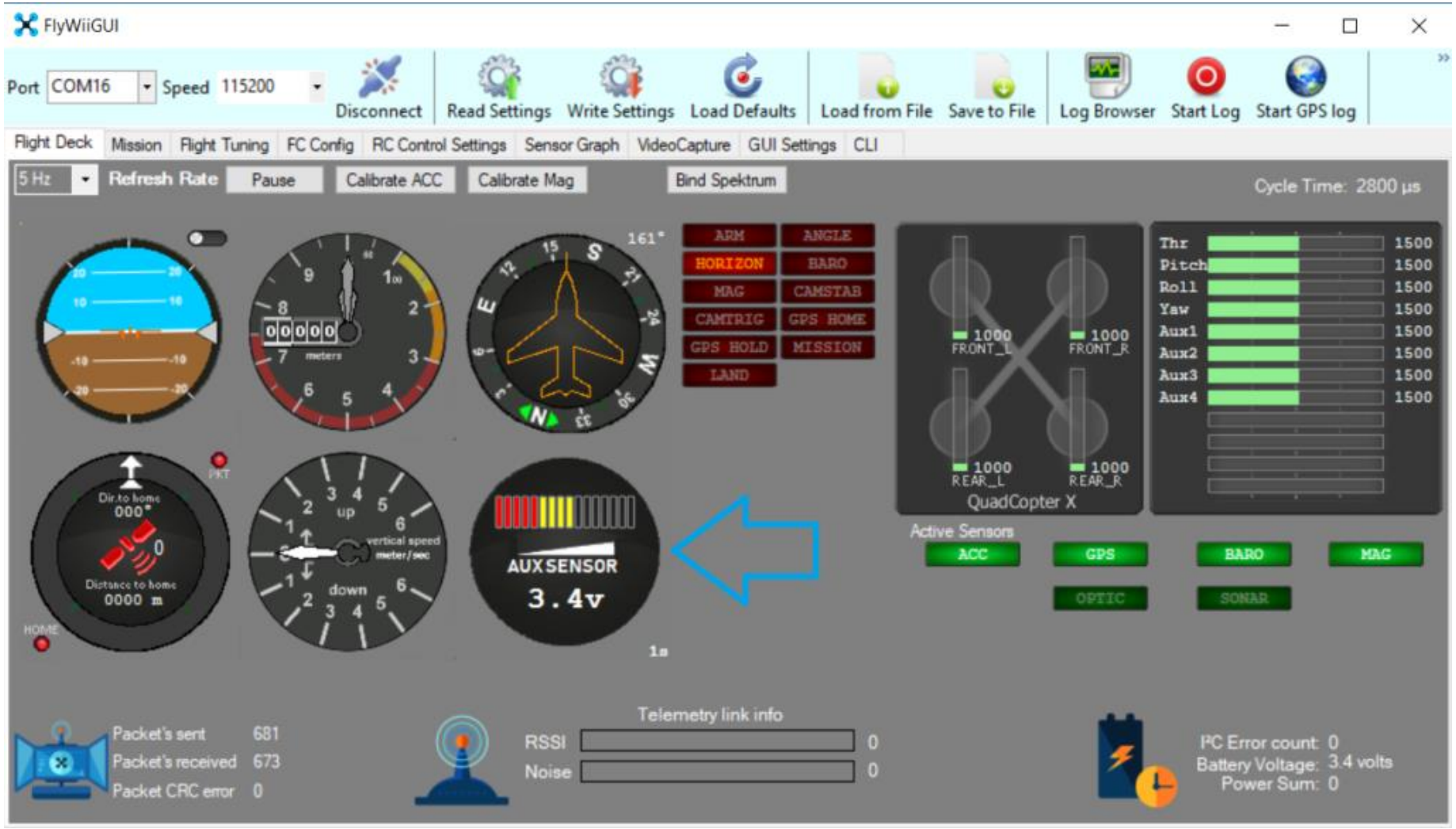
Change the Vbat scale to match the voltage range your sensors is capable of delivering (in my case 70) 0V-4.6V analog **Use the Vbat scale to adjust the output till the number Matches the ppm concentration numerical value of the Control Sensor**



Synerduino Arduino

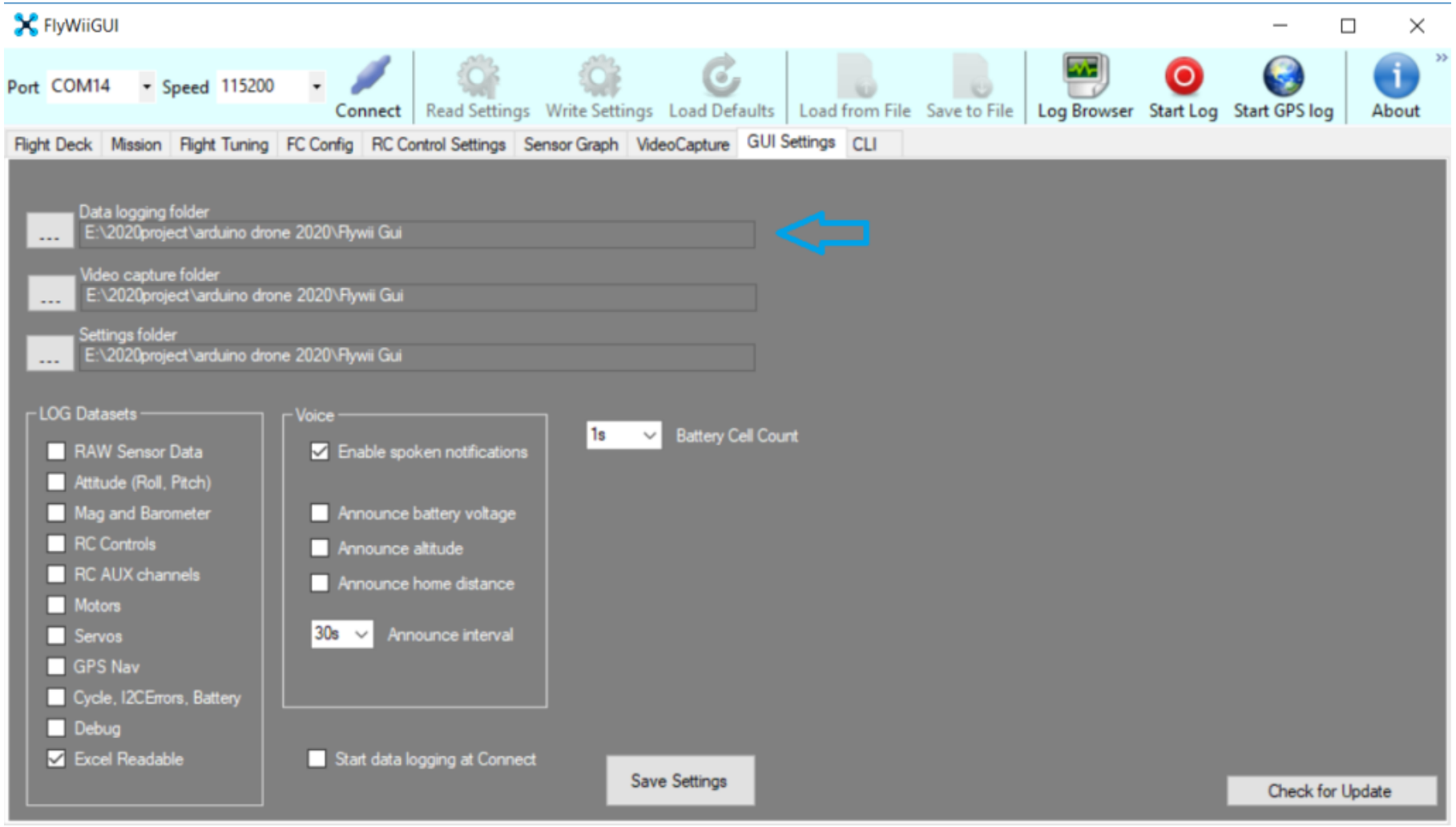
For A0 (Aux input)

Aux sensor should do a read out like this depending if your sensor is resistance base it go up or down



Synerduino Arduino

In GUI Settings you need ro indicate which folder the Log data is to be save to



Synerduino Arduino

Select Log button anytime you want to start and when you want to see the logs select log browser

For Information on how to export as CSV and create Charts and Graphs on Spread sheet

see :

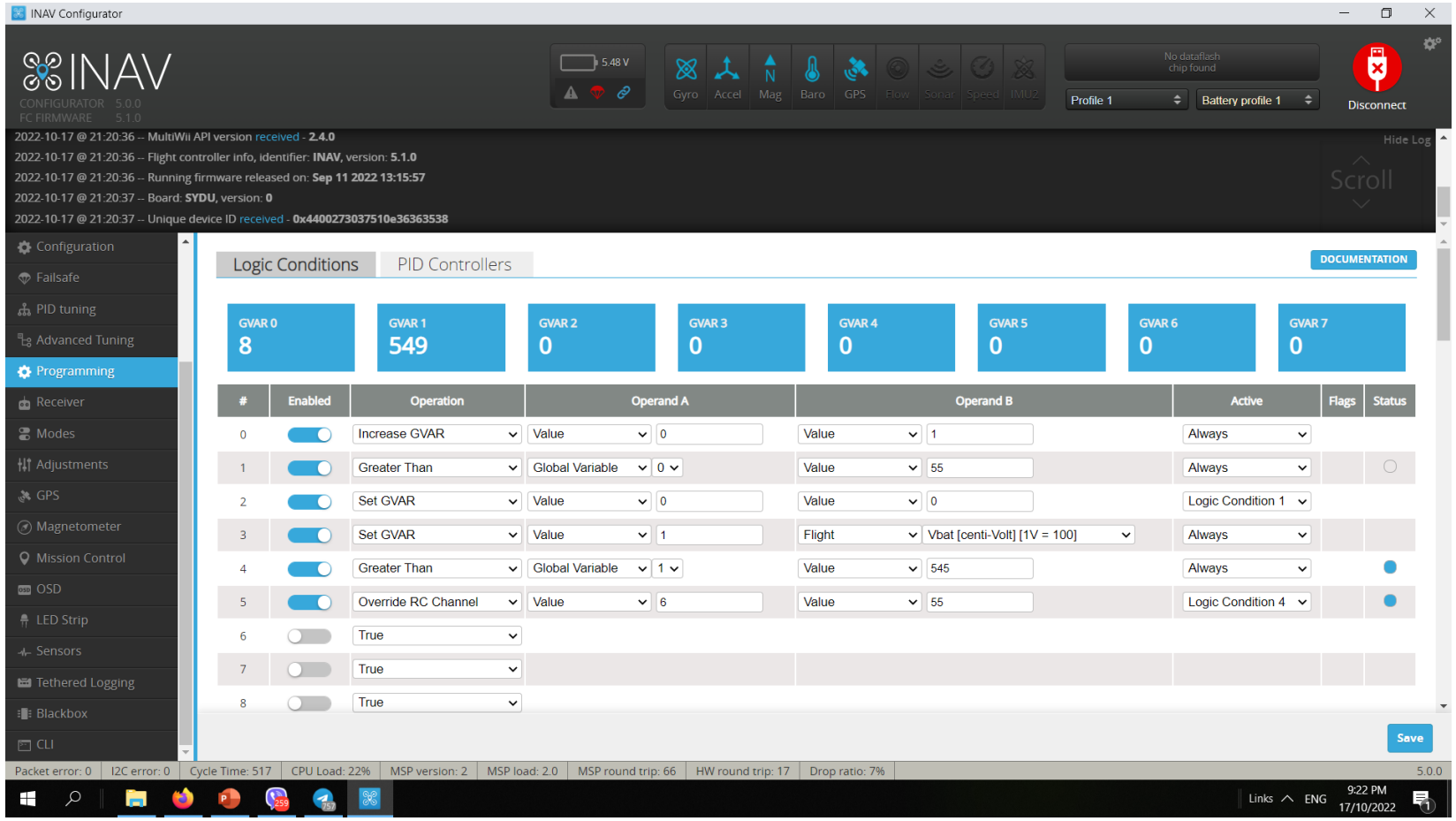
Add on Integration tab Gas Sensors and ADC Data Loggging

Synerduino STM

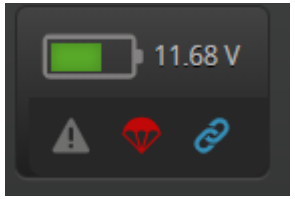
INAV Programming (PLC)

This is the definitive feature of INAV combine with the Synerduino Shield .

This PLC function allows you to program upto 8 GVAR and instructions from timer to sensor conditions to trigger a Flight mode action or control action of your Drone



Data Analytic Intervention Via Companion board



Configuration

Similar Low battery failsafe the DAI Starts by configure your Battery Cell Count ,Current and Voltage scale this way you can adjust to your sensors while calibrating the value output.

But in this case instead of the Battery its ADC data would be inputed from the Companion Controller (Arduino)

This would define your signal point on where to trigger.

ADC V (Battery Voltage Monitoring) could be input 1 D12
ADC I (Battery Current Monitoring) could be input2 D13

You can set your Arduino to input a specific PWM indicating a Data is send

Voltage and Current Sensors

Battery voltage monitoring	
ADC	Voltage Meter Type
Raw	Voltage source to use for alarms and telemetry
450	Voltage Scale
11.64	Battery Voltage
Battery current monitoring	
ADC	Current Meter Type
400	Current Meter Scale
0	Offset in millivolt steps
48.24	Battery Current

Battery Settings

3	Number of cells (0 = auto)
4.25	Maximum cell voltage for cell count detection
3.3	Minimum Cell Voltage
4.2	Maximum Cell Voltage
3.5	Warning Cell Voltage
mAh	Battery Capacity Unit
0	Capacity
	Warning Capacity (remaining %)
	Critical Capacity (remaining %)

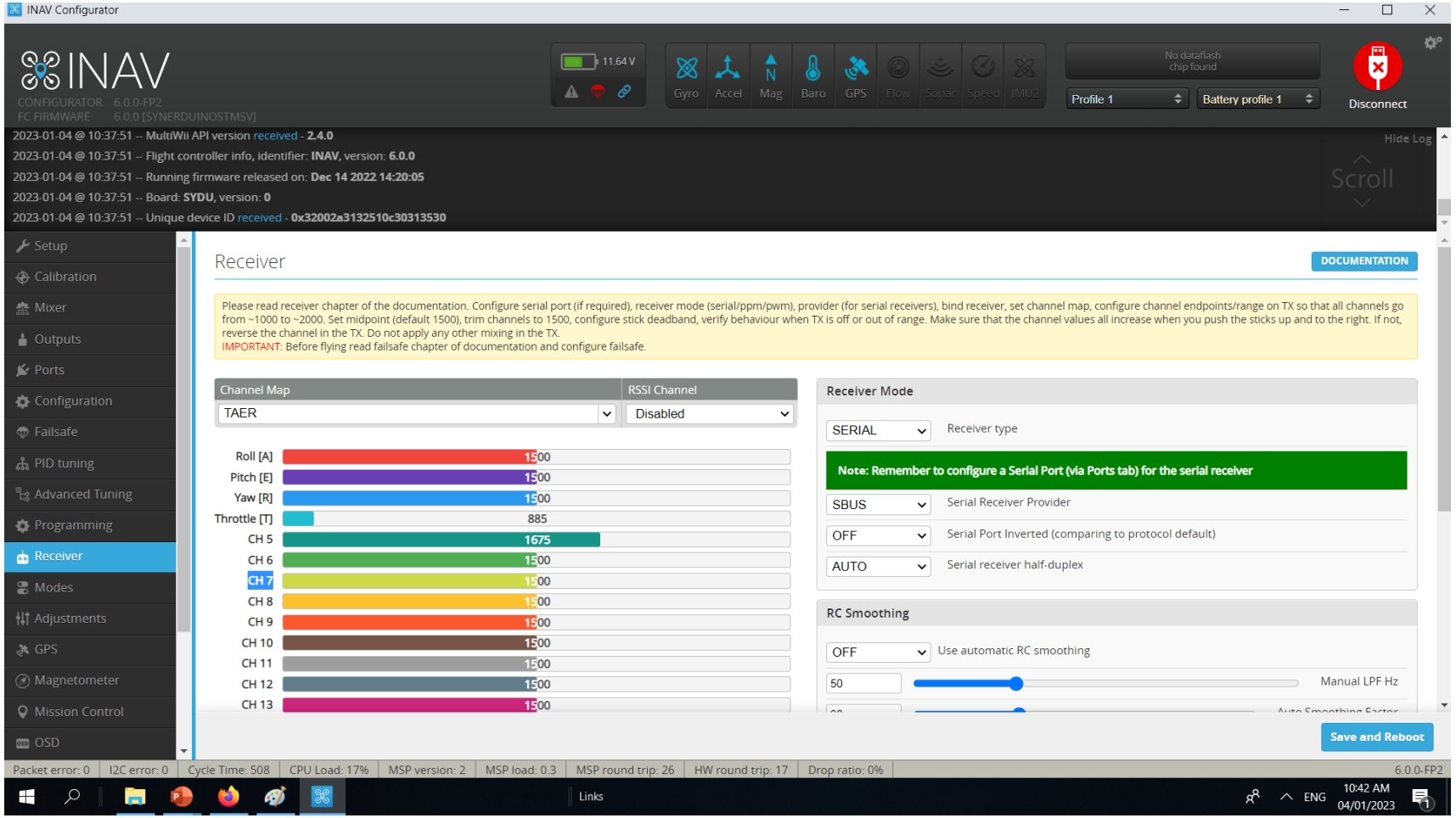
Data Analytic Intervention
SBUS Converter Method

In Sbus Converter Method We Wanted the RC Channel 7 and 8 to be free for the Mode Trigger this would be active when a an PWM value is send by the companion board (Arduino)

The PWM is RC Servo PWM this is useful if you need a properly define value

1. This can be trigger by setting the Channle respond to the ADC input on Programming

2. This can also be directly trigger by the Pwm Converter from the extra free channels Eg. 7 and 8



Data Analytic Intervention by ADC

In this sample we use ADC I (Flight Current) as a sample where we hook up the sensor to

0# set GVAR
Operand A = Value 0
Operand B = Flight Current / Vbat Volt
Active = Always

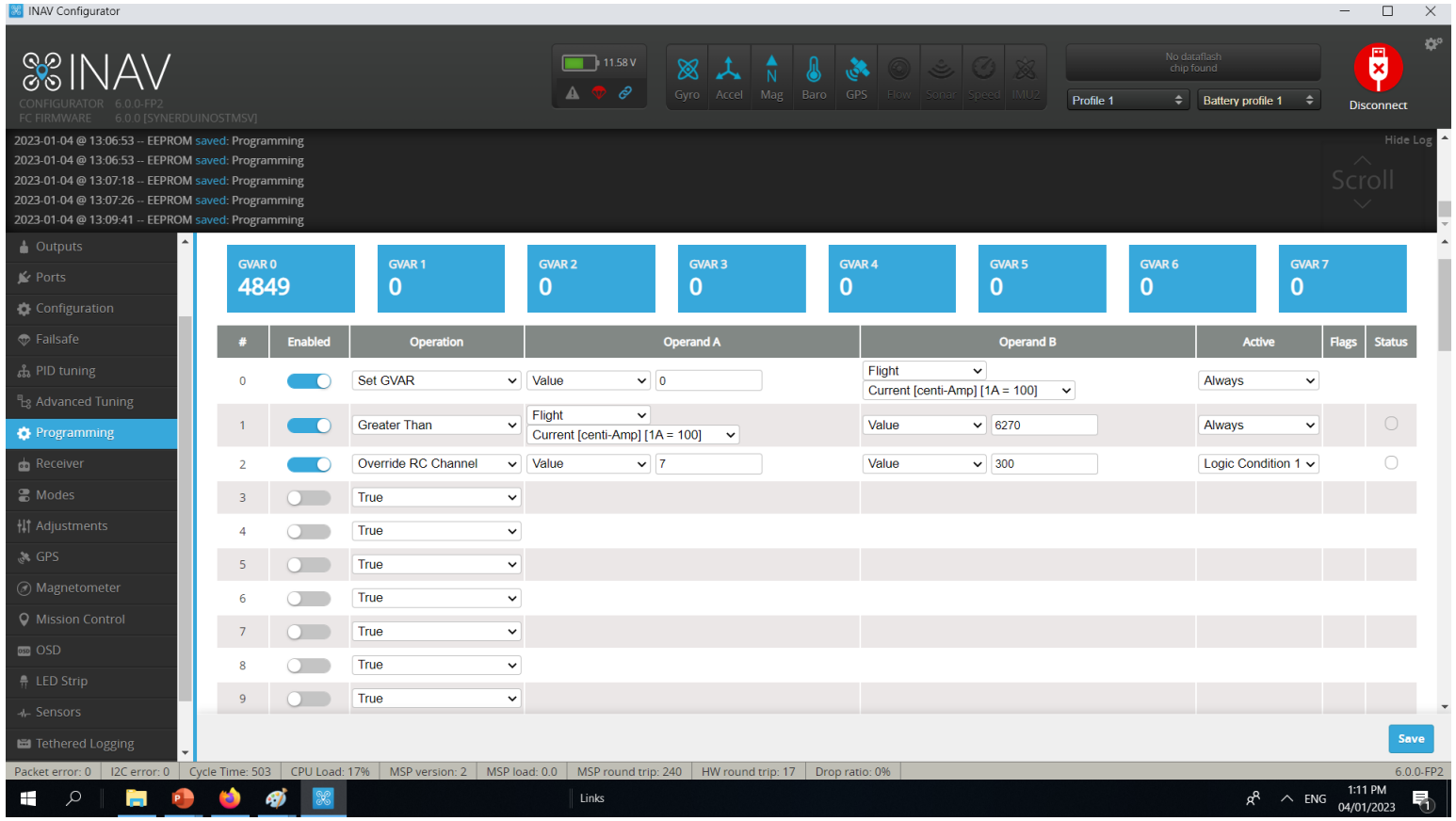
First we need to visualize the Current input in this case GVAR0 4849

1# Greater than
Operand A = Flight Current / Vbat Volt
Operand B = Value 2670
Active = Always

Operand B can be change depending on the Arduino input value want to trigger from

2# Override RC Channel
Operand A = Value 7
Operand B = Value 200
Active = Logic condition 1

Here is where the magic happens when the conditions are met in Logic #1 with an active Status this would trigger the Value 7 (Ch7) to set PWM to 200 overriding the RC input and triggering the servo or Payload



Note: if your RC Channel is occupied by an Receiver channel lets say 7 or 8 and you don't want to override it. You can set the next free channel 9 - 10

Synerduino STM

Data Logging

For Information on how to export as CSV and create Charts and Graphs on Spread sheet

See:

Add on Integration Synerduino STM ADC sensors in the Implementation Tab

Tethered Logging Page