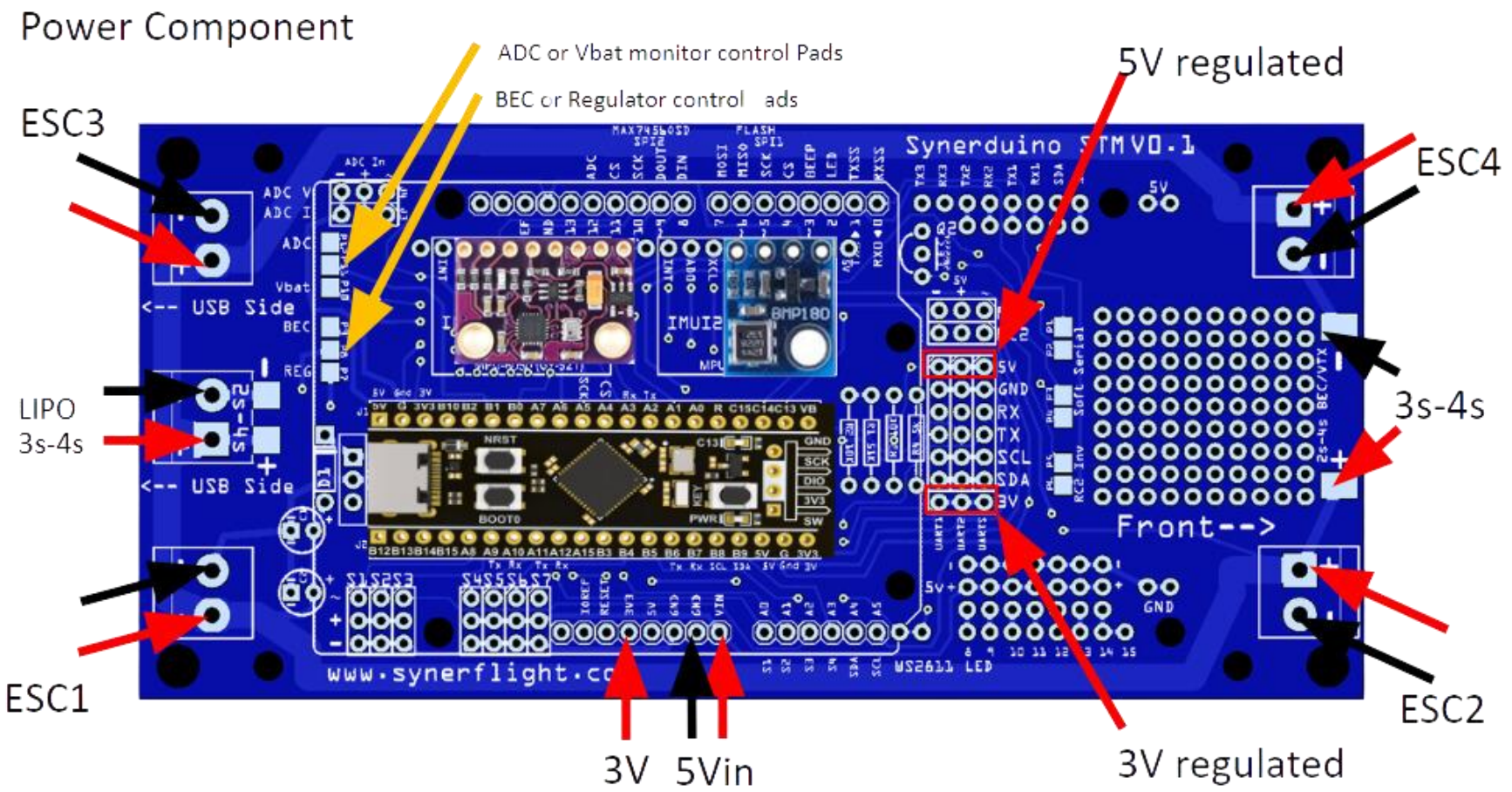
Program Logic

Synerduino STM

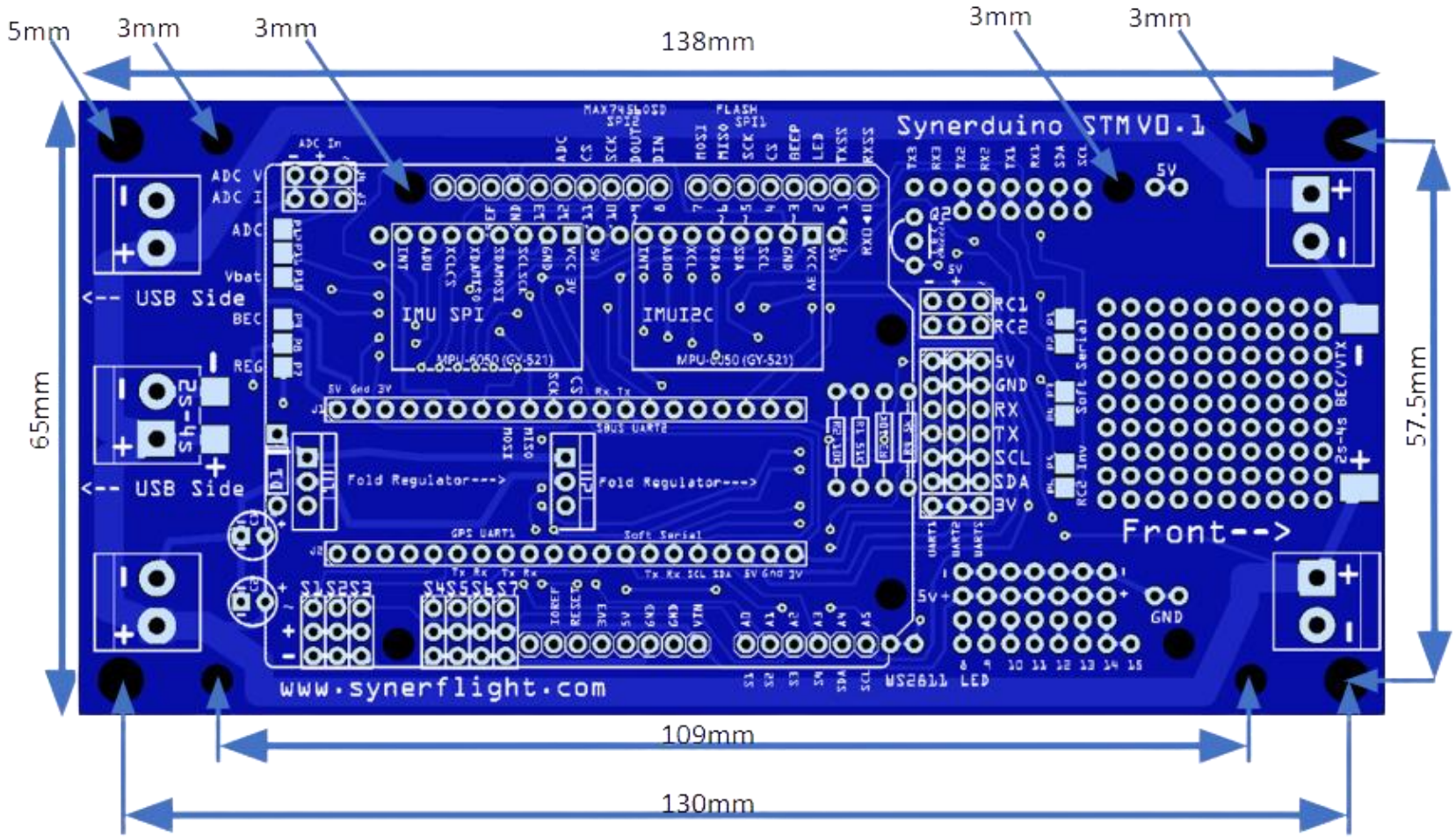
VERSIONS: F405, F411, H743

For more Information:
www.synerflight.com

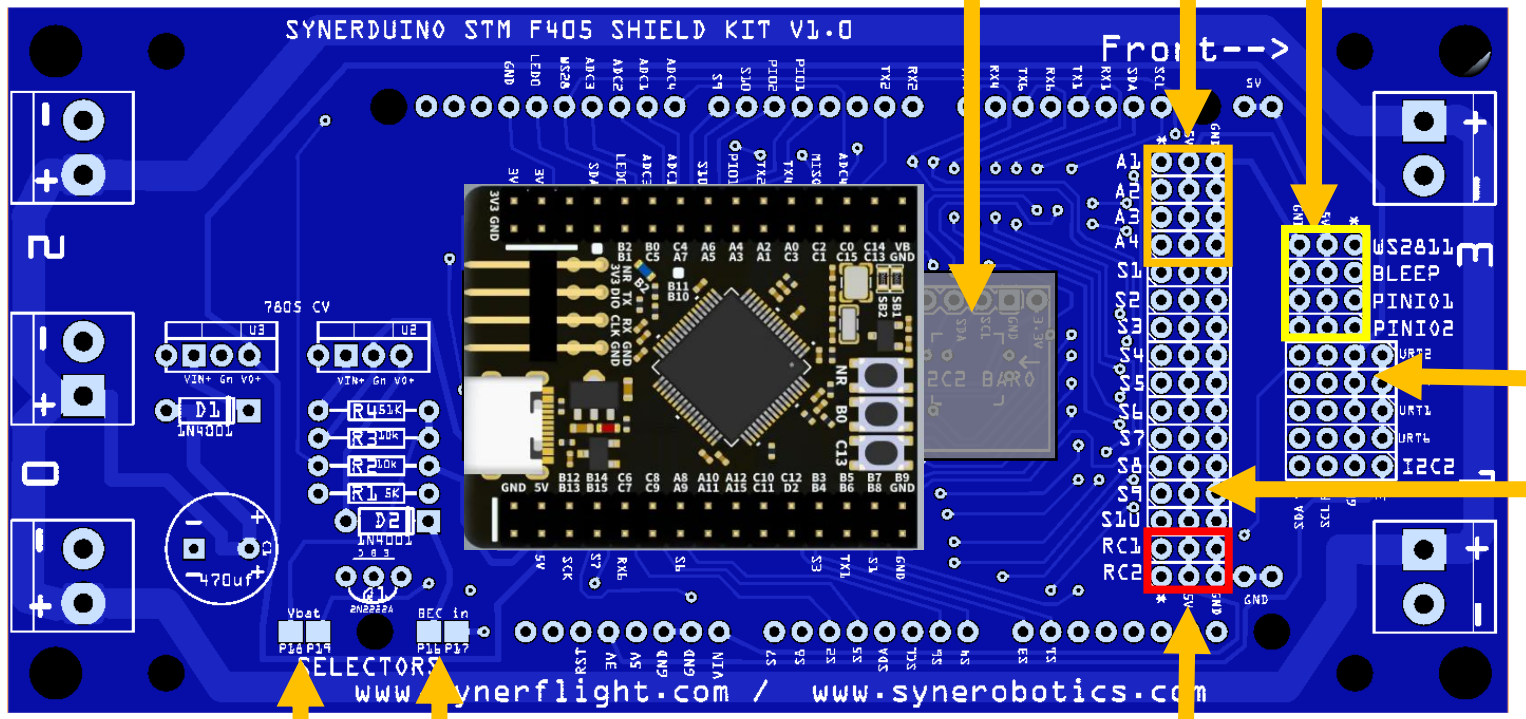
SYNERDUINO STM F411 SHIELD



SYNERDUINO STM F411 SHIELD



SYNERDUINO STM F405 SHIELD



Sensors must be covered with the provided housing glued into place using PVA white glue

ADC input headers

WS2811 & I/O

UART headers

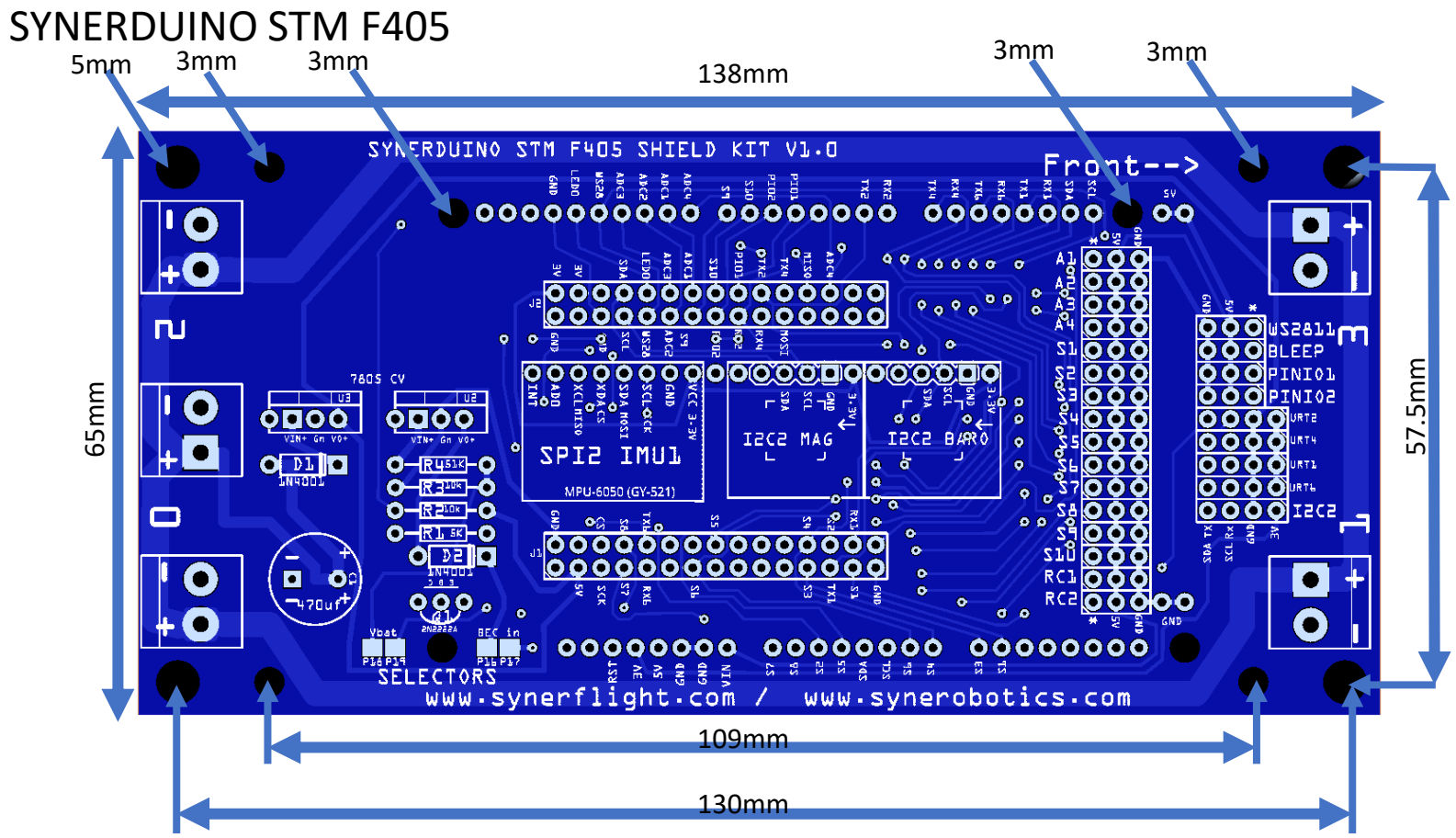
PWM output headers

RC1 & RC2

UBEC /Regulator Selection Pads

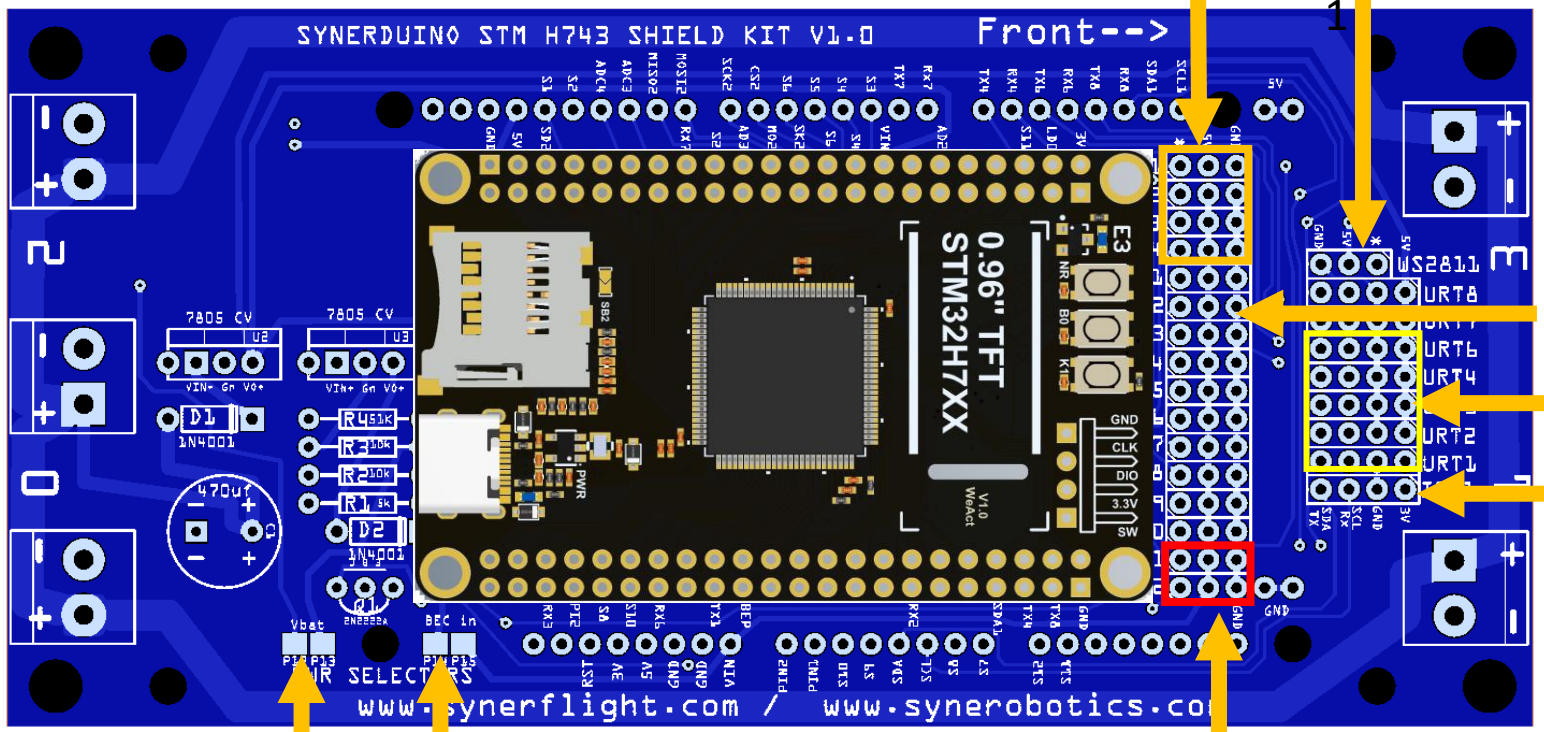
ADC and Battery Monitoring

SYNERDUINO STM F405 SHIELD



SYNERDUINO STM H743 SHIELD

Sensors must be covered with the provided housing glued into place using PVA white glue



ADC input headers

WS281 1

PWM output headers

UART headers

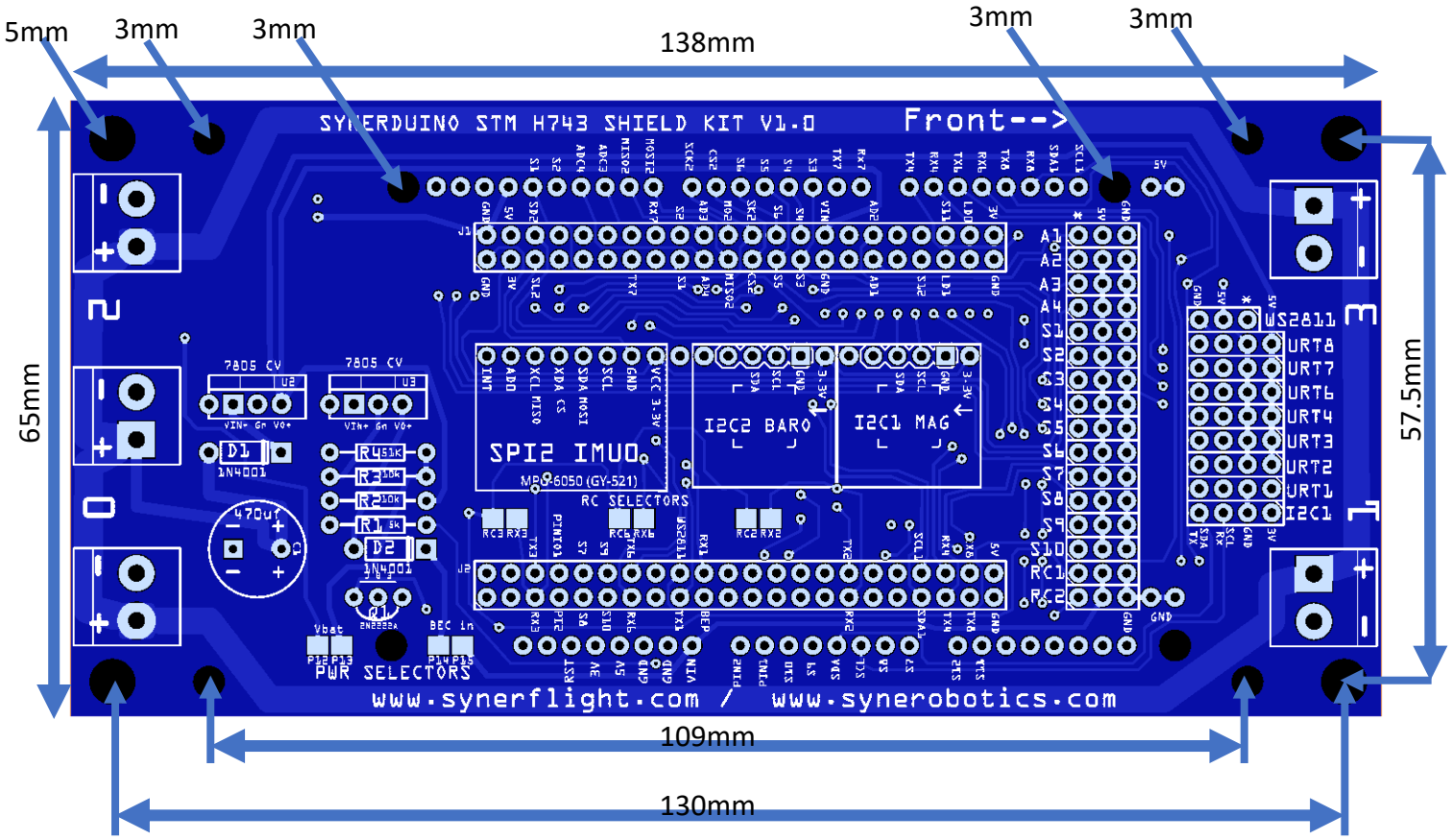
i2C Headers

UBEC /Regulator Selection Pads

RC1 & RC2

ADC and Battery Monitoring

SYNERDUINO STM H743 SHIELD



SYNERDUINO STM

One of the Benefits of Using Synerduino Paired with INAV is it opens a Host of Possibility in Applications

- MIXER SETUP
- PROGRAM LOGIC
- COMPANION BOARD

Here Represented just a Few samples

SETTING UP YOUR DRONE

MIXER for (INAV5-INAV6)

Note INAV5-INAV6 Synerduino STM has Two Firmware with different output arrangement for different vehicle types , (You can get creative in mixing for custom frame designs)

SYNERDUINOSTM.Hex (Default Loaded)

```
(TIM_USE_MC_MOTOR | TIM_USE_FW_MOTOR), // S1
(TIM_USE_MC_MOTOR | TIM_USE_FW_MOTOR), // S2
(TIM_USE_MC_MOTOR | TIM_USE_FW_SERVO), // S3
(TIM_USE_MC_MOTOR | TIM_USE_FW_SERVO), // S4
(TIM_USE_MC_MOTOR | TIM_USE_FW_SERVO), // S5
(TIM_USE_MC_MOTOR | TIM_USE_FW_SERVO), // S6
(TIM_USE_MC_SERVO | TIM_USE_FW_SERVO), // S7
```

Vehicle Preset Mix

```
QUAD X
QUAD +
QUAD A-Tail
Y4
Y6
Hex X
Hex +
Hex H
```

```
FlyingWing
Airplane
Airplane No Rudder
Airplane V-Tail 2 Aileron Servo
Airplane V-Tail 1 Aileron Servo
Other Stuff
```

SYNERDUINOSTMSV.Hex

```
(TIM_USE_MC_MOTOR | TIM_USE_FW_MOTOR), // S1
(TIM_USE_MC_MOTOR | TIM_USE_FW_MOTOR), // S2
(TIM_USE_MC_MOTOR | TIM_USE_FW_MOTOR), // S3
(TIM_USE_MC_MOTOR | TIM_USE_FW_MOTOR), // S4
(TIM_USE_MC_SERVO | TIM_USE_FW_SERVO), // S5
(TIM_USE_MC_SERVO | TIM_USE_FW_SERVO), // S6
(TIM_USE_MC_SERVO | TIM_USE_FW_SERVO), // S7
```

Vehicle Preset Mix

```
Quad X W/ Gimbal
Quad + W/ Gimbal
Single Copter
Bi-Copter
Tricopter
Rover
Boat
Camera Gimbal
```

```
FlyingWing Differential thrust
Airplane Differential Thrust
Airplane V-Tail Differential Thrust
Other Stuff
```

SETTING UP YOUR DRONE

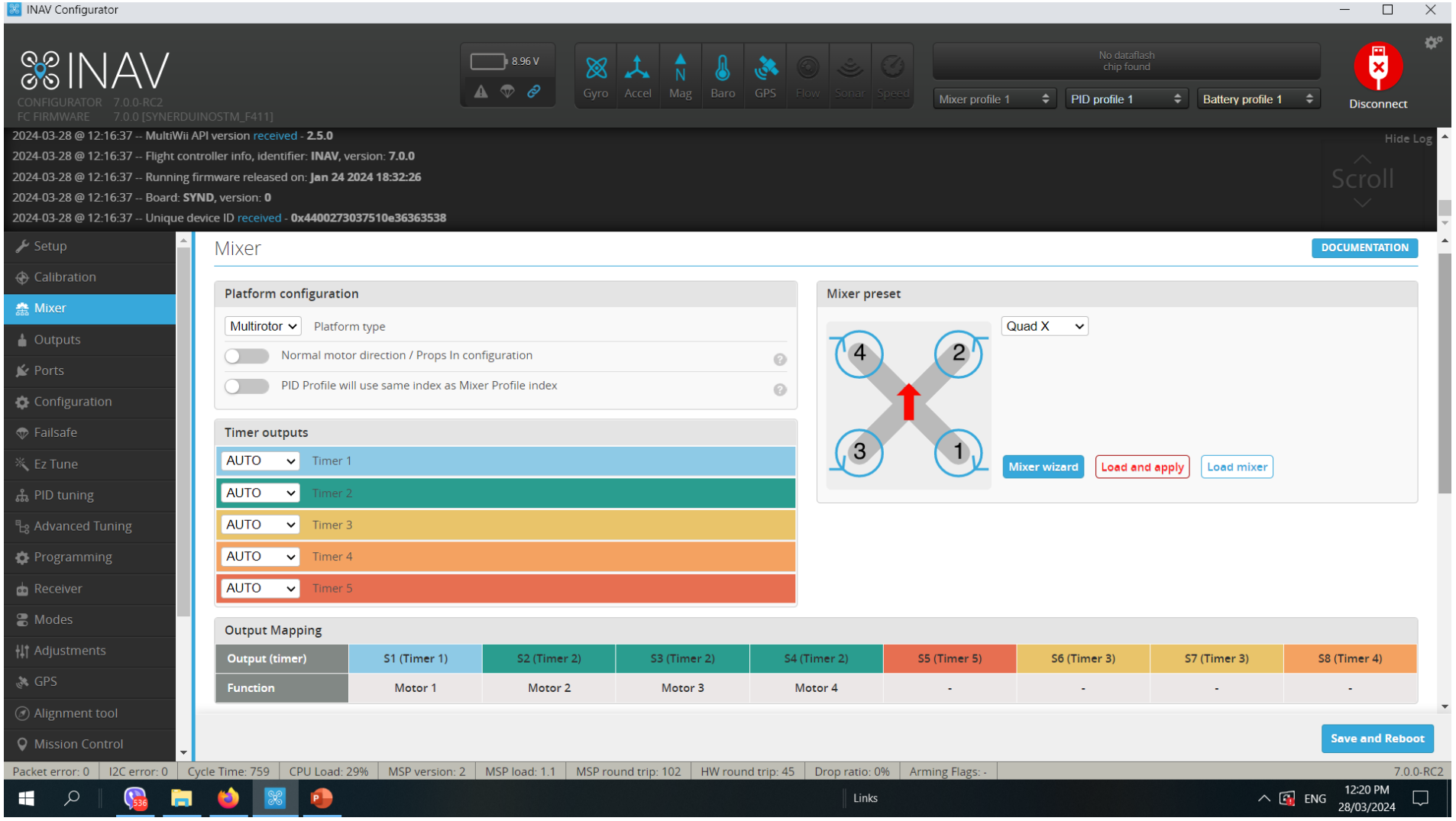
MIXER (INAV 7-INAV 8)

Airframe or Vehicle time Preset and mix selection

Load and apply when selected then Save Reboot

Mixing is now color coded to timer availability

The Timer is Automatically assign to the corresponding PWM Pins as needed



This allows you to assign motor and servo function to your custom drone frame or payload requirement this eliminates the need to recompile a new firmware for custom frame types

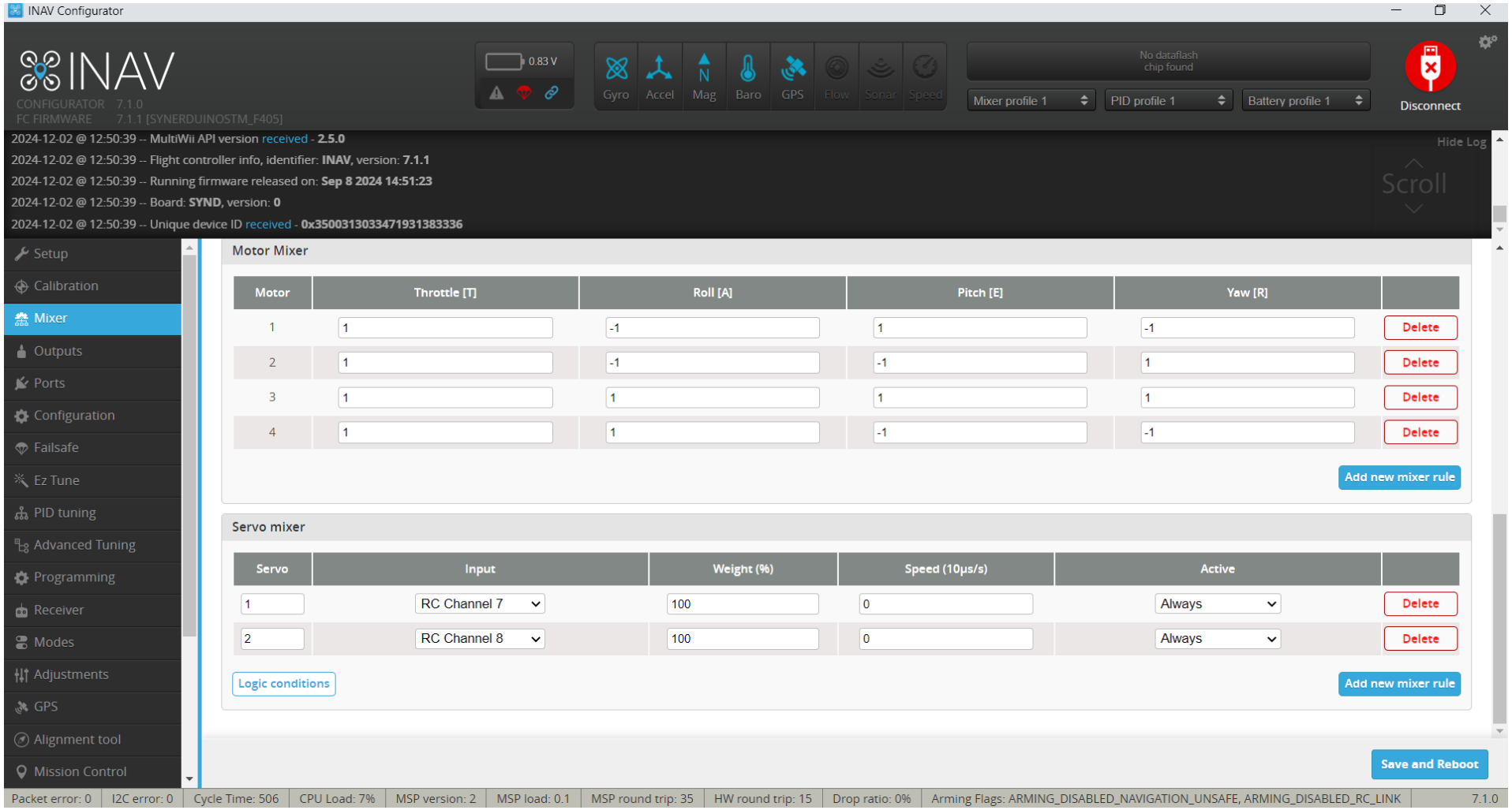
SETTING UP YOUR DRONE

Motor Mixer controls ESC control Elements of vehicle or Payload

Here you can add motors and Assign it Mixing

Servo Mixer control Servo elements of Vehicle or Payload

Here is where you can Add the Input assignment of the Servo



SERVO MIXING

Motor Mixing

A motor mixing rule is needed for each motor. Each rule defines weights that determine how the motor it applies to will change its speed relative to the requested throttle and flight dynamics: roll rate, pitch rate and yaw rate. The heigher a weight the more the input will have an impact on the speed of the motor. Refer to the following table for the meaning of each weight.

Weight	Definition
THROTTLE	Speed of the motor relative to throttle. Range [0.0, 1.0]. A motor with a weight of 0.5 will receive a command that will half of a motor with a 1.0 weight
ROLL	Indicates how much roll authority this motor imparts to the roll rate of the aircraft. Range [-1.0, 1.0]. For fixed wing models this is usually set to 0. A positive value means that the motor needs to accelerate for a positive roll rate request (rolling right). A negative value means that the motor needs to decelerate.
PITCH	Indicates how much pitch authority this motor imparts to the pitch rate of the aircraft. Range [-1.0, 1.0]. For fixed wing models this is usually set to 0. A positive value means that the motor needs to accelerate for a positive pitch rate request (pitching down). A negative value means that the motor needs to decelerate.
YAW	Indicates how much yaw authority this motor imparts to the yaw rate of the aircraft. Range [-1.0, 1.0]. For fixed wing models with more than one motor this weight can be used to setup differential thrust. For fixed wing models with only one motor this is usually set to 0. A positive value means that the motor needs to accelerate for a positive yaw rate request (clockwise yaw seen from the top of the model). A negative value means that the motor needs to decelerate

SERVO MIXING

Servo Mixing

At least one servo mixing rule is needed for each servo. Each rule defines how a servo will move relative to a specific input like a RC channel, or a requested flight dynamics rate or position from the flight controller.

Each servo mixing rule has the following parameters:

•Servo index: defines which servo the rule will apply to. The absolute value of the index is not important, what matters is only the relative difference between the used indexes. The rule with the smaller servo index will apply to the first servo, the next higher servo index to the second servo, etc. More than one rule can use the same servo index. The output of the rules with the same servo index are added together to give the final output for the specified servo.

•Input: the input for the mixing rule, see a summary of the input types table bellow.

•Weight: percentage of the input to forward to the servo. Range [-1000, 1000]. Mixing rule output = input * weight. If the output of a set of mixing rules is lower/higher than the defined servo min/max the output is clipped (the servo will never travel farther than the set min/max).

•Speed: maximum rate of change of the mixing rule output. Used to limit the servo speed. 1 corresponds to maximum 10µs/s output rate of change. Set to 0 for no speed limit. For example: 10 = full sweep (1000 to 2000) in 10s, 100 = full sweep in 1s.

CLI input ID	Mixer input	Description
0	Stabilised ROLL	Roll command from the flight controller. Depends on the selected flight mode(s)
1	Stabilised PITCH	Pitch command from the flight controller. Depends on the selected flight mode(s)
2	Stabilised YAW	Yaw command from the flight controller. Depends on the selected flight mode(s)
3	Stabilised THROTTLE	Throttle command from the flight controller. Depends on the selected flight mode(s)
4	RC ROLL	Raw roll RC channel
5	RC PITCH	Raw pitch RC channel
6	RC YAW	Raw yaw RC channel
7	RC THROTTLE	Raw throttle RC channel
8	RC channel 5	Raw RC channel 5
9	RC channel 6	Raw RC channel 6
10	RC channel 7	Raw RC channel 7
11	RC channel 8	Raw RC channel 8
12	GIMBAL PITCH	Scaled pitch attitude of the aircraft [-90°, 90°] => [-500, 500]
13	GIMBAL ROLL	Scaled roll attitude of the aircraft [-180°, 180°] => [-500, 500]
14	FEATURE FLAPS	This input value is equal to the flaperon_throw_offset setting when the FLAPERON flight mode is enabled, 0 otherwise
15	RC channel 9	Raw RC channel 9
16	RC channel 10	Raw RC channel 10
17	RC channel 11	Raw RC channel 11
18	RC channel 12	Raw RC channel 12
19	RC channel 13	Raw RC channel 13
20	RC channel 14	Raw RC channel 14
21	RC channel 15	Raw RC channel 15
22	RC channel 16	Raw RC channel 16
23	Stabilized ROLL+	Clipped between 0 and 1000
24	Stabilized ROLL-	Clipped between -1000 and 0
25	Stabilized PITCH+	Clipped between 0 and 1000
26	Stabilized PITCH-	Clipped between -1000 and 0
27	Stabilized YAW+	Clipped between 0 and 1000
28	Stabilized YAW-	Clipped between -1000 and 0
29	MAX	Constant value of 500

SYNERDUINO STM

One of the highlighted feature both hardware and software is its flexibility into adding new function to a Drone the combination of Synerduino hardware and INAV software makes this possible

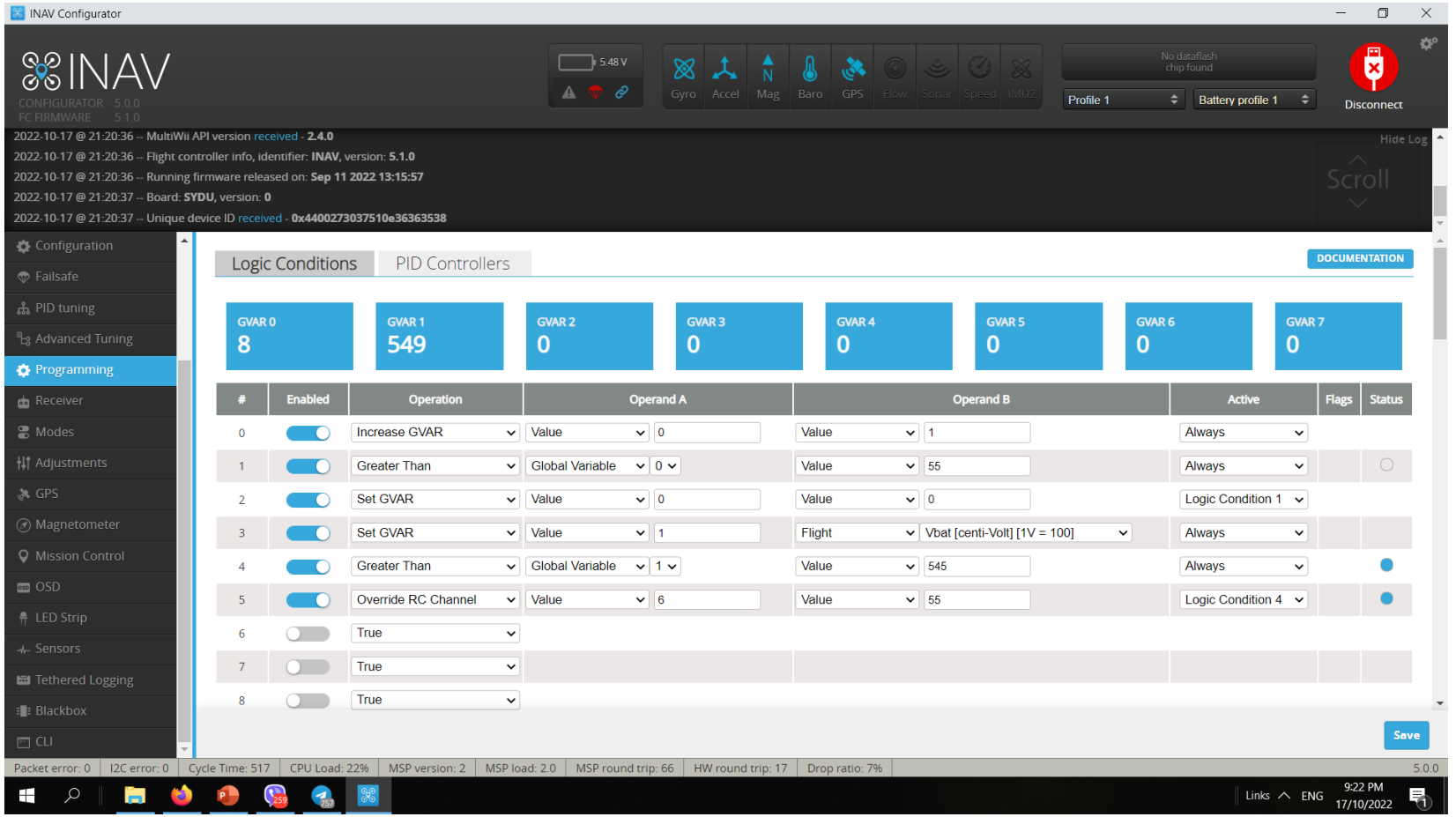
Programmable Logic and Data Analytic Intervention allows you to add block like behavior control to your drone to respond to Environment , Sensor inputs and Conditions.

Operations - is the Logic function to conduct

Operand - A and B are the individual values and directors

Active – this has 2 option being always live or only live when the condition are met on the assign Logic conditions

Here Represented just a Few samples



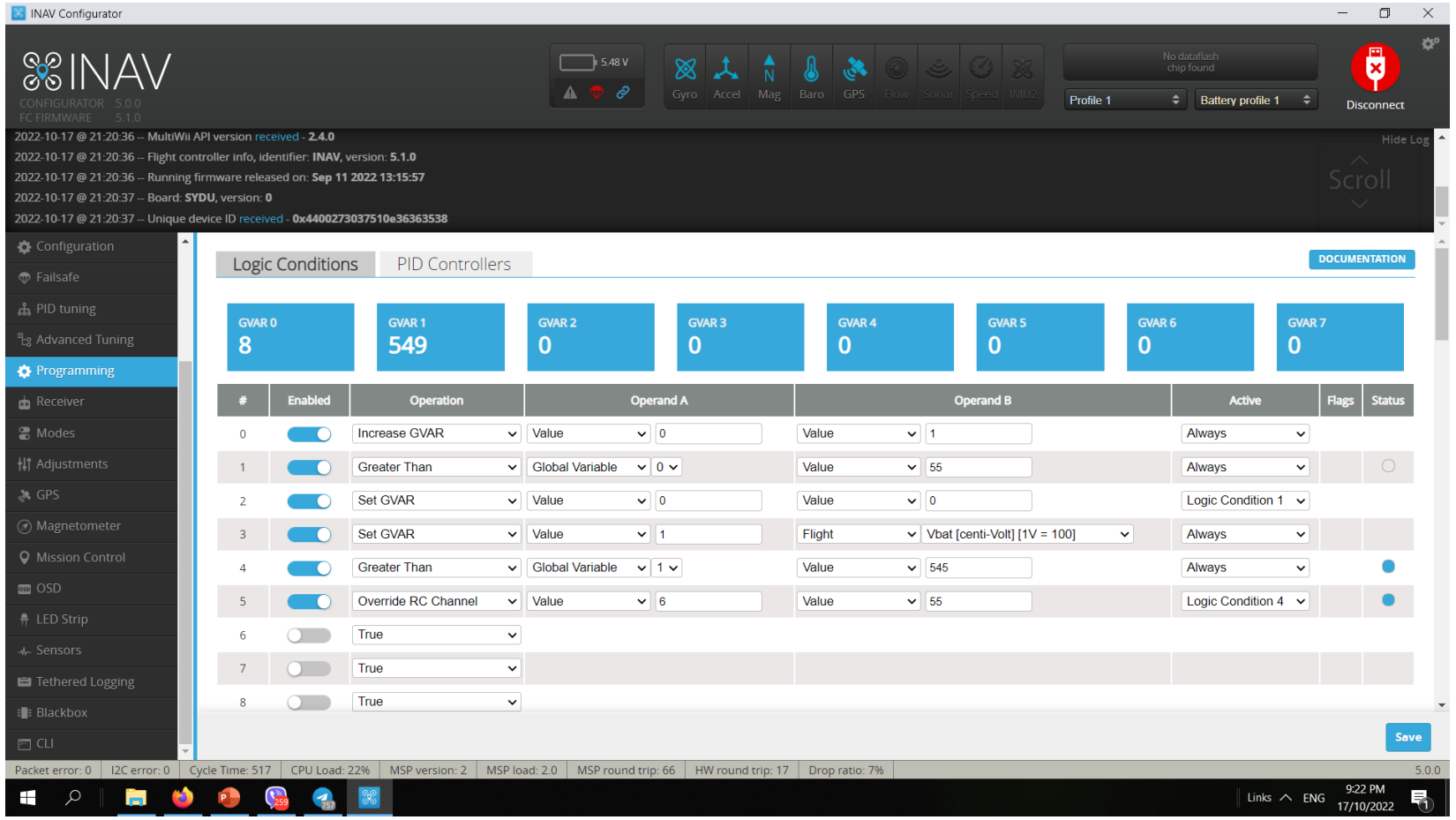
PROGRAMMING

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

Is the Combination of ADC sensor with Programmable Logic functionality.

This PLC function allows you to program upto 8 GVAR and 64 instructions line from timer to sensor conditions to trigger a Flight mode action or control action of your Drone

And in certain application it skips the need of an onboard companion computer reducing the vehicle platform size



This feature can be use in multiple applications . Selective payload deployment , hardware trigger , Gimbal stabilization , failsafe ,geofence , data respond and more

OPERATIONS

OPERATION ID	NAME	NOTES
0	TRUE	Always evaluates as true
1	EQUAL	Evaluates false if false or 0
2	GREATER_THAN	true if Operand A is a higher value than Operand B
3	LOWER_THAN	true if Operand A is a lower value than Operand B
4	LOW	true if <1333
5	MID	true if >=1333 and <=1666
6	HIGH	true if >1666
7	AND	true if Operand A and Operand B are the same value or both true
8	OR	true if Operand A and/or OperandB is true
9	XOR	true if Operand A or Operand B is true, but not both
10	NAND	false if Operand A and Operand B are both true
11	NOR	true if Operand A and Operand B are both false
12	NOT	The boolean opposite to Operand A
13	STICKY	Operand A is activation operator, Operand B is deactivation operator. After activation, operator will return true until Operand B is evaluated as true
14	ADD	Add Operand A to Operand B and returns the result
15	SUB	Substract Operand B from Operand A and returns the result
16	MUL	Multiply Operand A by Operand B and returns the result
17	DIV	Divide Operand A by Operand B and returns the result
18	GVAR SET	Store value from Operand B into the Global Variable addressed by Operand B. Bear in mind, that operand Global Variable means: Value stored in Global Variable of an index! To store in GVAR 1 use Value 1 not Global Variable 1
19	GVAR INC	Increase the GVAR indexed by Operand A with value from Operand B
20	GVAR DEC	Decrease the GVAR indexed by Operand A with value from Operand B
21	IO PORT SET	Set I2C IO Expander pin Operand A to value of Operand B. Operand A accepts values 0-7 and Operand B accepts 0 and 1
22	OVERRIDE_ARMING_SAFETY	Allows to arm on any angle even without GPS fix
23	OVERRIDE_THROTTLE_SCALE	Override throttle scale to the value defined by operand. Operand type 0 and value 50 means throttle will be scaled by 50%.

OPERATIONS

OPERATION ID	NAME	NOTES
24	SWAP_ROLL_YAW	basically, when activated, yaw stick will control roll and roll stick will control yaw. Required for tail-sitters VTOL during vertical-horizonral transition when body frame changes
25	SET_VTX_POWER_LEVEL	Sets VTX power level. Accepted values are 0-3 for SmartAudio and 0-4 for Tramp protocol
26	INVERT_ROLL	Inverts ROLL axis input for PID/PIFF controller
27	INVERT_PITCH	Inverts PITCH axis input for PID/PIFF controller
28	INVERT_YAW	Inverts YAW axis input for PID/PIFF controller
29	OVERRIDE_THROTTLE	Override throttle value that is fed to the motors by mixer. Operand is scaled in us. 1000 means throttle cut, 1500 means half throttle
30	SET_VTX_BAND	Sets VTX band. Accepted values are 1-5
31	SET_VTX_CHANNEL	Sets VTX channel. Accepted values are 1-8
32	SET_OSD_LAYOUT	Sets OSD layout. Accepted values are 0-3
33	SIN	Computes SIN of Operand A value in degrees. Output is multiplied by Operand B value. If Operand B is 0, result is multiplied by 500
34	COS	Computes COS of Operand A value in degrees. Output is multiplied by Operand B value. If Operand B is 0, result is multiplied by 500
35	TAN	Computes TAN of Operand A value in degrees. Output is multiplied by Operand B value. If Operand B is 0, result is multiplied by 500
36	MAP_INPUT	Scales Operand A from [0 : Operand B] to [0 : 1000]. Note: input will be constrained and then scaled
37	MAP_OUTPUT	Scales Operand A from [0 : 1000] to [0 : Operand B]. Note: input will be constrained and then scaled
38	RC_CHANNEL_OVERRIDE	Overrides channel set by Operand A to value of Operand B
39	SET_HEADING_TARGET	Sets heading-hold target to Operand A, in degrees. Value wraps-around.
40	MOD	Divide Operand A by Operand B and returns the remainder
41	LOITER_RADIUS_OVERRIDE	Sets the loiter radius to Operand A [0 : 100000] in cm. If the value is lower than the loiter radius set in the Advanced Tuning, that will be used.
42	SET_PROFILE	Sets the active config profile (PIDFF/Rates/Filters/etc) to Operand A. Operand A must be a valid profile number, currently from 1 to 3. If not, the profile will not change
43	MIN	Finds the lowest value of Operand A and Operand B
44	MAX	Finds the highest value of Operand A and Operand B
45	FLIGHT_AXIS_ANGLE_OVERRIDE	Sets the target attitude angle for axis. In other words, when active, it enforces Angle mode (Heading Hold for Yaw) on this axis (Angle mode does not have to be active). Operand A defines the axis: 0 - Roll, 1 - Pitch, 2 - Yaw. Operand B defines the angle in degrees
46	FLIGHT_AXIS_RATE_OVERRIDE	Sets the target rate (rotation speed) for axis. Operand A defines the axis: 0 - Roll, 1 - Pitch, 2 - Yaw. Operand B defines the rate in degrees per second

OPERANDS

OPERAND TYPE	NAME	NOTES
0	VALUE	Value derived from value field
1	GET_RC_CHANNEL	value points to RC channel number, indexed from 1
2	FLIGHT	value points to flight parameter table
3	FLIGHT_MODE	value points to flight modes table
4	LC	value points to other logic condition ID
5	GVAR	Value stored in Global Variable indexed by value. GVAR 1 means: value in GVAR 1
5	PID	Output of a Programming PID indexed by value. PID 1 means: value in PID 1

FLIGHT

Operand Value	Name	Notes
0	ARM_TIMER	in seconds
1	HOME_DISTANCE	in meters
2	TRIP_DISTANCE	in meters
3	RSSI	
4	VBAT	in Volts * 100, eg. 12.1V is 1210
5	CELL_VOLTAGE	in Volts * 100, eg. 12.1V is 1210
6	CURRENT	in Amps * 100, eg. 9A is 900
7	MAH_DRAWN	in mAh
8	GPS_SATS	
9	GROUD_SPEED	in cm/s
10	3D_SPEED	in cm/s
11	AIR_SPEED	in cm/s
12	ALTITUDE	in cm
13	VERTICAL_SPEED	in cm/s
14	TROTTLE_POS	in %
15	ATTITUDE_ROLL	in degrees
16	ATTITUDE_PITCH	in degrees
17	IS_ARMED	boolean 0/1
18	IS_AUTOLAUNCH	boolean 0/1
19	IS_ALTITUDE_CONTROL	boolean 0/1
20	IS_POSITION_CONTROL	boolean 0/1

FLIGHT

Operand Value	Name	Notes
21	IS_EMERGENCY_LANDING	boolean 0/1
22	IS_RTH	boolean 0/1
23	IS_WP	boolean 0/1
24	IS_LANDING	boolean 0/1
25	IS_FAILSAFE	boolean 0/1
26	STABILIZED_ROLL	Roll PID controller output [-500:500]
27	STABILIZED_PITCH	Pitch PID controller output [-500:500]
28	STABILIZED_YAW	Yaw PID controller output [-500:500]
29	ACTIVE_WAYPOINT_INDEX	Indexed from 1. To verify WP is in progress, use IS_WP
30	ACTIVE_WAYPOINT_ACTION	See ACTIVE_WAYPOINT_ACTION paragraph
31	3D HOME_DISTANCE	in meters, calculated from HOME_DISTANCE and ALTITUDE using Pythagorean theorem
32	CROSSFIRE LQ	Crossfire Link quality as returned by the CRSF protocol
33	CROSSFIRE SNR	Crossfire SNR as returned by the CRSF protocol
34	GPS_VALID	boolean 0/1. True when the GPS has a valid 3D Fix
35	LOITER_RADIUS	The current loiter radius in cm.
36	ACTIVE_PROFILE	integer for the active config profile [1..MAX_PROFILE_COUNT]
37	BATT_CELLS	Number of battery cells detected
38	AGL_STATUS	boolean 1 when AGL can be trusted, 0 when AGL estimate can not be trusted
39	AGL	integer Above The Groud Altitude in cm
40	RANGEFINDER_RAW	integer raw distance provided by the rangefinder in cm

FLIGHT

ACTIVE_WAYPOINT_ACTION

Action	Value
WAYPOINT	1
HOLD_TIME	3
RTH	4
SET_POI	5
JUMP	6
SET_HEAD	7
LAND	8

FLIGHT_MODE

Operand Value	Name
0	FAILSAFE
1	MANUAL
2	RTH
3	POSHOLD
4	CRUISE
5	ALTHOLD
6	ANGLE
7	HORIZON
8	AIR
9	USER1
10	USER2

PROGRAMING LOGIC SAMPLE

One of the Defining Feature of the Synerduino STM and INAV combine is you can configure the Controller as an PLC this means you can introduce an cause and effect behavior to the drone too do a certain action when condition variable is in effect.

Some sample one could create behavior

- Tilt Trigger
- Low Battery Failsafe
- Geofence
- Waypoint Trigger
- Camera Gimbal
- Data Analytic Intervention
- Flight Mode override
- etc,

The Following are selected sample to help you to understand how programming works

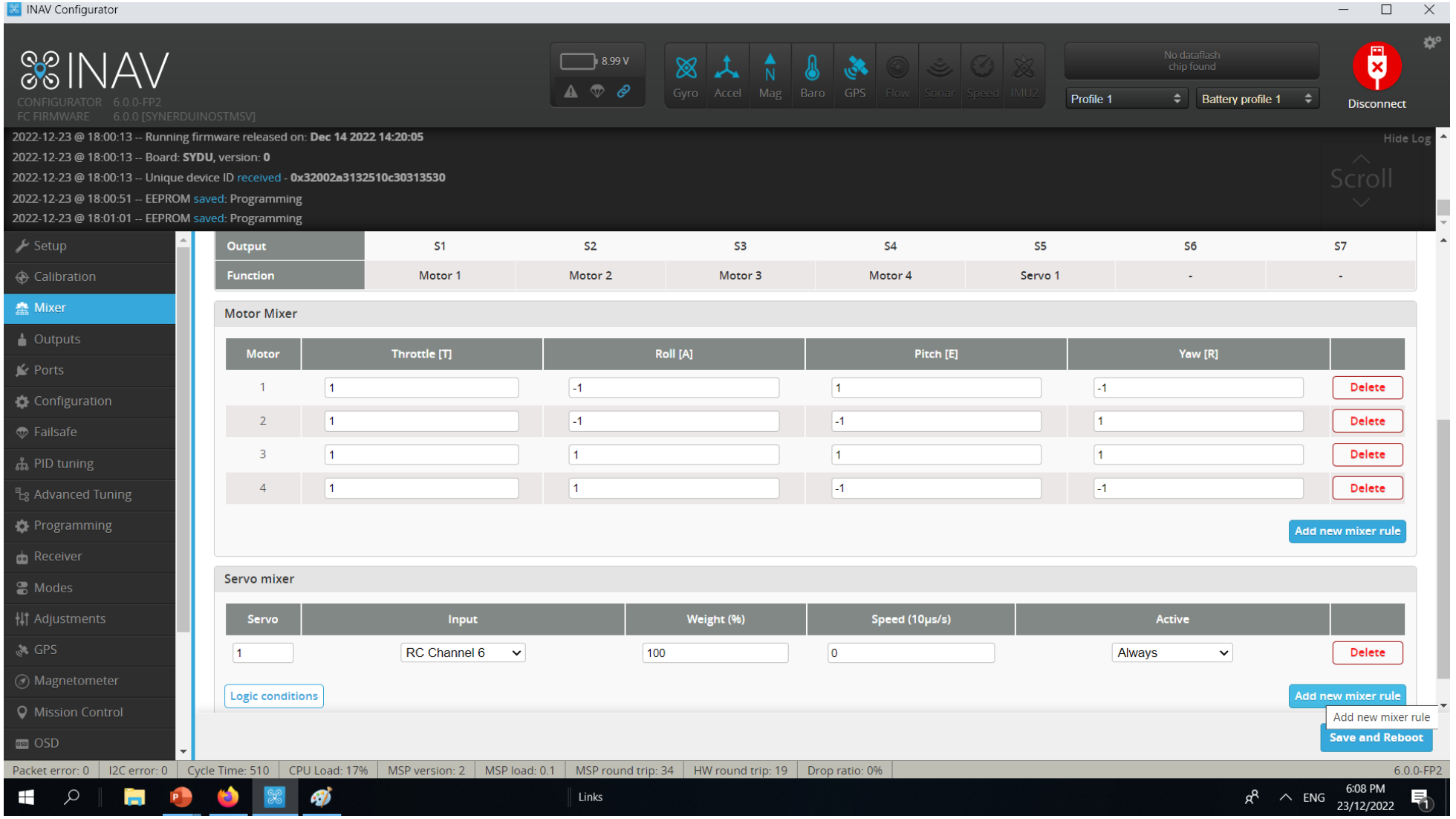
Tilt Trigger sample

In this Programing sample we'll be triggering a servo when the drone tilt more than 10 degrees to the right

This application can be use to

- deploy payload
- Stabilization gimbal
- trigger a companion hardware to perform a task

In Mixer tab add new Mixer Rule to RC channel 6



Servo S7 Pin SYNERDUINOSTM.Hex
Servo S5 Pin SYNERDUINOSTMSV.Hex

Tilt Trigger sample

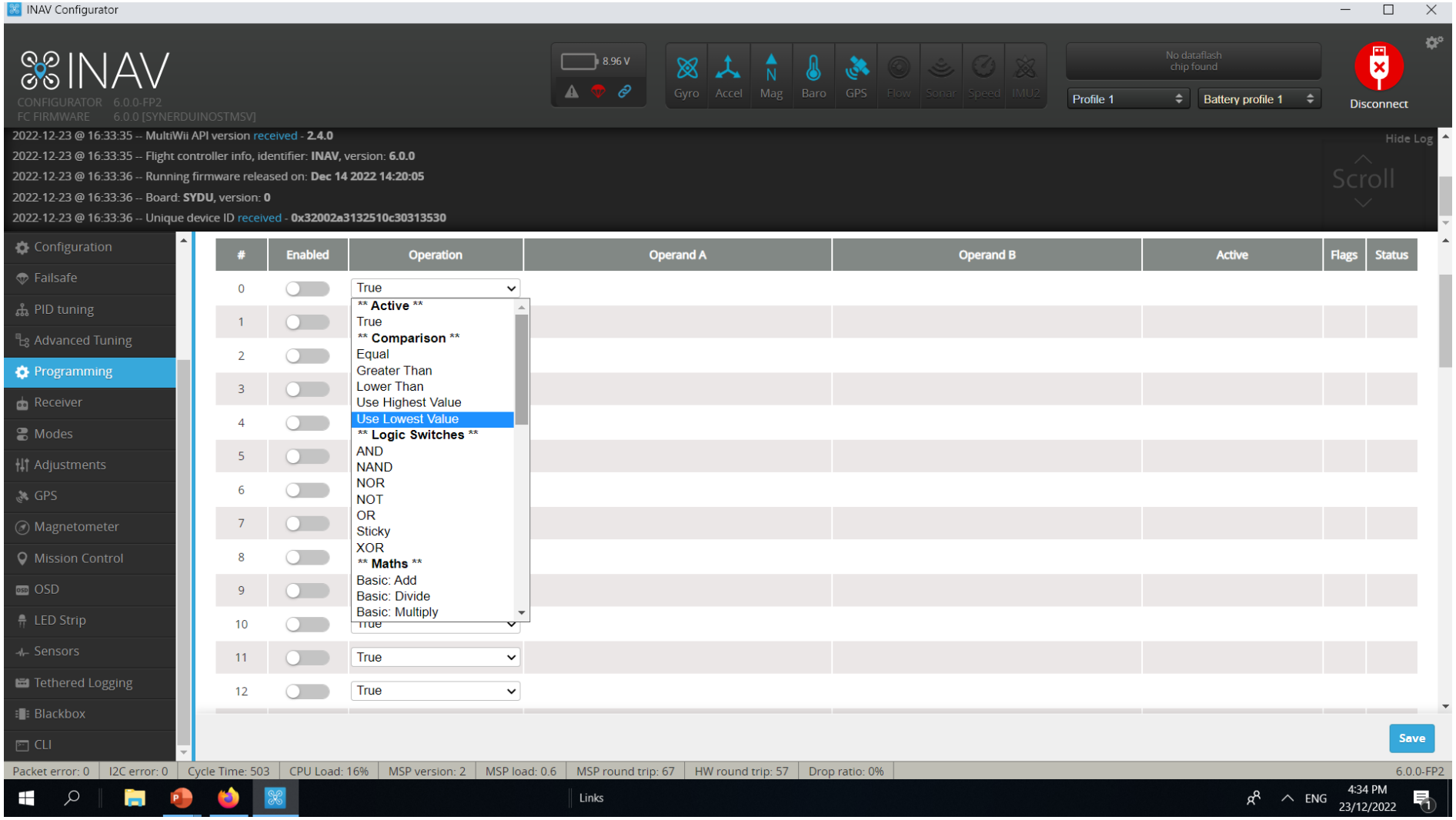
Inav has Programming value you can instruct Conditions to and when to conditions are Valid it can be use to trigger an action that the Synerduino would do.

This can serve basic

- failsafe function
- Geofence
- Servo trigger

To more complex

- Mission conditions
- Events Conditions



Tilt Trigger sample

In this Sample

Operation
-this instruct the type of operation the programming is to perform

#0

- Set GVAR

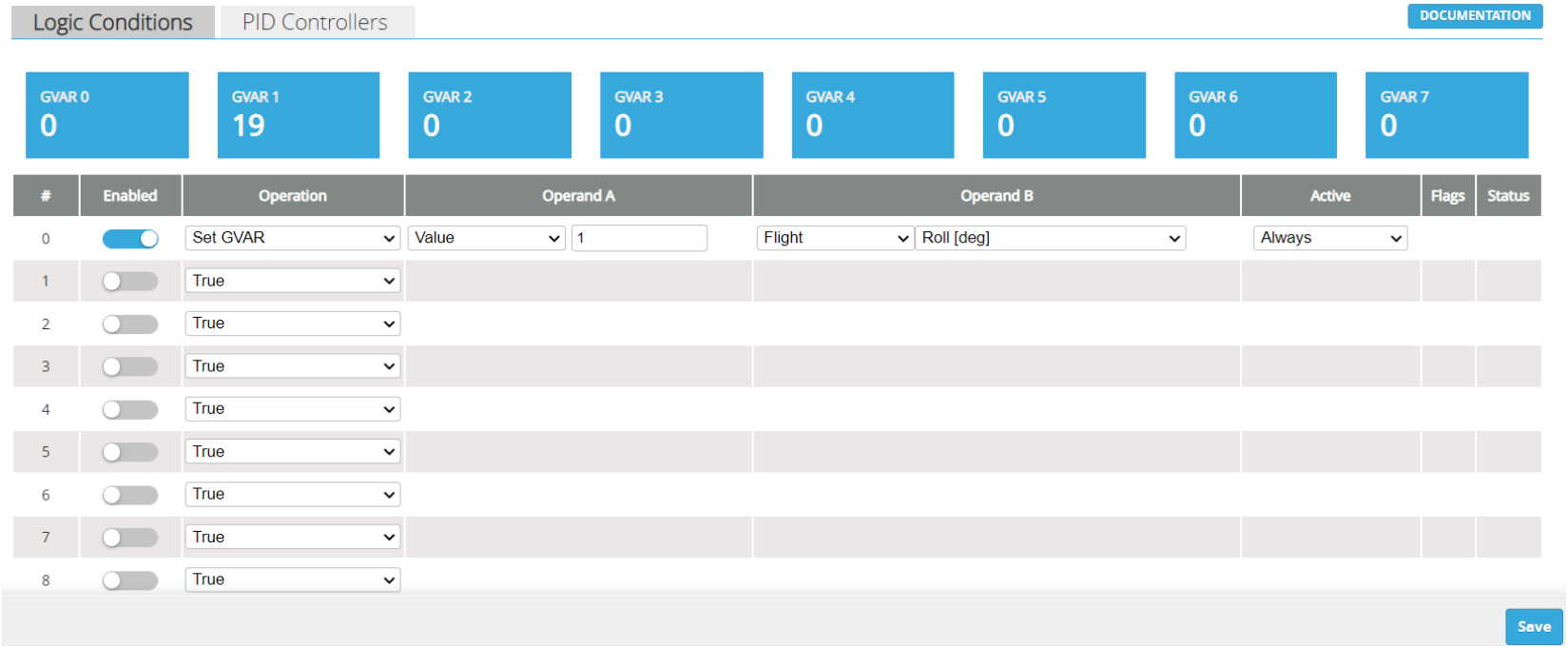
Operand A
- Then we instruct the value
– set which GVAR to write on

Set GVAR to Value 1
- GVAR1

Operand B

-here we want to capture flight data
- This example is Roll (degree)
- Meaning it captures the roll in angle of the Synerduino shield

Active
- means the platform is always active and constant listening to the Roll angle



In this Sample when the Drone roll greater than 10 degrees to the right the servo on S5 triggers

Note: Mixer Tab the Servo

Tilt Trigger sample

Operation
-this instruct the type of operation the programming is to perform

 # 1

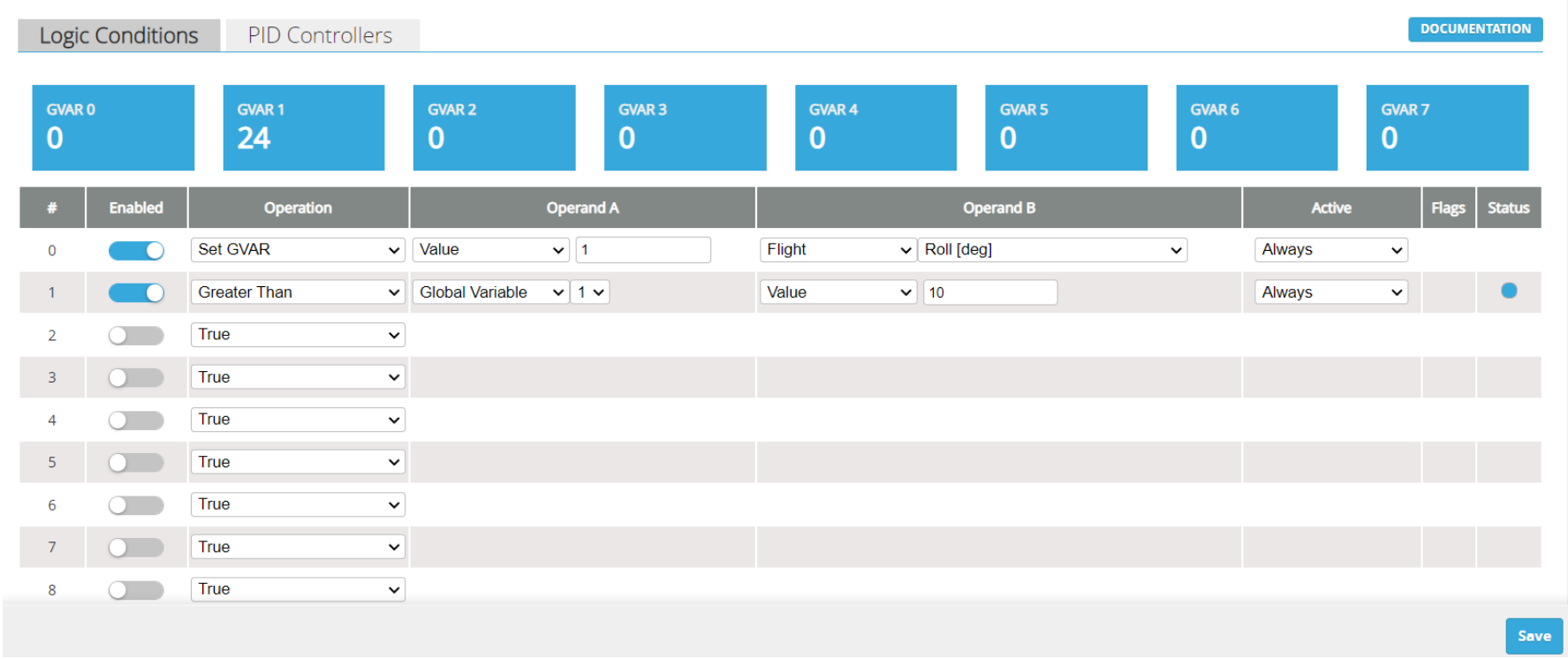
- Greater than (Conditions)

Operand A
-We want to Extract information from GAVR 1 (Global Variable 1)

Operand B
- if the Value is 10 (Degrees)
- This example is Roll (degree)

Active
means the platform is always active and constant listening to the Roll angle

Status
- This will trigger if the Value greater than 10 degrees is met it should indicate



Tilt Trigger sample

In this Sample

Operation
-this instruct the type of operation the programming is to perform

#2

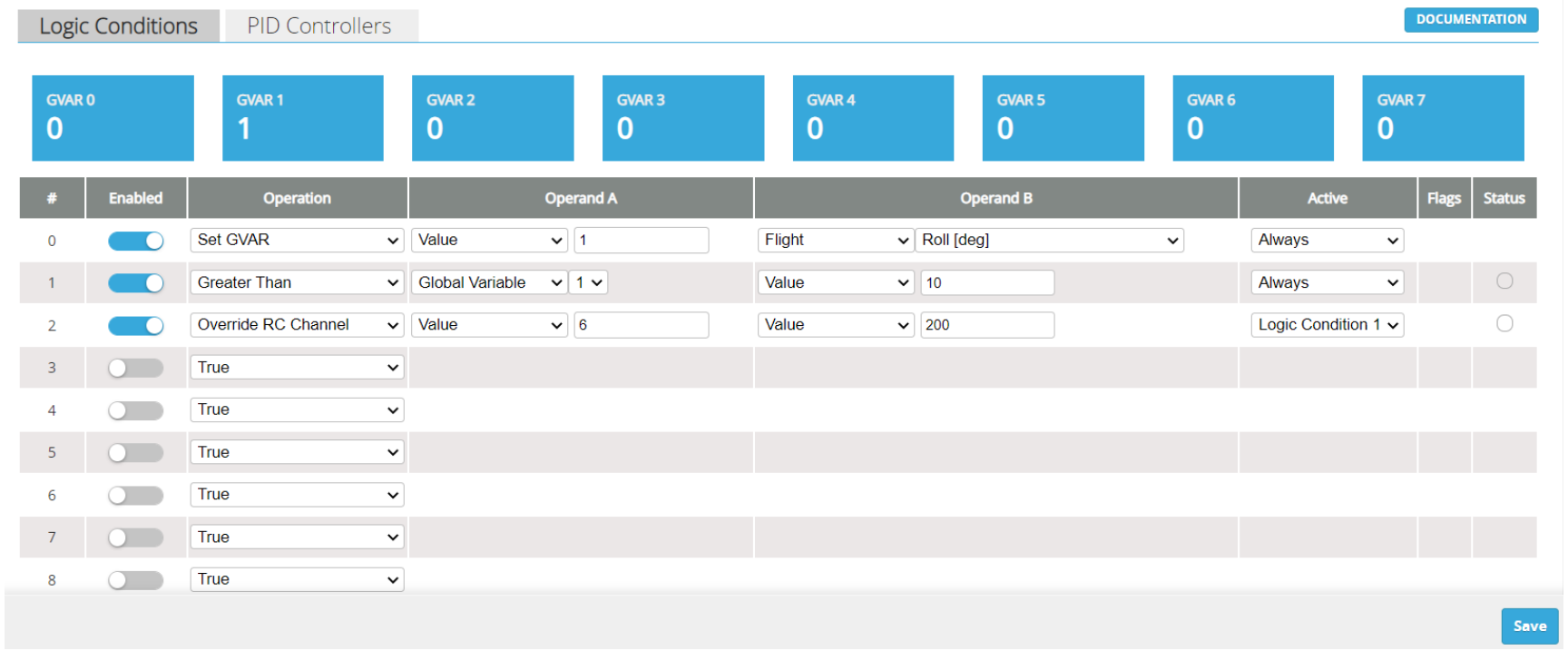
Override RC channel

Operand A
- Value6 (RC Channel 6 or Aux2)

Operand B
- Value 200 (Degrees of servo plugged into Channel 6)

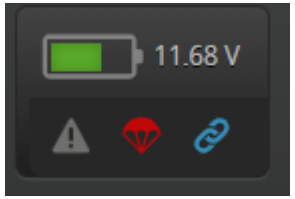
Active

- Logic condition 1
- It looks at the Condition 1 if its status is active then the condition is true to trigger the servo



Low Battery Failsafe

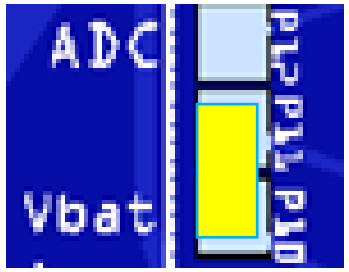
Configuration



Before a Low battery failsafe RTH can be setup you needed to configure your battery Cell count and Voltage scale

You can cross calibrate this with an separate battery monitoring buzzer

This would define your signal point on where to trigger.



P11-P10 is Vbat to read from the battery input as battery monitoring voltage

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

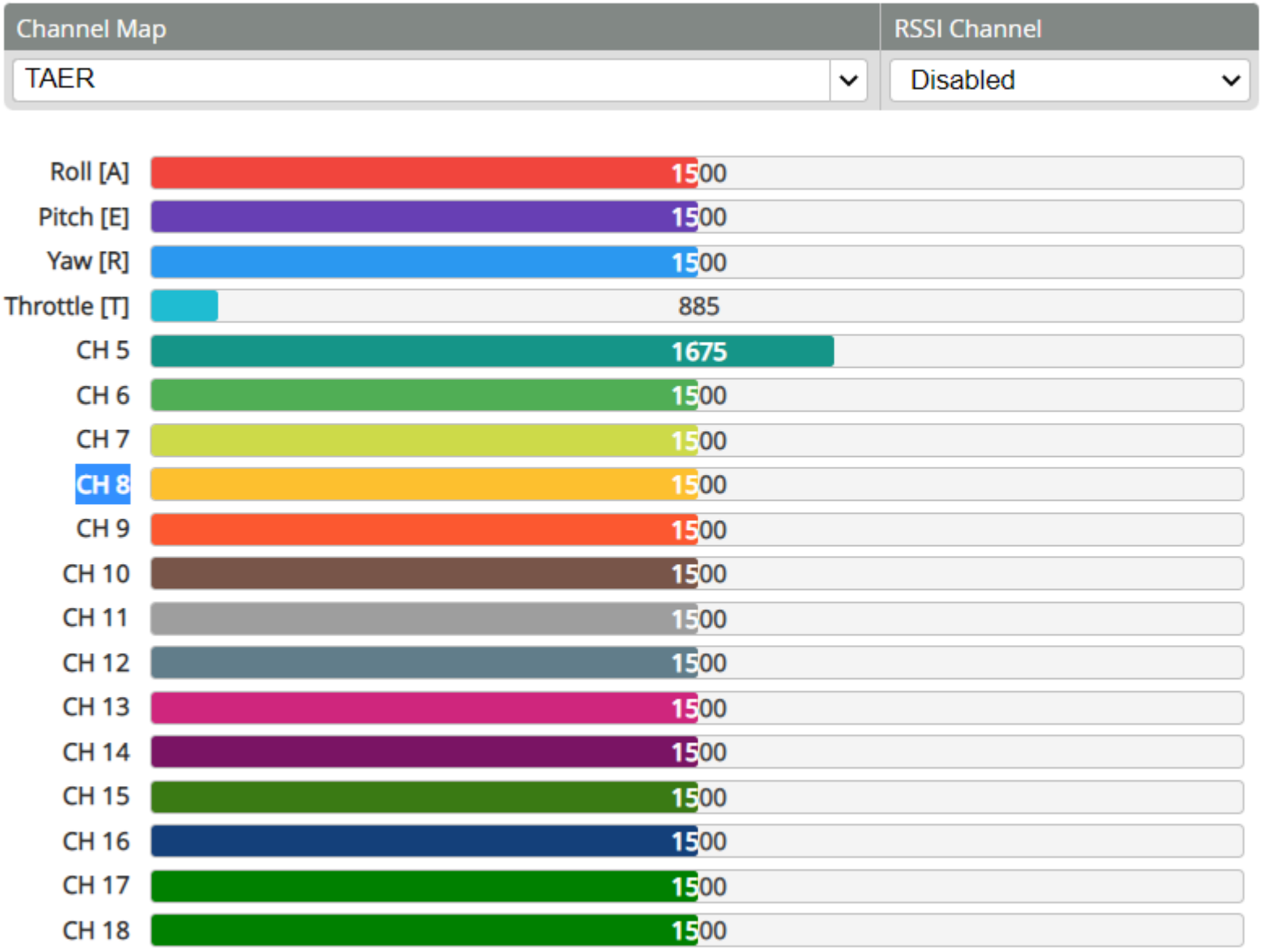
3	Number of cells (0 = auto)
4.25	Maximum cell voltage for cell count detection
3.3	Minimum Cell Voltage
4.2	Maximum Cell Voltage
3.5	Warning Cell Voltage
mAh	Battery Capacity Unit
0	Capacity
	Warning Capacity (remaining %)
	Critical Capacity (remaining %)

Low Battery Failsafe

Receiver

An empty Channel needed to be chosen for the trigger point.

In this case CH8

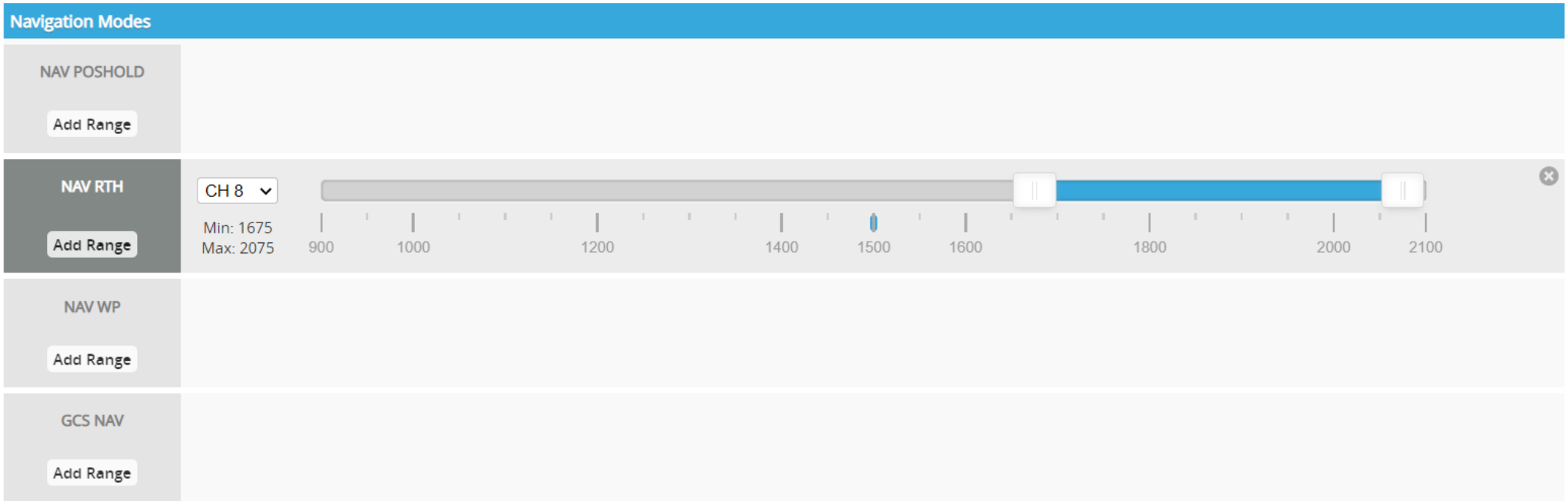


Low Battery Failsafe

In NAV RTH a channel associated to it needs to be configure to that flight mode

Modes

In this case CH8



Low Battery Failsafe

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

First we need to visualize the Voltage in this case 11.63v or GVAR0 1163

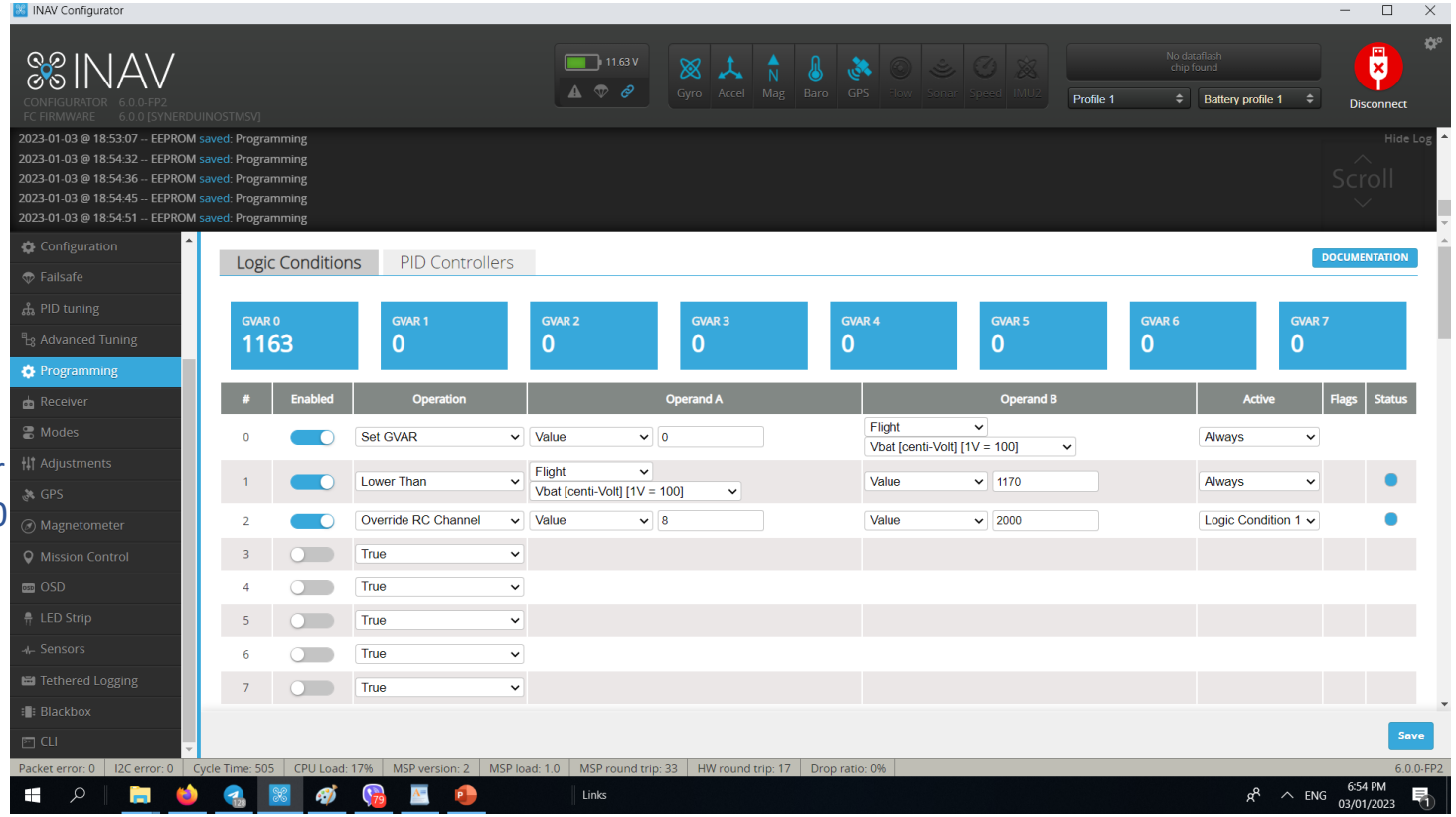
1# Lower than
 Operand A = Flight Vbat Volt
Operand B = Value 1170
Active = Always

Here we set an sample that if the Vbat Volt is is Lower than 11.70V or 1170 the status would trigger but if its higher lets say 12.00V or 1200 than the Status would stay off

2# Override RC Channel
Operand A = Value 8
Operand B = Value 2000
Active = Logic condition 1

Here is where the magic happens when the conditions are met in Logic #1 with a active Status this would trigger the Value 8 (Ch8) to set PWM to 2000 overriding the RC input and triggering the RTH in Modes. This puts the drone to a RTH mode.

This sample also can be utilized for Data Analytics intervention



Waypoint Trigger

Waypoint trigger is useful for triggering payload or drone function when its above a waypoint assign to it

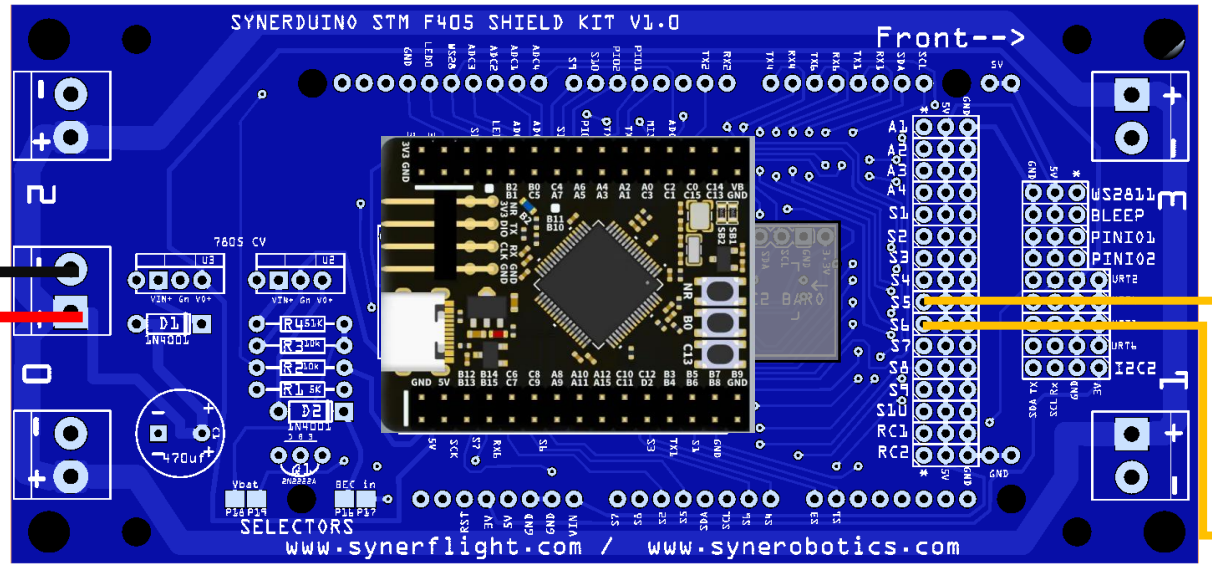
UBEC is Required for High current applications if it exceeded more than

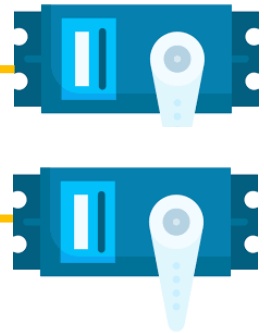
- 3A for STMF405 and H743
- 1.5A for F411 of use when driving servos

BEC input requirements must be 5V

External ADC Sensor

11.1V Lipo





Any unused Pins can be assigned in the mixer for the task

Waypoint Trigger

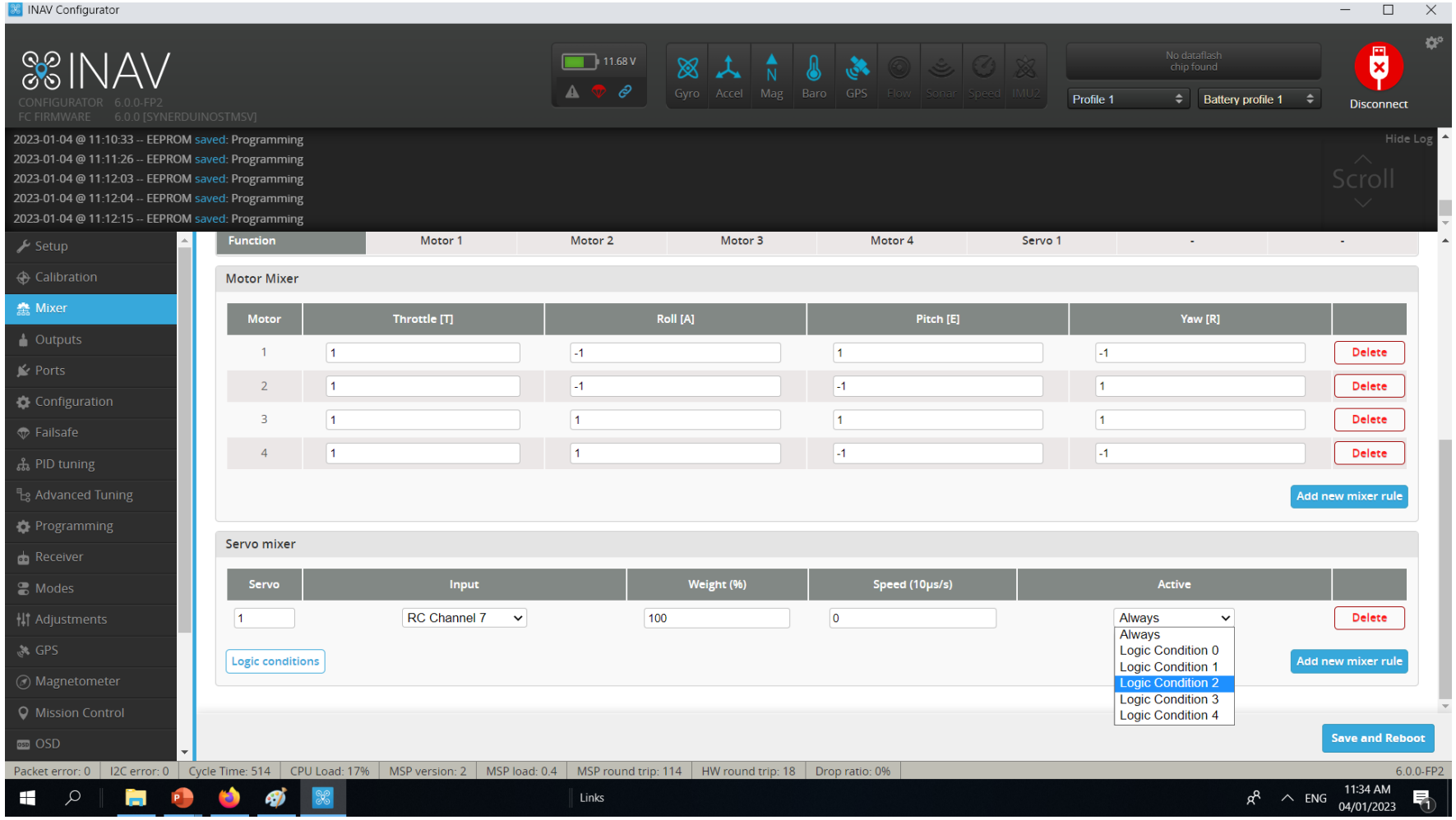
Mixer

INAV5 – INAV6

In Mixer RC channel 7 is associated to servo1
S1 on SYNERDUINOSTM.HEX
S5 on SYNERDUINOSTMSV.HEX

Logic condition can also be added to the mixer removing the need to add 3# Logic condition in the Programming

Servos can be reverse by giving it a negative value on Weight% -100%



INAV5 –INAV6 Be aware of your firmware each version has a specific number of servos available for use

1 servo on S7	SYNERDUINOSTM.Hex
3 Servos on S5-S7	SYNERDUINOSTMSV.Hex

Waypoint Trigger

Mixer

INAV 7 - INAV8

RC Channel 7 and 8 on the remote (Aux 7 and 8) Assign for servo input 1 and 2 respectively

On Board PWM Pins

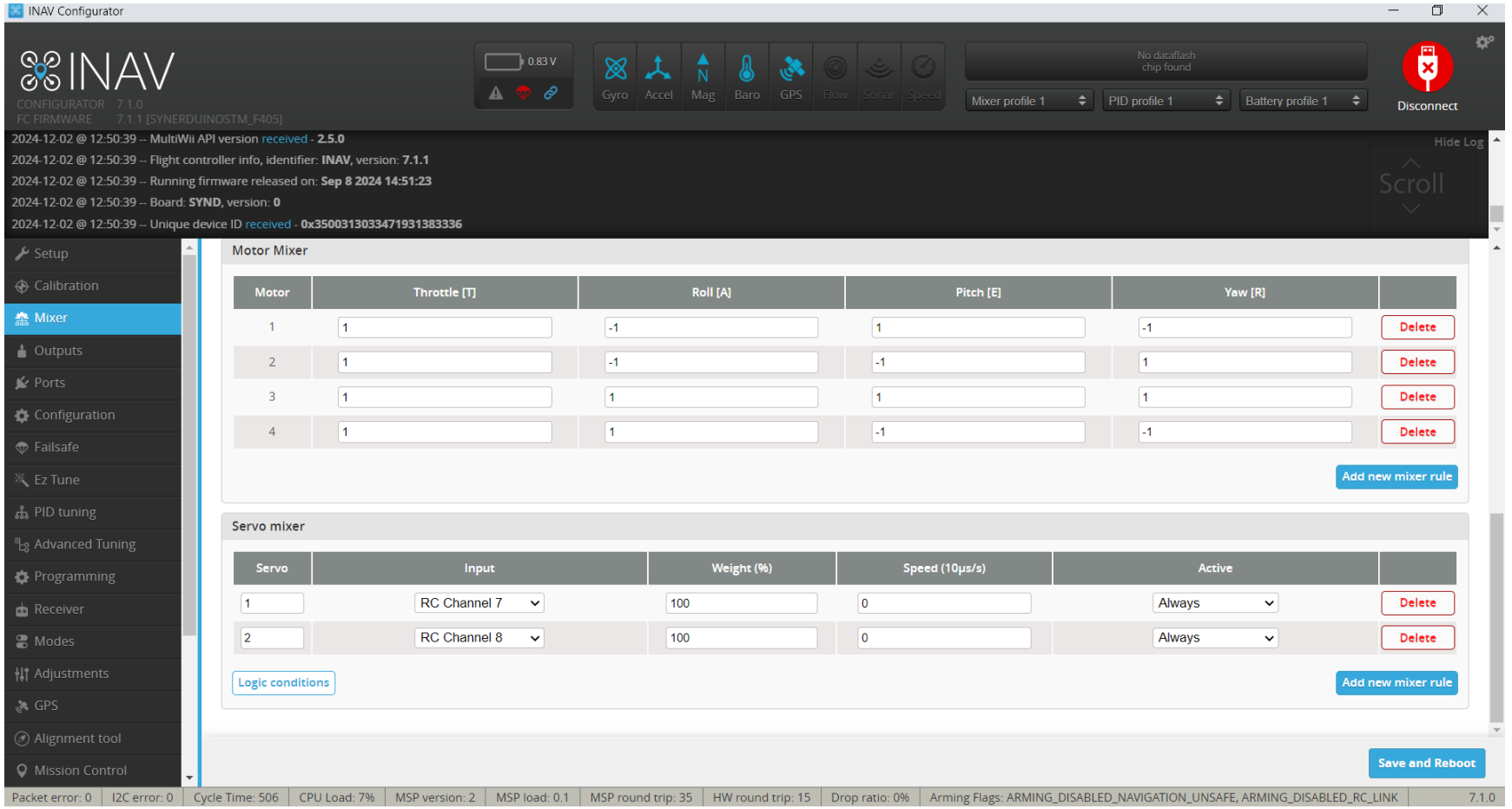
S7 – servo 1

S8 – Servo 2

Is assign automatically when save

Output Mapping

Output (timer)	S1 (Timer 4)	S2 (Timer 3)	S3 (Timer 3)	S4 (Timer 2)	S5 (Timer 1)	S6 (Timer 1)	S7 (Timer 12)	S8 (Timer 12)	S9 (Timer 14)	S10 (Timer 13)
Function	Motor 1	Motor 2	Motor 3	Motor 4	-	-	Servo 1	Servo 2	-	-



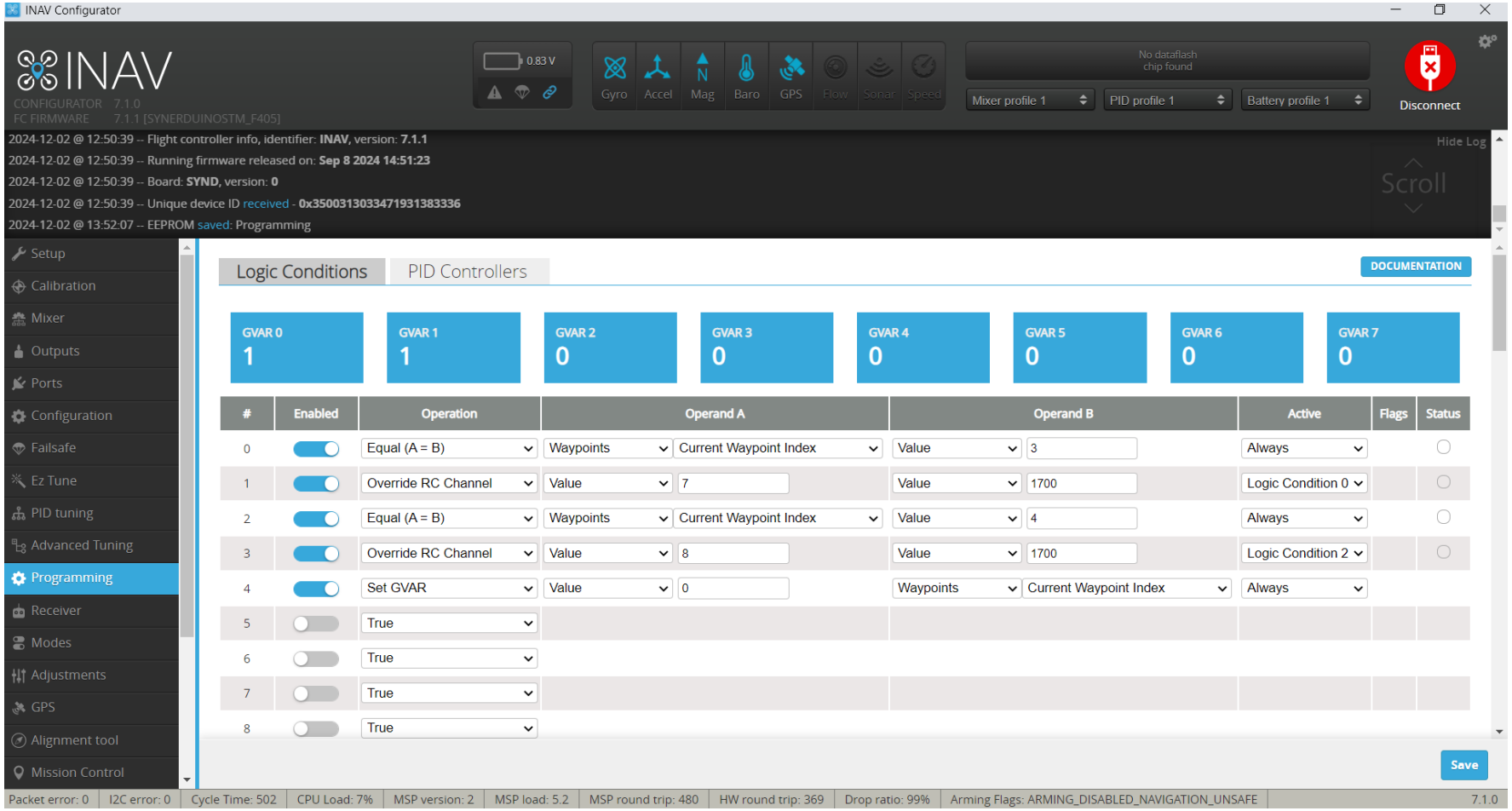
Waypoint Trigger

Waypoint trigger is a function to trigger a servo or a payload function of the drone when the drone reaches a define waypoint to trigger or when conditions are true.

#0 Equal A=B
Current Waypoint index = Value 3
#1 Override RC Channel
Channel Value7 to Value 1700
Logic Condition 0

#2 Equal A=B
Current Waypoint index = Value 3
#3 Override RC Channel
Channel Value7 to Value 1700
Logic Condition 2

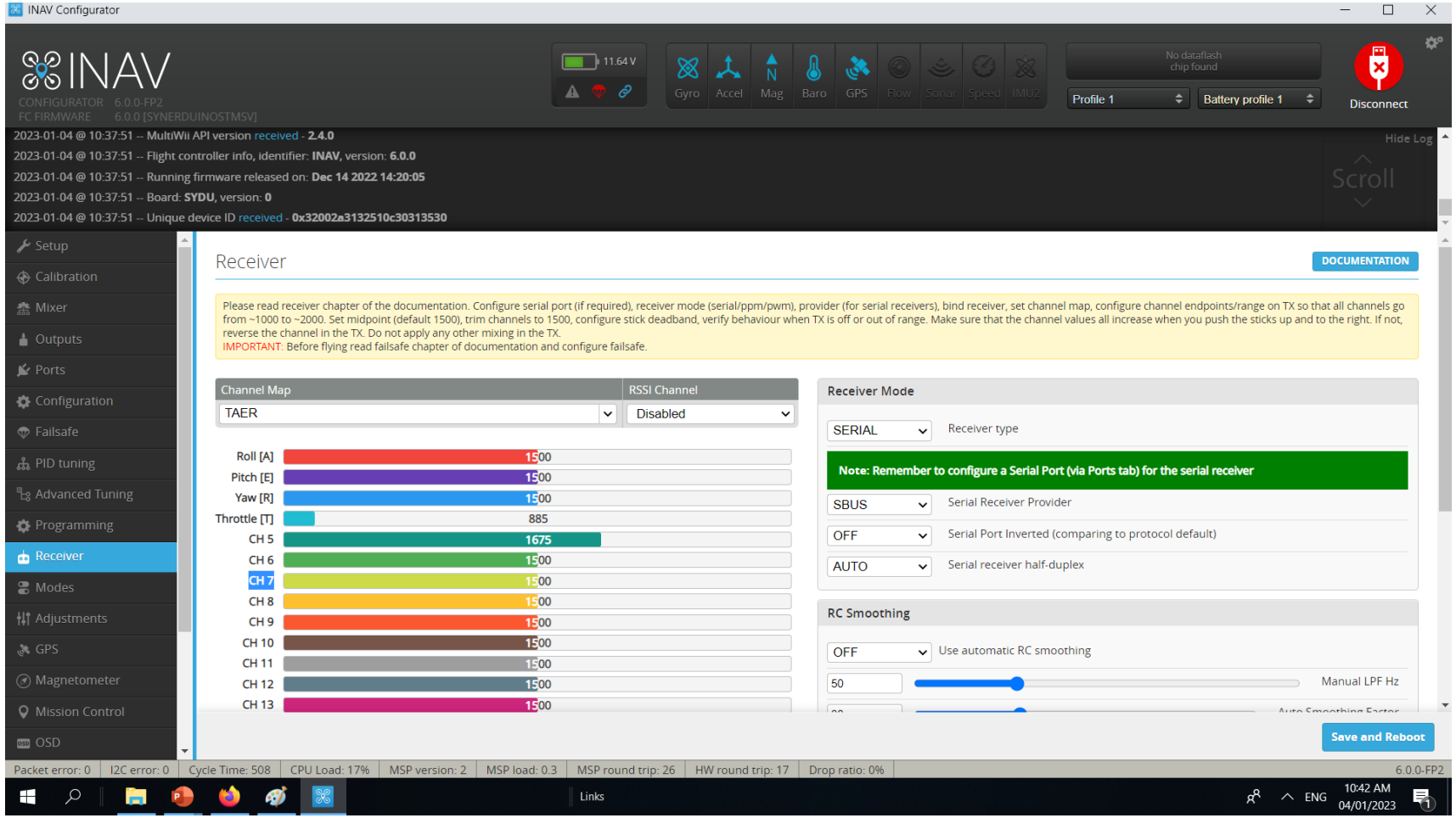
Set GVAR Value 0
(To help visualize Waypoint)
Current Waypoint index



Waypoint Trigger

Receiver

We Wanted the RC Channel 7 and Channel 8 to be free for the Mode Trigger



Waypoint Trigger

Mission Control

when the Drone is above the said Waypoint address

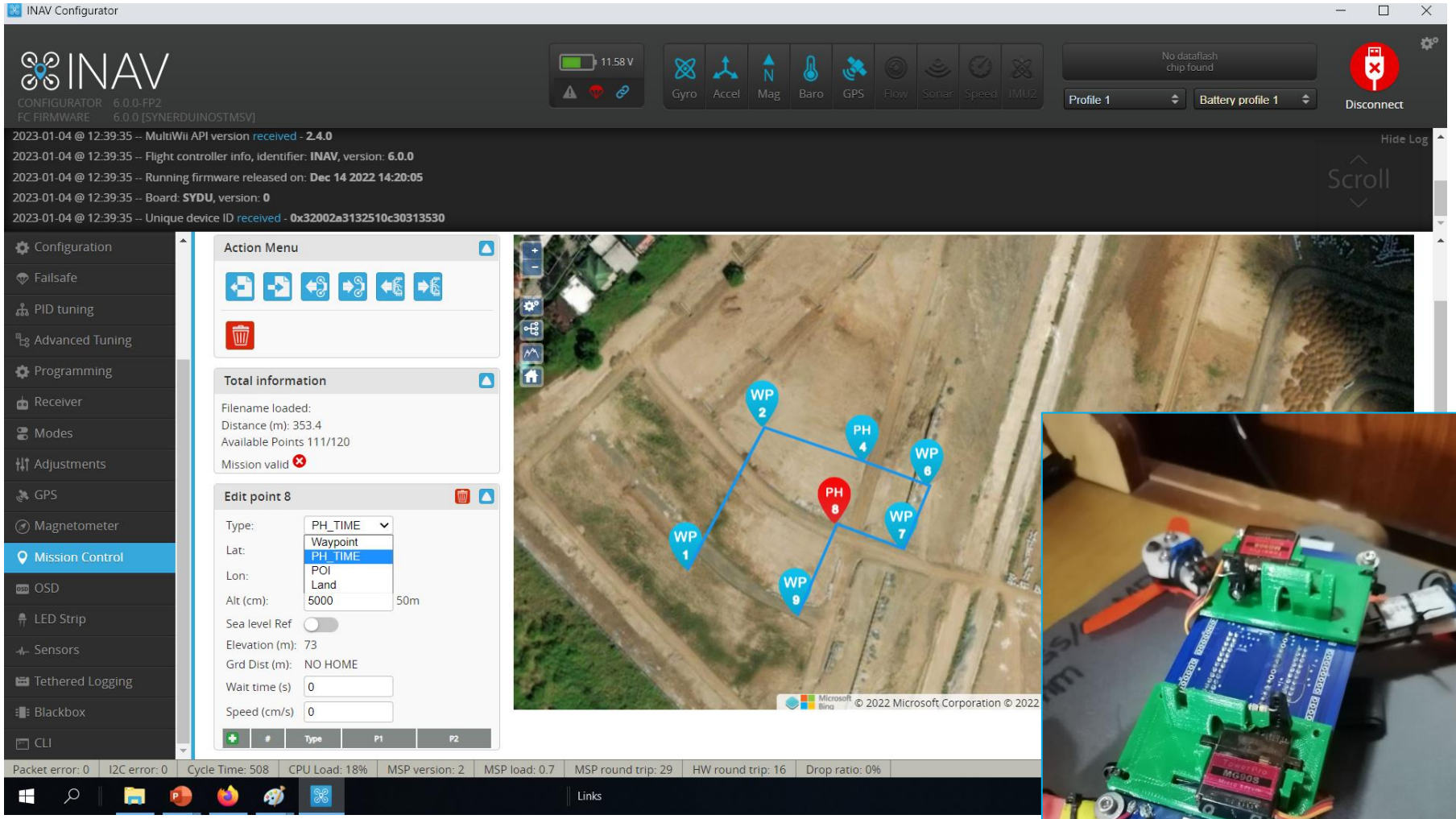
In this sample

Servo 1 is trigger by waypoint 3 waypoint index

And

Servo 2 is trigger by Waypoint 4 waypoint index

The payload would Activate



Geofence

This is a means to restrict a drone within a perimeter from its home point

It’s a feature use for beginners to prevent flying too far

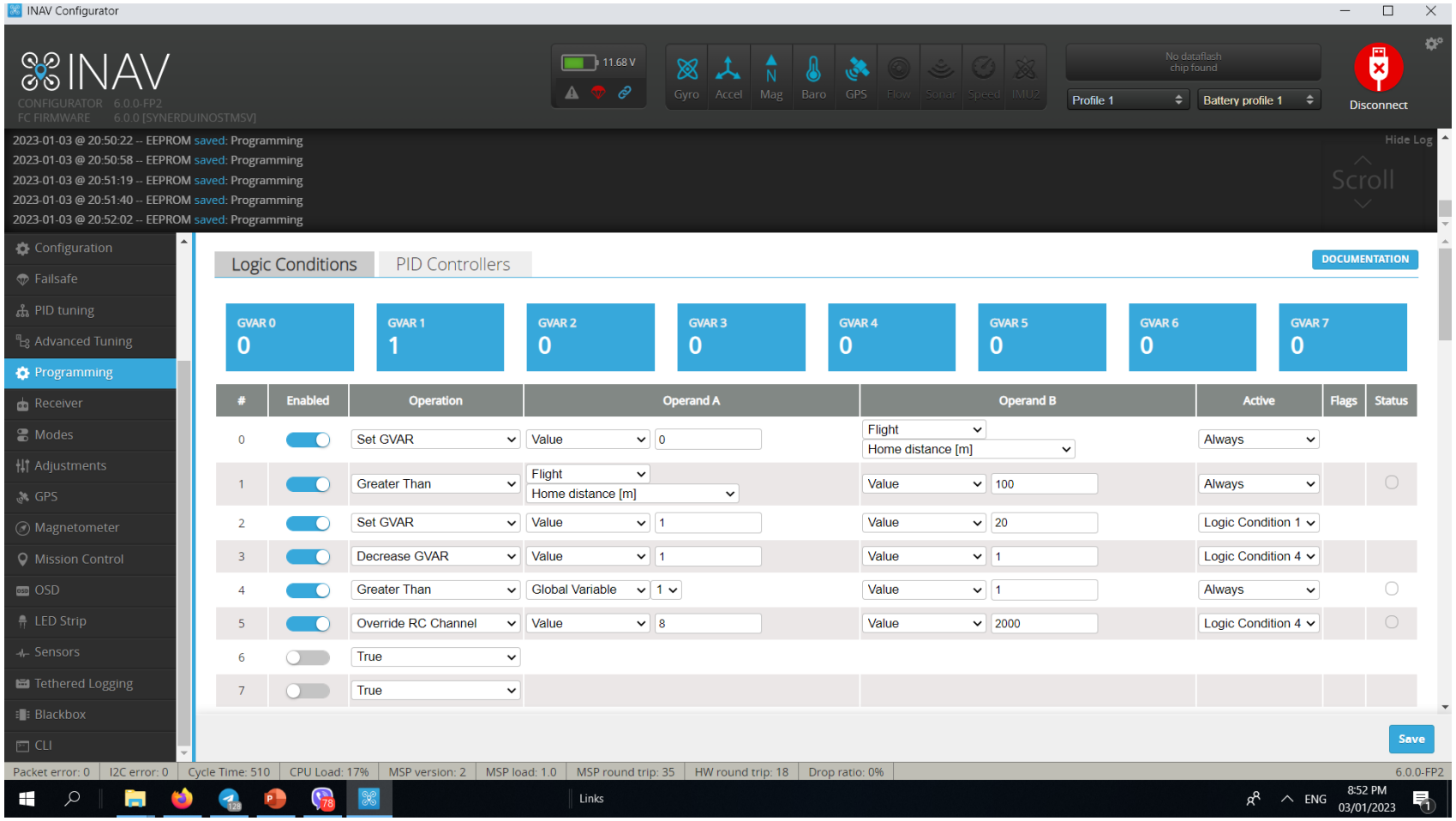
In this sample we want to keep the drone with in the 100m radius of an area from home point

0# set GVAR
Operand A = Value 0
Operand B = Flight Home Distance (m)
Active = Always

First we need to visualize Distance from home indicated by GVAR0

1# Greater than
Operand A = Flight Home Distance (m)
Operand B = Value 100(m)
Active = Always

Here we set the value of 100m from home this would be the boundary of the Geofence



Geofence

2# set GVAR
Operand A = Value 1
Operand B = Value 20
Active = Logic condition 1

Here we set our variable when logic 1 condition is triggered, we set GVAR 1 from 20 (2 sec) timer

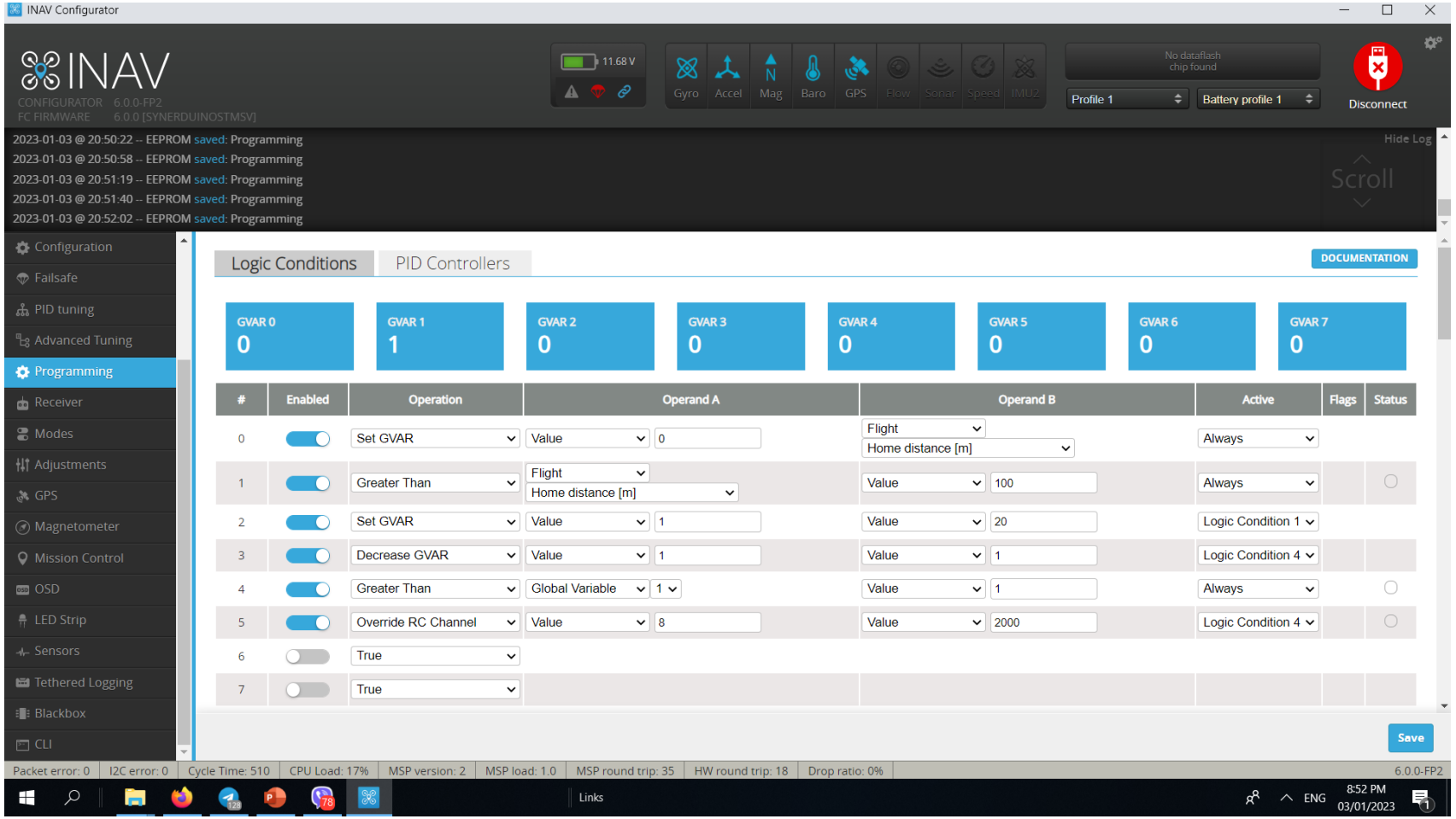
3# Decrease GVAR
Operand A = Value 1
Operand B = Value 1
Active = Logic condition 4

Here GVAR1 starts the count down
(this countdown only active when Condition 4 is Active)

4# Greater Than
Operand A = Global Variable 1
Operand B = Value 1
Active = Always

If GVAR1 has a Value more than 1 its Status would be Active.

5# Override RC Channel
Operand A = Value 8
Operand B = Value 2000
Active = Logic condition 4



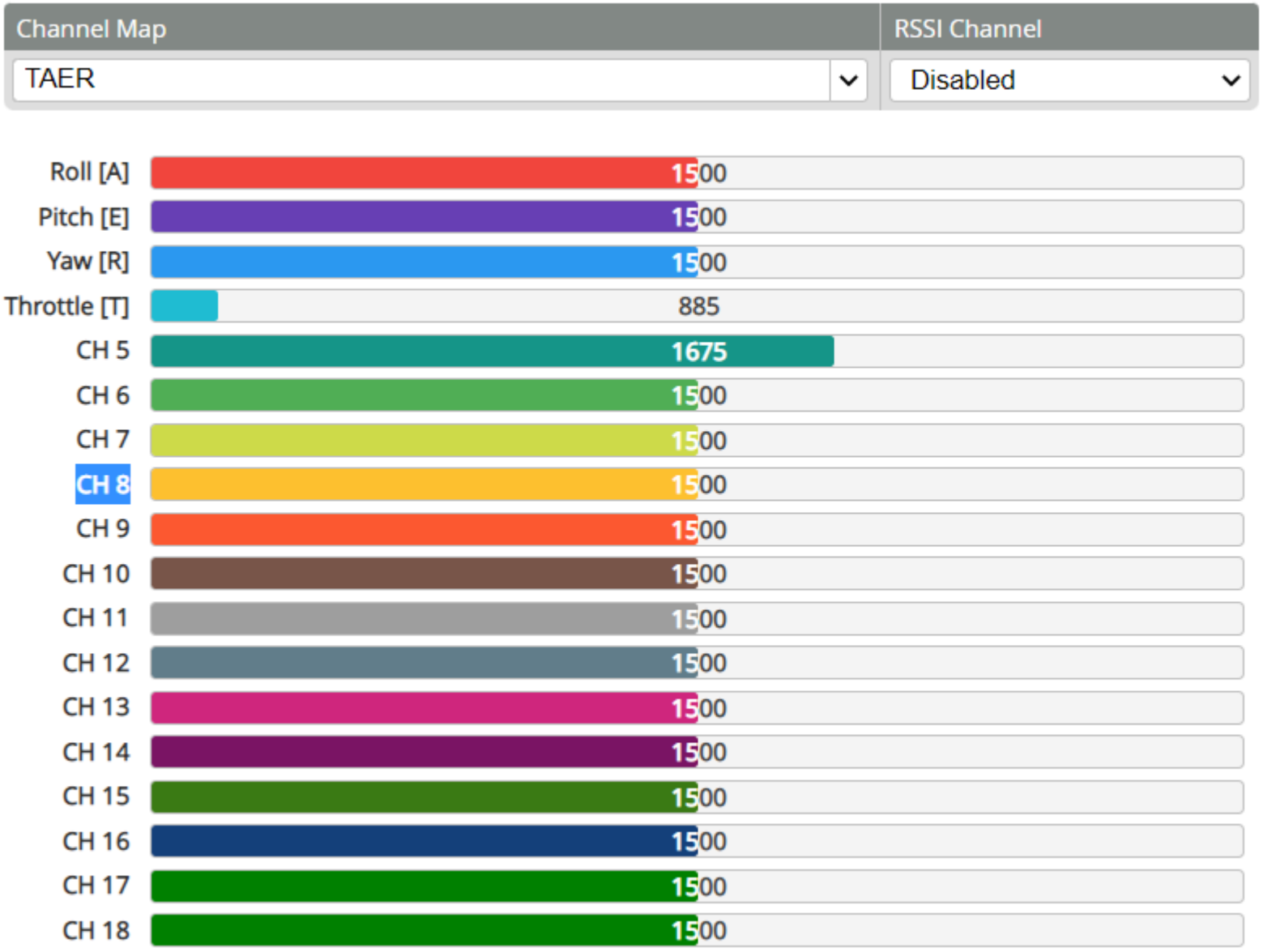
When the conditions are met in Logic #1 with an active Status this would trigger the Value 8 (Ch8) to set PWM to 2000 overriding the RC input and triggering the RTH in Mode. But this mode is only active for 2 seconds then the drone resumes last command. This way it keeps the drone in the 100m Perimeter without doing the complete RTH action.

Geofence

Receiver

An empty Channel needed to be chosen for the trigger point.

In this case CH8

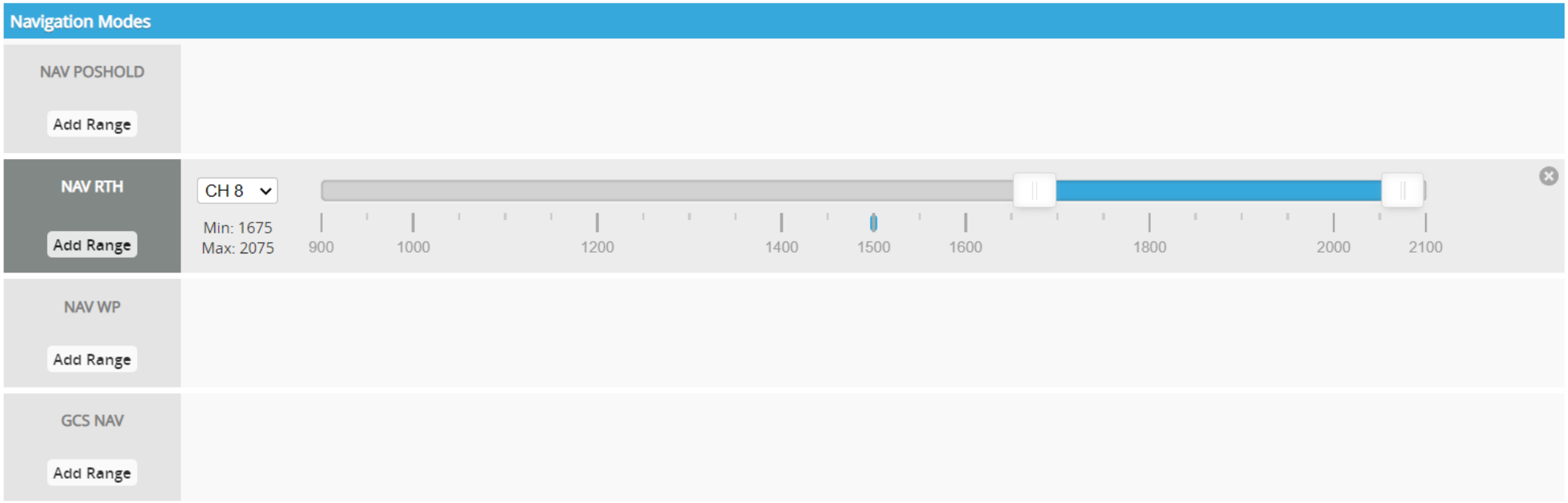


Geofence

In NAV RTH a channel associated to it needs to be configure to that flight mode

Modes

In this case CH8



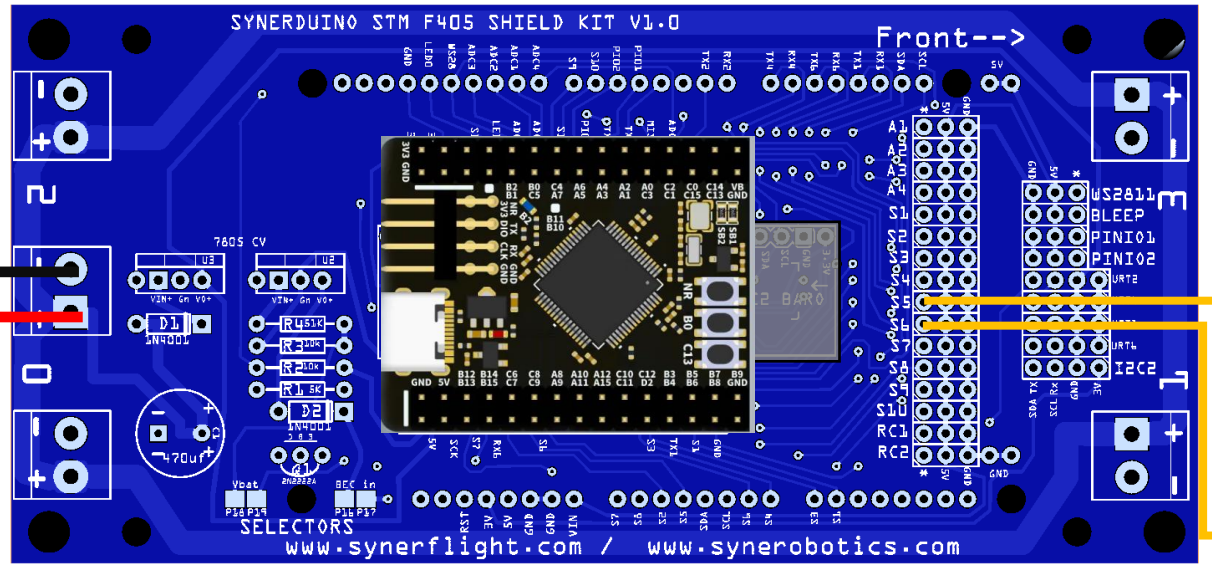
Gimbal

Mainly as Payload like cameras stabilization

Here we can assign Pitch and Roll Gimbal function to the given servo

External ADC Sensor

11.1V Lipo



UBEC is Required for High current applications if it exceeded more than

- 3A for STMF405 and H743
- 1.5A for F411 of use when driving servos

BEC input requirements must be 5V

S7

S8

Any unused Pins can be assigned in the mixer for the task

Gimbal

Mixer

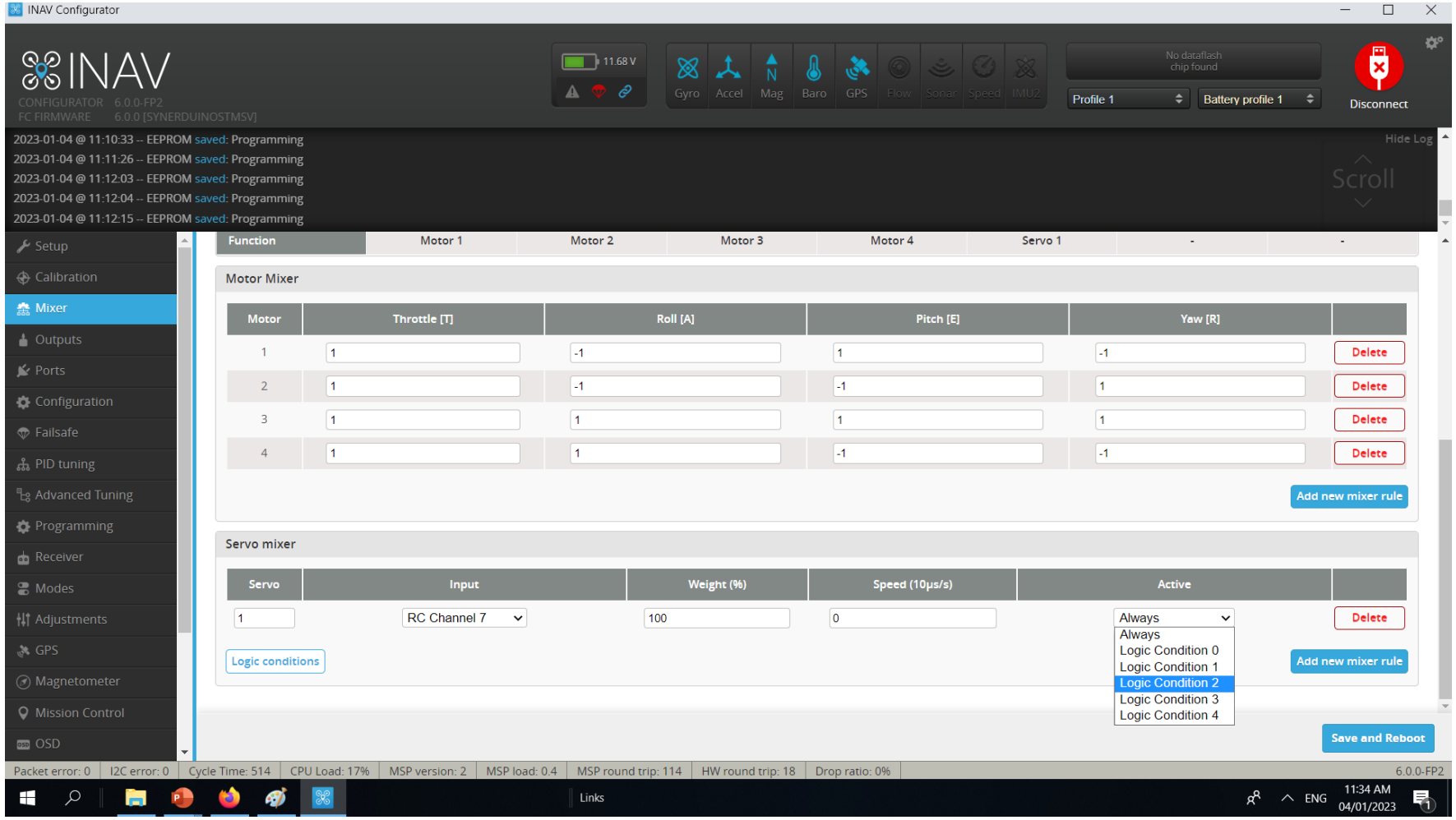
INAV5-6 In Mixer RC channel 7 is associated to servo1

S1 on SYNERDUINOSTM.HEX
S5 on SYNERDUINOSTMSV.HEX

Logic condition can also be added to the mixer removing the need to add 3# Logic condition in the Programming

Servos can be reverse by giving it a negative value on Weight% -100%

Note: only Applies to INAV5 and INAV6



INAV5 –INAV6 Be aware of your firmware each version has a specific number of servos available for use

1 servo on S7	SYNERDUINOSTM.Hex
3 Servos on S5-S7	SYNERDUINOSTMSV.Hex

Gimbal

INAV5-6 With Camera or other instrument that Needs Stabilization you can configure Pitch Roll and Yaw Axis available
such can also be augmented by extra channels from your RC when necessary
In this sample with setup a single roll axis

This case we are going to crunch a few math

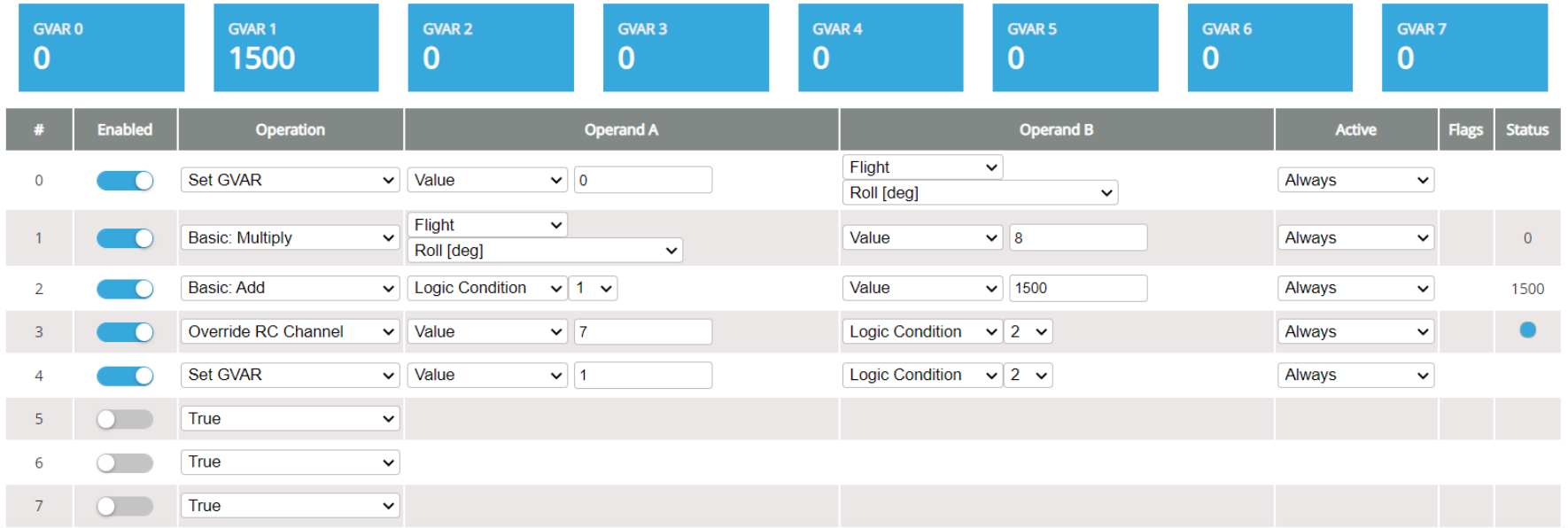
0# set GVAR
Operand A = Value 0
Operand B = Flight Roll Degrees
Active = Always

GVAR 0 is only for Visual reference to see what the roll degrees indicate when you tilt the drone (can be omitted or remove)

1# Basic Multiply
Operand A = Flight Roll Degrees
Operand B = Value 8
Active = Always

We multiply the roll degree X Value of 8
Ex. (10 Degree Roll X 8) this is also what to adjust when your servo needs alignment by multiplier

Note: only Applies to INAV5 and INAV6



Be aware of your firmware each version has a specific number of servos available for use

1 servo on S7	SYNERDUINOSTM.Hex
3 Servos on S5-S7	SYNERDUINOSTMSV.Hex

Gimbal

INAV5-6
2# Basic Add
Operand A = Logic condition 1
Operand B = Value 1500
Active = Always

Here we keep the Output to 1500 or Centerline 90 degree for servo

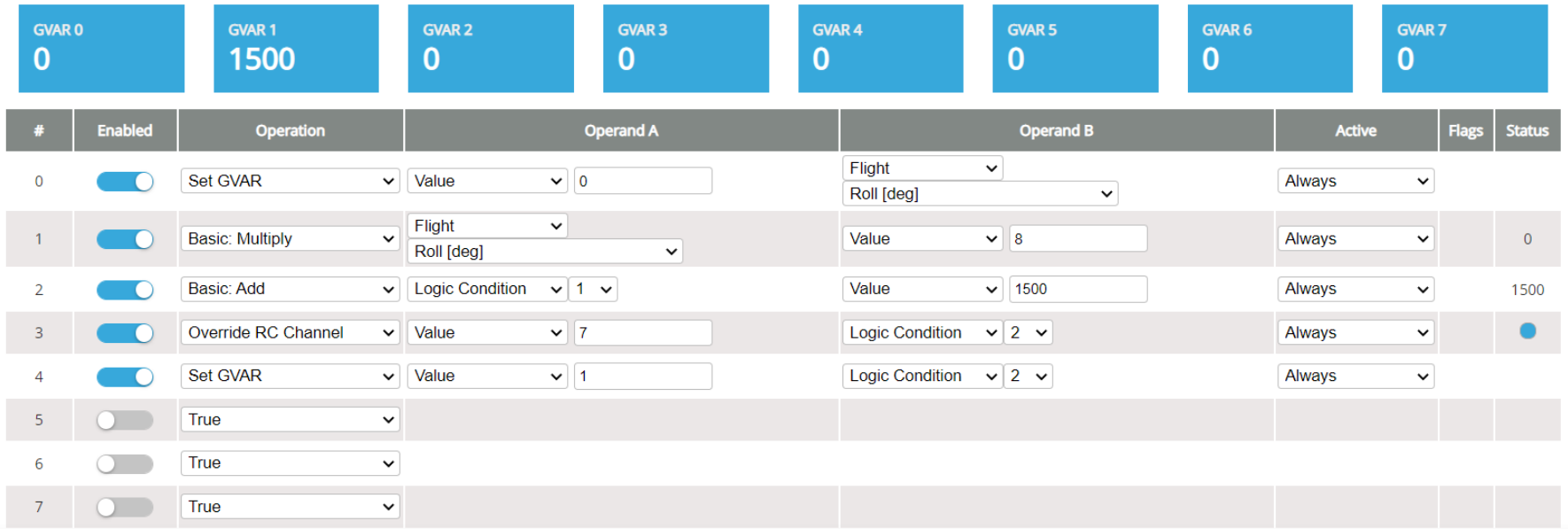
3# Override RC Channel
Operand A = Value 7
Operand B = Logic condition 2
Active = Always

Here the servo should respond to the status output on Logic 2 and should tilt to match the degrees tilt of the drone

4# is same as 0# it's a visualization of logic 2 output

Several chain of similar code set can be applied to Pitch and Yaw axis as well

Note: only Applies to INAV5 and INAV6



Be aware of your firmware each version has a specific number of servos available for use

1 servo on S7	SYNERDUINOSTM.Hex
3 Servos on S5-S7	SYNERDUINOSTMSV.Hex

Gimbal

Mixer

INAV 7 - INAV8

Gimbal Pitch and Roll
Assign for servo input
1 and 2 respectively

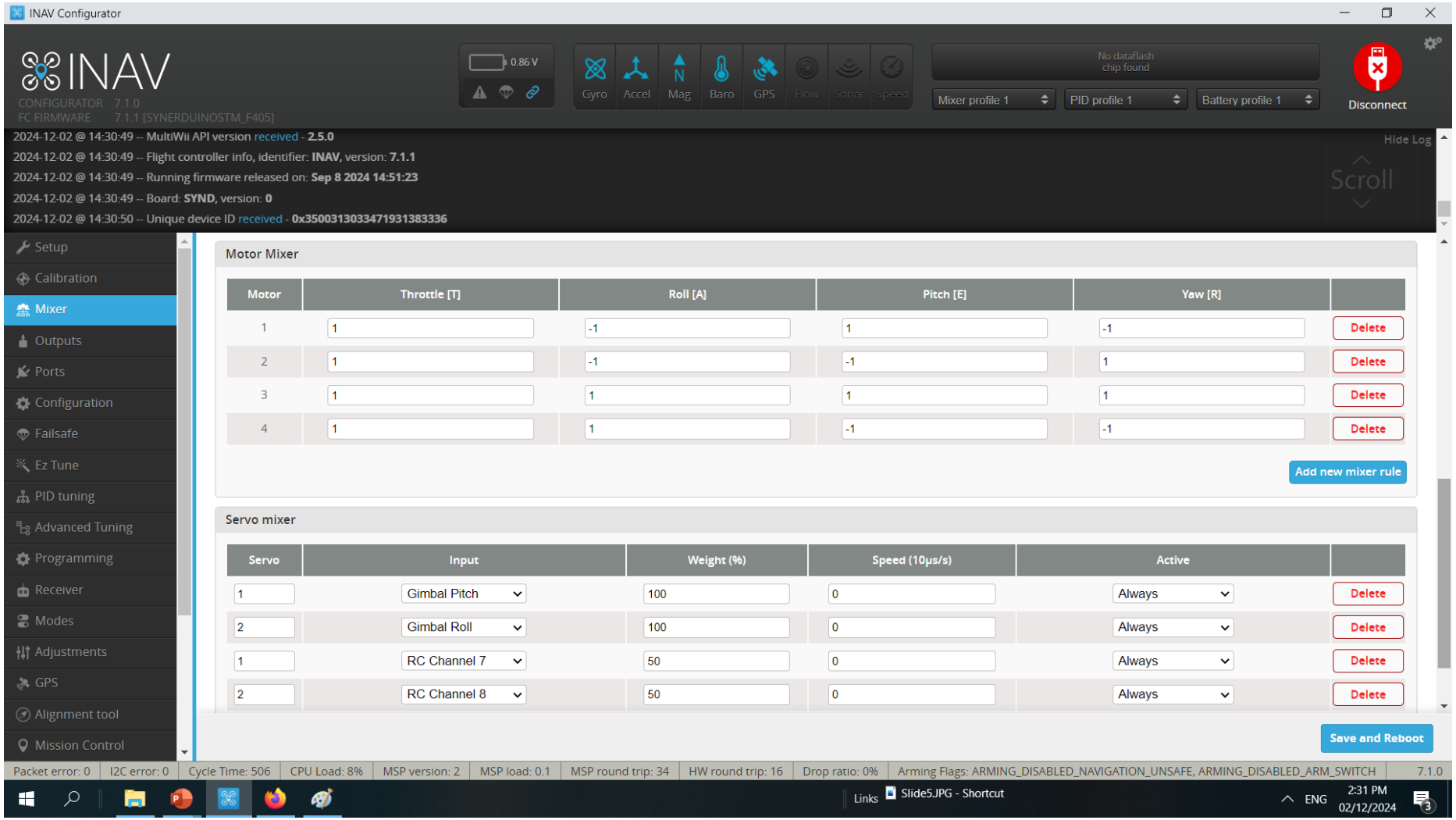
RC Channel 7 and 8
Respectively controls
50% of the Mixing
weight and its useful for
adjusting camera angle
with Remote Knob
Trim control

On Board PWM Pins

S7 – servo 1

S8 – Servo 2

Is assign automatically
when save

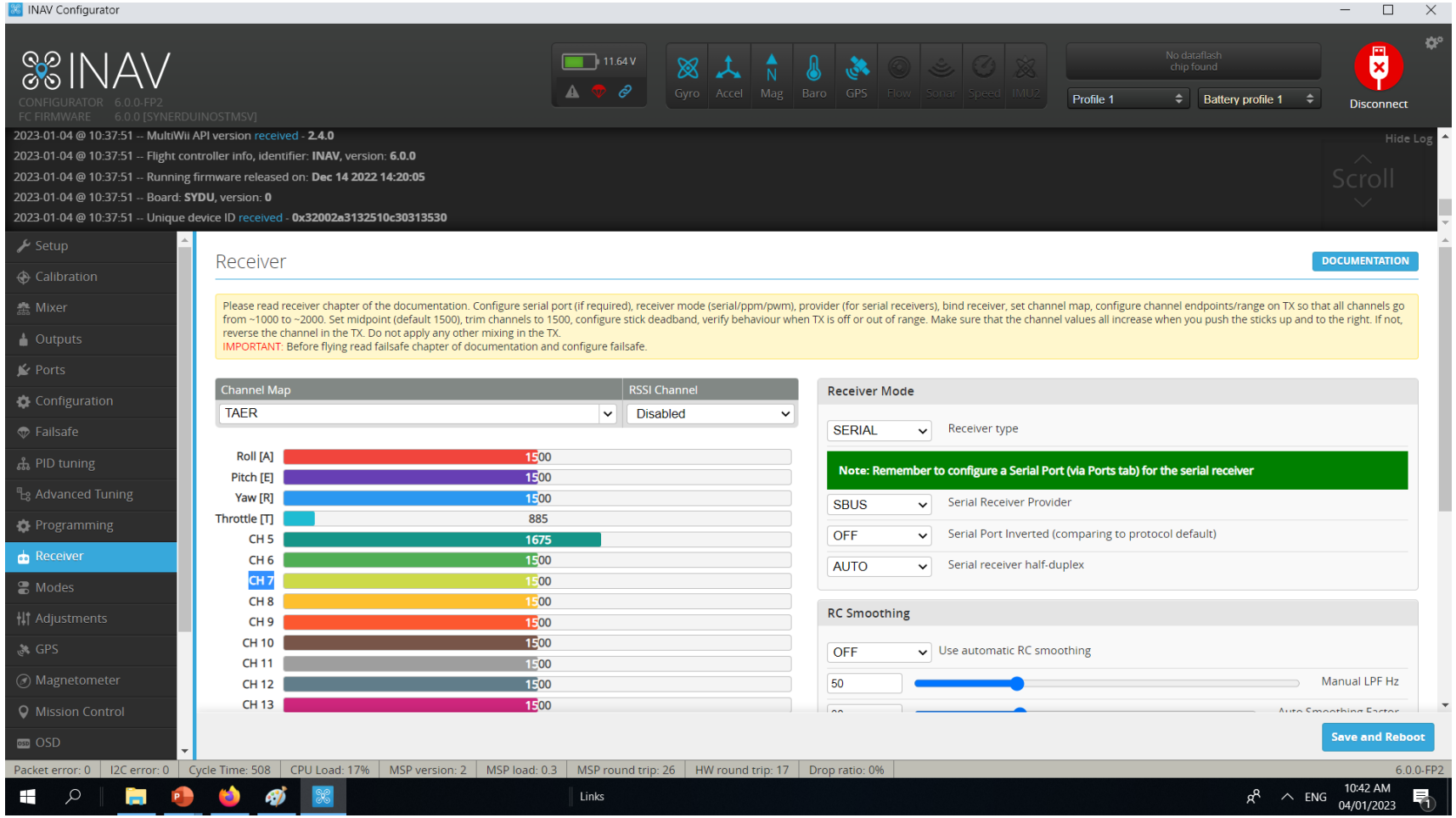


Gimbal

Receiver

We Wanted the RC Channel 7 and Ch 8 to be free for the gimbal function override

Which also doubles as a Pitch control if you wanted to adjust the angle of the camera

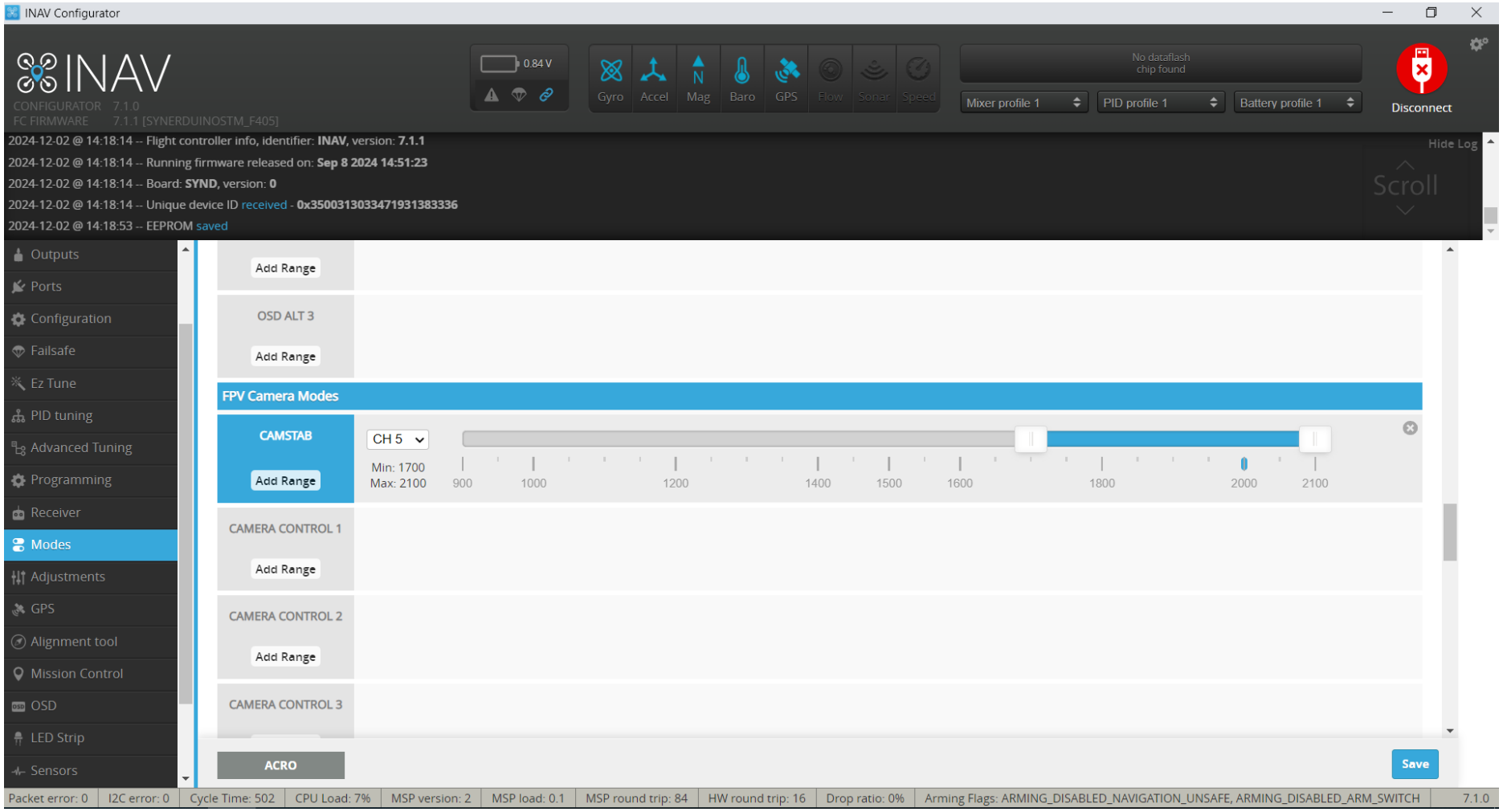


Gimbal

Modes

INAV 7 - INAV8

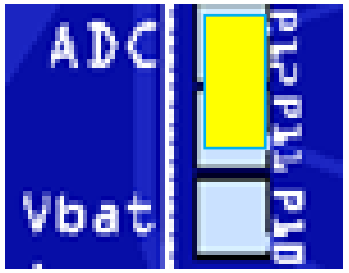
For Gimbal function to work you need to be active in the FPV camera Mode



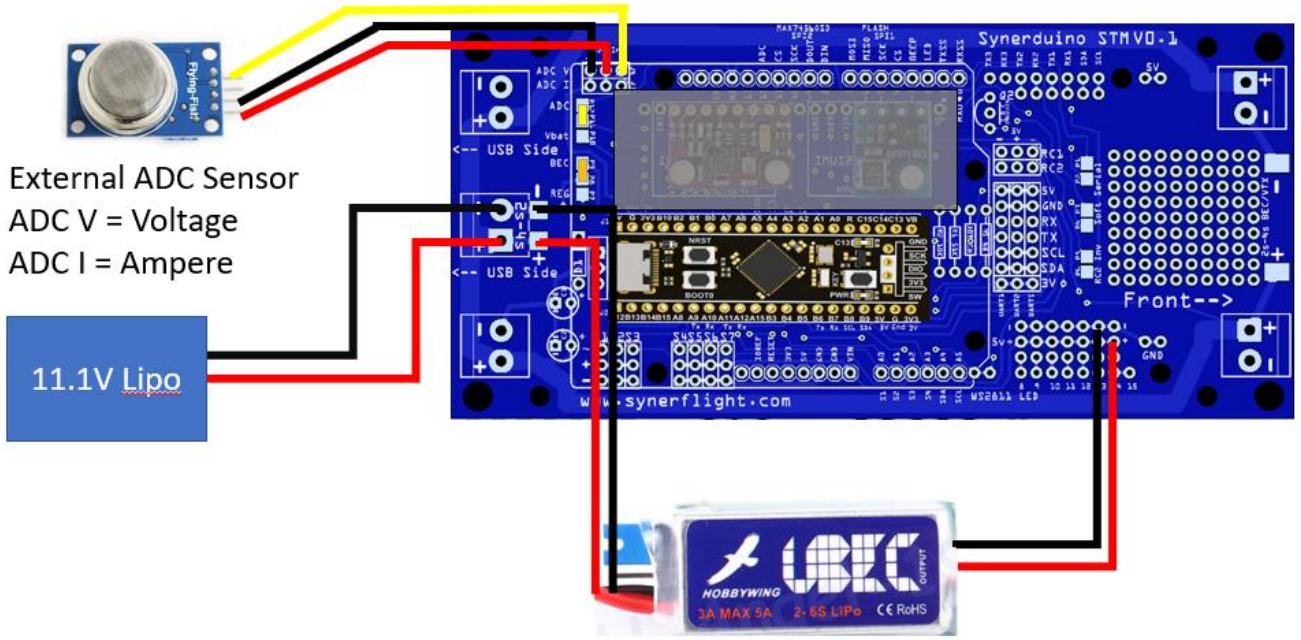
Data Analytic Intervention

As the name suggest if you know what you're looking for and you want the drone to intervene /Do something when it met a condition. doing the mission whether is triggering a payload or instrument when a potential positive value is match

For this sample we Hook up a sensor into the ADC V or ADC I



P12-P11 : ADC –this is use for having an external Analog sensor installed as it reads 0V-5V for Synerduino to convert to a Global Variable



UBEC source is same as ESC input and output is plug into any freely available 5V pin

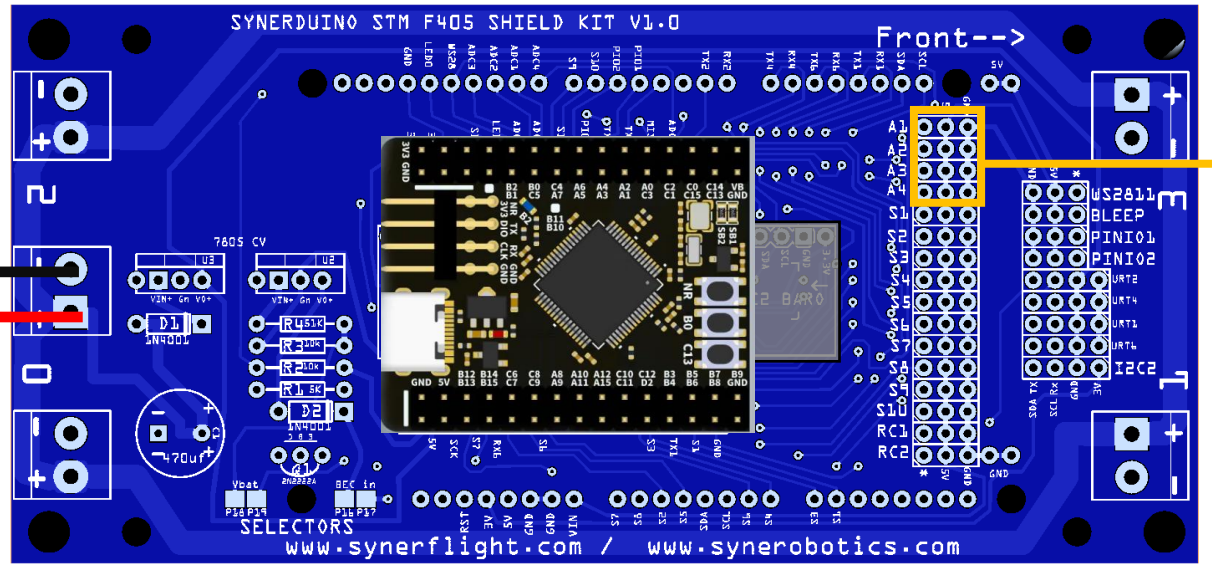
Data Analytic Intervention

Primary with 4 ADC pins

- Voltage
- Current
- RSSI
- Airspeed

External ADC Sensor

11.1V Lipo



This can be reassign for any Analog sensors outputting 0V -5V

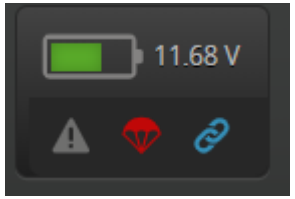
UBEC is Required for High current applications if it exceeded more than

- 3A for STMF405 and H743
- 1.5A for F411 of use when driving servos

BEC input requirements must be 5V

Data Analytic Intervention

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.

Note: its best to use a 2nd sensor acting as controlled to a specimen in order to calibrate its output

But in this case instead of the Battery its ADC sensor data we would input or Companion Controller input

This would define your signal point on where to trigger.

ADC V (Battery Voltage Monitoring) could be Sensor 1
ADC I (Battery Current Monitoring) could be Sensor 1

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

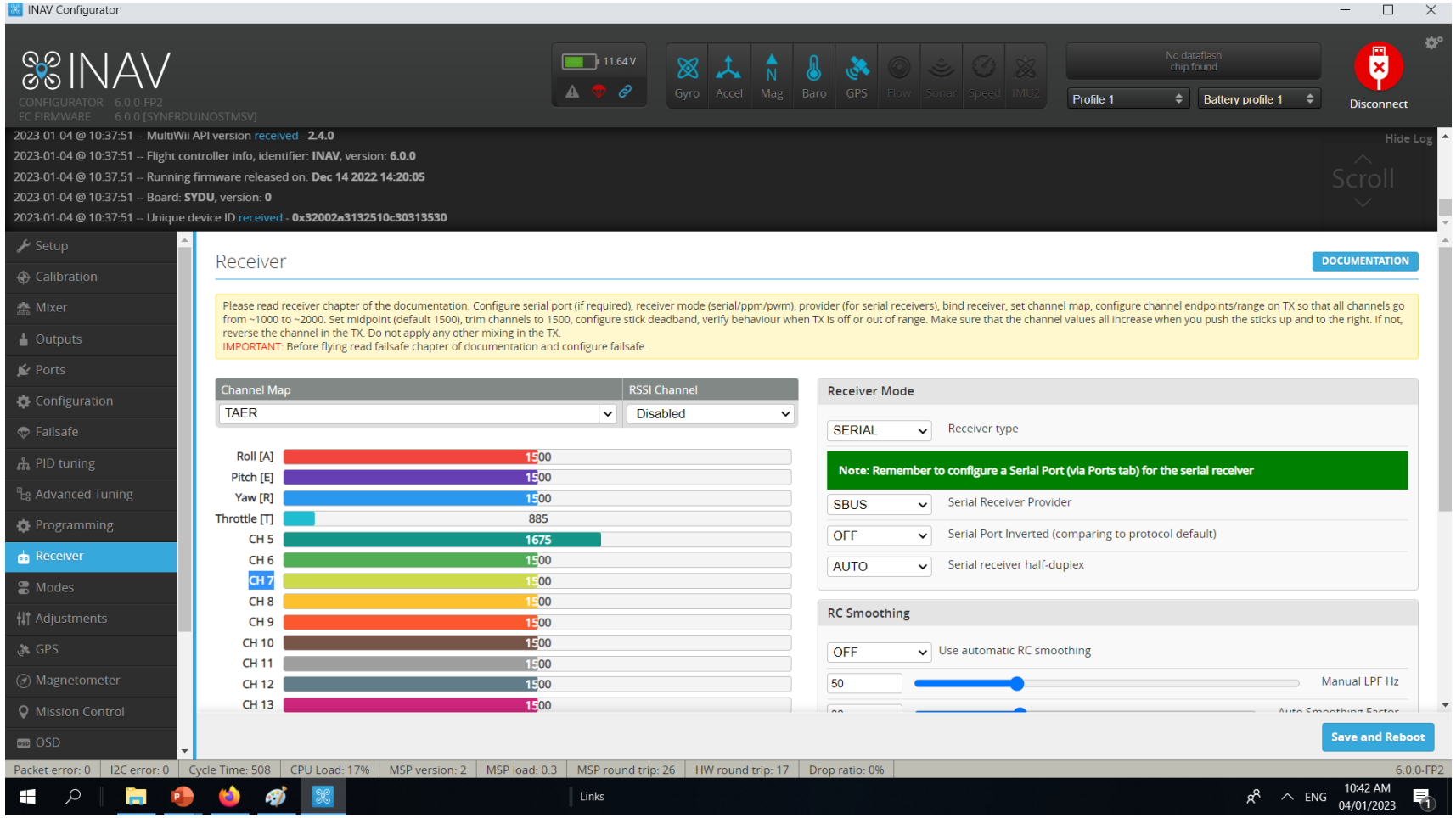
Battery Settings

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

Receiver

We Wanted the RC Channel 7 to be free for the Mode Trigger this would be active when a sensor value is met



Data Analytic Intervention

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

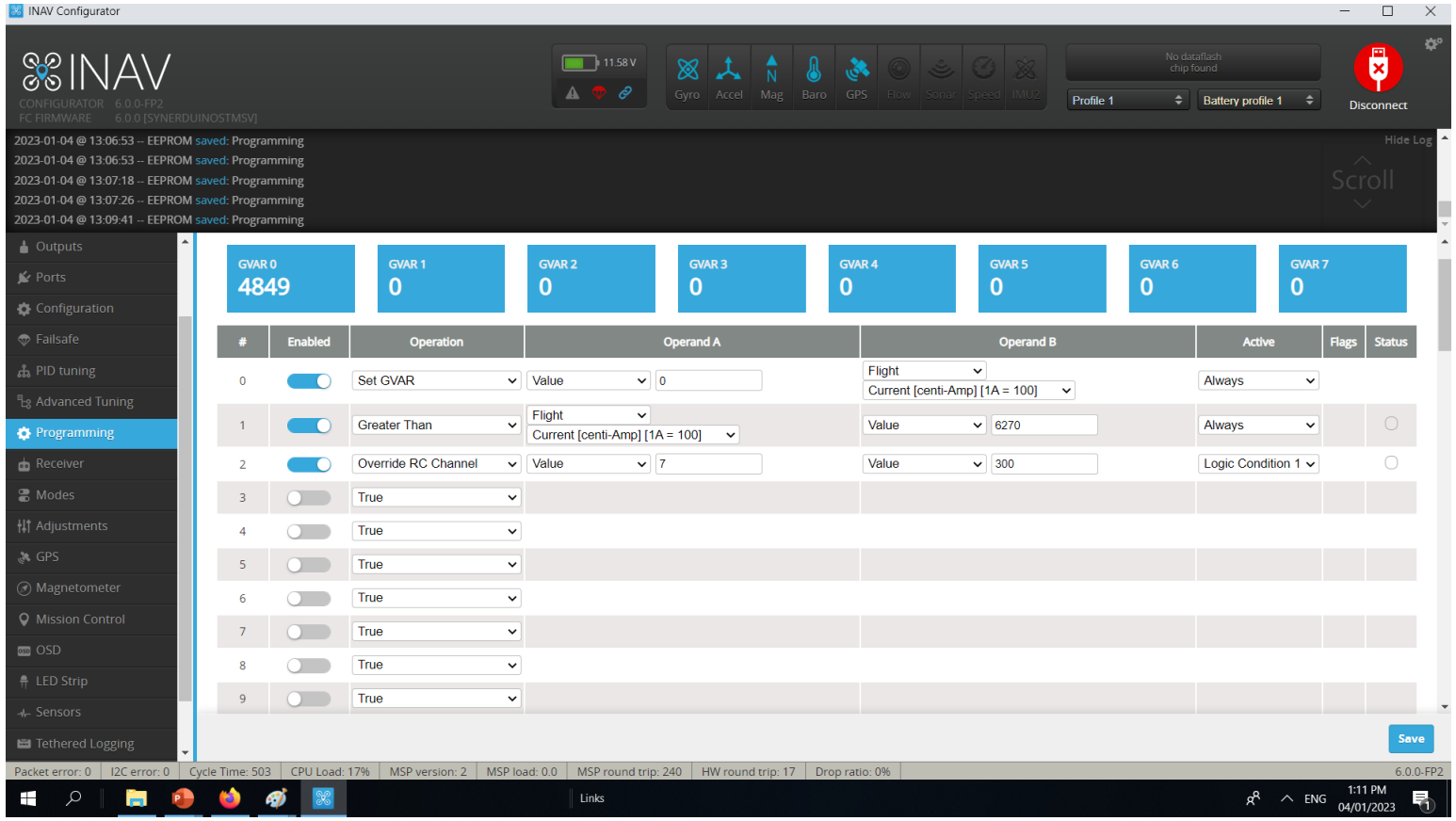
First we need to visualize the Current input in this case GVAR0 4849

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

Operand B can be change depending on the sensor 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

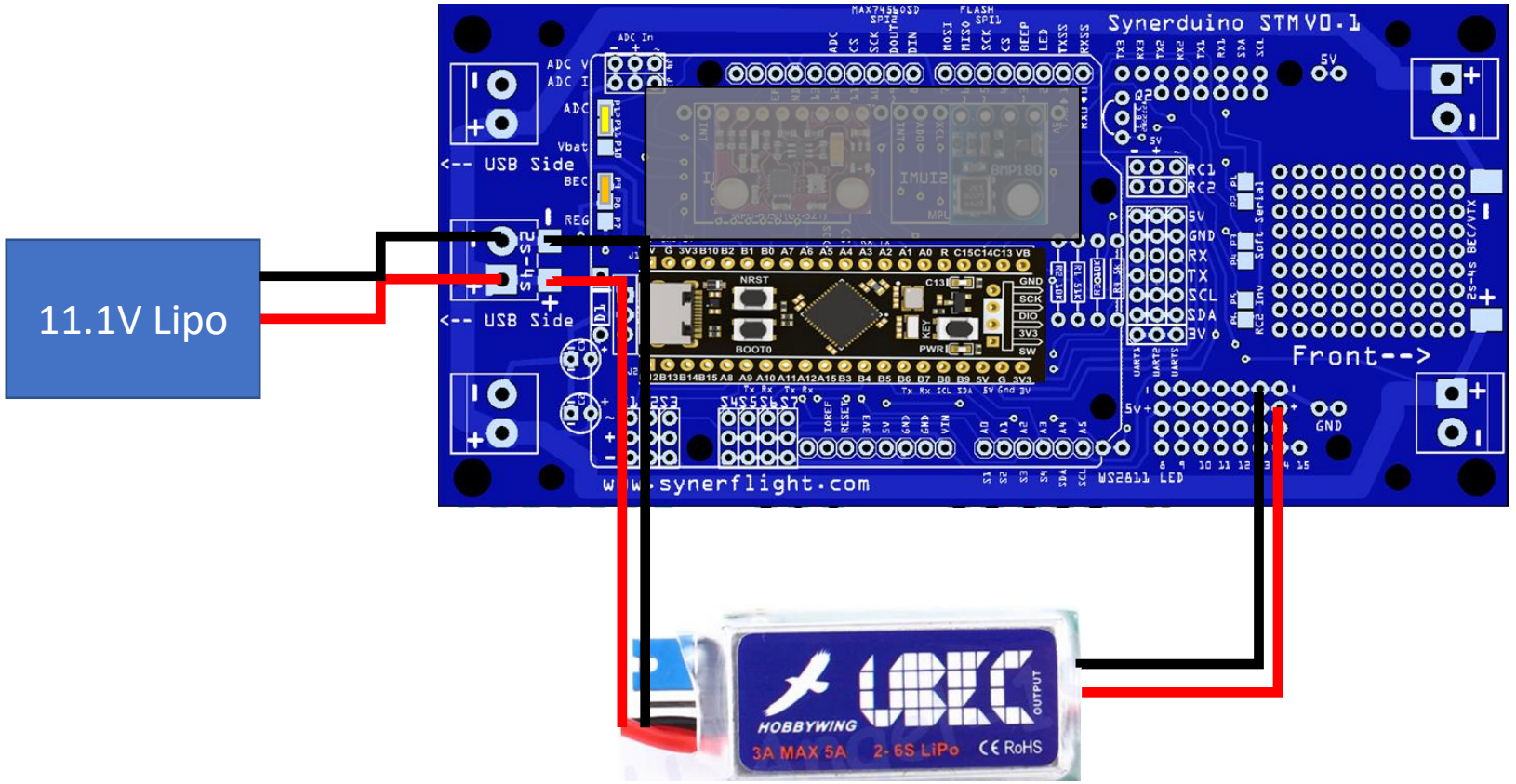


Companion board

There are several reason you may add a companion controller or computer to perform tasks separate from the Synerduino but need to be able to communicate with it to perform fight modifying action or Data Logging or even extended sensor array that is more than the standalone synerduino board physically accommodate

For Companion board add on options there are several Levels of installation

- ADC – Analog 0V-5V input
- Sbus/PWM – RC PWM 50 hz
- Serial – MSP Telemetry
- PWM Output



UBEC may be needed to add extra current to power a companion board source is same as ESC input and output is plug into any freely available 5V pin

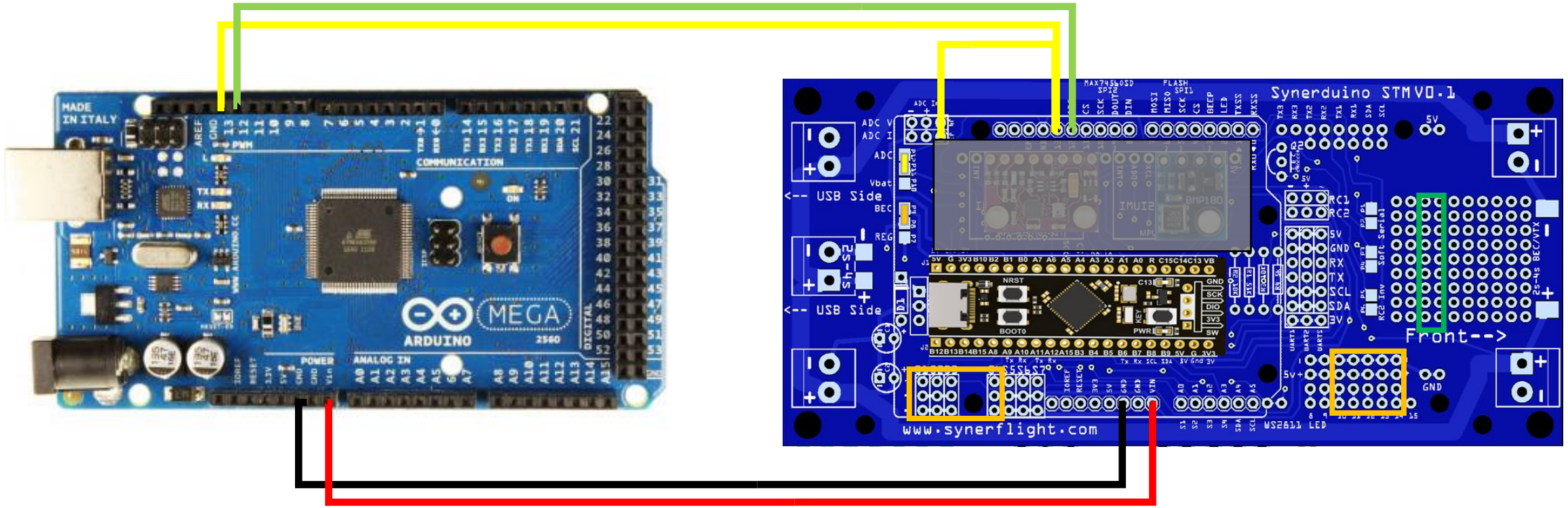
ADC Companion

F411 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 Mega

- D12 (ADC V) D13 (ADC I) D12 is align to the Synerduino V0.1 board ADC pin directly
- Analog pins A10 ,A11 ,A12,A13 , A14 are free for Sensor inputs
- DIY prototyping segment are also free for other Digital pins directly below it
- Motor PWM pins are also connected to the Analog pin S1-A0 S2-A1 S3-A2 S4-A3 as feedback loop to read pwm data off the motor



ADC Companion

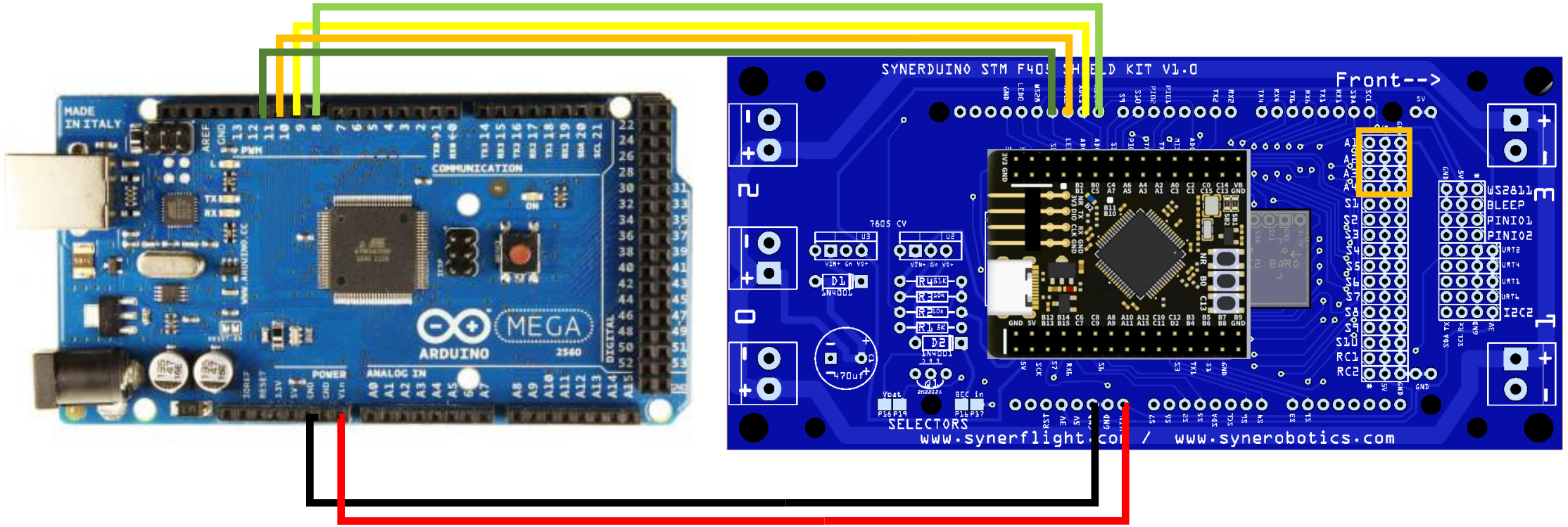
F405 and H743 The is useful for analog base inputs

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.

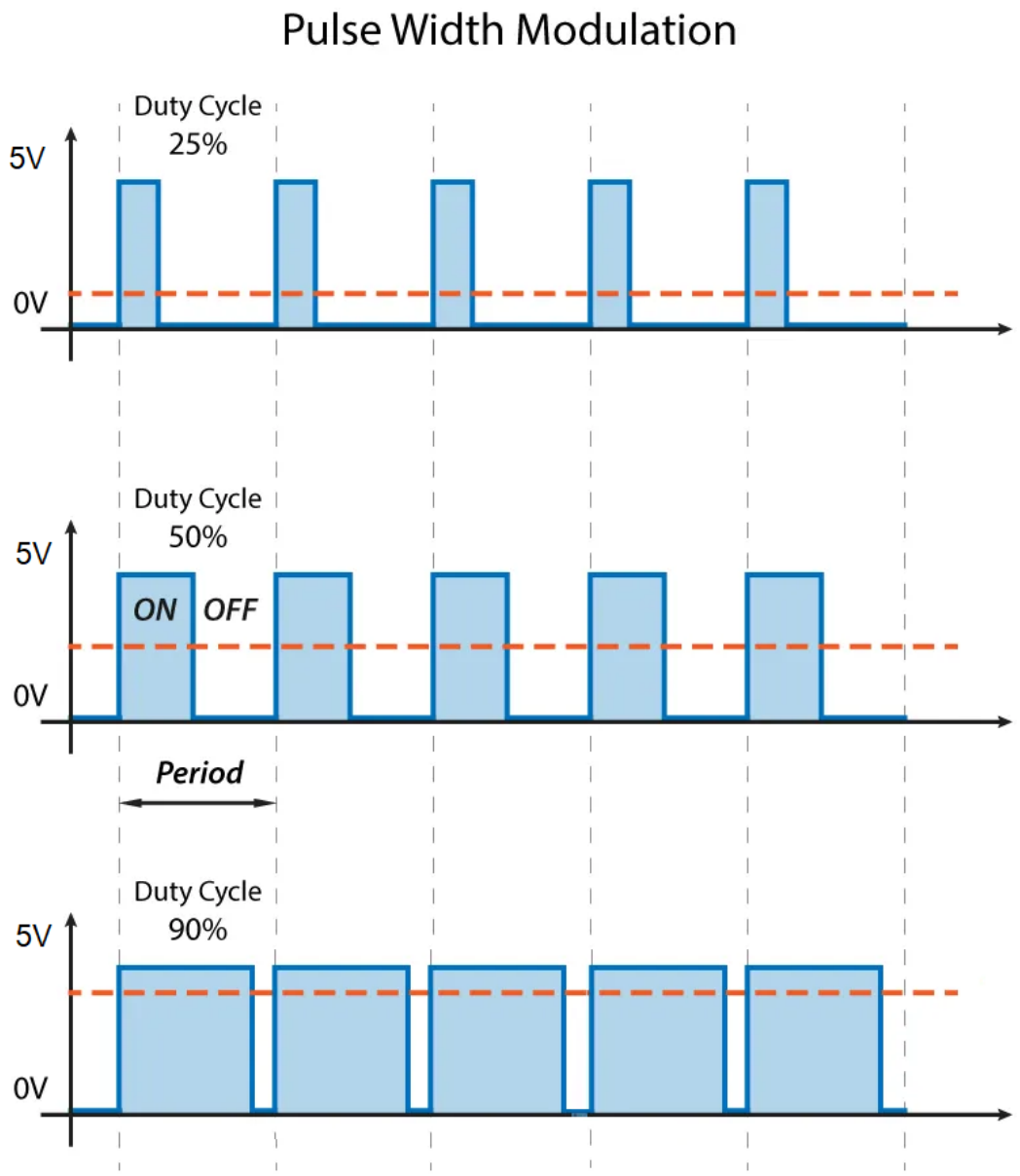
Arduino Mega

- Analog pins A8 ,A9 ,A10,A11 are free for Sensor inputs
- DIY prototyping segment are also free for other Digital pins directly below it

Note: pls be aware of the ADC pins Location they may Vary from board to board



ADC Companion



The average voltage depends on the duty cycle, or the amount of time the signal is ON versus the amount of time the signal is OFF in a single period of time.

This works with most Arduino PWM motor driver scripts

SBUS/PWM Companion

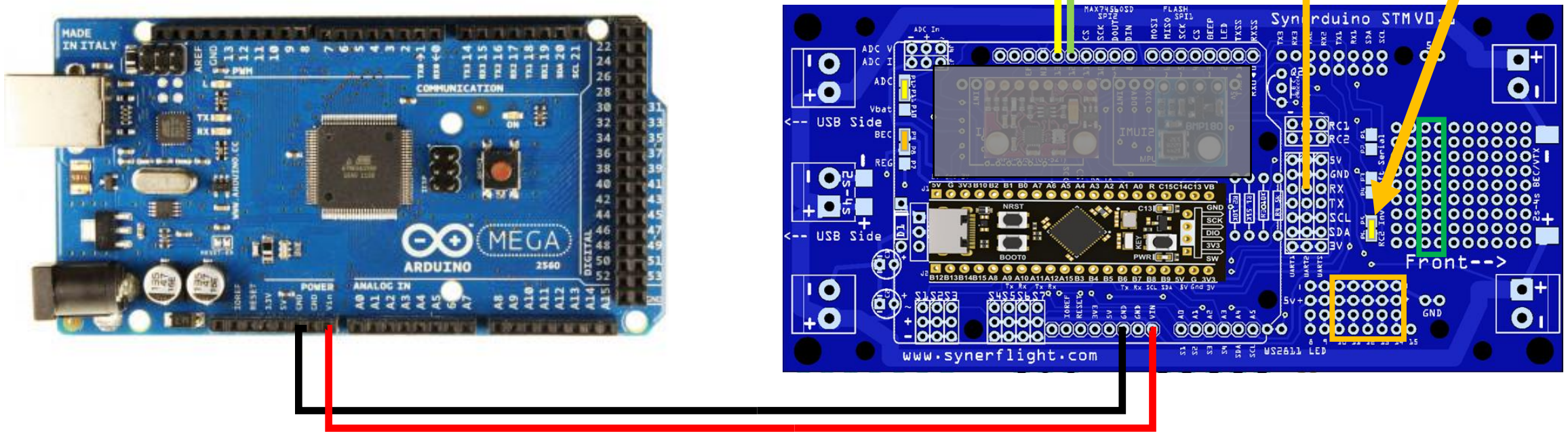
In the world of Robotics /RC /Drones
There is always a standard of PWM called the RC PWM its set at 50hz
And range from 1000us to 2000us
Of pulse this controls the value of the servos

This is said to be useful as interface when you need to have a semi digital range you want to instruct vehicle to do

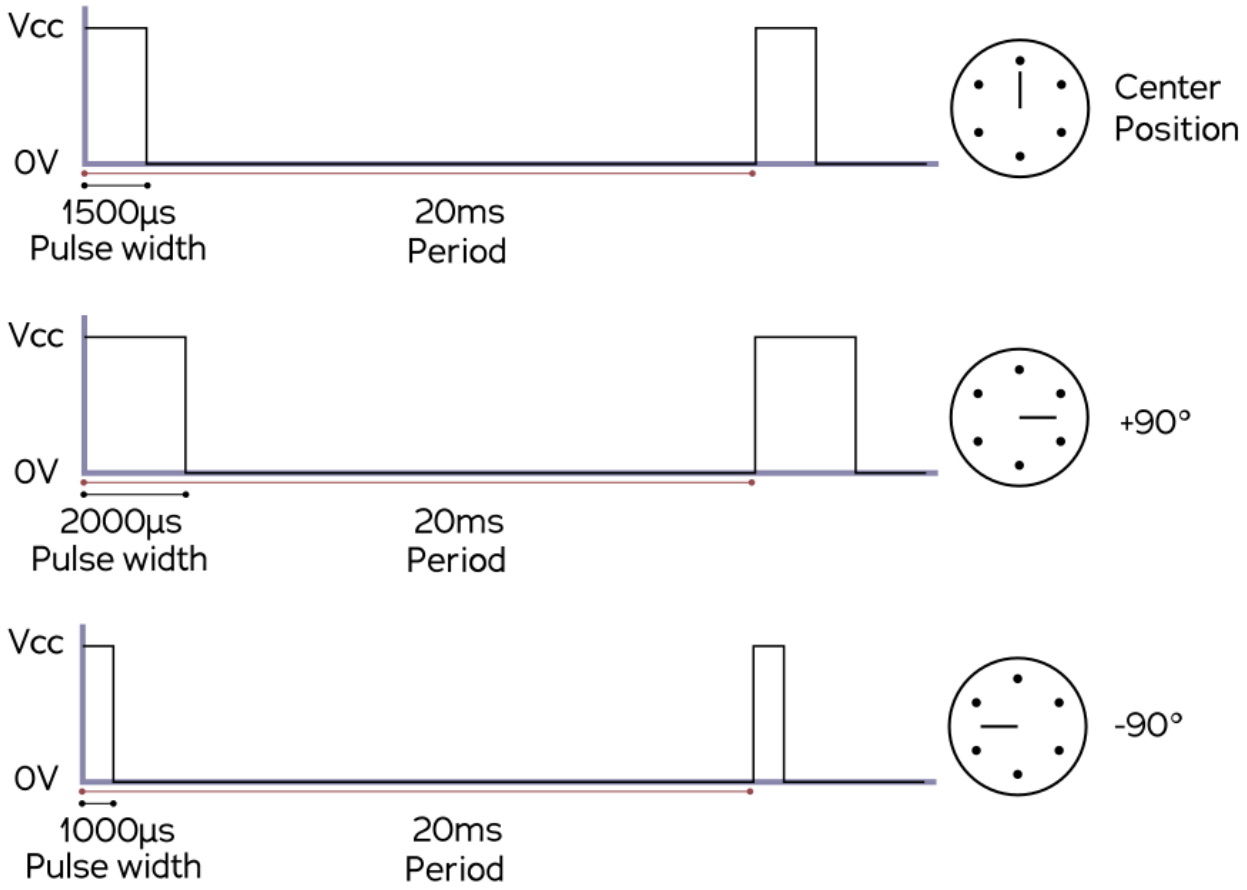
- Analog pins A10 ,A11 ,A12,A13 , A14 are free for Sensor inputs
- DIY prototyping segment are also free for other Digital pins directly below it
- Motor PWM pins are also connected to the Analog pin S1-A0 S2-A1 S3-A2 S4-A3 as feedback loop to read pwm data off the motor

This Setup requires an SBUS/PPM/PWM converter

- Arduino D12 Ch7 (ADC V)
- Arduino D13 Ch8
- RC Inv must be connected



SBUS/PWM Companion



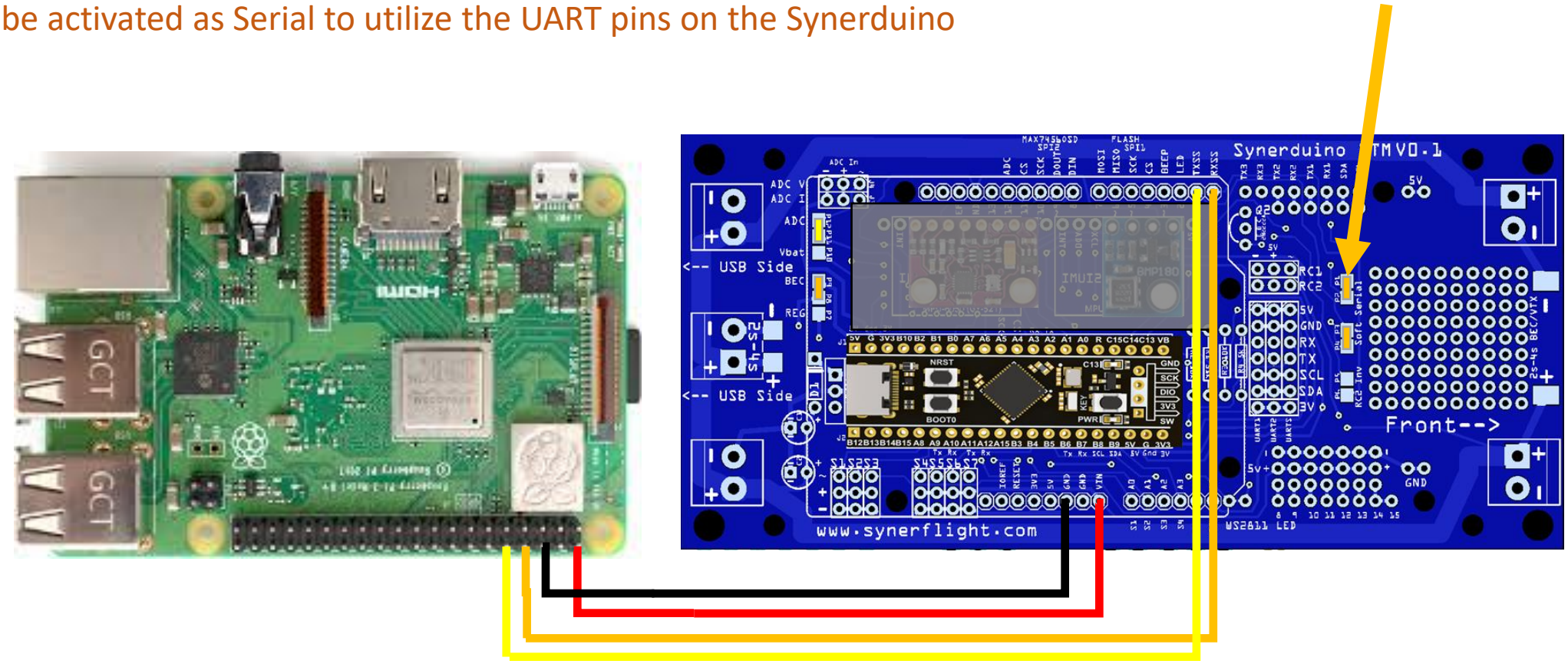
As RC servo standard this can be taken as an advantage for multiple data points for inputs

This works on Most Arduino Servo Scripts

Serial Companion

This is mostly done with Companion computer boards like Raspberry pi , Jetson Nano for higher levels of processing and applicable for larger vehicle platforms that can carry an extra onboard weight
Raspberry Pi UART is P8 TX(GPIO15) P10 RX(GPIO16)

Soft serial must be activated as Serial to utilize the UART pins on the Synerduino



Note: the Raspberry Pi 3 does come with a bluetooth an alternative Bluetooth connection can be done should weight and size is a concern and use the companion board as ground station instead
you can also utilize the other GPIO pints for the ADC input as well with raspberry pi

GPIO Companion

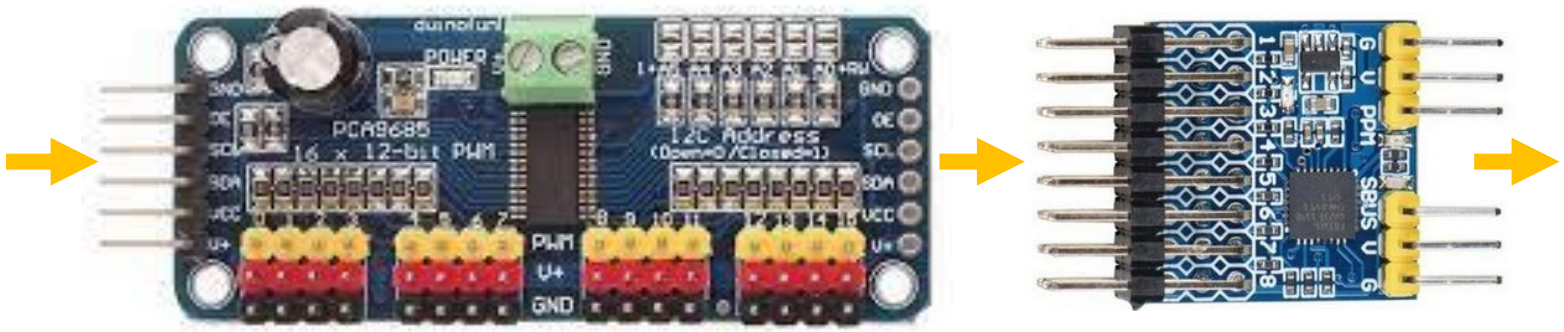
Raspberry Pi 3 Model B (J8 Header)

GPIO#	NAME			NAME	GPIO#
	3.3 VDC Power	1	2	5.0 VDC Power	
8	GPIO 8 SDA1 (I2C)	3	4	5.0 VDC Power	
9	GPIO 9 SCL1 (I2C)	5	6	Ground	
7	GPIO 7 GPCLK0	7	8	GPIO 15 TxD (UART)	15
	Ground	9	10	GPIO 16 RxD (UART)	16
0	GPIO 0	11	12	GPIO 1 PCM_CLK/PWM0	1
2	GPIO 2	13	14	Ground	
3	GPIO 3	15	16	GPIO 4	4
	3.3 VDC Power	17	18	GPIO 5	5
12	GPIO 12 MOSI (SPI)	19	20	Ground	
13	GPIO 13 MISO (SPI)	21	22	GPIO 6	6
14	GPIO 14 SCLK (SPI)	23	24	GPIO 10 CE0 (SPI)	10
	Ground	25	26	GPIO 11 CE1 (SPI)	11
30	SDA0 (I2C ID EEPROM)	27	28	SCL0 (I2C ID EEPROM)	31
21	GPIO 21 GPCLK1	29	30	Ground	
22	GPIO 22 GPCLK2	31	32	GPIO 26 PWM0	26
23	GPIO 23 PWM1	33	34	Ground	
24	GPIO 24 PCM_FS/PWM1	35	36	GPIO 27	27
25	GPIO 25	37	38	GPIO 28 PCM_DIN	28
	Ground	39	40	GPIO 29 PCM_DOUT	29

Attention! The GIPO pin numbering used in this diagram is intended for use with WiringPi / Pi4J. This pin numbering is not the raw Broadcom GPIO pin numbers.

http://www.pi4j.com

The P3(GPIO8) & P5 (GPIO9) can be utilize to drive an servo module which is run thru an converter then feeds the Sbus

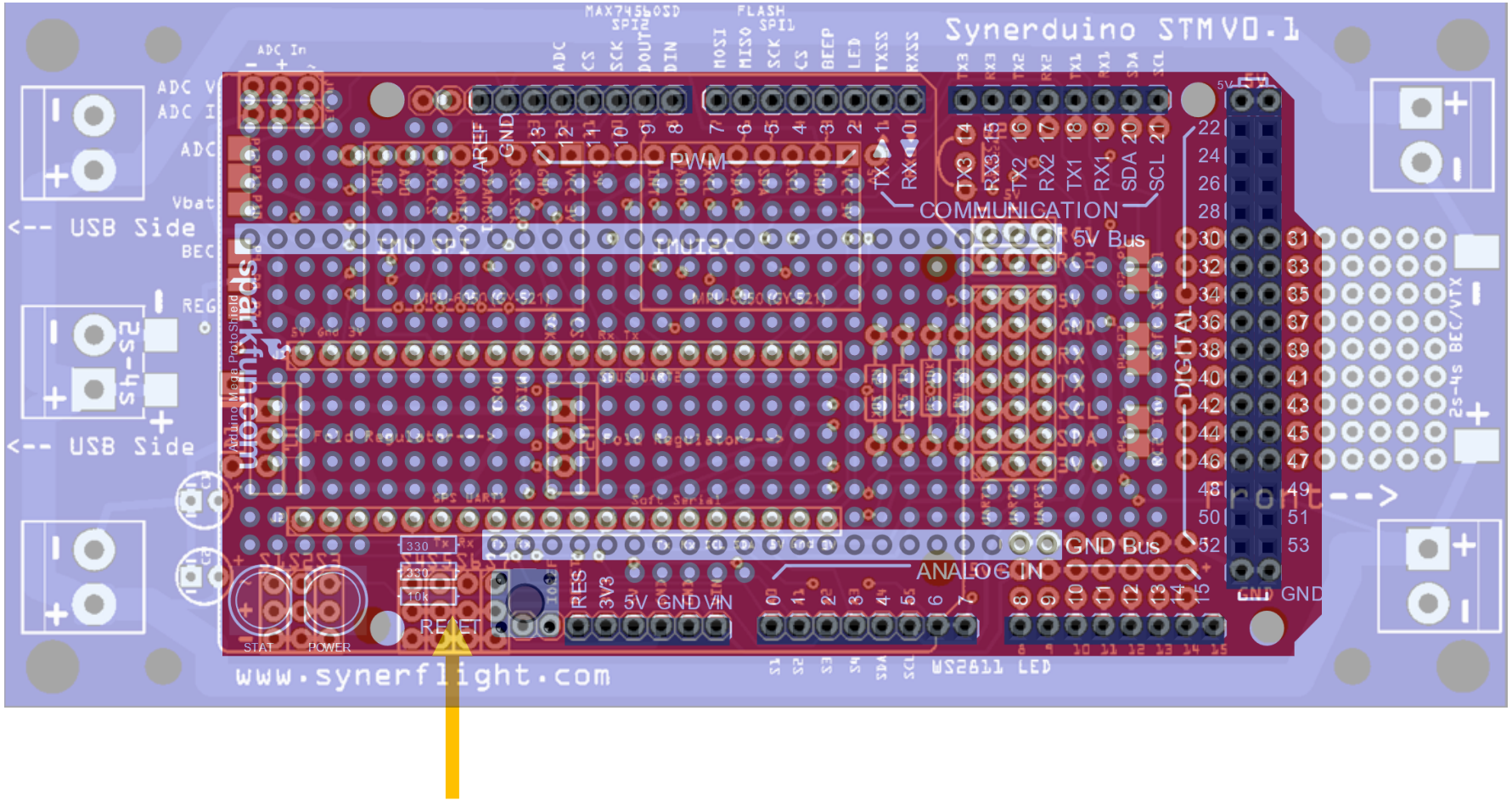


Note: Apart from serial the GPIO pins of the raspberry can also work in conjunction of the 12bit i2C servo module and Sbus/PPM/PWM Converter

SYNERDUINO STM and PROTOTYPING BOARDS

Since Synerduino is design as a shield board with an Arduino footprint its self means its compatible with a wide range of other shield prototyping board as well making it easy to fit custom or add on hardware reducing the need of additional wires and connectors

Male Headers are installed on the bottom of the board as required.



Prototyping Shield are to be place on the under side of the board

Final note: Program Logic is a very useful tool for Drone Vehicles it allows users with a wide flexibility to configure their Drone Vehicle's action and utilization of add on hardware as well . This sample is just a small slice of what can be done

Caution: Flight modification Flight modes entered in the Logic should be carefully studied and when testing fly slow 2 m/s and low 2meters both on missions or manually . Please attach tether when possible.

Avoid putting any Logic on Channel use by Arm/Disarm mode (this is your emergency shutoff switch)

RTH , Navigation modes , Arm/Disarm, Override (Ch1,2,3,4,5)

Multiple lines of code can also be stacked to form a instruction set for your vehicle which could accommodate companion board and daisy chaining of multiple synerduino boards