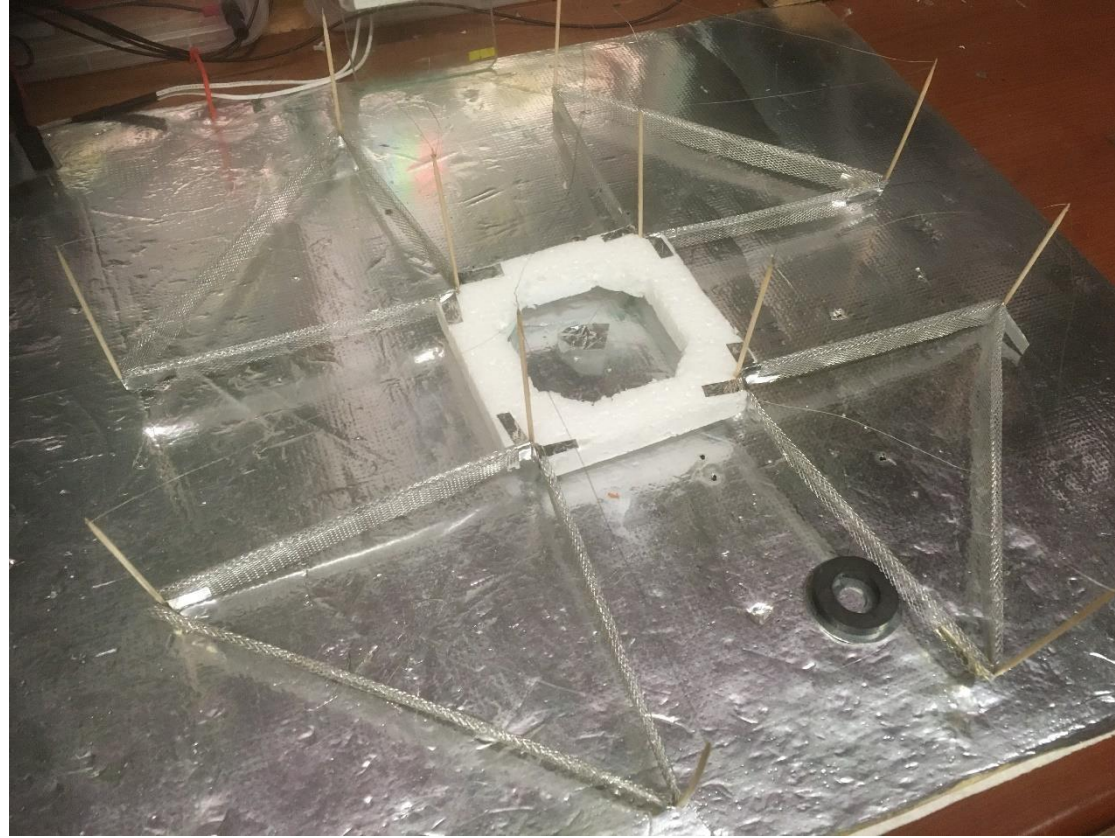


Simple Multi-Axis Command and Control of Ionic Lifter using ESP32



Christopher Jones January 2021

Introduction

This document presents a practical design implementation for a multi axis remote controlled ionic lifter.

The ionic lifter shown makes use of an octagonal shaped geometry for ease of control and has 3 degrees of freedom motion (pitch, roll, and lift). More degrees of freedom are possible, but only 3 are considered here.

The lifter control system is based around the ESP32 microcontroller and the lifter can be controlled manually using 4 potentiometers. The lifter may also be controlled remotely via Wi-Fi using a computer program and joypad control. General source code, block diagrams, and schematics are provided for others to replicate.

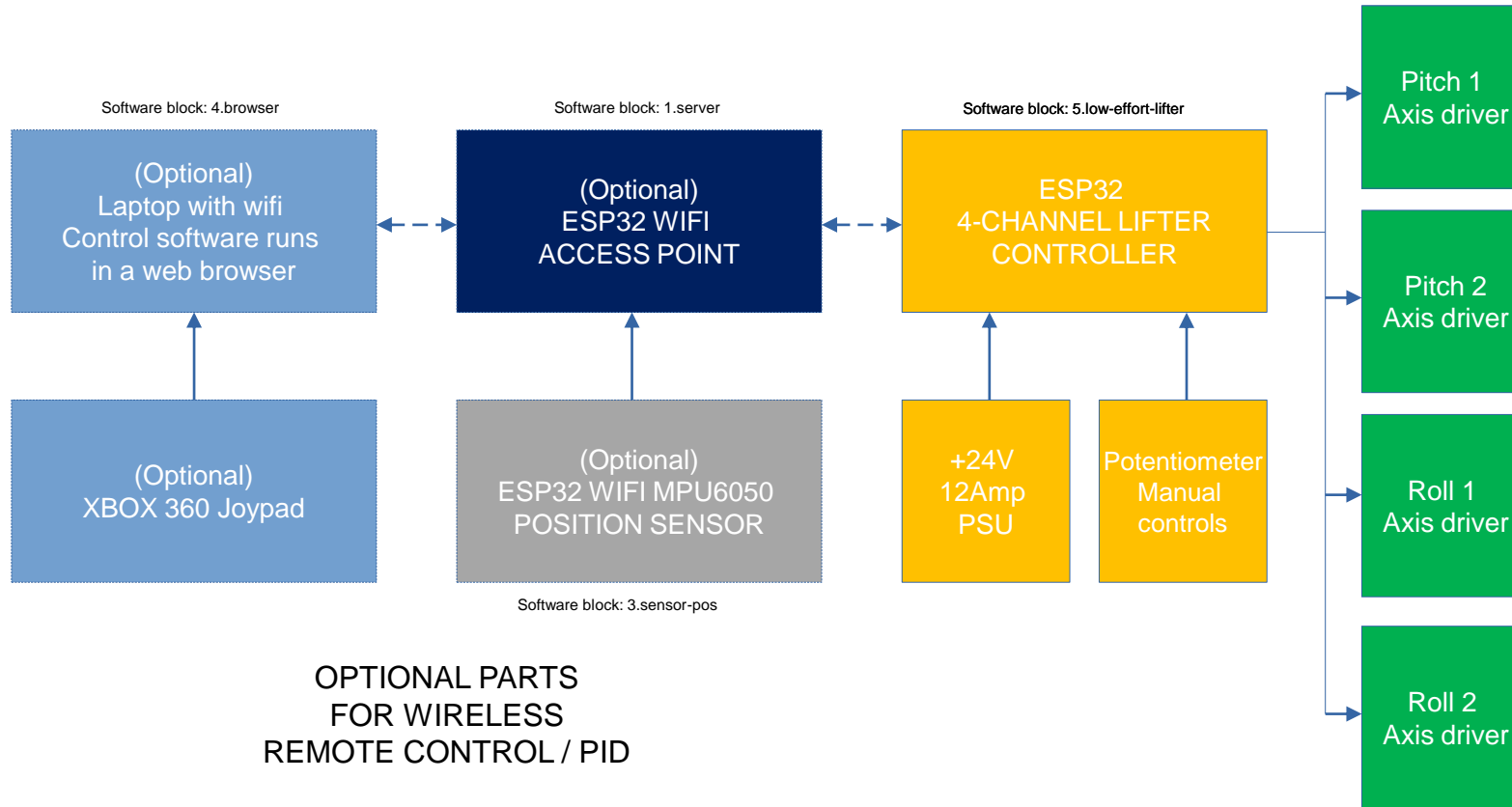
The intention is to present a system that is low cost and has a rapid assembly time, with low build effort. The main objective being to produce something that can be quickly and easily replicated and improved upon by any interested parties.

The general sprit of the project is explained in this youtube video:

https://youtu.be/C5L3u5OyNGA?list=PLiWshFxy5jDVE85_Qa8NKqKymudv9NQ1L&t=68

Warning: Lifters make use of high voltages and are a potential electric shock hazard. Please be careful when working with high voltages. EMC may also be a concern. Replication of this project is done entirely at your own risk.

Multi-Axis Command and Control Block Diagram



Shown here is a block diagram describing the main parts required for a 3 Degree of freedom (DOM) lifter.

The lifter has an octagonal geometry because this particular geometry greatly simplifies control. Pitch, roll, and lift are possible using four triangular ion thrusters (Axis drivers).

The axis drivers are driven using 100Hz PWM signals. The PWM for each axis can come from potentiometers connected to a 4 channel lifter controller (Manual control), or optionally as signals sent wirelessly over Wi-Fi using a dedicated command and control network.

For Wi-Fi control, an ESP32 may be configured as a wireless access point. This access point will also act as a router allowing position sensor data to be sent to a control computer, and control signals from the control computer to the 4-channel lifter controller.

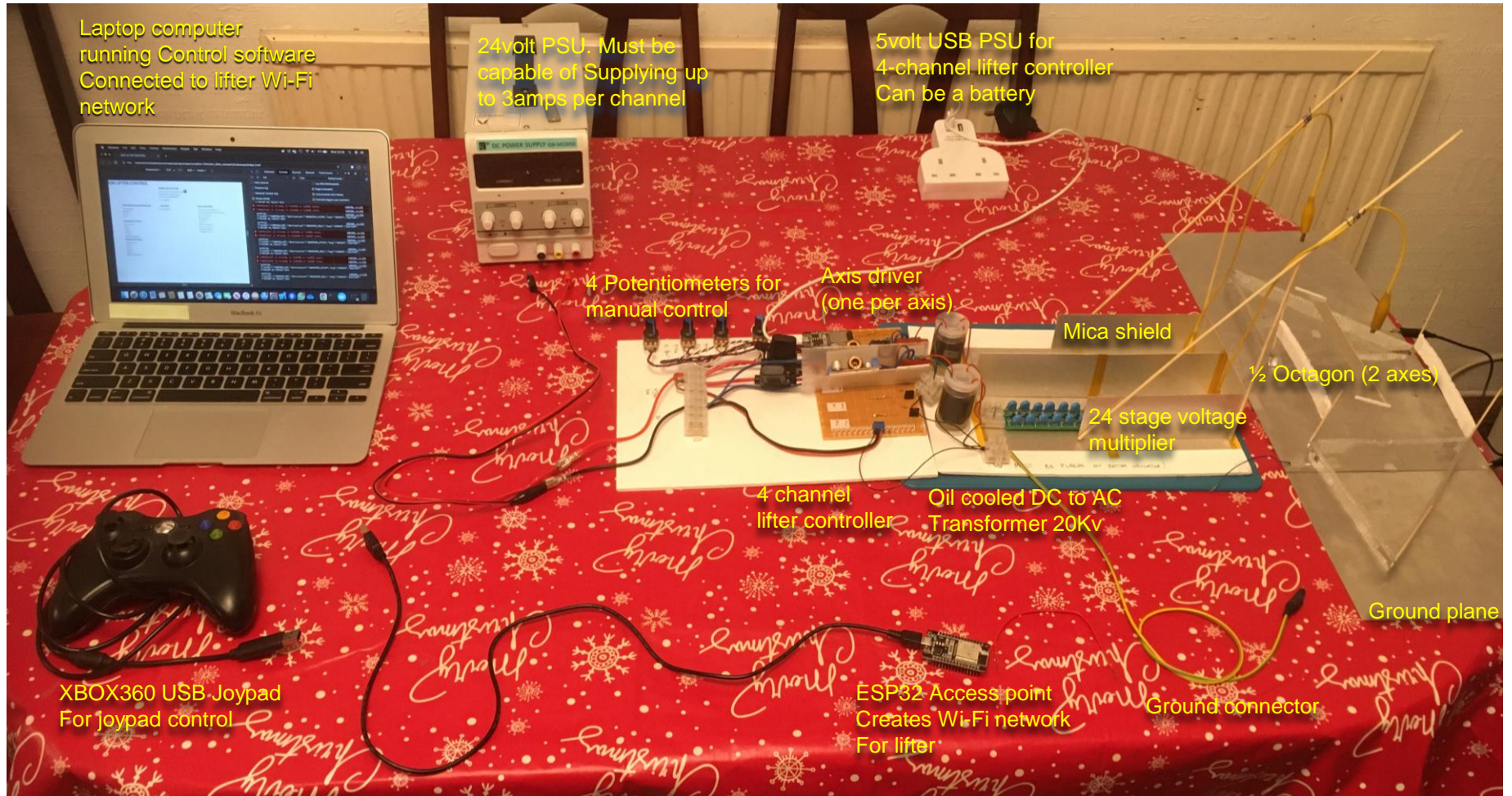
An ESP32 interfaced to an MPU6050 can be used to give lifter attitude data if it is mounted onto the lifter. It is likely that the lifter can lift the MPU6050 sensor, but not the ESP32 if its i2c interface and power are connected via 4 strands of 40AWG magnet wire.

A jQuery program runs in your internet browser, your computer must be connected to the wireless access point. The jQuery web application will display the potentiometer values from the 4-channel lifter controller, along with the lifter attitude. You may control the lifter from the program either using sliders in the software or an XBOX360 or similar joystick.

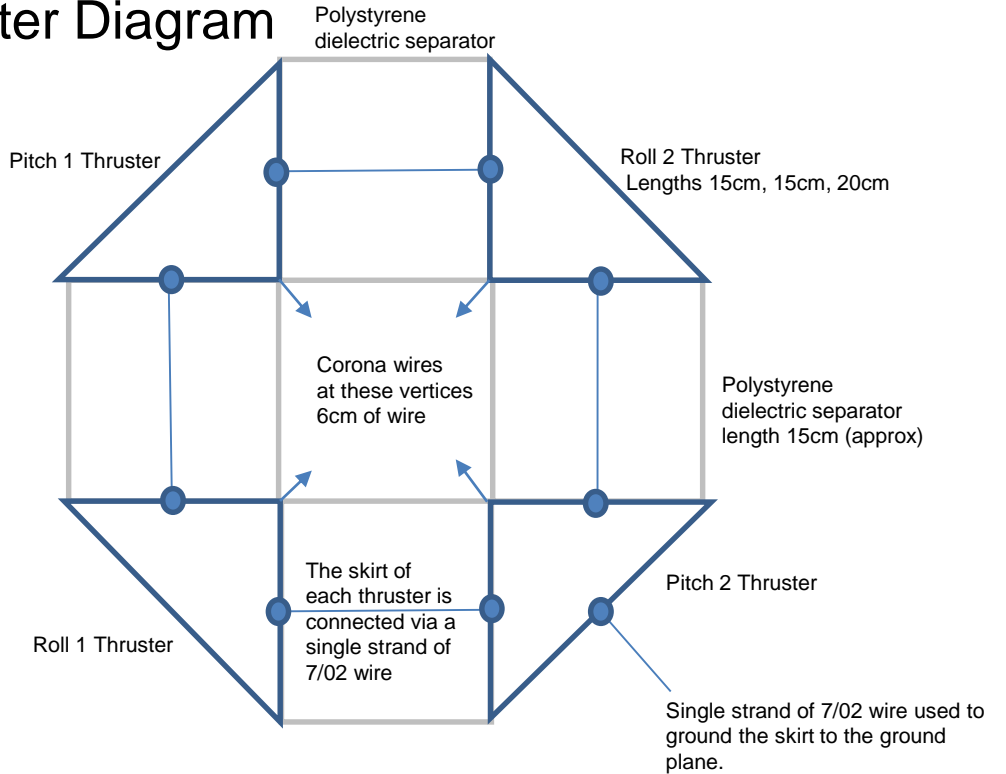
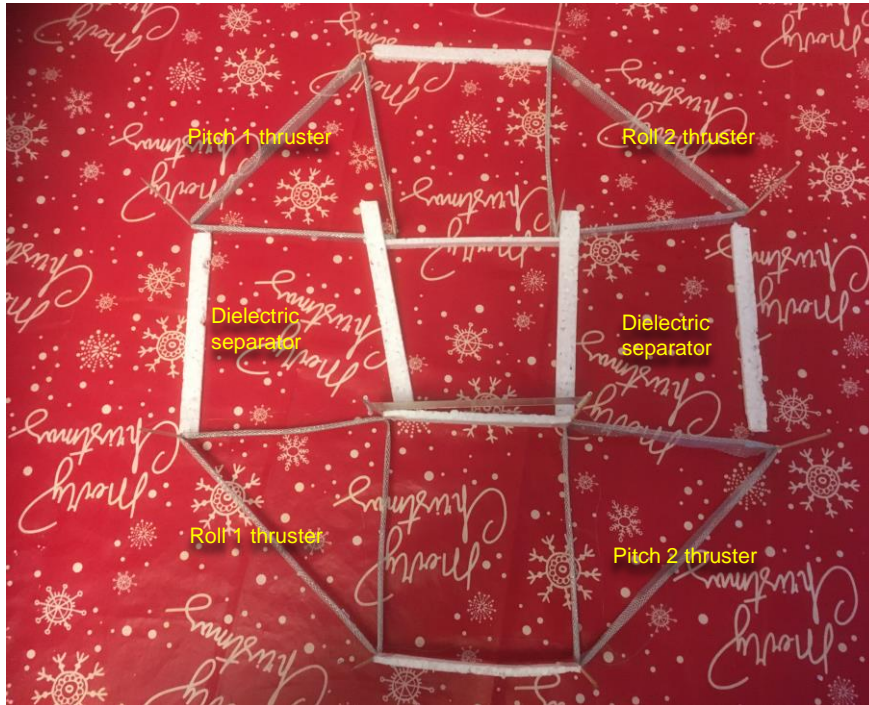
Code for all blocks is available here:

<https://github.com/cybernetic-research/ion-lifter-esp32-command-control-network>

Multi-Axis Command and Control



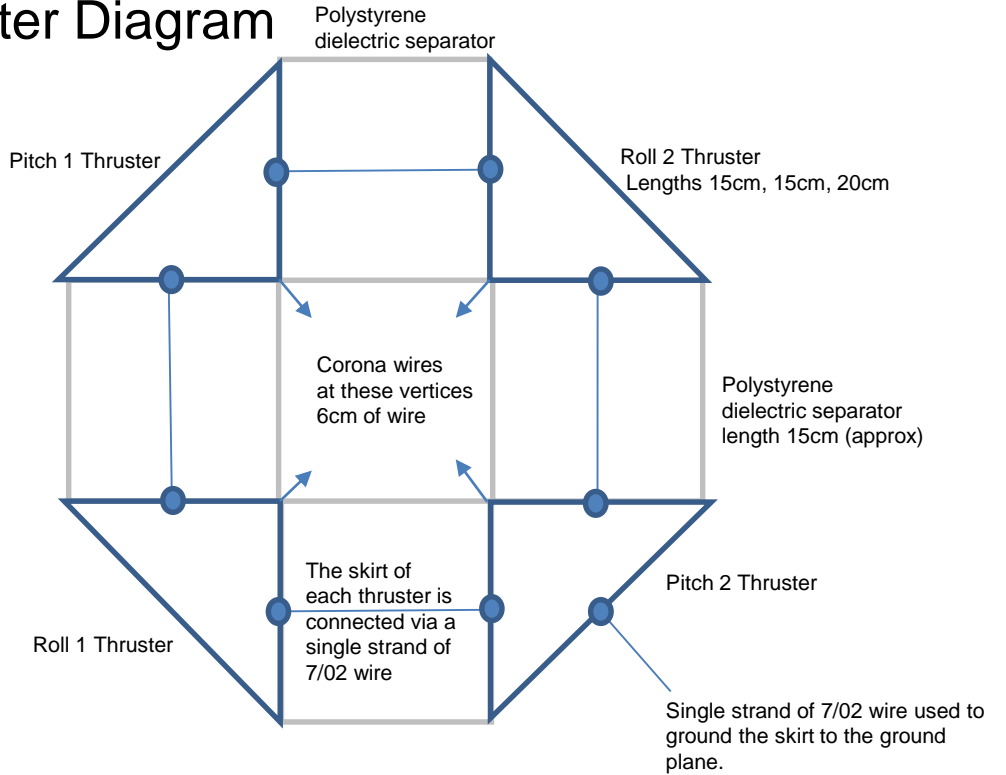
Octagon Lifter Diagram



- The lifter has an octagonal geometry because it lends itself to intuitive pitch roll thrust control.
- The construction is open frame as shown in the picture above
- The lifter is formed from 4 right angle triangle thrusters (total length 50cm each), coupled to each other via a lightweight polystyrene dielectric material.
- Separation distance between each thruster is 15cm. Separation is important because the electric field from one thruster corona can interfere with another thruster if it is too close.
- Total weight of the structure is assumed to be around 10grams, and must be kept as light as possible to maximise available thrust

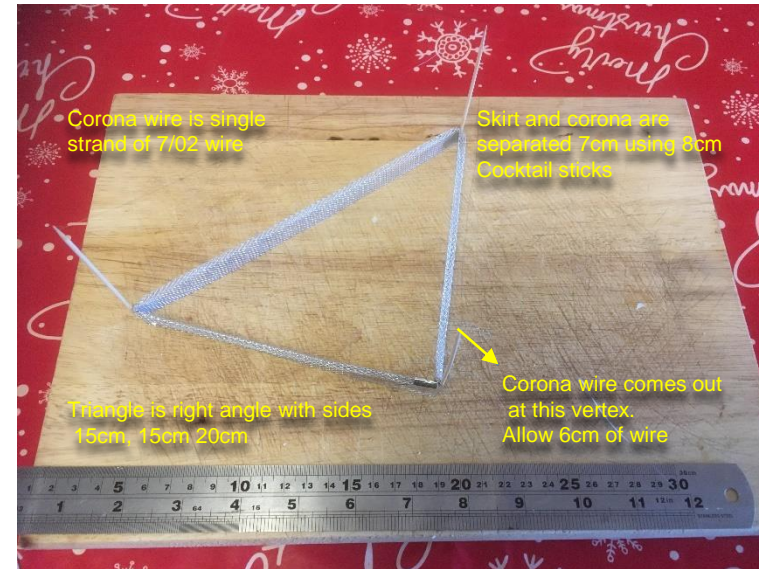
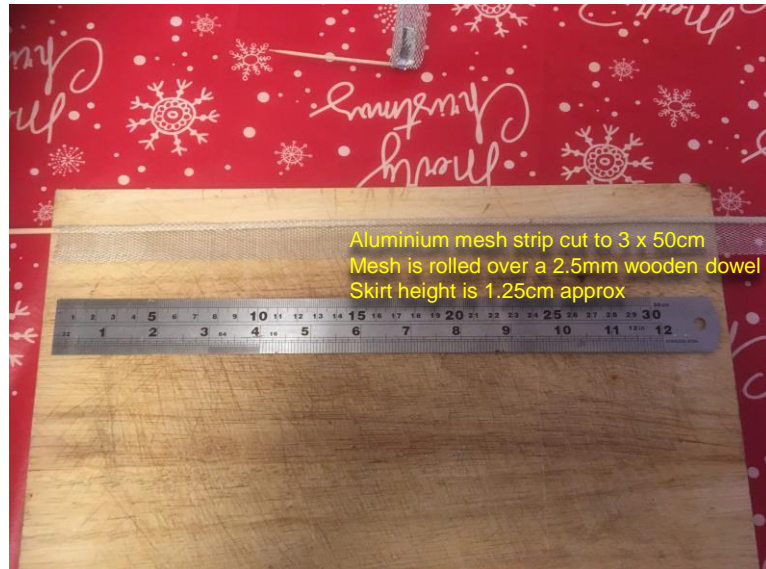
Be sure this wire is not free to float upwards and short against the corona wire.

Polystyrene
dielectric separator



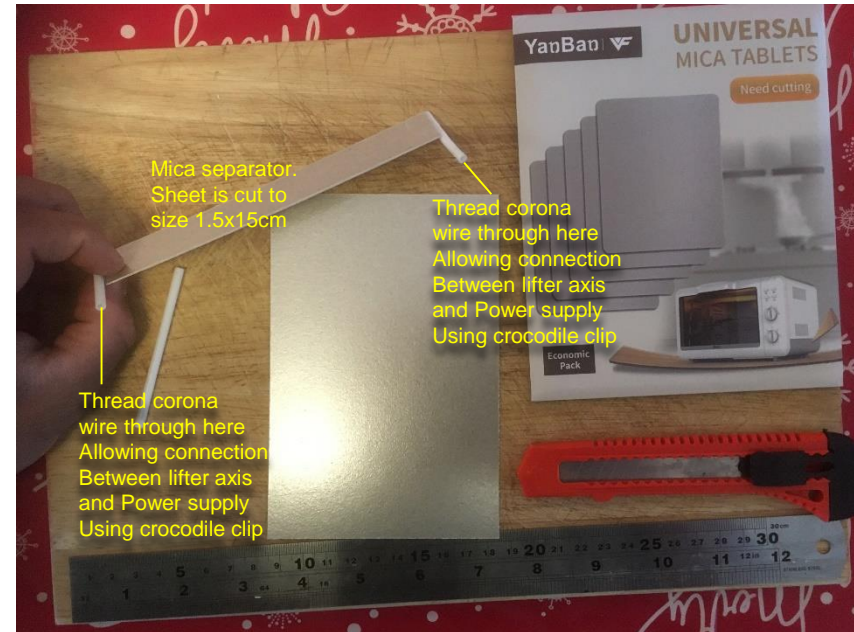
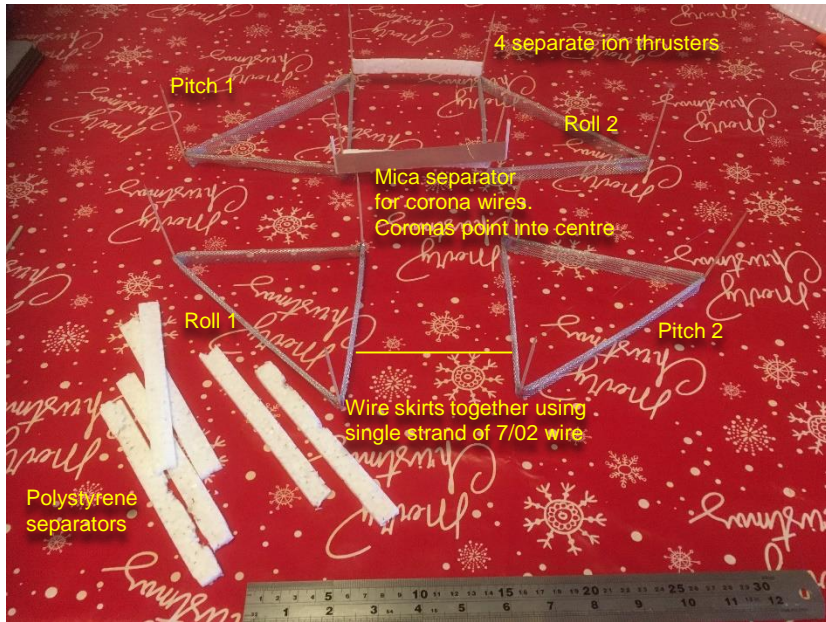
- Be sure this wire is not free to float upwards and short against the corona wire.**

Assembly of each ion thruster



- The octagon is made of 4 separate right angled triangle ion thrusters.
- Triangles have hypotenuse length 20cm and two 15cm lengths.
- Each right angle triangle is made of fine aluminium mesh cut to 3cm x 50cm. (You may use aluminium foil but it likely won't have the same rigidity as the mesh).
- The long side of the mesh is rolled over a 2.5mm wooden dowel to create a rounded edge.
- Mesh is folded into a triangle, to give 2 x 15cm sections and a 20cm section
- The skirt and corona are separated using cocktail sticks 8cm in length.
- The corona wire for each triangle is a single strand of 7/02 wire, or thinner bare wire (enamel copper wire does not work as well, even if it is thin).
- Triangles are connected to each other using two strips of expanded polystyrene 1 x 15 x 0.1 cm
- Each lifter triangle has its own separate power supply providing lift, pitch and roll control.

Assembly of octagon lifter

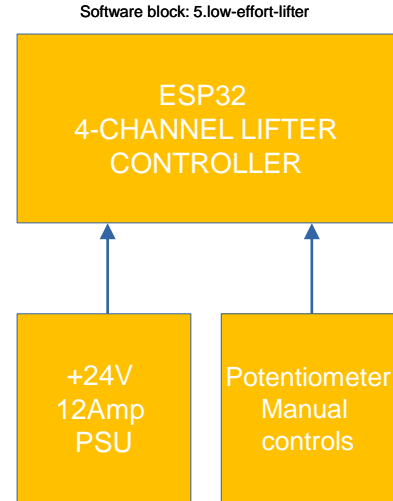


- The 4 separate right angled triangle lifters need to be joined together to form an octagon.
- Each thruster must be separated by 15cm using the polystyrene blocks as shown above
- The skirts need to be wired together using single strands of 7/02 wire. This is the ground connection.
- Corona wires for each thruster need to point into the centre of the octagon
- A mica separator is used to keep corona wires apart from each other

Lifter BOM

Item	Part	Quantity	Supplier	Notes
1	Aluminium mesh	1	Amazon / Bcreative	Cut 4 strips 3 x 50cm. Mesh hole size 1 x 2mm, part no M513. All lifter skirts are made of this because it gives rigidity
2	Expanded polystyrene	1	any	Cut 8 strips 1 x 15 x 0.1cm
3	Aluminium tape	1	any	Cut as appropriate (use sparingly to reduce weight)
4	Cocktail sticks	12	any	Important - must be 8cm long to prevent arcing at these voltages
5	Wooden dowel	1	any	2.5mm diameter, used to create a rounded edge on skirt facing corona
6	Electrical wire (single strand of 7/02)	4	any	4 strands of 7/02 wire of length 60cm each
7	Sellotape (scotch tape)	1	any	Cut as appropriate
8	Super glue	1	any	Use as appropriate
9	Mica sheet	1	Amazon	Yan Ban Universal Mica Tablets

ESP32 4-Channel Lifter controller Block Diagram



The 4 channel lifter is shown here as a block diagram. It is intended to be as simple as possible to replicate because it has a very low component count, and most of the hard work is done in software, which is available to anyone on GitHub. The ESP32 is programmed using the Arduino ESP32 libraries.

The controller is basically a circuit board with an ESP32 on it. The ESP32 generates a 100Hz PWM waveform as a drive signal to each of the 4 lifter axes. The duty cycle of the outputs is set by one of four potentiometers, one for each channel, allowing manual axis control.

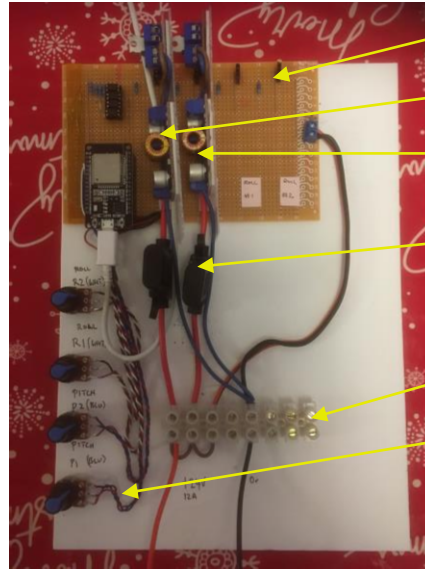
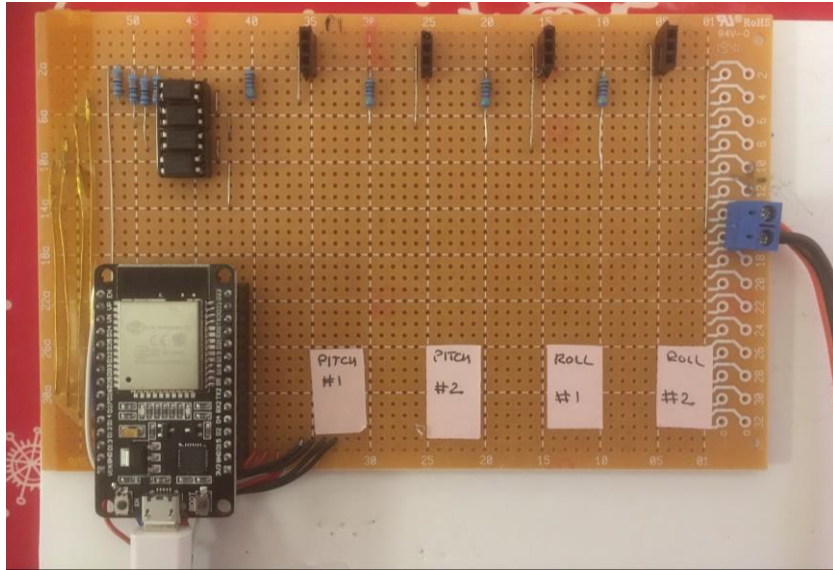
Opto isolators are used to electrically isolate the 3v3 PWM signals from the output stages, and prevent damage to the ESP32 chip. The only components needed to make this board are some resistors, connectors, optos and an esp32.

The output from the controller may also come from remotely via a wireless web socket using Wi-Fi. In this case the lifter may be remotely controlled using a joystick.

Code for this blocks is available here:

https://github.com/cybernetic-research/ion-lifter-esp32-command-control-network/tree/main/ion_lifter_network/5.low-effort-inverter/low-effort-inverter-100hz-pwm

ESP32 4-Channel Lifter controller



Space for roll 1 and 2 channels, add more channels if you want lateral thrust. A lifter is capable of 6DOF motion like a helicopter

Pitch 1 axis driver card (See axis drivers)

Pitch 2 axis driver card

3amp automobile fuses one per axis

Connection block

Potentiometers (50k linear)

- The 4 channel controller electronic design is kept as simple as possible with a minimal component count.
- Circuit board only contains ESP32, opto-isolators, and resistors. Connectors are provided for the individual axes, and +24v power to opto-isolators.
- The ESP32 can be used in manual or Wi-Fi remote control mode. Switching is done through remote control software running on a pc.
- In manual mode the controller reads 4 potentiometer inputs for each control channel and outputs a 100Hz PWM signal to the relevant channel (see wiring schedule)
- In remote control mode the controller will accept control signals from a computer program. The control demands may come from a joystick, or slider controls.

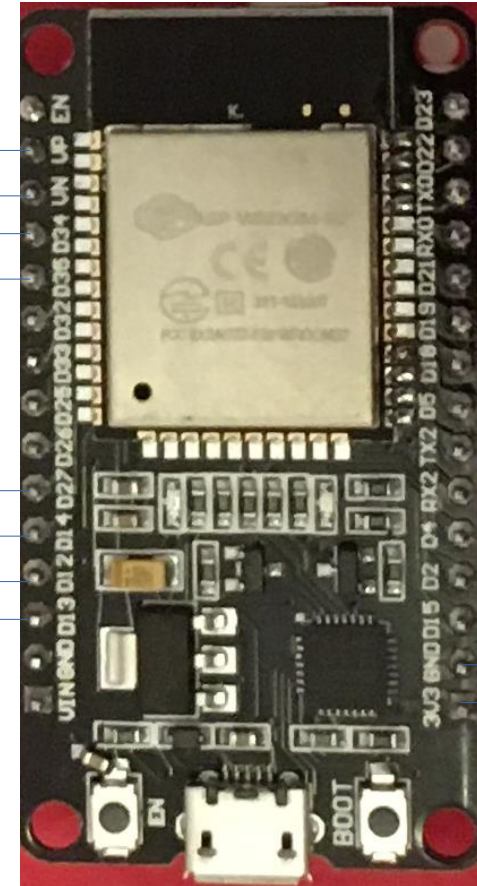
ESP32 4-Channel Lifter Controller Wiring schedule

Pin	Alias	Type	Function
36	ADC1_0	Analog	Pitch 1 demand
39	ADC1_3	Analog	Pitch 2 demand
34	ADC1_6	Analog	Roll 1 demand
35	ADC1_7	Analog	Roll 2 demand
16	U2_TXD	TXD	TX
17	U2_RXD	RXD	RX
27	GPIO33	PWM	Roll 2 PWM Out
14	GPIO25	PWM	Roll 1 PWM Out
12	GPIO26	PWM	Pitch 2 PWM Out
13	GPIO27	PWM	Pitch 1 PWM Out

Roll 2 Demand 50KPot Wiper
 Roll 1 Demand 50KPot Wiper
 Pitch 2 Demand 50KPot Wiper
 Pitch 1 Demand 50KPot Wiper

Roll 2 PWM Output
 Roll 1 PWM Output
 Pitch 2 PWM Output
 Pitch 1 PWM Output

0V
 +3v3



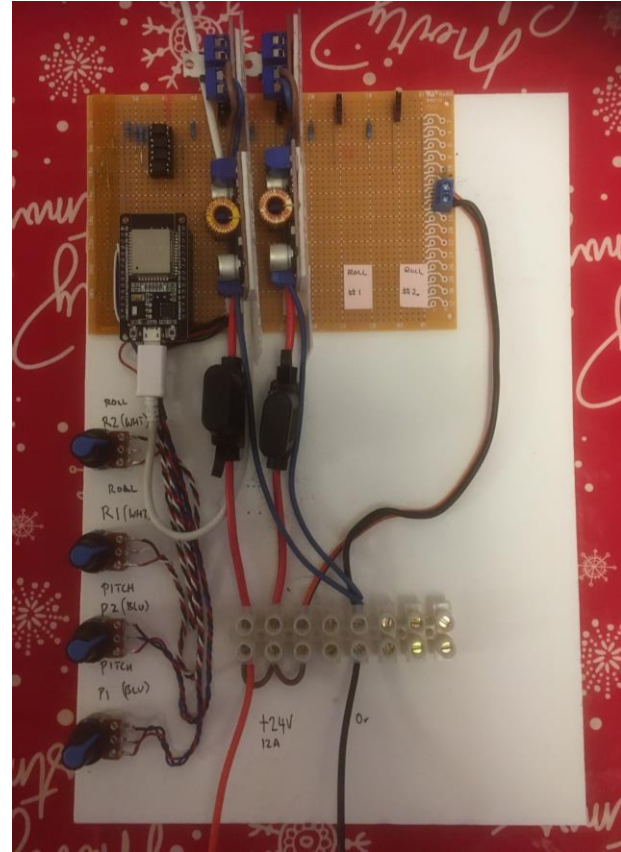
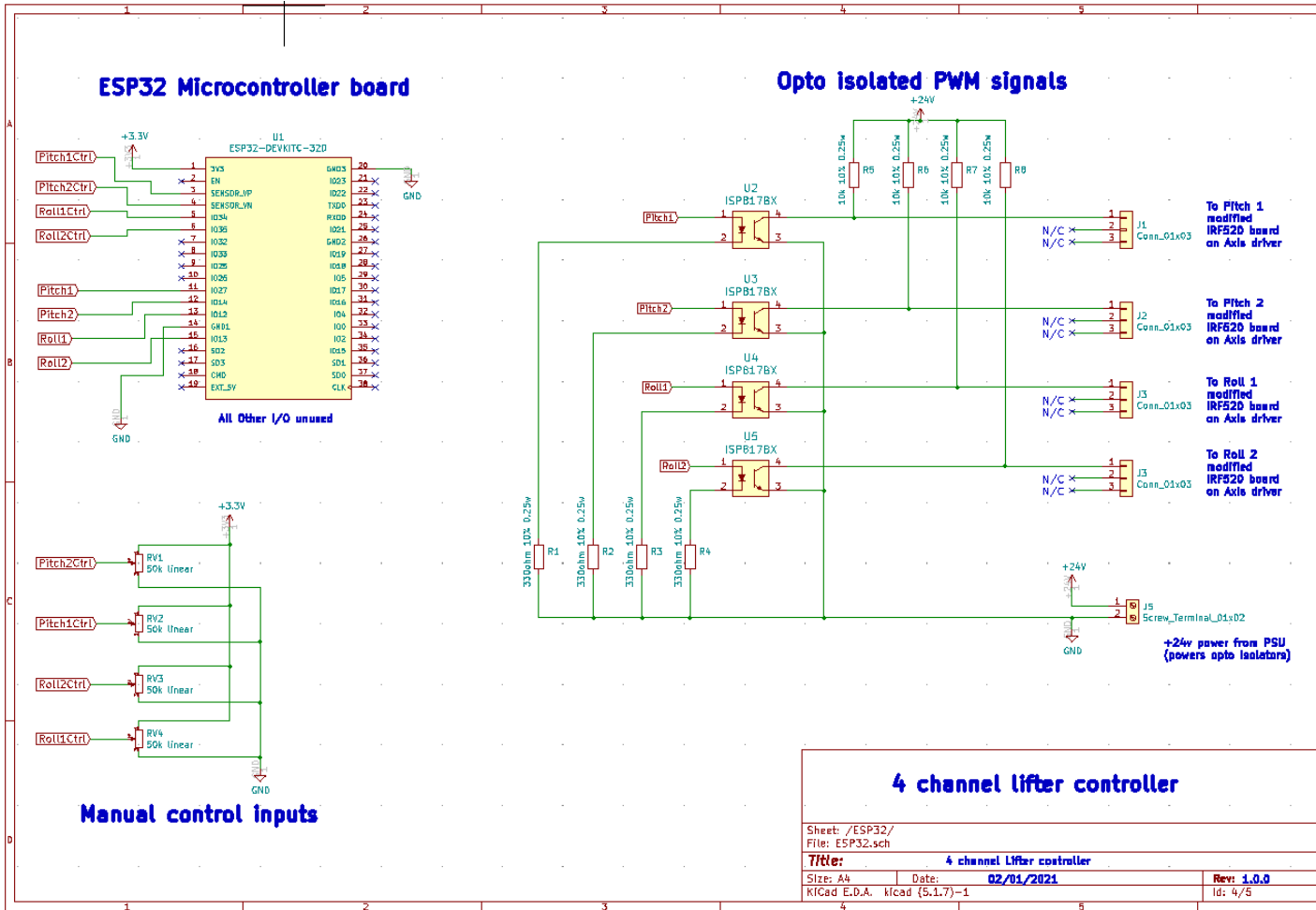
All other I/O unused

The firmware for this is software block: 5.low-effort-inverter

Code available here:

<https://github.com/cybernetic-research/ion-lifter-esp32-command-control-network>

ESP32 4-Channel Lifter Controller Schematic



ESP32 4-Channel Lifter Controller Programming

The code that runs on the 4 – channel lifter controller is available on github here:

https://github.com/cybernetic-research/ion-lifter-esp32-command-control-network/tree/main/ion_lifter_network/5.low-effort-inverter/low-effort-inverter-100hz-pwm

The code for the 4-channel lifter controller is called `low-effort-inverter-100hz-pwm.ino`. It was written using Arduino ide version 1.8.9 and has dependencies on libraries

- WebsocketsClient_Generic
- ArduinoJson

[illegible]

ESP32 4-Channel Lifter Controller Web socket API

When the 4-channel controller is connected to the access point, it will regularly transmit a JSON message to the laptop in the format shown.

In this case the Pitch axes are reporting the ADC values, for pitch 1, and pitch 2, the last control values sent to the controller for the pitch axis by the laptop, and whether remote control is currently being used. The controller will alternate between sending messages from the INVERTER_PITCH or INVERTER_ROLL axis. It is possible to record potentiometer values at the laptop and play them back later as a motion profile for the octagon lifter.

```
{"sender": "INVERTER_PITCH", "destination": "CONTROLLER", "msg": "STATUS", "payload": {"ADC0": 0, "ADC1": 0, "CTRL0": 0, "CTRL1": 0, "REMOTE_CTRL": true, "}}
```

↑
Potentiometer value (0-255), for control of Pitch1, or Pitch2

↑
Lifter thrust for this axis (0-255), in this case Pitch1, and Pitch2

The 4-channel controller may be addressed as "INVERTER_PITCH", or "INVERTER_ROLL" It can be controlled by sending it JSON messages in the format below.

Below are commands to enable remote control and set the thrust to both pitch axes to 0.

Laptop ID (don't change)

```
{"sender": "CONTROLLER", "destination": "INVERTER_PITCH", "msg": "UPDATE", "payload": {"REMOTE_CONTROL_ENABLE": true, "}}
```

↑
Set to INVERTER_PITCH or INVERTER_ROLL

↑
Set to true or false to enable/disable remote control

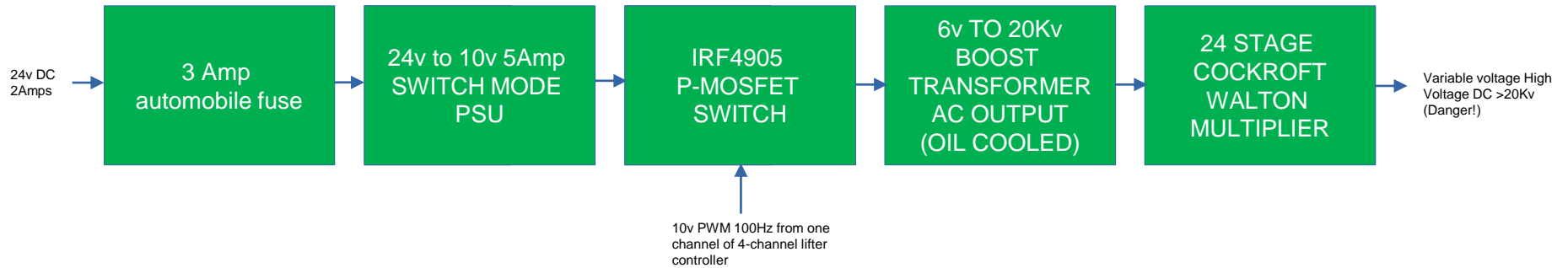
```
{"sender": "CONTROLLER", "destination": "INVERTER_PITCH", "msg": "UPDATE", "payload": {"AXIS0": 0, "AXIS1": 0, "}}
```

↑
PWM demand value – a number between 0 and 255 – sets lifter thrust for that axis, in this case Pitch1, and Pitch2

ESP32 4-Channel Lifter Controller BOM

Item	Part	Quantity	Supplier	Notes
1	Stripboard	1	any	10 x 15cm or larger
2	ESP32 wroom32 arduino devkit	1	Amazon / any	-
3	Sil header 3 ways	4	Amazon / any	Aussel 20 Pieces 2.54mm Breakaway PCB Board 40Pin Male and Female Header Connector for Arduino Shield (PIN-2 20PCS) Cut to length as needed (3 pins for axis cards) (15 pins for ESP32 module)
4	Resistor 330 Ohm (through hole 0.25watt 10%)	4	any	-
5	Resistor 10KOhm (through hole 0.25watt 10%)	4	any	-
6	Screw terminal header 2pin	1	any	-
7	Potentiometer 50K linear + knobs etc	4	any	Power from 3v3
8	Wire 7/02 black, white, red, blue	4 each	any	Cut to 25cm
9	Wire 17Amp red	1	any	Cut to 25cm
10	Wire 17Amp Black	1	any	Cut to 25cm
11	13 Amp terminal block 8-way	1	any	
12	Polystyrene plastic panel	1	any	A4 Paper size (21 x 30cm)
13	Double sided adhesive tape	As required	any	-
14	USB Micro cable (power ESP32)	1	any	-
15	+5v USB PSU (power ESP32)	1	any	-
16	Opto-isolator chip	4	RS	RS 161-1063 (ISP817BX)
17	+24v 12Amp PSU	1	any	Need at least 2 amps per powered axis at 24v

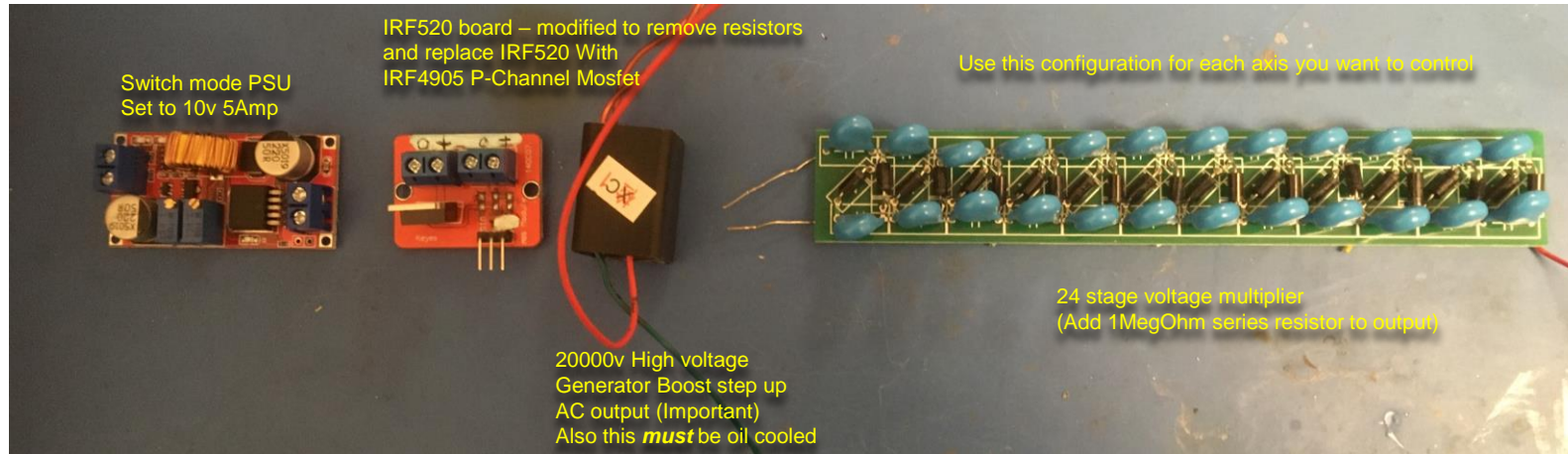
Lifter Axis Driver Block Diagram (One for each axis)



- In the octagon lifter there are 4 axis drivers, 2 for pitch, and 2 for roll.
- The axis drivers are the amplifier circuits that convert the PWM from the 4-channel controller to the high DC voltages necessary to produce ion thrust.
- Like the 4-channel controller, the design is minimal and requires low assembly effort.
- The 24volt incoming supply is regulated to no more than 10volts by a low cost switch mode PSU readily available from eBay or amazon
- The 10volt DC output of the switch mode PSU is switched by a P-Channel mosfet using one of the PWM outputs of the 4-channel controller. Switching is done using a modified IRF520 mosfet switch PCB. Again this element is low cost and available from Amazon or eBay. Note, it must be modified to use a P-Channel mosfet
- The PWM signal switched by the mosfet is used to power a low cost 6v to 20Kv AC Boost transformer available from Amazon or eBay. ***Note as shown here, the component is being run outside of manufacturers specification, and must be oil cooled, and only run for periods of time less than 3 minutes.***
- The output from the boost transformer is fed into a 24-stage Cockcroft Walton multiplier with a 1Megohm series resistor attached to it.
- The high voltage output may be adjusted using the PWM from the 4-channel controller, this provides proportional control over a lifter axis.

Warning: Lifters make use of high voltages and are a potential electric shock hazard. Please be careful when working with high voltages. EMC may also be a concern. Replication of this project is done entirely at your own risk.

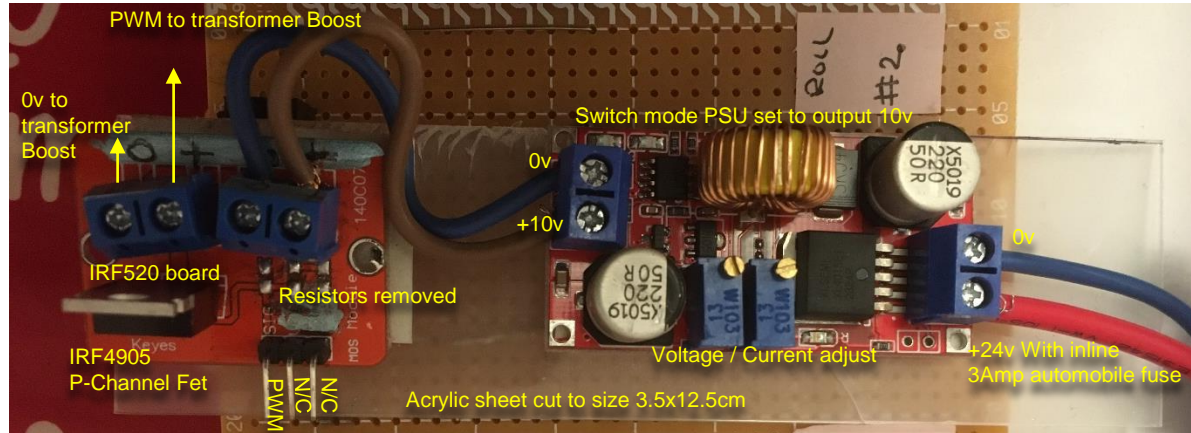
Lifter Axis driver (Pitch, Roll, Lateral Thrust etc)



- All Axis drivers use the circuit modules shown above. These parts are relatively low cost and readily available from suppliers such as Amazon, or ebay
- There are 4 axis drivers, 2 for pitch and 2 for roll. The lifter would be capable of forward motion and turning, if more axes were added
- Wiring between modules is deliberately kept simple
- Axis drivers make use of a switch mode power supply converting 24v DC to 10v DC at up to 5 Amps per channel
- An IRF3205 mosfet PCB is used for each channel to switch a DC to AC high voltage transformer module using the PWM from the 4-channel controller
- **Important – the IRF520 PCB Absolutely Must be modified to use a P-Channel Mosfet type IRF4905, this also involves removing all LEDs resistors on that PCB too. A different polarity needs to be used on the signals at the screw connectors.** This allows the high voltage stages to use the same ground and allow multiple high voltage stages to be energised at the same time.
- **The DC to AC transformer must be oil cooled to allow it to run at higher voltage that 7.8v.**
- High voltage transformer must not be run higher than 10v, and should only be run for 2 to 3 minutes at a time, before letting it cool for a while.
- The DC to AC high voltage transformer module drives a 24 stage Cockcroft Walton voltage multiplier with series 1MOhm limiting resistor. This drives the corona wire for a given axis.
- The PWM input signal applied to a lifter axis driver allows for a theoretical voltage output in excess of 30Kv, and can be controlled proportionally

Warning: Lifters make use of high voltages and are a potential electric shock hazard. Please be careful when working with high voltages. EMC may also be a concern. Replication of this project is done entirely at your own risk.

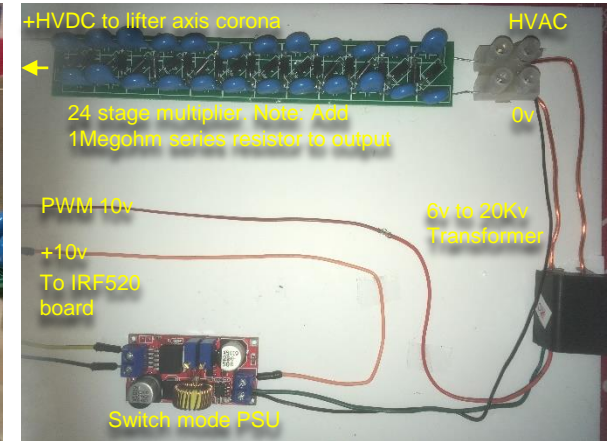
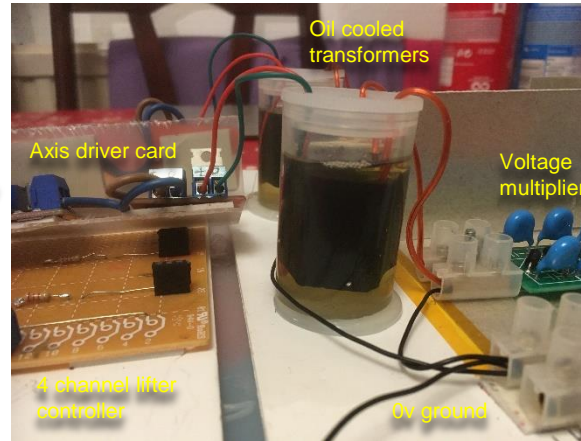
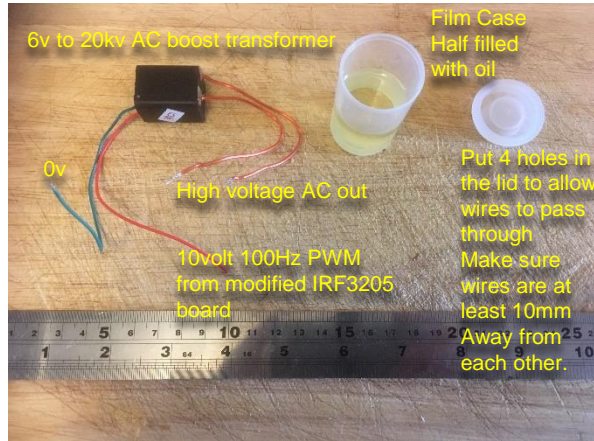
Lifter Axis driver – Modification of IRF3205 board and wiring



- The IRF3205 board is low cost and readily available but cannot be used in its basic form this system.
- The transformer **must** be referenced to the same ground used by everything else, and this is not possible with a N-channel Fet switching a high side load.
- The IRF3205 mosfet must be replaced by a P-channel part such as IRF4905, and also the resistors and LED on the board also need to be removed, because they will upset the gate biasing
- After the modification the transformer can be driven by PWM. The PWM duty cycle will vary the output voltage of the transformer, allowing variable thrust to any given lifter axis.

Warning: Lifters make use of high voltages and are a potential electric shock hazard. Please be careful when working with high voltages. EMC may also be a concern. Replication of this project is done entirely at your own risk.

Lifter Axis driver – Oil cooling the transformers and output wiring notes

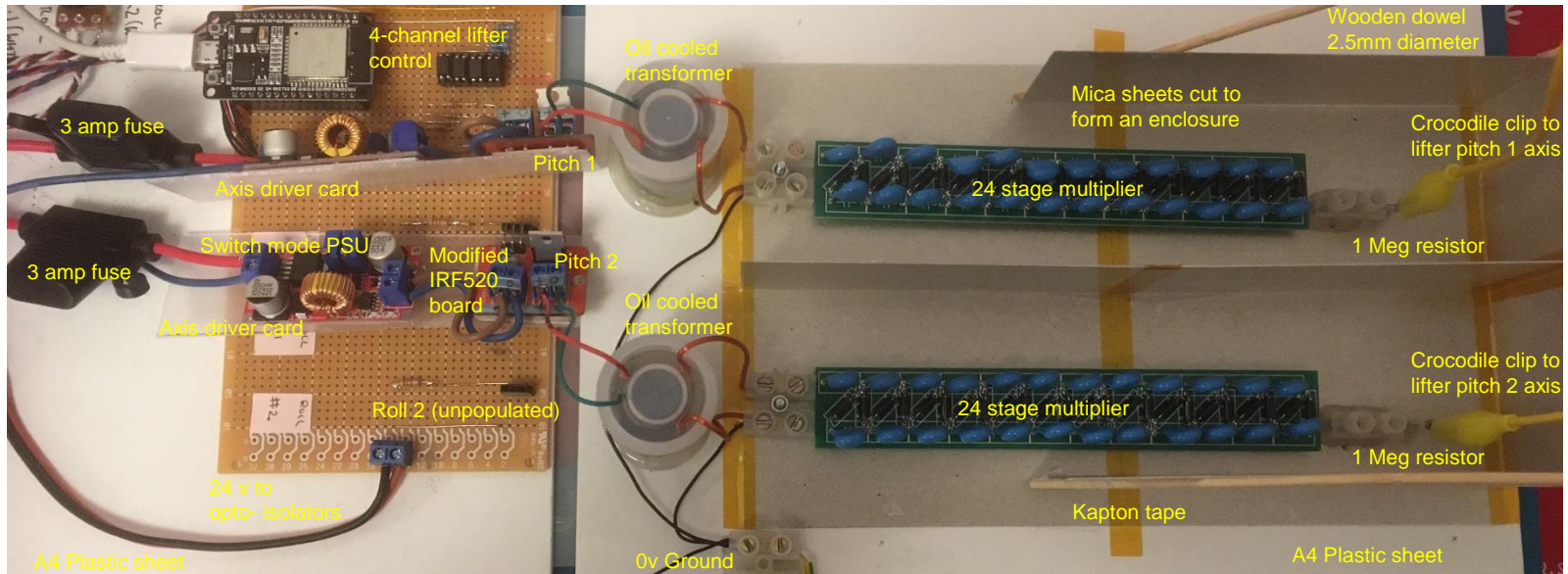


- The DC to AC transformer is intended for use at 6v DC.
- In this application it is run with 100Hz PWM at 10v, which is beyond the manufacturers rating. It is known that the device cannot be used above 11volts.
- The transformer can be run oil cooled, and in this application they were run with vegetable oil.
- The transformers were placed into film cases half filled with oil. Transformer wires
- Tests suggest that it can be run oil cooled for periods of two to three minutes at a time, but should then be given time to cool (lifter performance begins to fade off)

Note: it is entirely possible that better and more reliable results may be obtained with different transformers, but this is the only configuration that was tested.

Warning: Lifters make use of high voltages and are a potential electric shock hazard. Please be careful when working with high voltages. EMC may also be a concern. Replication of this project is done entirely at your own risk.

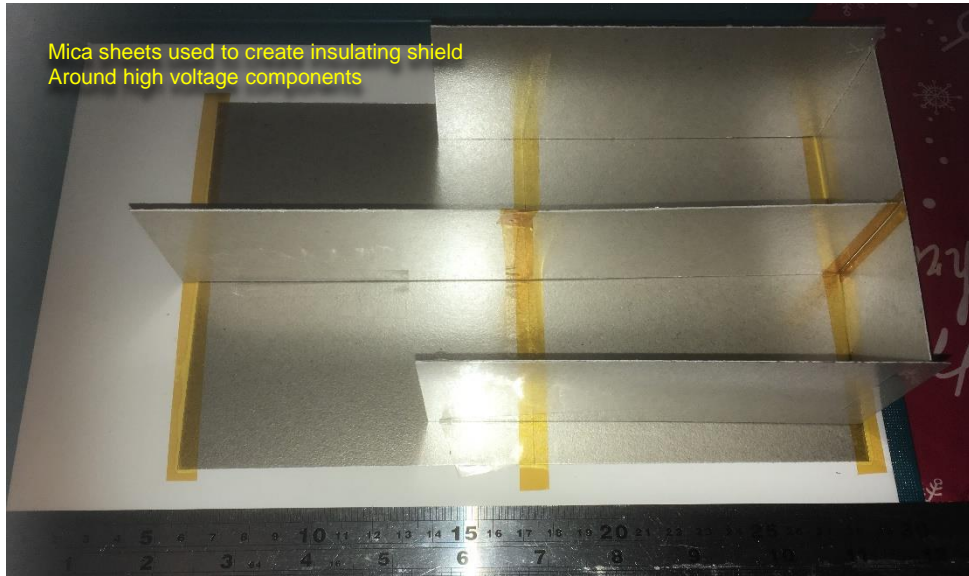
Lifter Axis driver – Enclosure



- The 24stage multipliers are mounted on top of mica sheets. This is done to prevent electrical arcing.
- The mica sheets should be mounted on top of a plastic insulator such as ABS to insulate anything underneath from the high voltage
- Wooden dowels are used to create a frame outside of the enclosure.
- Crocodile clips can be attached to the dowels, and are used to connect the voltage multipliers to each lifter axis via a 1megohm resistor

Warning: Lifters make use of high voltages and are a potential electric shock hazard. Please be careful when working with high voltages. EMC may also be a concern. Replication of this project is done entirely at your own risk.

Lifter Axis driver – Enclosure



- The 24stage multipliers are mounted on top of mica sheets. This is done to prevent electrical arcing.
- The mica sheets should be mounted on top of a plastic insulator such as ABS to insulate anything underneath from the high voltage
- Wooden dowels are used to create a frame outside of the enclosure.
- Crocodile clips can be attached to the dowels, and are used to connect the voltage multipliers to each lifter axis via a 1megohm resistor

Warning: Lifters make use of high voltages and are a potential electric shock hazard. Please be careful when working with high voltages. EMC may also be a concern. Replication of this project is done entirely at your own risk.

Lifter Axis Driver BOM (one per axis)

Item	Part	Quantity	Supplier	Notes
1	Inline automotive fuse holder + fuse	1 (per axis)	any	At 24v each axis is expected to draw around 2Amps. 3Amp fuses have been chosen
2	3 Amp (or higher) wire Black, Red	As needed per axis	any	-
3	Switch mode PSU 5Amp	1 (per axis)	Amazon	Dealikee 4 Pack 5A DC-DC Adjustable Buck Converter, XL401. Set to 10volt output
4	Film Canisters	1 (per transformer)	Amazon	Firgus Film Canisters, clear 3cm diameter
5	Oil	Half fill canister	any	A dielectric oil to cool transformer. Tests were run with vegetable oil.
6	IRF520 board	1 (per axis)	Amazon	WMYCONGCONG 10 Pcs IRF520 MOSFET Driver Module for Arduino Raspberry Pi (Need to remove mosfet and resistors on the board)
7	IRF4905 P-channel mosfet	1 (per axis)	any	Must replace IRF3250 mosfet with this
8	DC to AC high voltage transformer	1 (per axis)	ebay	DC 3.6V-6V High Voltage Generator AC20KV Boost Electric Module Replacement UK
9	1Megohm Resistors (through hole 0.25watt 10%)	1 (per axis)	any	Use in series with 24 stage multiplier output, reduces current of arc discharge, does not degrade lift performance.
10	Double sided adhesive	As needed	any	-
11	24 stage voltage multiplier	1 (per axis)	ebay	Voltage Doubler Rectifying 24 Times Rectifier 60000V High Voltage Multiplier PSU
12	3 Amp (or higher) terminal strip	Cut as needed	any	Simplifies electrical connection
13	Mica sheet	Cut as needed	Amazon	Yan Ban Universal Mica Tablets
14	Crocodile clips	1 (per axis)	any	Used to connect axis driver to ion thruster corona wire

ESP32 Access Point Block Diagram

Wifi

Network: LIFTER

Password: ionocraft

Uses websockets

Can have up to 5 client nodes

Software block: 1.server

(Optional)
ESP32 WIFI
ACCESS POINT

The ESP32 access point is based around the Arduino library WebSockets_Generic, and relatively high speed control is possible using websockets.

It creates an wifi network called LIFTER, password is ionocraft. All other ESP32 devices in the system will automatically connect to this server.

The access point also performs a basic routing function to ensure that signals from sensors are routed to a laptop, and signals from the laptop are routed to the 4-channel lifter controller or lifter inverter units.

The access point is required for remote control, and in order to control the lifter with a joypad, you must connect your computer to the wifi access point created by this module.

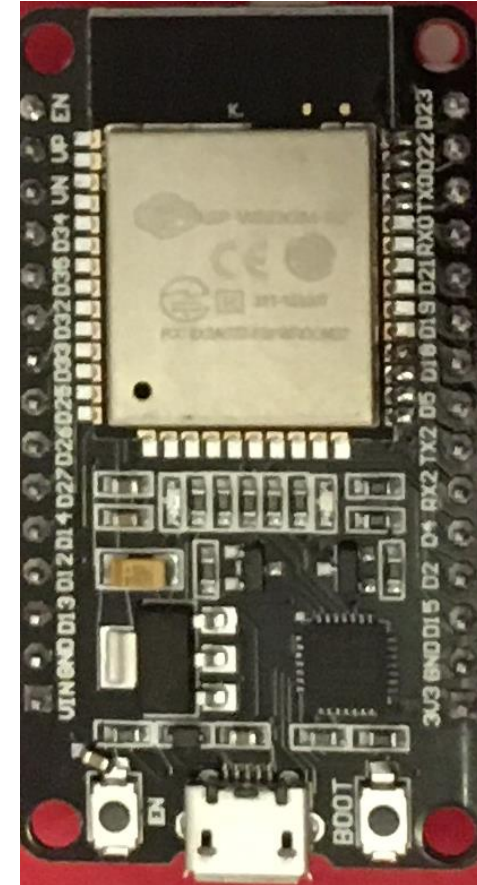
Code for this block is available here:

https://github.com/cybernetic-research/ion-lifter-esp32-command-control-network/tree/main/ion_lifter_network/1.server/ESP32_WebSocketServer-accesspoint

ESP32 Access Point Wiring schedule

Pin	Alias	Type	Function
-	-	-	-

There is no circuit diagram or I/O for an ESP32 board flash with this firmware, to use it you simply flash the firmware onto an ESP32 chip, and then plug it into a USB port to get power, then wait 5 seconds for the network to appear. That is it...



All other I/O unused

Code for this block is available here:

https://github.com/cybernetic-research/ion-lifter-esp32-command-control-network/tree/main/ion_lifter_network/1.server/ESP32_WebSocketServer-accesspoint

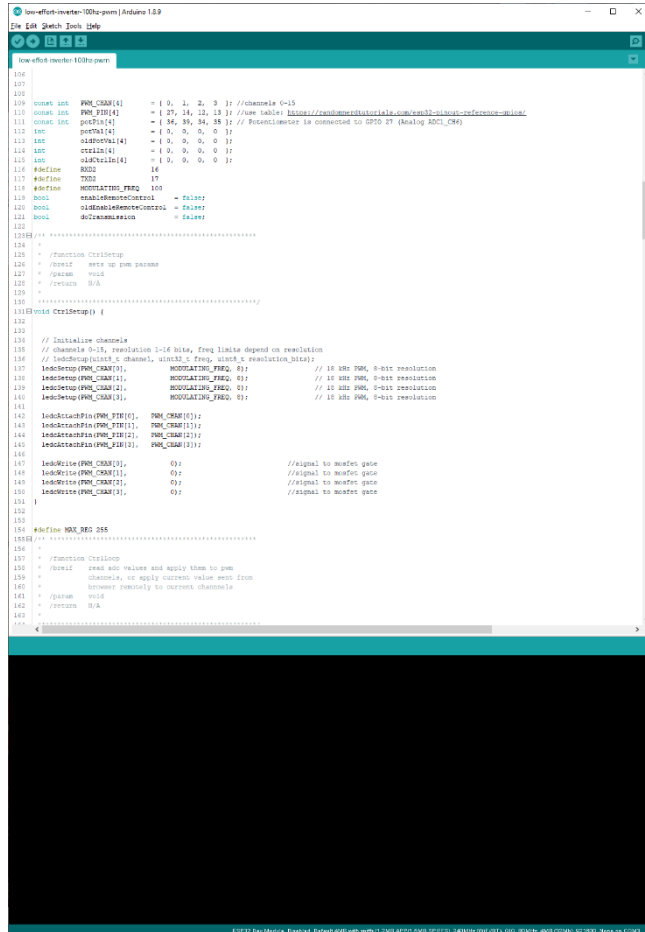
ESP32 Access Point Programming

The code that runs on the access point is available on github here:

https://github.com/cybernetic-research/ion-lifter-esp32-command-control-network/tree/main/ion_lifter_network/1.server/ESP32_WebSocketServer-accesspoint

The code for the access point is called ESP32_WebSocketServer-accesspoint.ino It was written using Arduino ide version 1.8.9 and has dependencies on libraries

- WebsocketsClient_Generic
- ArduinoJson



Access Point BOM

Item	Part	Quantity	Supplier	Notes
1	ESP32 wroom32 arduino devkit	1	Amazon / any	-
2	USB Micro cable (power ESP32)	1	any	-
3	+5v USB PSU (power ESP32)	1	any	-

ESP32 Position Sensor Module Block Diagram



(Optional)
ESP32 WIFI MPU6050
POSITION SENSOR

Software block: 3.sensor-pos

The ESP32 wifi MPU6050 module is intended to wirelessly stream positional data for pitch and roll to the laptop computer. An MPU6050 is interfaced to the ESP32 by means of I2C.

Experiments have been run that demonstrate that it is possible for an ion lifter to lift this particular sensor.

See:

https://youtu.be/C5L3u5OyNGA?list=PLiWsHfxy5jDVE85_Qa8NKqKy mudv9NQ1L

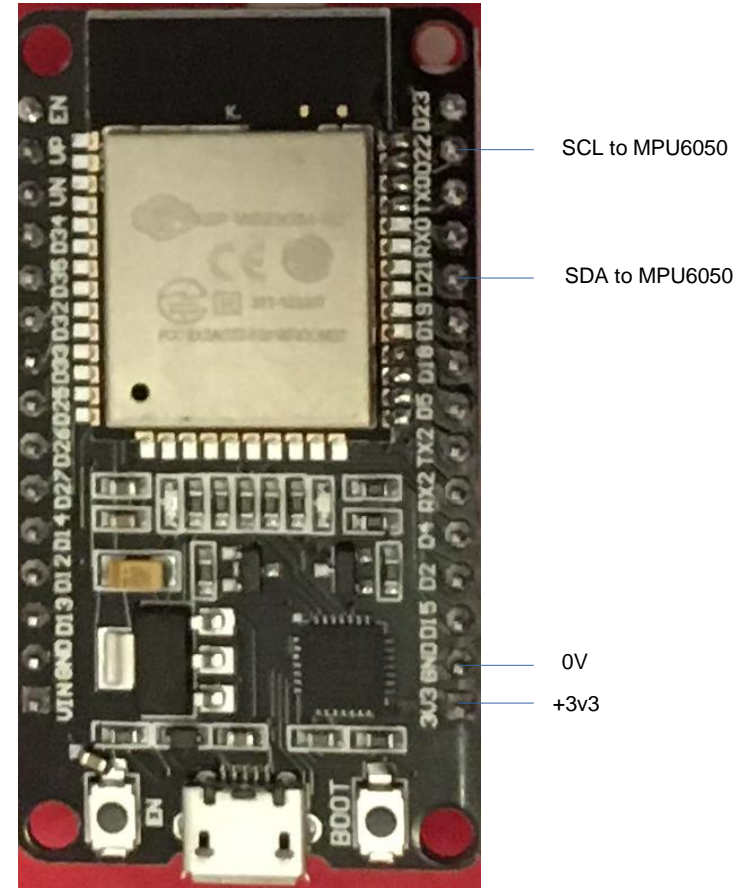
This module shall stream JSON encoded data to the laptop encoding the orientation of the sensor.

Code for this block is available here:

https://github.com/cybernetic-research/ion-lifter-esp32-command-control-network/tree/main/ion_lifter_network/3.sensor-pos/ESP32_WebSocketClient-mpu6050

ESP32 Position Sensor Module Wiring schedule

Pin	Alias	Type	Function
22	SCL	I2C	SCL to MPU3050
21	SDA	I2C	SDA to MPU6050
3v3	-	Power	Power to MPU3050
GND	-	Power	0v to MPU3050

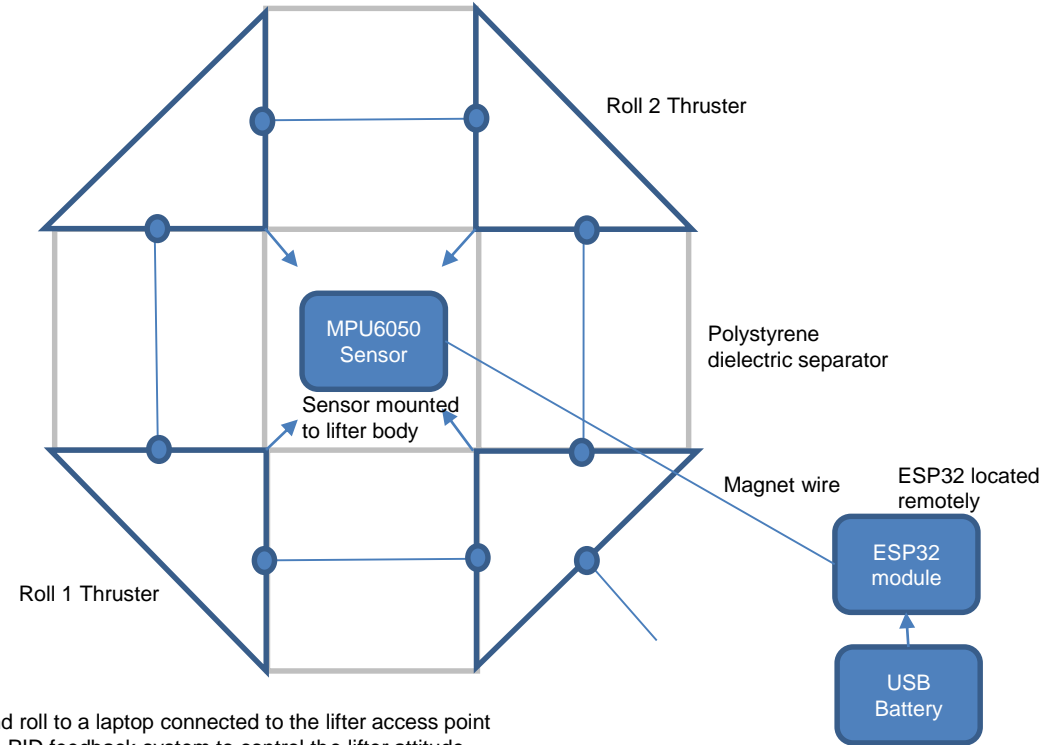
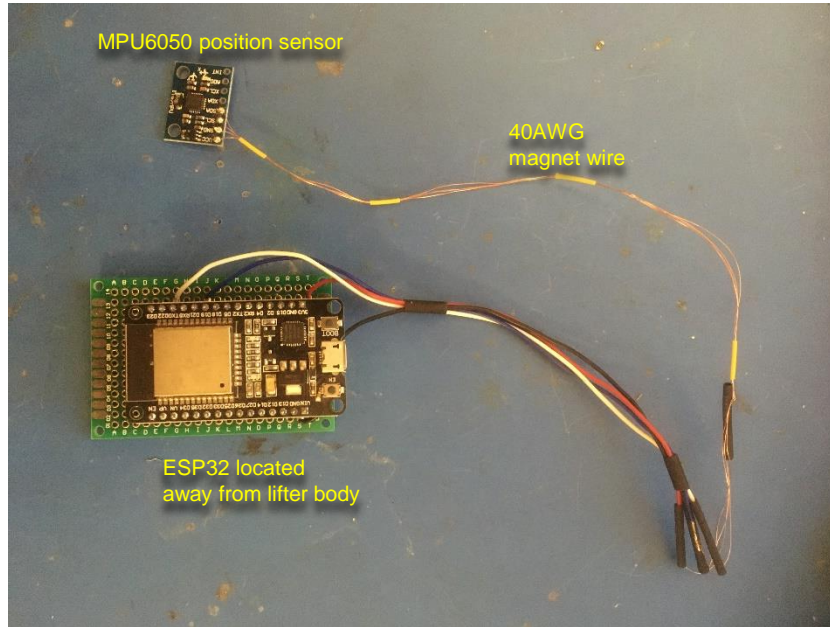


All other I/O unused

Code for this block is available here:

https://github.com/cybernetic-research/ion-lifter-esp32-command-control-network/tree/main/ion_lifter_network/3.sensor-pos/ESP32_WebSocketClient-mpu6050

ESP32 Position Sensor Module Wiring



- The position sensor module wirelessly streams MPU6050 Euler angles for heading, pitch and roll to a laptop connected to the lifter access point
- The purpose of the sensor is to be used as the positional feedback element of a closed loop PID feedback system to control the lifter attitude.
- Experiments have been performed with an MPU6050 sensor and an octagonal ion lifter, and it is thought to be feasible that the lifter can carry such a sensor.
- Only the barest minimum interface exists between the ESP32 and MPU6050, the interface is I2C on two wires and power supplied by the ESP32 to the MPU6050 (3v3 and ground)
- It is important to mention that the wiring between the lifter and sensor should be as lightweight and flexible as possible, to this end only 4 strands of 40AWG magnet wire are used. Thicker wire will likely prove too heavy for the lifter to carry.
- The sensor itself can likely be mounted in the centre of the lifter body as shown in the diagram to the upper right.
- **An important point to mention is that the ESP32, should remain on the floor, and for safety reasons should have its own isolated power supply such as a mobile phone portable charger.**

ESP32 Position Sensor Module Programming

```
low effort inverter 100hz-pwm.ino
// Arduino 1.8.9

// low effort inverter 100hz-pwm
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

The code that runs on the ESP32 Positon sensor module is available on github here:

https://github.com/cybernetic-research/ion-lifter-esp32-command-control-network/tree/main/ion_lifter_network/3.sensor-pos/ESP32_WebSocketClient-mpu6050

The code for the 4-channel lifter controller is called low-effort-inverter-100hz-pwm.ino It was written using Arduino ide version 1.8.9 and has dependencies on libraries

- WebsocketsClient_Generic
- ArduinoJson
- MPU6050 Basic Example with IMU by Kris weiner

ESP32 Position Sensor Module Web socket API

When the ESP32 position sensor module is connected to the access point, it will regularly transmit a JSON message to the laptop in the format shown. The message encodes to the Euler angles of the sensor and can be used in the laptop as feedback for a PID control loop/

```
{"sender": "SENSOR_ORIENTATION ", "destination": "CONTROLLER ", "msg": "STATUS", "payload": {"HEADING": "0", "PITCH": "0", "ROLL": "0", "ERROR": "0"}}
```

Heading pitch roll to 1 decimal place in degrees.
Note in the software implementation heading is unstable and unreliable, but it is not an important part of lifter control

Sensor initialisation error

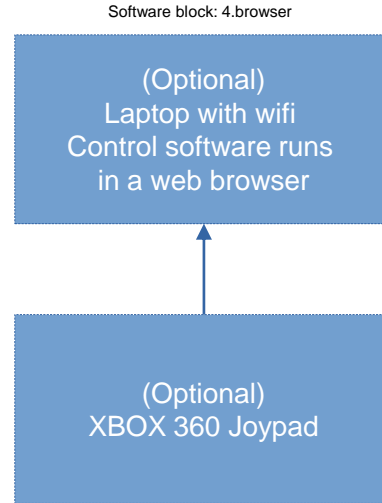
ESP32 Position Sensor Module BOM

Item	Part	Quantity	Supplier	Notes
1	ESP32 wroom32 arduino devkit	1	Amazon / any	-
2	USB Micro cable (power ESP32)	1	any	-
3	+5v USB PSU (power ESP32)	1	any	Preferably a battery / mobile phone charger battery
4	MPU6050 PCB	1	any	Only using i2c pins +3v3, 0v
5	40AWG Magnet wire	4 strands 20cm	any	-

Laptop Software Block Diagram

Wifi
Network: LIFTER
Password: ionocraft

**Connect your laptop
to this network to control
the lifter remotely**



The laptop is any modern wifi enabled laptop capable of running the chrome browser. It will work on linux windows or mac, and requires no installation.

Software to control the lifter is essentially a web application written in jQuery/html. Simply download the contents of the folder from github and open index.html with your chrome browser. (make sure the js folder is in the same directory as index.html)

The software will receive data streamed from the 4-channel lifter controller, and/or position sensor and display it on screen.

You may control the lifter using an xbox360 joypad if you like using this software.

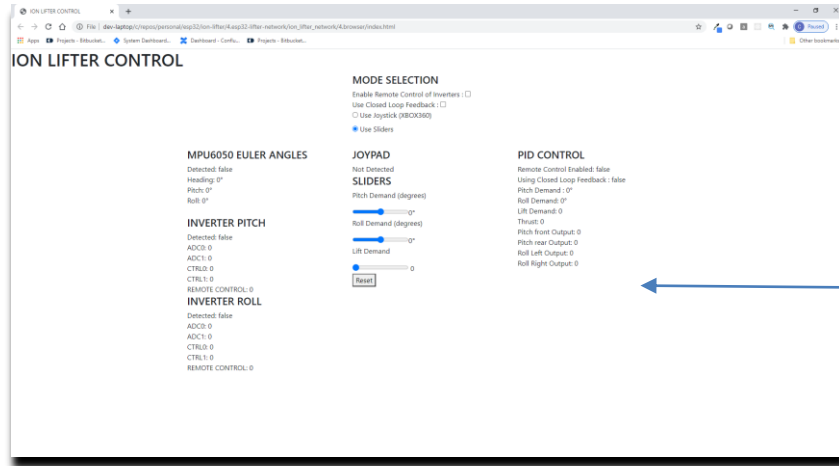
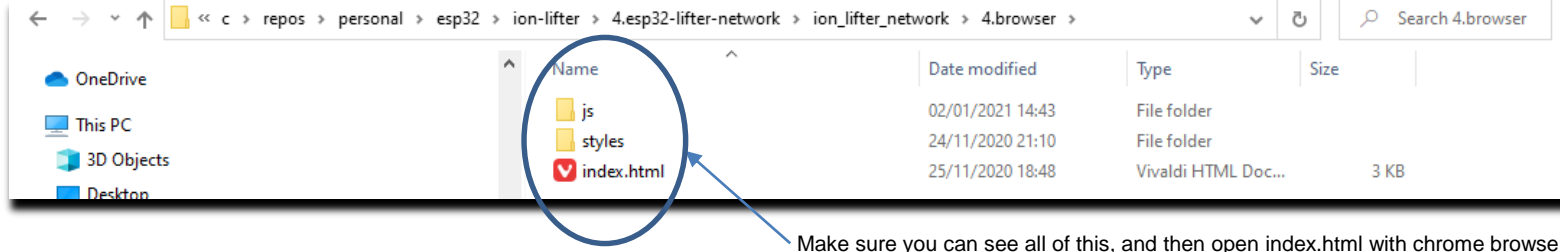
Code for this block is available here:

https://github.com/cybernetic-research/ion-lifter-esp32-command-control-network/tree/main/ion_lifter_network/4.browser

Laptop Software

Make sure the Access point is running and your laptop is connected to it.

Download the browser code from github here: https://github.com/cybernetic-research/ion-lifter-esp32-command-control-network/tree/main/ion_lifter_network/4.browser

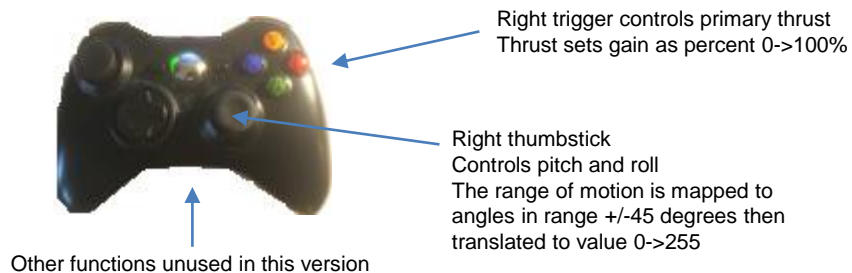


You will see this screen. If the sensors or 4-channel controller are connected you will see updates coming in for MPU6050 Euler Angles and or Inverter Pitch/Inverter Roll channels. You may control the thrusters using the sliders, but if you have an xbox 360 joypad, you may use that instead

Code for this block is available here:

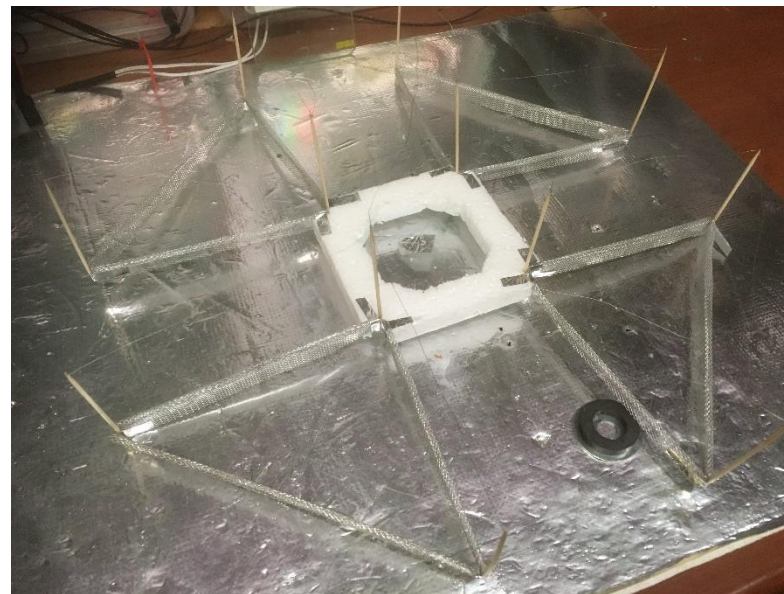
https://github.com/cybernetic-research/ion-lifter-esp32-command-control-network/tree/main/ion_lifter_network/4.browser

Laptop Software Joypad Control + Lifter Control Law



Control law:

```
Front = Pitch1 = thrust + sin((pitch_angle)*thrust)
Rear  = Pitch2 = thrust - sin((pitch_angle)*thrust)
Left  = Roll1  = thrust + sin((roll_angle)*thrust)
Right = Roll2  = thrust - sin((roll_angle)*thrust)
```



THIS HAS BEEN TESTED FOR ROLL AXIS

Code for this block is available here:

https://github.com/cybernetic-research/ion-lifter-esp32-command-control-network/tree/main/ion_lifter_network/4.browser

Laptop Software Joypad Control + Lifter PID Feedback



Right trigger controls primary thrust
Thrust sets gain as percent 0->100%

Right thumbstick
Controls pitch and roll
The range of motion is mapped to
angles in range +/-45 degrees then
translated to value 0->255

Other functions unused in this version

Control law:

Front = Pitch1 = thrust + sin((pitch_angle)*thrust)

Rear = Pitch2 = thrust - sin((pitch_angle)*thrust)

Left = Roll1 = thrust + sin((roll_angle)*thrust)

Right = Roll2 = thrust - sin((roll_angle)*thrust)

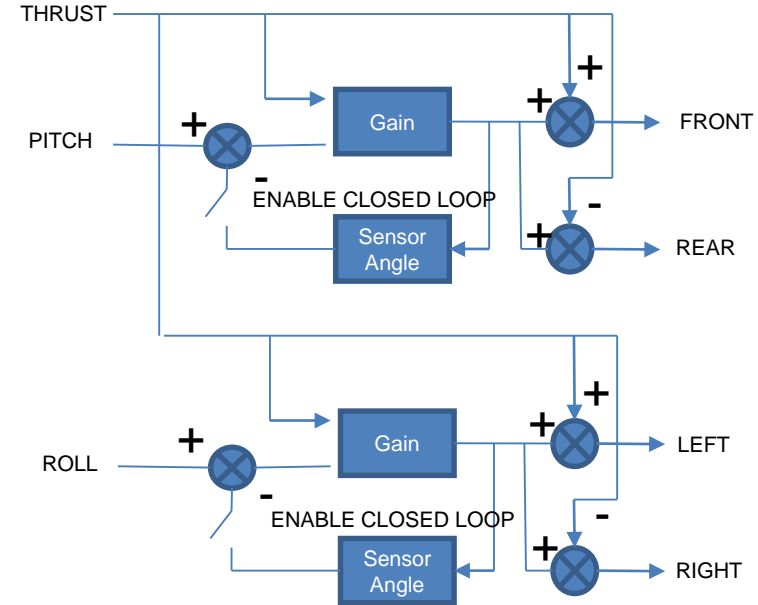


Image above shows a possible implementation of a PID closed loop feedback system
For the ionic lifter. At the time of release this implementation has not yet been tested.

Code for this block is available here:

https://github.com/cybernetic-research/ion-lifter-esp32-command-control-network/tree/main/ion_lifter_network/4.browser

Laptop Software Module BOM

Item	Part	Quantity	Supplier	Notes
1	Modern Laptop with USB and wifi capability	1	any	Software is web based, should work on any computer operating systems
2	XBOX360 joypad optional	1	any	-