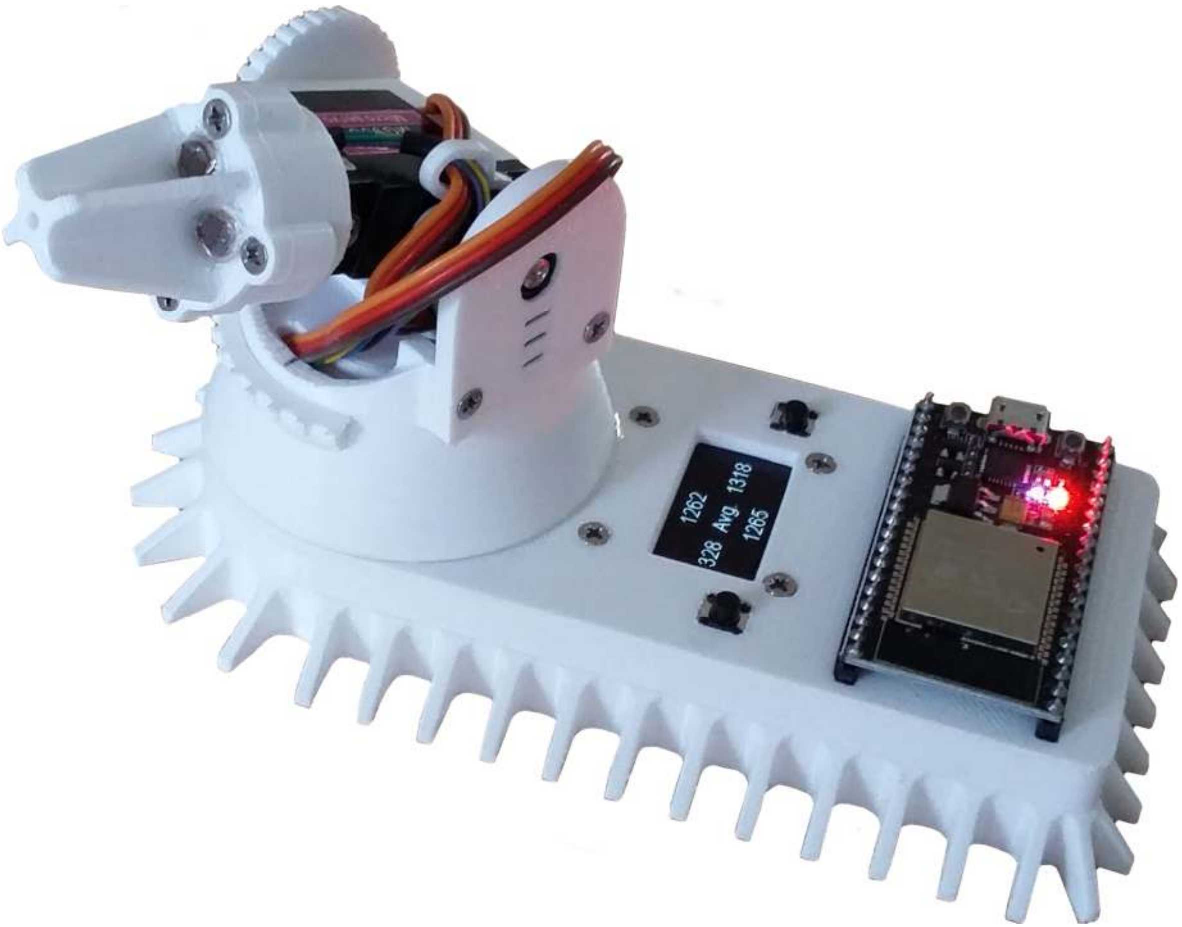
LightBot

Servo & Sensor Calibration



Issue: 1.1 Released: 25/10/2022 TechKnowTone

Why do we need to calibrate the servos?

IMPORTANT
IMPORTANT
IMPORTANT

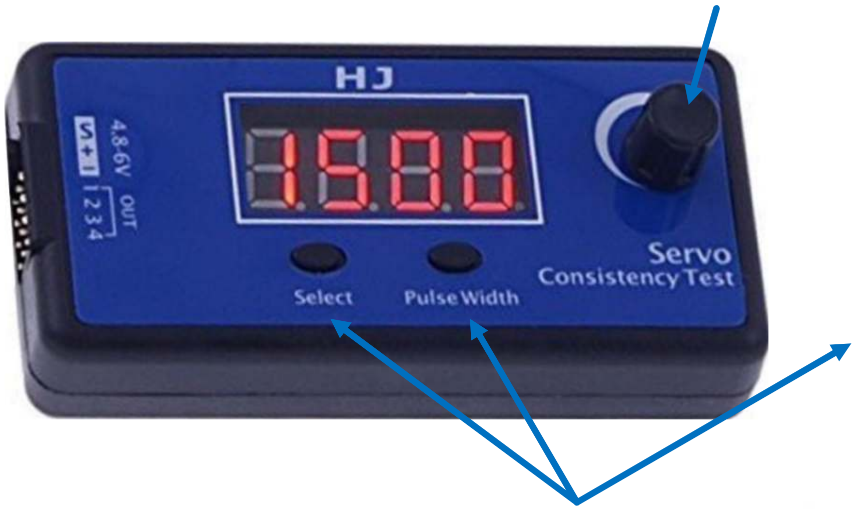
- No two servos are the same
- Servos can be damaged if not setup correctly
- Course calibration must be performed prior to the assembly process
- This sets approximate positions for the leaver arms
- Course calibration ensure servos are within mechanical limits
- Fine calibration determines min/max robot physical limits
- The ESP32 C++ code needs limit values in order to work accurately
- Hence, each robot has a unique set of calibrated values

Servo calibration is performed in two stages:

- Pre-set ensures mechanical parts are assembled correctly
- Fine calibration, performed during testing

- Repeat this process for a given servo if it is ever replaced.

HJ Servo Consistency Tester



Pot

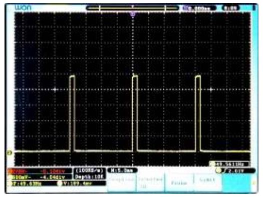
Select Button:

- Variable (Pot) pulse width 800 – 2200 µs (default mode)
- Fixed constant pulse width 1500 µs
- Sweep (Pot) pulse width from 800 -> 2200 -> 800 µs

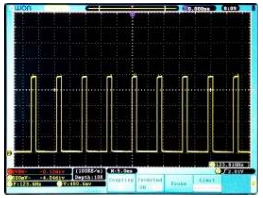
Pulse Width Button:

This is actually pulse frequency (Hz)

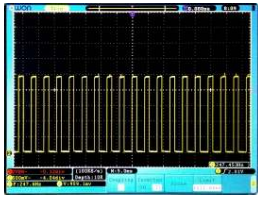
- 50H = 50 Hz - run MG90D at this frequency (default)
- 125H = 125 Hz
- 250H = 250 Hz



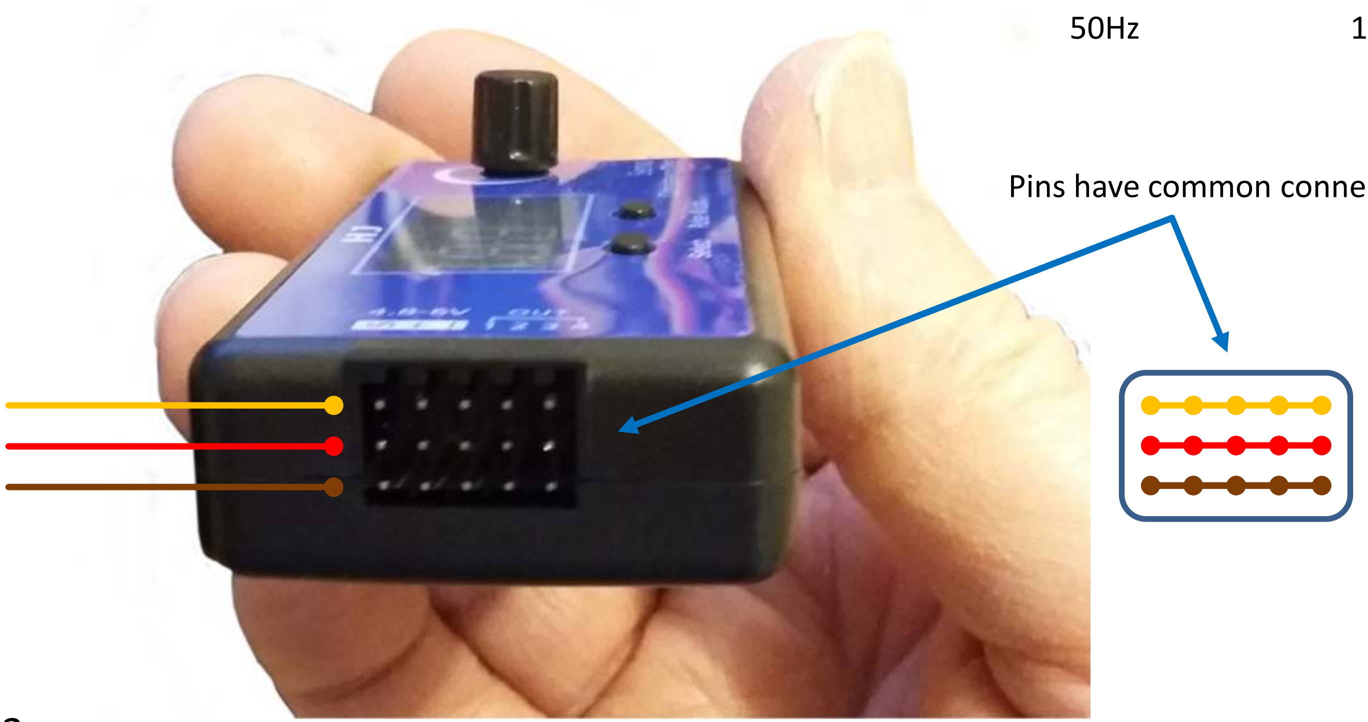
50Hz



125Hz



250Hz



Pins have common connections

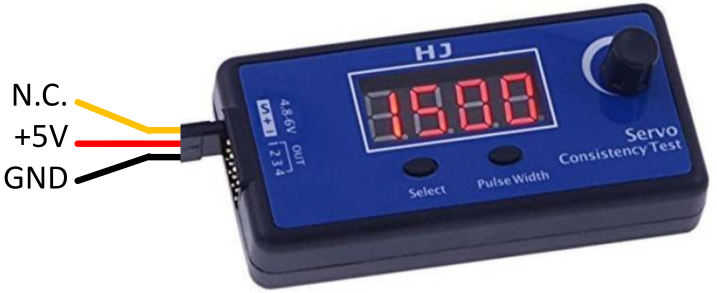
Signal
+4.8 - 6V
GND

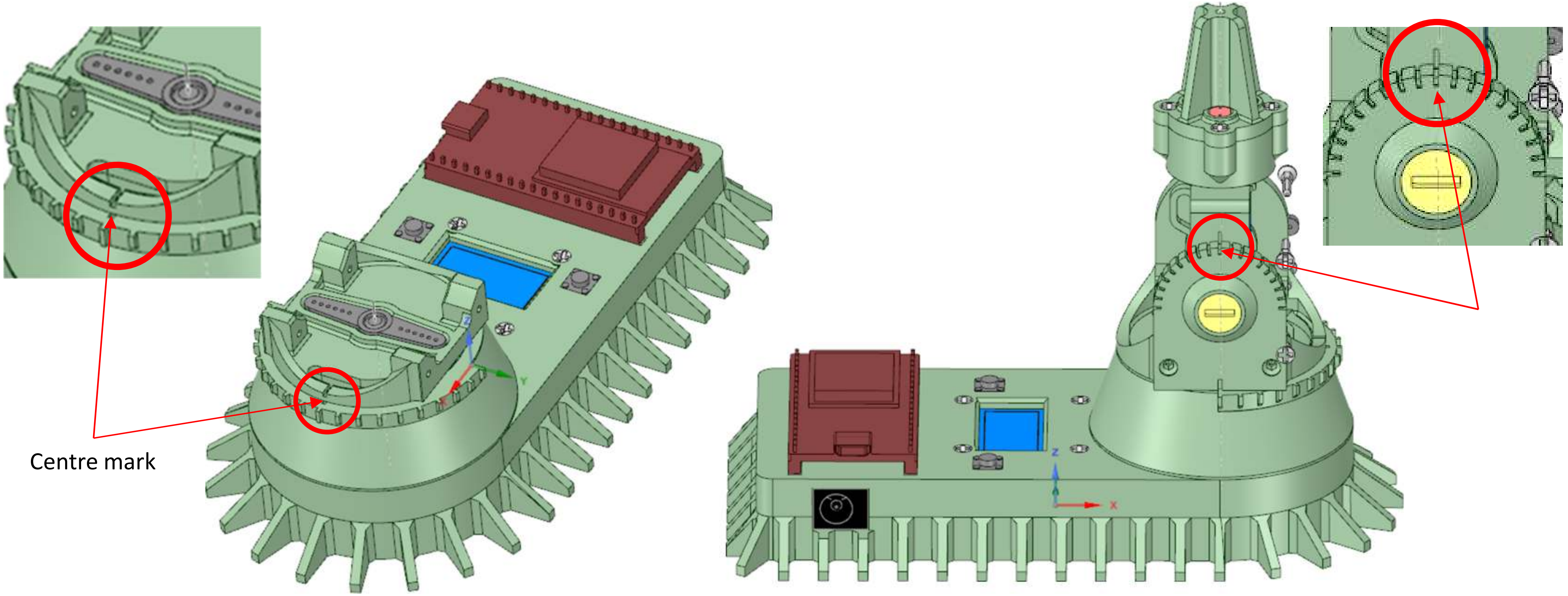
Signal
+4.8 - 6V
GND

Servo Pre-set For Assembly:

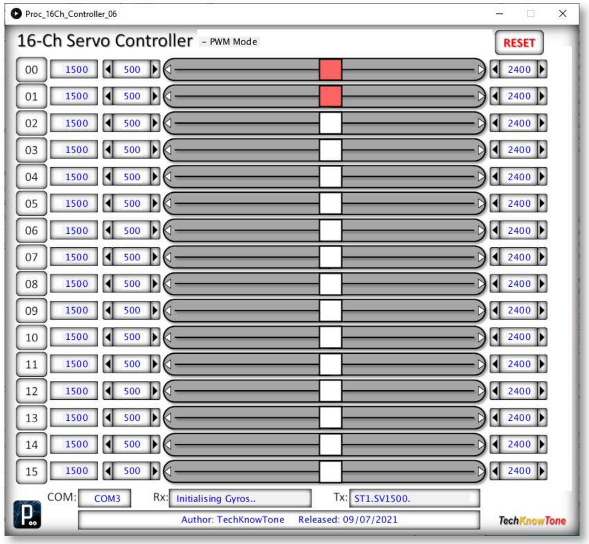
This ensures that attached mechanical part will have sufficient range.

- Select the correct servo plug. ie. Servo 0 for Pan turntable
- Connect servo to consistency tester and set centre value of 1500µs
- Attach mechanical part to the servo in the positions indicated.




Centre mark

Note: Servo splined shaft has 20 teeth. So the arm can only be attached in 18° intervals. The closest compromise position for 0° has to be found for the servo arm at 1500µs.



Proc_16Ch_Controller_06

16-Ch Servo Controller - PWM Mode

RESET

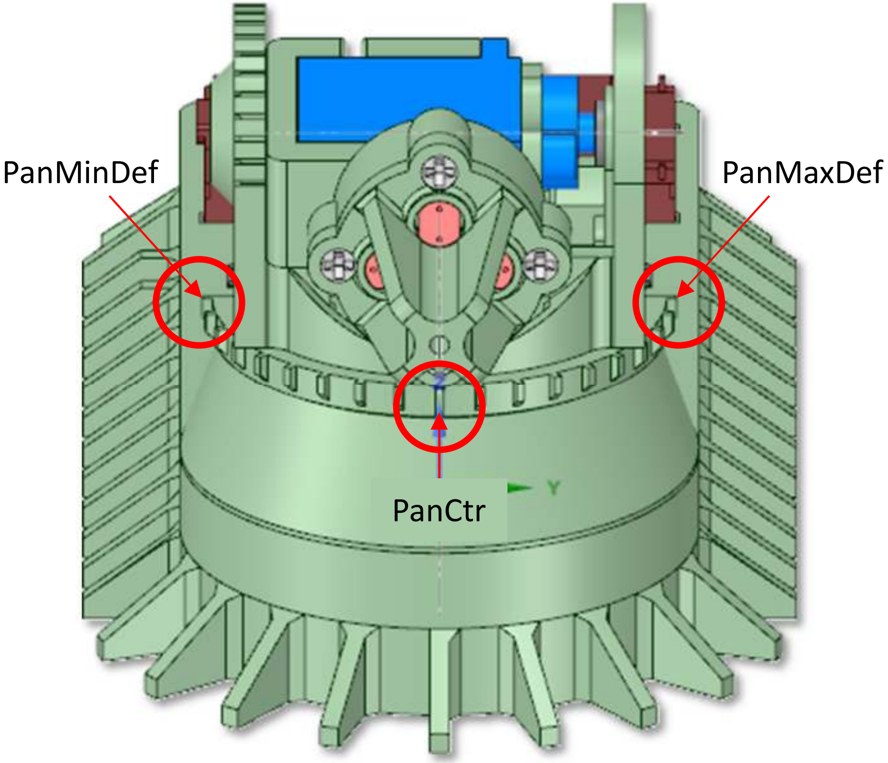
00	1500	500	2400
01	1500	500	2400
02	1500	500	2400
03	1500	500	2400
04	1500	500	2400
05	1500	500	2400
06	1500	500	2400
07	1500	500	2400
08	1500	500	2400
09	1500	500	2400
10	1500	500	2400
11	1500	500	2400
12	1500	500	2400
13	1500	500	2400
14	1500	500	2400
15	1500	500	2400

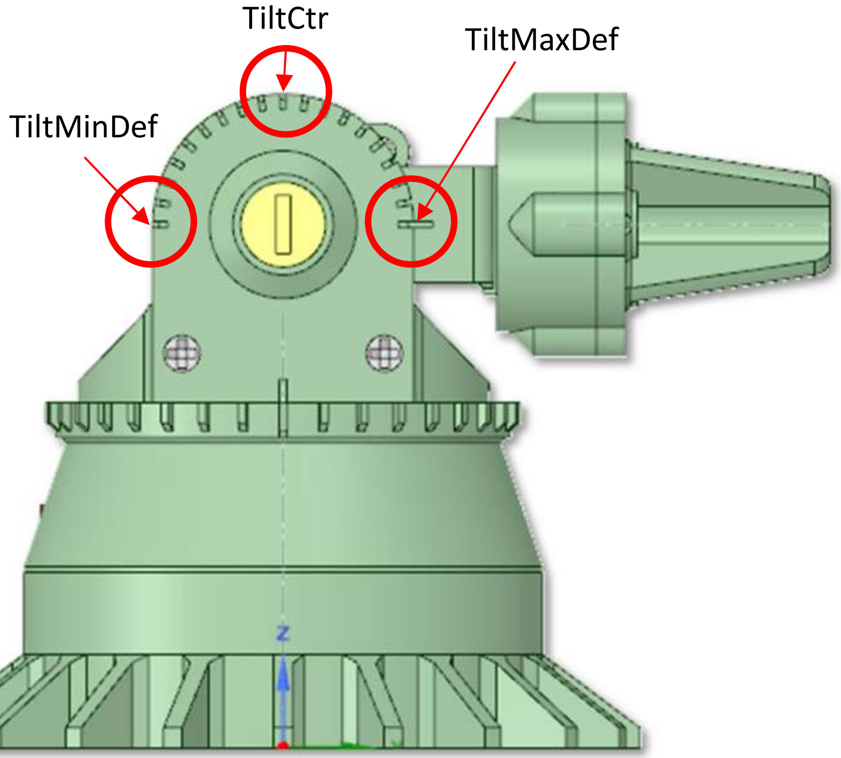
COM: COM3 Rx: Initialising Gyros.. Tx: ST1.SV1500.

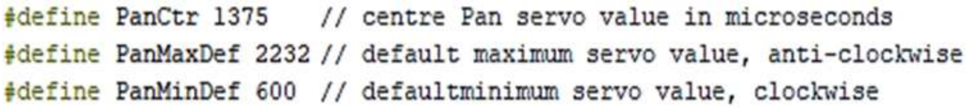
Author: TechKnowTone Released: 09/07/2021

Fine Calibration:

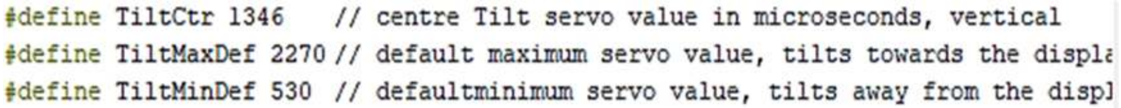
Use the provided 16-channel servo controller app to position the servos (Ch00 and Ch01) at the desired locations, and then record their values in the code definitions. The app connects to the ESP32 via the USB link, and lists the COM number when connected.







```
#define PanCtr 1375    // centre Pan servo value in microseconds
#define PanMaxDef 2232 // default maximum servo value, anti-clockwise
#define PanMinDef 600  // defaultminimum servo value, clockwise
```

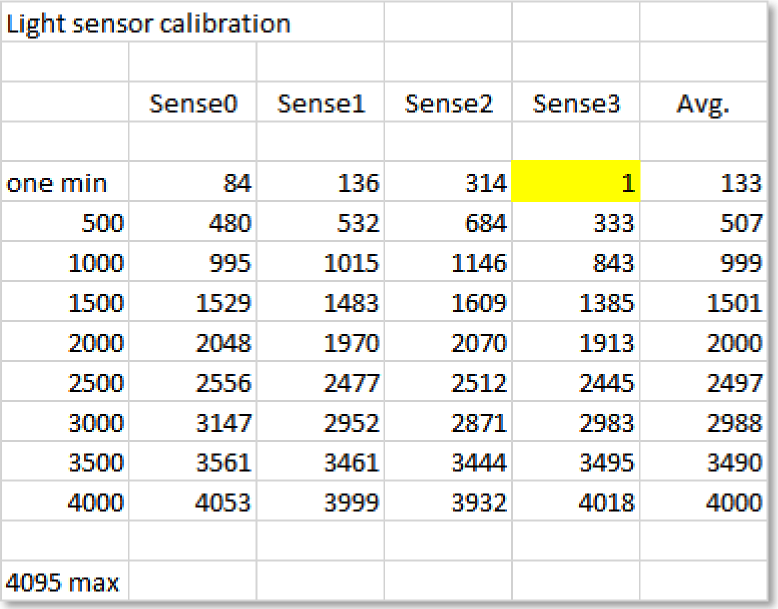


```
#define TiltCtr 1346    // centre Tilt servo value in microseconds, vertical
#define TiltMaxDef 2270 // default maximum servo value, tilts towards the displa
#define TiltMinDef 530  // defaultminimum servo value, tilts away from the displ
```

Sensor Calibration:

As the four GM5537 photo resistor sensors are discrete components, and not monolithic built on a chip as one sensor, there is a need to calibrate and match their sensitivity to different light levels in the code. Following this procedure will greatly improve the performance of your robot.

You need to print and use the calibration tube, which fits over the group of light sensors, and effectively shades them from all light, other than that entering the end of the tube. By placing varying thickness of paper over the end of the tube, we can then vary the intensity of light from a torch, and produce a table like the one below:

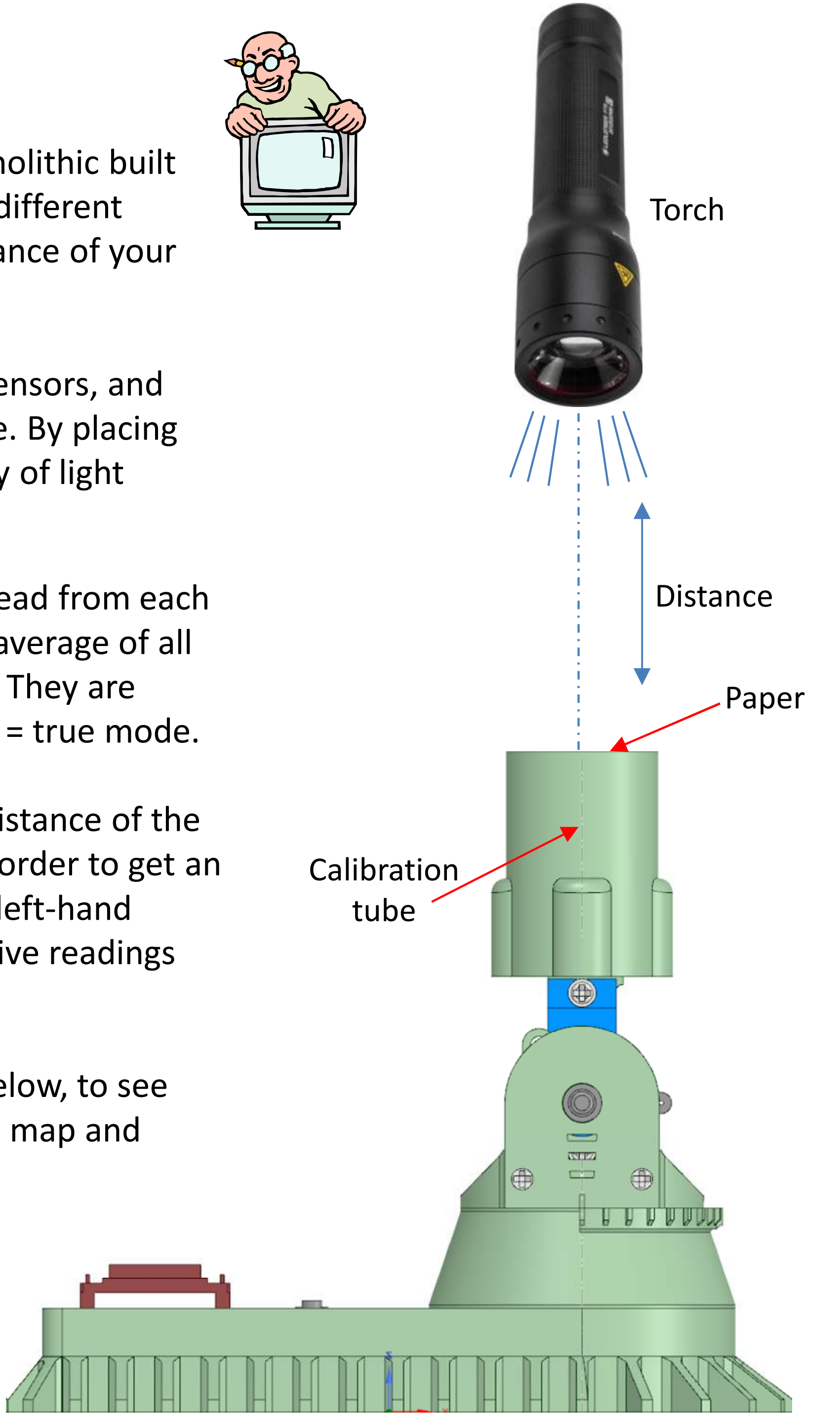


Light sensor calibration					
	Sense0	Sense1	Sense2	Sense3	Avg.
one min	84	136	314	1	133
500	480	532	684	333	507
1000	995	1015	1146	843	999
1500	1529	1483	1609	1385	1501
2000	2048	1970	2070	1913	2000
2500	2556	2477	2512	2445	2497
3000	3147	2952	2871	2983	2988
3500	3561	3461	3444	3495	3490
4000	4053	3999	3932	4018	4000
4095 max					

This table shows the values read from each of the four sensors, plus the average of all of them, by the ESP32 micro. They are shown on the display in TEST = true mode.

For example, we adjust the distance of the torch and paper thickness in order to get an average reading close to the left-hand column, and then record all five readings for that light level.

Study the code mentioned below, to see how these values are used to map and correct the sensor readings.



Code:

These values are entered into a function in your code, getAdjSense(Sensor, Value) in the form of a look-up table and mapping function, one for each sensor.