**Dome RFID Documentation**

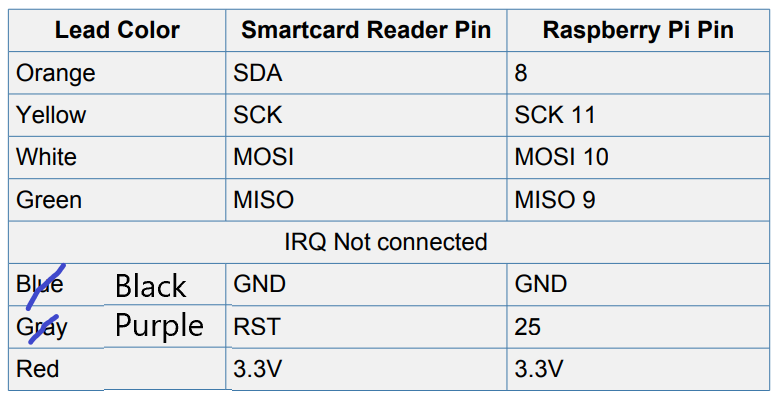
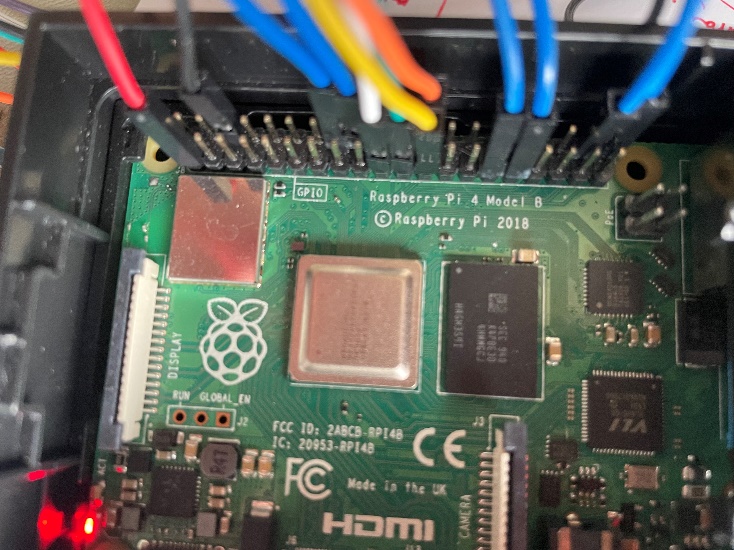
**Introduction:**

This document contains all relevant information on the use and maintenance of the system which measures the Draco2 dome’s azimuthal position. The system is composed of 6 RFID readers each in their own case attached to a mount. The readers are all wired to a single raspberry pi which has Ip address 192.168.184.19 on the astrolab network. On the inside of the dome 41 tags have been placed with equal separation on a 126-degree arc.

**Reader Assembly**:

The reader case was designed with fusion 360 and those files are in this folder. The case is assembled with 8 self-tapping screws (some force when screwing will be required due to the plastics resistance). The case is attached to the mount with a nut and bolt which can be tightened and loosened with an allen key.

The wires are attached to the RFID readers in the order (from left to right) of: Orange, Yellow, White, Green, GAP, Black, Purple, Red; see picture below for more detail.

The Pi needs to be set up with the following wire – pin combinations: Red – 1; Black – 6; Orange – 24 (GPI08); Yellow – 23 (GPIO11); White – 19 (GPIO10); Green – 21 (GPIO9); Blue – 29, 31, 38, 40, 16, 18, (GPIO5, GPIO6, GPIO20, GPIO21, GPIO23, GPIO24).

**Code:**

To get an instantaneous value for the azimuthal angle for the dome from any astrolab computer a Pyro5 name server needs to be run on the raspberry pi. This is done by inputting ‘pyro5-ns -n 192.168.184.19’ in a terminal on the pi and then running ‘python server.py’ in the ‘RFID/’ directory in another terminal. Then the file ‘client.py’ can be run by any other computer on the astrolab network which returns the dome’s azimuth.

01\_read.py – Reads from an RFID reader and waits until it can read a tag before continuing. It can only run with 1 reader connected.

02\_write.py – Writes to an RFID reader and waits until it can write to a tag before continuing. It can only run with 1 reader connected.

client.py – Remotely requests the azimuth of the Draco2 dome and returns it in degrees. It requires the name server and server code to both be running on the raspberry pi.

dome\_data.json – A dictionary containing several variables all in degrees: ‘home’ is the home position of the dome in degrees; ‘gap’ is the mean angular distance between 2 tags in the dome calculated by dividing the difference in angle between the 40th and 0th tags by 40; the reader variables (in the form ‘reader0’) are the positions of the dome when the 0th tag is sitting on the corresponding reader.

id\_dict.json – A dictionary of all the tags in the dome where the keys are the ids and the corresponding values are the tag’s positional numbers.

multiple\_handler.py – Defines a class which handles multiple RFID readers. The main read and write functions used by the subsequent files are methods of this class.

read\_multiple\_id.py – Continuously reads and returns only ids from 6 RFID readers.

read\_multiple.py – Continuously reads and returns ids and texts from 6 RFID readers.

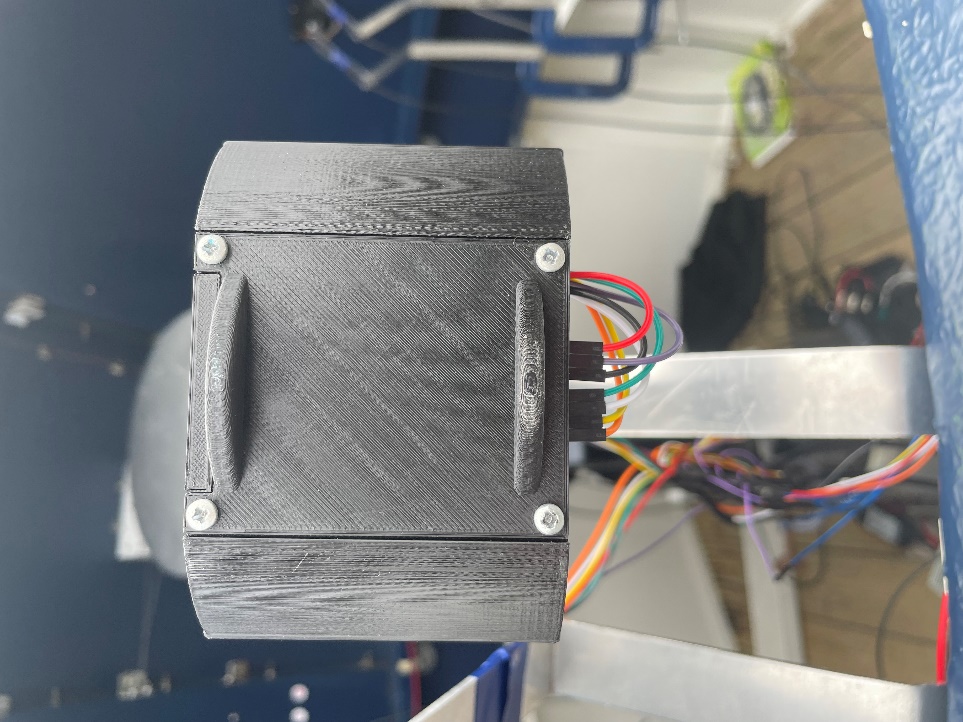
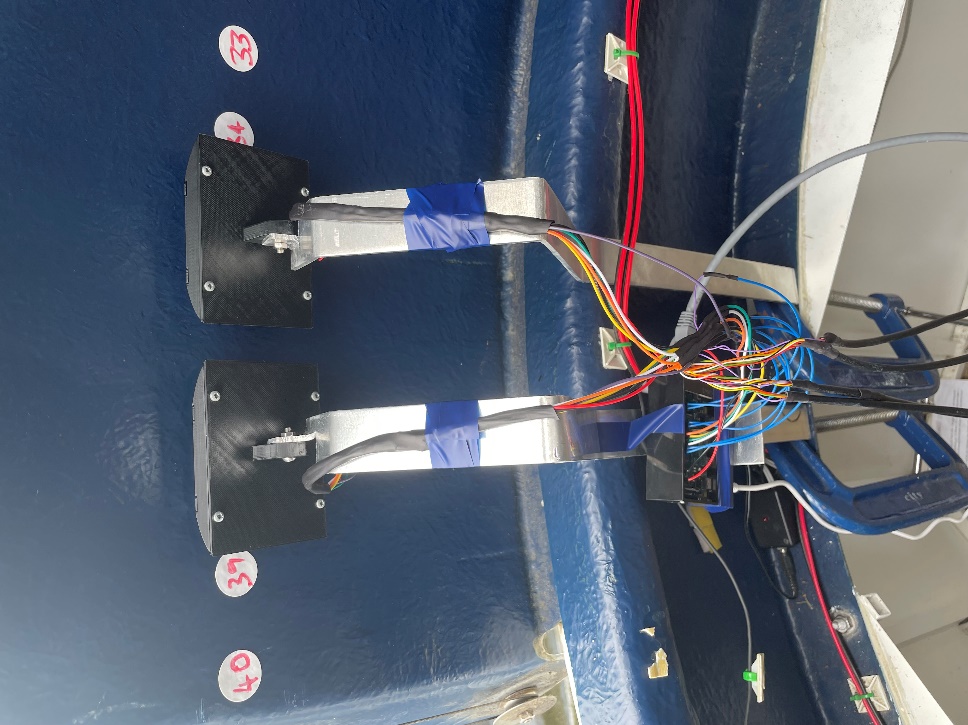
read\_position.py - Continuously reads from the 6 RFID readers and then calculates and returns the domes azimuth. Does this by taking the readers value from the dome\_data.json file and adding the tags number from the id\_dict.json multiplied by the value for ‘gap’ also in the dome\_data.json.

server.py – Needs to be run to allow other computers on the astrolab network to request the dome azimuth using client.py. Requires a name server to be running on the raspberry pi as explained above.

write\_id\_dict.py – Used to create the id\_dict.json file by reading from the RFID readers and creating a dictionary of all the tag texts where the corresponding ids are the keys.

Write\_many.py – Used to number tags starting from the variable ‘count’. It can only run with 1 reader connected.

**Setup and Calibration:**

The 41 tags have been placed at a height of ~23cm and each separated by 5cm measured from tag centre to centre. The ‘write\_many.py’ and ‘write\_id\_dict.py’ files can be used renumber the tags and register them in the ‘id\_dict.json’ file. The aluminium mounts need to be placed in the dome in 3 equidistant pairs, with the first in each pair placed on a tag and the second placed 2.5 tags on, so that the tags sit at the same height as the readers. The mounts can be bent into the required position which is with the curves of the faces of the reader cases just touching the dome, so they slide over it. To calibrate the positions of each reader spin the dome until the 0th tag sits on the reader and read the azimuthal position off the dome GUI, do this a few times to get a mean to record in the ‘dome\_data.json’ file.