Arduino Due (Programing Port) program to command azimuth and elevation antenna motors to track space objects. Works with any type of motors.

INTRODUCTIONS-

This is a CT4BB program that uses ARDUINO's open-source libraries to read SD cards, RTC Watch, I2C, SPI, etc. The system is designed to track DSN spacecraft probes and celestial objects using Horizons NASA's Ephemeris @ https://ssd.jpl.nasa.gov/horizons.cgi#top The data from this site is in txt and are printed in files with the name of the space objects. Ex: Copy Sun data and paste it in the SD card as SUN.txt, and Mars Reconnaissance Orbiter (MRO) data and save it in the SD card as MRO.txt.
The angle position are mapped in voltages(0->5V) to 0-> 4095 bytes using 12 bits resolution of ARDUINO DUE so, in must put on the Setup the instruction analogReadResolution(12).
The model of the (ARDUINO DUE Programming Port) used does not AutoStart when the voltage is applied. To fix this, you must solder a surface mount resistor (0603 size) or 1/4W tubular onto the board: solder a 10k resistor between the ERASE line and + 3.3V, close to Fet T3.
In practice, solder it across the 2 upper pins of T3.
https://forum.arduino.cc/index.php?topic=256771.60
Arduino due is 3.3 V logic. To connect with I2C LCD or RTC1307 with Logic 5V, a bidirectional voltage converter is required. The SD Card reader is 3,3V Logic but feed with 5V DC
External Pullup resistors of 5K must be placed to stabilize the digital inputs of pins 3,5,6,7 and 13, because the internal Pullup resistors of ARDUINO DUE are 50K and this high value causes instability and errors in the readings caused by induced noise, namely alternating current.

SYSTEM WORKING –

1 - The tracking system will only work when RTC date equals the recorded Ephemeris, because it will compare the actual date with the date of the Object Ephemeris in the SD card.
If the dates are equal, it will read all file in the SD. Next, operator must choose an object on the Menu with a single Button click, wait for object and select it. After selecting object all data of time and azimuth/Elevation will be read.
Now, the LCD will show alternately two frames: First frame- the name of the object, antenna offset the azimuth and the elevation in Second frame: Date and UTC time.

2 – Every minute the system reads the data file of the object and checks if the time is equal to RTC time. If time it is equal it reads the azimuth and the elevation. Then, compares these values with the previous Azimuth and Elevation to decide to go on CCW or CW exciting respective motor. Motors will run driving a multiturn potentiometer, with pre-calibrate voltages that are mapped to the angles. Motors will stop @ that precise Voltage/angle.

3 – A calibration must be made if you change mechanical hardware.
Normally it is only necessary to do it once. Calibration will make the motors of the Az and El to run from the minimum angle/voltage (0 °) where the microswitch is activated (on to off) to stop and after, motors will travel to the maximum angle/voltage (90° or 360°) to switch (on to off) the end microswitch to stop movement. (Microswitches are Normally Closed).
The azimuth switches must be placed collinear in the same plane, because the 0° coincides with 360°.
The microswitches or reed relays are normally closed contacts. Its opening places 3,3 voltage on pins 3, 5, 6, 7 that are powered by Pullups. This high logic level causes the Rotors to stop.

Note: It is possible to use only two microswitches (one for each motor) but this implies a change in the software due to the existence of a protection instruction intended to stop the motors in case the potentiometer reading fails. Motors will continue in permanent motion! A stop is needed which is done by the end microswitches. With some study and changes in the software it is possible to use only 2 microswitches but we think that the mechanical process is more effective and accurate using 4 microswitches.

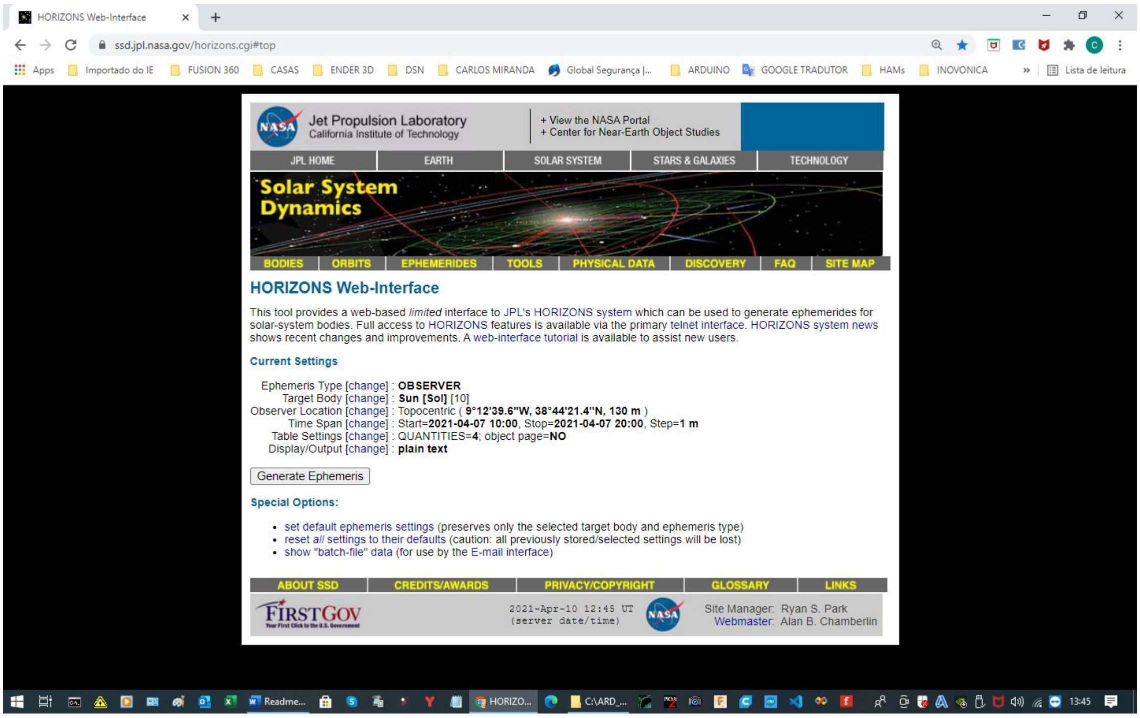
HARDWARE-

The pinout used on the Arduino Due is as follows:
Pin 11 digital OUTPUT: CCW control of Elevation motor
Pin 10 digital OUTPUT: Elevation motor CW command
Pin 8 digital OUTPUT: Azimuth engine CW command
Pin 9 digital OUTPUT: Azimuth motor CCW command
Digital pin 7 INPUT_PULLUP: Elevation end limit switch sensor 90°
Digital pin 6 INPUT_PULLUP: Elevation start switch sensor 0°
Digital pin 5 INPUT_PULLUP: Azimuth end limit switch of 360°
Digital pin 3 INPUT_PULLUP: Azimuth start switch sensor of 0°
Digital Pin 13 INPUT_PULLUP: Clicks of the Menu Button (single click, double click or long press).
Connections of 6 Pin IDC for SPI located in the middle of ARDUINO DUE to communicate with SD Card
IDC SPI-Pin 1 pin (dot or number 1) is digital MISO .
IDC SPI-Pin 2 Not used,
IDC SPI- Pin 3 SCK Clock communication with SD CARD.
IDC SPI- Pin 4 MOSI.
IDC SPI- Pin 5 RESET Not used
IDC SPI- Pin 6 GND Not used
Digital pin 4 CS (Chip select) for SPI communication with SD CARD
Analog A4 pin SDA for I2C communications RTC Watch DS1307 and I2C LCD
Analog A5 SCL pin for I2C communications RTC Watch DS1307 and I2C LCD
Analog A0 pin INPUT for analyzing the voltage of the azimuth potentiometer
Analog A1 pin INPUT for analyzing the voltage of the elevation potentiometer

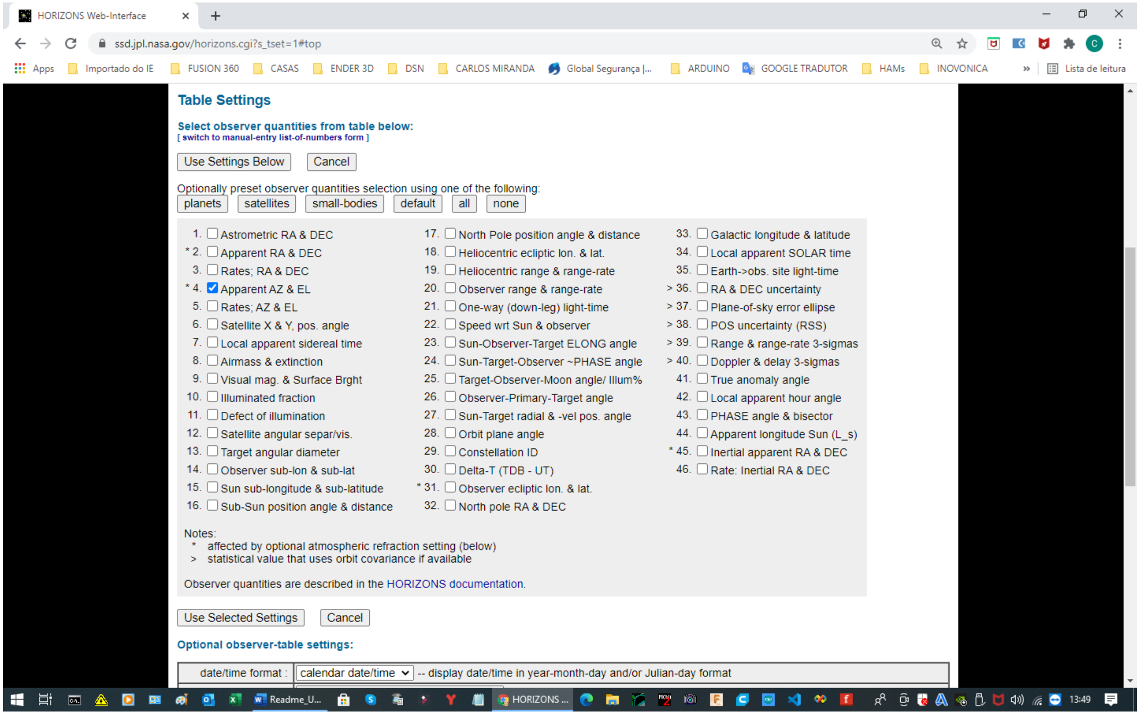
How get and format data from NASA Horizons Ephemeris to put in the SD Card:

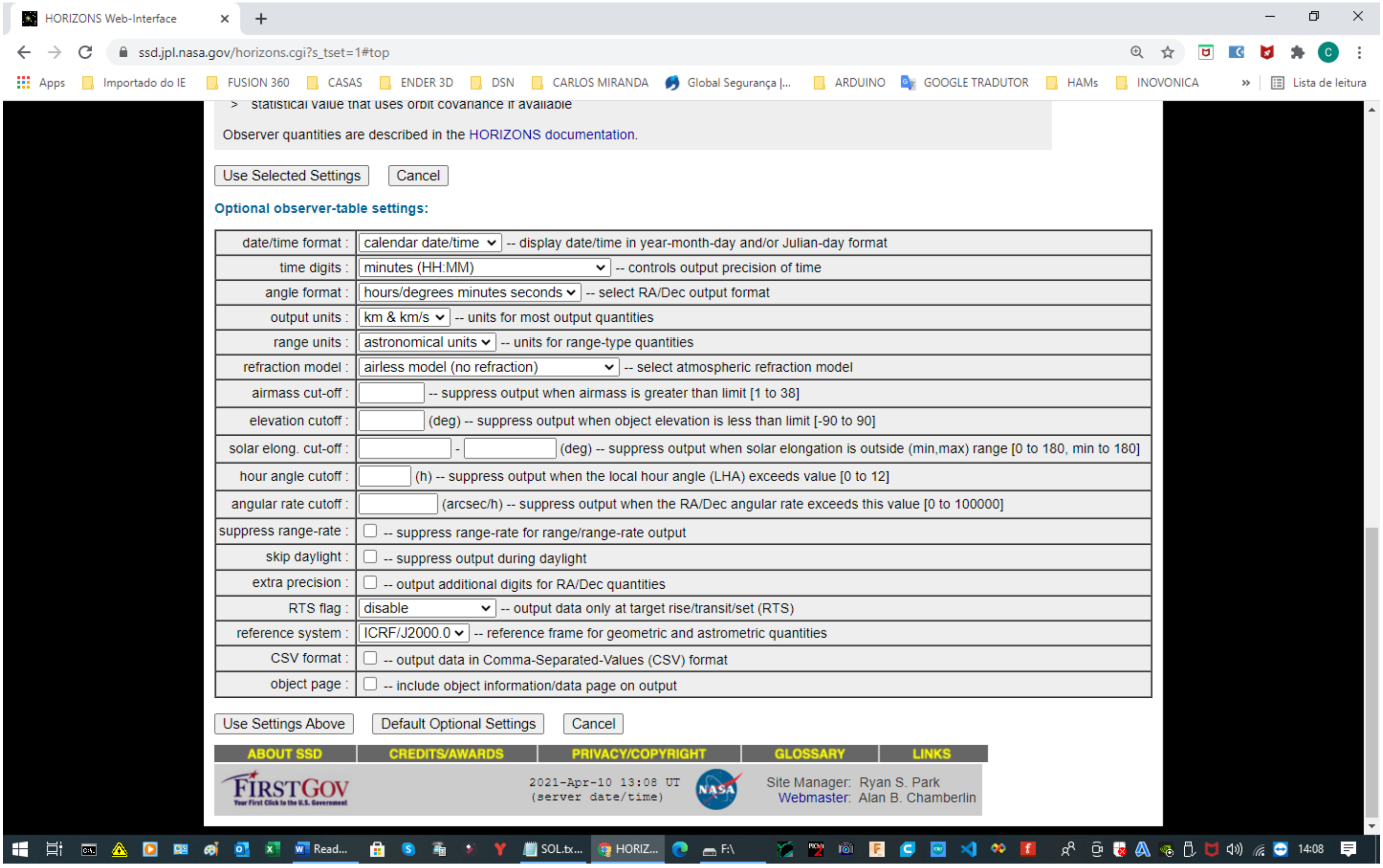
1 – Go to the site https://ssd.jpl.nasa.gov/horizons.cgi#top

2 - Fill the Page with your requirements



3 - Open the Table Settings and check as shown



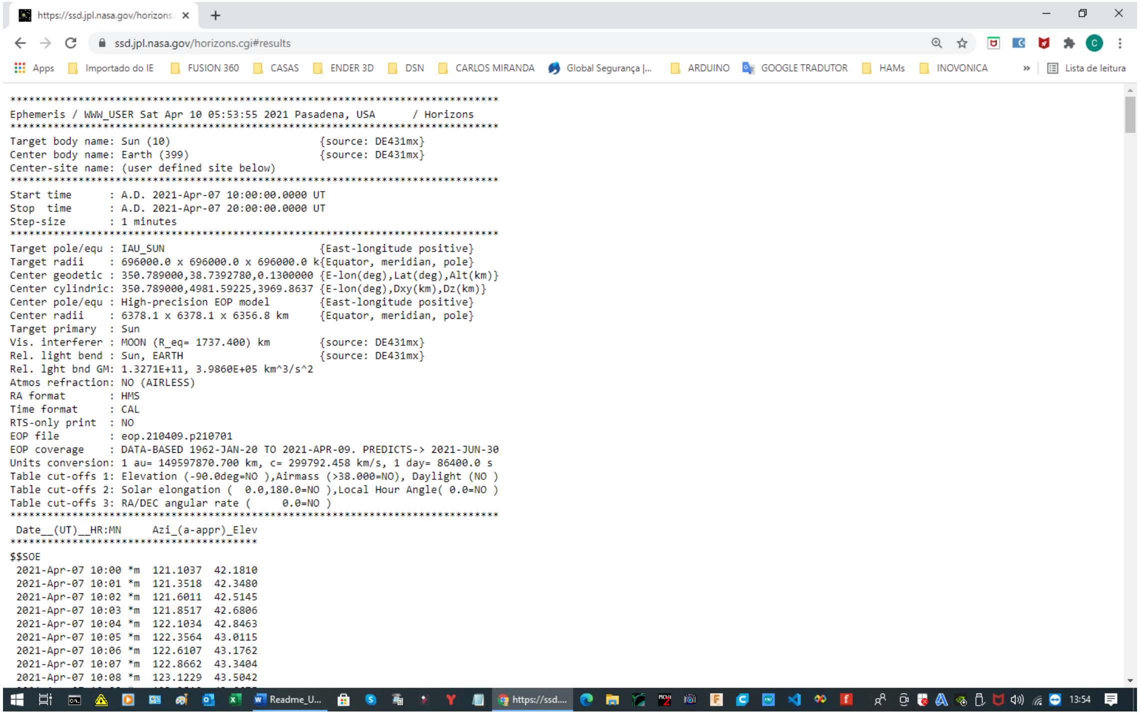


4 – Click Use Selected Settings

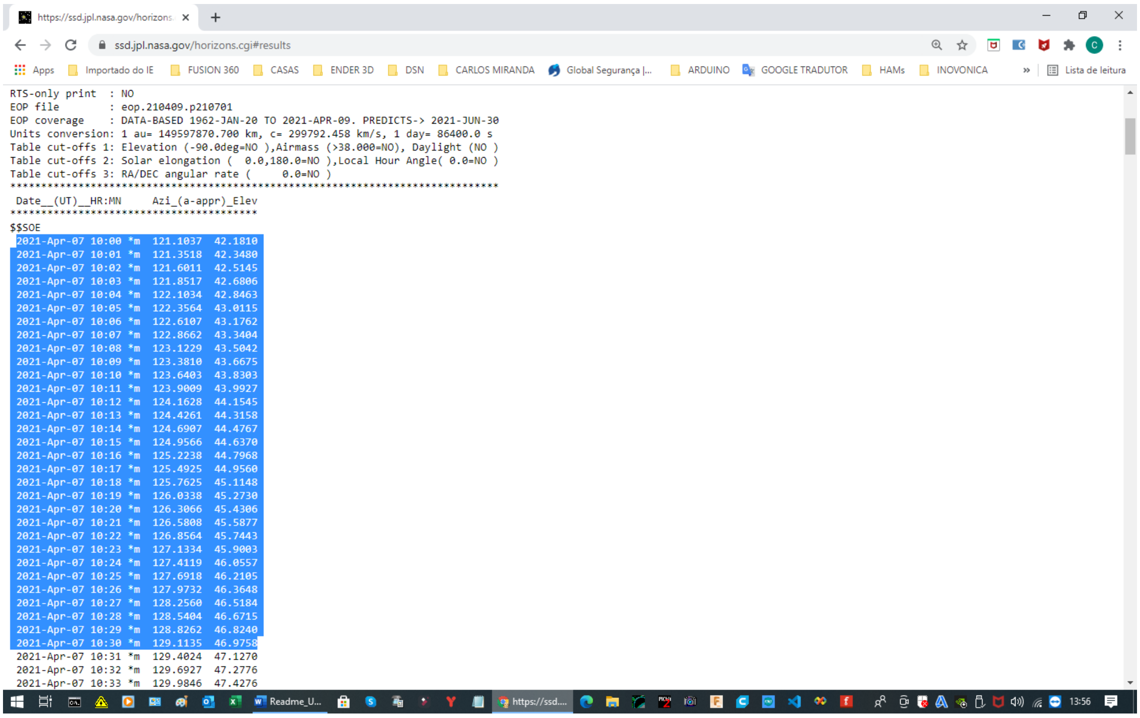
5 - Select Target Body (Sun, Moon, MRO, Perseverance, Parker Solar Probe, etc)

6 – Click Generate Ephemeris

7 – You get:



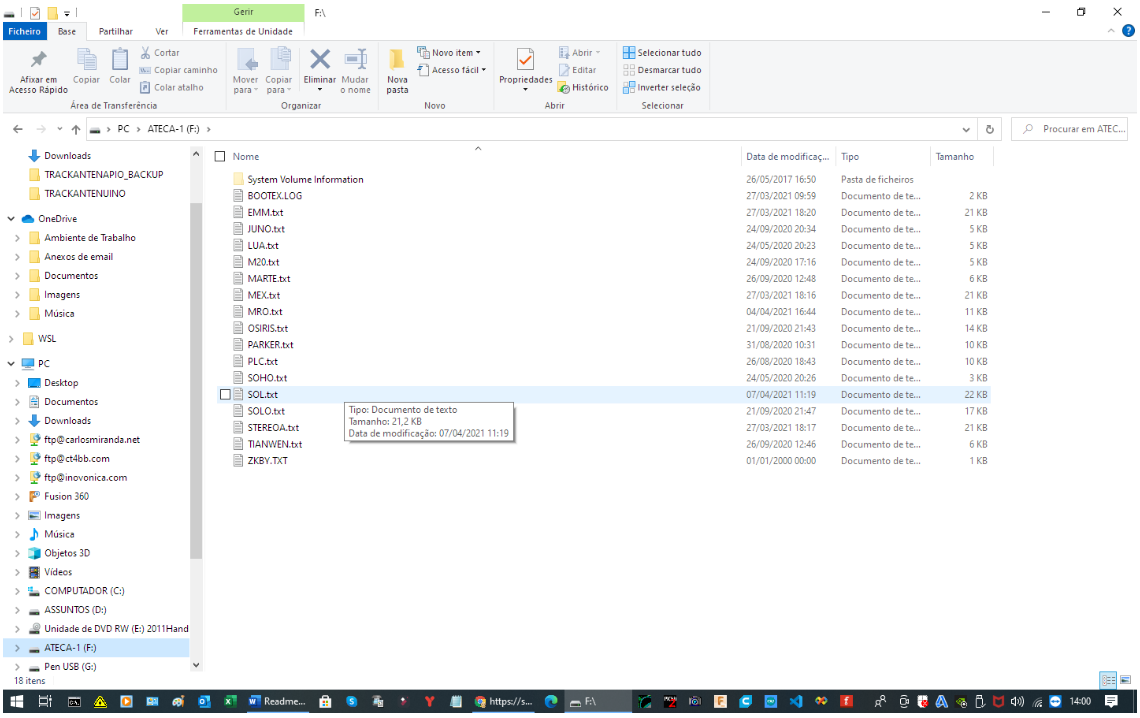
8- Select data columns with Date Azimuth and Elevation o object like this



This will track the object between 10:00 and 10:30 AM UTC @ 21 Apr 07

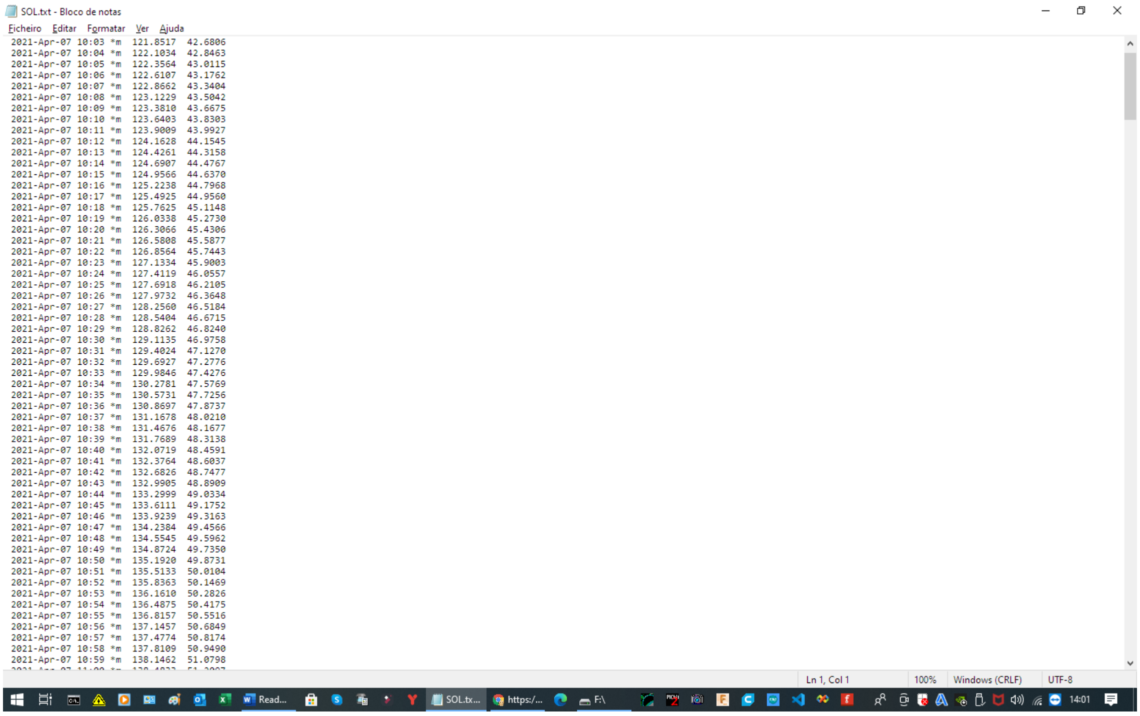
Copy this and put on the SD Card inside a folder with the object name.

Like this SD Card shows some txt files where to paste the data:



Data is from SOL (Sun), so, open the folder SOL.txt (Sun.txt) and paste inside.

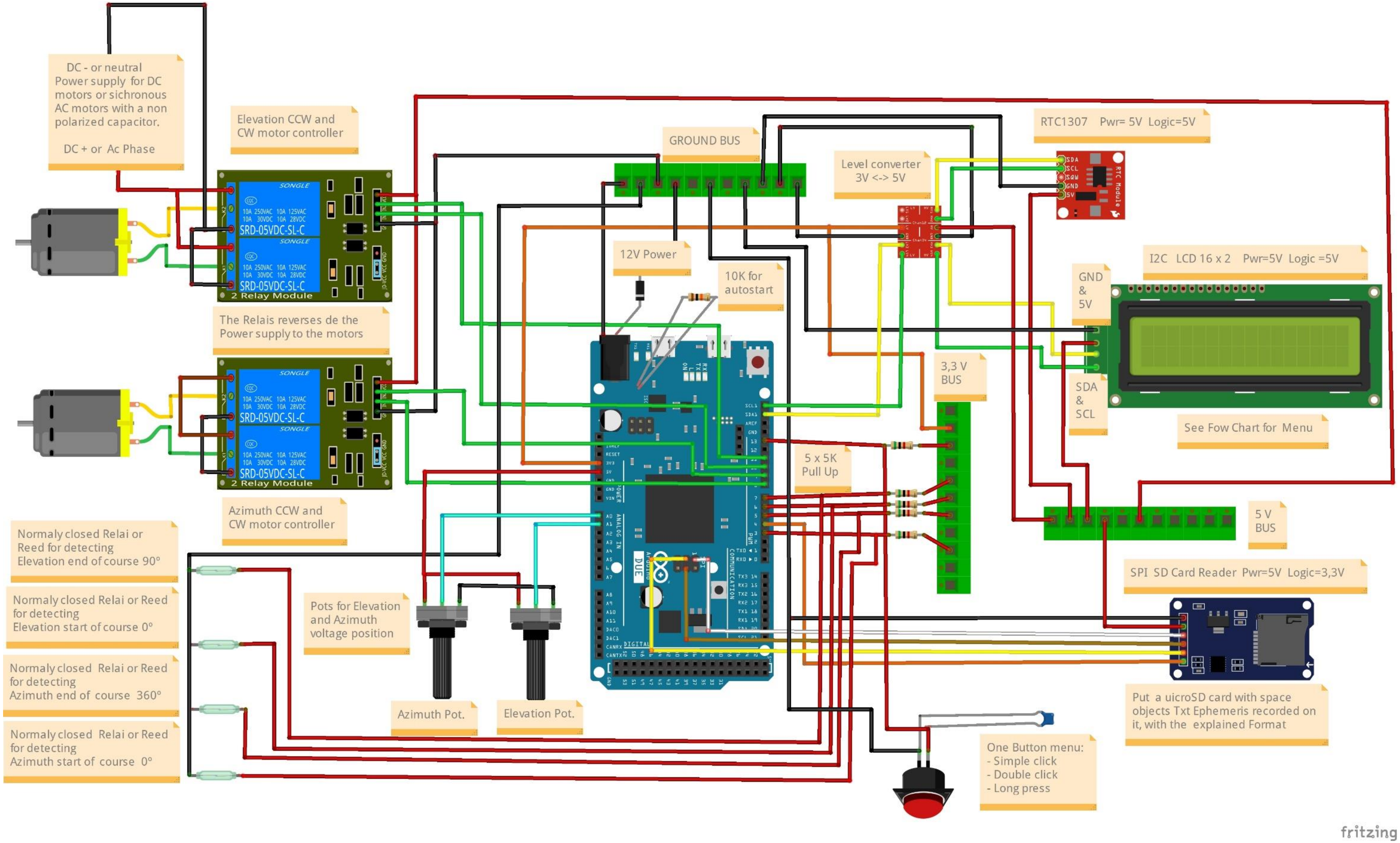
Remember the first column must be a space!

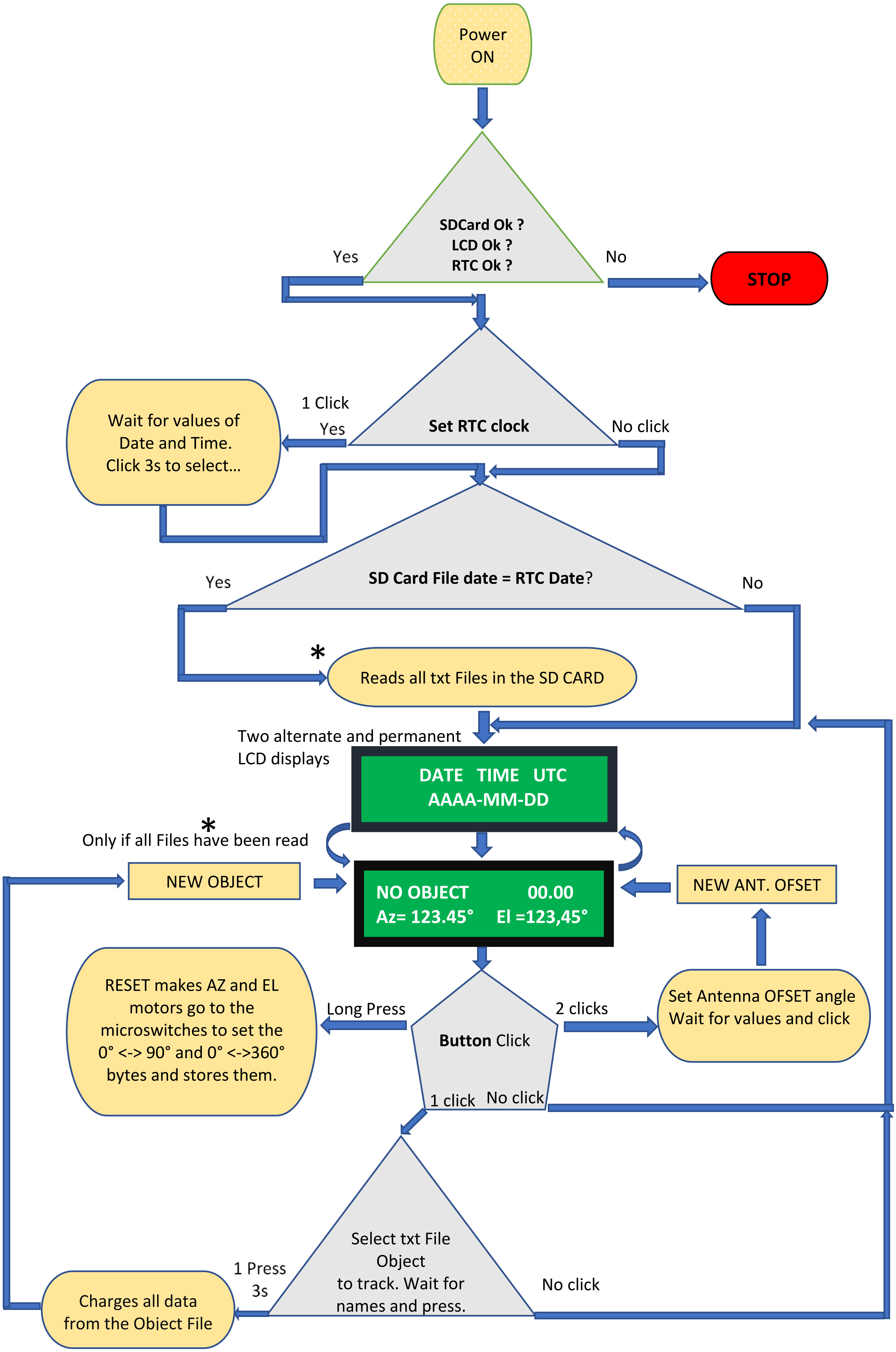


Save.

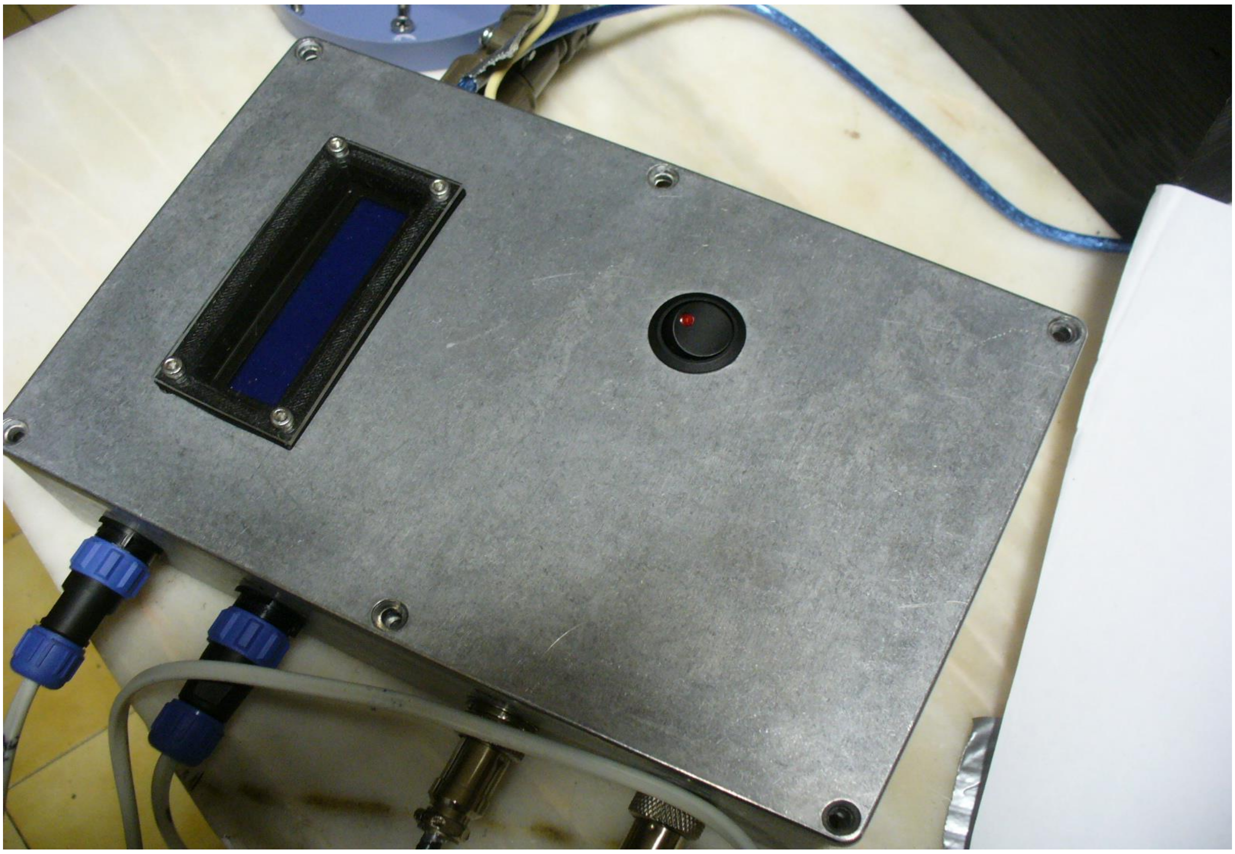
Do the same for all the objects you want to track on a specific day

Put the card in the Tracker and follow the menu to set the clock if needed, to select the object, and antenna offset. Wait for the automatic commands…

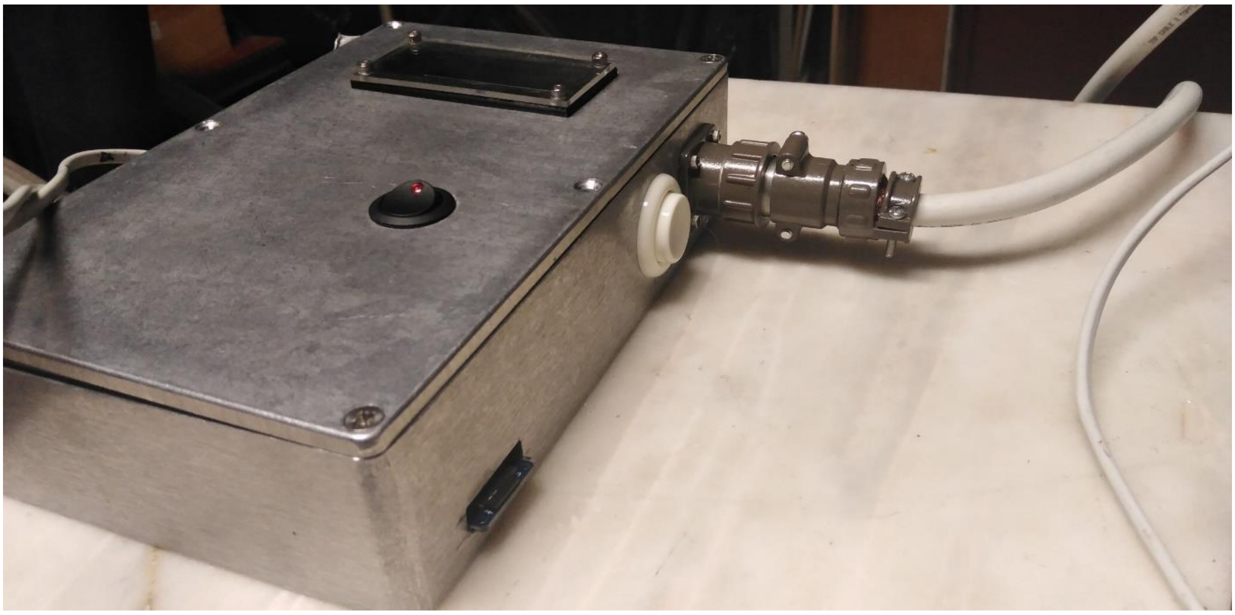

DC - or neutral
Power supply for DC motors or sichronous AC motors with a non polarized capacitor.
DC + or Ac Phase
Elevation CCW and CW motor controller
GROUND BUS
RTC1307 Pwr= 5V Logic=5V
Level converter 3V <-> 5V
12V Power
10K for autostart
I2C LCD 16 x 2 Pwr=5V Logic =5V
GND & 5V
The Relais reverses de the Power supply to the motors
3,3 V BUS
SDA & SCL
See Fow Chart for Menu
5 x 5K Pull Up
Azimuth CCW and CW motor controller
5 V BUS
Normaly closed Relai or Reed for detecting Elevation end of course 90°
SPI SD Card Reader Pwr=5V Logic=3,3V
Normaly closed Relai or Reed for detecting Elevation start of course 0°
Pots for Elevation and Azimuth voltage position
Normaly closed Relai or Reed for detecting Azimuth end of course 360°
Azimuth Pot.
Elevation Pot.
Put a uicroSD card with space objects Txt Ephemeris recorded on it, with the explained Format
Normaly closed Relai or Reed for detecting Azimuth start of course 0°
One Button menu:
- Simple click
- Double click
- Long press
fritzing



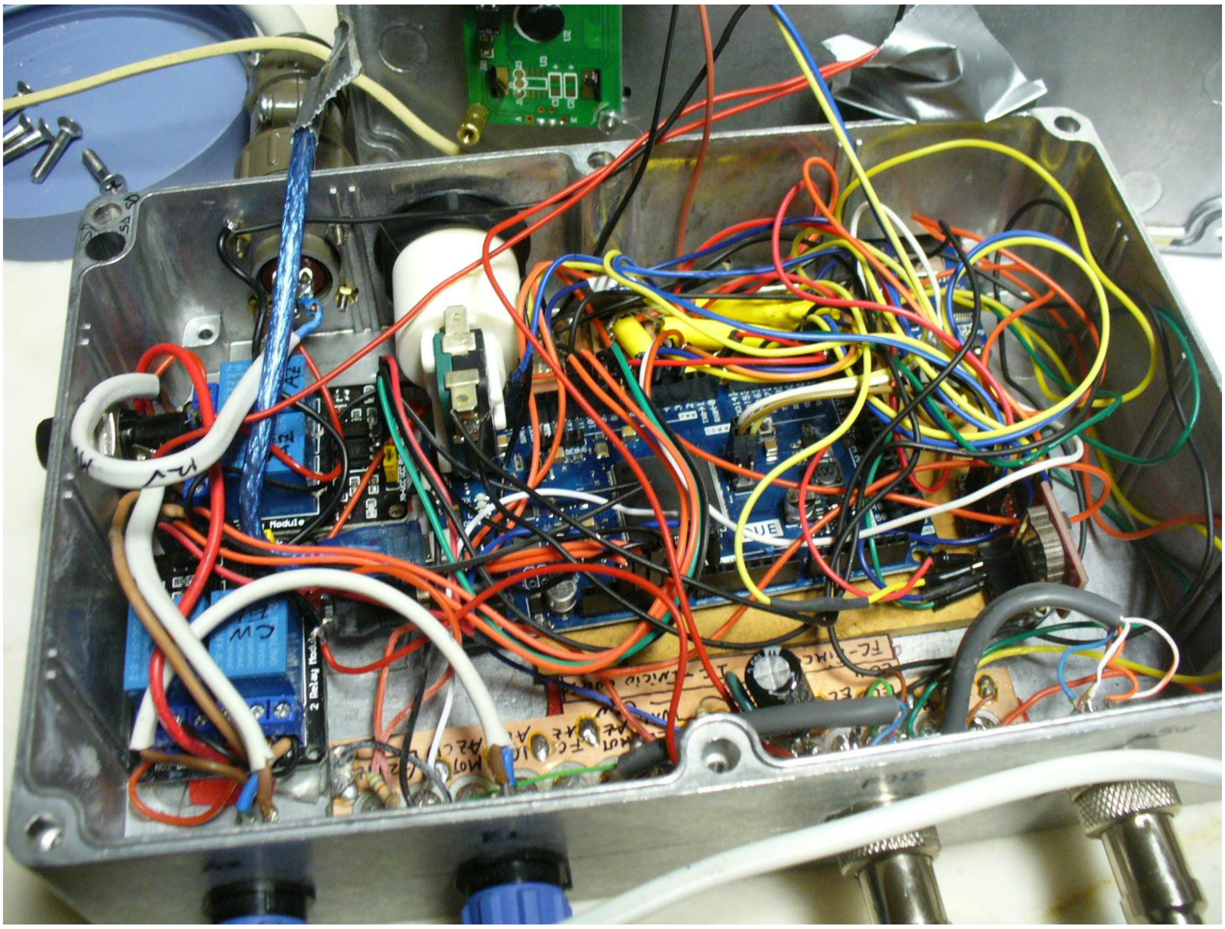
Flow chart Menu



Panel, ON/Off switch AZ/EL and micro-switches connectors



- Connector for ARDUINO 12 V and AC or DC Power for motors
- One robust button for Menu
- Micro-SD Card Slot



Inside the Mule Prototype

Date Time screen

OBJECT Name, OFFSET, Azimuth and Elevation @ the Time of previous screen

3D printing Elevation and Azimuth rotators to test the Tracker.
With this white plate and the rood shadow, it got a zero shadow in one hour of tracking the Sun.

More tests are now underway with a set of two-rotor type CDR Master Rotors @ 24 V AC and a 200 uF condenser without polarity to reverse the moving direction of the synchronous motors.
Good work and 73

CT4BB

Carlos