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 NASA's Ephemeris @ https://ssd.jpl.nasa.gov/horizons.cgi#top The data from this NASA site is in txt, 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. You can copy data of at least 8 space objects to track in a specific day. Libraries in the folder you download. Any new versions of the libraries will force to change the software in some parts to agree with the libraries changes. This is a long program with mapped voltages(0->5V) to 0-> 4095 bytes using 12 bits resolution so, have on it the instruction analogReadResolution(12); The model of the (ARDUINO DUE Programming Port) that I used does not start when the voltage is turned on. The fix is to solder a surface mount resistor (0603 size) onto the board: solder to 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 SD readers or 5V RTC, a bidirectional voltage converter is required. See the schematic.
Note2: 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 –
The tracking system will only work on that day because it will compare the actual date with the date of the saved Object Ephemeris in the SD card If the dates are equal, it will read the actual Time- Hour Minute and Seconds - to compare with an equal Time in the file of the selected space object. If The times are equal, it will catch the azimuth and the elevation of the object and will activate the relays of the Azimuth and Elevation motors. Motors will run driving a multiturn potentiometer, with pre-calibrate voltages that are mapped to the angles. Motors will stop @ that precise Voltage/angle. 2 - Calibration is made before tracking. 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 a microswitch is activated to stop and after, motors travel to the maximum angle/voltage (360°) to switch on another microswitch to stop movement. 3 - The system reads the file and checks if the date is correct (if not, everything is null). If it is correct, the information loop for the actual UTC time and the Azimuth / Elevation in which the antenna is located begins. Every 5 minutes it compares the real time with those in the file if it is the same, reads the azimuth and the respective elevation. Then, compare these values with the previous Azimuth and Elevation to decide the CCW or CW movement of the respective engines. For good LCD messages, you must translate to your language in the firmware, all LCD comments, and menus

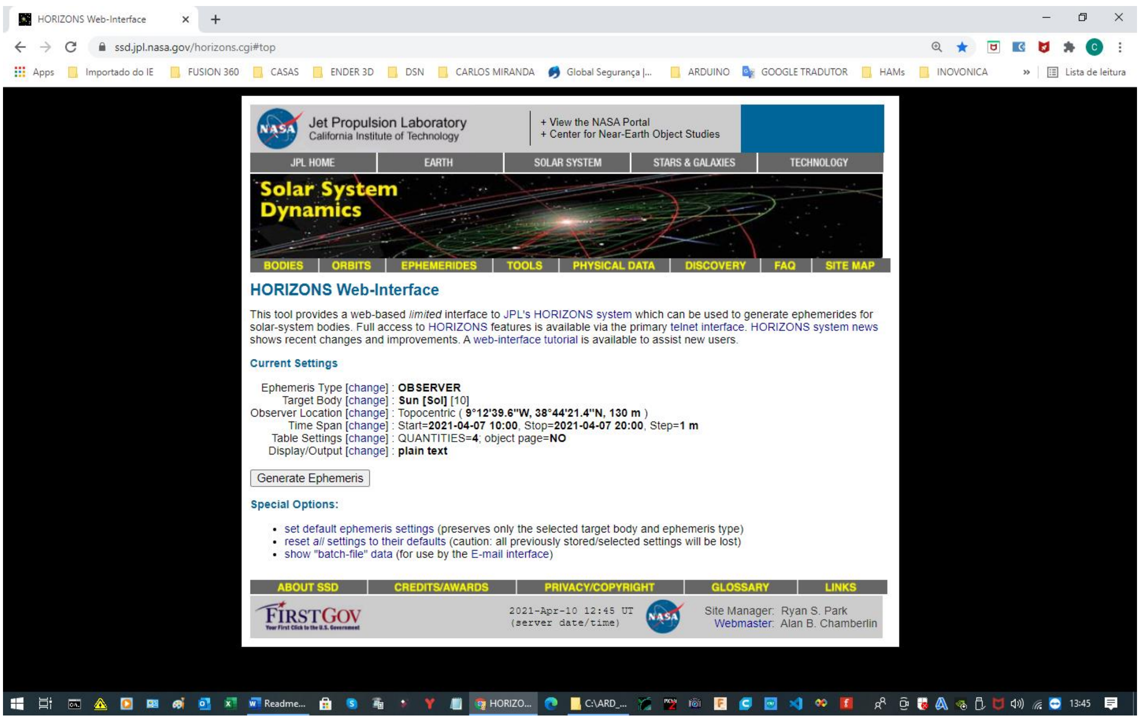
HARDWARE-
The pinout used on the Arduino Due is as follows:
Pin 11 digital OUTPUT: CCW control of Elevation motor
Pin 10 digital OUTPUT: Elevation motor CW command
Pin 8 digital OUTPUT: Azimuth engine CW command
Pin 9 digital OUTPUT: Azimuth motor CCW command
Digital pin 7 INPUT_PULLUP: Elevation limit switch sensor
Digital pin 6 INPUT_PULLUP: Elevation start sensor
Digital pin 5 INPUT_PULLUP: Azimuth limit switch
Digital pin 3 INPUT_PULLUP: Azimuth start sensor
Digital Pin 13 INPUT_PULLUP: Clicks of the Menu Button (3 options: single click, double click or long press 3 seconds).
Connections of 6 Pin IDC for SPI located in the middle of ARDUINO DUE. Is 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

STRUCTURE AND OPERATION - (In the text, SOL means Sun).
When connecting, the system runs the Settings and Setup (). In the setup (), will read the Date and Time and the day in the DataHoraRTC () function, then in the setup (), it will compare the Year, Month, day of the card with the actual RTC. System uses as start the reading of the date of the file SOL.txt defined inside the setup () as a start reference to put the Tracking structure pointed to the Sun. The function that does this operation is ler_SOL_Data (). In this function, at the end of reading and registering, it will return to the setup () to make Date comparation with the function compara_Datas run () . Comparing the dates and, if they are the same, it will read the directories of all SDCard files. With the function * ler_DIRECTORIA_SD (File dir, int numTabs) reads the directory and stores in the Strings A, B, C, D, E, F, G, H the menu is prepared for choosing the file to open. Space Objects (Probes) selection, is done in PROBES with sondastr = "Click to open" that will appear on the LCD. When clicking to open with the probes () function, the object file to be followed is chosen. When defining this choice, all data from the object in question is loaded into the Data (u) buffer in a matrix that will be carefully accessed. The Loop then starts with the selected object where, every 5 minutes, it will be checked whether the actual Hour and Minutes are in the respective file. If there are HOURS = HORASSD and MINUTES = MINUTOSSD, the respective AzNovo and ELNovo will be read soon using the lerHM_AZ_EL () function. These data are available for processing the movement of the antenna. During the 5-minutes interval to read new data, the LCD will inform the UTC Date / Time and the azimuth and also where the Az/El is pointed or, in the case of movement, where the antenna is going to new Az/El. The program uses ARDUINO IDE by opening the file TRACKANTENUINO.ino which is inside the folder TRACKANTENUINO. The libraries used in the program are all within this folder. The use of more recent versions of the libraries may require the correction of the instructions in the software. For the operation of the system, it is not necessary to use more recent versions of the libraries. This program, essentially, operates relays that can excite other power relays or power contactors that, in turn, excite bidirectional power motors by switching contacts and connections. The stopping point of the motors is defined by a voltage 0-> 5V mapped in movements 0-> 90 degrees for Elevations and 0-> 360 degrees for Azimuths. These voltages can be obtained by rotating a multi-turn potentiometer engaged in the motors. It is a mechanical question that each one can study and develop, making the turns of the potentiometers to be mapped in the voltage of 0-> 5V and in the respective angular excursions of 0-> 90 degrees Elevations and 0-> 360 degrees Azimuths.

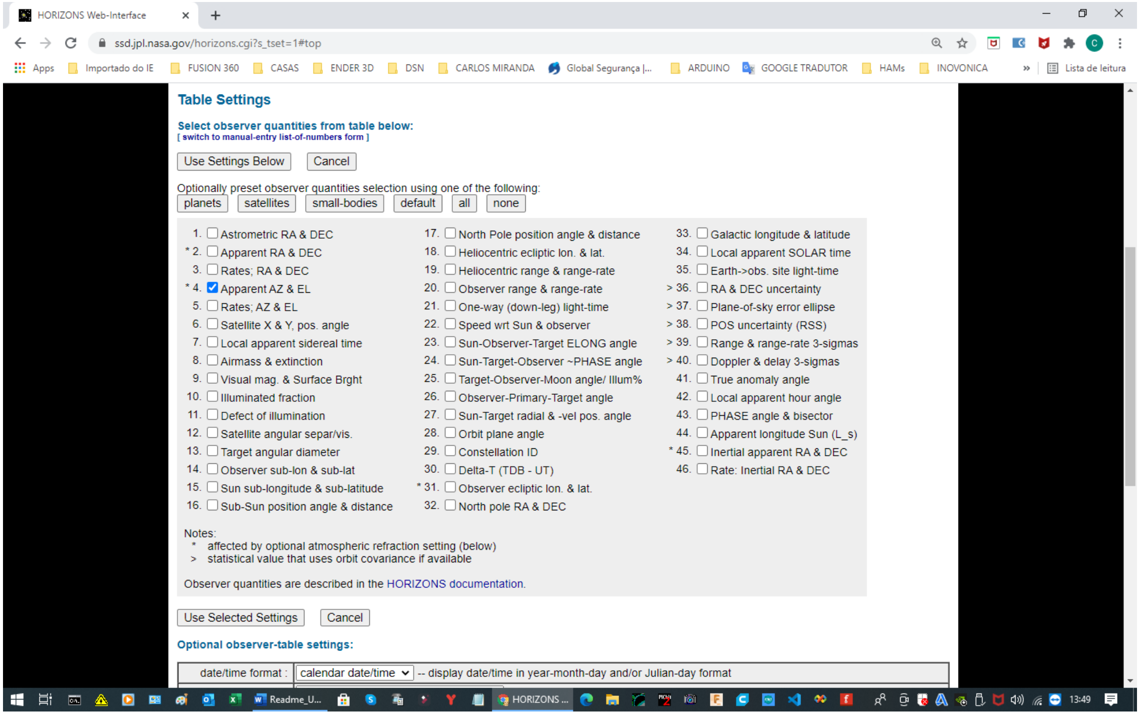
How get and format data from NASA Horizons Ephemeris to put in the SD Card:

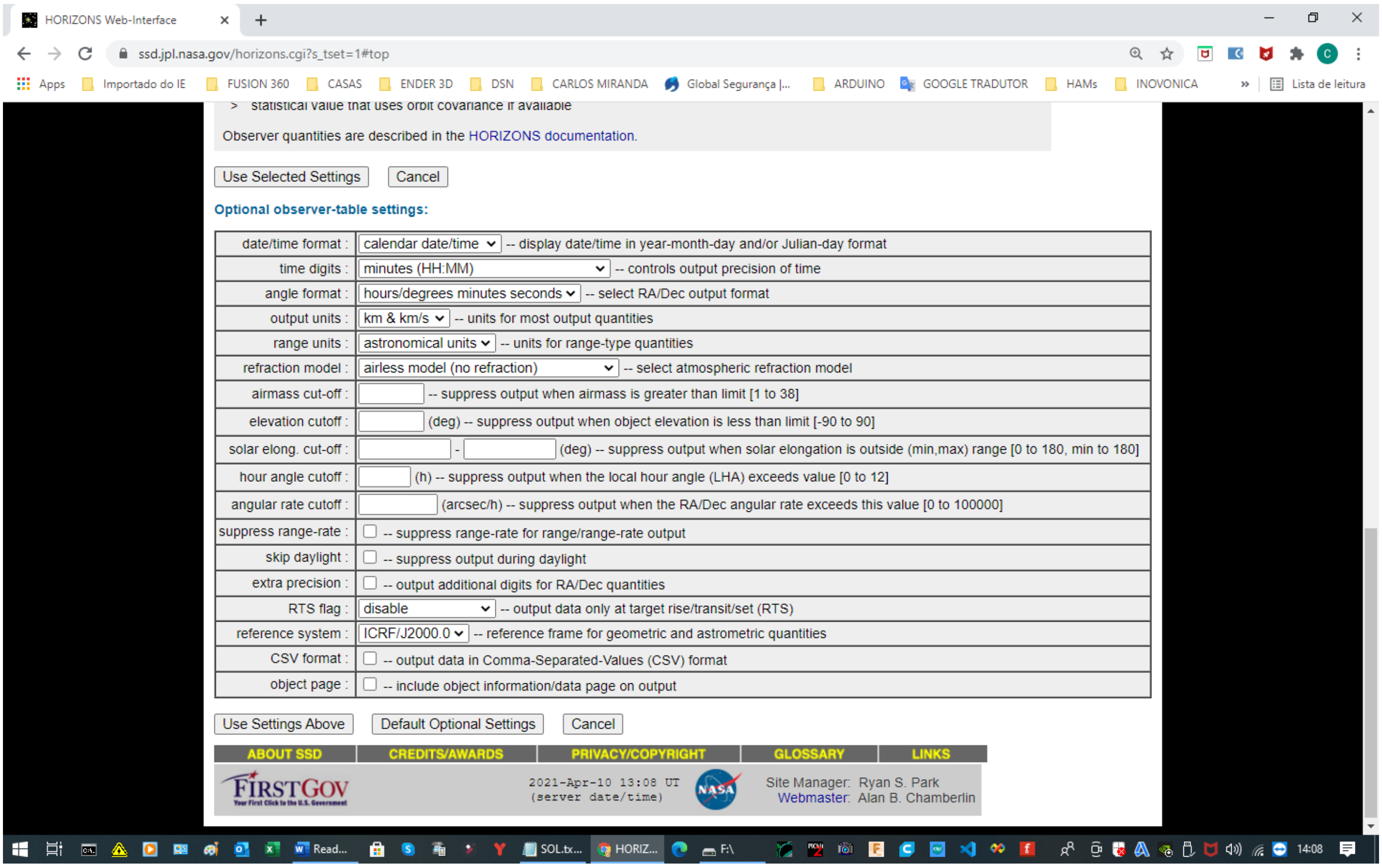
1 – Go to the site https://ssd.jpl.nasa.gov/horizons.cgi#top

2 Fill the Page with your requirements



3 Open the Table Settings and check as shown



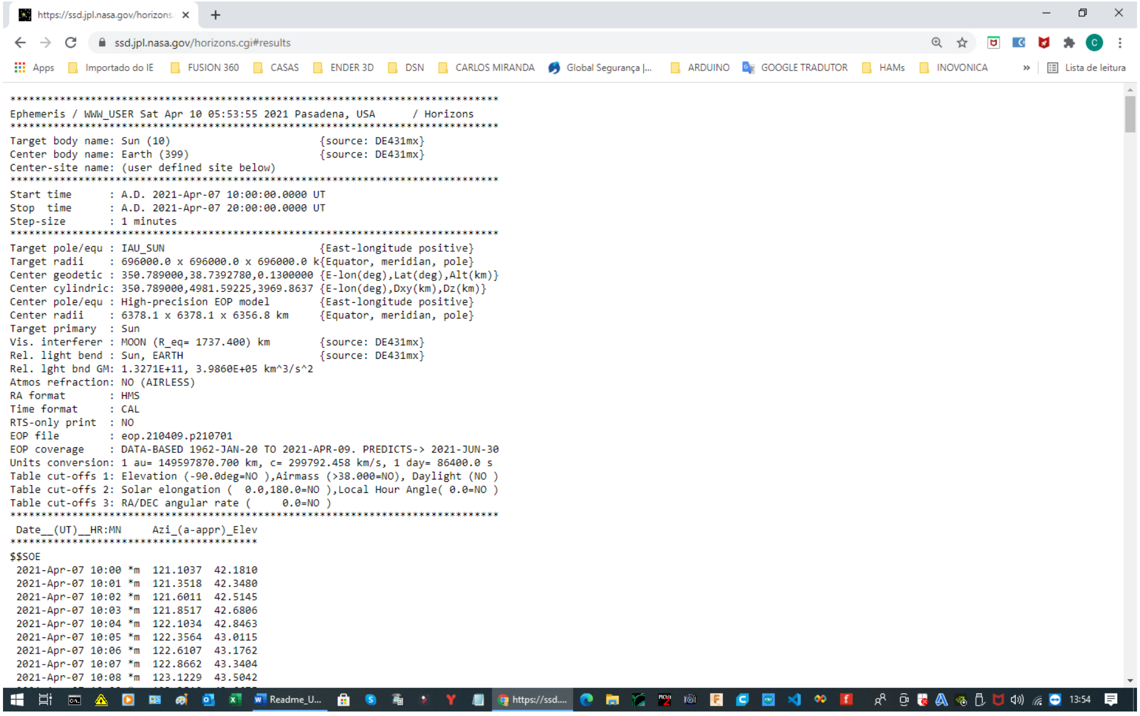


4 – Click Use Selected Settings

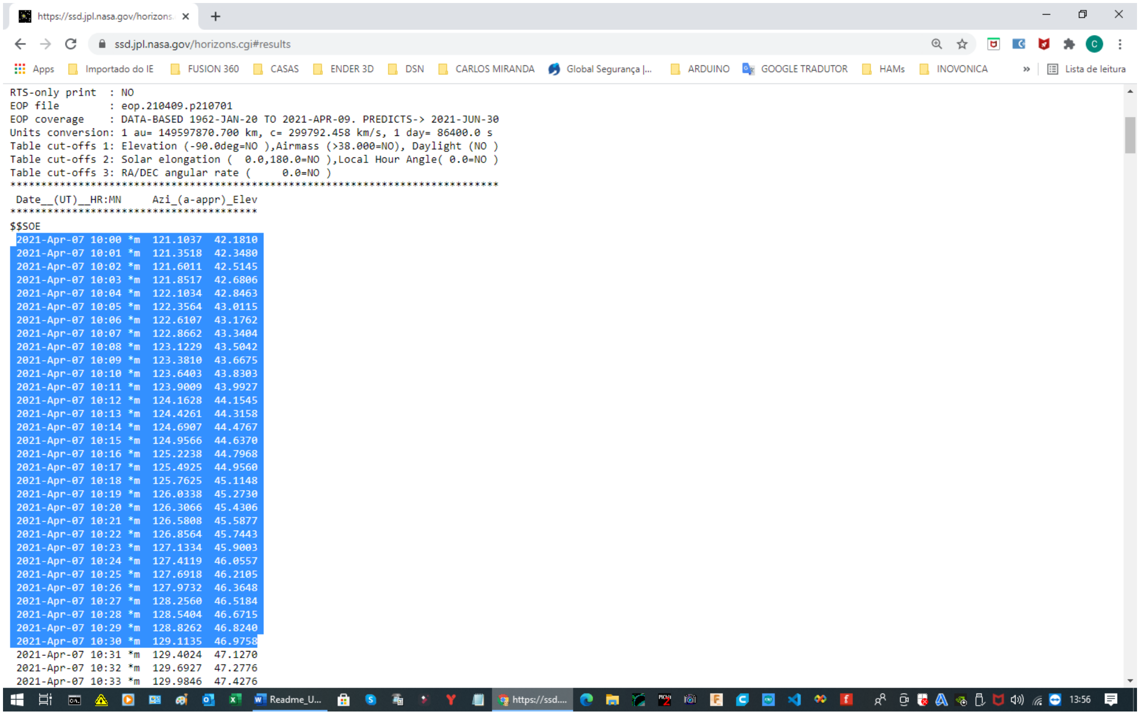
5- Select Target Body (Sun, Moon, MRO, Perseverance, Parker Solar Probe, etc)

6 – Click Generate Ephemeris

7 – You get:



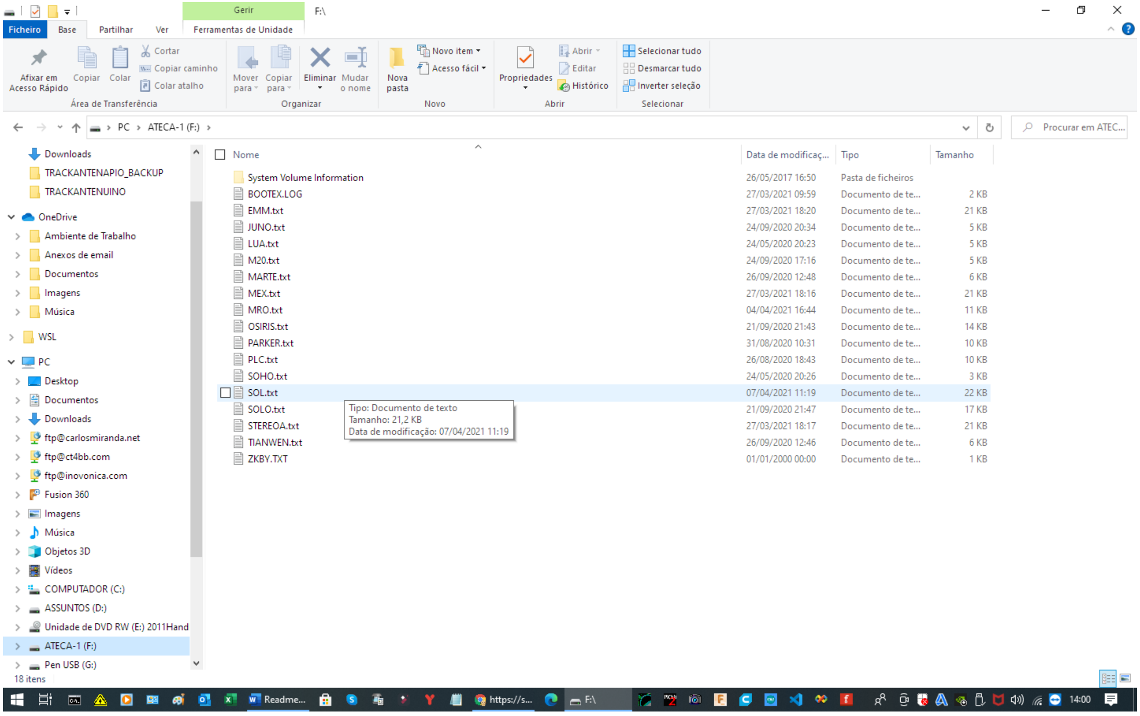
8 Select data columns with Date Azimuth and Elevation o object like this



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

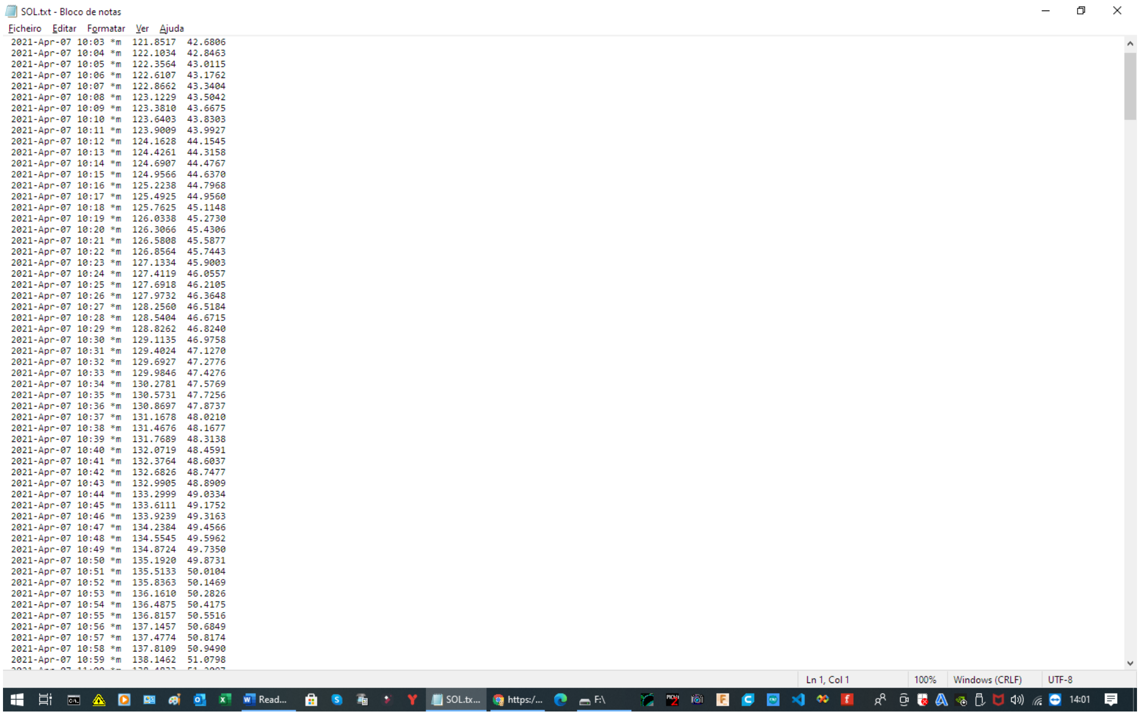
Copy this and put on the SD Card inside a folder with the object name.

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 !

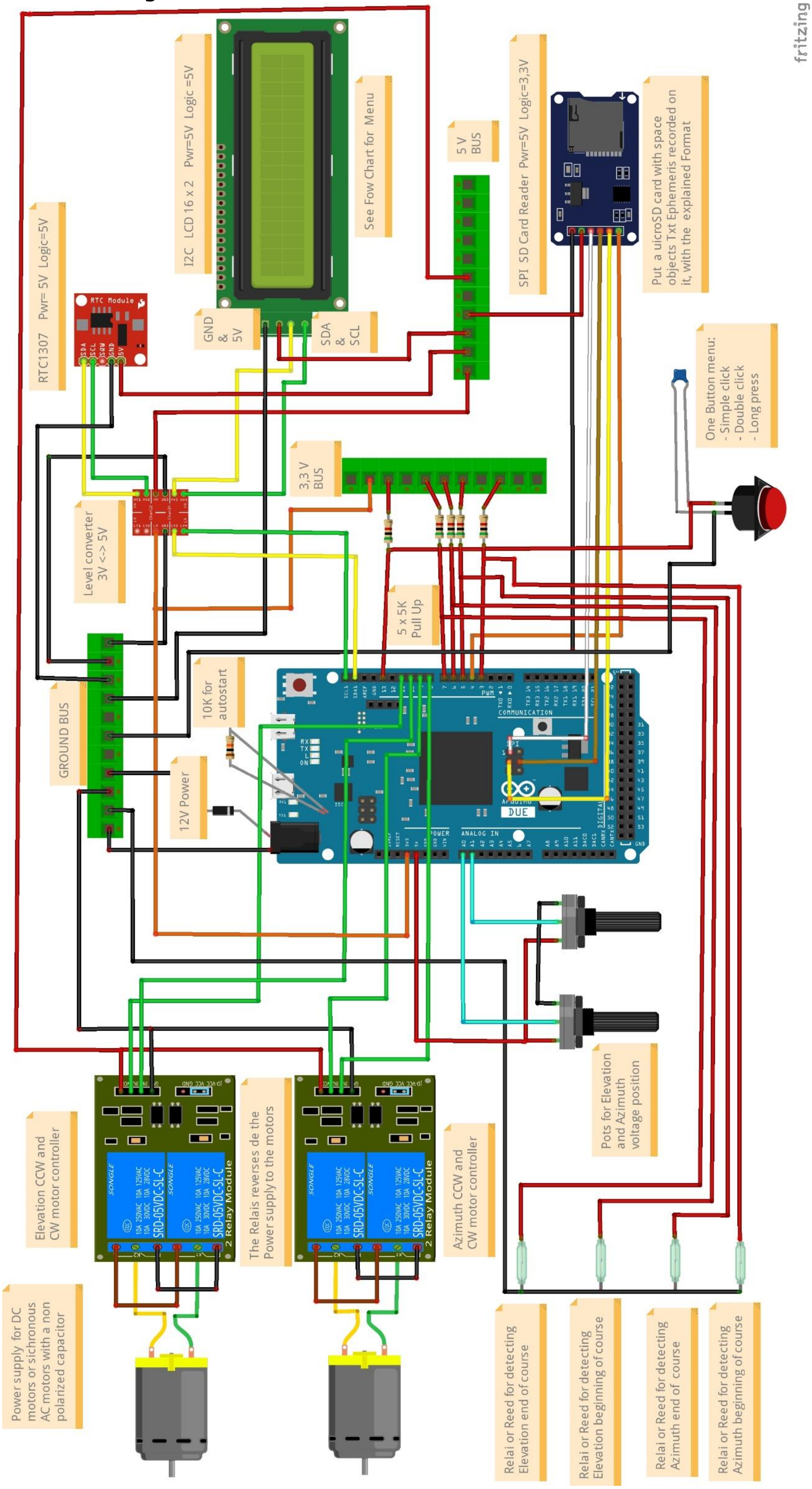


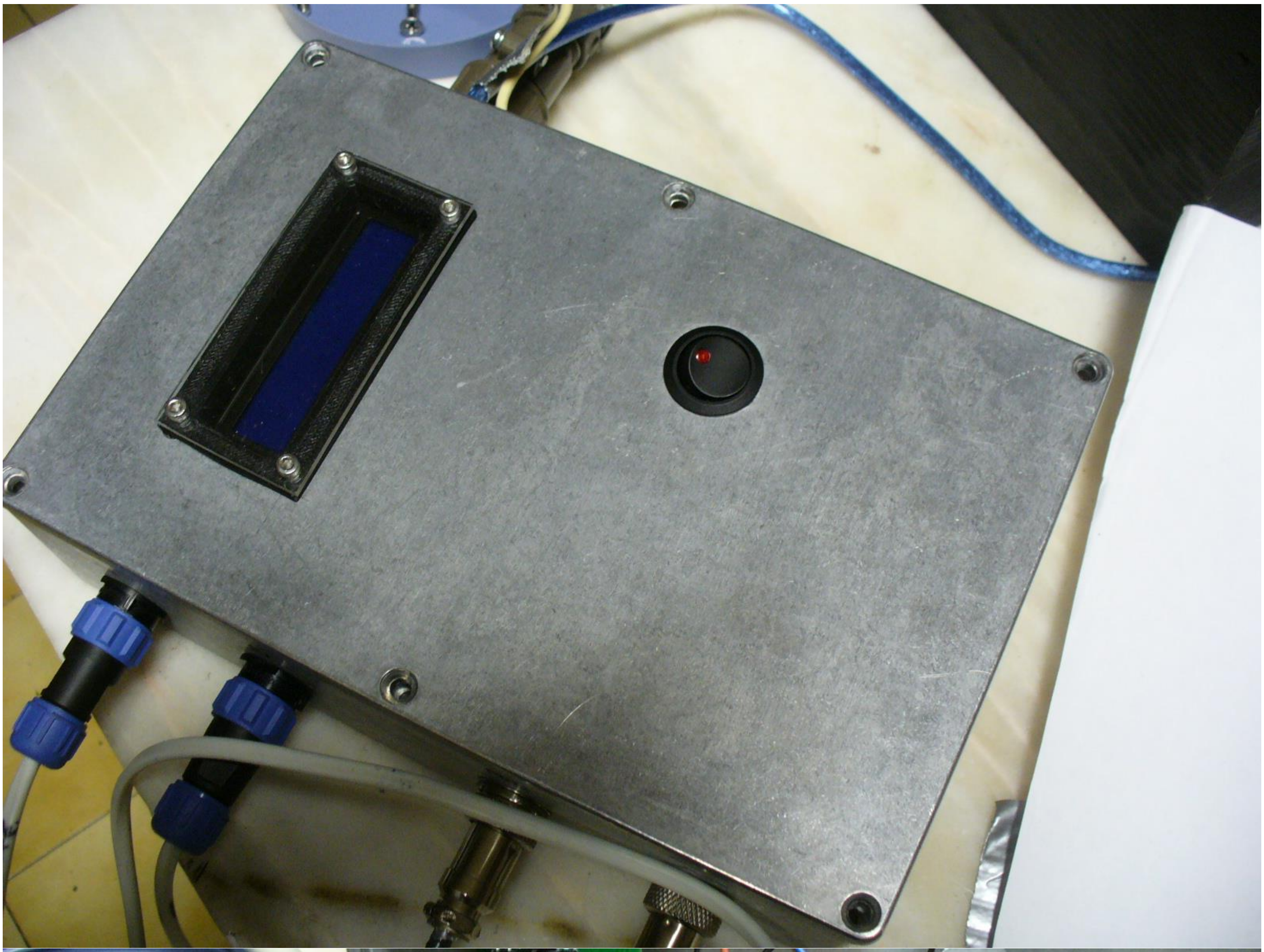
Save.

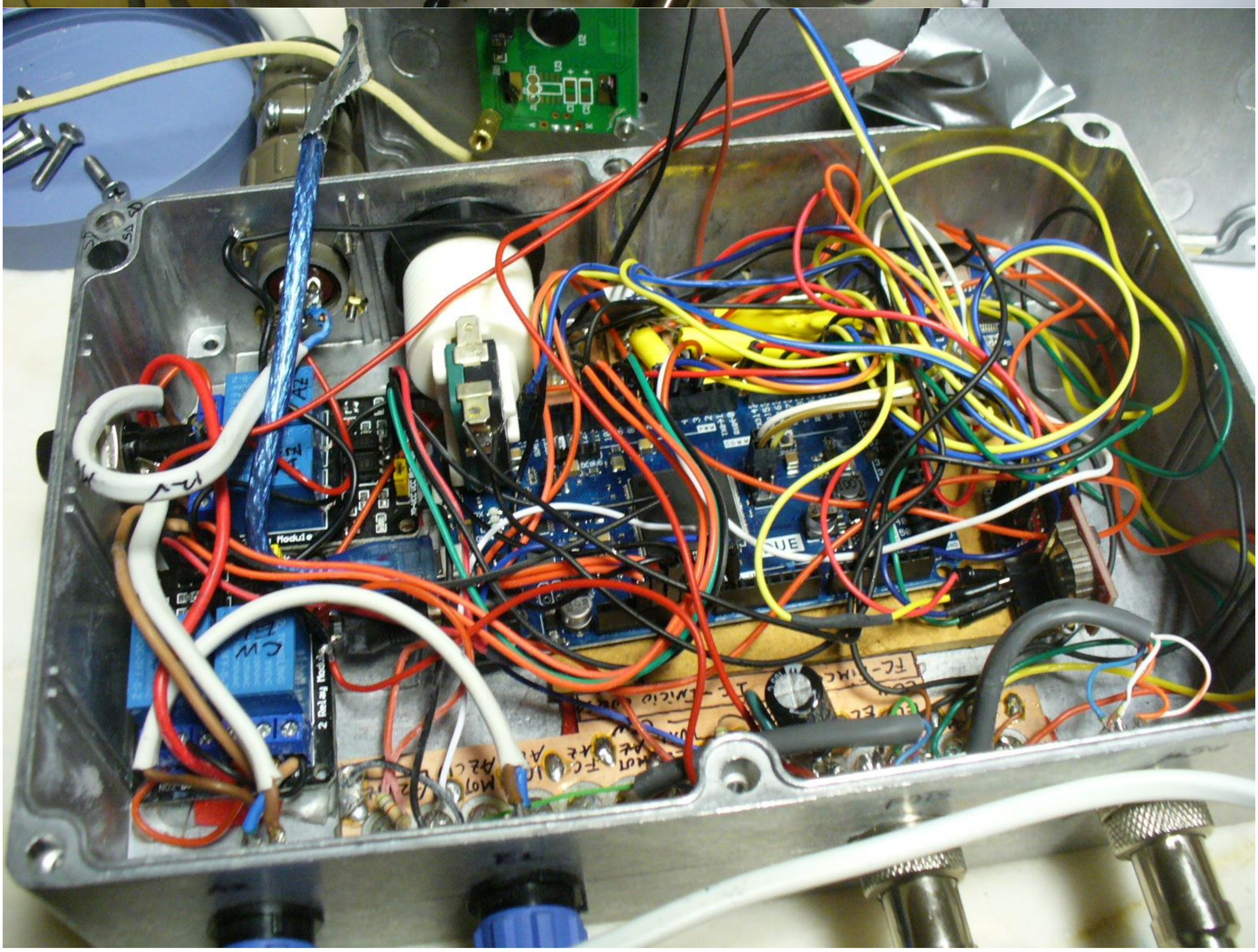
Do the same for all the objects you want to track on a specific day

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

ARDUINO Diagram connections:







Good work es 73

CT4BB

Carlos