Version 1.0

08-08-2023

# STARTUP TRACKING SYSTEM



Translated by CT1BYM

# INDICE


STARTUP TRACKING SYSTEM .......... 2
Encoders .......... 4
  Elevation SOLAR-360 MB .......... 4
    Verification .......... 6
  Azimuth SE58Sx .......... 7
    Verification .......... 7
DriverDish .......... 8
JPLastroserver .......... 12
Verification of communications .......... 13
SNserver .......... 15

# STARTUP TRACKING SYSTEM

This is an antenna tracking system that integrates dedicated hardware and software designed for amateur use in Earth-Moon-Earth (EME) communications, radio astronomy, amateur Deep Sky Network (DSN) and other Space Communications applications where an accurate and high precision tracking is needed.

Proper use of the system requires knowledge of electronics and some software skills. Therefore, it is recommended for users with previous experience in antenna tracking systems only.

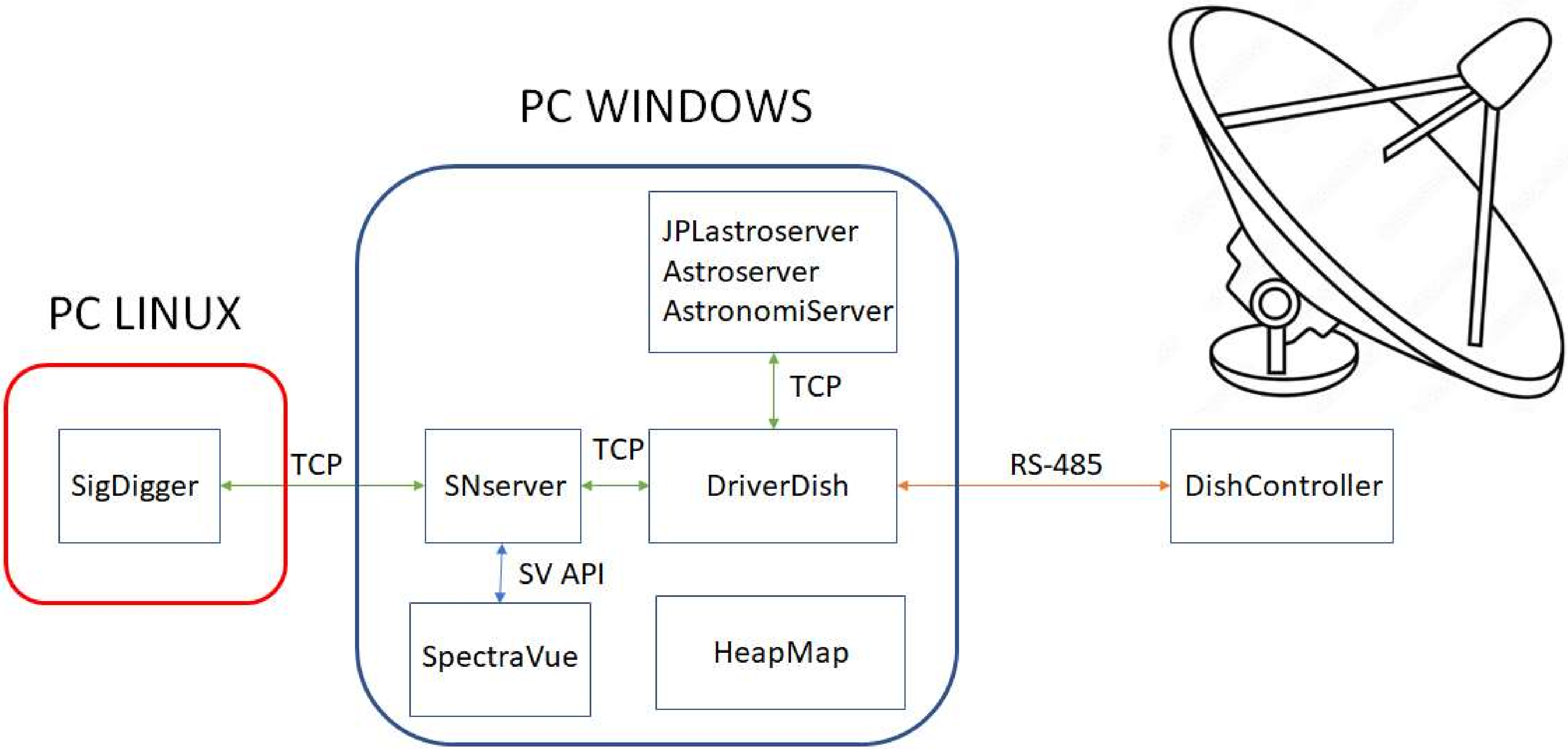
The hardware components of this system have been specifically selected to support the software applications that run on it. These components include a DishController unit that uses an ESP32 microcontroller (MCU), a DC motor driver board with high-performance, high-current control devices, RS485 input/output devices, and a DC step-up converter (versions <1.5). Other peripherals can be connected, such as a variety of digital absolute encoders and inclinometers to sense the motion and position of the motorized turning system, and a remote control unit.

The application software includes the DriverDish program, responsible for requesting data from the ephemeris server and for communicating with the DishController unit, and a suite of server programs (JPLastroserver, Astroserver, or Astronomyserver) that obtain the most accurate positioning data possible, are available from reliable sources.

The integration of dedicated hardware and software in such a system offers several advantages. First, it improves system performance and reliability by allowing the DishController unit to directly access and control specialized hardware components such as motor drivers and position sensors. This reduces overhead and improves response times. Second, it simplifies system design and development by providing a standardized set of hardware and software components optimized for narrow-beam antenna tracking. Third, unlike other antenna tracking system designs, motion and position sensors are located at the last link in the motion chain to actively monitor the aim of the antenna. In this way, the system is able to minimize the impact of mechanical tolerances and gear backlash.

The system is made up of three blocks that work together:

- Ephemeris servers (Astroserver, JPLastroserver or Astronomyserver)
- Control software, SO Windows (DriverDish.App)
- Antenna Controller (Controller Dish)




# EA3HMJ Anntena Tracking System

# Encoders

The first thing we must do is configure the encoders, both SOLAR-360 elevation and SE58xx azimuth.

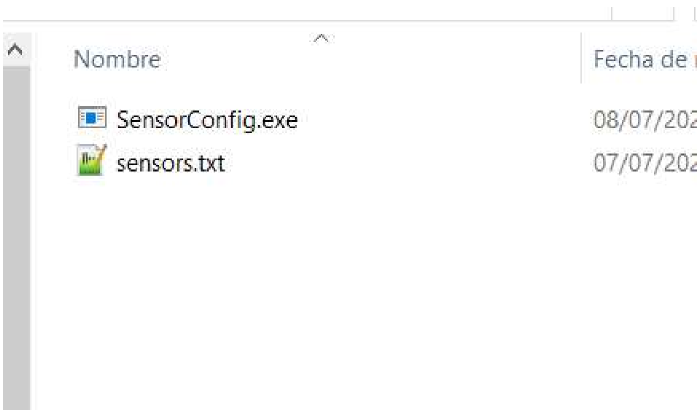
To simplify the process I have created a utility that does the job automatically, we just have to connect the sensor to the computer through the USB-RS485 adapter and power it.

We will use two programs, one to configure and the other to verify.

To configure: https://ea3hmj.net/download/SensorConfig.rar

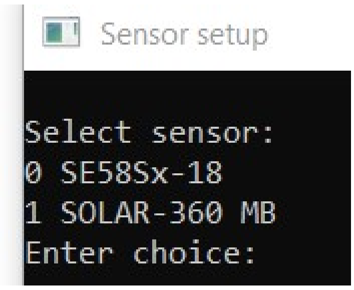
To verify: https://ea3hmj.net/download/ModRTUClient.rar

Unzip the file to a directory and run SensorConfig.exe

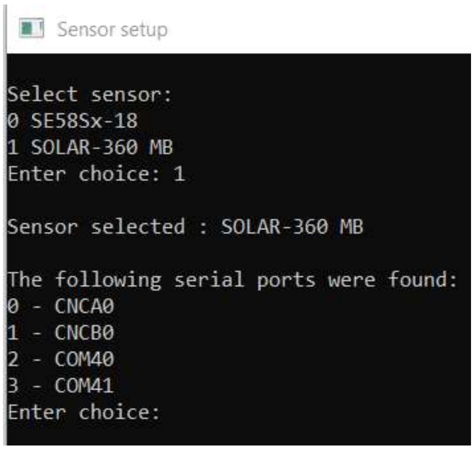



# Elevation SOLAR-360 MB

We connect the encoder and feed it.




Option 1 ask wich port it is connected



We select the port and data will begin to appear on the screen.

```
Select sensor:
0 SE58Sx-18
1 SOLAR-360 MB
Enter choice: 1

Sensor selected : SOLAR-360 MB

The following serial ports were found:
0 - CNCA0
1 - CNCB0
2 - COM40
3 - COM41
Enter choice: 2
Select port : COM40
Verifying correct parameters
Tx =>  02 03 00 04 00 02 85 F9
Rx =<  00 00 00 00 00 00
Check 19200 bps
Tx =>  02 03 00 04 00 02 85 F9
Rx =<  00 00 00 00 00 00
Tx =>  01 03 00 04 00 02 85 CA
Rx =<  00 00 00 00 00 00
Tx =>  02 03 00 04 00 02 85 F9
Rx =<  00 00 00 00 00 00
```

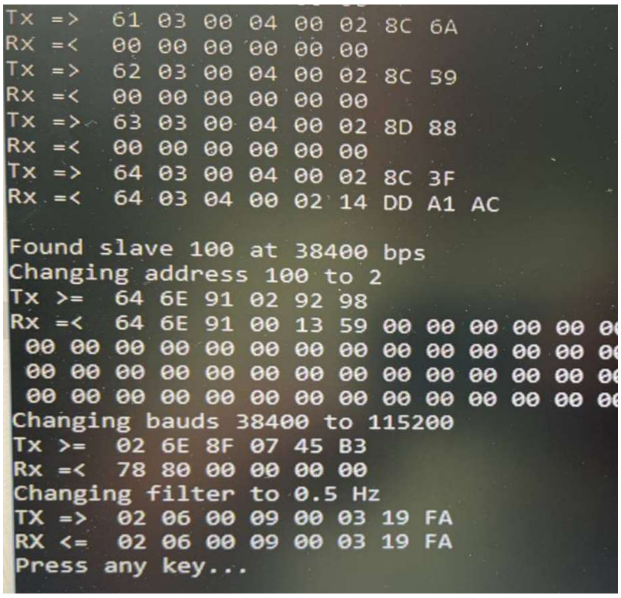
It ends with the following screen

```
Sensor setup
Tx =>  7B 03 00 04 00 02 8E 50
Rx =<  00 00 00 00 00 00
Tx =>  7C 03 00 04 00 02 8F E7
Rx =<  00 00 00 00 00 00
Tx =>  7D 03 00 04 00 02 8E 36
Rx =<  00 00 00 00 00 00
Tx =>  7E 03 00 04 00 02 8E 05
Rx =<  00 00 00 00 00 00
Tx =>  7F 03 00 04 00 02 8F D4
Rx =<  00 00 00 00 00 00

Press any key...
```

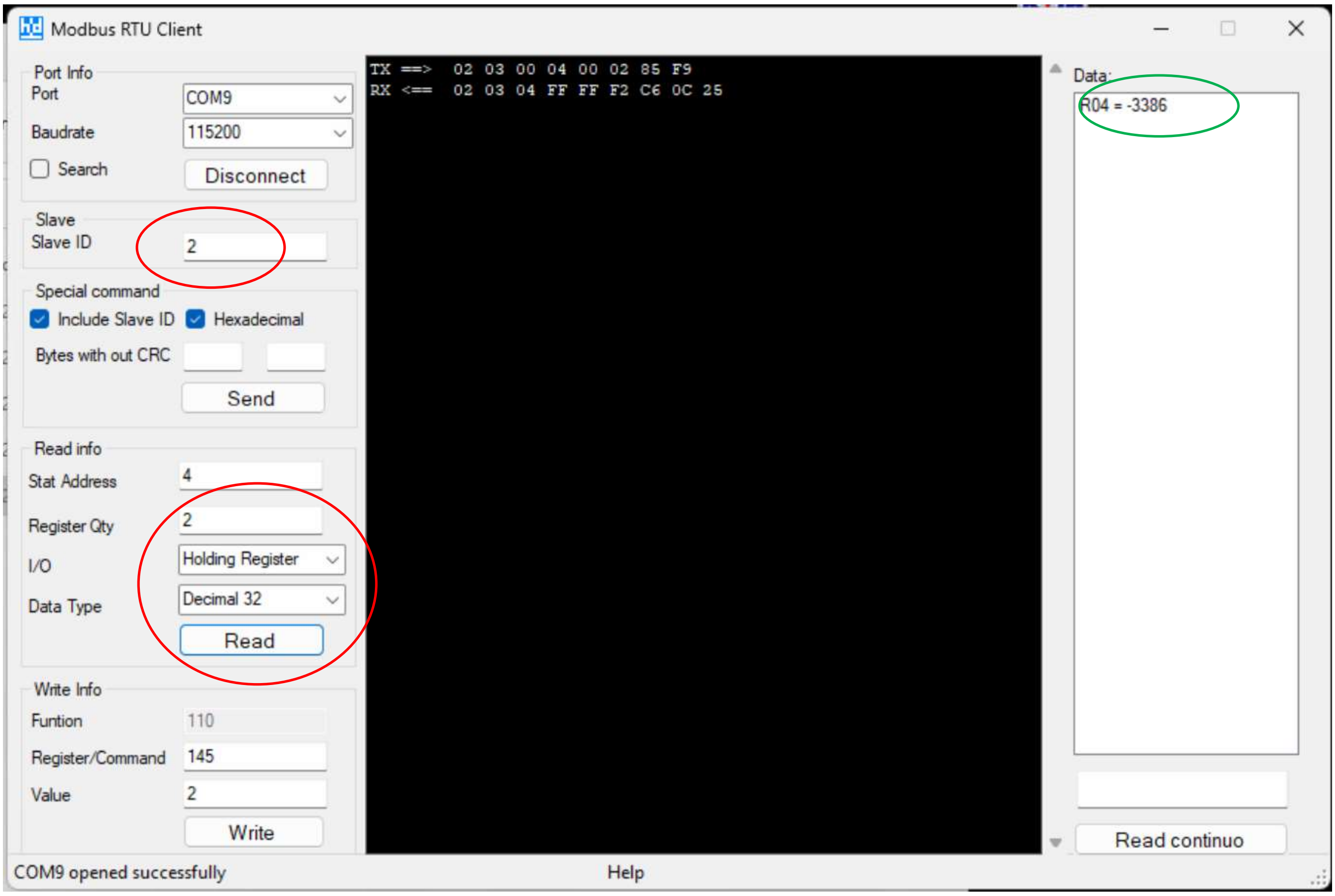
If the system is unable to find the encoder, change the order of cables A and B, maybe they are reversely connected.

When everything has gone correctly, it presents us with the following screen.




## Verification

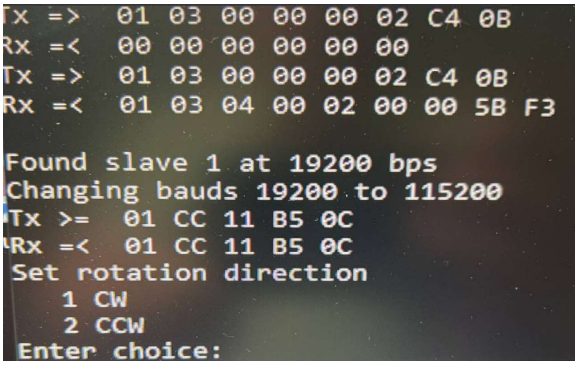
We open the Modbus RTU Client program, select the port that we have connected to the inclinometer and speed 115200 and Slave ID 2. We press Connect.



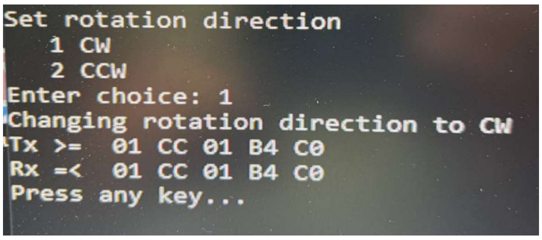

We configure the parameters that are shown on screen and press the Read button at the top, it indicates the value of the reading.

## Azimuth SE58Sx

We will use the same procedure and, if everything is correct, it will ask us the direction of rotation.



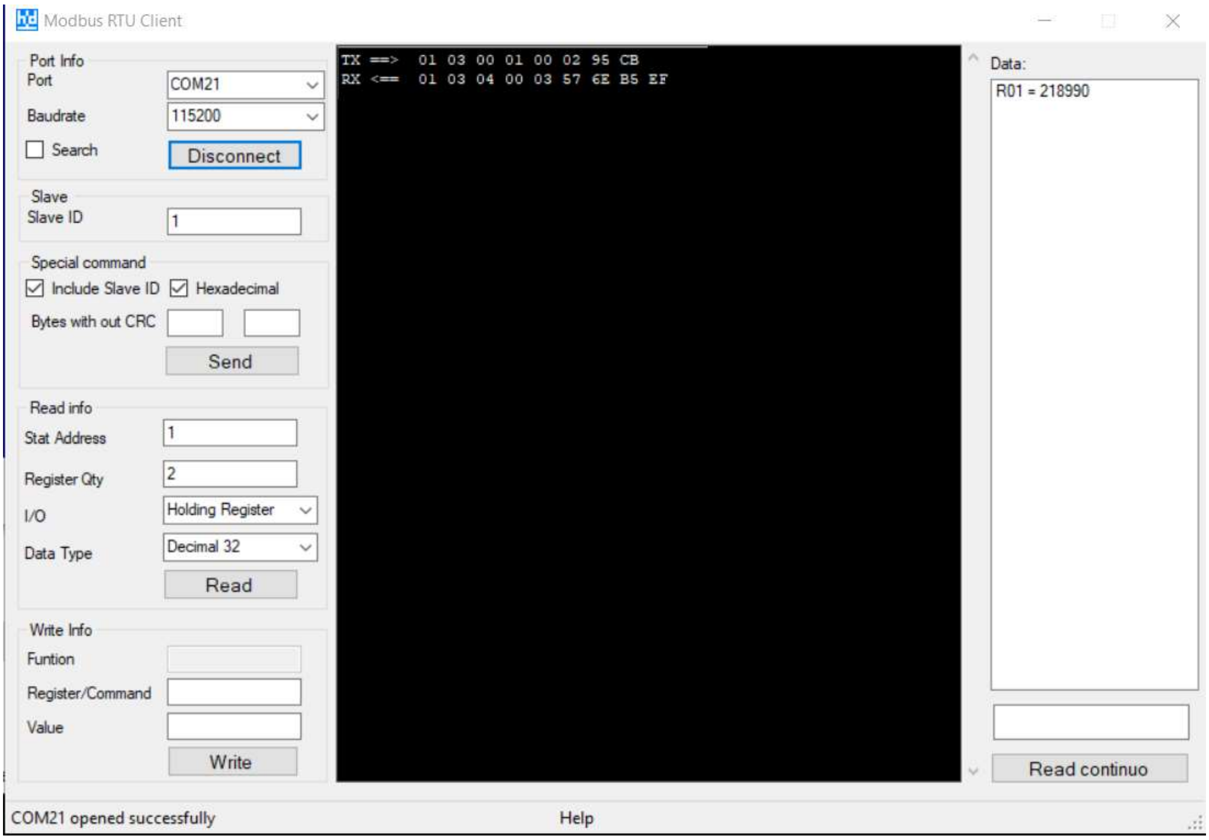

Select address




The encoder is already configured.

## Verification

We open the Modbus RTU Client program, select the port that we have connected to the encoder and speed 115200 and Slave ID 1. We press Connect.

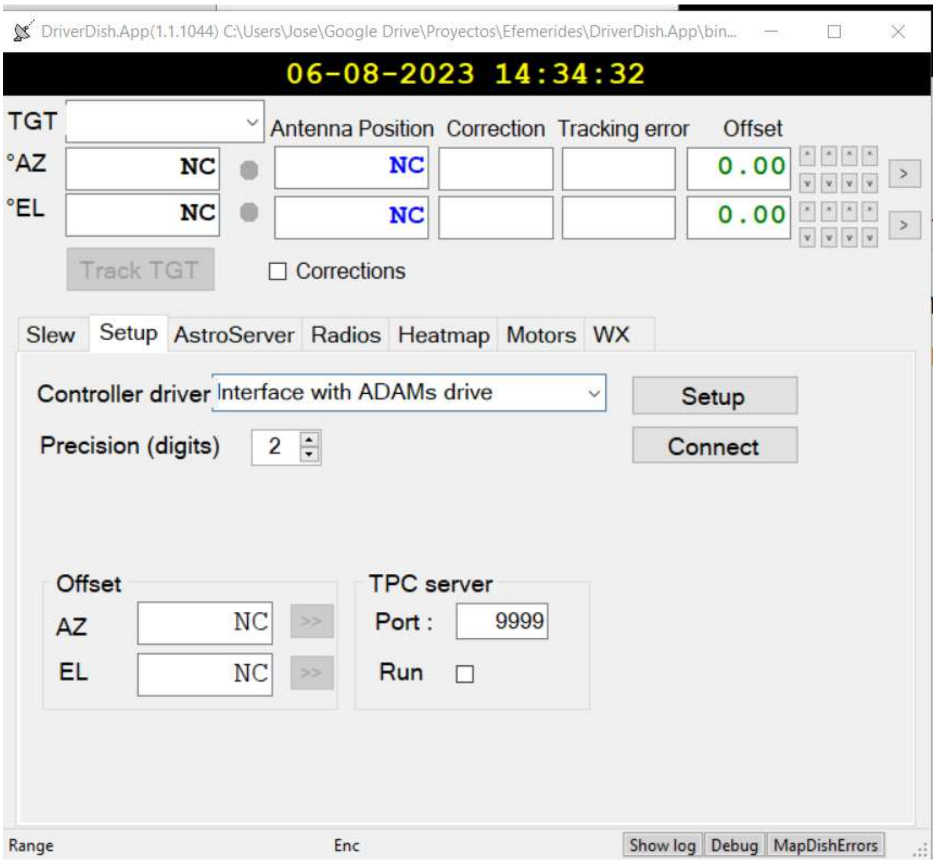



We configure the parameters, the same that are shown on screen, and press the Read button at the top, it indicates the value of the reading.

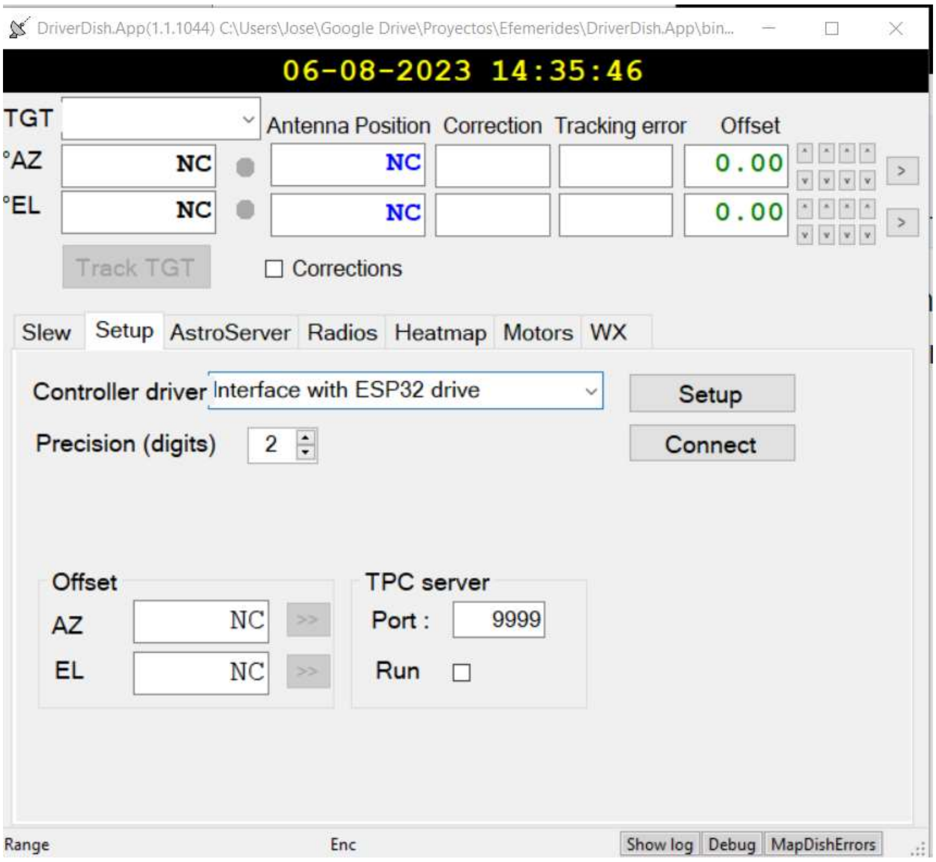
# DriverDish

We connect the encoders to the controller and the controller to the PC and restart the program.

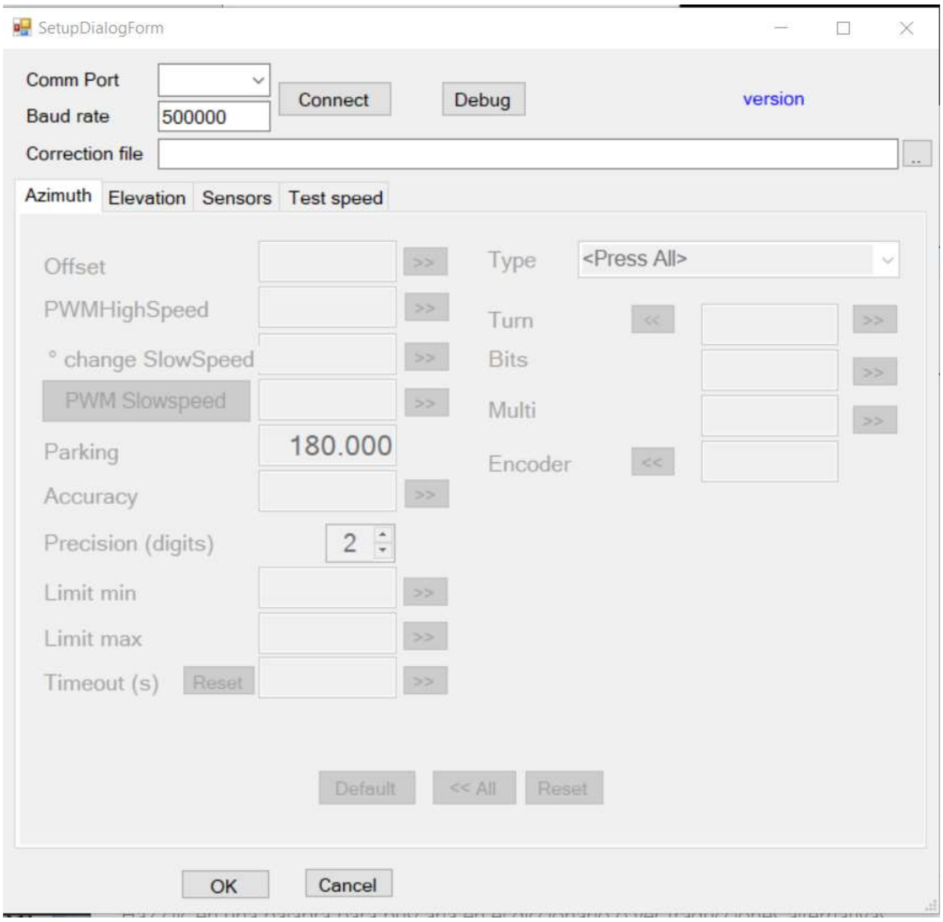
We select the setup tab.



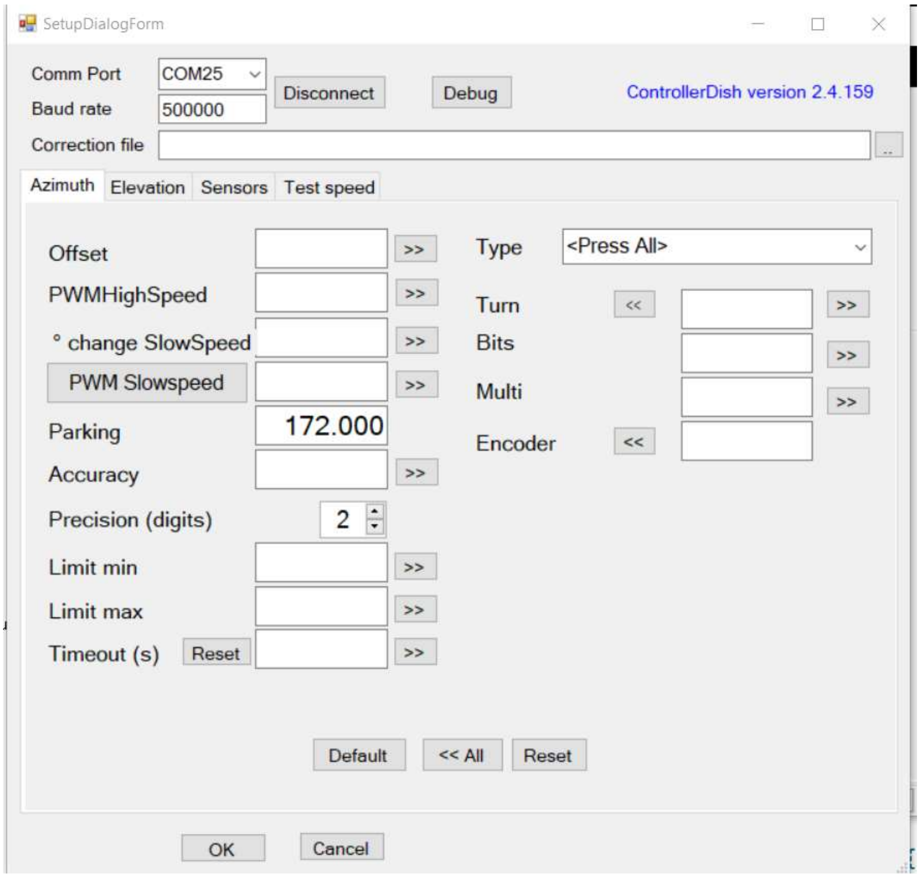

In controller driver we select ***Interface with ESP32 drive.***



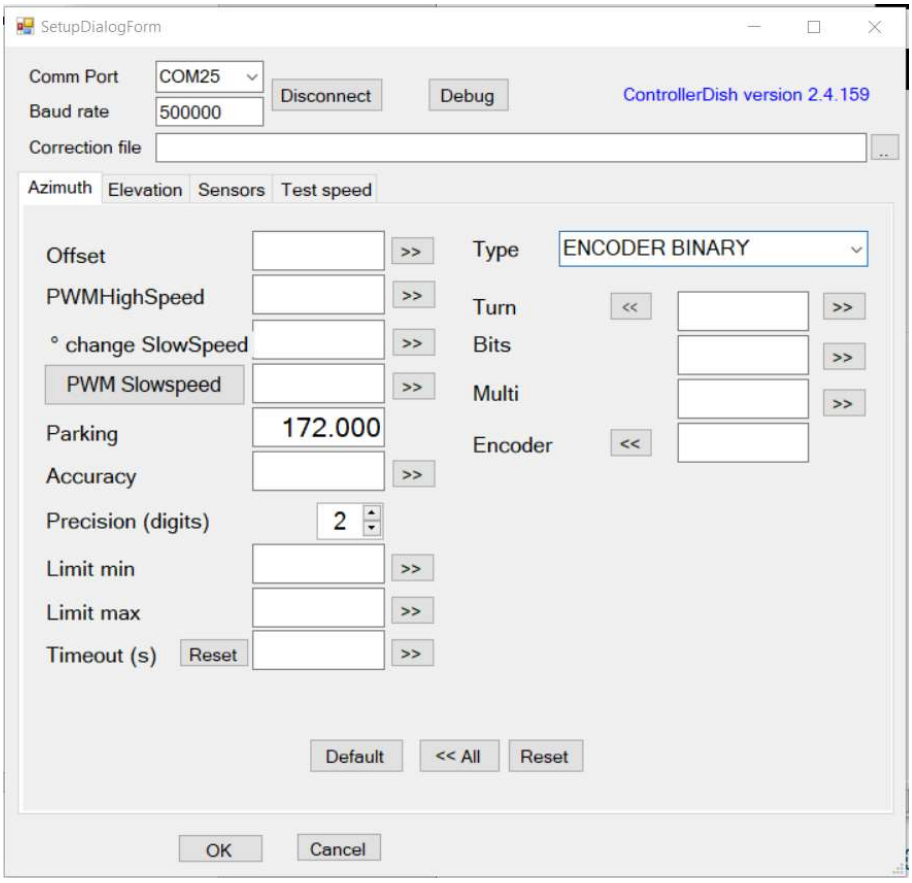
Press ***setup***.



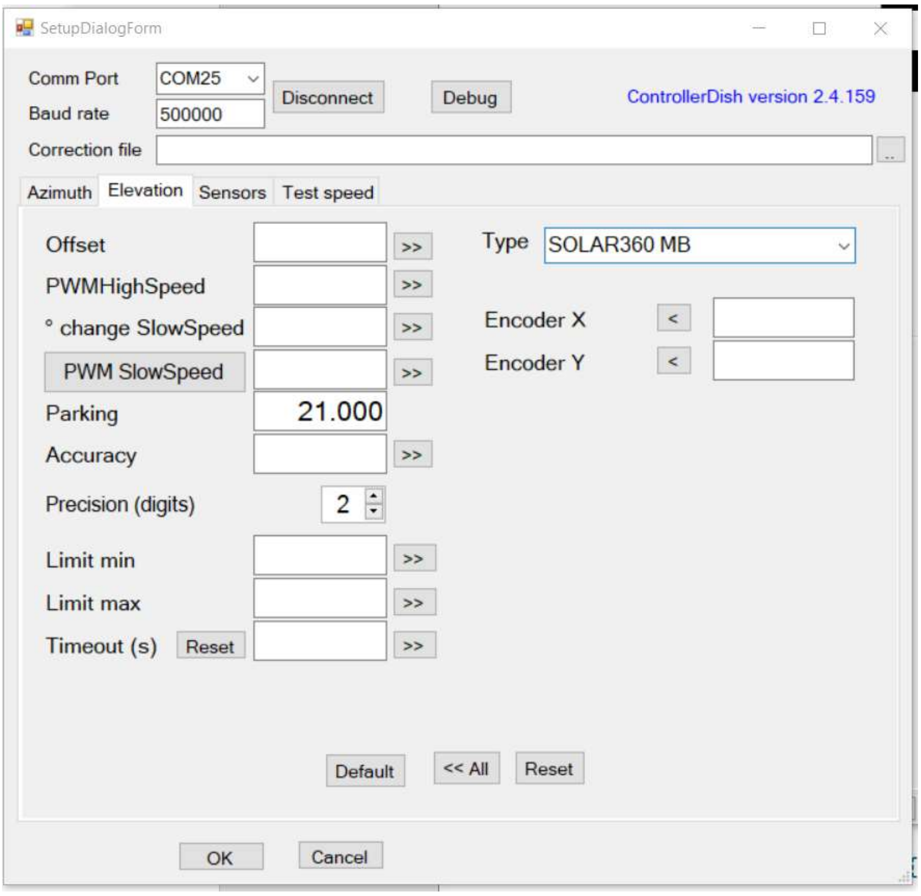

We select the COMM Port and press connect, until the controls that are now gray are enabled.



In the azimuth tab we select type: **ENCODER BINARY**.

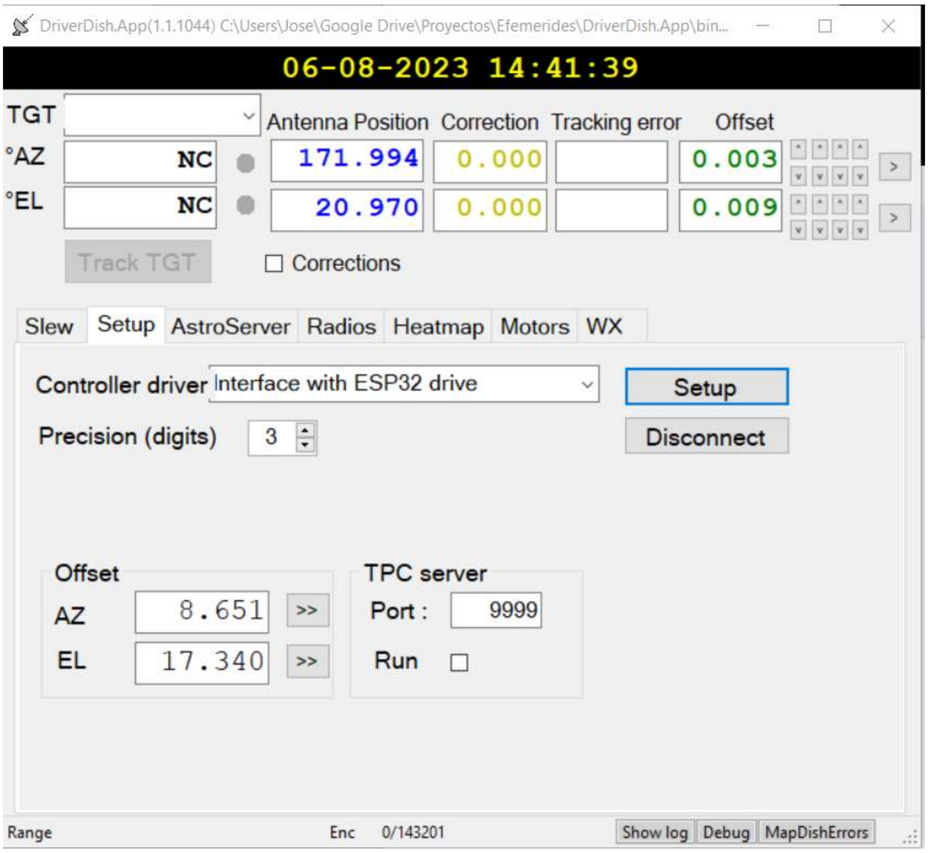



In the elevation tab we select type: **SOLAR360MB**

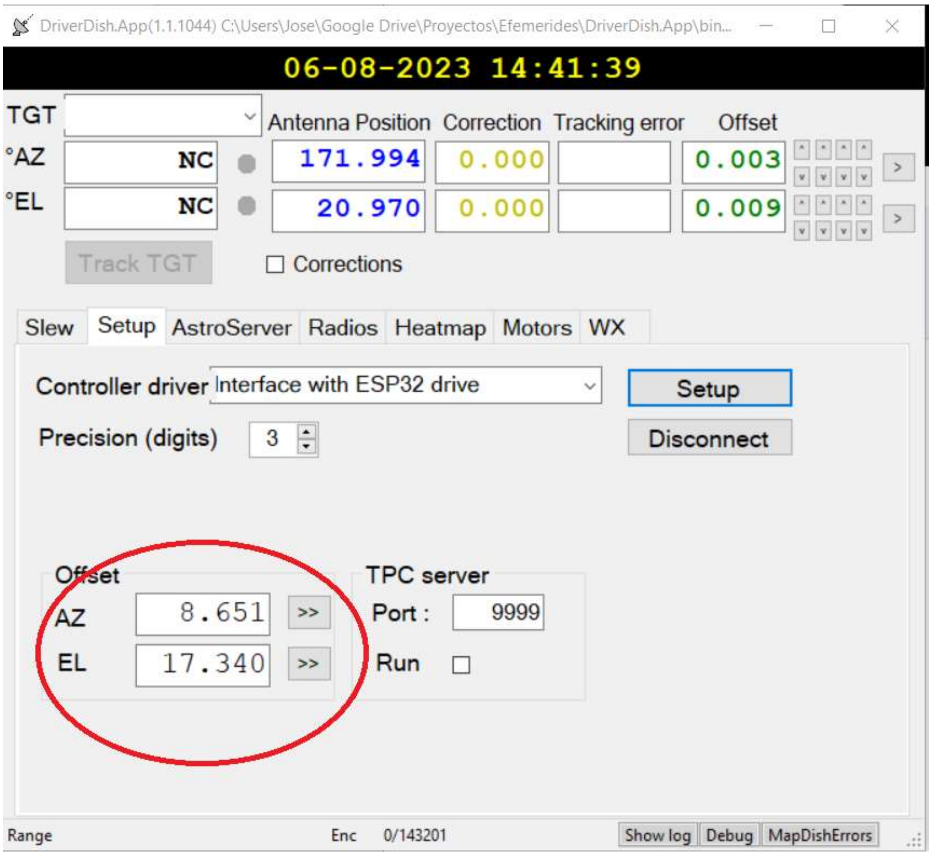



Click OK to exit to the main screen.

We can now verify that we have already information about antenna position.



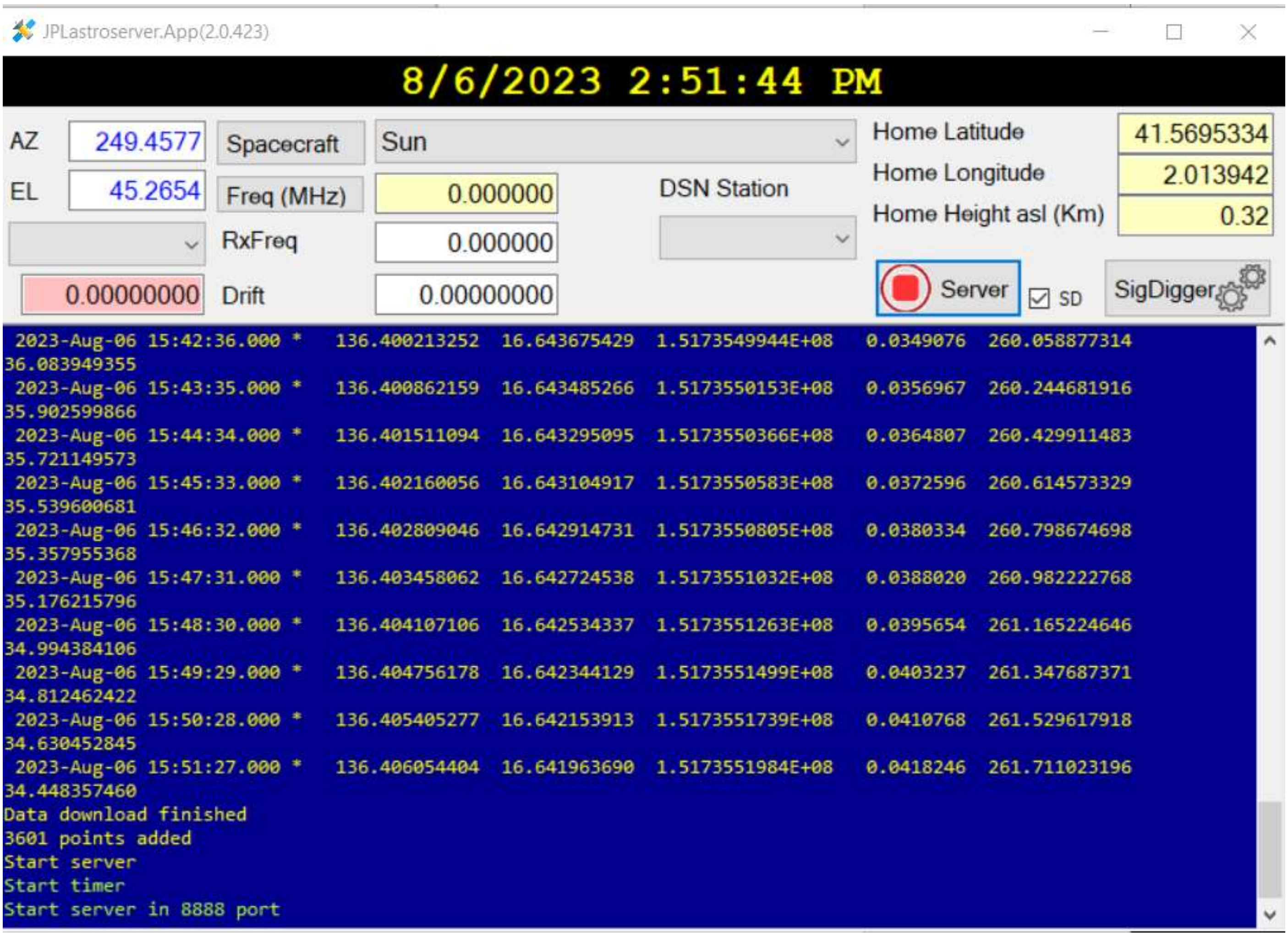

With the offset values we can adjust the value of the antenna position as close to reality.




We must start from a known position, example 180° in AZ and if we have a 21° elevation offset, that is when the antenna is perpendicular to the ground. The button on the right must be pressed to send the new data to the controller. We close the program.

## JPLastroserver

We start the program

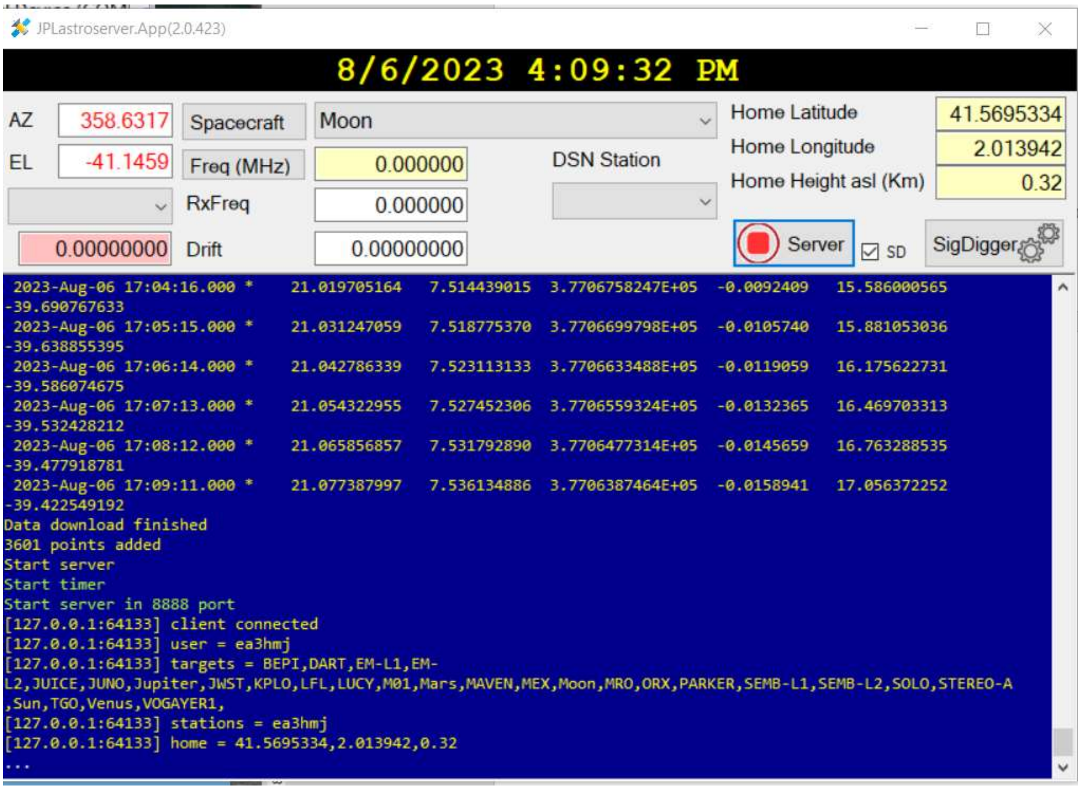



We must put the appropriate values of latitude, longitude and height of our location.

We close the program.

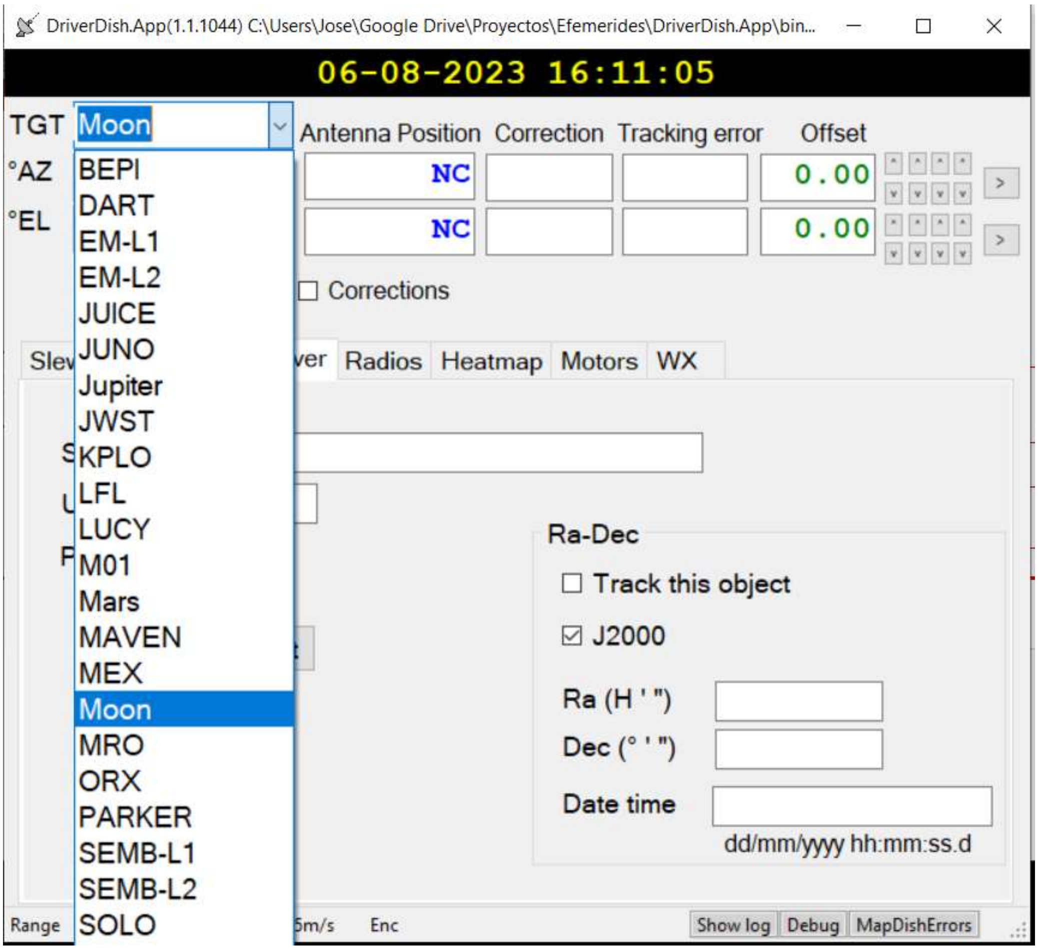
# Verification of communications

We open JPLastroserver first and then open DriverDish.

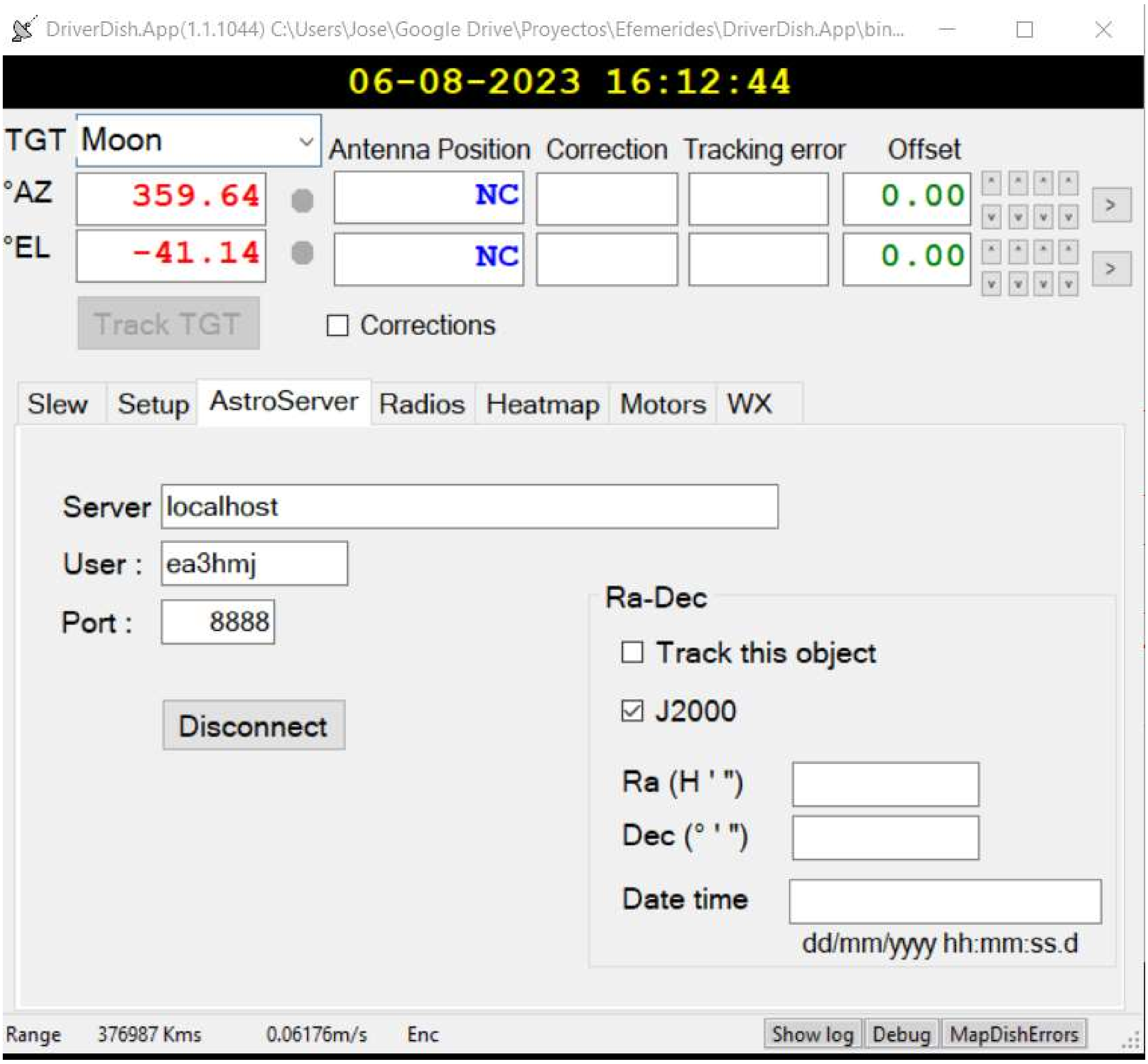



In the last lines of the log we can verify that driverdish has been connected.

We select an object in DriverDish.



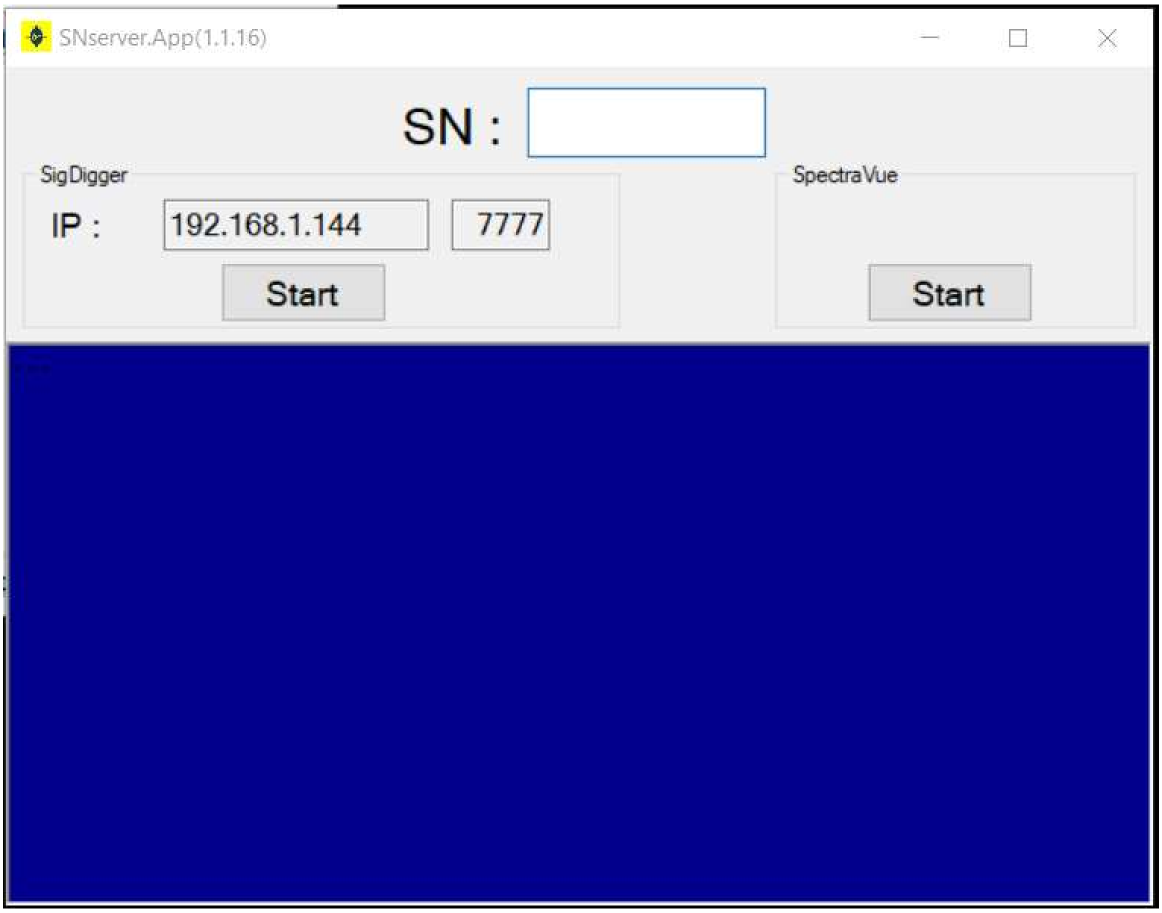
The program already obtained ephemeris from the server and we can track the object selected.



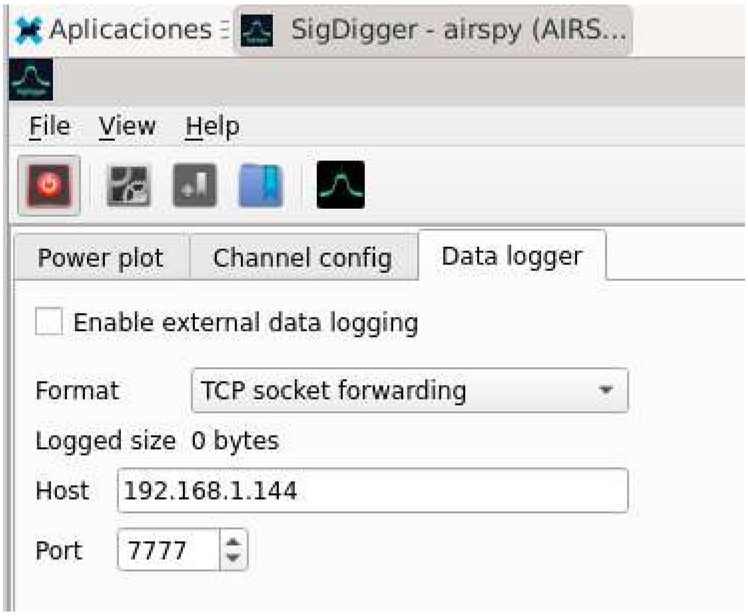
# SNserver

We can connect DishDriver with SpectraVue or SigDigger through SNserver.

This program acts as an interface between the information it receives from these programs and DriverDish.

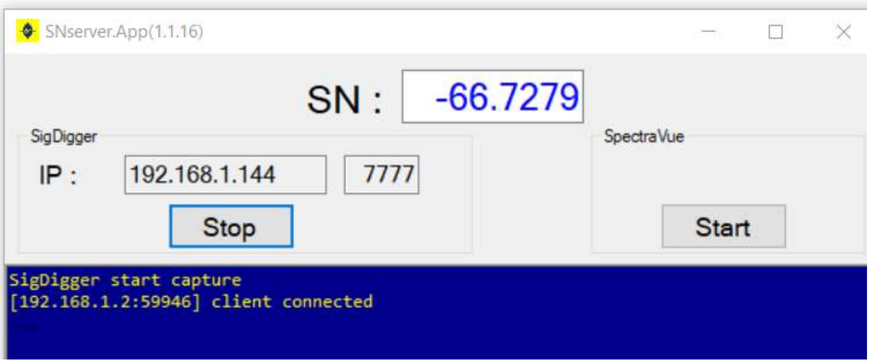



This screen gives us the information to configure SigDigger.

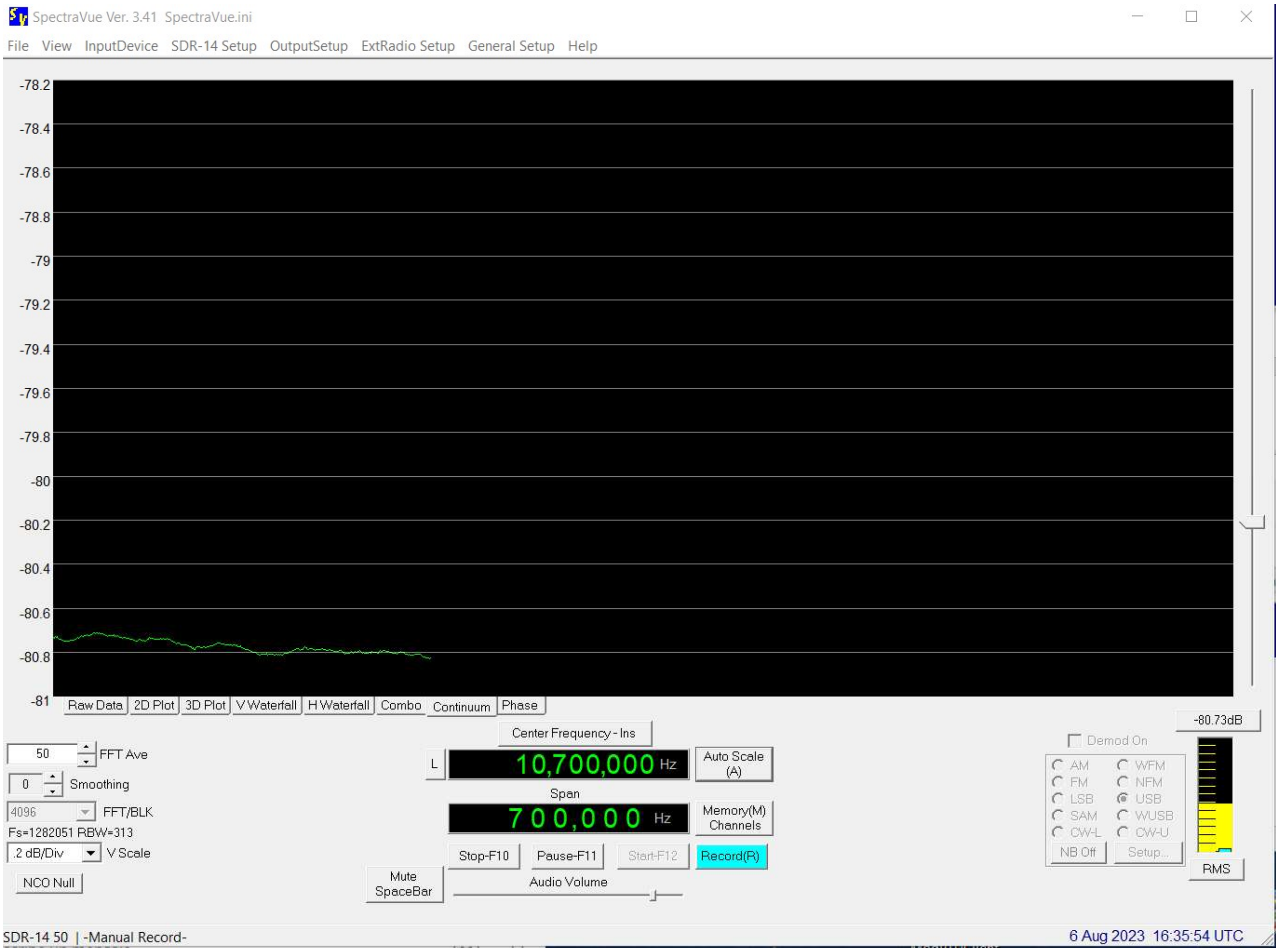



This information must be configured at SigDigger to receive information.

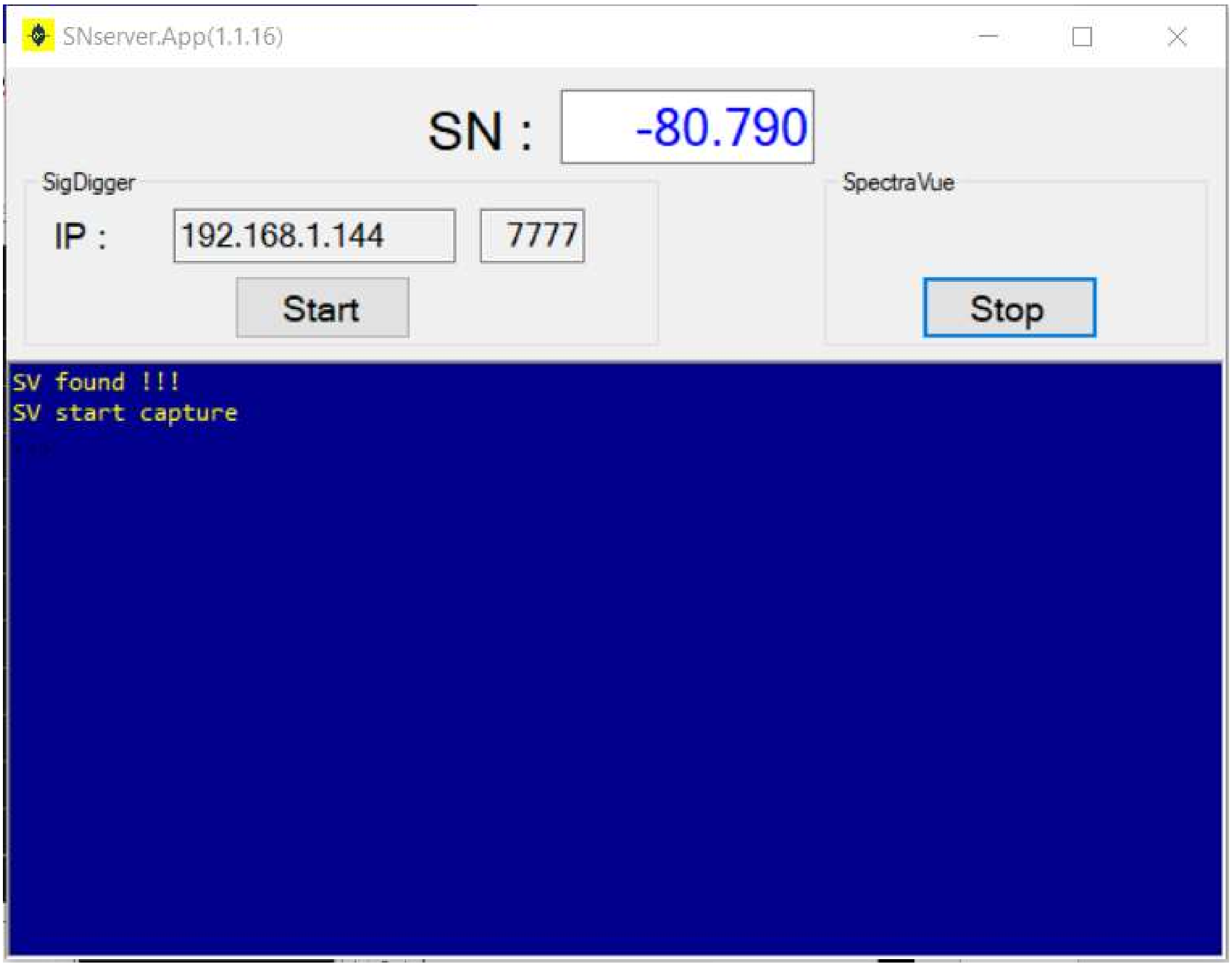
Press Start in SNserver at the SigDigger side.



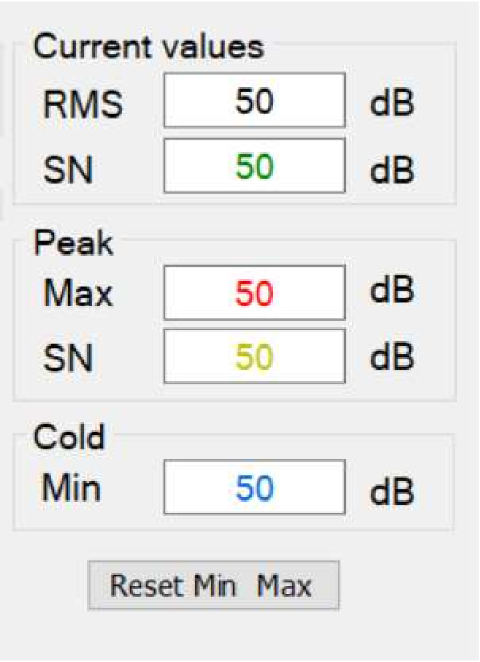

But if we want to connect through SpectraVue it is easier, we must have the program started.

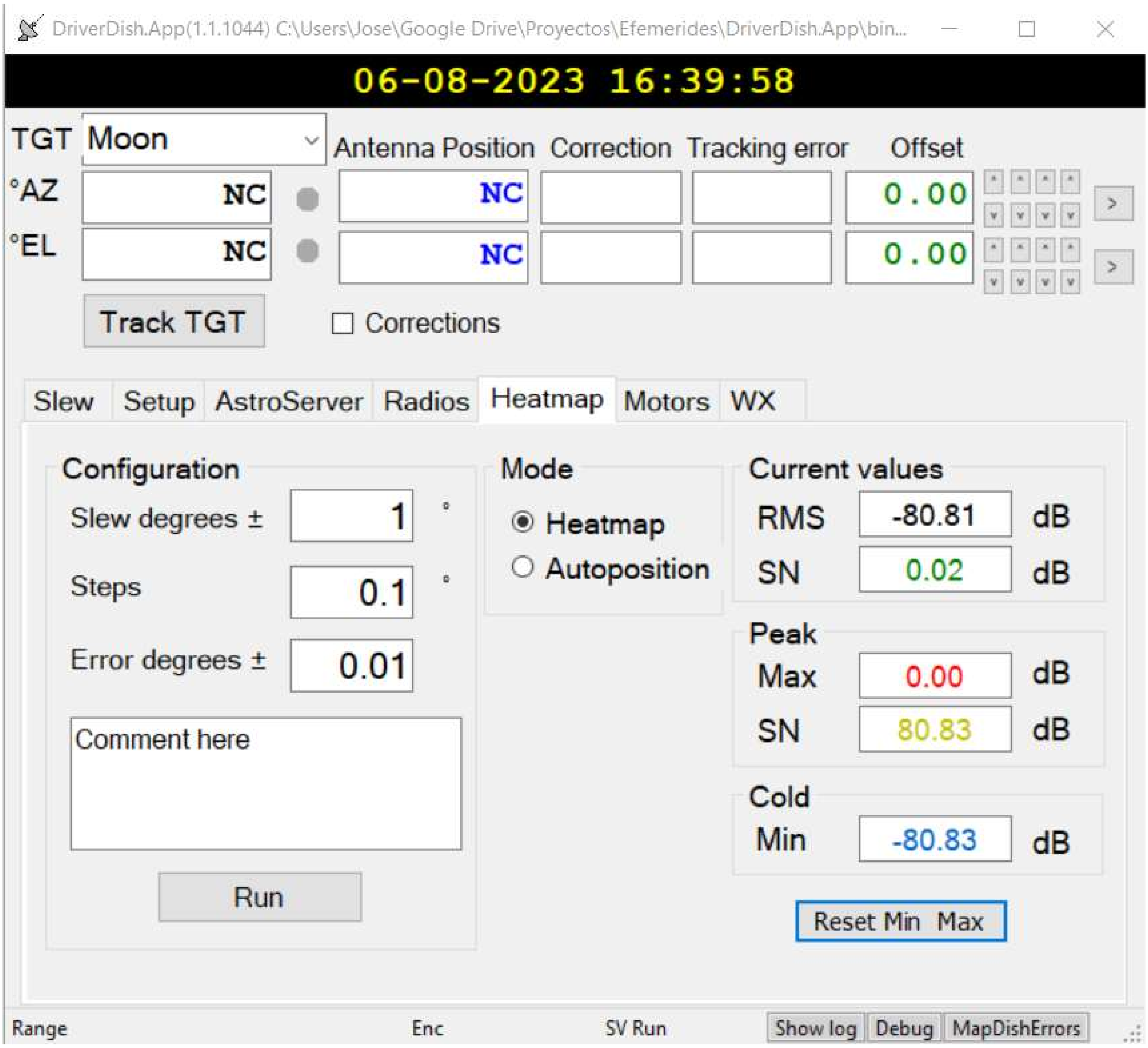


Press Start in SNserver at the SpectraVue side.




If we have started DriverDIsh before and SNserver did not connect, we must do it manually by pressing the **Reset Min Max** button.






We already have the information.

If we first start SNserver when starting DriverDish it automatically connects.