Callisto DiSEqC Antenna Tracker

Fig. 1: Final product backside with connectors and front side with LED and switch

Description

The goal of the project was to make an antenna tracker that could not only follow the Sun, but also other objects in the sky (Moon, Satellite, etc.). Our main priority was to keep the price as low as possible to ensure everyone could build the same thing. The key components (and most expensive ones) are two DiSEqC Rotors and an Arduino Micro. The first DiSEqC is in charge of the x-axis (azimuth or hour angle), while the second one controls the remaining one (elevation or declination). A single Arduino controls the movement of both. For some latitudes near 45° north +/- ~10° it might be sufficient to track the Sun with a single SAT-rotor parallel to the path of the geostationary satellites, assuming the beam angle of the antenna covers at least 20°. In such a special case, the SAT-rotor can be used like in a satellite application. For example, see solution as in Glasgow/UK here: https://e-callisto.org/coverage/AntennaFinalPos.JPG

Content

Technical Data

Attributes	Value
Diseqc Motor	TechniSat, HH100, HH120 and similar ones
Vin	13-18V
Code	Arduino and Python 2.7 and above
Signal from Arduino	22 KHz
Diseqc return	None
Angle	Depending on rotor: +/- (62°…78°)
Angle resolution	1/16°
Current when idle	30-50mA
Current when moving	200-350mA
Prize tag	~270 CHF +/- currency fluctuations

Table 1: Technical specifications

Start the Setup

1. Connect the F-Cables to the DiSEqC motors
2. Plug in the Arduino via USB-cable
3. Connect the Supply voltage and turn it on
4. Start the Python script and let it do the work

The connections are straightforward. There is a micro USB cable for the communication between the Arduino and the PC (Python Script). The F-Cables are for the communication between the Arduino and the motors. Any voltage between 13 and 18 volts works just fine. The higher the voltage the faster the drive velocity.

Nr.	Cable	Function
1	Micro USB	Arduino
2	F-Cable	DiSEqC Elevation
3	F-Cable	DiSEqC Azimuth
4	Power supply cable	Supply 13 V ... 18 V / 0.5 A

Table 2: Cable connectors.

Schematic

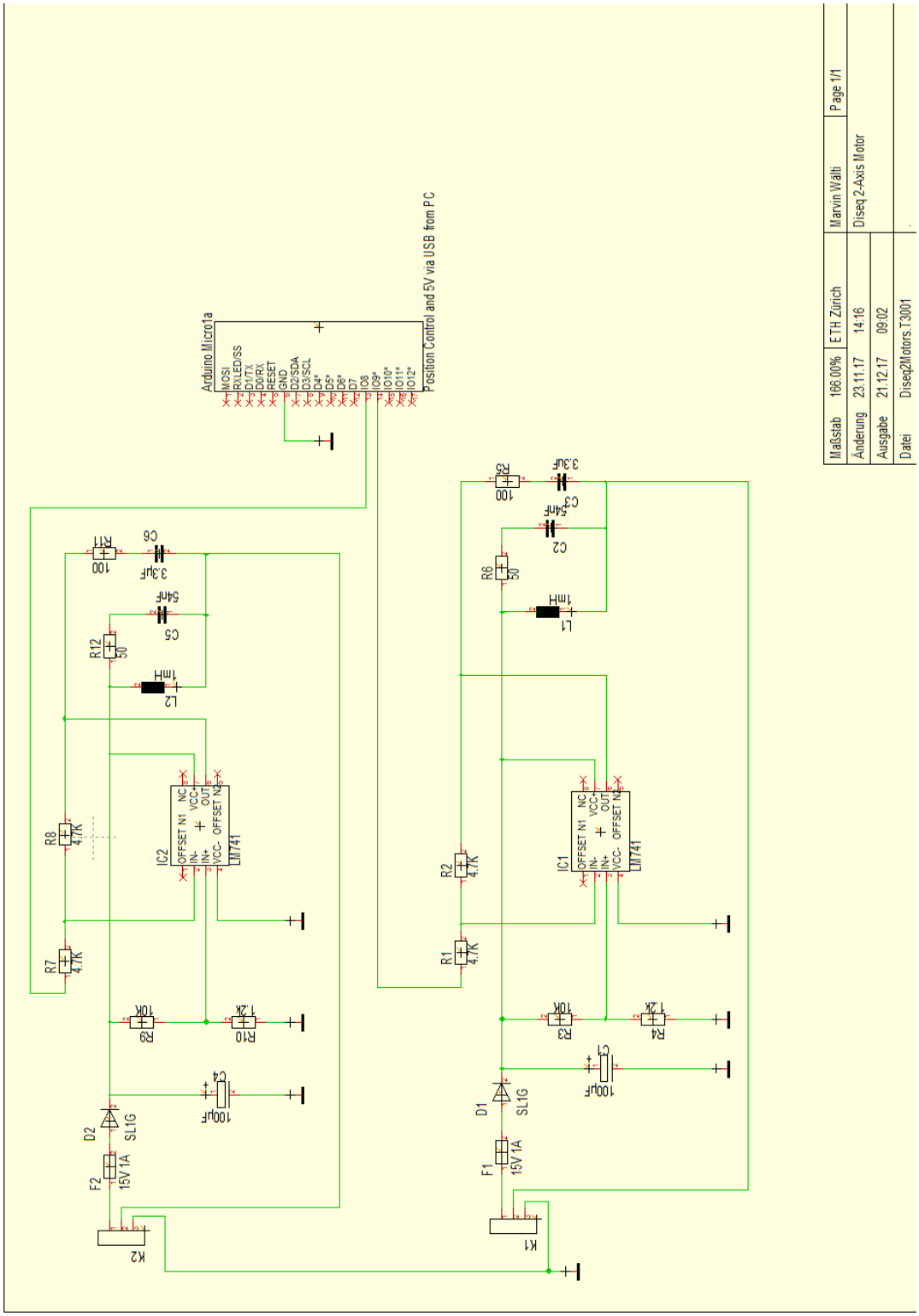
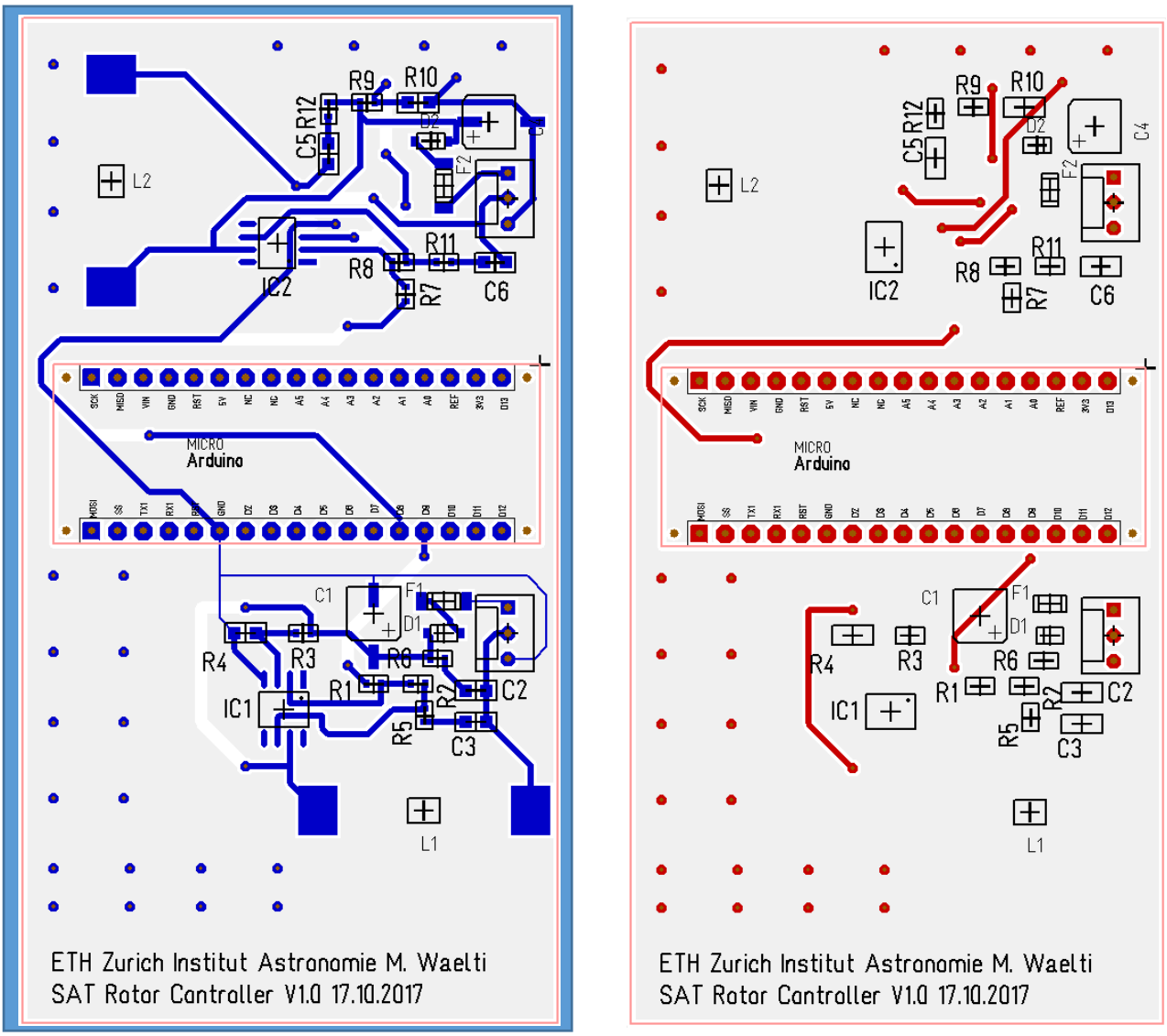

Arduino Micro1a
Position Control and 5V via USB from PC
Maßstab 166.00% ETH Zürich Marvin Wälti Page 1/1
Änderung 23.11.17 14:16 Diseq 2-Axis Motor
Ausgabe 21.12.17 09:02
Datei Diseq2Motors.T3001
IC1
IC2
LM741
K1
K2
F1
F2
D1
D2
SL1G
15V 1A
L1
L2
1mH
C1
C2
C3
C4
C5
C6
100µF
3.3µF
5nF
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
4.7K
10K
2K
100
50

Fig. 2: Schematics

Layout



Figures 3a, b: Top- and bottom layout of PCB

Components

Amt.	Nr.	Part	Value	Distrelec Nr.
4	R1/2/7/8	Resistor	4.7kΩ	
2	R3/R9	Resistor	10kΩ	
2	R4/10	Resistor	1.2kΩ	
2	R5/11	Resistor	100Ω	
2	R6/12	Resistor	50Ω	
2	F1/2	Fuse	15V 1A	300-79-840
2	D1/2	Diode	SL1G	170-02-190
2	C1/4	Capacitor	100uF	300-47-883
2	C2/5	Capacitor	56nF	
2	C3/6	Capacitor	3.3uF	
2	IC1/2	Operational amplifier	LM741	173-01-406
2	L1/2	Choke	1mH	158-10-652
1	Arduino	Microcontroller	Micro	
2	K1/2	Connector	3 Pin	
2	F1/2	DiSEqC Connector	F	
1	R1	DC-Connector	Black	
1	LED1	LED	Red	
1	S1	Switch	ON/OFF	
2	Mot1/2	DiSEqC Rotors		Technisat or similar
1	PSU	Power supply unit	18V/0.5A	Distrelec 300-41-879 or Conrad 1294213-62
1	PCP	Printed Circuit Board	AU-201806/26572	Beta-Layout
1	BP	Backpanel		Beta-Layout
1	FP	Frontpanel		Beta-Layout
1	Case	Aluminum enclosure		CONRAD 522 945

Table 3: Parts list of PCB, enclosure and power supply

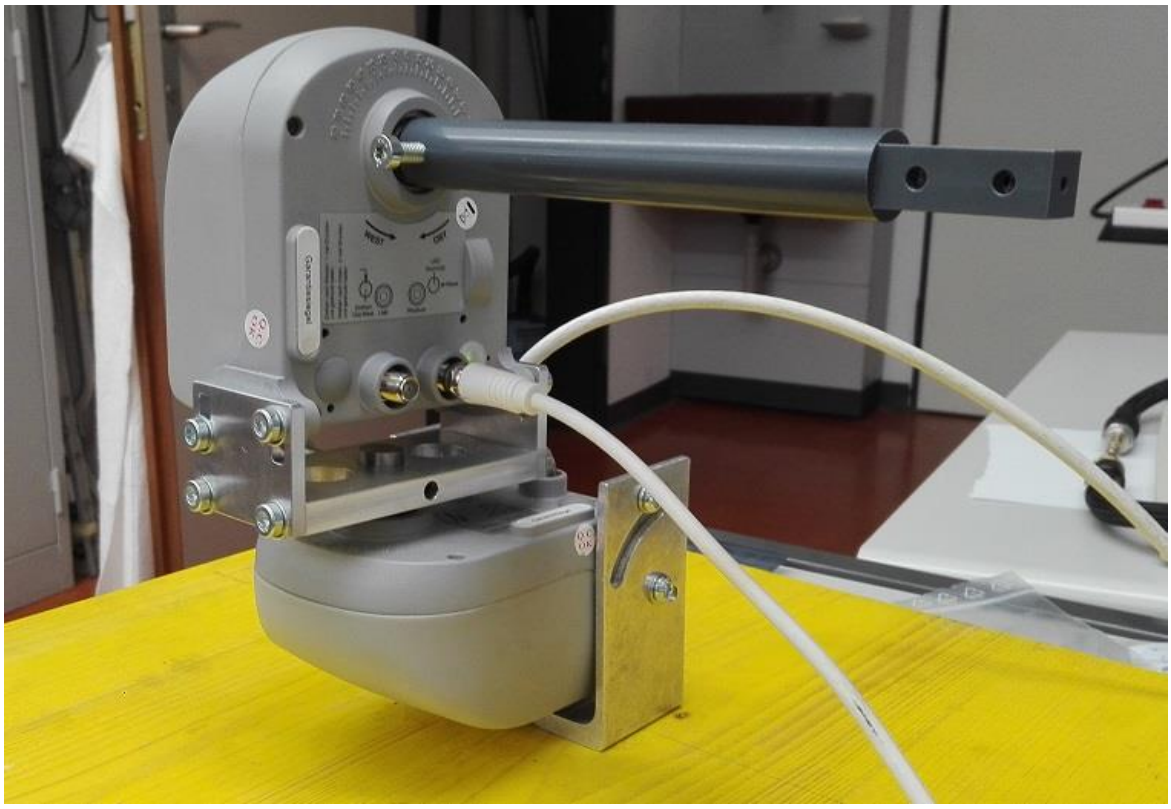


Fig. 4: How to stack 2 SAT-rotors for an Azi/Ele-drive system. Drive shafts where replaced by individual adapters to finally get a two-axis system.

Fig. 5: How to install SAT-rotor for pseudo parallactic mount with tracking in local hour angle only at fixed declination. Depending on mounting method the command for hour angle needs to be changed from ha to -ha.

Low cost installation with one rotor

With manually adjusted, fixed declination we only need one satellite rotor, controlled by hour angle (HA).

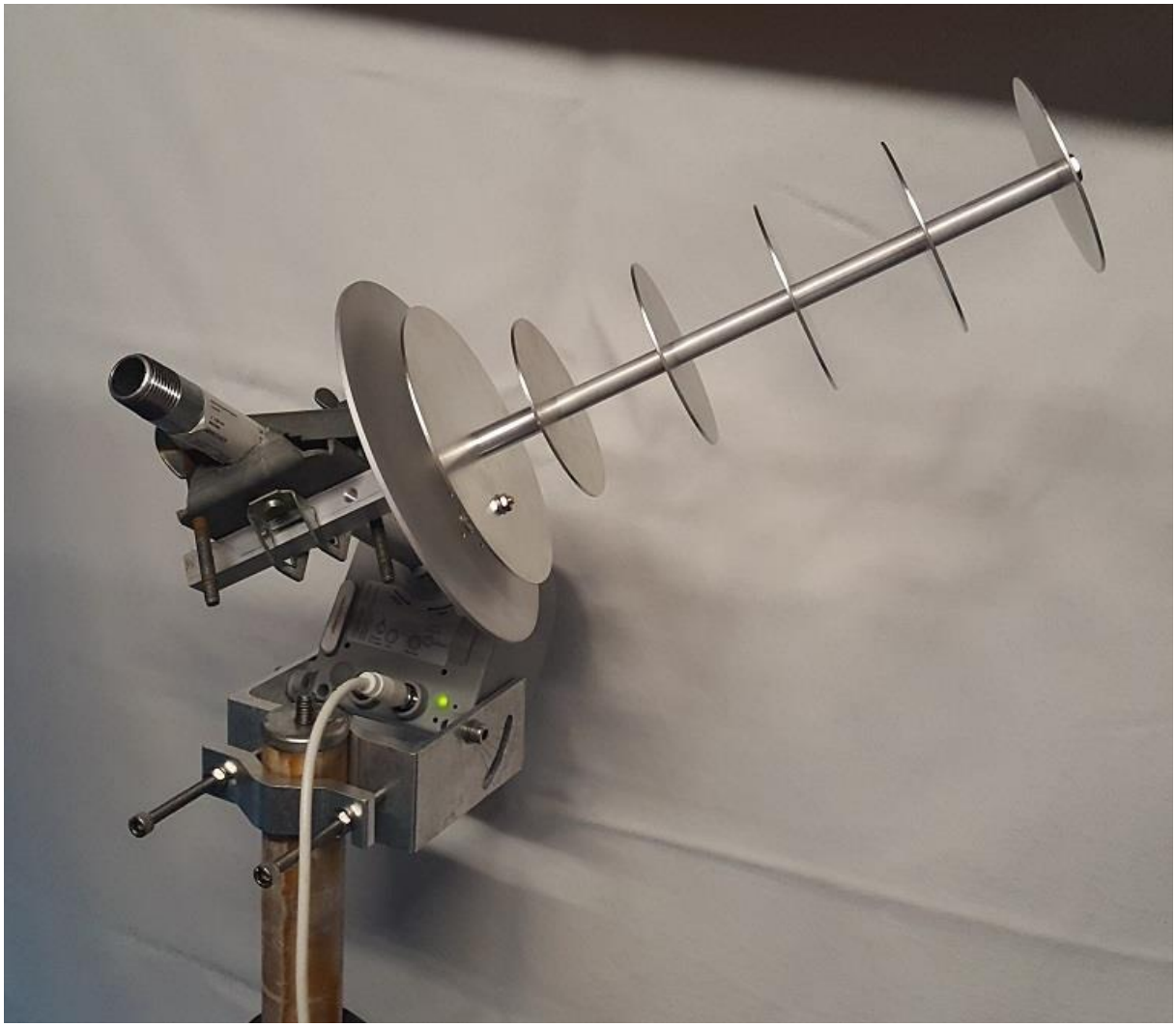


Fig. 6: Simple rotor with a shaft extension by a piece of water pipe 1/2". For demonstration a small L-band antenna has been attached.

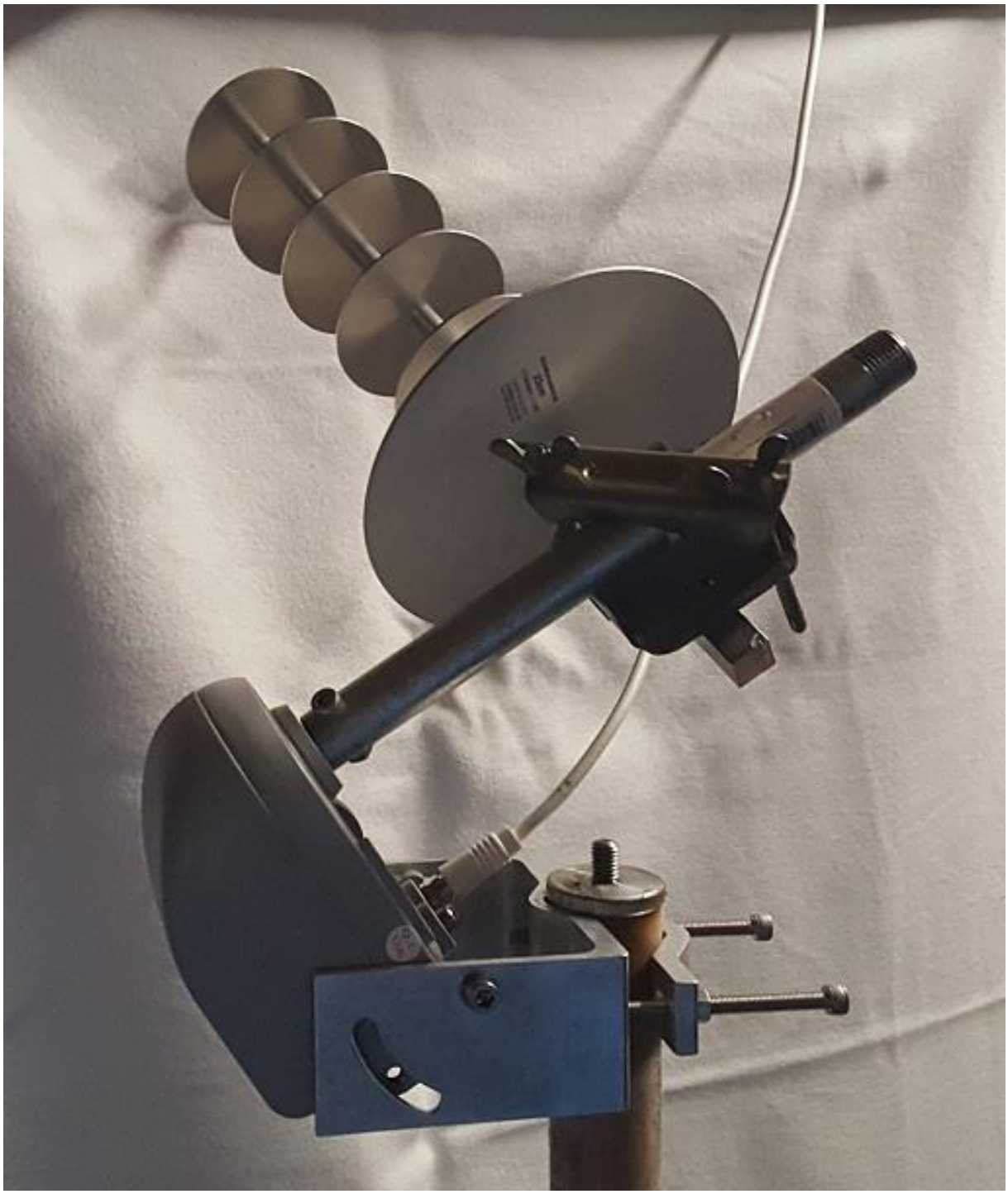


Fig. 7: Side view of an hour angle drive system. Shaft extension by 1/2" water pipe.

S-Band example

Fig. 8: Control PC and rotor controller box. A Python script is called every 5 minutes and commands the rotor to the actual Sun position. The system can also be used to track satellites, e.g. IRIDIUM.

Fig. 9: S-band antenna with Sat-rotor on a tripod tracking the Sun.

Software Installation

Required: Python 2.7 or higher which is tested, no guarantee for other versions
In case you need to modify the Python script, I recommend Anaconda and Spyder
Scripts can be found in chapter 4 here: https://www.e-callisto.org/Hardware/Callisto-Hardware.html

Install following extra Python libraries:
1. pip install serial to update type: pip install --upgrade serial (or Pyserial)
2. pip install ephem to update type: pip install --upgrade ephem

3. Install Arduino IDE or at least the driver for Arduino from here:
https://www.arduino.cc/en/main/software

4. Install system scheduler from here:
https://www.filecluster.com/downloads/System-Scheduler.html

5. Edit parameter MyLocation according to your latitude, longitude, altitude, pressure and temperature. For MaxRange check the specification of your SAT-rotor and for communication port check Windows device manager.

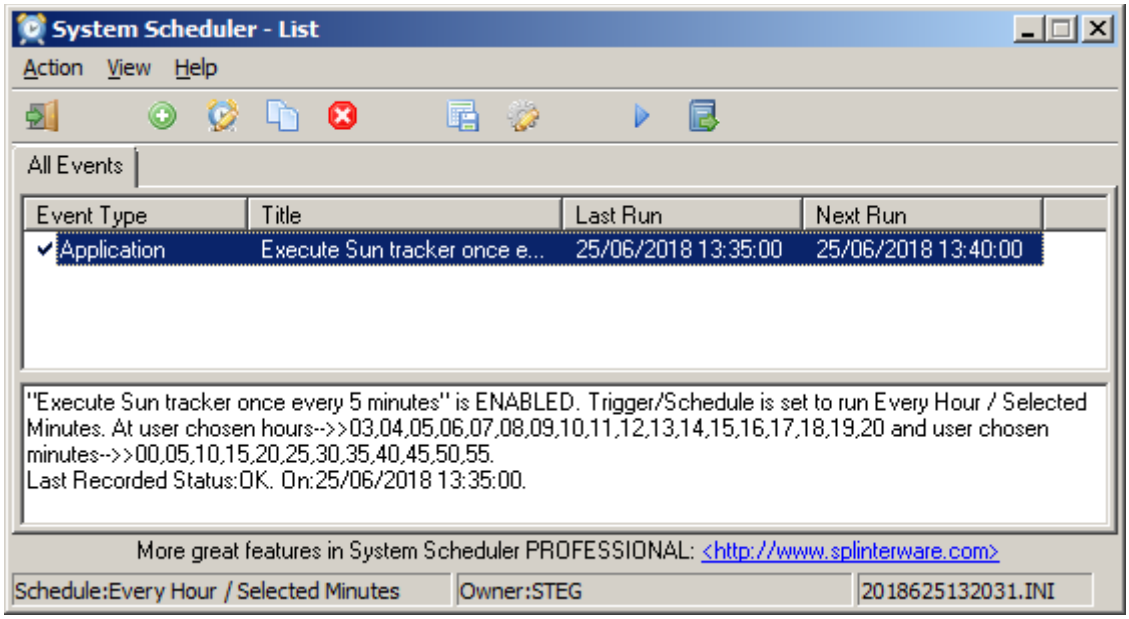


Fig. 10: Screenshot system scheduler

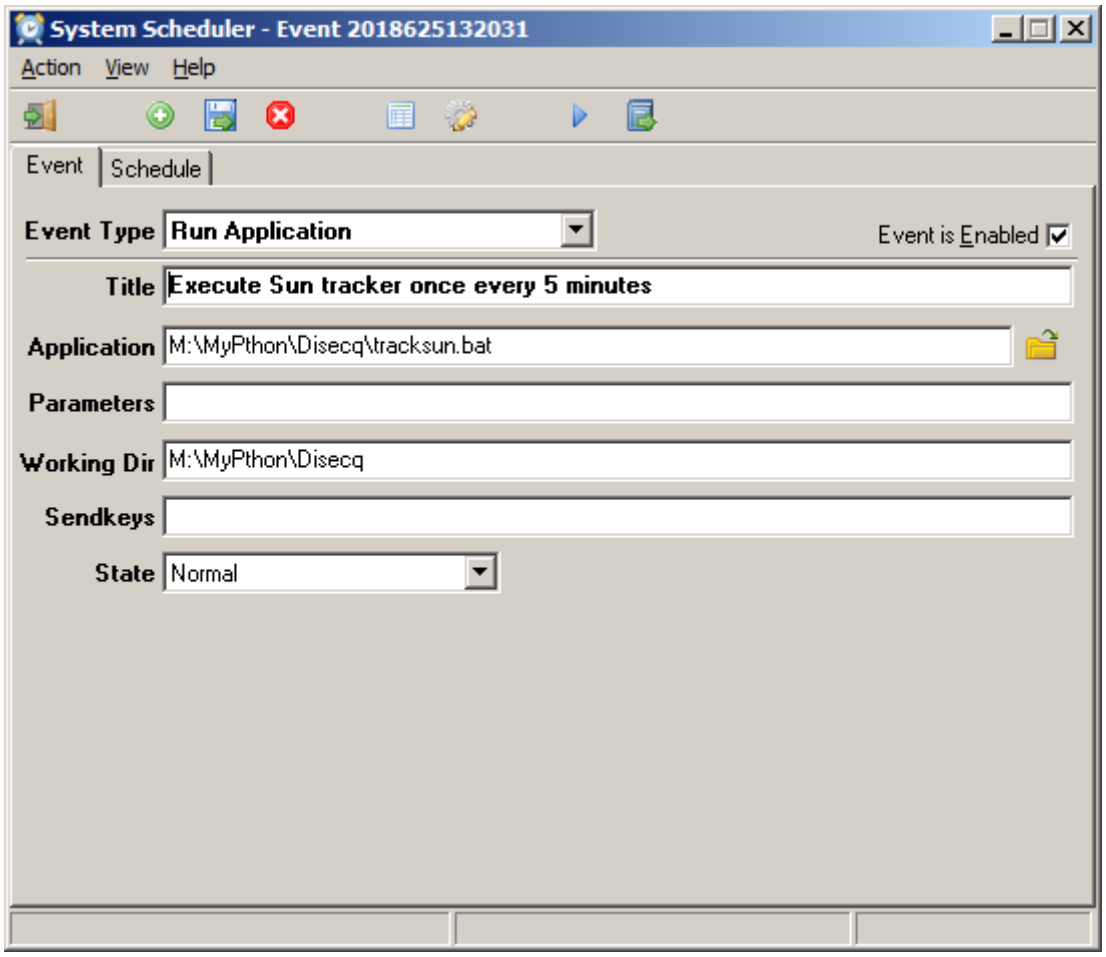


Fig. 11: Screenshot for Sun tracking event. Application and Working Dir path may look different for each user.

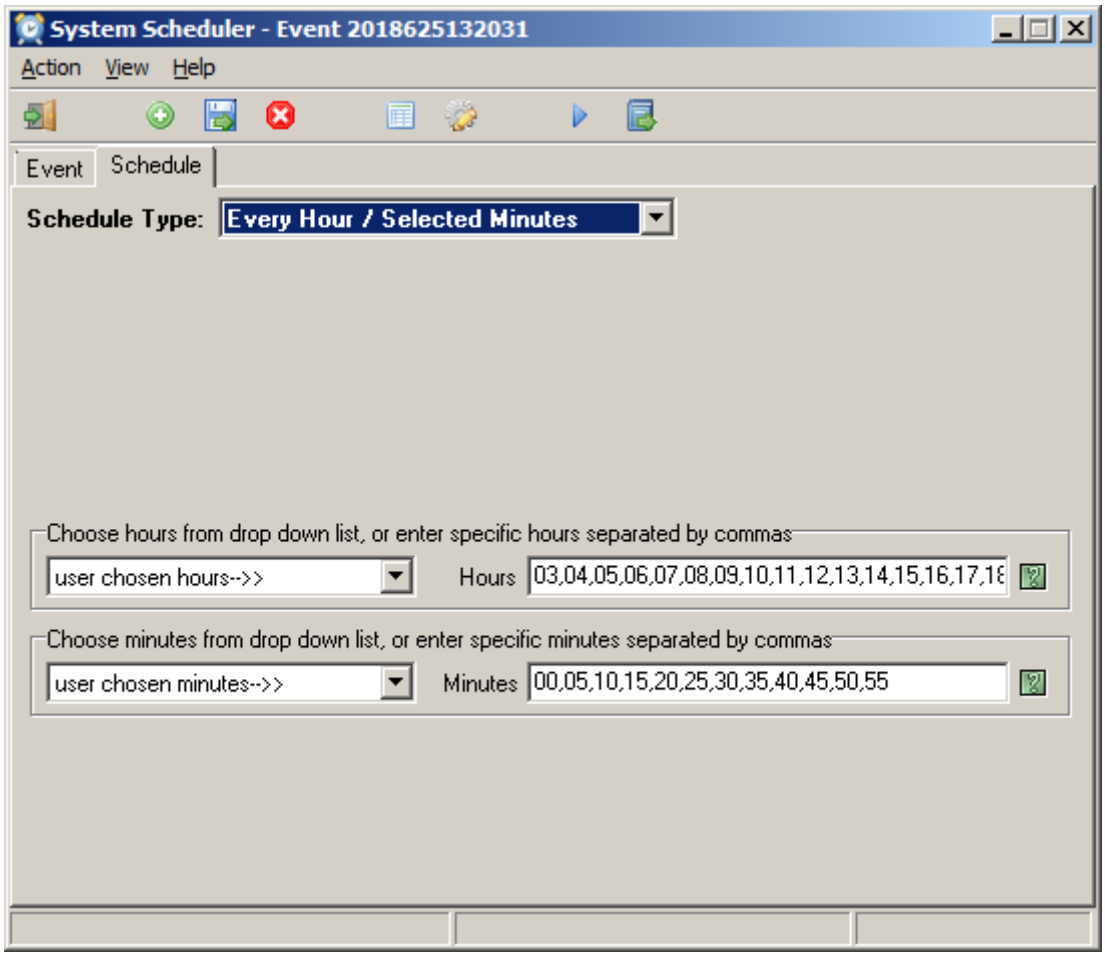


Fig. 12: Screenshot of schedule. Tracking in Europe is useful between 03 and 20 every 5 minutes. Other continents/countries need to find out their active Sun time expressed in UT.

Executing batch file **tracksun.bat** to run Python script

```
rem this batch file executes the sun tracker according to location, date and time

rem Chr. Monstein, 2018-06-25

c:\users\HB9SCT\Anaconda2\python.exe
C:/Users/HB9SCT/Documents/Disecq/sunpos_HA_DEC.py

rem python M:/MyPthon/Disecq/sunpos_AZI_ELE.py

timeout /t 5
```

Path to sunpos_HA_DEC.py may or sunpos_AZI_ELE.py as well as to python.exe may look different for each user.
Timeout is meant for testing to see error-messages and angles.

If you are neither happy with PYTHON nor with the system scheduler ssfree.exe, you always can send commands to the rotor manually, based on a simple terminal program like PUTTY or WinSCP or HYPERTERMINAL.
Command to send azimuth value: azi55.3 <ENTER>
Command to send elevation value: ele-12.8 <ENTER>
Command to send limit value: max65 <ENTER>
Command to get response from Arduino: ? <ENTER> {older versions}
Command to get help from Arduino: -h <ENTER>
Command to get software version from Arduino: -v <ENTER>

Hardware Installation

Install the antenna pole in a vertical position as precise as possible, every error in tilt produces pointing error in hour angle, declination, elevation or azimuth. Adjust the rotors azimuth in exactly north-south direction using a magnetic compass or even better find out when the Sun is in the meridian and adjust azimuth accordingly. Edit the Python script and set hour angle (ha) to 0.0 and run it. Then you may adjust rotor and antenna together in a way that the shadow of the front-dipole is exactly in the center of the antenna. Change Python script back into original version and run it. Depending on your location and depending on mechanical mounting method you may change the sign of hour angle ha -> -ha

Document history

Date	Version	Comment
21.12.2017	Very draft	Apprentice Marvin Wälti
15.07.2018	V03b	Several updates Monstein
14.12.2018	V04	Update for manual mode
15.04.2021	V05	Major hardware upgrade
10.08.2022	V06	http -> https
01.09.2022	V07	Resolution 1° -> 1/16°