

Simple Rotator control according to the N1VF controller

Marco, DL1BMH and Matthias, DD1US

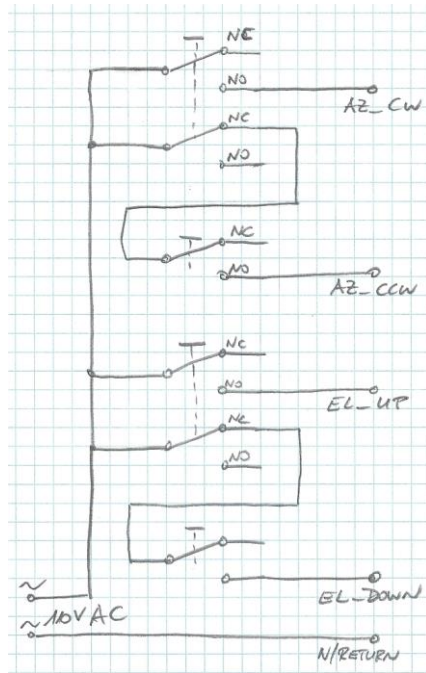
Rev 4, November 1st 2018

Hello,

recently, we acquired a batch of PZT drives VICON V380PTX. Those drives are pan / tilt rotors for TV cameras. They are quite solid and can be used as Az/EI-Rotators. However they do not feature any position sensors.



The first thing we did to try out the PZTs was to build a manual controller for the drives. Only 4 push buttons for EL_UP, EL_DOWN, AZ_CW and AZ_CCW and a fuse were needed. Two of the push buttons included double SPDT switches which were used to protect the motors from getting actuated simultaneously in opposite directions. Here is the simple schematic:



The simple circuit was built into a plastic encasing which serves as a nice hand controller. Input is from a 110V AC power supply, the output signals going to the motors in the PZT were connected by very rugged standard 12 pin MS Connectors (MIL-C-5015). We use these connectors as they provide enough pins to connect later also encoders which we plan to integrate into the PZTs.



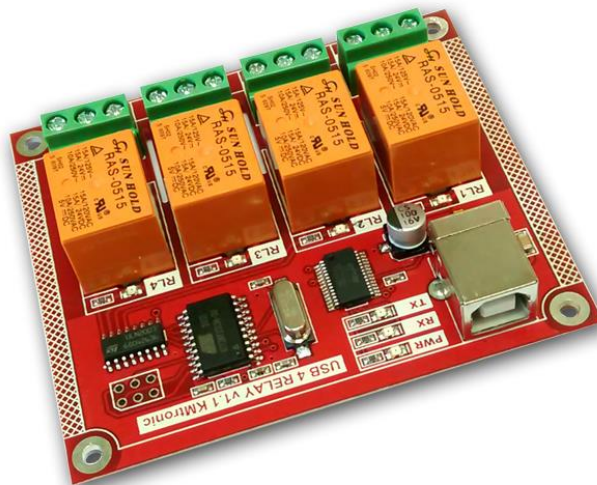


We intended to integrate magnetic encoders to the drives in order to sense the azimuth and elevation . However, we were also interested in a simple interim solution to control them from a PC without having position encoders. Fortunately, Ben Gelb, N1VF, had developed a simple rotator controller for a similar PZT drive and published it at Github (https://github.com/bgelb/sat_rotor).

Ben has been using a simple and inexpensive board controlling 4 relays with an ATMEL controller (KMtronic V1.0). It includes a FTDI USB-UART interface and thus provides a virtual COM port interface to the PC.



Unfortunately, later versions of this board look similar but are using a different ATMEL controller where the code for Ben's firmware will not fit. We only got the later version V1.1 of this board from https://www.kmtronic.com/usb-relays.html?product_id=53 (price was 28,80 Euros).



Thus, we had to replace the ATMEL controller on the board and insert an AT-Tiny 4313. Fortunately, the AT-Tiny 4313 has the same pinout. The controller concept developed by Ben N1VF relies on the hardware limit switches in the pan/tilt base of the VICON V330 being set at 0 degrees azimuth and 90 degrees elevation. Upon start up, it drives the pan/tilt base to the stops to acquire initial position. From that point on, it relies on timer-based dead reckoning to control the pan/tilt position. This is a fairly accurate method with the synchronous motors in the V330APT, as they drive at a constant speed based on the 60Hz rate of the AC power supply.

Well, we are living in Europe and use 230V 50Hz mains supplies. Thus, we have to use either 230V/110V transformers or AC/AC converters. Both approaches result in driving the rotator not with 60Hz but 50Hz supply. Also, it turned out that the VICON V380PTX is rather similar but not identical to the VICON V330 PZT drive. The speed is for instance different. Thus, the firmware had to be adapted. Marco spent quite some time optimizing the software for our needs which included changes to the timers, interrupts, get around the integer limitations of the AT4313 controller and several more issues. As the firmware from Ben N1VF is utilizing almost the complete memory this was a quite difficult task. Thus, one limitation is that the hysteresis of the controller is 5 degrees.

That means that the rotator will only be actuated when the error between the present and new position is more than 5 degrees, both Azimuth and Elevation.

Finally, Marco got the modified firmware successfully. As the original firmware it emulates a YAESU GS-232 interface. The RS232 parameters are 19200bd, 8bit, no parity. Not all commands of the GS-232 command set are supported.

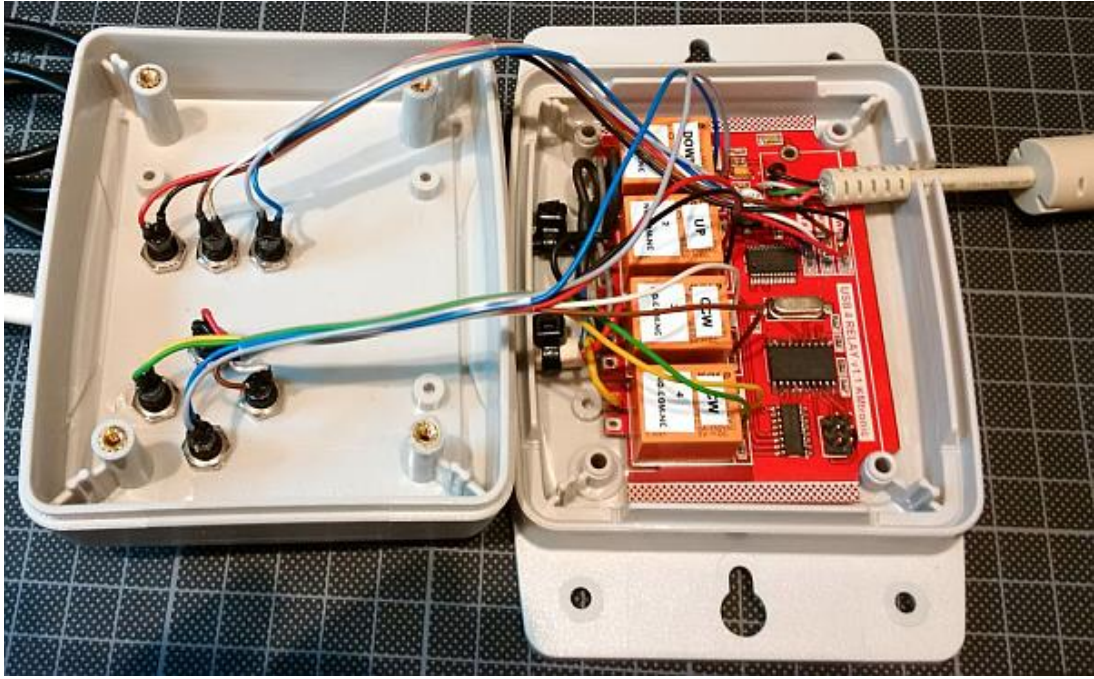
Only the following commands are supported:

- C → Output of present Azimuth Value
- C2 → Output of present Azimuth and Elevation Values
- B → Output of present Elevation Value
- Mxyz → Input Target Azimuth Value
- Wxyz0abc → Input Target Azimuth and Elevation Value

Each command has to be terminated with a <CR>

Here are some pictures of the N1VF controller built into a small plastic encasing. I had to cut off part of the PCB to fit it in and also removed the USB-Jack and soldered a USB cable directly to the PCB. Finally replacing the on-board SMD-LEDs with 3mm LEDs mounted in the encasing is very helpful during testing and later operations.





The controller is powered by the USB-port. The mains cable will supply the 110V for the rotator. The relays inside the controller switch the respective rotator ports for turning in CW/CCW respectively UP/DOWN directions.

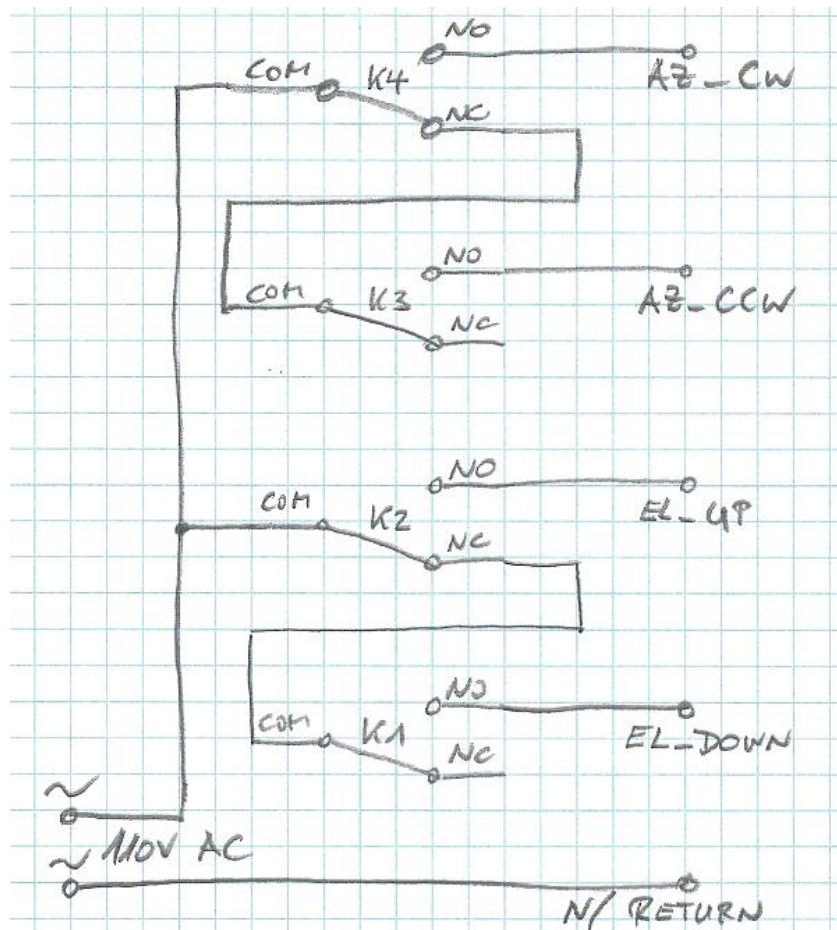
The relays used have Normally Open (NO) and Normally Closed (NC) Contacts each capable of switching max:

- 12VDC/15A
- 24VDC/15A
- 125VAC/15A
- 250VAC/10A

The relays are wired as such:

- K1 EL_DOWN
- K2 EL_UP
- K3 AZ_CCW
- K4 AZ_CW

Even though the firmware in the controller should prevent activating the supply voltage of a motor in both directions simultaneously, we decided to use the free contacts of the relays for additional protection. Below you can find the used wiring which ensures that EL_UP&EL_DOWN respectively AZ_CCW&AZ_CW can never be actuated simultaneously.



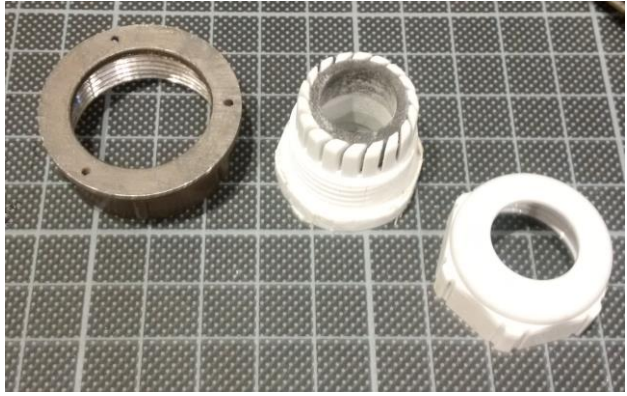
We decided to use very rugged connectors to connect the various controllers to the different PZTs.



These standard MS Connectors (MIL-C-5015) are designed to operate in electric AC or DC applications with frequencies up to 3MHz. The connector back shell is made from high grade aluminium alloy. These connectors have a threaded coupling nut for ease of application and require not tooling as the contacts are manufactured for soldered connections. These circular connectors, specially designed for military applications, are nowadays widely used also in the industry where strength, reliability and high availability in other countries is essential. We modified them as we liked to have the jack in the part with the outside thread and the plug in the part with the threaded coupling nut. As the inserts can be exchanged this can be easily done. Also, we changed the strain-relief to have a better protection of the cable and the wires.

Below please find some pictures of our setup:





By the way it turned out to be helpful to use a standardized wiring scheme for the rotators and connectors and here is what we ended up with:

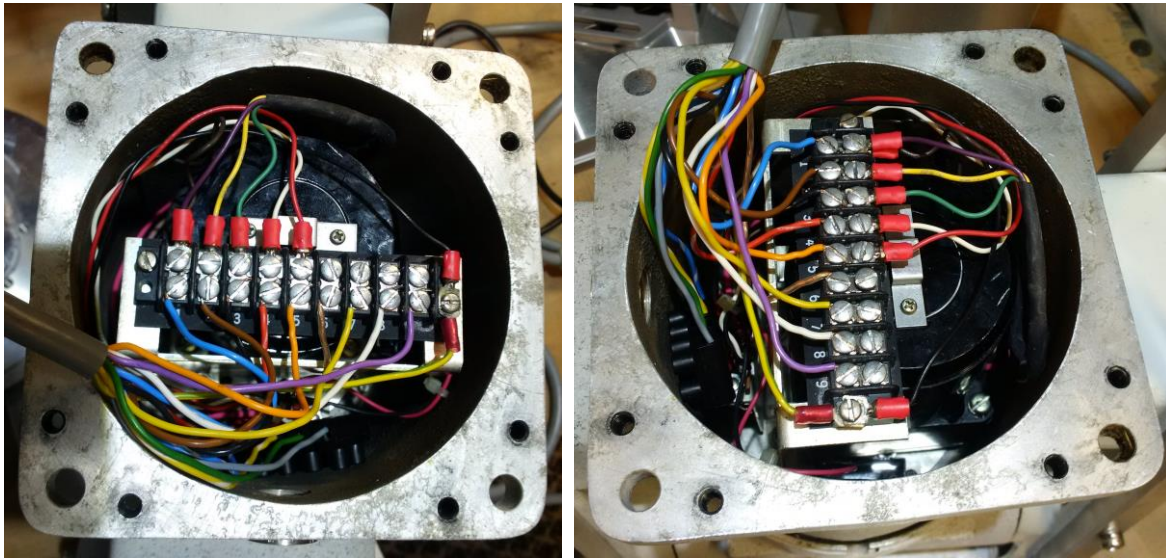
PIN# Rotor	1	2	3	4	5	6	7	8	9	10		
Signal	N/Return	EL_Down	EL_Up	AZ_CCW	AZ_CW	Sensor1	Sensor2	Sensor3	Sensor4	Ground	Sensor5	Sensor6
Colour wire in rotator	brown & purple	yellow	green	white	red					Black		
Colour wire to control interface	blue	brown	black	red	orange	transparent	yellow	white	purple	green/yellow	green	gray
PIN# connector	4	2	1	11	12	3	5	8	7	9	6	10

Of course, the colour coding of the cable will depend on the cable you have on hands. We are using a 14 wire cable with only 12 wires used (blue/white and yellow/white are unused).

In the case of the N1VF interface the sensor wires are not used and thus I ended up using only a 6 wire cable. Here is the connection table:

PIN# Rotor	1	2	3	4	5
Signal	N/Return	EL_Down	EL_Up	AZ_CCW	AZ_CW
Colour wire in rotator	brown & purple	yellow	green	white	red
Colour wire to control interface	Blue & green	gray	black	red	yellow
PIN# connector	4	2	1	11	12

Here are some pictures of the connector block in the rotator. On PIN#1 of that connector block 2 wires (brown and purple) are connected.



So far, I could successfully operate the interface with the rotator driver of Ham Radio Deluxe and also with my favourite rotator controller PSTRotator. In PSTRotator you have to select in "Setup" and then "EL /AZ+EL Controller" the line item "ST3 Fox Delta". For satellite tracking you can use the built in Satellite Tracker of PSTRotator or Orbitron connected with the "MyDDE" interface to PSTRotator.

We are always interested in feedback. Please send it to the Email address given below.

Best regards

Marco and Matthias

Email: dd1us@amsat.org

Homepage: www.dd1us.de