Auto Guider interface using Raspberry Pico (Document version 21)

Warning: Please be careful when interfacing to your telescope. I cannot be responsible for any damage caused due any errors, either due to your wiring, modifying the motor etc or any software. I have tested this on my LX10.

INDEX:

Figures:

1.0 Introduction.

This interface enables guiding using guiding software such as PHD2 on a PC, or my software for a Raspberry Pi.

Communications between the PC / RPi and Pico is via USB. The interface also gets its 5v from the USB. It can be interfaced to a Meade LX10, ST-4 or other types of mount. The ST-4 interface uses a 4-channel Phototransistor Output Optocoupler, the ILQ74. The Meade LX10 uses either mechanical or solid state relays, or for the more adventurous PWM.

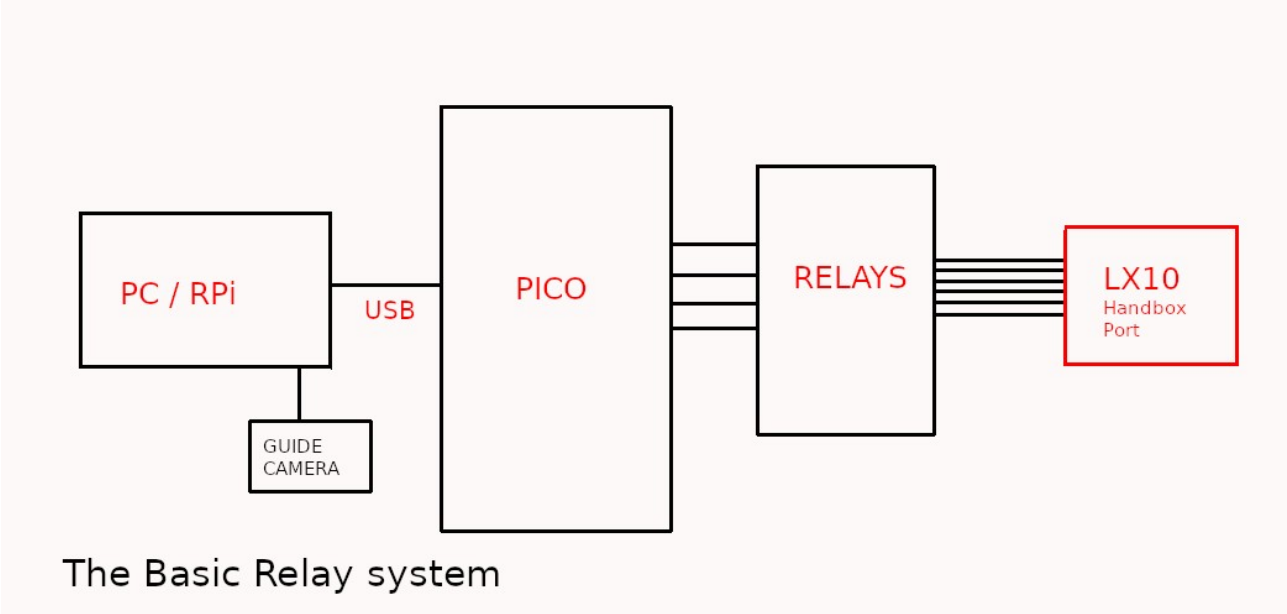


Figure 1.

With the Relay system the LX10 is operated as normal, the interface provides pulses into the Handbox port to guide the telescope.

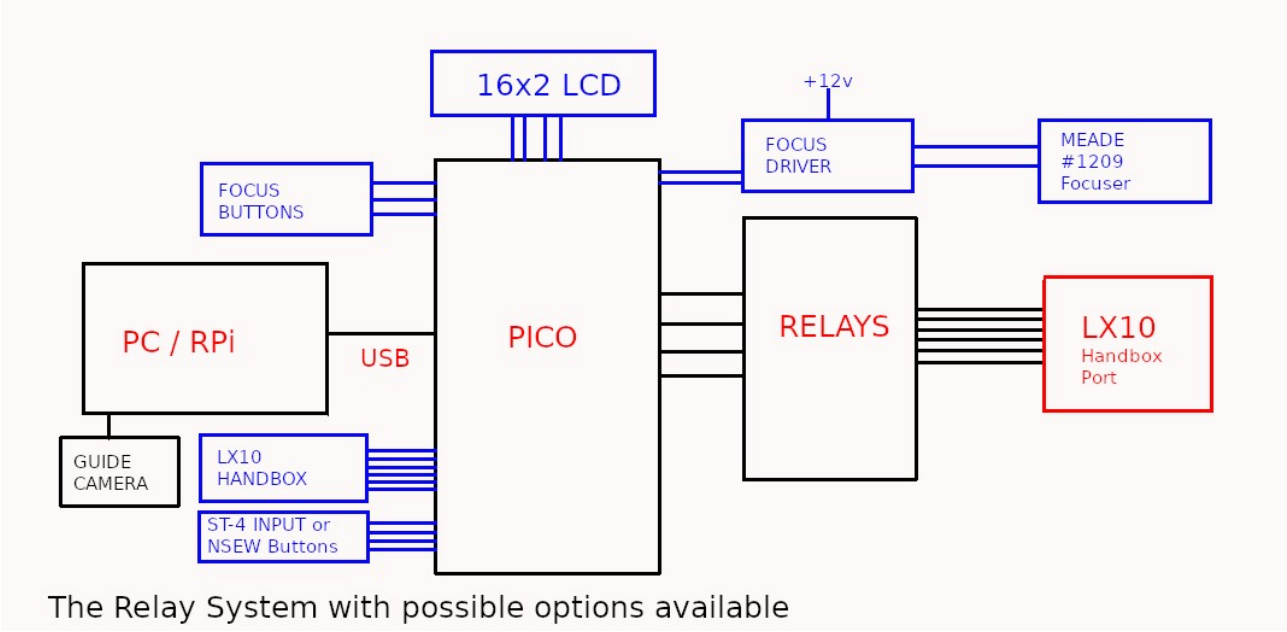


Figure 2.

You will need to provide the interface from the Pico to the telescope/mount port. The Pico provides 4 outputs for N,S,E & W (Dec +,Dec -,RA- & RA+). These are +3.3v when active (although these can be inverted in the Micropython code for active low relay boards), note many boards available are for Arduinos requiring 5v so may not work with the Pico.

1.1 Wiring schematic using relays.

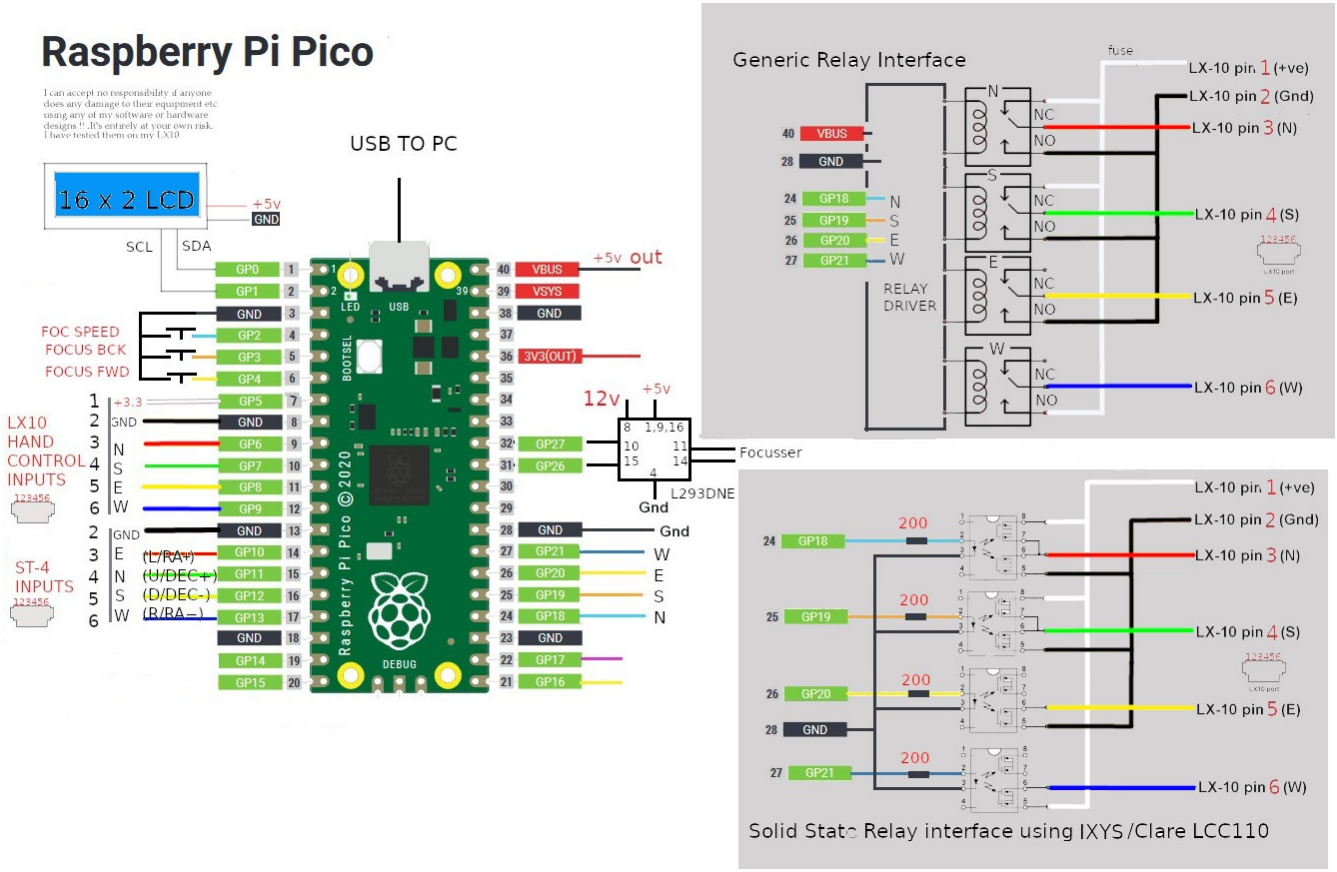


Figure 3.

Here is one example of an available board that should work as it uses the same GP outputs, connect as shown in the generic relay interface.

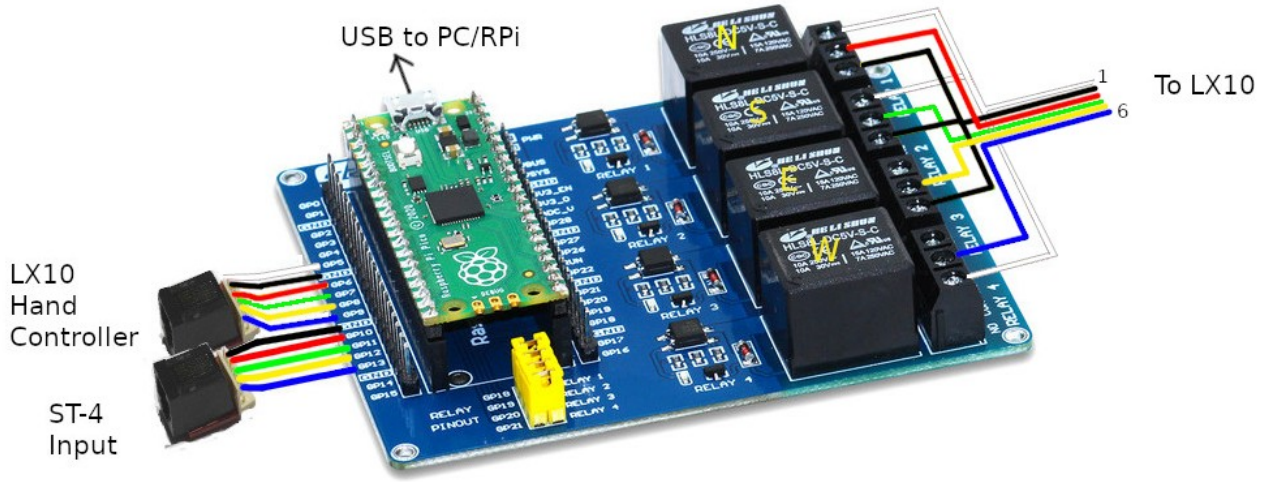


Figure 4.

1.2 Parts for SB-components Relay Board Version:

Essential:

https://thepihut.com/products/raspberry-pi-pico
https://thepihut.com/collections/pico/products/raspberry-pi-pico-relay-board
Cable to LX10

Optional:

https://proto-pic.co.uk/product/sparkfun-bob-14021-rj11-breakout/
https://proto-pic.co.uk/product/sparkfun-prt-00132-rj11-6-pin-connector/
https://proto-pic.co.uk/product/sparkfun-prt-10366-jumper-wire-0-1-6-pin-4/
https://proto-pic.co.uk/product/sparkfun-prt-10371-jumper-wire-0-1-6-pin-6/
https://proto-pic.co.uk/product/sparkfun-prt-10365-jumper-wire-0-1-5-pin-4/

You can connect a LX10 hand controller into the Pico to also provide manual control. There are also ST-4 control inputs, or you could make your own hand controller using 4 push buttons from the ST-4 inputs to gnd.

1.3 Fitting a 16x2 LCD.

There is also an option to fit a 16x2 i2c LCD display.
Example https://www.amazon.co.uk/GeeekPi-Character-Backlight-Raspberry-Electrical/dp/B07PGZ9B51/ref=sr_1_3?crid=38WYQ7K2UVZZS&keywords=i2c+16x2+lcd+module&qid=1644850602&sprefix=16x2+i2c+lcd%2Caps%2C1014&sr=8-3

You need to install the drivers **lcd_api.py** and **pico_i2c_lcd.py** from:

https://raw.githubusercontent.com/dhylands/python_lcd/master/lcd/lcd_api.py
and
https://raw.githubusercontent.com/T-622/RPI-PICO-I2C-LCD/main/pico_i2c_lcd.py

advice on installing the 16x2 lcd here..

https://www.tomshardware.com/uk/how-to/lcd-display-raspberry-pi-pico
In the pico main.py script set use_lcd = 1 if you have connected the LCD, and installed the drivers.

1.4 Prototype Design for Solid State Relay board, using LCC110s or LCC120s.

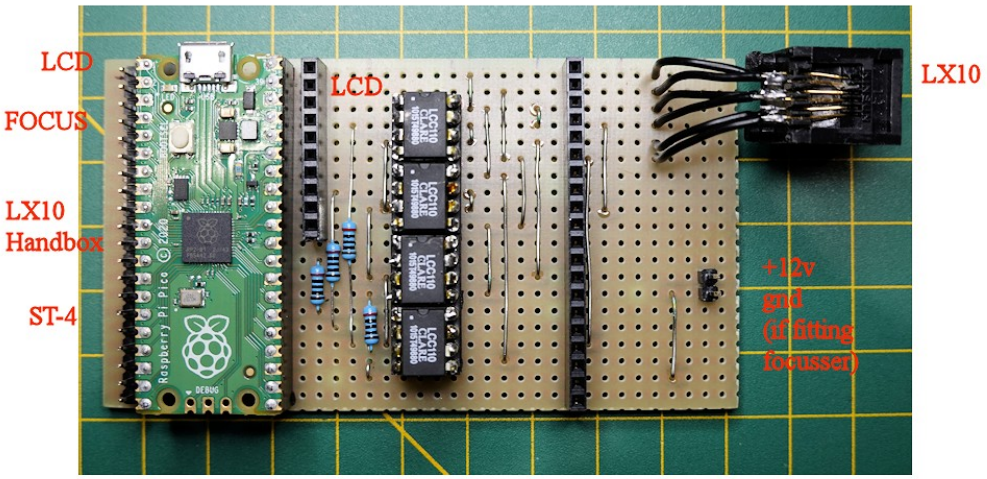


Figure 5.

SSR Relay Board layout - Top

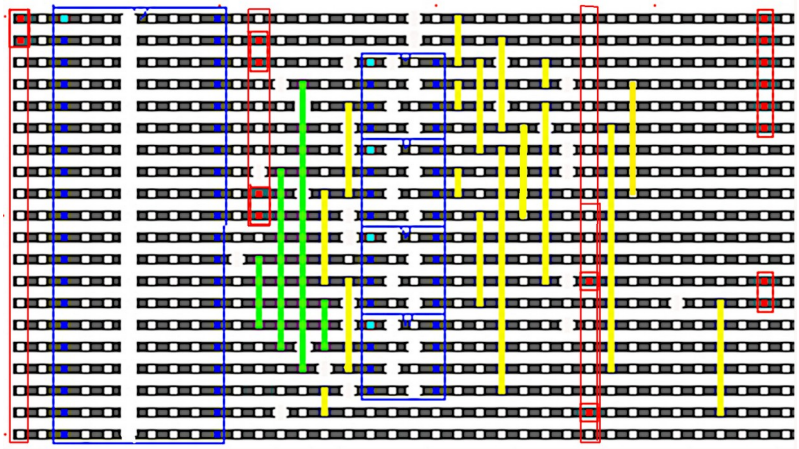


Figure 6.

Bottom showing track cuts

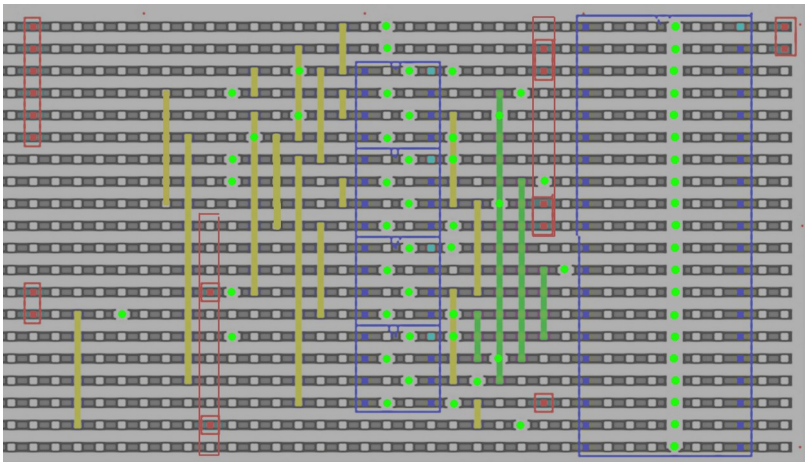


Figure 7.

Yellow – Wire links
Green – 200ohm Resistors
Blue - ICs (LCC110/120) / PICO
Red – Connectors

Note this board is the same size as the PWM version and has the option of a daughter board for the Meade #1209 Focus controller.

2.0 PWM version.

There is also the option to try using PWM to drive the LX10. This is for experimenting and is not proven to be sufficiently stable for astrophotography yet. At your own risk !!.

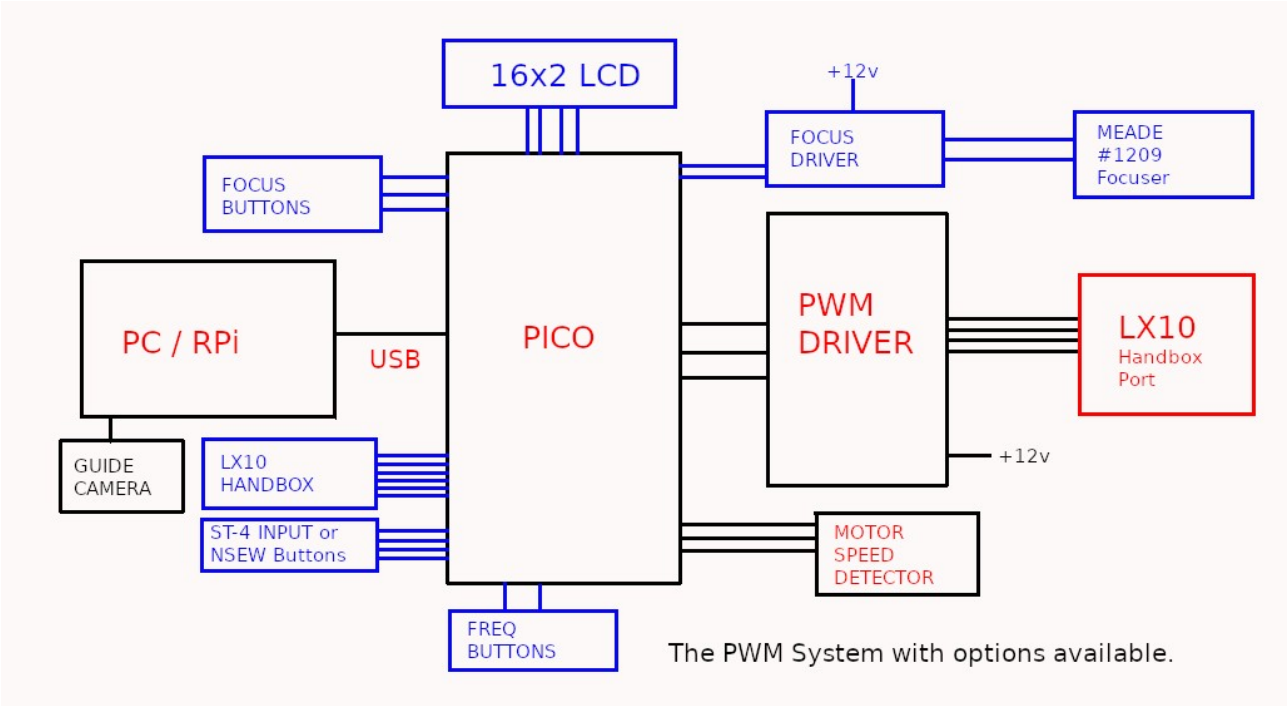


Figure 8.

To provide motor speed detection you need to fit a led transmitter and detector to the motor in the LX10, which involves drilling a hole in the motor. (see section 2.5)

This was a modification proposed by Mark on the LX10 forum.
https://groups.io/g/LX-10/files/LX10%20Tracking%20Drive%20Replacement/PMDC%20frequency%20control%20notes%20002.pdf

The interface provides a PWM signal to power the RA motor at the required rate, and by monitoring the motor speed can maintain a constant rate (set for 20.89Hz). It also provides PWM for the DEC motor. PWM is disabled in the software as supplied, simply change it to pwm_on = 1 and motor_ctrl = 1 to enable.

The outputs from the L293DNE are connected to the LX10 port.
ENSURE the LX10 is switched OFF and no power connected.
This can also be controlled by PHD2 etc.

2.1 PWM Prototype, with optional LCD, FOCUS and N,S,E & W buttons (connected to ST-4 inputs)

Figure 9.
PWM Prototype with optional daughter board for Meade #1209 Focuser control

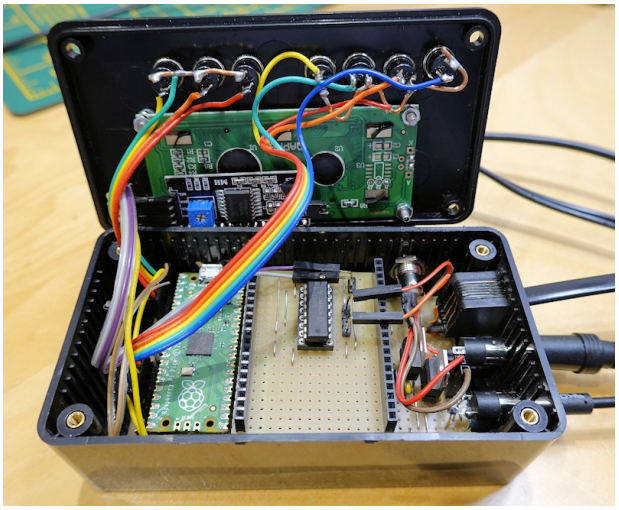


Figure 10.
PWM Prototype Circuit Board

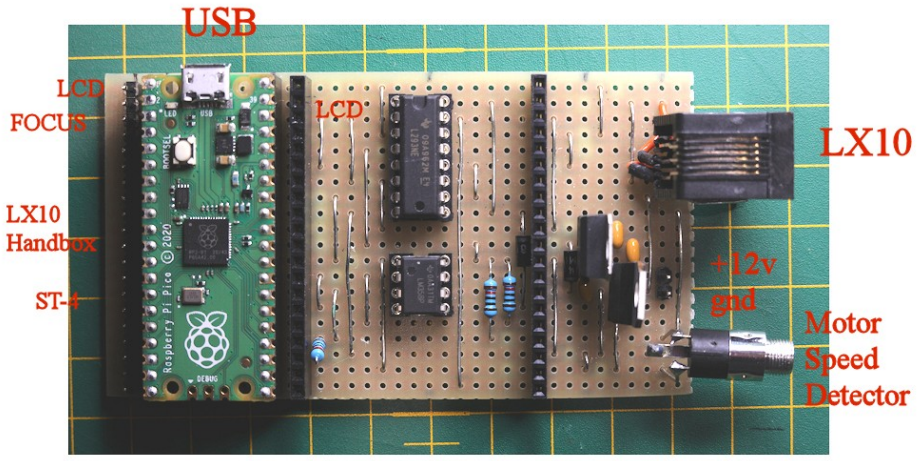


Figure 11.

The two connectors either side of the ICs are for a daughter board which can carry the IC for the focus controller for the Meade #1209. Two pins on the left hand connector are used to power the optional LCD Display.

2.2 Circuit Board layout for PWM Prototype

Top

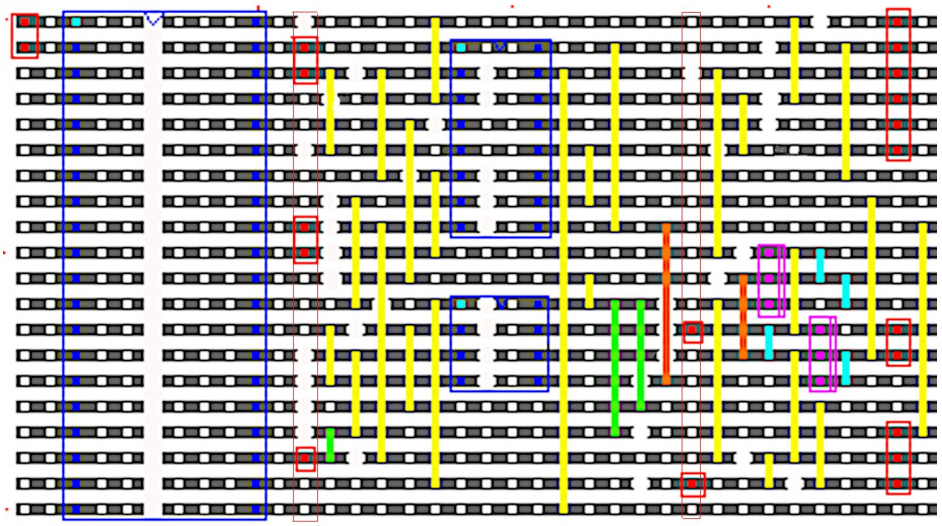


Figure 12. **Bottom** showing track cuts

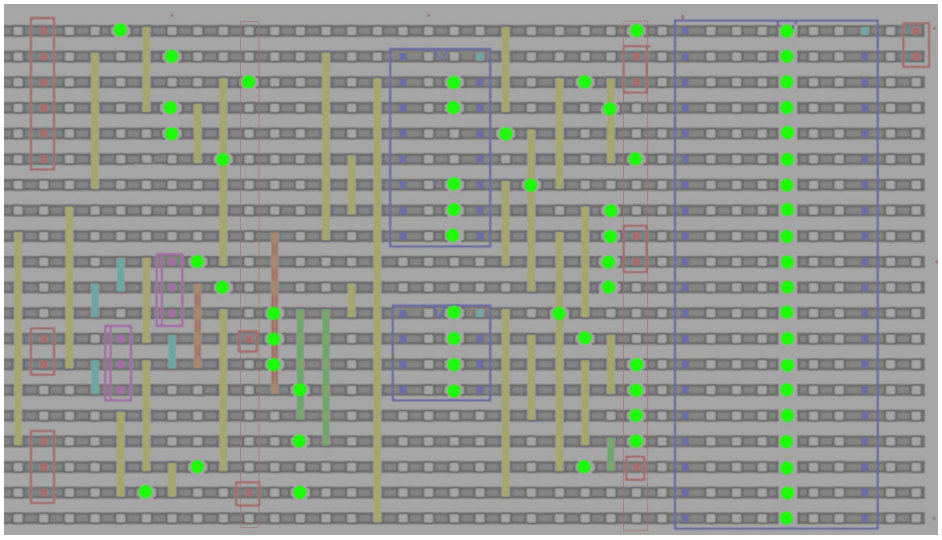


Figure 13. **Components**

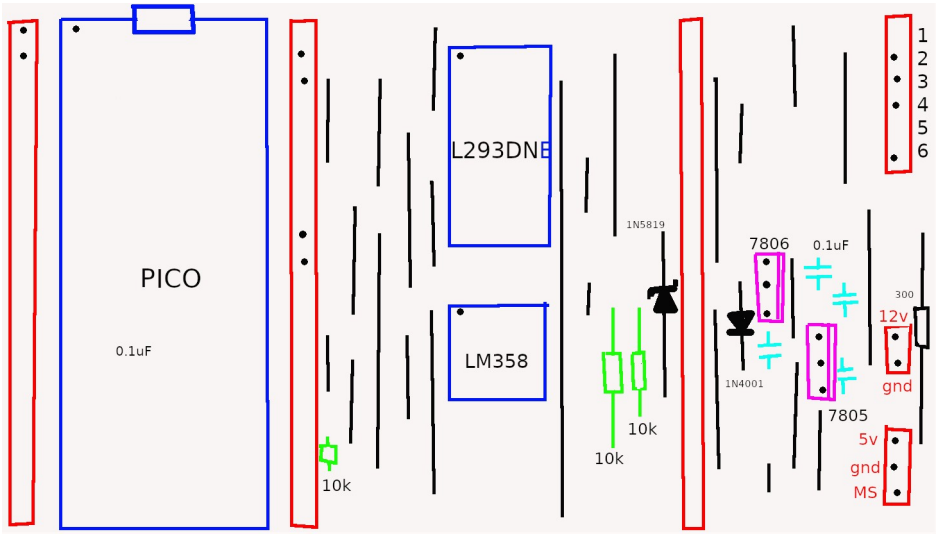


Figure 14.

2.3 Prototype Daughter Board for Meade #1209 Focuser control

Top

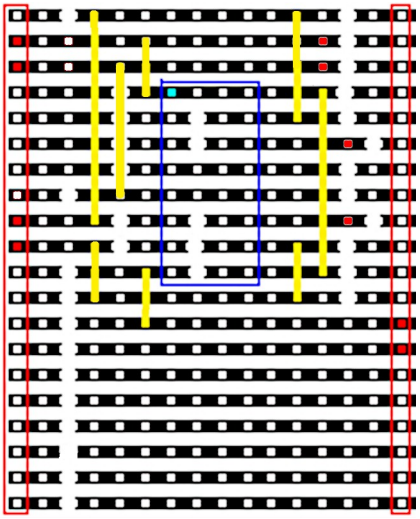


Figure 15.

Bottom showing track cuts

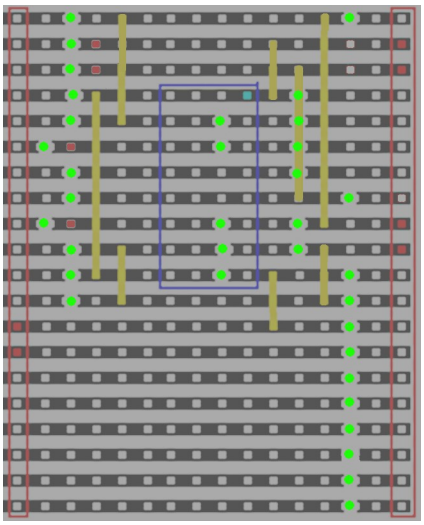


Figure 16.

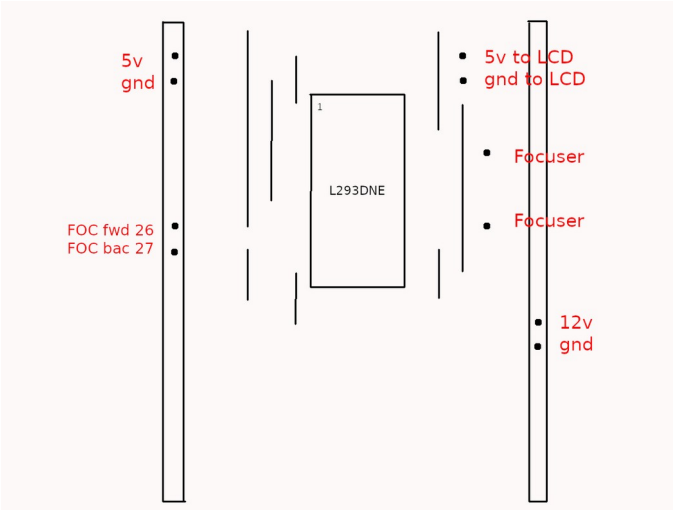
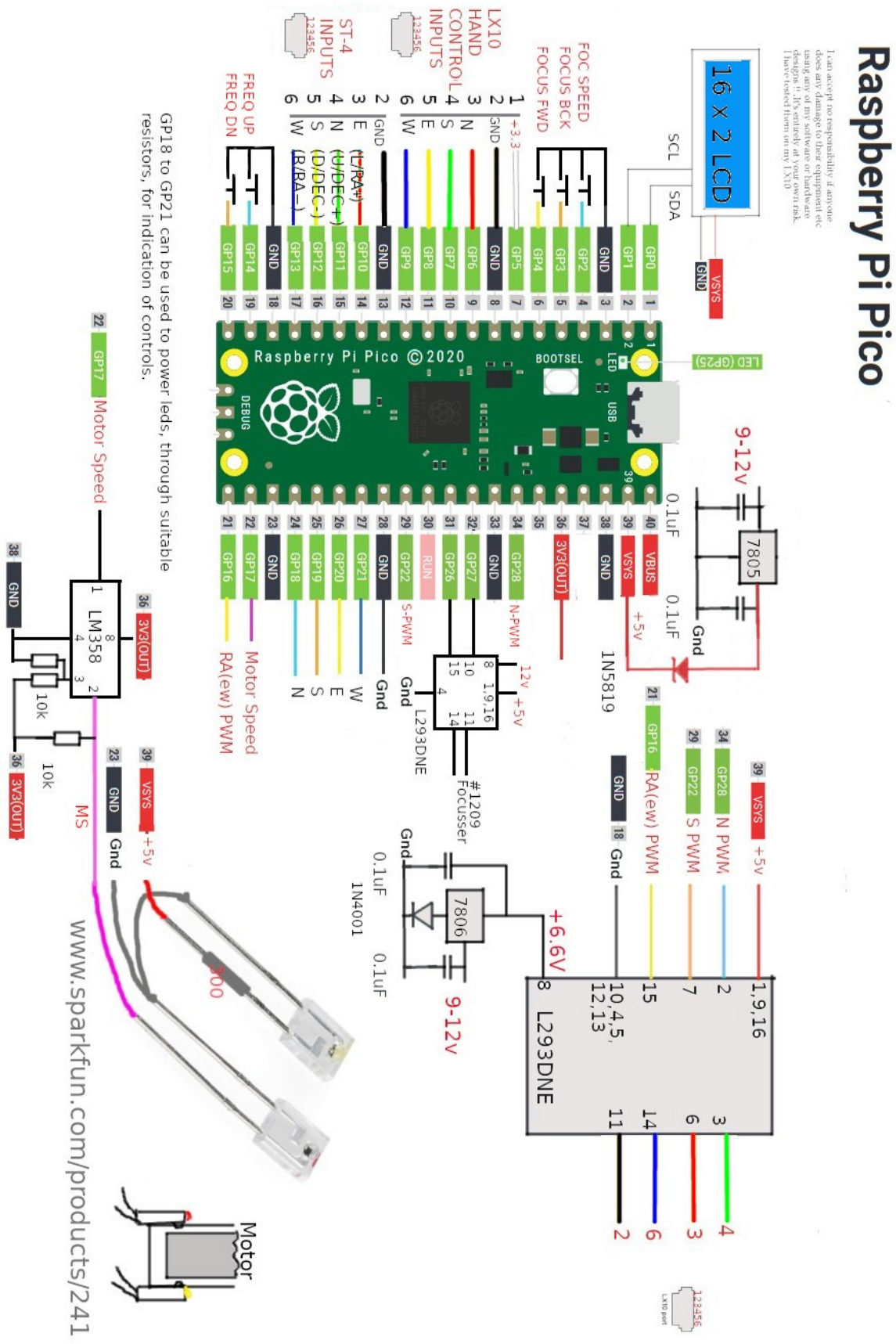


Figure 16a. Top

This board could be modified to mount on the SB Components relay board if required by fitting a socket for the left hand connector and connecting the 12v to the board rather than using the right hand connector.

2.4 PWM Schematic

Figure 17.



2.5 Modifying the LX10 Motor for speed detection.

You need to remove the motor from the LX10 and **CAREFULLY** drill a hole in the opposite side to the existing ‘spare’ hole in the body. **Ensure you don’t damage the coils within the motor.** You then need to mount the LED transmitter and receiver on the motor so they are operating through the holes on either side of the motor.

The LED on the back of the motor can be held in place by the spring clip that holds the motor, for the one on the front of the motor will either require the plastic disc, which sits between the motor and the gearbox, to be modified to allow space for the device or making a new disc to suit.

I then routed the wires to a 3.5mm stereo socket mounted on the back of the battery compartment.

Ensure you insulate all leads.

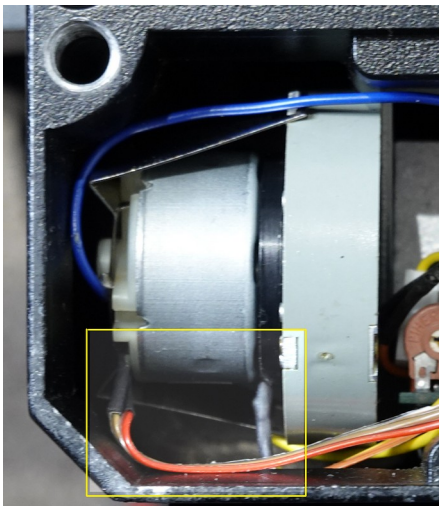


Figure 18. The LX10 motor showing the connections to the LED Tx and Rx.

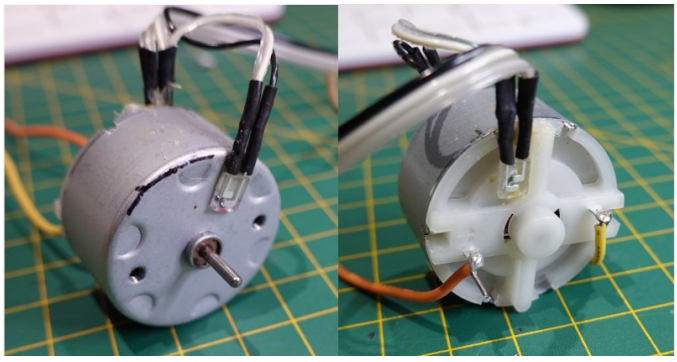


Figure 18a. A spare motor showing where the LED Tx and Rx are mounted.

3.0 ST-4 output interface (NOT for LX10).

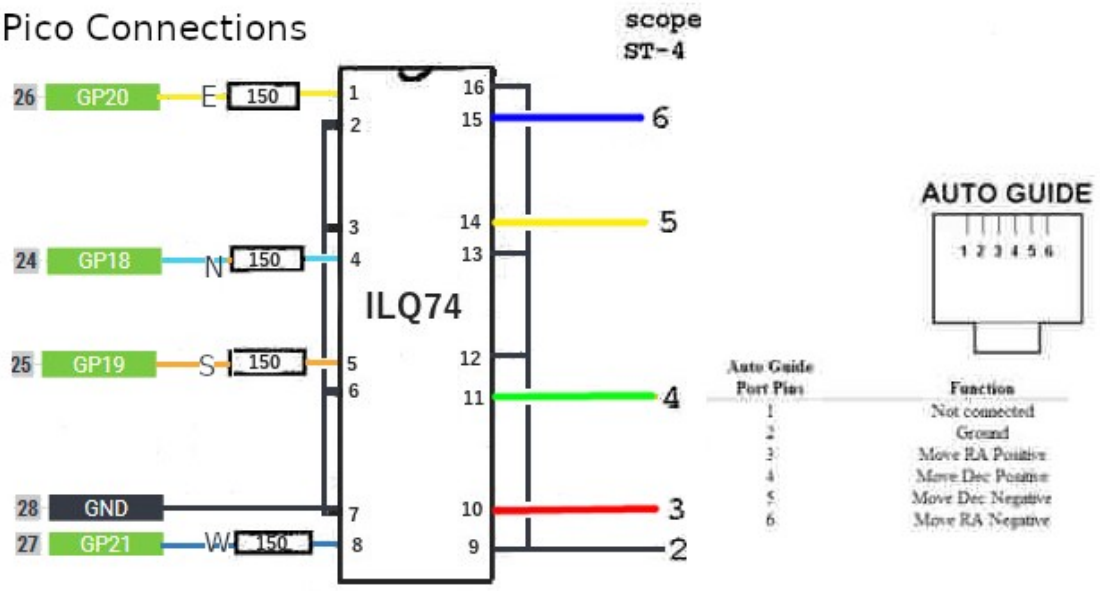


Figure 19.

3.1 The optional LCD display

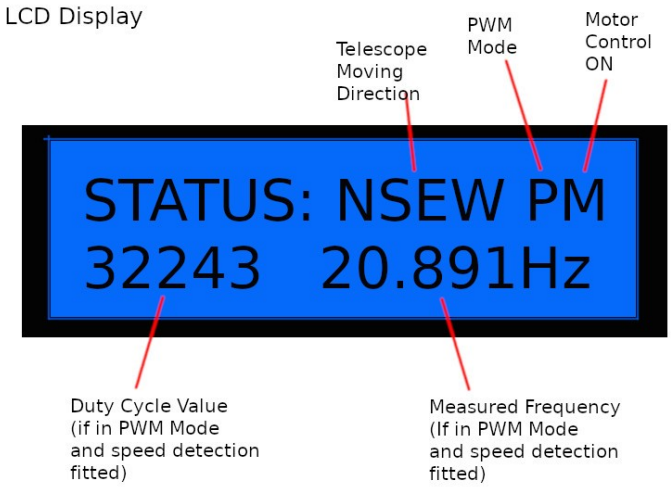


Figure 20.

4.0 LX200 ASCII Commands between PC / Raspberry Pi and Pico.

When sending any commands you MUST use the exact format or you may crash the software

:Mx move in required direction, where x = n,s,e or w. eg. **:Mn**
:MgxYYYY move in required direction for a required period where x = n,s,e or w and Y = 0000 to 9999. eg **:Mgn1000** move North for 1 second.
:Qx quit move, where x = n,s,e,w or # (all). eg. **:Qn**

Additions if using PWM control:

:RAVXXXXX will set the Duty Cycle (and volts to the motor) Eg :RAV32000
:RAFXXXXX will set the target motor freq Eg. :RAF02089 set 20.89Hz
:MCO motor control ON
:MCo motor control OFF

4.1 Additions if using a Meade #1209 focuser controls:

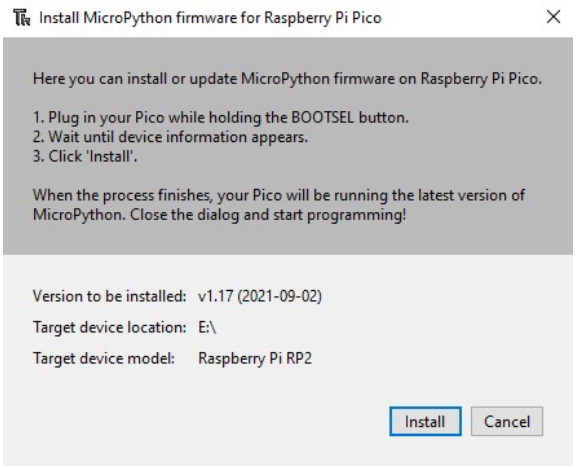
:F+ Focus forward
:F- Focus Backwards
:FS or :F1 Focus speed SLOW
:FF or :F4 Focus speed FAST
:Fx where x = 1,2,3 or 4 Focus speed 1 to 4
:FP+XXXXX Focus Forwards for XXXXX mS eg FP+01000 for 1 second
:FP-XXXXX Focus Backwards for XXXXX mS
:FQ Focus QUIT/STOP

The following commands are also acknowledged to enable connection with PHD2:
:GR,:GS,:GW,:AT0,:GD,:GA,:Gt,:GVT,:GVP,:GVF,:GVN,:GVD,:?+

4.2 Programming the PICO.

The easiest is with Thonny, this should be already installed if using the Raspberry Pi OS and can be installed on a PC from www.thonny.org

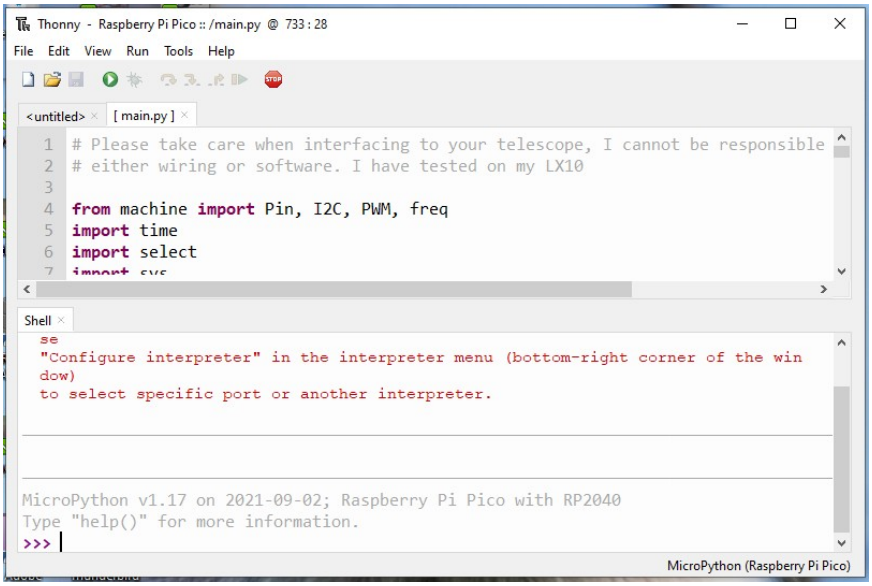
On the PICO hold down the BOOTSEL button whilst connecting the PICO to the PC or Rpi with the USB cable. The following screen should appear, press INSTALL.



Now run THONNY , if it doesn't say Micropython (Raspberry Pi PICO) in the bottom right hand corner then click on it and choose Micropython (Raspberry Pi PICO) from the list.

Now from file open main.py , then click on file then Save as.. and save it to the PICO as main.py.

Click on the RUN symbol. At the bottom of the window type :Mgn2000 and press return.
The green LED on the PICO should light for 2 seconds and the N relay operate if connected.
(ignore any warnings about unexpected echo)



4.3 Using PHD2 on a Windows PC.

I have tested it on a Windows PC with PHD2 v2.6.11, ASCOM Platform 6.6.SP1 and drivers for Meade LX200 Classic and Autostar #494, #495, and #497.

https://openphdguiding.org/
https://ascom-standards.org/Downloads/Index.htm
https://ascom-standards.org/Downloads/ScopeDrivers.htm

Run PHD Guiding 2. Click on **Guide** and **Connect Equipment** and choose the required **Serial Port**. (Use Device Manager to find it if unknown). Click on **OK**

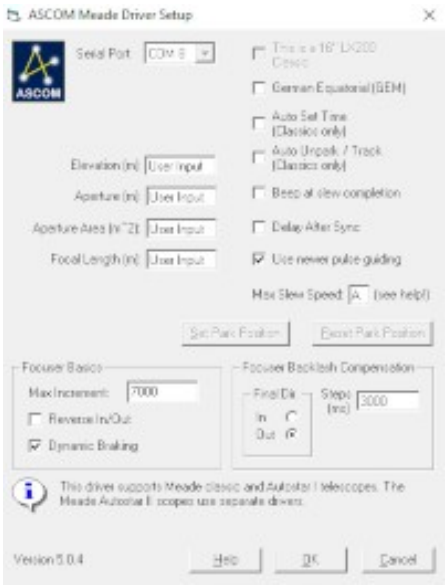


Figure 21.

Choose the **Meade Classic and Autostar 1 (ASCOM)** Mount (assuming you have installed the driver) and click **Connect.** Click **Close** if successfully connected.

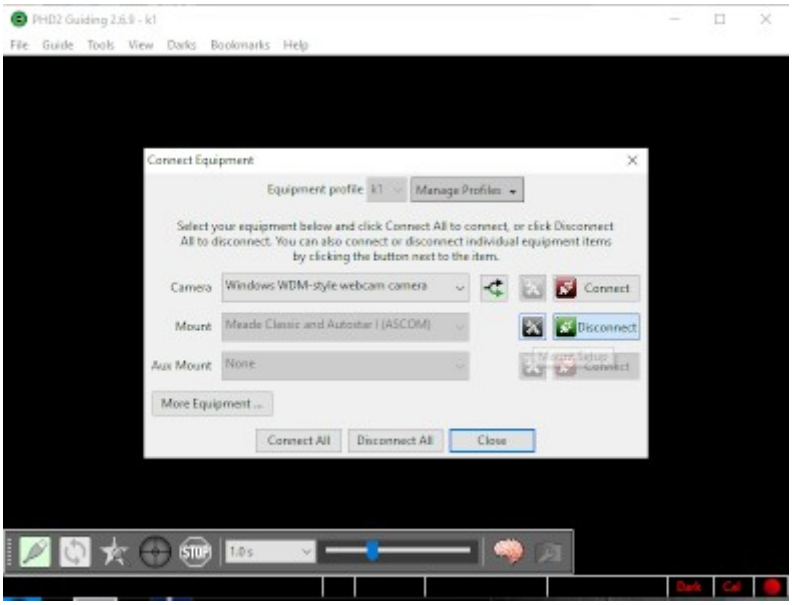


Figure 22.

To test click on **Tools** and then **Manual Guide**, then click on **North, South, East or West** and listen for the scope moving and / or watch the PICO green LED light and relay activate.

4.4 Using Pi_Autoguider (PiAG.py or PiLCAG.py) for guiding with a Raspberry Pi.

https://github.com/Gordon999/Pi-AutoGuider
https://github.com/Gordon999/Pi-LC-Autoguider
Pi-AutoGuider works with RaspiOS based on BUSTER
Pi-LC-Autoguider works with RaspiOS based on BULLSEYE

Copy PiAGL.py or PiLCAG.py into '/home/pi' on the Raspberry Pi.
Install opencv with 'sudo apt install opencv-python3'

Change 'serial_connected = 1', then Python3 PiAG.py or PiLCAG.py to run.

You can use a Pi Camera or a Philips 740/900 USB webcam attached to your guidescope.
Connect the pico to a USB port on the Pi. It should appear as '/dev/ttlAMC0'.

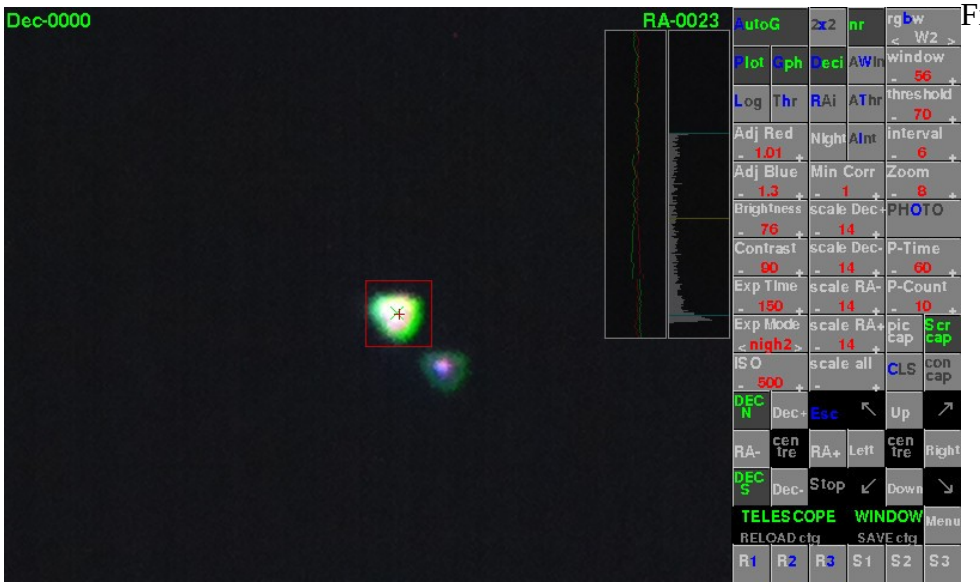


Figure 23.

In the python script set the following parameters to suit your setup:

CONNECTION SETTINGS

camera_connected = 1 # FOR TESTING with no camera attached / simulation ONLY, 1 for normal use
# setting camera_connected = 0 will run in demo / simulation mode
serial_connected = 1 # set serial_connected = 0 if you don't have the Rpi PICO connected.
use_Pi_Cam = 1 # if using the RPi camera set use_Pi_Cam = 1 , if set to 0 the program will use a USB camera.
Webcam = 0 # 0 = Philips, 1 = Logitech
usb_max_res = 1 # usb_max_res - Webcam max available resolution, depends on your Webcam
# all USB camera images will be taken using this resolution
# 0 = 320x240, 1 = 640x480, 2 = 800x600, 3 = 960x720, 4 = 1280x960,
# 5 = 1620X1215, 6 = 1920x1440, 7 = 2592x1944, 8 = 3280x2464, 9 = 5184x3688
# eg. Philips NC900 and Toucam Pro 740 won't work above 1, Logitech C270 max 4.
# (Pi camera uses a max of 7 (v1) or 9 (v2) automatically)

5.0 Some test results using the PWM interface.

Oscilloscope screen shot showing it maintaining 20.89Hz.

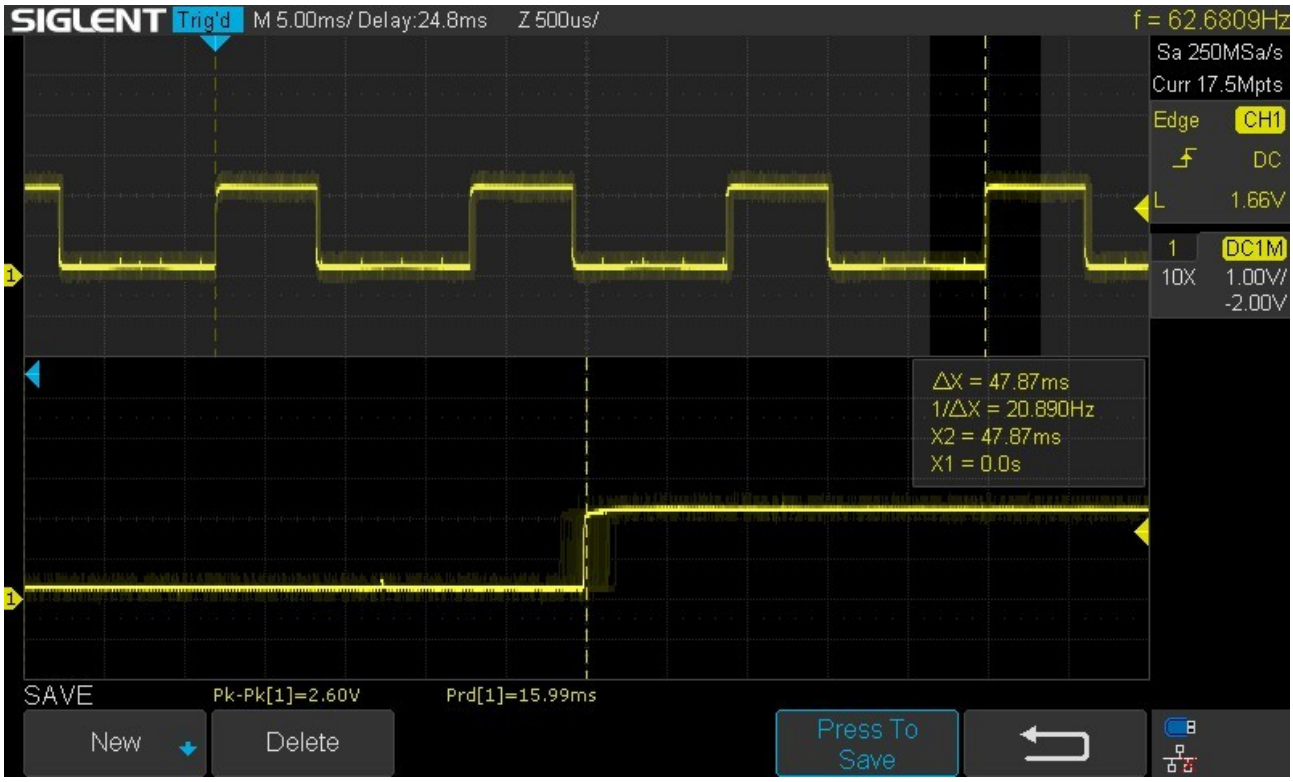


Figure 24.

The upper trace is the signal fed from the speed detection circuit, through the op-amp to the pico. The lower trace is a zoom of the area around the 3rd pulse showing the timings and their variations.

Difference between int (LX10 standard control), mco (PWM control with feedback switched OFF) and pwm (PWM control with feedback switched ON).

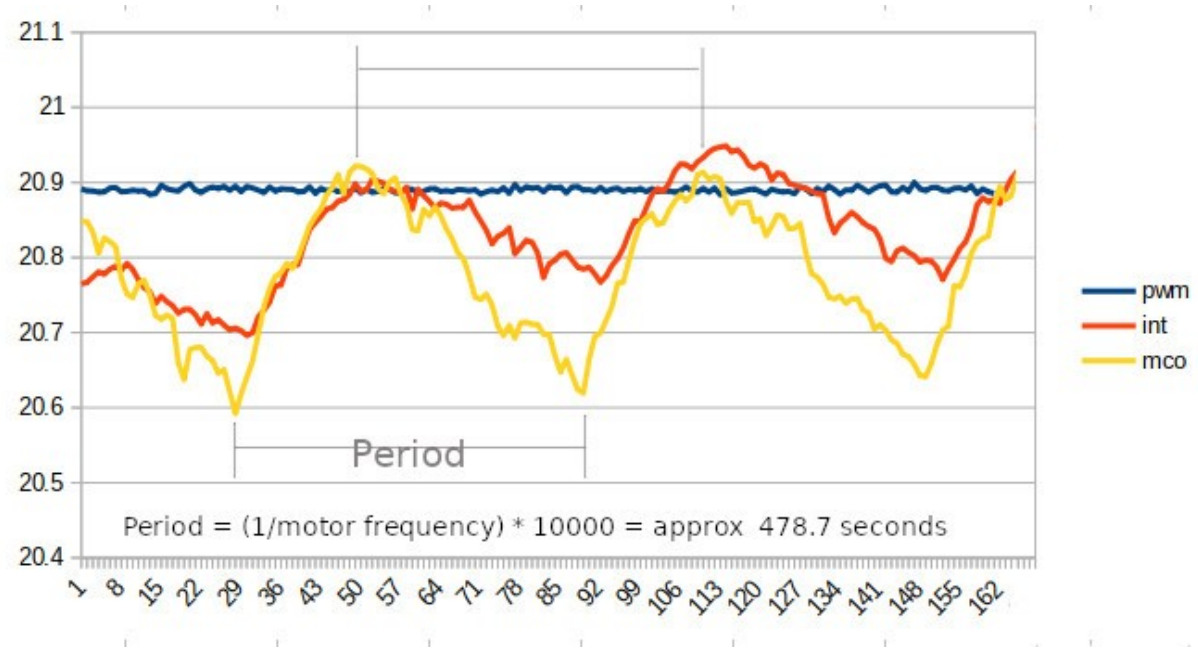


Figure 25.

The standard LX10 RA control shows Periodic Error at the rate of the drive worm (1/10000th of the motor rate).

The PWM with no feedback control also shows Periodic Error and this is worse than the standard LX10 RA control, presumable this is due the reduced torque from the motor whilst using PWM. (Note this has hopefully been improved by changing the PWM frequency to 376Hz, the 18th harmonic of 20.89Hz which should synchronise the motor, more testing required!)

The PWM control with feedback enabled shows significant improvement, maintaining a good rate at the set motor frequency of 20.89Hz.