MK1 Replacement of Celestron G-4 Dual Axis Motor Drive Model 93522

Paul Byrne

Table of Contents

1 Introduction

After my G-4 motor drive failed, and after looking at the cost of replacements, I decided to

Fig1 : G-4 Drive (“It's dead Jim”)

replace it by building one myself using as many components as I had at hand. After searching in various forums, I found a few projects using the Arduino UNO. These used timed interrupts to generate the required pulses to drive the stepper motors. However the Arduino UNO is based on the ATmega328p processor which I know has counter timers that could be used in pwm mode to produce these pulses instead. Arduino UNO clones are at hand, so I shall use these for this project.

2 Requirements

The G4 equatorial telescope mount uses two bipolar stepper motors one for driving the Right Ascension (RA) axis and the other for Declination (DEC). The RA is the celestial equivalent of longitude, and DEC the celestial equivalent of latitude (fig2).

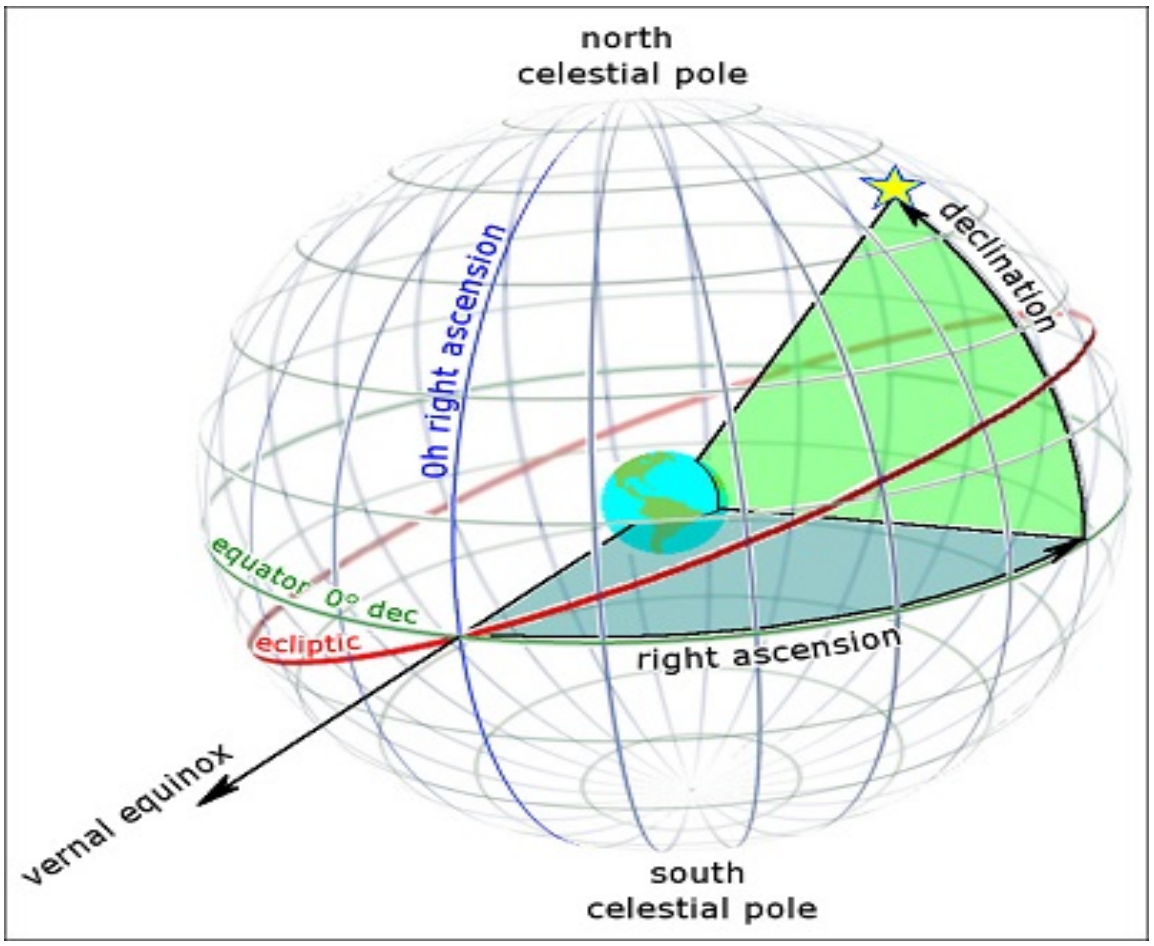

fig 2: Celestial Sphere

The telescopes mount RA axis is aligned so that it is parallel to the Earth's axis of rotation (fig 3). The RA motor which drives the axis, allows tracking of celestial objects by compensating for the Earth's rotation, and hence must run continuously rotating the RA axis by 15º an hour (24 hours x 15º = 360º) which is known as the sidereal rate. The DEC axis motor is used to assist in locating celestial objects, and thus only need to operate when a particular object is being located, and hence the rotation rate does not require great accuracy.

To assist in locating objects both the RA and DEC drives will have a fast mode equivalent to 8 times the normal sidereal rate, but the RA drive will automatically revert to the normal 15º per hour rate. The original drive had a X4 mode but was little used so is not required.

The stepper motor drivers and the Arduino UNO must operate from a single DC supply.

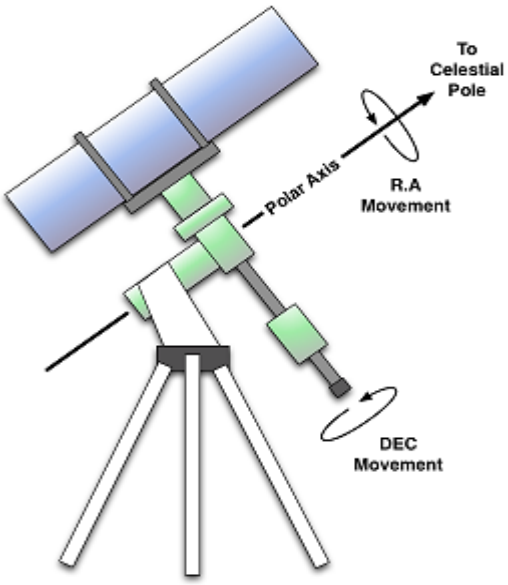

fig3: Equatorial Mount

3 Design Concept

The pulses required for driving the stepper motors will be produced using the ATmega328p timer counters. When configured these will continuously produce pulses independent of what the software on the processor is doing. For this project an Arduino UNO clone will be used which is based on the ATmega328p processor. However to utilise all the counter timers it will mean that the code will have to access the ATmega328p internal registers to control these timers, and not use any Arduino UNO features which use them such as **tone** and **delay**.

The pulses produced will feed into a DRV8834 stepper motor driver carrier which will drive the stepper motor.

4 Design

The RA and DEC stepper motors are driven using the DRV8834 driver, via a DRV8834 driver carrier https://www.hobbytronics.co.uk/drv8834-stepper-motor-driver. Both stepper motors on the mount (see Appendix) have 20Ω resistance on each phase, requiring 450mA of current when using a 9V supply. The DRV8834 can supply up to 1.5A per phase without requiring a heat sink or forced air flow.

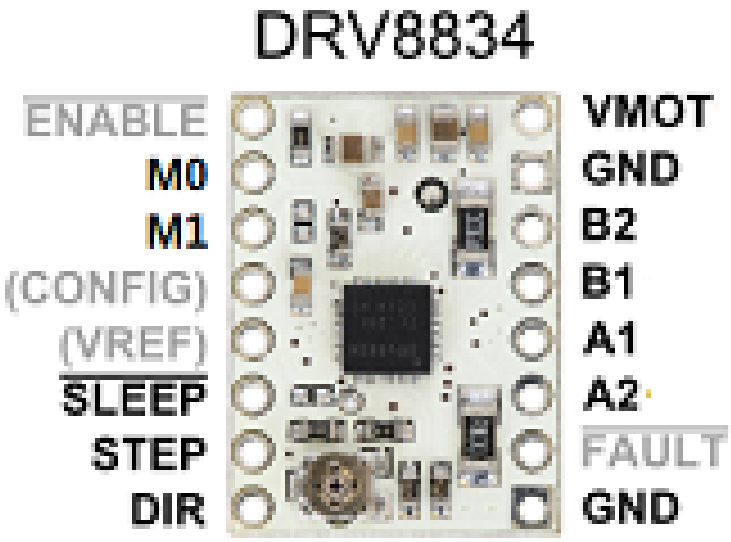

fig 4: DRV8834 Low Voltage Stepper Motor Driver Carrier

The pins with grey text are not required and are set by internal defaults. The M0 and M1 pins are used to set the stepping mode. For this project the stepping mode it set to ‘Full step’ (see Table 1).

Table 1

M0	M1	Microstep Resolution
Low	Low	Full step
High	Low	Half step
Floating	Low	1/4 step
Low	High	1/8 step
High	High	1/16 step
Floating	High	1/32 step

The full step sequence is shown in Diagram 1.

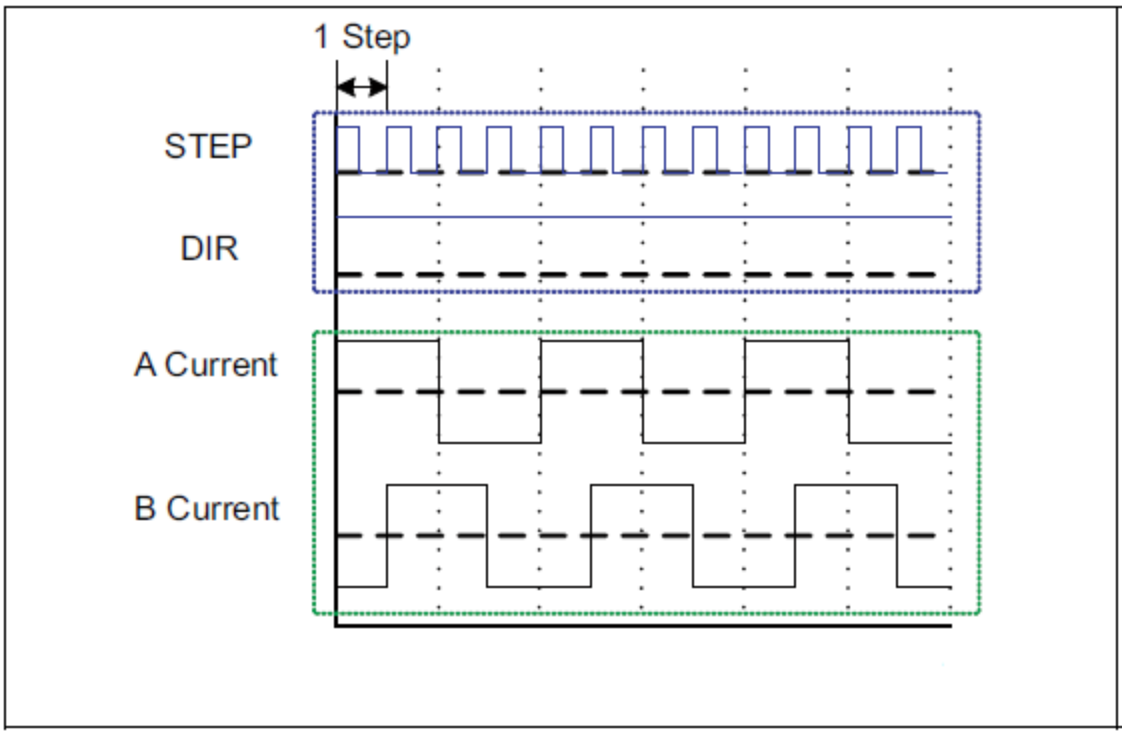

Diagram 1: Full Step sequence (from DRV8834 data sheet)

The step pulses are fed into the DRV8834 driver carrier **STEP** input from the timer counter outputs of the Arduino UNO. Other outputs are used to set the **SLEEP** and **DIR (**direction) control inputs. The **A1 A2 B1 B2** outputs from the DRV8834 drive the stepper motor. The Ardunio UNO is powered from the DC barrel plug and the recommend voltage range is 9V → 12V. The DRV8834 operating voltage is specified as 2.5V → 10.8V. A single power supply with a 9V output was selected to power the entire circuit (see Diagrams 2 and 3).

The RA step pulses will be driven using TIMER 1 on the ATmega328p which is a 16 bit timer. The required range of pulses (see **6. Timing**) for the DEC motor could not be obtained using TIMERS 0 and 2 individually as they have only 8 bit resolution. To overcome this, these timers are used in concert with the output of TIMER 2 fed into TIMER 0 clock input utilising the flexibility provided by direct register control.

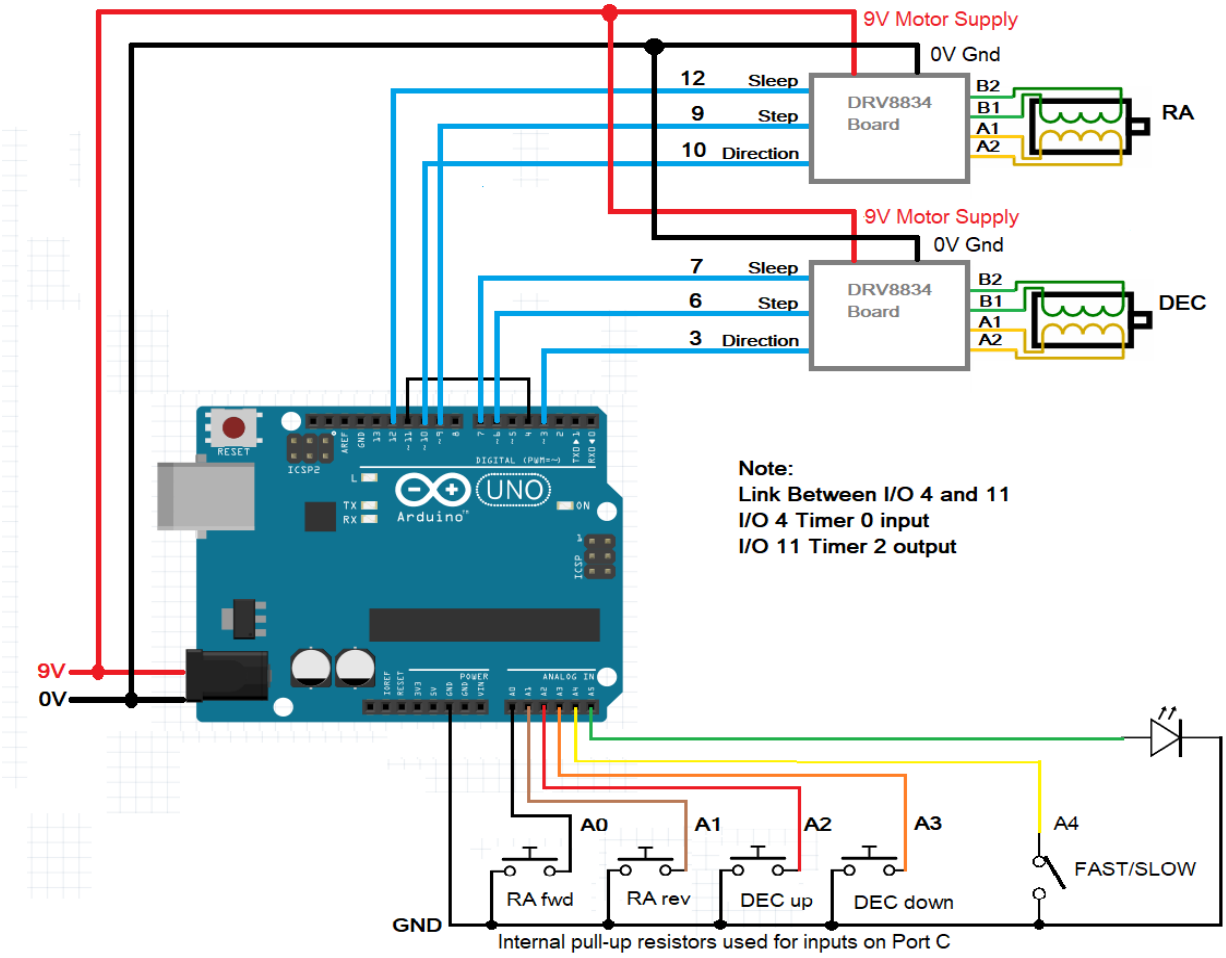

Diagram 2: full circuit

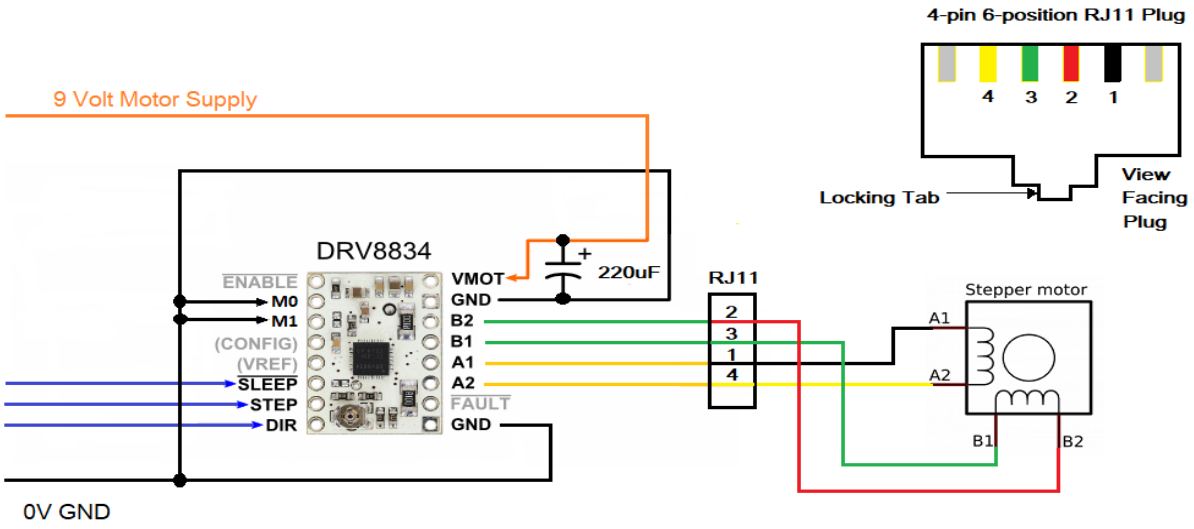

Diagram 3: DRV8834 board

5 Construction

The DRV8834 driver carriers and associated components were mounted on veroboard and the whole circuit placed in a standard plastic case (figures 4 and 5).

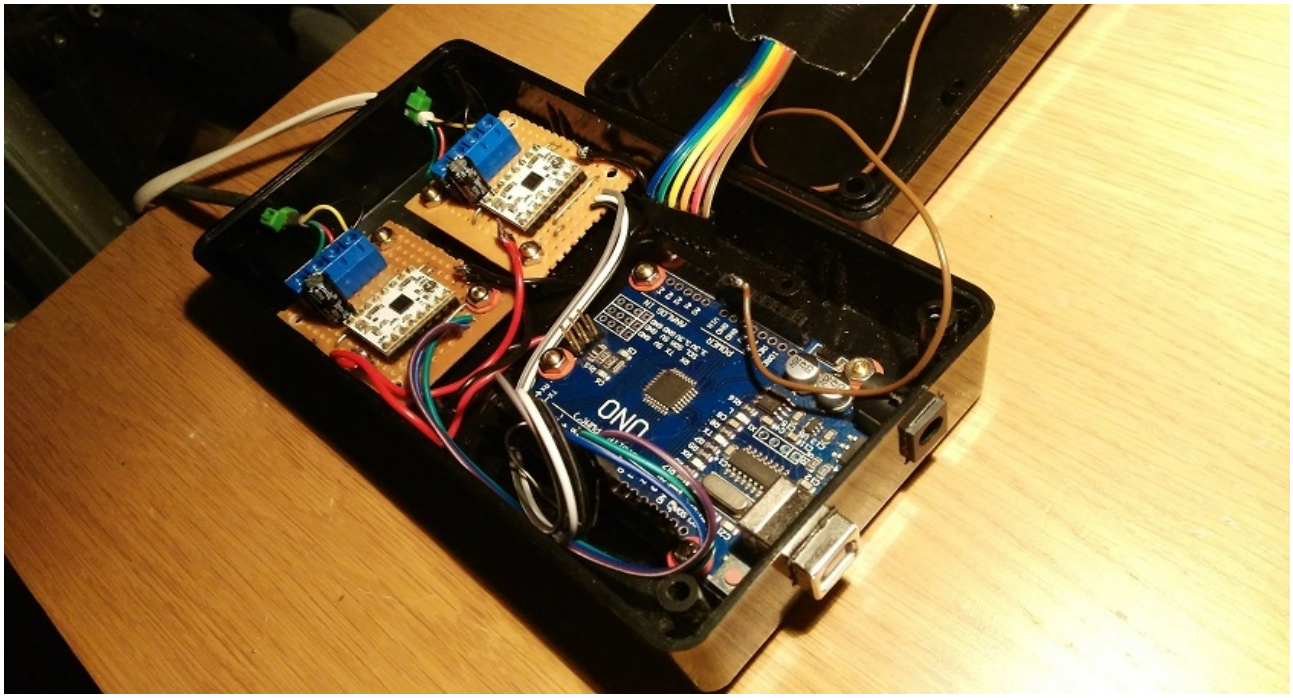

fig 4

fig 5

6 Timing

I found some information specifying that the RA stepper motor received a 400ms (2.5 Hz) pulse from the original G-4 #93522 driver for a sidereal rate. Examining Diagram 1, this would correspond to a 10Hz (100ms) step pulse input to the DRV8834 driver. However I discovered this was too fast. By experiment I established that a 110ms (9.09090...Hz) step pulse produced a 15º per hour rotation about the RA axis.

The DEC axis motor required an approx 3Hz step input to rotate at 15º hour, but accuracy is not required.

7 Initial Testing

RA tracking was tested by measuring the rotation about the RA axis over a period of two hours. The RA setting circle was allowed to rotate freely as if aligning. On this mount there is sufficient friction so that the freed setting circle rotates with the RA axis.


R. A.
15.1.2021 14:22


R. A.
15.1.2021 15:22

This initial test shows that the RA axis is being driven at the required rate of 15º per hour.

8 In Use

The RA and DEC axes were driven as specified in the requirements, the RA operating continuously at the 15º rate when the x8 forward/reverse was not selected. The DEC axis motor did have a tendency to initially stall for a few seconds before moving. This problem was traced to the power supply. The supply used was a variable DC 30V 2A overload protected set to 9 Volts. Examining diagram 1 it can be seen that the current fed into the A and B coils coincide at parts of the sequence, which corresponds to a 900mA pulsed load. When two motors (RA and DEC) are running, the maximum pulsed load will be 1800mA which the 2A power supply used struggled to meet. This problem can be corrected using a higher current rated DC supply or a 9V D cell pack.

9 Further Development

For a MK2 version an Arduino UNO will not be used, instead a stand alone ATmega328p programmed using a AVR programmer and the Atmel development system will be utilised. As register commands have already been extensively used, moving the software to the Atmel IDE should be straight forward.

Using a stand alone ATmega328p will free up some I/O lines which have reserved use or just not available on the Arduino. The MK2 version will still use a 16MHz clock and be powered by a single DC power supply.

The principle enhancements planned are;

a) Control the microstep resolution (see table 1) using the M0 and M1 inputs on the DRV8834 for the RA stepper motor. For tracking purposes the resolution could be increased e.g. ½ step ¼ step etc, and changed back to full step for fast mode,

b) Use an analogue input with a potentiometer for fine adjustment of tracking rate.

10 Appendix

RA Motor

DEC Motor