

A Barndoor Platform for Dobsonians

Henk Aling, April 20, Santa Barbara

Version 1.0

1 Introduction

The Dobsonian telescope offers the best value in terms of aperture. Some are motorized, which makes them very good for visual observing and even makes them capable of limited Alt/Az astrophotography. This feature adds a significant amount to the price though, and most Dobsonians in use today don't have it. I own a 35 year old 10" Coulter Odyssey Dobsonian. I picked it up at an astronomy fair of the local astronomy club for only \$100 along with a nice set of eyepieces and some other accessories. My experience with visual observing is limited, and I have always wanted to add a tracking capability to it that is good enough for modest astrophotography. We will show in this document how this can be done using the barndoor mechanism. While this is possible, keep in mind that cheap and simple home built contraptions such as this are generally less accurate than professional mounts so do not expect the same type of results.

1.1 Equatorial Platforms

The equatorial platform is the traditional tracking solution for non-tracking Dobsonians. It uses circular or elliptical sections on rollers along a virtual cone whose axis points at the celestial North Pole. The main purpose is to track accurately enough to keep an object in view for prolonged visual observing. There are very few examples of equatorial platforms that are good enough for auto-guided astrophotography. Here is a picture of a commercial high end system:



Figure 1: High-end commercial VNS-style EQ platform

The price of ready-built EQ platforms ranges from \$400 to \$2000. Kits start anywhere from \$200. So, price-wise you will be better off with either a non-Dobsonian system or with an Alt/Az motorized Dobsonian than with an EQ platform unless you build one yourself. For apertures of 16 inches and up however, a quality EQ platform has a real economic advantage compared to other types of tracking mounts. Equatorial mounts for large mirrors are extremely expensive.

1.2 Barndoor Trackers for Cameras

The barndoor tracker is a tracking solution for cameras for without the use of a telescope for wide field astrophotography. The mechanical parts consist of two boards connected with a hinge. A screw controls the angle between the boards. A motor drives the screw at the appropriate speed for a linear angular motion. This can be a circularly bent screw or a straight one if a trigonometric control correction factor is applied. The availability of cheap stepper motors and embedded computers makes this possible. See the pictures below for an example that I built on a rainy day for my Canon SX-40.



Figure 2: A traditional camera barndoor platform.

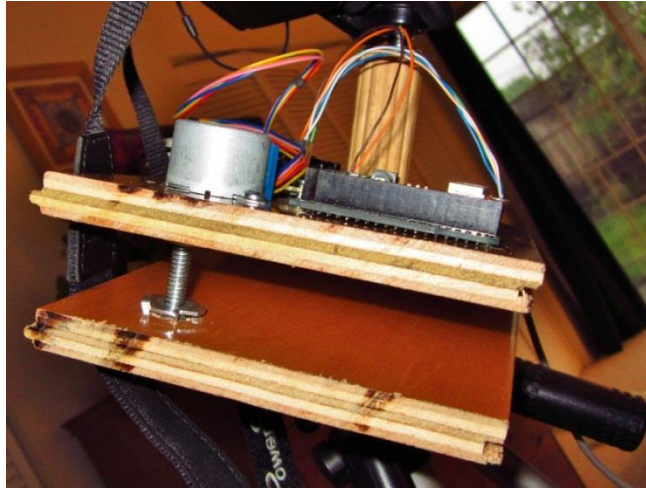


Figure 3: The motor supports the weight of the entire platform.

1.3 Barndoor Platforms for Dobsonians

The goal of this document is to show how to build a barndoor tracker for a Dobsonian. This is unusual and new but it can be done successfully and quite easily. The basic design consists of a box split in two parts connected by a hinge placed parallel to the equatorial axis. The pictures below show the basic design. The purpose of the box shape is to have a flat top to place the Dobsonian on top of, and also to provide sufficient stiffness because flexibility is bad for tracking.



Figure 4: Barndoor box with manual crank (this crank was never used and was removed).



Figure 5: Barndoor box showing the hinges pointing at the celestial North Pole.

The main differences between the barndoor tracker and the equatorial platform are the following:

- A barndoor platform is much simpler and easier to construct than an equatorial platform. It uses a hinge instead of sectors on rollers like with an equatorial platform. Though the use of sectors and rollers seems simple, many builders have difficulty determining the right shape from spreadsheets or mathematical formulas. Manual cutting often results in inaccuracies that are not noticed by the builder. Using a jig is the best way to avoid math and inaccuracies but building a good jig is not easy. A barndoor construction can be slapped together from scrap boards and a hinge in a matter of hours without having to worry about all that. Also, because of its simplicity, it is easier to detect error sources and to tune the design for accurate tracking.
- Equatorial platforms are designed such that the center of mass is close to the axis of the virtual cone, which implies that it is displaced minimally during tracking. Because of that, only a small motor is required that uses very little power. The barndoor tracker moves the center of mass in a bad way. This must be mitigated by an added-on counterforce mechanism for instance by hooking up a bungee cord. This is decidedly less elegant and convenient.
- Equatorial platforms are smaller. Barndoor trackers are generally bulky and heavy. Their size depends on the latitude. An incidental advantage of such a box however is that it elevates the Dobsonian to a convenient height that makes viewing a lot easier.
- An equatorial platform can be reset easily by a single pull. To reset a barndoor tracker, a screw must be wound back to its original position.
- Because the size of the box of a barndoor design depends on the latitude, the range of latitudes for which it is suitable is narrower than that of equatorial platforms. For the latter, different types of designs are available each of which work best for a certain latitude range.

After I designed and built the first version I read about the Hiss drive by Tom Fangrow. It looks quite similar to my design. However, it does not have levers and uses a deflating inner tube for tracking, which makes it useless for astrophotography. My barndoor design uses a stepper motor for precise tracking (two steppers for the dual axis configuration).

2 The Mechanical Components

We will illustrate how the barndoor tracker is built by going through the setup procedure for an auto-guided observation session. We will look at the individual features as we go along.

2.1 The Barndoor Construction

The initial barndoor design was shown above. The final design has some additional features as shown in the picture below.

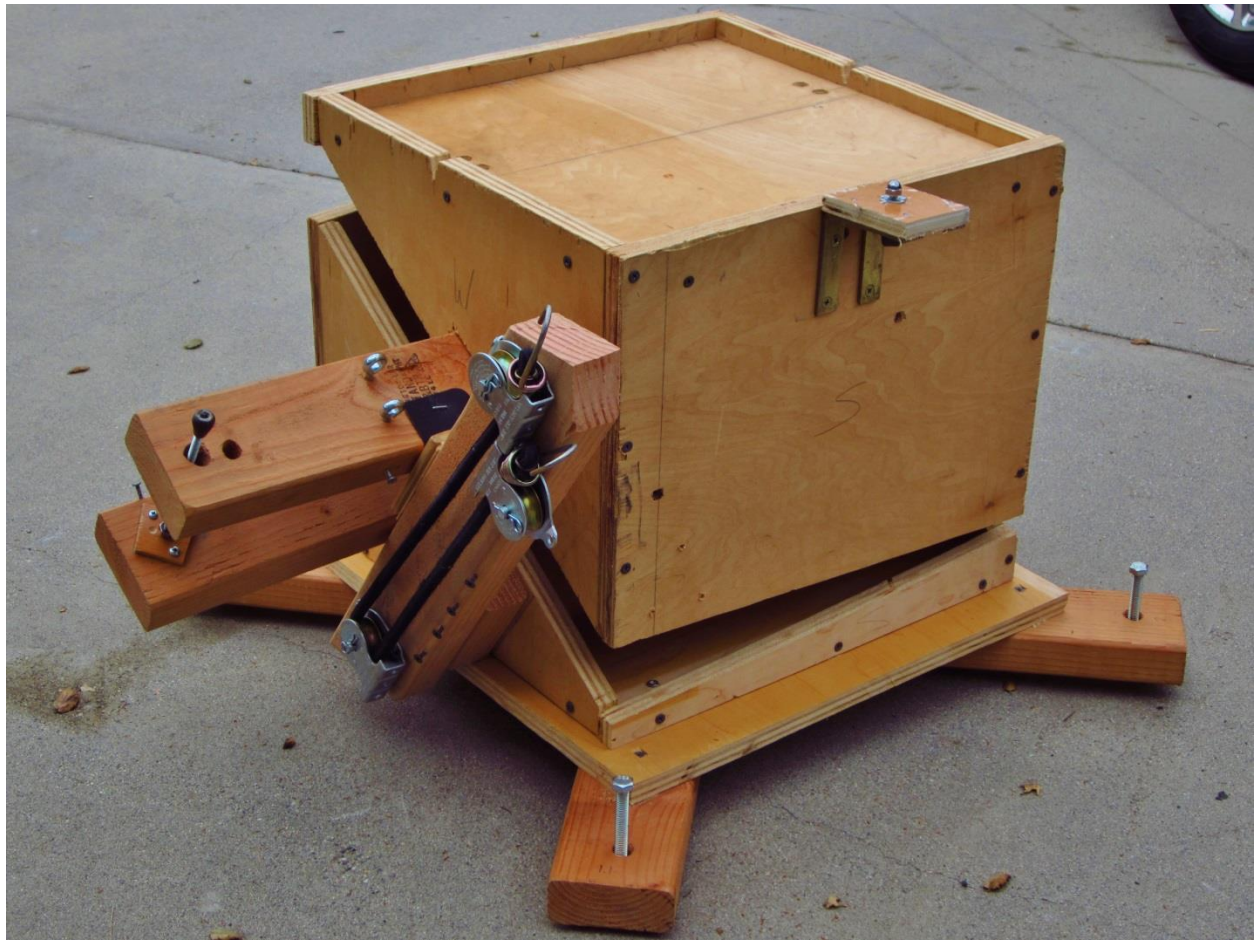


Figure 6: Basic barndoor tracker with support slider and bungee cord holder.

The main mechanical design features are as follows:

- The hinge axis is parallel to the equatorial axis, which is 34 degrees in this case.
- The top and bottom parts of the wedge do not constitute a complete box. A wedge shaped portion is missing. The tip of this wedge is on the hinge side. The gap between the parts on the other side is about 2 inches.
- The levers of the barndoor mechanism consist of two 2x4s that are attached to the top and bottom parts. How this was done is seen best in one of the pictures below where the box is opened.

- A screw keeps the two beams separated and slowly lowers the top as it turns counterclockwise. This means that the total weight of the top and the Dobsonian rests on it.
- The top side beam rests on a support slider. This reduces the warping of the box under its heavy load compared to when only the hinges would hold the top in place. It increases the stiffness and reduces the vibrations significantly.
- The support slider doubles as a bungee cord holder. The bungee provides a counterforce against the weight parallel to the direction of motion.
- A base was added consisting of a board with four pieces of 2x4 as feet. This board could have replaced the bottom board of the bottom section of the box to reduce weight although this was not done here.

2.2 Polar Alignment

The first step of the setup procedure is polar alignment. First we level the base. This is done using a bidirectional level by adjusting the screws in the feet. The box has to be opened up because the level is attached to the bottom.

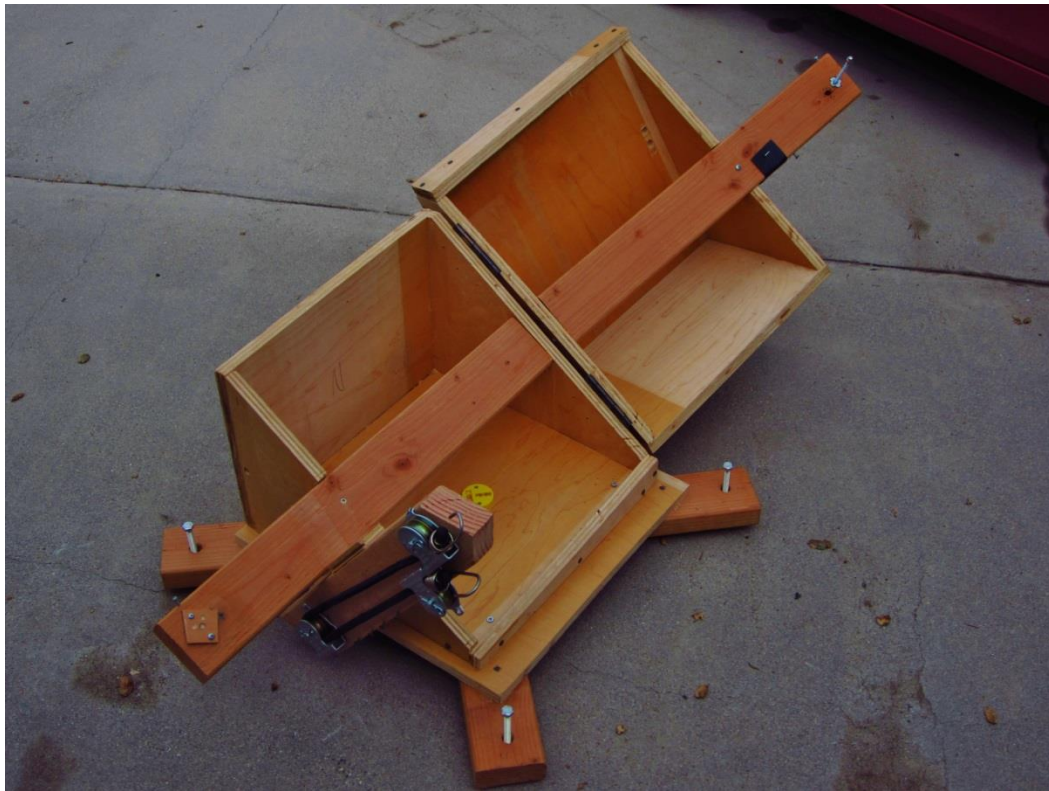


Figure 7: The barndoor tracker opened up with the yellow level visible.



Figure 8: Bidirectional level attached to the base.

The second polar alignment step is to align the hinge board with Polaris. This is done by pressing an alignment plank against the lower hinge board, lying on the ground and shifting the box until the plank points at Polaris. The top of the plank is cut off at the latitude angle. Afterwards we need to re-check if the box is still level and correct if needed.



Figure 9: The alignment plank is pressed against the hinge board to help with polar alignment.

2.3 The Screw Mechanism

After polar alignment the box is closed. The top lever then rests on a T nut that the screw runs through. The stepper motor axis will be placed in a fitting hole in a glob of epoxy on top of the screw. This hole was shaped by pressing the axis into the epoxy while it was hardening. The motor will lower the top section by turning the screw counterclockwise.

Note that the weight will rest on the T nut and not on the stepper. This is different from the case of the camera barndoor drive that was shown earlier. In that case the screw carried the weight of the top part including camera and turned clockwise to lower it. The weight of the Dobsonian does not allow this.

One matter that deserves attention is the fact that the screw angle changes as the top section is lowered while the motor turns. The problem is that the screw is a critical factor in the tracking accuracy. The screw has a cap nut at the bottom that rotates in a dimple drilled into a piece of floorboard. The screw is designed to be perpendicular to the top lever throughout the trajectory. The dimple must be straight under the hole in the lever under full load. Its location must therefore be determined with the Dobsonian installed. Note the two little screws in the top and bottom lever. The controller has a slot that fits around them and helps position the controller mechanically parallel to the screw. See the pictures below.



Figure 10: The screw supports the top lever through a loosely attached T nut.



Figure 11: The screw rotates in a dimple in a small piece of floorboard. A glob of epoxy around the top has a hole in the shape of the stepper motor axis that was pressed into it.

Anything that causes the screw angle to change from its intended position will result in a torque on both the cap nut (so it may wiggle out of the dimple) and/or the T nut (which gets transferred to the

telescope and will be visible for astrophotography). Any factors that can cause this must be eliminated as best as possible. For instance:

- The screw used is $\frac{1}{4}$ inch thick and has 20 TPI. It may buckle after repeated use.
- The cap nut may not be centered on the screw as a result of manufacturing errors.
- The hole in the epoxy in which the stepper motor axis is lodged may be off-center.
- The T nut and screw are cheaply made and do not lock very tightly. This means that the weight bearing contact location may turn along with the screw.
- The screw may not be greased well or may have grains of sand in it.

At the time of the writing of this document the autoguider still shows periodic errors at the same frequency as the screw turns. This means that the screw must be realigned by somehow fixing any of the causes above. Do not underestimate these issues. Also realize that they can be fixed easily one way or another because of the simplicity of the system. The main problem is to recognize them.

2.4 The Support Beam

The support beam simply consists of a piece of floorboard that it slides on. Special features are:

- A piece of binder plastic was stapled on to the top lever to reduce the stiction that occurred without it. Otherwise the sliding will have a noticeably bumpy behavior.
- The support beam is offset from the box with a piece of board in between because the top section extends over the bottom section at the lower part of the screw trajectory. Without this offset the top section would bump into the support beam.
- The position of the support beam must be chosen such that the top lever has continuous support along the trajectory and can move smoothly - not too tight or too loose.



Figure 12: The support slider with anti-stiction binder plastic.

2.5 The Bungee Cord

The weight of the Dobsonian would stop the stepper motor dead in its tracks without some form of mechanical compensation. This was done in the form of a super-bungee wound around some pulleys as shown below.

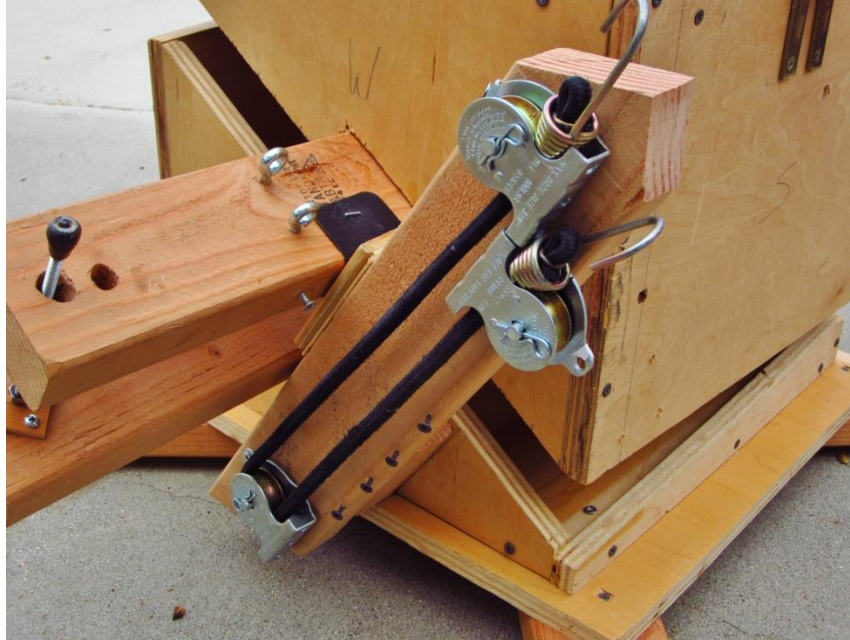


Figure 13: The support beam doubles as a super-bungee holder.

Special features:

- A choice of two eye hooks and a screw at the top lever to hook into for tension direction.
- A choice of four screws on the support beam to hook into for tension force.
- At the start of each run the bungee needs a jerk because of the friction in the pulleys. The goal of this jerk is to relieve the tension at the beginning. Note that the force is the largest when the T nut hits the bottom, this is why we need the jerk when it is at the top or else the stepper will stall.

2.6 The Optional Declination Axis

The mechanics discussed so far support a motion in Right Ascension (RA). Autoguiders generally work in two directions, RA and Declination (DEC). This is why the declination axis was added.

The declination axis is implemented as an optional add-on board as shown below. It sits on a steel brace and has holes where the Dobsonian base has any protrusions (three feet and a screw) to keep it in place. That way we can turn the Dobsonian without backlash. The position of the brace was chosen such that the board can be moved with little force. In fact in some positions it tilts the other way but it can be kept in place with a bungee cord. The board sticks out over the side where the DEC controller will push up against it from the brace mounted on the bottom section. The amount of motion can be very small so we don't need a long screw.



Figure 14: The declination board positioned at a place where it causes a narrow balance margin.

The declination control screw shown below sits upside down and has a similar glob of epoxy that the declination motor axis fits into. It is pulled up from the declination board because that distance is supposed to remain the same on average.

By this time you may have noticed several rubber bands. These are all cut out from an old road bike inner tube. That kind of rubber is fantastic because it does not dry out and does not tear either.



Figure 15: The declination control screw with a rubber pull-up band for the declination controller.

To keep the declination board from tipping while we are aiming the OTA at a target we push a little wedge shaped piece of 2x4 in between the declination board and the top section of the barndoor tracker. See below.



Figure 16: A wedge shaped piece of wood keeps the declination board from tipping while we are aiming at a target.

2.7 Telescope and Accessories

The telescope is mounted next as shown below. Note the purple bungee that was added to keep it from tilting backwards. Maybe if I moved the steel brace on the declination board back a bit this would not be necessary.



Figure 17: The telescope mounted on the barndoor tracker.

It is well known that Newtonians can have a so-called back-focus problem. This means that the focuser may protrude too far out to achieve focus with a DSLR mounted on the focuser. Indeed this was the case. Being practical as always, I removed the focuser altogether. I replaced it with a piece of ABS pipe and the T-ring that doubles as a helical focuser over a very limited distance. The DSLR, a Canon Rebel XT, is clamped on to the telescope using rubber bands. This has worked fine so far. At some point I need to check how to collimate this properly – sticking a DSLR onto a telescope with screws lodged into the ABS pipe in cardboard holes is not the ideal fine-tuning environment needed for astrophotography. But hey it works. The downside is that after I aimed using the Telrad, I have to find the target looking through the DSLR eyepiece. This is somewhat frustrating but without a goto system, this is what every astrophotographer will eventually have to do. Fine focus is achieved using a Bahtinov mask – it cannot be done using the human eye in any way.



Figure 18: Rebel XT attached with rubber bands next to the Telrad.

Next we mount the guide scope, see the picture below. The guide scope is optional. It can be used with or without the declination board because PHD has an option to guide for RA only.



Figure 19: The guide scope mounted near the OTA center of mass.



Figure 20: Rear view of the Dobsonian mounted on the barndoor tracker.

3 The Electronic and Software Components

Several hobbyists will boast how easy it is to put controllers together but this is really the hard part especially if you start from scratch like me. Luckily we live in an age where hobbyists have created their own space in the embedded world in the form of the Arduino. For simple embedded control tasks this is truly a Swiss army knife. It is perfect for implementing single-threaded control algorithms on cheap 8-bit hardware.

3.1 The Controllers

The term “controller” is shorthand for the assembled parts consisting of an Arduino, the stepper motor controller board and the stepper motor. You can see them in the pictures below sitting on the back of my car. The RA controller has the parts on the outside of the floorboard frame. The DEC controller has them on the inside. Note the slot in the RA controller board that fits tightly around the two positioning screws mentioned earlier. Note also how the stepper motor axes protrude through a round hole.

The RA controller stays in place by gravity. Backlash is not a problem once the motor is engaged because it only moves in one direction. The DEC controller is held in place by a rubber band pulling it up into the screw socket. There are two backlash sources: the socket in the epoxy blob, and any play in the

controller unit hanging down from the rubber band. The latter can be minimized by clamping it against the barndoor with a rubber band, if needed. Usually gravity keeps it in place but it is something to look out for.



Figure 21: RA and DEC controllers.

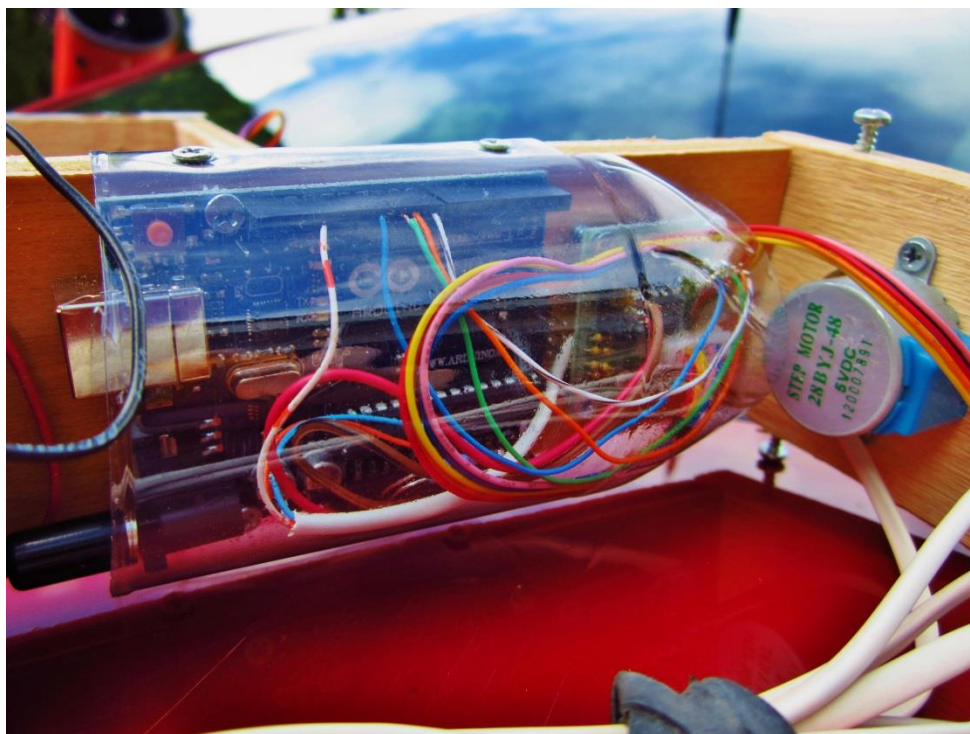


Figure 22: DEC controller detail.

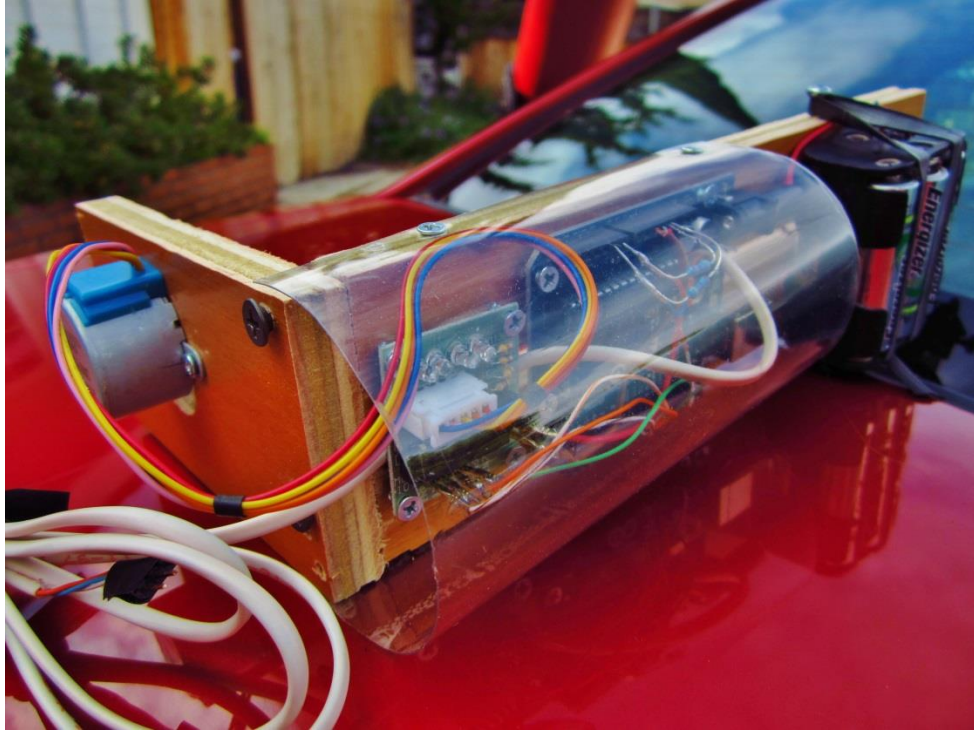


Figure 23: RA controller detail.

3.2 The Arduino Computer

The Arduino is a wonderful 8-bit hobby computer that exists in all sorts of varieties. They have both digital and analog inputs and outputs, and serial-over-USB for downloading code and debug output on a serial console. No knowledge is needed about an embedded OS because there is none – well whatever there is is hidden from the user. This means that it is extremely easy to use for simple tasks. The user interface is also very friendly. A simple pushbutton builds the executable and flashes the Arduino. Soldering is not needed when using the right gauge of wire. They can simply be stuck straight into the sockets, folded double when needed. The Arduino computer used here is the Arduino Uno R2. It was available for \$14 a piece while it was phased out!

3.3 The Stepper Motor

The motor is a 28BYJ-48 four phase stepper motor controlled by a ULN2003 controller board. It has an internal gear reduction by a factor 64 using a measly 500 mW. One internal revolution takes 64 steps. Thus, one external revolution takes 4096 steps. The torque is enough to turn the screw if the bungee cord relieves the load. Being able to move about 65 lbs. of weight using a 500 mW motor is an amazing feat! The Arduino and 28BYJ stepper motor combined runs for 6 to 8 hours on a battery pack of 6 rechargeable AAs.

3.4 The Orion SSAG Mini-guider ST-4 Signals

The Orion SSAG Mini-guider consists of a small telescope and a sensor/controller. The sensor sends images to PHD on the laptop via USB. PHD sends a control signal back to the controller via USB. The controller converts this to resistance signals via an optocoupler. These so-called ST-4 signals can easily

be converted to a voltage signal with the Arduino. Namely, by placing it in series with a 500 Ohm resistor in between the ground and 3.3 Volt pin. The plus side of this voltage is then fed into a 0-5 Volt analog pin. To be precise, the resistor is connected from the 3.3 Volt pin to the analog input that is connected to the ST-4 signal. The 10-bit AD conversion then shows a drop from more than 700 to around 40 when a pulse comes in and the optocoupler resistance is minimal. A picture taken during testing is shown below. There are four ST-4 outputs total: DEC and RA forward and backward. The SSAG manual has the details.

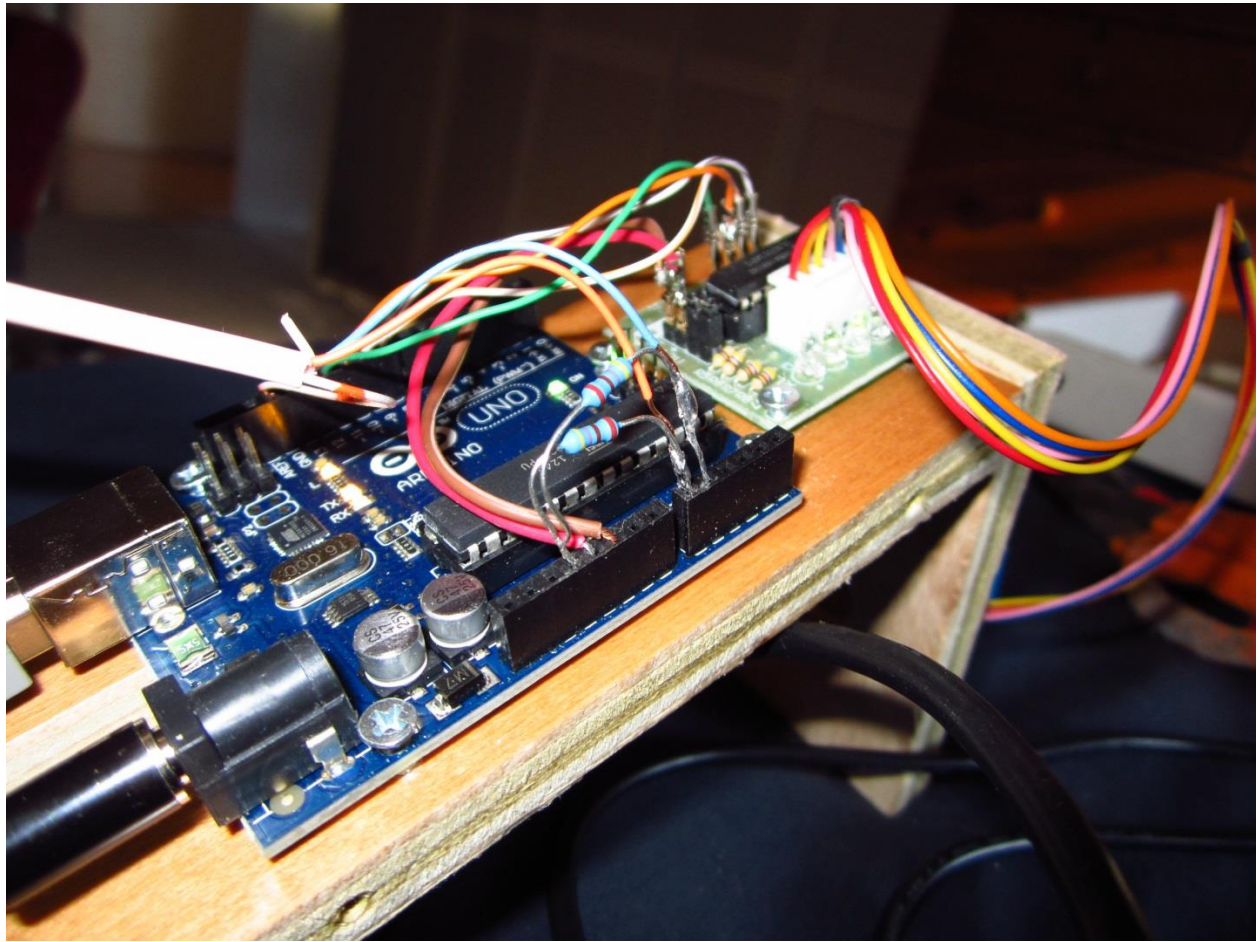


Figure 24: The ST-4 signals of the RA controller are connected to A0 and A1 that are each connected to the 3.3 Volt pin through a 500 Ohm resistor.

The final system with all controllers hooked up is shown below.

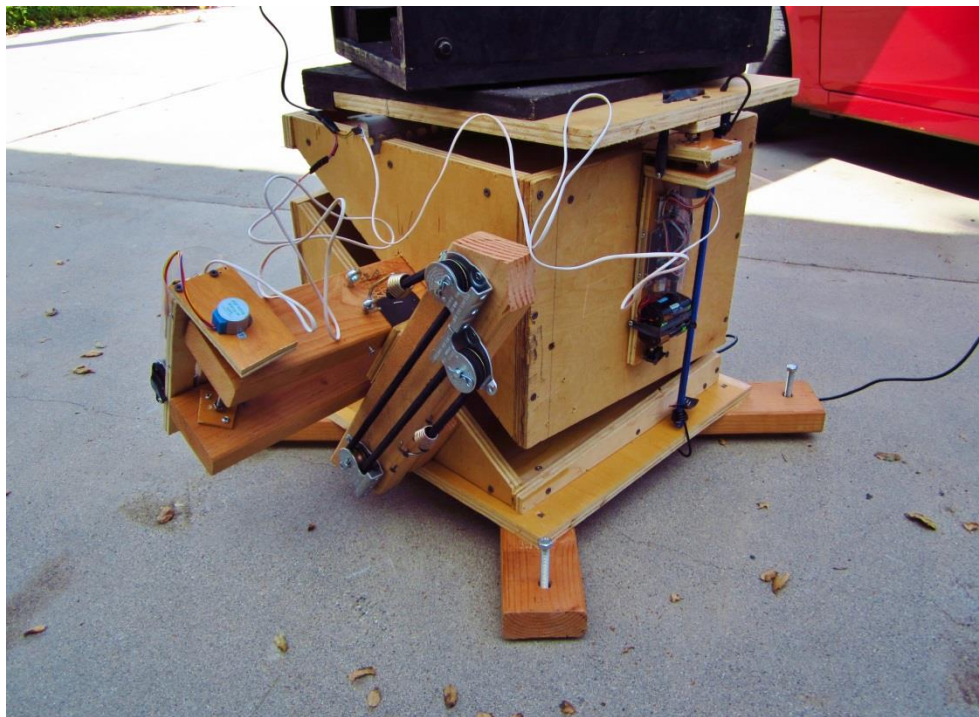


Figure 25: Barndoor tracker with everything hooked up.

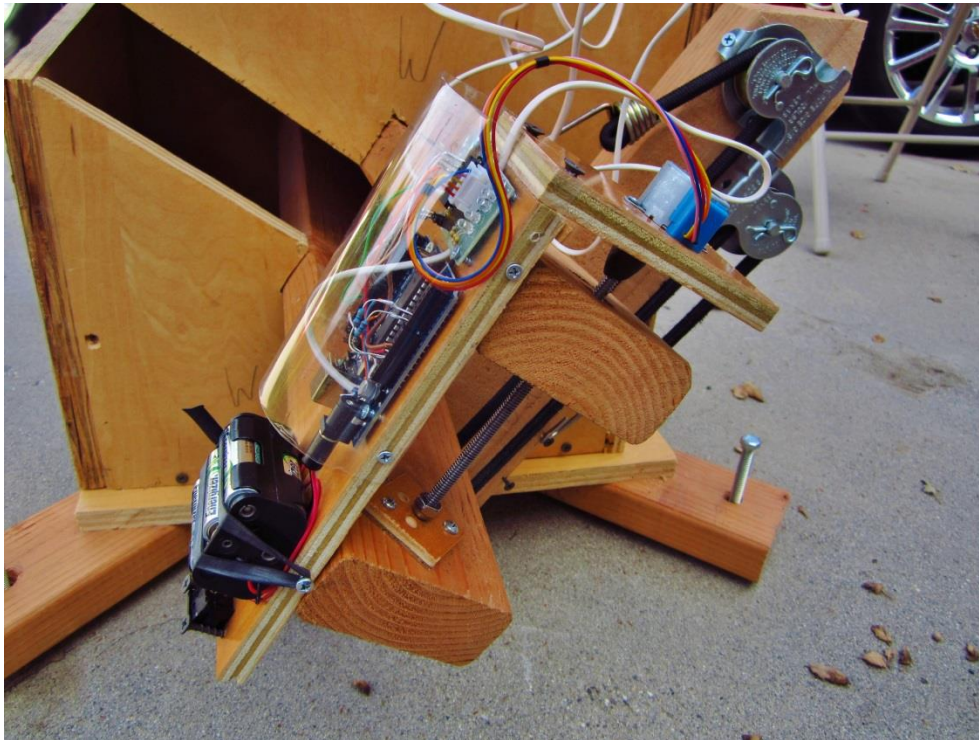


Figure 26: RA controller hooked up.

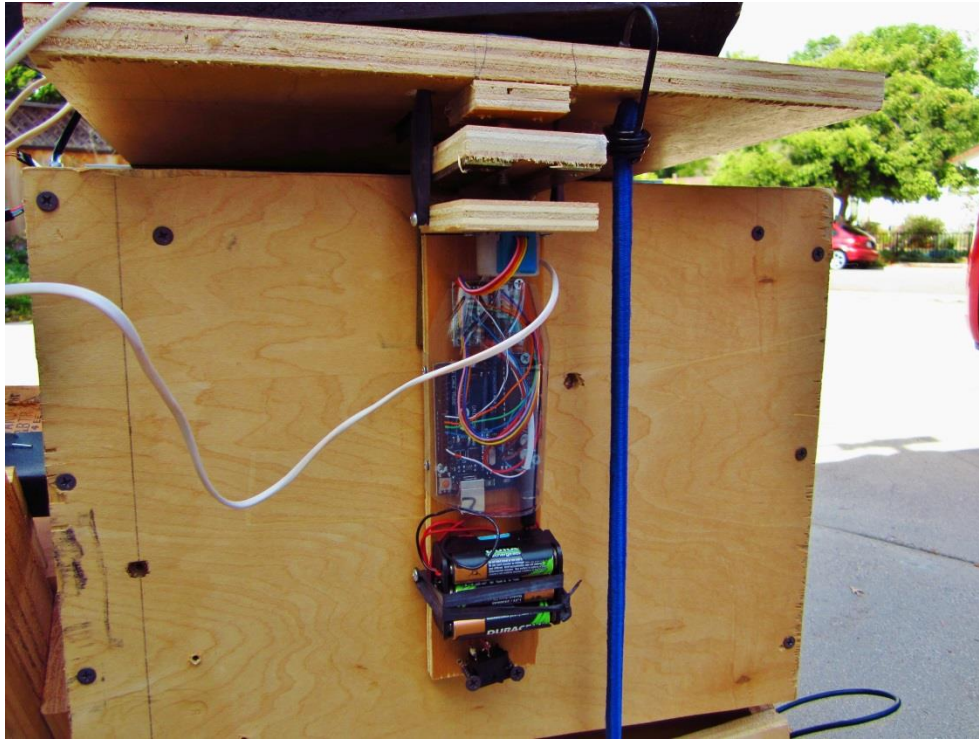


Figure 27: DEC controller hooked up.

3.5 Software and Control

The RA controller program has the following features:

- Digital output pins 8-11 are configured for wires 1-4 (blue, pink, yellow, orange).
- Computation of basic constants for the conversion from the earth's angular rotation to pulse period in seconds.
- A table of 8 sequences of HIGH/LOW pulse values for each of the four pins. The program cycles through this table, one sequence for each step.
- The timeline is used as a basis for control. A delay is issued for each pulse until the pre-computed time step. By doing this, any cumulative errors caused by computational delays are avoided.
- A trigonometric correction factor is implemented to account for the changing screw angle along the trajectory.
- A speedup/slowdown factor is applied for pins 0/1 when an ST-4 pulse is registered.
- An optional debug feature produces a character plot on the serial console.

The DEC controller program is a modified version of the RA one. It has the following features:

- Digital output pins 8-11 are configured for wires 1-4 (blue, pink, yellow, orange).
- Computation of basic constants for the conversion from the earth's angular rotation to pulse period in seconds.

- A table of 8 sequences of HIGH/LOW pulse values for each of the four pins. The program cycles through this table, one sequence for each step.
- The timeline is used as a basis for control. A delay is issued for each pulse until the pre-computed time step. By doing this, any cumulative errors caused by computational delays are avoided.
- A speedup/slowdown factor is applied for pins 0/1 when an ST-4 pulse is registered.
- An optional debug feature produces a character plot on the serial console.
- An incremental value of -1/1 is used to determine in which direction the motor turns when a pulse is registered on pins 0/1.
- The same slowdown/speedup factor is applied in both directions.

The Arduino software source code is listed in the appendix.

PHD applies a calibration before it starts controlling the system. It senses the control action and backlash of the controller. This means that the controlled system is somewhat insensitive to the gains that were set in the controller software. This does not mean that we can pick any value. The best strategy is to spend some time testing with various gain values while letting PHD calibrate each time. This can be done for RA only to get that one right first (the DEC action can be turned off with PHD). Once the optimal RA gain is determined we can do the same for DEC. Currently,

- The RA action is sped up by decreasing the per-pulse delay by a factor 0.5.
- The RA action is slowed down by increasing the per-pulse delay by a factor 1.5.
- The DEC per-pulse delay is set to 2 times the nominal RA per-pulse delay.

4 Results

At this point I have tried the autoguided dual axis configuration only once due to circumstances. The result was very encouraging though, and more and better results will follow. Here is a picture of the laptop screen with the PHD graph during an 11 minute and 29 seconds exposure at ISO 100 and the corresponding image:

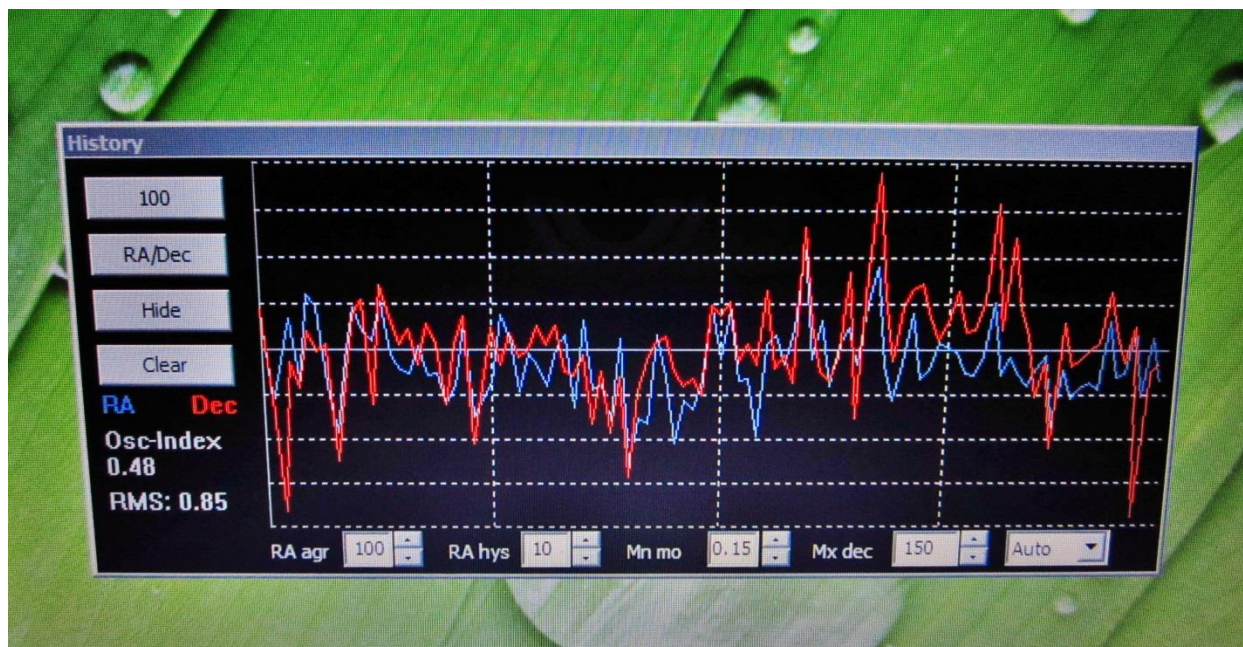


Figure 28: PHD graph during the first trial.



Figure 29: Nearly round stars with M101 in the upper right corner. The exposure was 11 minutes and 29 seconds at ISO 100 and was taken under poor conditions.

Unfortunately the seeing was not very good due to a full moon and humid air so there was a lot of reflected and dispersed light. However, the stars are nearly round, which is simply unthinkable without autoguiding. Usually after 30 seconds the star trails are already much larger than here. So, this is a resounding success especially for a Dobsonian considering what equipment we started out with.

From earlier single axis (RA only) trials there are a few more results. Here is a 30 second exposure at 400 ISO of the Orion nebula:



Figure 30: A single 30 sec ISO 400 shot of Orion with the Dobsonian on the barndoor box.

True, this is an easy target but still nice to see a decent result with rich colors and a fair amount of detail. In addition to my 10.1" F/4.5 Odyssey I also tried a simple department store telescope that I bought in the store for about \$100. Some of the exposures were 140 seconds, most were 30. The result is quite remarkable for such a cheap scope:

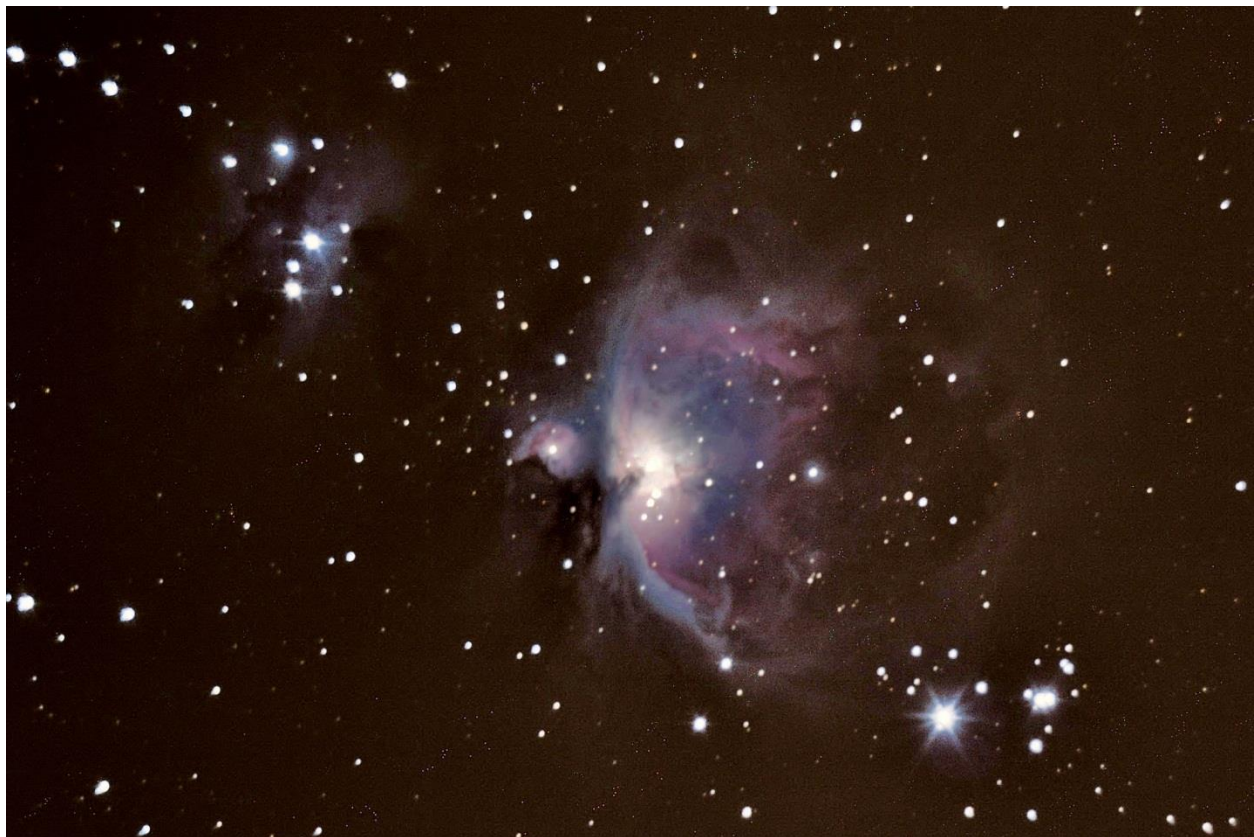


Figure 31: A stack of 26 images taken with my Venture RX-7 on the barndoor tracker.

In the introduction I showed a picture of a small barndoor tracker for my Canon SX-40 point-and-shoot camera. Below is a 256 second exposure at ISO 1600 of the Milky Way. It has the Andromeda galaxy at the lower left, the Pleiades at the upper left, the California nebula above the center, the Double Cluster near the center and Capella at the upper right. The blue glow from above is caused by Jupiter.

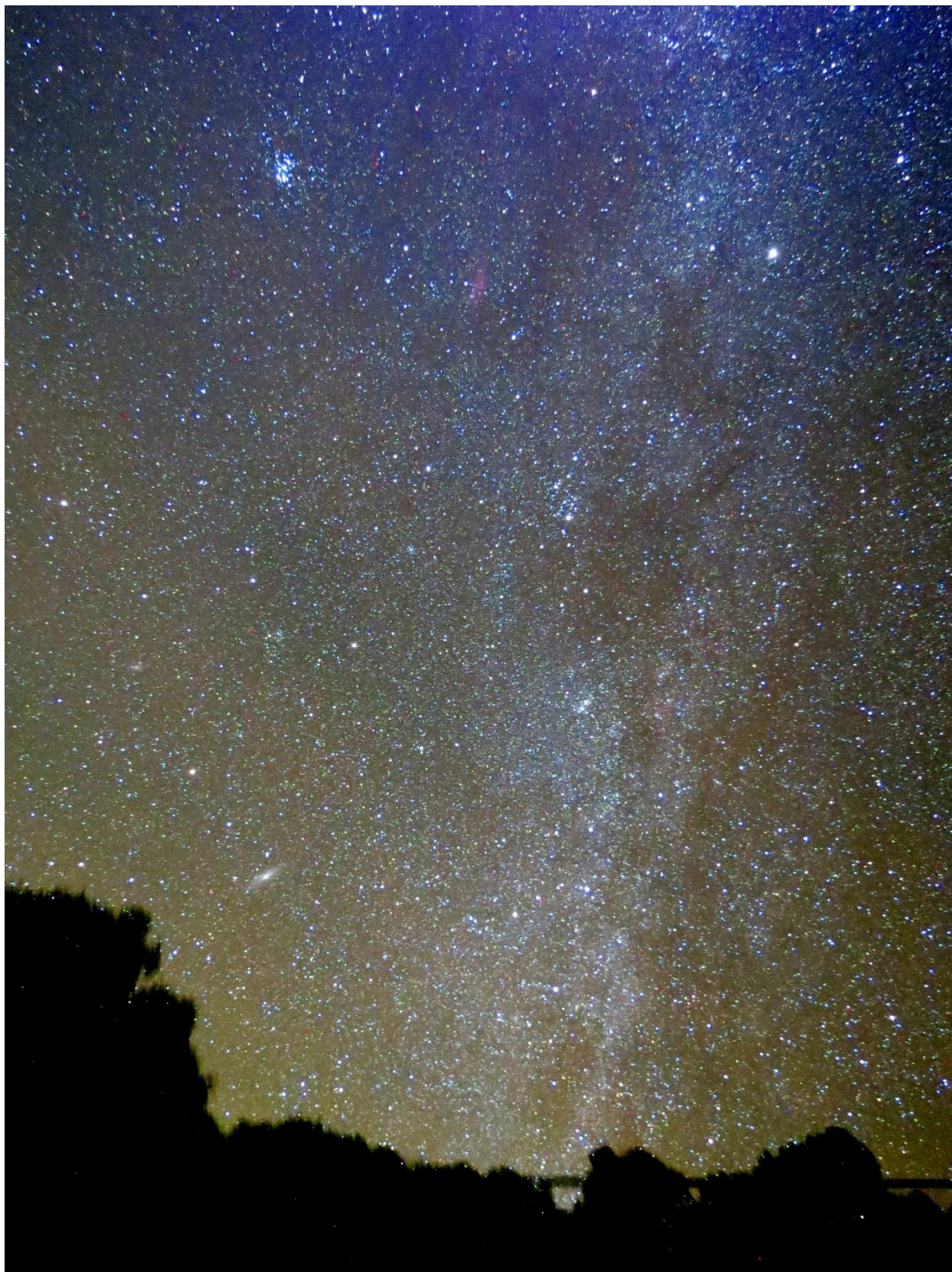


Figure 32: Wide field with the barndoor camera, 256 seconds at ISO 1600 with my SX-40.

5 Controller Source Code

The source code below has several debug statements that can be left out and may not be relevant anymore. It is presented as is.

5.1 RA Controller

```
////////////////////////////////////
// Simple character bar plot that shows x on a scale of xmin to xmax from
// left to right using a marker character and a | to indicate xmax.
////////////////////////////////////
void plotData(double xmin, double xmax, double x, char marker)
{
    char buf[80];
    int i;
    if ((x >= xmin) && (x <= xmax)) {
        for (i = 0; i < 78; i++) {
            if (double(i)/78 >= (x - xmin)/(xmax - xmin))
                buf[i] = ' ';
            else
                buf[i] = marker;
        }
        buf[78] = '|';
        buf[79] = char(0);
        Serial.println(buf);
    } else {
        Serial.print(marker);
        Serial.print(" is out of range, value = ");
        Serial.println(x);
    }
}

// Basic constants
double pi = 4*atan(1.0);           // Pi = 3.1415927
double inch2cm = 2.54;             // One inch is 2.54 cm
double secSidDay = 86164.1;        // Seconds in a siderial year
double eRot = 2*pi/secSidDay;      // Earth's rotation in radians per sec

// Barndoor geometry (RA)
double tpi = 20;                   // Threads per inch of the screw
double beam = 65.7;                // Hinge to T-nut/cap nut in cm
double y0 = 9.0;                   // Initial vertical T-nut position in cm
double y1 = 1.0;                   // Final vertical T-nut position in cm

// Motor characteristics
int ppir = 64;                     // Pulses per internal rev (360/5.625)
int ppr = 64*ppir;                 // Pulses per revolution (gear reduction factor = 64)
double cmpr = inch2cm/tpi;         // cm per revolution (1 thread of the screw)
double phipr = cmpr/beam;          // Angle (phi) difference per revolution (in radians)
double dtpr = phipr/eRot;          // The time it takes the sky to rotate that much
double dtp = dtpr/ppr;             // Pulse time in sec
int motorPin[4] = { 8, 9, 10, 11}; // blue=1, pink=2, yellow=3, orange=4

int ccw[8][4] =                   // counter-clockwise
{
```

[illegible]

```

// We don't need this loop, let's write our own for better control
int nRev = (np - npMin)/512;
while (ip + npMin <= np) {
    int ic = ip % 8; // Internal cycle index
    double correction = cos((phi0*(np - ip))/(2*np));
    double dt1;
    double dt = dtpp/correction;
    int a0 = analogRead(0); // faster
    int a1 = analogRead(1); // slower
    if (DEBUG == 1) {
        Serial.print(a0); Serial.print(' '); Serial.println(a1);
    }
    if (a0 < 400) { // faster
        dt = 0.5*dt;
    }
    if (a1 < 400) { // slower
        dt = 1.5*dt;
    }
    for (int j = 0; j < 4; j++)
        digitalWrite(motorPin[j], ccw[ic][j]);
    t = t + 1000000*dt;
    dt1 = t - micros();
    if (dt1 > 0)
        delayMicroseconds(dt1);
    ip = ip + 1;

    if (DEBUG == 1) {
        if (ip % 512 == 0) { // Each revolution has 8*64 pulses
            double tmp = micros();
            int iRev = ip/512;
            plotData(0, nRev, nRev-iRev, 'I');
            plotData(27400000, 27600000, tmp - t0, 'A');
            plotData(27400000, 27600000, t - t0, 'R');
        }
    }
}
}

```

5.2 DEC Controller

```

/////////////////////////////////////////////////////////////////
// Simple character bar plot that shows x on a scale of xmin to xmax from
// left to right using a marker character and a | to indicate xmax.
/////////////////////////////////////////////////////////////////
void plotData(double xmin, double xmax, double x, char marker)
{
    char buf[80];
    int i;
    if ((x >= xmin) && (x <= xmax)) {
        for (i = 0; i < 78; i++) {
            if (double(i)/78 >= (x - xmin)/(xmax - xmin))
                buf[i] = ' ';
            else
                buf[i] = marker;
        }
        buf[78] = '|';
    }
}

```

```

        buf[79] = char(0);
        Serial.println(buf);
    } else {
        Serial.print(marker);
        Serial.print(" is out of range, value = ");
        Serial.println(x);
    }
}

// Basic constants
double pi = 4*atan(1.0);          // Pi = 3.1415927
double inch2cm = 2.54;           // One inch is 2.54 cm
double secSidDay = 86164.1;      // Seconds in a siderial year
double eRot = 2*pi/secSidDay;    // Earth's rotation in radians per sec

// Barndoor geometry (RA)
double tpi = 20;                 // Threads per inch of the screw
double beam = 65.7;             // Hinge to T-nut/cap nut in cm
double y0 = 9.0;                // Initial vertical T-nut position in cm
double y1 = 1.0;                // Final vertical T-nut position in cm

// Motor characteristics
int ppir = 64;                   // Pulses per internal rev (360/5.625)
int ppr = 64*ppir;               // Pulses per revolution (gear reduction factor = 64)
double cmpr = inch2cm/tpi;       // cm per revolution (1 thread of the screw)
double phipr = cmpr/beam;        // Angle (phi) difference per revolution (in radians)
double dtpr = phipr/eRot;        // The time it takes the sky to rotate that much
double dtp = dtpr/ppr;           // Pulse time in sec
int motorPin[4] = { 8, 9, 10, 11}; // blue=1, pink=2, yellow=3, orange=4

int ccw[8][4] = // counter-clockwise
{
    {HIGH, LOW, LOW, LOW},
    {HIGH, HIGH, LOW, LOW},
    {LOW, HIGH, LOW, LOW},
    {LOW, HIGH, HIGH, LOW},
    {LOW, LOW, HIGH, LOW},
    {LOW, LOW, HIGH, HIGH},
    {LOW, LOW, LOW, HIGH},
    {HIGH, LOW, LOW, HIGH}
};

// Global variables used by the stepper algorithm
int DEBUG;
long ip, np, npMin;
double phi0, t, t0;

////////////////////////////////////
void setup() {

    DEBUG = 0;
    if (DEBUG == 1) {
        // Initialize the serial device
        Serial.begin(9600);
        Serial.println("Entering setup...");
    }
}

```

```

}

// Motor 0 (RA)
phi0 = y0/beam;           // Initial angle between the beams (approximate)
np    = (y0/inch2cm)*tpi*ppr; // Number of pulses needed from y0 down to 0
npMin = (y1/inch2cm)*tpi*ppr; // Turn off at the last y1 cm
ip    = 0;                 // Pulse counter

// Declare the motor pins as outputs
for (int j = 0; j < 4; j++)
    pinMode(motorPin[j], OUTPUT);
analogReference(DEFAULT);

t = micros(); // Time in micorseconds. Note, the math above is in seconds.
t0 = t;
ip = 0;

if (DEBUG == 1) {
    Serial.print("dtp = "); Serial.println(dtp);
    Serial.print("dtp = "); Serial.println(dtp);
    Serial.print("np = "); Serial.println(np);
    Serial.print("npMin = "); Serial.println(npMin);
    Serial.println("Setup complete...");
    Serial.println("I = Revolution number counting down from nRev to nMin");
    Serial.println("A = Actual time per revolution");
    Serial.println("R = Requested time per revolution");
}
}

////////////////////////////////////
void loop(){
    // We don't need this loop, let's write our own for better control
    int nRev = (np - npMin)/512;
    while (ip + npMin <= np) {
        int ic = ip % 8; // Internal cycle index
        if (ic < 0) ic = ic + 8;
        int inc = 0;      // 1=up, -1=down
        double dtl;
        double dt = 2.0*dtp; // speed = 100% of nominal RA speed
        int a0 = analogRead(0); // up
        int a1 = analogRead(1); // down
        if (DEBUG == 1) {
            Serial.print("inc="); Serial.print(inc); Serial.print(' ');
            Serial.print("ic="); Serial.print(ic); Serial.print(' ');
            Serial.print("a0="); Serial.print(a0); Serial.print(' ');
            Serial.print("a1="); Serial.println(a1);
        }
        if (a0 < 400) { // up
            inc = 1;
        }
        if (a1 < 400) { // down
            inc = -1;
        }
        if (inc != 0) {

```

```
        for (int j = 0; j < 4; j++)
            digitalWrite(motorPin[j], ccw[ic][j]);
    }
    t = t + 1000000*dt;
    dt1 = t - micros();
    if (dt1 > 0)
        delayMicroseconds(dt1);
    ip = ip + inc;
}
}
```