Parabolic Dish Antenna

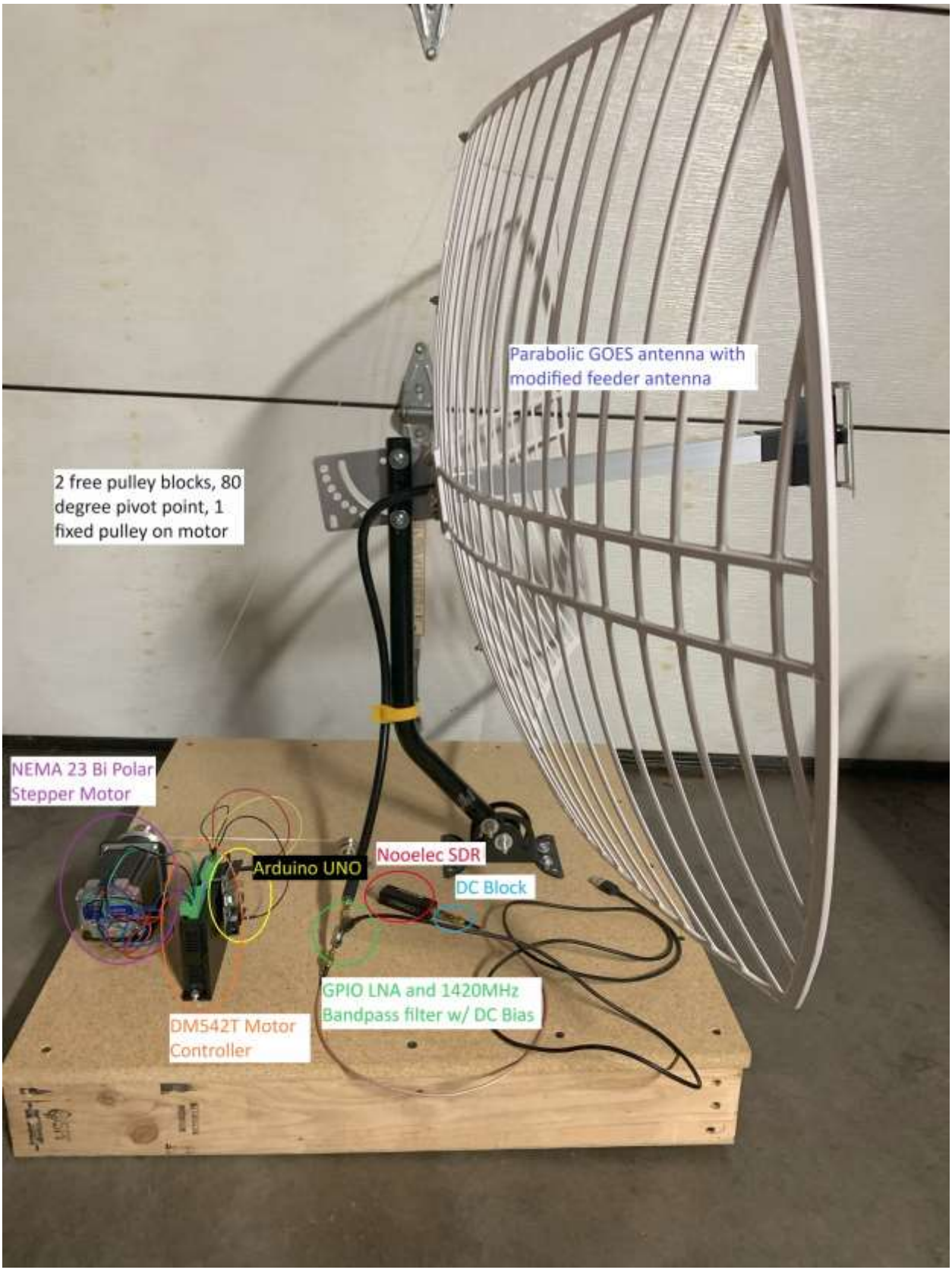
Radio Components			
Component	**Price**	**URL**	**Notes**
Nooelec NESDR Smart v5 SDR	$33.95	Nooelec	
LNA Filtered Hydrogen Line with Bias Tee	$78.00	gpio	
SDTC Tech Cable Coax	$8.99	Amazon	
SMA In-Line DC Block 2W	$24.95	Nooelec	
50 Ohm Terminator	$10.69	Amazon	5 pack
Nooelec GOES Parabolic Antenna	$109.95	Amazon	
USB Cable 3.0	$10.00		
Cost	$276.53		

*Prices may vary

Equatorial Mount Components			
Component	**Price**	**URL**	**Notes**
Adjustable Antenna Mount	$17.50	Amazon	
Small Pulley Block	$8.99	Amazon	32 mm
Aluminum single groove 8mm fixed bore pulley	$12.50	Amazon	Dia 58mm*Aperture 8mm
Sterpperonline CNC stepper motor driver	$25.99	Amazon	
Stepperonline Nema 23 stepper motor alloy steel mounting bracket	$7.87	Amazon	
Stepperonline Nema 23 stepper motor	$31.98	Amazon	
Arduino UNO REV3	$32.00	Amazon	
24V DC 4A power supply	$29.99	Amazon	
Cost	$166.82		

*Prices may vary

If you wanted to build both the parabolic dish antenna and the equatorial mount it would cost $443.35.


Parabolic GOES antenna with modified feeder antenna
2 free pulley blocks, 80 degree pivot point, 1 fixed pulley on motor
NEMA 23 Bi Polar Stepper Motor
Arduino UNO
Nooelec SDR
DC Block
GPIO LNA and 1420MHz Bandpass filter w/ DC Bias
DM542T Motor Controller

Motor Controller

Purpose:
Build an arduino code to use a DM542T to rotate a NEMA 23 stepper motor from 0 degrees to 80 degrees over 30 minutes and then from 80 degrees to 0 degrees over 30 minutes. The DM542T is connected as follows: PUL+ to pin 9, PUL- and DIR- to GND, DIR+ to pin 8 on the arduino.

To control a NEMA 23 stepper motor using a DM542T stepper driver and an Arduino, you can use the AccelStepper library to simplify the task of generating the step pulses and controlling the direction. Make sure you have the AccelStepper library installed in your Arduino IDE. You can install it via the Library Manager.
Here's a sample Arduino code that rotates the NEMA 23 stepper motor from 0 to 80 degrees over 30 minutes and then back to 0 degrees over 30 minutes

```
#include <AccelStepper.h>

// Define stepper motor connections
#define PUL_PIN 9
#define DIR_PIN 8

// Define the number of steps per revolution for your motor
#define STEPS_PER_REV 200  // Change this according to your motor's specifications

// Create an instance of the AccelStepper class
AccelStepper stepper(1, PUL_PIN, DIR_PIN);
void setup() {

  // Set the maximum speed and acceleration
  stepper.setMaxSpeed(1.2);  // Adjust this value as needed
  stepper.setAcceleration(100);  // Adjust this value as needed

  // Set the initial direction to move clockwise
  stepper.setSpeed(stepper.maxSpeed());

  // Set up the serial communication (for debugging)
  Serial.begin(9600);
  Serial.Println("Stepper test!");
}

void loop() {
  // Move from 0 to needed position over 30 minutes (counterclockwise)
  stepper.moveTo(-STEPS_PER_REV * 80 / 360*50);  // Convert degrees to steps and multiplies by a
factor to rotate it enough times to get dish into position
  stepper.runToPosition();  // Blocking call until the target position is reached

  // Move back to home over 30 minutes (clockwise)
  stepper.moveTo(0);  // Return to the initial position
  stepper.runToPosition();

}
```

Notes:
We include the AccelStepper library and set up the stepper motor's connections and parameters.

In the setup() function, we set the maximum speed and acceleration for the stepper motor and initialize the serial communication for debugging (you can remove this if not needed).

In the loop() function, we repeatedly rotate the motor from 0 to 80 degrees and then back to 0 degrees over 30 minutes each.

Make sure to adjust the STEPS_PER_REV value according to your motor's specifications, as it represents the number of steps required for a full revolution of your NEMA 23 stepper motor. Additionally, fine-tune the speed and acceleration settings based on your specific motor and application requirements.

An example wiring diagram is shown below: