

```

/** Sumner sketch to control a stepper motor for a turntable with TB6600 stepper motor driver and Arduino without a library
    /// 1st Arduino

// Define stepper motor connections and buttons:
const int dirPin = 2 ;    // To Stepper Motor (DIR+)
const int stepPin = 3 ;   // To Stepper Motor (PUL+)
const int buttonPin1 = 4 ; // To hand controller clockwise rotation (Blue Wire)
const int buttonPin2 = 5 ; // To hand controller counter-clockwise rotation (Yellow Wire)
const int signalPin2 = 7; // output pin 7 to 2nd Arduino LED pulse pin 5
const int Pin8 = 8;      // From hand controller starts indexing sequence. (Purple Wire)
const int Pin9 = 9;      // Input from Hall Effect.
const int distance = 400; // Make this less or more to set the final rotation place.
    // A0          // Reads the POT for rotation speed (Brown Wire)
// variables will change:

int buttonState1 = 0;    // variable for reading the pushbutton status
int buttonState2 = 0;    // variable for reading the pushbutton status
int Stop = HIGH;
int Start = LOW;
int Rotate = 0;
int i = 0;

void setup() {
  Serial.begin(9600);
  pinMode(stepPin, OUTPUT); // Declare pins as output:
  pinMode(dirPin, OUTPUT); // Declare pins as output:
  pinMode(signalPin2, OUTPUT); // Declare pins as output:

  pinMode(buttonPin1, INPUT); // initialize the pushbutton pin as an input:
  pinMode(buttonPin2, INPUT); // initialize the pushbutton pin as an input:

  pinMode(Pin8, INPUT);      // Input from hand control
  pinMode(Pin9, INPUT);      // Input from Hall Effect

  Stop = HIGH;
  Start = LOW;
  Rotate = 0;

// Set initial state of stepper motor
digitalWrite(stepPin, LOW); // Set stepping off
digitalWrite(dirPin, LOW); // Set the spinning direction CW:
digitalWrite (signalPin2, LOW); // Set LED pulse signal to 2nd Arduino low
}

void loop(){

    /// Index Track

```

```

Start = digitalRead (Pin8);
if (Start == HIGH){
Rotate = 1;
}
if (Rotate == 1){
Stop = digitalRead (Pin9); //Reads Pin9, Hall Effect, It goes low when it gets to the magnet.
digitalWrite(dirPin, LOW); // Set the spinning direction Clockwise
digitalWrite(stepPin, HIGH);
delayMicroseconds(500);
digitalWrite(stepPin, LOW);
delayMicroseconds(500);

if (Stop == LOW) { // Hall effect is now low and turn table stops
for(int i = 0; i<distance; i++) {
digitalWrite(stepPin, HIGH);
delayMicroseconds(5000);
digitalWrite(stepPin, LOW);
delayMicroseconds(5000);
}
Rotate = 0;
}
}

///< Read Button for clockwise rotation and rotate if it is pressed.

buttonState1 = digitalRead(buttonPin1); // read the state of the forward pushbutton value:

digitalWrite(signalPin2, LOW);
if (buttonState1 == HIGH){ // check if the pushbutton is pressed.
digitalWrite(dirPin, LOW);} // Set the spinning direction Clockwise

///< Read Button for counter-clockwise rotation and rotate if it is pressed.

buttonState2 = digitalRead(buttonPin2); // read the state of the reverse pushbutton value:

if (buttonState2 == HIGH){ // check if the pushbutton is pressed.
digitalWrite(dirPin, HIGH);} // Set the spinning direction counter-clockwise.}

if (buttonState1 == HIGH || buttonState2 == HIGH ) {

digitalWrite(signalPin2, HIGH); // Send LED pulse signal to 2nd Arduino high

int x = ( analogRead(A0) ); //Reads the analog value on pin A0 into x. Following code determines rotation speed. Set the delays for speed desired.

if (x > 854) {
digitalWrite(stepPin, HIGH);
delayMicroseconds(500);
digitalWrite(stepPin, LOW);
delayMicroseconds(500);}

```

```
else if (x > 683) {
  digitalWrite(stepPin, HIGH);
  delayMicroseconds(1000);
  digitalWrite(stepPin, LOW);
  delayMicroseconds(1000);}

else if (x > 513) {
  digitalWrite(stepPin, HIGH);
  delayMicroseconds(2000);
  digitalWrite(stepPin, LOW);
  delayMicroseconds(2000);}

else if (x > 400) {
  digitalWrite(stepPin, HIGH);
  delayMicroseconds(3000);
  digitalWrite(stepPin, LOW);
  delayMicroseconds(3000);}

else if (x > 300) {
  digitalWrite(stepPin, HIGH);
  delayMicroseconds(5000);
  digitalWrite(stepPin, LOW);
  delayMicroseconds(5000);}

else if (x > 100) {
  digitalWrite(stepPin, HIGH);
  delayMicroseconds(16000);
  digitalWrite(stepPin, LOW);
  delayMicroseconds(16000);}

else {digitalWrite(stepPin, LOW);}

else { digitalWrite(signalPin2, LOW); } // Skip past the rotation speed

}
```