Post date:
Sunday, October 5, 2014 - 00:15
Last modified:
Sunday, December 14, 2014 - 14:20

This article/page is part of a larger project, the main project page is: MINI NC Controller for Z-axis

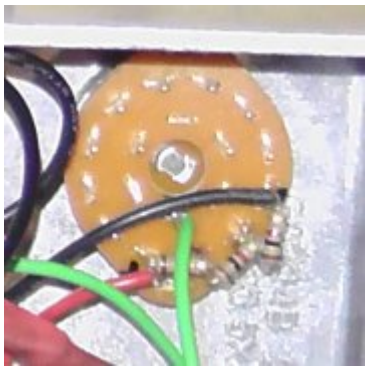
Here is the DOC (Depth Of Cut) box.
Box consists of:

- On/Off toggle switch
- LED bulb
- Four position rotary switch
- KY-040 Rotary Encoder Module
- 9 Pin D-Sub connector
- Aluminum box
- 9 wire shielded cable

The On/Off switch (bottom left) is just that, it will turn the box on or off so when the machine is in use there is no way to accidentally move the head, the Green LED is for visual notification (the on/off switch has a light in it also, but it was not bright enough).

The four position rotary switch (top left in image) is used to control the depth of cut increments .0005 / .001 / .010 / .100
If the selector is turned to .010, every "click" that the Rotary Encoder Knob (bottom right) is turned, the mill head with move ten thousands.
The four position switch is wired with three 1 Ohm resistors (to create a resistor ladder/divider) to an analog input on the Arduino board. By using the "resistor ladder" you can feed the first contact with power, ground the last leg/contact, place three resistors on the legs between and "pull" off the main pole which is connected to the Arduino board. Using three resistors will divide the voltage, depending on which position the switch is in gives different readings to the analog pin on the Arduino board. This saves on the number of pins used on the Arduino board (one analog compared to four digital with out the resistors).

Heres a shot (sorry not the best) of the resistors. You will see the Red power wire on the left, the Black ground on the right, the three resistors inbetween and the Green off the inner pole which hooks to the analog pin.



Heres the C++ for the four position rotary switch, im using a broad span for the readings, they can be tighten up alot.

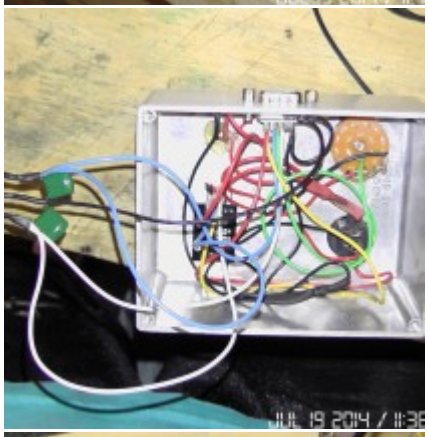
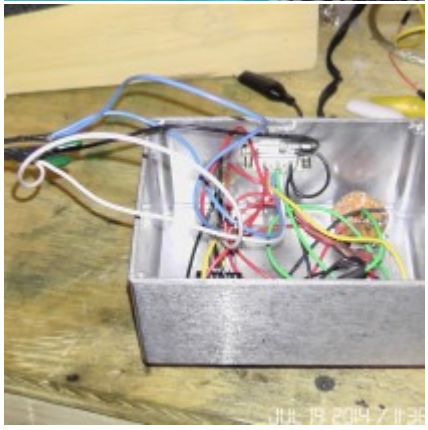
```
const int pinDepthCutSwitch = A5; //green with black strip
int depthCutSwitchReading = 0;
volatile unsigned long handWheelDepthOfCut = 100;

void loop() {
// depth of cut switch
    depthCutSwitchReading = analogRead(pinDepthCutSwitch);
    // Serial.println( depthCutSwitchReading );

    if(depthCutSwitchReading > 950){
        handWheelDepthOfCut = 1000;
    } else if(depthCutSwitchReading > 601 && depthCutSwitchReading < 699){
        handWheelDepthOfCut = 100;
    } else if(depthCutSwitchReading > 301 && depthCutSwitchReading < 399){
        handWheelDepthOfCut = 10;
    } else if(depthCutSwitchReading >= 0 && depthCutSwitchReading < 99){
        handWheelDepthOfCut = 1;
    }
}
```

The KY-040 Rotary Encoder Module is used to jog the mills head.
In the image below you will see two capcitors hooked to the White and Blue wires, these capcitors are being used to try and "smooth" the noise out of the line and not give false "readings" when the plate inside of the rotary encode "bounces" over the detents with in the switch when it is being turned.

Using Arduinos attachInterrupt routine to pick up any Rotary Encoder movment.
Below is the core C++ for that.

```
    const int handWheelBoxTogglePin = 6; // green
    const int handWheelBoxLEDPin = 5; // blue with black strip

// Hand Wheel Encoder
    const int handWheekDT = 2; // white
    const int handWheekCLK = 3; // blue
    const int handWheekSW = 4; // yellow

    volatile unsigned int HandwheelEncoderPos = 1;  // a counter for the dial
    volatile unsigned int HandwheelLastReportedPos = 1;   // change management
    volatile static boolean rotating=false;      // debounce management


void setup() {
    pinMode(handWheelBoxTogglePin, INPUT_PULLUP);
    pinMode(handWheelBoxLEDPin, OUTPUT);

// hand wheel
    // http://forum.arduino.cc/index.php/topic,209005.0.html
    pinMode(handWheekCLK, INPUT_PULLUP);
```

```
    pinMode(handWheekDT, INPUT_PULLUP);
    pinMode(handWheekSW, INPUT_PULLUP);

   attachInterrupt(0, handwheelChange_attachInterrupt, FALLING);  // CHANGE FALLING RISING
}


void loop() {

// hand wheelmoveStepper
    if(digitalReadFast(handWheelBoxTogglePin) == LOW){

        ledOnOff(true, handWheelBoxLEDPin);

        rotating = true;  // reset the debouncer
        if (HandwheelLastReportedPos != HandwheelEncoderPos){
            stepper.setCurrentPosition(0);

            // handWheelDirction is defaulted to 'n'
            // this is set in the handwheelChange_attachInterrupt() functions
            if(handWheelDirction != 'n'){
                moveStepperDistance(handWheelDirction, handWheelDepthOfCut, rapidSpeed);
                HandwheelLastReportedPos = HandwheelEncoderPos;
            }

        }
        if (digitalReadFast(handWheekSW) == LOW )  {
            HandwheelEncoderPos = 0;
        }

    } else {

        ledOnOff(false, handWheelBoxLEDPin);
    }

}


void handwheelChange_attachInterrupt(){

    // Exit out if Box Switch is Off.
    if(digitalReadFast(handWheelBoxTogglePin) == HIGH) return;

    if ( rotating ) delay (2);

    if( digitalReadFast(handWheekCLK) == LOW ) {
        HandwheelEncoderPos += 1;
        handWheelDirction='d';
    } else {
        HandwheelEncoderPos -= 1;
        handWheelDirction='u';
    }

    rotating = false;

}
```

- 
- 
- 
- 

Mobile QR Barcodes:

Save Article as PDF

Current URL Address

Other site pages which link here:

- MINI NC Controller for Z-axis

Category:
Industrial Hobbies Mill Modifications

Tags:
Arduino