## Arduino-RE controller

Input to Arduino-RE from controller Arduino-Control:

1. 5V
2. GND

Output to Arduino-Control from Arduino-RE:

1. Tic pulse. Short .5ms pulse for each tic. The degree rotation for each tic can be set up in the Arduino code.
2. Turn direction: high for clockwise, low for counter clockwise. It stays at the last state until it changes.
3. [optional] Active: ? not sure if I need this. Would just show high if the animal is pulling saving some processing on the Arduino-control side of things. Would go high when tics are detected and stay high for say 100ms.

Input from the Rotary encoder.

1. Green = A phase,
2. White = B phase,
3. Red = Vcc positive power supply 5-24V
4. Black = V0 ground, please wire as above.
5. Another metal shielding mesh layer is anti-interference. This can be shorted to black and also go in shielding around any cabling.

We have the 600 pulse/rotation. It’s a bit much so we have an Arduino in between to do some of the processing.

How to detect the direction and code.

///////////////////////////////////////////////////////

// Code to sense a rotary encoder.

// LPD3806 – Optical Rotary Encoder

//

///////////////////////////////////////////////////////

// Cowen and also code from

// https://www.instructables.com/Tutorial-of-Rotary-Encoder-With-Arduino/

// https://www.circuitschools.com/rotary-encoder-with-arduino-in-detail-with-example-codes/

///////////////////////////////////////////////////////

#define encoder0PinA 2

#define encoder0PinB 3

#define outPinTic 4

#define outPinDirection 5

#define TTL\_DELAY\_MS 2

#define TICS\_PER\_SIGNAL 100

int encoderPos = 0;

int tempEncoderPos = 0;

int ticCountCW = 0;

int ticCountCCW = 0;

void setup() {

Serial.begin(115200);

pinMode(encoder0PinA, INPUT\_PULLUP);

pinMode(encoder0PinB, INPUT\_PULLUP);

pinMode(outPinTic, OUTPUT);

pinMode(outPinDirection, OUTPUT);

digitalWrite(outPinTic,0);

digitalWrite(outPinDirection,0);

attachInterrupt(0, doEncoder, RISING); // Must be rising

}

int lastValRotary;

int valRotary;

int send\_signal = 0;

void loop() {

if (send\_signal > 0 ){

send\_signal = 0;

digitalWrite(outPinTic,HIGH);

delay(TTL\_DELAY\_MS);

digitalWrite(outPinTic,LOW);

if(valRotary>lastValRotary) {

digitalWrite(outPinDirection, HIGH);

Serial.print(" CW ");

// Serial.print(ticCountCW);

}

if(valRotary<lastValRotary) {

digitalWrite(outPinDirection, LOW);

Serial.print(" CCW ");

//Serial.print(ticCountCCW);

}

Serial.print(" re ");

Serial.print(valRotary);

Serial.println(" ");

}

lastValRotary = valRotary;

}

void doEncoder()

{ // Only runs on rising edge

if (digitalRead(encoder0PinA) != digitalRead(encoder0PinB))

{

encoderPos++;

tempEncoderPos++;

//ticCountCW++;

//ticCountCCW = 0;

}else{

encoderPos--;

tempEncoderPos--;

// ticCountCCW++;

// ticCountCW = 0;

}

valRotary = encoderPos;

if (abs(tempEncoderPos) >= TICS\_PER\_SIGNAL){

tempEncoderPos = 0;

send\_signal = 1;

}

}

## Rotary Encoder

Pinout:

Green = A phase,

White = B phase,

Red = Vcc positive power supply 5-24V

Black = V0 ground, please wire as above.

Another metal shielding mesh layer is anti-interference. This can be shorted to black and also go in shielding around any cabling.

We have the 600 pulse/rotation. It’s a bit much so we have an Arduino in between to do some of the processing.

AB two-phase output rectangular orthogonal pulse, the circuit output is NPN open collector output type, this output type can be directly connected with single-chip microcomputer or PLC with internal pull-up resistor, such as 51 single-chip or Mitsubishi PLC (PLC input mode Should be connected to the switch to 0V function), Note: If the encoder is not connected to the device, the oscilloscope cannot be directly oscillating (the open collector output has no voltage output when there is no pull-up resistor). Two pull-up resistors are added to the two-phase output of the AB;

Uses:

Used to measure the rotation speed, angle, acceleration and length of an object.

Scope of application:

It is suitable for intelligent control of various displacement change measurement, automatic fixed length automatic suede machine, steel fixed length cutting controller, civil height measuring human body scale, college student competition robot, etc.

Features:

It has the advantages of small size, light weight, convenient installation and high cost performance.

Note:

A phase and B phase output lines must not be connected directly to VCC, otherwise, the output three-stage tube will be burned.

# Connecting the camera to the neural recording system

The Matlab Image Acquisition Toolbox Graphical User Interface (GUI) and a second circuit were used to configure the camera to send a TTL pulse to the Intan data acquisition system (Intan Technologies, Los Angeles, California, United States) every time a frame was exposed. This allowed the timing of each frame to be aligned with the recorded neural activity. This circuit consisted of an open collector trigger from the camera and a pull up resistor to process the signal for triggering the Intan system. After collection, video information was processed by DeepLabCut (Mathis et al., 2018) and custom Matlab code (github.com/GiaJordan/Behavior\_Quantification).

NOTE: Open collector means no voltage goes out – do you need a **pullUP** resistor? According to <https://en.wikipedia.org/wiki/Open_collector> this is true ( I was mistakenly using a pulldown resistor)

Looks like the pullup can be in the range of 3KOhm to 6KOhm.

A picture containing text, diagram, technical drawing, plan

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence