Encoders

Rotary encoder: Type of position sensor. It generates an electrical signal which maybe analog or digital.

There are different types of rotary encoders.

OUTPUT SIGNAL

INCREMENTAL

ROTARY ENCODER

ABSOLUTE

MAGNETIC

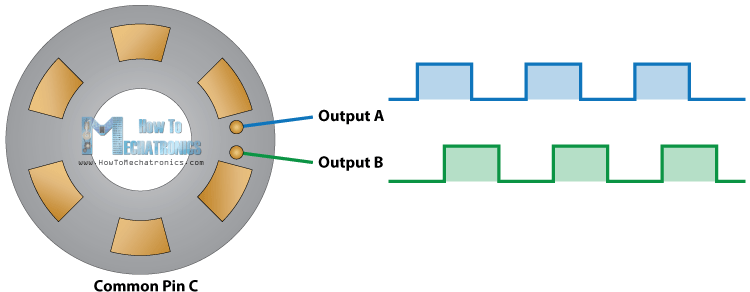
SENSING TECHNOLOGY

OPTICAL

LASER

WORKING:

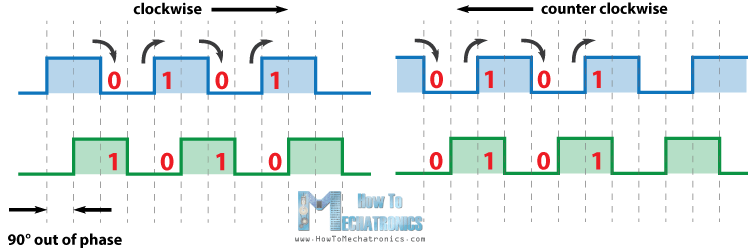
. Here’s how the square wave pulses are generated: The encoder has a disk with evenly spaced contact zones that are connected to the common pin C and two other separate contact pins A and B, as illustrated below.



When the disk will start rotating step by step, the pins A and B will start making contact with the common pin and the two square wave output signals will be generated accordingly.

Any of the two outputs can be used for determining the rotated position if we just count the pulses of the signal. However, if we want to determine the rotation direction as well, we need to consider both signals at the same time.

We can notice that the two output signals are displaced at 90 degrees out of phase from each other. If the encoder is rotating clockwise the output A will be ahead of output B.



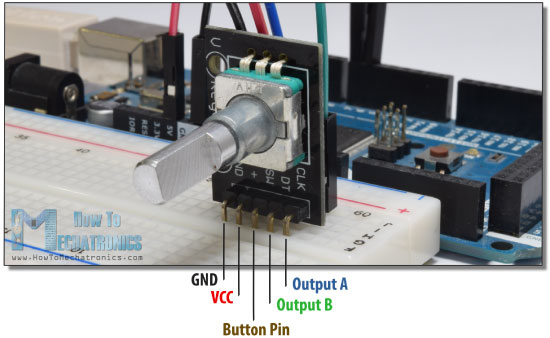
So if we count the steps each time the signal changes, from High to Low or from Low to High, we can notice at that time the two output signals have opposite values. Vice versa, if the encoder is rotating counter clockwise, the output signals have equal values. So considering this, we can easily program our controller to read the encoder position and the rotation direction.

## Source Code

Here’s the Arduino code:

1. #define outputA 6
2. #define outputB 7
3. **int** counter = 0;
4. **int** aState;
5. **int** aLastState;
6. **void** setup() {
7. pinMode (outputA,INPUT);
8. pinMode (outputB,INPUT);
9. Serial.begin (9600);
10. // Reads the initial state of the outputA
11. aLastState = digitalRead(outputA);
12. }
13. **void** loop() {
14. aState = digitalRead(outputA); // Reads the "current" state of the outputA
15. // If the previous and the current state of the outputA are different, that means a Pulse has occured
16. **if** (aState != aLastState){
17. // If the outputB state is different to the outputA state, that means the encoder is rotating clockwise
18. if (digitalRead(outputB) != aState) {
19. counter ++;
20. } **else** {
21. counter --;
22. }
23. Serial.print("Position: ");
24. Serial.println(counter);
25. }
26. aLastState = aState; // Updates the previous state of the outputA with the current state
27. }

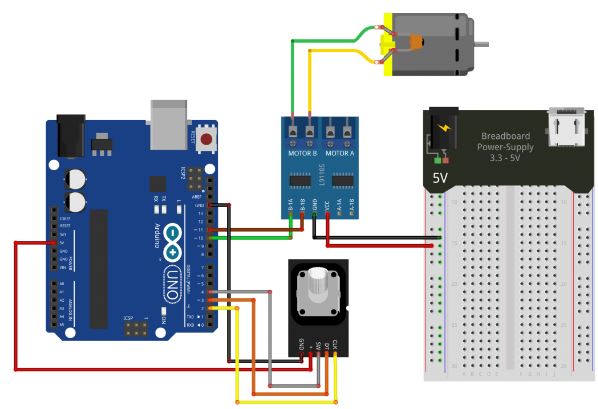
CONNECTION:



Code and explanation: <http://howtomechatronics.com/tutorials/arduino/rotary-encoder-works-use-arduino/>

**Control a DC Motor with L9110S Rotary Encoder and Arduino**

**Connections:**



**Code and working:**

<https://brainy-bits.com/blogs/tutorials/l9110s>

**INFRARED ENCODER:**

Infrared encoder:requirements ir led emitter and phototransistor(datasheet of ir led : <https://cdn-shop.adafruit.com/datasheets/IR333_A_datasheet.pdf>)

Components req:

Ir led

Phototransistor

**Code and connections for IR encoder to track transitions between 2 colors black and white :**

<https://www.youtube.com/watch?v=WKdsLo1gTKo>

const int led1Pin=2;

const int led2Pin =3;

const int led3Pin=4;

const int led4Pin=5;

const int sensorPin=7;

const int threshold\_value=700;

int distance=0;

int current\_color=0;//black=0,white=1

void setup()

{

Serial.begin(9600);

pinMode(led1Pin,OUTPUT);

pinMode(led2Pin,OUTPUT);

pinMode(led3Pin,OUTPUT);

pinMode(led4Pin,OUTPUT);

}

void loop(){

int reading=analogRead(sensorPin);

Serial.print("Distance: ");

Serial.print(distance);

Serial.print('\n');

Serial.print("Color :");

if(current\_color ==0)

Serial.print("White");

else

Serial.print("Black");

Serial.print('\n');

if(reading>threshold\_value){

if(current\_color ==0){

distance++;

current\_color =1;

}

}

else

{

if(current\_color ==1)

distance++;

current\_color=0;

}

}

**NOTE:** Over here we are measuring the distance as a counter is running for distance in the form of discrete values as the color changes therefore we should try and keep as many partions between black and white color to get better precission.

Also since it is the distance which we need or the analog values which is more important we can test the bot number of times to see to what distance measured by counter distance++ corresponds to the distance travelled in real world.