**Task 2**

**ROTARY ENCODERS**

**PREPARED BY :-**

**Debashis Pradhan**

**Mechanical Engineering**

**First Year**

**VSSUT Robotics Society**

**OVERVIEW**

[**Rotary Encoders**](http://www.ia.omron.com/support/glossary/atoz/179/index.html) are sensors that detect position and speed by converting rotational mechanical displacements into electrical signals and processing those signals. It is also called Shaft Encoders. Sensors that detect mechanical displacement for straight lines are referred to as Linear Encoders. Rotary Encoders measure the number of rotations, the rotational angle, and the rotational position. It is a type of position sensor which is used for determining the angular position of a rotating shaft. It generates an electrical signal, either analog or digital, according to the rotational movement.

**USES OF ROTARY ENCODERS**

Rotary encoders are used in a wide range of applications that require monitoring or control, or both, of mechanical systems, including industrial controls, [robotics](https://en.wikipedia.org/wiki/Robotics), [photographic lenses](https://en.wikipedia.org/wiki/Photographic_lens), computer input devices such as optomechanical [mice](https://en.wikipedia.org/wiki/Computer_mouse) and [trackballs](https://en.wikipedia.org/wiki/Trackball), controlled stress [rheometers](https://en.wikipedia.org/wiki/Rheometers), and rotating [radar](https://en.wikipedia.org/wiki/Radar) platforms.

Usually this is for the purpose of monitoring or controlling motion parameters such as speed, rate, direction, distance or position. When applying encoders, selecting the optimum model and specifying the appropriate configuration are critical for success.  Proper encoder selection begins by understanding the role of the encoder in the motion control system.

**TYPES OF ROTARY ENCODERS**

There are two main types of rotary encoder: **absolute** and **incremental**. The output of an absolute encoder indicates the current shaft position, making it an [angle transducer](https://en.wikipedia.org/wiki/Transducer). The output of an incremental encoder provides information about the *motion* of the shaft, which typically is processed elsewhere into information such as position, speed and distance.

The two types of Rotary Encoders :-

**▪ Absolute :**An absolute encoder maintains position information when power is removed from the encoder. The position of the encoder is available immediately on applying power. The relationship between the encoder value and the physical position of the controlled machinery is set at assembly; the system does not need to return to a calibration point to maintain position accuracy.

An absolute encoder has multiple code rings with various binary weightings which provide a [data word](https://en.wikipedia.org/wiki/Data_word) representing the absolute position of the encoder within one revolution. This type of encoder is often referred to as a parallel absolute encoder.

A multi-turn absolute rotary encoder includes additional code wheels and gears. A high-resolution wheel measures the fractional rotation, and lower-resolution geared code wheels record the number of whole revolutions of the shaft.

**Advantages of Absolute Encoders**

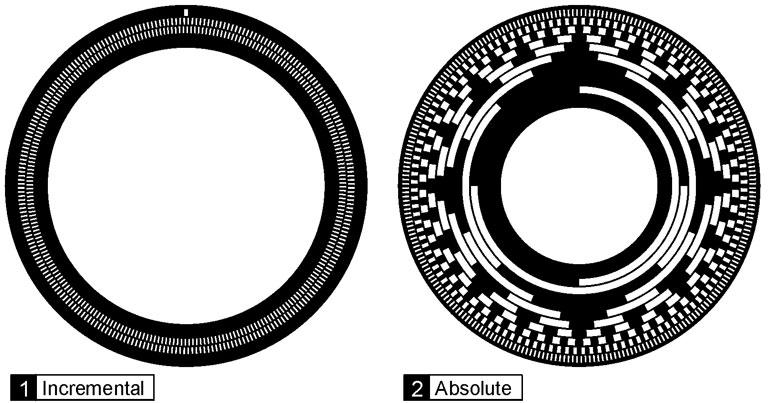
* Remembers its position after a power outage and offers continuous position monitoring
* Typically have speed, scaling, preset, and fieldbus functions
* Allow you to determine the exact position of a machine and control over the storage of electronic data
* Multiple interface options: Analog, Ethernet, Fieldbus, Parallel, Serial
* Single-turn and Multi-turn revolution options available
* Optical a magnetic measuring principle
* Absolute encoders have a resolution of up to 16 bits, or 65,536 pulses per revolution (PPR).

**▪ Incremental :**The output signal of incremental encoders is generated each time the shaft rotates a certain amount and the resolution of the device is defined by the number of signals per turn. Each time the encoder is powered on it begins counting from zero, regardless of where the shaft is or its previous position. It is necessary for a reference point to be determined in all positioning tasks, both in the startup of the control system and when the power to the encoder has been disrupted. It is necessary for incremental encoders to re-home upon the reference point when powered down.

Rotary incremental encoders store the data in an external buffer or counter. Battery backups can help eliminate the need for re-homing after planned or unplanned shutdowns. Incremental encoders are typically simpler to use and cheaper than absolute encoders.

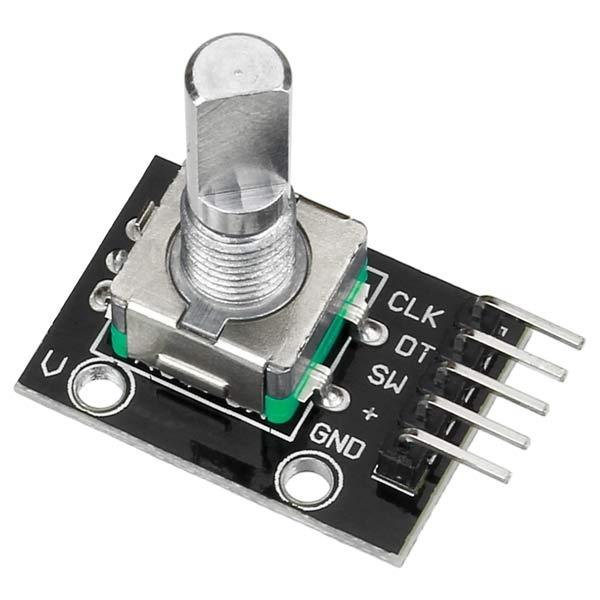
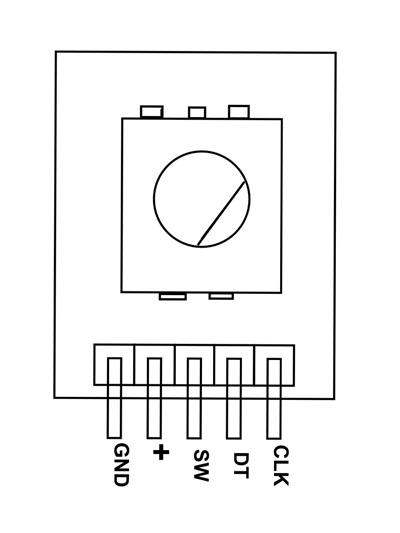
**Advantages of Incrememtal Encoders**

* Good for simple pulse counting or frequency monitoring applications such as speed, direction, and position monitoring
* More cost-effective and less complex than an absolute encoder
* A, B, Z, and inverted signals as HTL (Push-Pull) or TTL (RS422).
* Any pulse count up to 16384 PPR available
* Flexible scaling functionality
* Magnetic measuring principle
* Incremental encoders have a resolution of up to 50,000 PPR.



**ROTARY ENCODER MODULE**

**Rotary Encoder** is a device used for knowing the **axial movement and its direction**. Although they are available in various types here we are going to see about simple contact type encoder module. Here we are going to use **M274 Rotary Encoder Module**.

**Rotary Encoder Rotary Encoder Pinout**

### **Rotary Encoder Module Pin Configuration**

Rotary Encoder Module is five pin device as shown in the **M274 pinout**. In them four pins are compulsory for using the module.

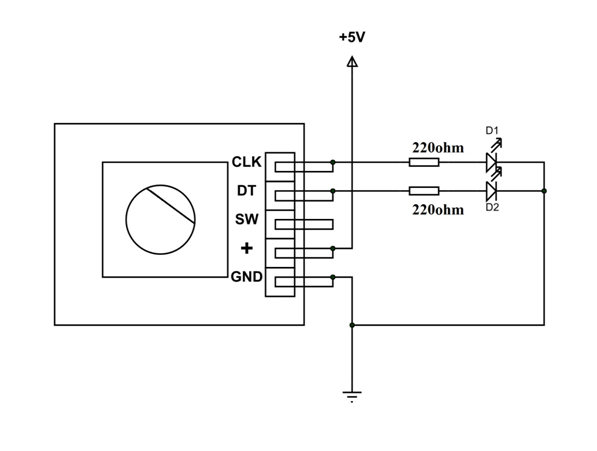
|  |  |
| --- | --- |
| **Pin Name** | **Description** |
| GND | Connected to Ground |
| **+** | Connected to +5V |
| SW | Output of internal button |
| DT | Contact A output or DATA |
| CLK | Contact B output or CLOCK |

In the above pins Contact A or DT and Contact B or CLK are used to measure the axis movement and direction.

**Rotary Encoder Features and Specifications**

* 360º free rotation.
* 20 steps or cycles per revolution
* Incremental type encoder
* Can work on low voltages
* Maximum operating temperature: 0°C to + 80°C
* Easy interface
* Long life

**ROTARY ENCODER MODULE IN CRCUIT CONNECTION**

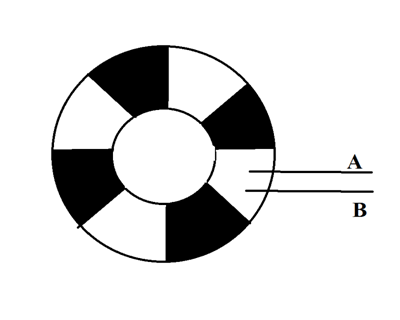


Here we have connected two outputs to two LEDs to check the state of the module. The button is of no use here so it left alone. The module is powered by +5V power source.

**WORKING OF ROTARY ENCODERS**

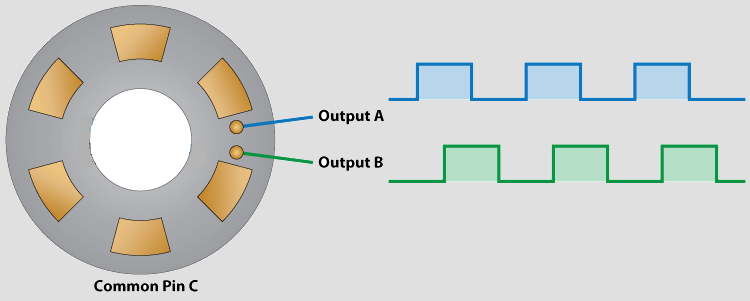
**Step 1: Output Channels**

The most common type of encoder uses two main output channels, A and B. The two output channels of the encoder assist in indicating the position and the direction of rotation. Suppose, if B leads A, then the disk of the encoder rotates in a clockwise direction. While, if A leads B, then the disk of the encoder rotates in a counter-clockwise direction.



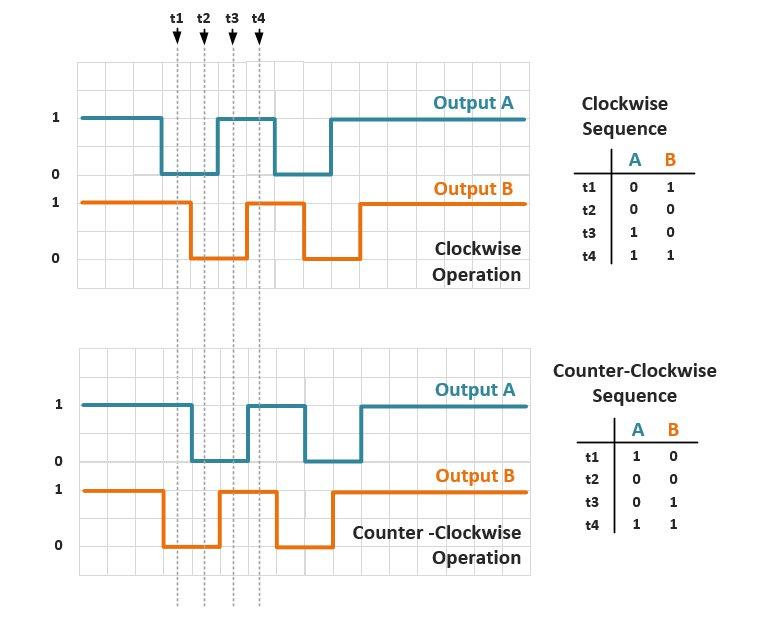
**Step 2: Observing the Pulse**

You will be able to follow the direction and the position of the rotation by observing both the pulses and the signals of A and B. 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.



**Step 3: Differentiating the Pulse**

We can see 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.

**Step 4: Third Output Channel**

Some of the encoders also include a third output channel which is called an index signal. It helps in supplying a single pulse per revolution, which is used for determining the reference position.

Also, keep in mind that if these encoders are turning fast, then a backward count might be occurring

**INTERFACING ROTARY ENCODERS IN ARDUINO**

Reading a control encoder with an Arduino is actually fairly straightforward. We just need to read input pulses and count them. We also need to determine which set of pulses is occurring first, so that we can determine the direction of rotation.

Most simple encoders like this only make use of 4 pins, one of those is ground and other Vcc. Those other two pins change state and are always either high or low, so they can only have a total of 4 combinations. 00, 01, 10, and 11. This is known as 2 bit gray code. So when you turn it, the arduino can say… you were at 01, and now you are at 00 so you move this way. Or you were at 01, but now you are at 10 so you must have moved the other way. You can see the encoder has 5 pins, the other 2 are just a simple switch that is engaged when you press down.

**Code**

In addition to the rotary encoder there are couple of LEDs. These will indicate the direction that we are spinning the encoder shaft.   
Here is how we will hook up our first rotary encoder experiment code.

|  |  |
| --- | --- |
|  | // Rotary Encoder Inputs  #define inputA 4  #define inputB 5    // LED Outputs  #define ledCW 8  #define ledCCW 9    int counter = 0;  int currentStateA;  int previousStateA;    String encdir ="";    void setup() {     pinMode (inputA,INPUT); // Set encoder pins as inputs  pinMode (inputB,INPUT);     pinMode (ledCW,OUTPUT); // Set LED pins as outputs     pinMode (ledCCW,OUTPUT);     Serial.begin (9600); // Setup Serial Monitor       previousStateA = digitalRead(inputA);    // Read the initial state of inputA     // Assign to previousStateA variable  }    void loop() {       currentStateA = digitalRead(inputA);    // Read the current state of inputA  if (currentStateA != previousStateA){ // If the previous and the current state  of the inputA are different then a  pulse has occured        if (digitalRead(inputB) != currentStateA) { // If the inputB state is different  than the inputA state then  counter --; // the encoder is rotating counterclockwise         encdir ="CCW";         digitalWrite(ledCW, LOW);         digitalWrite(ledCCW, HIGH);         } else {         counter ++;    // Update previousStateA with the  current state         encdir ="CW";         digitalWrite(ledCW, HIGH);         digitalWrite(ledCCW, LOW);         }       Serial.print("Direction: ");       Serial.print(encdir);       Serial.print(" -- Value: ");       Serial.println(counter);     }     previousStateA = currentStateA; // Update previousStateA with the  current state  } |

**SOURCES**

▪ <https://en.wikipedia.org/wiki/Rotary_encoder>

▪ <https://howtomechatronics.com/tutorials/arduino/rotary-encoder-works-use-arduino/>

▪ <https://www.doityourself.com/stry/how-does-a-rotary-encoder-work>

▪ <https://components101.com/sensors/rotary-encoder-pinout-circuit-working>

▪ <https://www.machinedesign.com/motion-control/what-s-difference-between-absolute-and-incremental-encoders>

▪ <https://dronebotworkshop.com/rotary-encoders-arduino/>

▪ <http://www.ia.omron.com/support/guide/34/introduction.html>

▪ <https://www.youtube.com/watch?v=v4BbSzJ-hz4>