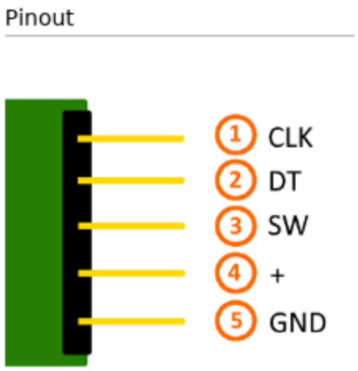
Rotary Encoder

Description:

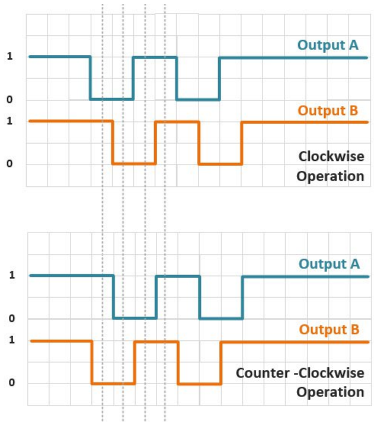
A rotary encoder is an electromechanical device used to measure the angular position of a rotating object. It consists of a rotating shaft and a sensing mechanism that generates electrical signals corresponding to the rotation. The encoder can provide information about the direction and speed of rotation, as well as precise position data. It is commonly used in various applications such as robotics, industrial control systems, and it is preferred to be used indoor.

Datasheet:

- CLK is pin A
- DT is pin B
- SW is the switch pin
- + is the vcc
- GND is the ground



when A leads B this means that the shaft rotates clockwise but when B leads A it rotates counter clockwise.



Code:

1. Define rotary encoder pins and initialize a counter variable to count the number of pulses.

```
#define ENCODER_CLK_A 2
#define ENCODER_DT_B 3

long long pulse_counter = 0;
```

2. Firstly, we need to set the pins as inputs with pull up resistors. Then we should use attachInterrupt(digitalPinToIterrupt(PIN_NAME , METHOD_NAME , MODE);
so, when the pulse state changes its mode it should call the method given.

```
void setup() {

  Serial.begin(9600);

  pinMode(ENCODER_CLK_A, INPUT_PULLUP);
  pinMode(ENCODER_DT_B,INPUT_PULLUP);

  attachInterrupt(digitalPinToInterrupt(ENCODER_CLK_A),ISR_ENCODER_PIN_A,CHANGE);
  attachInterrupt(digitalPinToInterrupt(ENCODER_DT_B),ISR_ENCODER_PIN_B,CHANGE);
}
```

3. When ISR_ENCODER_PIN_A() is called it checks if pin A is not equal to pin B it increments the pulse counter by 1 as it means that this is a rising edge and A leads B, otherwise it decrements the counter by 1 as it means that pin A read a Falling edge.

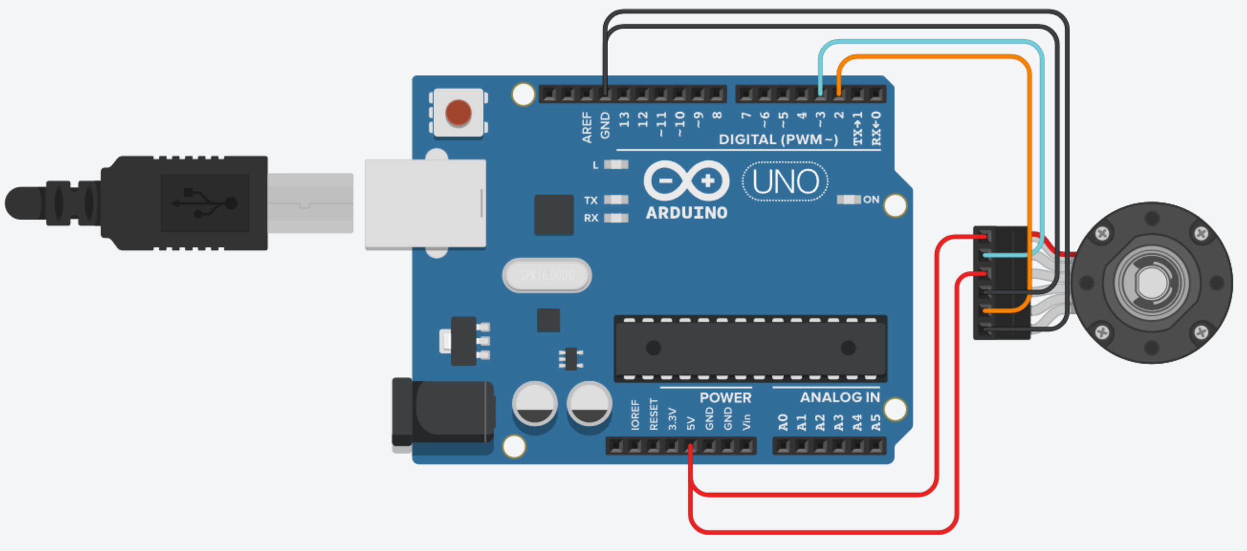
```
void ISR_ENCODER_PIN_A() {
  if(digitalRead(ENCODER_CLK_A) != digitalRead(ENCODER_DT_B))
    pulse_counter++;
  else
    pulse_counter--;
}
```

4. When ISR_ENCODER_PIN_B() is called it checks if pin B is equal to pin A it increments the pulse counter by 1 as it means that this is a rising edge and A leads B, otherwise it decrements the counter by 1 as it means that pin B read a Falling edge.

```
void ISR_ENCODER_PIN_B() {
  if(digitalRead(ENCODER_CLK_A) == digitalRead(ENCODER_DT_B))
    pulse_counter++;
  else
    pulse_counter--;
}
```

Here is the whole code and its simulation:

- simulation link on wokwi
- simulation link on Tinkercad



```
#define ENCODER_CLK_A 2
#define ENCODER_DT_B 3
//initializing counter variable for pulses
long long pulse_counter = 0;

void setup() {

  Serial.begin(9600);
  //set encoder pins as inputs with internal pull up resistors
  pinMode(ENCODER_CLK_A, INPUT_PULLUP);
  pinMode(ENCODER_DT_B,INPUT_PULLUP);
  //attach interupts to encoder pins at any rise or fall
  attachInterrupt(digitalPinToInterrupt(ENCODER_CLK_A),ISR_ENCODER_PIN_A,CHANGE);
  attachInterrupt(digitalPinToInterrupt(ENCODER_DT_B),ISR_ENCODER_PIN_B,CHANGE);
}

void loop() {
  //
}

void ISR_ENCODER_PIN_A() {
  if(digitalRead(ENCODER_CLK_A) != digitalRead(ENCODER_DT_B)) //moving clockwise
    pulse_counter++;
  else //moving counter clockwise
    pulse_counter--;
}

void ISR_ENCODER_PIN_B() {
  if(digitalRead(ENCODER_CLK_A) == digitalRead(ENCODER_DT_B)) //moving clockwise
    pulse_counter++;
  else //moving counter clockwise
    pulse_counter--;
}
```