San Francisco Bay University

CE450 Fundamentals of Embedded Engineering
Lab 8 Rotary Encoder Design

Objectives:

The rotary encoder is popularly used in the control system. In this week, students will design the program to control rotary encoder through GPIO ports on Raspberry Pi bord and do hands-on exercise through lab assignments

Introduction:

A rotary encoder is an electro-mechanical device that converts the angular position or motion of a shaft or axle to analog or digital code. Rotary encoders are usually placed at the side which is perpendicular to the shaft. They act as sensors for detecting angle, speed, length, position, and acceleration in automation field.

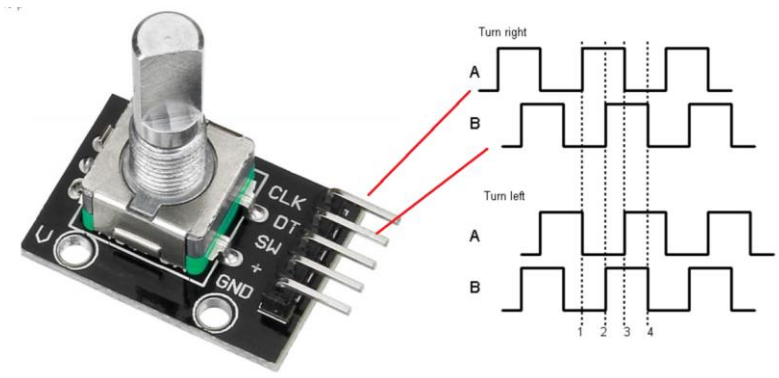
Equipment:

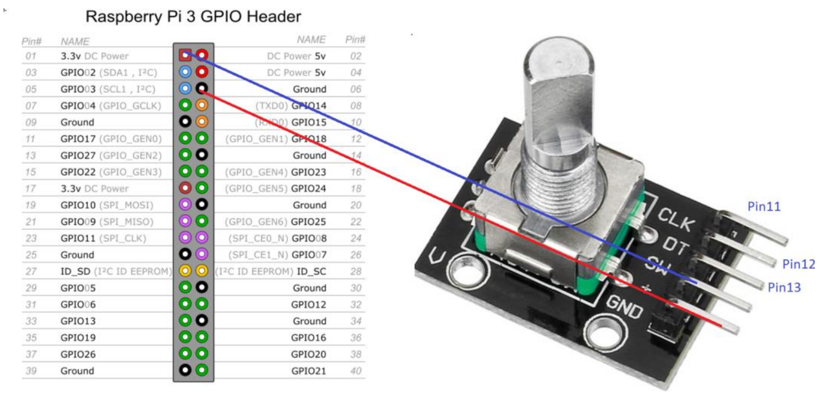
The equipment you require is as follows:

- Laptop & Raspberry Pi 3 model Board
- SunFounder Super Starter Kit V2.0 for Raspberry Pi
- One Rotary Encoder module

The Laboratory Procedure:

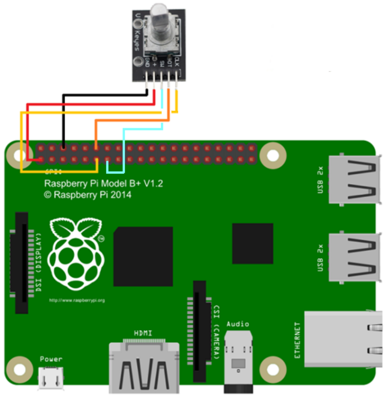
1. **Hardware connection**





Raspberry Pi 3 GPIO Header

Pin#	NAME	NAME	Pin#
01	3.3v DC Power	DC Power 5v	02
03	GPIO02 (SDA1 , I²C)	DC Power 5v	04
05	GPIO03 (SCL1 , I²C)	Ground	06
07	GPIO04 (GPIO_GCLK)	(TXD0) GPIO14	08
09	Ground	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Ground	14
15	GPIO22 (GPIO_GEN3)	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Ground	20
21	GPIO09 (SPI_MISO)	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	(SPI_CE0_N) GPIO08	24
25	Ground	(SPI_CE1_N) GPIO07	26
27	ID_SD (I²C ID EEPROM)	(I²C ID EEPROM) ID_SC	28
29	GPIO05	Ground	30
31	GPIO06	GPIO12	32
33	GPIO13	Ground	34
35	GPIO19	GPIO16	36
37	GPIO26	GPIO20	38
39	Ground	GPIO21	40



2. Control program in Python

```
# Python Program

import RPi.GPIO as GPIO
import time

RoAPin = 11     # pin11 -> Connected to CLK
RoBPin = 12     # pin12 -> Connected to DT
RoSPin = 13     # pin13 -> Connected to SW

globalCounter = 0

flag = 0
Last_RoB_Status = 0              # two var. for pin B’s value
Current_RoB_Status = 0

def setup():
        GPIO.setmode(GPIO.BOARD)        # Numbers GPIOs by physical location
        GPIO.setup(RoAPin, GPIO.IN)     # input mode
        GPIO.setup(RoBPin, GPIO.IN)
        GPIO.setup(RoSPin,GPIO.IN, pull_up_down=GPIO.PUD_UP)   # Bottom pin
        rotaryClear()

def rotaryDeal():
        global flag
        global Last_RoB_Status
        global Current_RoB_Status
        global globalCounter
        Last_RoB_Status = GPIO.input(RoBPin)  # Read in data from DT
        while (not GPIO.input(RoAPin)):
                Current_RoB_Status = GPIO.input(RoBPin)
                flag = 1
        if flag == 1:
                flag = 0
                if (Last_RoB_Status == 0) and (Current_RoB_Status == 1):
                        globalCounter = globalCounter + 1
                        print 'globalCounter = %d' % globalCounter
                if (Last_RoB_Status == 1) and (Current_RoB_Status == 0):
                        globalCounter = globalCounter - 1
                        print 'globalCounter = %d' % globalCounter

def clear(ev=None):
                globalCounter = 0
        print 'globalCounter = %d' % globalCounter
        time.sleep(1)

def rotaryClear():
        GPIO.add_event_detect(RoSPin, GPIO.FALLING, callback=clear)
        # wait for falling

def loop():
        global globalCounter
        while True:
                rotaryDeal()
#               print 'globalCounter = %d' % globalCounter

def destroy():
        GPIO.cleanup()             # Release resource
setup()
try:
        loop()
except  KeyboardInterrupt:  # When 'Ctrl+C' is pressed, the child program
destroy() will be  executed.
        destroy()
```

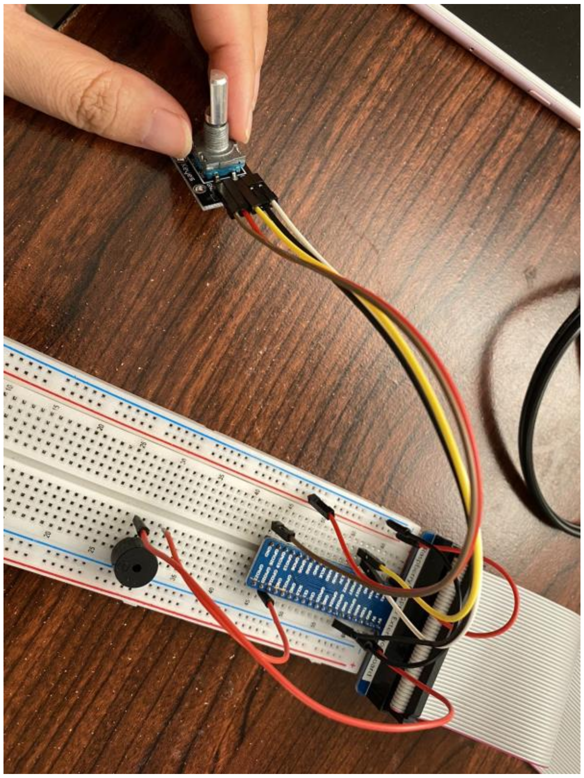
**Note: Hardware connection reference and running command*

https://learn.sunfounder.com/lesson-12-rotary-encoder/

https://learn.sunfounder.com/category/super-kit-v3-0-for-raspberry-pi/

The Laboratory Assignments:

1. Build up the hardware circuit and run the example program to observe what will happen



2. Add a buzzer to the above circuit and design the program to make “z” sound as the turning indicator if the encoder is turned one circle
   Youtube Link:
   https://youtu.be/sD1wTkyleBE

```
import RPi.GPIO as GPIO
import time
RoAPin = 11    # pin11 -> Connected to CLK
RoBPin = 12    # pin12 -> Connected to DT
RoSPin = 13    # pin13 -> Connected to SW
BuzzerPin = 36 # pin36 -> connected to Buzzer 1, BCM GPIO16
globalCounter = 0
flag = 0
Last_RoB_Status = 0 # two var. for pin B’s value
Current_RoB_Status = 0
def setup():
   GPIO.setmode(GPIO.BOARD)      # Numbers GPIOs by physical location
   GPIO.setup(RoAPin, GPIO.IN)    # input mode
   GPIO.setup(RoBPin, GPIO.IN)
   GPIO.setup(BuzzerPin, GPIO.OUT)
   GPIO.setup(BuzzerPin, GPIO.HIGH)
   GPIO.setup(RoSPin,GPIO.IN, pull_up_down=GPIO.PUD_UP)  # Bottom pin
   rotaryClear()
def rotaryDeal():
   global flag
```

```
    global Last_RoB_Status
    global Current_RoB_Status
    global globalCounter
    Last_RoB_Status = GPIO.input(RoBPin) # Read in data from DT
    while(not GPIO.input(RoAPin)):
        Current_RoB_Status = GPIO.input(RoBPin)
        flag = 1
        GPIO.setup(BuzzerPin, GPIO.HIGH)
    if flag == 1:
        flag = 0
        if (Last_RoB_Status == 0) and (Current_RoB_Status == 1):
            globalCounter = globalCounter + 1
            GPIO.setup(BuzzerPin, GPIO.LOW)
            print ('globalCounter = %d' % globalCounter)
        if (Last_RoB_Status == 1) and (Current_RoB_Status == 0):
            globalCounter = globalCounter - 1
            GPIO.setup(BuzzerPin, GPIO.LOW)
            print ('globalCounter = %d' % globalCounter)
def clear(ev=None):
    globalCounter = 0
    print ('globalCounter = %d' % globalCounter)
    time.sleep(1)
def rotaryClear():
    GPIO.add_event_detect(RoSPin, GPIO.FALLING, callback=clear)
# wait for falling
def loop():
    global globalCounter
    while True:
        rotaryDeal()
        # print 'globalCounter = %d' % globalCounter
def destroy():
    GPIO.cleanup()              # Release resource

setup()
try:
    loop()
except  KeyboardInterrupt:  # When 'Ctrl+C' is pressed, the child program
destroy() will be  executed.
    destroy()
```