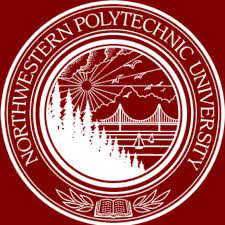
Rotary encoder Using RASPBERRY PI 3 GPIO



Aditi Vaidya

vaidya19598@mail.npu.edu



Northwestern Polytechnic University Fremont, CA Copyright © 2017, All Rights Reserved .

Table of Contents

|  |  |  |  |
| --- | --- | --- | --- |
| 1. | Introduction | |  |
|  | 1.1 | Objective |  |
|  | 1.2 | Requirement |  |
|  | 1.3 | Principle |  |
| 2. | Hardware connections | |  |
| 3. | Working | |  |
|  | 3.1 | Programming |  |
|  | Objective The main objective is to achieve operations like increament and decreament. Using a rotary encoder, such operations can be performed. With Raspberry Pi 3 model B development board and inbuilt interface thony ide.   Requirement  * 1 \* Raspberry Pi * 1 \* Breadboard * 4 \* Jumper wires (Male to Male, 2 red and 2 black) * 1 \* Network cable (or USB wireless network adapter) * 1 \* Rotary Encoder module * 1 \* 5-Pin anti-reverse cable  Principle In this experiment ,when Raspberry Pi output is supplied with low or 3.3 V by programming, when the rotary encoder is spun clockwise or anticlockwise the output varies based on encoding as shown below: 2.0 Hardware connections Use bread board and make connections as shown in visual representation below:   Working Once the wireless connection is established then open thonny Python IDE which mostly is pre installed in raspberry Pi’s.   1. Make sure the Python library is uploaded and running . 2. Type in the program, and check for any errors. 3. Run the program 4. Turn on the power module . 5. Check for incremented or decremented value on terminal based on encoder. | |  |

|  |  |
| --- | --- |
| **3.1** | **Programming**  Enter the following program on Thonny Python IDE : |

import RPi.GPIO as GPIO

import time

# Set up pins

# Rotary A Pin

RoAPin = 17

# Rotary B Pin

RoBPin = 18

# Rotary Switch Pin

RoSPin = 27

def print\_message():

print ("========================================")

print ("| Rotary Encoder |")

print ("| ------------------------------ |")

print ("| Pin A connect to GPIO0 |")

print ("| Pin B connect to GPIO1 |")

print ("| Button Pin connect to GPIO 2 |")

print ("| |")

print ("| Use a Rotary Encoder |")

print ("| Rotary to add/minus counter |")

print ("| Press to set counter to 0 |")

print ("| |")

print ("| Aditi Vaidya|")

print ("========================================\n")

print 'Please press Ctrl+C to end the program...'

raw\_input ("Press Enter to begin\n")

def setup():

global counter

global Last\_RoB\_Status, Current\_RoB\_Status

GPIO.setmode(GPIO.BCM)

GPIO.setup(RoAPin, GPIO.IN)

GPIO.setup(RoBPin, GPIO.IN)

GPIO.setup(RoSPin,GPIO.IN, pull\_up\_down=GPIO.PUD\_UP)

# Set up a falling edge detect to callback clear

GPIO.add\_event\_detect(RoSPin, GPIO.FALLING, callback=clear)

# Set up a counter as a global variable

counter = 0

Last\_RoB\_Status = 0

Current\_RoB\_Status = 0

# Define a function to deal with rotary encoder

def rotaryDeal():

global counter

global Last\_RoB\_Status, Current\_RoB\_Status

flag = 0

Last\_RoB\_Status = GPIO.input(RoBPin)

# When RoAPin level changes

while(not GPIO.input(RoAPin)):

Current\_RoB\_Status = GPIO.input(RoBPin)

flag = 1

if flag == 1:

# Reset flag

flag = 0

if (Last\_RoB\_Status == 0) and (Current\_RoB\_Status == 1):

counter = counter + 1

if (Last\_RoB\_Status == 1) and (Current\_RoB\_Status == 0):

counter = counter - 1

print 'counter = %d' % counter

# Define a callback function on switch, to clean "counter"

def clear(ev=None):

global counter

counter = 0

def main():

print\_message()

while True:

rotaryDeal()

def destroy():

# Release resource

GPIO.cleanup()

# If run this script directly, do:

if \_\_name\_\_ == '\_\_main\_\_':

setup()

try:

main()

# When 'Ctrl+C' is pressed, the child program

# destroy() will be executed.

except KeyboardInterrupt:

destroy()