#Import GPIO module សម្រាប់គ្រប់គ្រងទៅលើRaspberry Pi GPIO channels

import RPi.GPIO as GPIO

#Python time module អនុញ្ញាតអោយយើងធ្វើការជាមួយfunctionទាំងអស់ដែលទាក់ទងtime

import time

#កំណត់Pinរបស់LCDទៅRaspberry Pi

LCD\_RS = 7

LCD\_E = 8

LCD\_D4 = 25

LCD\_D5 = 24

LCD\_D6 = 23

LCD\_D7 = 18

# Device constants

LCD\_WIDTH = 16 # កំណត់characterធំបំផុតត្រឹម១៦មួយជួរ

LCD\_CHR = True

LCD\_CMD = False

LCD\_LINE\_1 = 0x80 # LCD RAM address សម្រាប់ជួរទី១

# Timing constants

E\_PULSE = 0.0005 # លោត0.0005 វិនាទី

E\_DELAY = 0.0005 # Delay0.0005 វិនាទី

def main():

# Main program block

GPIO.setwarnings(False) # សម្រាប់ disable warning

GPIO.setmode(GPIO.BCM) # ប្រើលេខBCM GPIO

GPIO.setup(LCD\_E, GPIO.OUT) # Set pin E ជា​ output

GPIO.setup(LCD\_RS, GPIO.OUT) # Set pin RS ជា​ output

GPIO.setup(LCD\_D4, GPIO.OUT) # Set pin DB4 ជា​ output

GPIO.setup(LCD\_D5, GPIO.OUT) # Set pin DB5 ជា​ output

GPIO.setup(LCD\_D6, GPIO.OUT) # Set pin DB6 ជា​ output

GPIO.setup(LCD\_D7, GPIO.OUT) # Set pin DB7 ជា​ output

# Initialize display

lcd\_init()

while True:

lcd\_string("Hello World",LCD\_LINE\_1) #display hello world លើ​ LCD នៅជួរទី១

time.sleep(3) # delay 2 វិនាទី

def lcd\_init():

# Initialize display

lcd\_byte(0x33,LCD\_CMD) # 110011 Initialise

lcd\_byte(0x32,LCD\_CMD) # 110010 Initialise

lcd\_byte(0x06,LCD\_CMD) # 000110 Cursor move direction

lcd\_byte(0x0C,LCD\_CMD) # 001100 Display On,Cursor Off, Blink Off

lcd\_byte(0x28,LCD\_CMD) # 101000 Data length, number of lines, font size

lcd\_byte(0x01,LCD\_CMD) # 000001 Clear display

time.sleep(E\_DELAY)​​​ #delay នៅក្នុងដំណើរការ execution​ របស់​ program

def lcd\_byte(bits, mode):

# Send byte to data pins

# bits = data

# mode = True for character

# False for command

GPIO.output(LCD\_RS, mode) # Send RS ទៅ​ output pin

# High bits

GPIO.output(LCD\_D4, False)

GPIO.output(LCD\_D5, False)

GPIO.output(LCD\_D6, False)

GPIO.output(LCD\_D7, False)

if bits&0x10==0x10:

GPIO.output(LCD\_D4, True)

if bits&0x20==0x20:

GPIO.output(LCD\_D5, True)

if bits&0x40==0x40:

GPIO.output(LCD\_D6, True)

if bits&0x80==0x80:

GPIO.output(LCD\_D7, True)

# Toggle 'Enable' pin

lcd\_toggle\_enable()

# Low bits

GPIO.output(LCD\_D4, False)

GPIO.output(LCD\_D5, False)

GPIO.output(LCD\_D6, False)

GPIO.output(LCD\_D7, False)

if bits&0x01==0x01:

GPIO.output(LCD\_D4, True)

if bits&0x02==0x02:

GPIO.output(LCD\_D5, True)

if bits&0x04==0x04:

GPIO.output(LCD\_D6, True)

if bits&0x08==0x08:

GPIO.output(LCD\_D7, True)

# Toggle 'Enable' pin

lcd\_toggle\_enable()

def lcd\_toggle\_enable():

# Toggle enable

time.sleep(E\_DELAY)

GPIO.output(LCD\_E, True)

time.sleep(E\_PULSE)

GPIO.output(LCD\_E, False)

time.sleep(E\_DELAY)

# Send string សម្រាប់ display

def lcd\_string(message,line):

message = message.ljust(LCD\_WIDTH," ")

lcd\_byte(line, LCD\_CMD)

for i in range(LCD\_WIDTH):

lcd\_byte(ord(message[i]),LCD\_CHR)

if name == 'main':

try:

main()

except KeyboardInterrupt:

#មិនដំណើរការទៀត

pass

finally:

lcd\_byte(0x01, LCD\_CMD)

lcd\_string("Goodbye!",LCD\_LINE\_1)

GPIO.cleanup() #clean up port ទាំងអស់ដែលបានប្រើ