Embedded systems term project

# project info

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | NAME | | Title | |
| 20133277 | | Seongsil Heo | | Speed Game |

# Project description

It’s a speed game. There’s a 8\*8 LED Matrix. When you take turns to push the Red and Green button, it turns on the 8\*8 LED Matrix one by one. And when the all Matrix are turned on, the speed of your game is shown through LDE Matrix. If you push the yellow button, you could end the game, and restart.

# system overview

.

Add\_event\_detect

if

pushed == 64

If 20 pressed

exit

Try:

Except

Distplay.set\_pixel

Display.write\_display

Add\_event\_detect(16)

setScore()

Add\_event\_detect(21)

# 4. environment

1. sensors or disply modules I used : 8x8 Bi-color LED Matrix, switches

2. SW packages : RPi.GPIO, time, Adafruit\_LED\_Backpack

# 5. source code

import RPi.GPIO as GPIO  
import time  
  
from Adafruit\_LED\_Backpack import BicolorMatrix8x8  
  
global pushed  
pushed = 0  
global count  
count = 0  
global x  
x = 0  
global y  
y = 0

GPIO.setmode(GPIO.BCM)

#set 3 switches  
GPIO.setup(16, GPIO.IN, pull\_up\_down = GPIO.PUD\_UP)   
GPIO.setup(20, GPIO.IN, pull\_up\_down = GPIO.PUD\_UP)  
GPIO.setup(21, GPIO.IN, pull\_up\_down = GPIO.PUD\_UP)

display = BicolorMatrix8x8.BicolorMatrix8x8()  
  
display.begin()

#print speed  
ten = [[0,0,0,0,0,0,0,0],  
      [0,1,0,0,1,1,1,0],  
      [0,1,0,0,1,0,1,0],  
      [0,1,0,0,1,0,1,0],  
      [0,1,0,0,1,0,1,0],  
      [0,1,0,0,1,0,1,0],  
      [0,0,1,0,0,1,0,0],  
      [0,0,0,0,0,0,0,0]]

Eleven = [[0,0,0,0,0,0,0,0],  
         [0,0,1,0,0,1,0,0],  
         [0,0,1,0,0,1,0,0],  
         [0,0,1,0,0,1,0,0],  
         [0,0,1,0,0,1,0,0],  
         [0,0,1,0,0,1,0,0],  
         [0,0,1,0,0,1,0,0],  
         [0,0,0,0,0,0,0,0]]  
  
twelve = [[0,0,0,0,0,0,0,0],  
         [0,1,0,0,1,1,1,0],  
         [0,1,0,0,0,0,1,0],  
         [0,1,0,0,1,1,1,0],  
         [0,1,0,0,1,0,0,0],  
         [0,1,0,0,1,0,0,0],  
         [0,1,0,0,1,1,1,0],  
         [0,0,0,0,0,0,0,0]]

def setScore(ls,c):  
        for x in range(8):  
                for y in range(8):  
                        if ls[x][y] == 1:  
                                display.set\_pixel(x,y,c)  
                        display.write\_display()  
  
def my\_callback(channel):  
        global pushed  
        global count  
        global x  
        global y  
        pushed += 1  
  
        if pushed == 1:  
                count = time.time()  
        display.set\_pixel(x,y,1)  
        display.write\_display()  
        x += 1  
       if x == 8:  
                x = 0  
                y += 1  
        if pushed == 64:  
                pushed = 0  
                x = 0  
                y = 0  
                score = time.time() - count  
                score2 = int(score)  
                print "Time:%01f" % score

                count = 0  
  
                if score2 == 10:  
                        setScore(ten, BicolorMatrix8x8.RED)  
                elif score2 == 11:  
                        setScore(eleven, BicolorMatrix8x8.RED)  
                else:  
                        setScore(twelve, BicolorMatrix8x8.RED)  
                time.sleep(1)  
                print "If you want to end the game, press the yellow button."  
  
GPIO.add\_event\_detect(16, GPIO.FALLING, callback=my\_callback, bouncetime=300)  
GPIO.add\_event\_detect(21, GPIO.FALLING, callback=my\_callback, bouncetime=300)  
  
try:  
        GPIO.wait\_for\_edge(20,GPIO.RISING)  
except KeyboardInterrupt:  
        GPIO.cleanup()