Class – TE B

Title – Traffic Signal

Input –

# Import standard python libraries

import sys

import time

LED\_1 = (0 \* 32) + 3

LED\_2 = (0 \* 32) + 23

LED\_3 = (0 \* 32) + 2

LED\_4 = (0 \* 32) + 26

LED\_5 = (1 \* 32) + 17

LED\_6 = (1 \* 32) + 15

LED\_7 = (0 \* 32) + 15

LED\_8 = (1 \* 32) + 14

LED\_9 = (0 \* 32) + 14

LED\_10 = (0 \* 32) + 27

LED\_11 = (0 \* 32) + 22

LED\_12 = (2 \* 32) + 1

NORTH\_GREEN = LED\_1

NORTH\_YELLOW = LED\_5

NORTH\_RED = LED\_9

EAST\_GREEN = LED\_2

EAST\_YELLOW = LED\_6

EAST\_RED = LED\_10

SOUTH\_GREEN = LED\_3

SOUTH\_YELLOW = LED\_7

SOUTH\_RED = LED\_11

WEST\_GREEN = LED\_4

WEST\_YELLOW = LED\_8

WEST\_RED = LED\_12

# PATH of a GPIO specific sysfsinterfce directory on Linux system

SYSFS\_GPIO\_DIR = "/sys/class/gpio"

defgpioExport (gpio):

try:

fo = open(SYSFS\_GPIO\_DIR + "/export","w")

fo.write(gpio)

fo.close()

return

except IOError:

return

defgpioUnexport (gpio):

try:

fo = open(SYSFS\_GPIO\_DIR + "/unexport","w")

fo.write(gpio)

fo.close()

return

except IOError:

return

defgpioSetDir (gpio, flag):

try:

fo = open(SYSFS\_GPIO\_DIR + "/gpio" + gpio + "/direction" ,"w")

fo.write(flag)

fo.close()

return

except IOError:

return

defgpioSetVal (gpio, val):

try:

fo = open(SYSFS\_GPIO\_DIR + "/gpio" + gpio + "/value" ,"w")

fo.write(val)

fo.close()

return

except IOError:

return

deflightExit (gpio):

gpioSetVal(gpio, val="0")

gpioUnexport(gpio)

return

deflightInit (gpio):

gpioExport(gpio)

gpioSetDir(gpio, flag="out")

gpioSetVal(gpio, val="0")

return

deflightOn (gpio):

gpioSetVal(gpio, val="1")

return

deflightOff (gpio):

gpioSetVal(gpio, val="0")

return

deftrafficInitAll():

lightInit(str(NORTH\_GREEN))

lightInit(str(NORTH\_YELLOW))

lightInit(str(NORTH\_RED))

lightInit(str(EAST\_GREEN))

lightInit(str(EAST\_YELLOW))

lightInit(str(EAST\_RED))

lightInit(str(SOUTH\_GREEN))

lightInit(str(SOUTH\_YELLOW))

lightInit(str(SOUTH\_RED))

lightInit(str(WEST\_GREEN))

lightInit(str(WEST\_YELLOW))

lightInit(str(WEST\_RED))

return

deftrafficExitAll():

lightExit(str(NORTH\_GREEN))

lightExit(str(NORTH\_YELLOW))

lightExit(str(NORTH\_RED))

lightExit(str(EAST\_GREEN))

lightExit(str(EAST\_YELLOW))

lightExit(str(EAST\_RED))

lightExit(str(SOUTH\_GREEN))

lightExit(str(SOUTH\_YELLOW))

lightExit(str(SOUTH\_RED))

lightExit(str(WEST\_GREEN))

lightExit(str(WEST\_YELLOW))

lightExit(str(WEST\_RED))

return

defnorthSouthOn():

lightOff(str(EAST\_YELLOW))

lightOff(str(WEST\_YELLOW))

lightOff(str(NORTH\_RED))

lightOff(str(SOUTH\_RED))

lightOn(str(EAST\_RED))

lightOn(str(WEST\_RED))

lightOn(str(NORTH\_GREEN))

lightOn(str(SOUTH\_GREEN))

time.sleep(10)

lightOff(str(NORTH\_GREEN))

lightOff(str(SOUTH\_GREEN))

lightOn(str(NORTH\_YELLOW))

lightOn(str(SOUTH\_YELLOW))

time.sleep(1)

return

defeastWestOn():

lightOff(str(NORTH\_YELLOW))

lightOff(str(SOUTH\_YELLOW))

lightOff(str(EAST\_RED))

lightOff(str(WEST\_RED))

lightOn(str(NORTH\_RED))

lightOn(str(SOUTH\_RED))

lightOn(str(EAST\_GREEN))

lightOn(str(WEST\_GREEN))

time.sleep(10)

lightOff(str(EAST\_GREEN))

lightOff(str(WEST\_GREEN))

lightOn(str(EAST\_YELLOW))

lightOn(str(WEST\_YELLOW))

time.sleep(1)

return

try:

print "\nTraffic Signal Light Simulation using Python"

print "-----------------------------------------------"

trafficExitAll() # Cleanup all the LED GPIOs first

trafficInitAll() # Initialize all LEDs and Turn them OFF

for x in range(0, 10): # Loop for 10 times

print "\nNORTH-SOUTH --> [GO]" # Prints

print "EAST-WEST --> [STOP]\n" # Prints

northSouthOn() # Call function to make NORTH-SOUTH traffic ON

time.sleep(1) # Sleep for 1 second

print "\nEAST-WEST --> [GO]" # Prints

print "NORTH-SOUTH --> [STOP]\n" # Prints

eastWestOn() # Call function to make EAST-WEST traffic ON

time.sleep(1) # Sleep for 1 second

trafficExitAll() # Cleanup all the LED pins

print "done" # Prints

exit() # Exit from Program

except KeyboardInterrupt: # CTRL-C Exception Handler to cleanup and exit safely from program

trafficExitAll()

print "Program Exit due to CTRL-C"

exit()

sys.exit(0)