# ASSIGNMENT NO. :13 BATCH NO. : T4 ROLL NO. :

# TITLE :

# Develop Robotics(stepper motor) Application using Beagle Board.

**# stepper.py**

#!/usr/bin/python

import sys

import time

import select

from stepperpins import \*

from gpio import \*

def stepperExit (gpio):

gpioSetVal(gpio, val="0")

gpioUnexport(gpio)

return

def stepperInit (gpio):

gpioExport(gpio)

gpioSetDir(gpio, flag="out")

gpioSetVal(gpio, val="0")

return

# This function turns the stepper PIN ON

# It will setup a PIN as GPIO, sets the direction as OUT and then makes the PIN high

def stepperOn (gpio):

gpioSetVal(gpio, val="1")

return

# This function turns the stepper PIN OFF

# It will setup a PIN as GPIO, sets the direction as OUT and then makes the PIN low

def stepperOff (gpio):

gpioSetVal(gpio, val="0")

return

def stepperInitAll():

stepperInit(str(STEPPER\_1))

stepperInit(str(STEPPER\_2))

stepperInit(str(STEPPER\_3))

stepperInit(str(STEPPER\_4))

def stepperExitAll():

stepperExit(str(STEPPER\_1))

stepperExit(str(STEPPER\_2))

stepperExit(str(STEPPER\_3))

stepperExit(str(STEPPER\_4))

print "\n=== Demonstration END ===\n"

return

# This function writes the sequence 5(5 in binary "0101") on stepper motor pins

# While writing start from LSB, hence LSB goes to stepper pin #1

def stepperSeq5():

stepperOn(str(STEPPER\_1)) # LSB first i.e. write 1 on STEPPER\_1

time.sleep(0.0001) # pause for 100 micro seconds

stepperOff(str(STEPPER\_2)) # Then next bit i.e. write 0 on STEPPER\_2

time.sleep(0.0001)

stepperOn(str(STEPPER\_3)) # Next bit i.e. write 1 on STEPPER\_3

time.sleep(0.0001)

stepperOff(str(STEPPER\_4)) # Final bit (MSB) i.e. write 0 on STEPPER\_4

time.sleep(0.0001)

return

# This function writes the sequence 9(9 in binary "1001") on stepper motor pins

# While writing start from LSB, hence LSB goes to stepper pin #1

def stepperSeq9():

stepperOn(str(STEPPER\_1)) # LSB first i.e. write 1 on STEPPER\_1

time.sleep(0.0001) # pause for 100 micro seconds

stepperOff(str(STEPPER\_2)) # Then next bit i.e. write 0 on STEPPER\_2

time.sleep(0.0001)

stepperOff(str(STEPPER\_3)) # Next bit i.e. write 0 on STEPPER\_3

time.sleep(0.0001)

stepperOn(str(STEPPER\_4)) # Final bit (MSB) i.e. write 1 on STEPPER\_4

time.sleep(0.0001)

return

# This function writes the sequence 10(10 in binary "1010") on stepper motor pins

# While writing start from LSB, hence LSB goes to stepper pin #1

def stepperSeq10():

stepperOff(str(STEPPER\_1)) # LSB first i.e. write 0 on STEPPER\_1

time.sleep(0.0001) # pause for 100 micro seconds

stepperOn(str(STEPPER\_2)) # Then next bit i.e. write 1 on STEPPER\_2

time.sleep(0.0001)

stepperOff(str(STEPPER\_3)) # Next bit i.e. write 0 on STEPPER\_3

time.sleep(0.0001)

stepperOn(str(STEPPER\_4)) # Final bit (MSB) i.e. write 1 on STEPPER\_4

time.sleep(0.0001)

return

# This function writes the sequence 6(6 in binary "0110") on stepper motor pins

# While writing start from LSB, hence LSB goes to stepper pin #1

def stepperSeq6():

stepperOff(str(STEPPER\_1)) # LSB first i.e. write 0 on STEPPER\_1

time.sleep(0.0001) # pause for 100 micro seconds

stepperOn(str(STEPPER\_2)) # Then next bit i.e. write 1 on STEPPER\_2

time.sleep(0.0001)

stepperOn(str(STEPPER\_3)) # Next bit i.e. write 1 on STEPPER\_3

time.sleep(0.0001)

stepperOff(str(STEPPER\_4)) # Final bit (MSB) i.e. write 0 on STEPPER\_4

time.sleep(0.0001)

return

# This function rotates the stepper motor one step in left direction (anti-clockwise)

def stepperDirLeft():

stepperSeq5()

time.sleep(0.01)

stepperSeq9()

time.sleep(0.01)

stepperSeq10()

time.sleep(0.01)

stepperSeq6()

time.sleep(0.01)

return

# This function rotates the stepper motor one step in right direction (clockwise)

def stepperDirRight():

stepperSeq6()

time.sleep(0.01)

stepperSeq10()

time.sleep(0.01)

stepperSeq9()

time.sleep(0.01)

stepperSeq5()

time.sleep(0.01)

return

try:

print "\nStepper Motor Driver using Python\n"

print "-----------------------------------------------\n"

stepperInitAll()

while True:

for i in range(0, 12): # for 12 times (to comeplete on revolution)

stepperDirLeft() # Rotate stepper left

time.sleep(3) # Sleep for 3 seconds

for i in range(0, 12):

stepperDirRight() # Rotate stepper right

time.sleep(3)

stepperExitAll

exit() # Exit from program

except KeyboardInterrupt:

stepperExitAll()

print "Program Exit due to CTRL-C"

exit()

sys.exit(0)