

GPIO 프로그램

[gpioControl.py]

gpioControl.py

```
import RPi.GPIO as GPIO
from time import sleep

ina1 = 33
ina2 = 35
ena = 37

inb1 = 31
inb2 = 29
enb = 23

fire = 16

UpDownCam=12
RightLeftCam=18

GPIO.setmode(GPIO.BOARD)

GPIO.setup(UpDownCam, GPIO.OUT, initial=1)
GPIO.setup(RightLeftCam, GPIO.OUT, initial=1)
GPIO.setup(11, GPIO.OUT, initial=1) # light

GPIO.setup(ina1,GPIO.OUT)
GPIO.setup(ina2,GPIO.OUT)
GPIO.setup(ena,GPIO.OUT)

GPIO.setup(inb1,GPIO.OUT)
GPIO.setup(inb2,GPIO.OUT)
GPIO.setup(enb,GPIO.OUT)

p1 = GPIO.PWM(RightLeftCam, 50) # 50 Hz
p2 = GPIO.PWM(UpDownCam, 50) # 50
p1.start(0)
p2.start(0)
p1.ChangeDutyCycle(0)
p2.ChangeDutyCycle(0)
```

```
pa=GPIO.PWM(ena,1000)
pa.start(25)

pb=GPIO.PWM(enb,1000)
pb.start(25)

initVerticalVal = 6.9
initHorizontalVal = 6.6

verticalVal = initVerticalVal
horizontalVal = initHorizontalVal

cameraPositionX = initVerticalVal
cameraPositionY = initHorizontalVal

# Set up camera constants
IM_WIDTH = 640
IM_HEIGHT = 480

class GpioControl(object):

    def __init__(self):
        global verticalVal
        global horizontalVal
        global p1
        global p2
        p1.ChangeDutyCycle(verticalVal)
        p2.ChangeDutyCycle(horizontalVal)
        sleep(0.1)
        p1.ChangeDutyCycle(0)
        p2.ChangeDutyCycle(0)
        print("> Init Vert=" + str(verticalVal) + ",Hort=" + str(horizontalVal))

    def click(self):
        GPIO.output(11, GPIO.LOW)
        sleep(0.5)
        GPIO.output(11, GPIO.HIGH)
        sleep(1)

    def __del__(self):
        global p1
        global p2
        p1.stop()
        p2.stop()
        print(" GPIO.__del__() ")
        GPIO.cleanup()

    def startUp(self):
        GPIO.setmode(GPIO.BOARD)
```

```

GPIO.setup(UpDownCam, GPIO.OUT, initial=1)
GPIO.setup(RightLeftCam, GPIO.OUT, initial=1)

GPIO.setup(ina1,GPIO.OUT)
GPIO.setup(ina2,GPIO.OUT)
GPIO.setup(ena,GPIO.OUT)

GPIO.setup(inb1,GPIO.OUT)
GPIO.setup(inb2,GPIO.OUT)
GPIO.setup(enb,GPIO.OUT)

p1.start(0)
p2.start(0)
p1.ChangeDutyCycle(0)
p2.ChangeDutyCycle(0)
pa.start(25)
pb.start(25)

def cleanUp(self):
    global p1
    global p2
    p1.stop()
    p2.stop()
    print(" GPIO.cleanup() ")
    GPIO.cleanup()
    sleep(2)

def initMotorPosition(self):
    # Init
    global p1
    global p2
    p1.ChangeDutyCycle(cameraPositionX)
    p2.ChangeDutyCycle(cameraPositionY)
    sleep(0.1)
    p1.ChangeDutyCycle(0)
    p2.ChangeDutyCycle(0)
    print("> Init Vert=" + str(cameraPositionX) + ",Hort=" + str(cameraPositionY))
    return "Vert=" + str(cameraPositionX) + ",Hort=" + str(cameraPositionY)

def moveUp(self):
    global verticalVal
    global horizontalVal
    global p2
    if(verticalVal<12):
        verticalVal = round(verticalVal+0.2, 1)
        p2.ChangeDutyCycle(verticalVal)
        print("> UP Vert=" + str(verticalVal) + ",Hort=" + str(horizontalVal))
        sleep(0.1)
        p2.ChangeDutyCycle(0)
    return "Vert=" + str(verticalVal) + ",Hort=" + str(horizontalVal)

```

```

def moveDown(self):
    global verticalVal
    global horizontalVal
    global p2
    if(verticalVal>5):
        verticalVal = round(verticalVal-0.2, 1)
        p2.ChangeDutyCycle(verticalVal)
        print("> Down Vert=" + str(verticalVal) + ",Hort=" + str(horizontalVal))
        sleep(0.1)
        p2.ChangeDutyCycle(0)
    return "Vert=" + str(verticalVal) + ",Hort=" + str(horizontalVal)

def moveRight(self):
    global verticalVal
    global horizontalVal
    global p1
    if(horizontalVal>2.5):
        horizontalVal = round(horizontalVal-0.2, 1)
        p1.ChangeDutyCycle(horizontalVal)
        print("> Right Vert=" + str(verticalVal) + ",Hort=" + str(horizontalVal))
        sleep(0.1)
        p1.ChangeDutyCycle(0)
    return "Vert=" + str(verticalVal) + ",Hort=" + str(horizontalVal)

def moveLeft(self):
    global verticalVal
    global horizontalVal
    global p1
    if(horizontalVal<10):
        horizontalVal = round(horizontalVal+0.2, 1)
        p1.ChangeDutyCycle(horizontalVal)
        print("> Left Vert=" + str(verticalVal) + ",Hort=" + str(horizontalVal))
        sleep(0.1)
        p1.ChangeDutyCycle(0)
    return "Vert=" + str(verticalVal) + ",Hort=" + str(horizontalVal)

def goForward(self,time):
    print("Go forward")
    pa.ChangeDutyCycle(75)
    GPIO.output(ina1,GPIO.HIGH)
    GPIO.output(ina2,GPIO.LOW)
    pb.ChangeDutyCycle(75)
    GPIO.output(inb1,GPIO.HIGH)
    GPIO.output(inb2,GPIO.LOW)
    sleep(0.1*int(time))
    GPIO.output(ina1,GPIO.LOW)
    GPIO.output(ina2,GPIO.LOW)
    GPIO.output(inb1,GPIO.LOW)

```

```

GPIO.output(inb2,GPIO.LOW)
return " >> Go Forward"

def goBackward(self,time):
    print("Go backward")
    pa.ChangeDutyCycle(75)
    GPIO.output(ina1,GPIO.LOW)
    GPIO.output(ina2,GPIO.HIGH)
    pb.ChangeDutyCycle(75)
    GPIO.output(inb1,GPIO.LOW)
    GPIO.output(inb2,GPIO.HIGH)
    sleep(0.1*int(time))
    GPIO.output(ina1,GPIO.LOW)
    GPIO.output(ina2,GPIO.LOW)
    GPIO.output(inb1,GPIO.LOW)
    GPIO.output(inb2,GPIO.LOW)
    return " >> Go Backward"

def goTurnleftback(self,time):
    print("Go Turn Left back")
    pa.ChangeDutyCycle(75)
    GPIO.output(inb1,GPIO.LOW)
    GPIO.output(inb2,GPIO.HIGH)
    sleep(0.1*int(time))
    GPIO.output(ina1,GPIO.LOW)
    GPIO.output(ina2,GPIO.LOW)
    GPIO.output(inb1,GPIO.LOW)
    GPIO.output(inb2,GPIO.LOW)
    return " >> Go Turn Left back"

def goTurnrightback(self,time):
    print("Go Turn Right back")
    pb.ChangeDutyCycle(75)
    GPIO.output(ina1,GPIO.LOW)
    GPIO.output(ina2,GPIO.HIGH)
    sleep(0.1*int(time))
    GPIO.output(ina1,GPIO.LOW)
    GPIO.output(ina2,GPIO.LOW)
    GPIO.output(inb1,GPIO.LOW)
    GPIO.output(inb2,GPIO.LOW)
    return " >> Go Turn Right back"

def goTurnleftforward(self,time):
    print("Go Turn Left forward")
    pa.ChangeDutyCycle(75)
    GPIO.output(inb1,GPIO.HIGH)
    GPIO.output(inb2,GPIO.LOW)
    sleep(0.1*int(time))
    GPIO.output(ina1,GPIO.LOW)
    GPIO.output(ina2,GPIO.LOW)

```

```

GPIO.output(inb1,GPIO.LOW)
GPIO.output(inb2,GPIO.LOW)
return " >> Go Turn Left forward"

def goTurnrightforward(self,time):
    print("Go Turn Right forward")
    pb.ChangeDutyCycle(75)
    GPIO.output(ina1,GPIO.HIGH)
    GPIO.output(ina2,GPIO.LOW)
    sleep(0.1*int(time))
    GPIO.output(ina1,GPIO.LOW)
    GPIO.output(ina2,GPIO.LOW)
    GPIO.output(inb1,GPIO.LOW)
    GPIO.output(inb2,GPIO.LOW)
    return " >> Go Turn Right forward"

def move_to_position(self,ObjX, ObjY):
    global cameraPositionX
    global cameraPositionY
    global p1
    global p2
    print(" >> Move to location x="+str(ObjX)+", y="+str(ObjY))
    moveLoop = True
    movX = int(IM_WIDTH/2)-ObjX
    movY = int(IM_HEIGHT/2)-ObjY
    print(" >> Center location movX="+str(movX)+", movY="+str(movY))

    xx = 1
    xy = 0
    yx = 1
    yy = 0
    while(moveLoop):
        if( xy < abs(movX) ):
            p1.ChangeDutyCycle(cameraPositionX)
            sleep(0.1)
            p1.ChangeDutyCycle(0)
            print("xx=" + str(xx) + ", xy="+ str(xy) + " cameraPositionX="
+str(round(cameraPositionX,1)))
            xy = xx*xx * 12
            xx = xx + 1
            if(movX > 0): cameraPositionX = cameraPositionX - 0.2
            else: cameraPositionX = cameraPositionX + 0.2
        if( yy < abs(movY) ):
            p2.ChangeDutyCycle(cameraPositionY)
            sleep(0.1)
            p2.ChangeDutyCycle(0)
            print("yx=" + str(yx) + ", yy="+ str(yy) + " cameraPositionY="

```

```

+str(round(cameraPositionY,1)))
        yy = yx*yx * 12
        yx = yx + 1
        if(movY > 0): cameraPositionY = cameraPositionY + 0.2
        else: cameraPositionY = cameraPositionY - 0.2
    elif( xy >= movX and yy >= movY):
        print(" >> Center location movX="+str(movX)+" , movY=" + str(movY))
        print(" >> Position location xy="+str(xy)
              +", yy="+str(yy)
              +" cameraPositionX="+str(round(cameraPositionX,1))+
              " cameraPositionY="+str(round(cameraPositionY,1))+ " .. ")
        moveLoop = False

def shooting(self):
    GPIO.setup(fire,GPIO.OUT)
    GPIO.output(fire,GPIO.HIGH)
    sleep(0.5)
    GPIO.output(fire,GPIO.LOW)
    GPIO.cleanup()
    self.startUp()
    return "Fired !"

```

🕒Revision #1

★Created 30 December 2023 13:53:29 by Hyeon Su Ryu

✎Updated 30 December 2023 14:23:53 by Hyeon Su Ryu