임베디드 응용 및 실습 과제5

- 5. Bluetooth serial

학과	전기공학과
학년	3학년
학번	2020161110
이름	전현서
과목명	임베디드 응용 및 실습
분반	01분반
담당교수님	우성민 교수님

```
import threading
import serial
import time
import RPi.GPIO as GPIO
PWMA =18
PWMB =23
AIN1 =22
AIN2 =27
BIN1 =25
BIN2 =24
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(PWMA,GPIO.OUT)
GPIO.setup(PWMB,GPIO.OUT)
GPIO.setup(AIN1,GPIO.OUT)
GPIO.setup(AIN2,GPIO.OUT)
GPIO.setup(BIN1,GPIO.OUT)
GPIO.setup(BIN2,GPIO.OUT)


L_Motor =GPIO.PWM(PWMA,500)
R_Motor =GPIO.PWM(PWMB,500)
L_Motor.start(0)
R_Motor.start(0)
Motor_Control =[0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0]
bleSerial =serial.Serial("/dev/ttyS0", baudrate=9600, timeout=1.0)
gData =""
def serial_thread():
    global gData
    while True:
        data =bleSerial.readline()
        data =data.decode()
        gData =data
def main():
    global gData
    try:
        while True:
            if  gData.find("B0") >=0:
                gData =""
                print("ok go")
                GPIO.output(AIN1,Motor_Control[4*1])
                GPIO.output(AIN2,Motor_Control[4*1+1])
                GPIO.output(BIN1,Motor_Control[4*1+2])
                GPIO.output(BIN2,Motor_Control[4*1+3])
                L_Motor.ChangeDutyCycle(100)
                R_Motor.ChangeDutyCycle(100)
                time.sleep(0.05)
            elif gData.find("B3") >=0:
```

```
                gData =""
                print("ok back")
                GPIO.output(AIN1,Motor_Control[4*4])
                GPIO.output(AIN2,Motor_Control[4*4+1])
                GPIO.output(BIN1,Motor_Control[4*4+2])
                GPIO.output(BIN2,Motor_Control[4*4+3])
                L_Motor.ChangeDutyCycle(100)
                R_Motor.ChangeDutyCycle(100)
                time.sleep(0.05)
            elif gData.find("B1") >=0:
                gData =""
                print("ok left")
                GPIO.output(AIN1,Motor_Control[4*3])
                GPIO.output(AIN2,Motor_Control[4*3+1])
                GPIO.output(BIN1,Motor_Control[4*3+2])
                GPIO.output(BIN2,Motor_Control[4*3+3])
                L_Motor.ChangeDutyCycle(100)
                R_Motor.ChangeDutyCycle(100)
                time.sleep(0.05)
            elif gData.find("B2") >=0:
                gData =""
                print("ok right")
                GPIO.output(AIN1,Motor_Control[4*2])
                GPIO.output(AIN2,Motor_Control[4*2+1])
                GPIO.output(BIN1,Motor_Control[4*2+2])
                GPIO.output(BIN2,Motor_Control[4*2+3])
                L_Motor.ChangeDutyCycle(100)
                R_Motor.ChangeDutyCycle(100)
                time.sleep(0.05)
            elif gData.find("B4") >=0:
                gData =""
                print("ok stop")
                GPIO.output(AIN1,Motor_Control[4*0])
                GPIO.output(AIN2,Motor_Control[4*0+1])
                GPIO.output(BIN1,Motor_Control[4*0+2])
                GPIO.output(BIN2,Motor_Control[4*0+3])
                L_Motor.ChangeDutyCycle(0)
                R_Motor.ChangeDutyCycle(0)
    except KeyboardInterrupt:
        pass
if __name__=='__main__':
    task1 =threading.Thread(target =serial_thread)
    task1.start()
    main()
    bleSerial.close()
```