CODE [ags.py] [Python3]

```
from gpiozero import LightSensor
from numpy import interp
import Adafruit_CharLCD as LCD
import RPi.GPIO as GPIO
import time
import spidev
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)


# Firebase Configuration
import pyrebase

config = {
        "apiKey":"<api-key>",
        "authDomain": "automatic-gardening-system.firebaseapp.com", "databaseURL":"https://automatic-
        gardening-system.firebaseio
.com/gardening-data",
        "storageBucket": "automatic-gardening-system.appspot.com"
}

firebase = pyrebase.initialize_app(config)
db = firebase.database()

def send_data(id, data, parent):
   par = db
   if (len(parent) > 0):
      for p in parent:
         par = par.child(p)
   par.child(id).set(data)
   print("Sent ", id, " = ", data)

def get_data(id):
   data = db.child(id).get().val()
   print(id, "=", data)
   return data
custom_control = get_data("customControl")
```

```
# Moisture_Sensor_Setup
threshold = 10
# Start_SPI_Connection
spi = spidev.SpiDev()
spi.open(0,0)

# Read_MCP3008_Data
def analogInput(channel):
  spi.max_speed_hz = 1350000
  adc = spi.xfer2([1,(8+channel)<<4,0])
  data = ((adc[1]&3) << 8) + adc[2]
  return data

# LCD_Setup

# Raspberry_Pi_Pin_Setup
lcd_rs = 25
lcd_en = 24
lcd_d4 = 23
lcd_d5 = 17
lcd_d6 = 18
lcd_d7 = 22
lcd_backlight = 2

# LCD_Rows_And_Columns
lcd_rows = 2
lcd_columns = 16

lcd = LCD.Adafruit_CharLCD(lcd_rs, lcd_en, lcd_d4, lcd_d5, lcd_d6, lcd_d7, lcd_columns, lcd_rows,
lcd_backlight)

# Pin_Declaration
ldr = 19
buzzer = 26
motor = 12
tank_level = 13
ldr_red_led = 16
moisture_blue_led = 20
tank_green_led = 21
```

```
# GPIO_Setup
GPIO.setup(buzzer,GPIO.OUT)
GPIO.setup(motor,GPIO.OUT)
GPIO.setup(tank_level,GPIO.OUT)
GPIO.setup(ldr_red_led,GPIO.OUT) GPIO.setup(moisture_blue_led,GPIO.OUT)
GPIO.setup(tank_green_led,GPIO.OUT)

# Default_Pin_Setup
GPIO.output(buzzer,GPIO.LOW)
GPIO.output(motor,GPIO.LOW)
GPIO.output(ldr_red_led,GPIO.LOW)
GPIO.output(moisture_blue_led,GPIO.LOW)
GPIO.output(tank_green_led,GPIO.LOW)

# Values
ldr_read = LightSensor(ldr)
ldr_value = (ldr_read.value)*100
tank_value = GPIO.input(tank_level)
moisture_value = int(interp(analogInput(0), [0, 1023], [100, 0]))

# Main_Code
lcd.message("    Automatic   ")
lcd.message("\nGardening System")
time.sleep(5)
lcd.clear()
while True:
   lcd.clear()
   ldr_value = (ldr_read.value)*100
   send_data("lightIntensity", ldr_value/100, [])
   customControl = get_data("customControl")
   if(customControl):
      led_state = get_data("ledState")
      if (led_state):
         GPIO.output(ldr_red_led, GPIO.HIGH)
         print("Custom control: ON LIGHT")
      else:
                     GPIO.output(ldr_red_led, GPIO.LOW)
          print("Custom control: OFF LIGHT")
```

```
            elif (ldr_value < 50):
        GPIO.output(ldr_red_led,GPIO.HIGH)
        print(ldr_value)
        time.sleep(0.5)
       else:
        GPIO.output(ldr_red_led,GPIO.LOW)
        print(ldr_value)
        time.sleep(0.5)

moisture_value = int(interp(analogInput(0), [0, 1023], [100, 0]))
send_data("moisture", moisture_value, [])

  if(moisture_value < threshold):
     GPIO.output(moisture_blue_led, GPIO.LOW)
     lcd.message("  Moisture LOW  ")
  else:
     GPIO.output(moisture_blue_led, GPIO.HIGH)
     lcd.message("  Moisture HIGH ")

  tank_value = GPIO.input(tank_level)
  send_data("waterLevel", 1, [])

  if(tank_value==1):
     GPIO.output(tank_green_led, GPIO.LOW)
     lcd.message("\nTank Level HIGH ")
  else:
     GPIO.output(tank_green_led,GPIO.HIGH)
     lcd.message("\n Tank Level LOW ")

  if(customControl):
     motor = get_data("landMotor")
     waterLevel = get_data("waterLevel")
     print("Read Water Level = ", waterLevel)
     waterLevel = 1
     while(customControl and motor and waterLevel):
        print("Custom Control: MOTOR ON")
        lcd.clear()
        lcd.message("  Motor ON  \n")
        lcd.message("  Tank Filled ")
        GPIO.output(buzzer, GPIO.HIGH)
```

```
            time.sleep(1)
            GPIO.output(buzzer, GPIO.LOW)
            customControl = get_data("customControl")
                        if(not customControl):
               break
            motor = get_data("landMotor")
            waterLevel = get_data("waterLevel")

    elif(moisture_value<threshold and tank_value==1):
        while(moisture_value<threshold and tank_value==1):
            lcd.clear()
            lcd.message("  Moisture LOW  ")
            lcd.message("\n    Motor ON    ")
            GPIO.output(buzzer,GPIO.HIGH)
            time.sleep(1)
            GPIO.output(buzzer,GPIO.LOW)
            time.sleep(1)
            GPIO.output(motor,GPIO.HIGH)
            print("Motor  On")
moisture_value = int(interp(analogInput(0), [0, 1023], [100, 0]))
send_data("moisture", moisture_value, [])
              if(moisture_value>threshold):
            lcd.clear()
            lcd.message("  Moisture HIGH ")
            lcd.message("\n    Motor OFF   ")
            GPIO.output(moisture_blue_led, GPIO.HIGH)
            GPIO.output(motor, GPIO.LOW)
            time.sleep(2)
moisture_value = int(interp(analogInput(0), [0, 1023], [100, 0]))
send_data("moisture", moisture_value, [])
              if(tank_value==0):
            lcd.clear()
            lcd.message(" Tank Level LOW ")
            lcd.message("\n    Motor OFF   ")
            GPIO.output(tank_green_led,GPIO.HIGH)
            GPIO.output(motor,GPIO.LOW)
            time.sleep(2)
            tank_value = GPIO.input(tank_level)
            send_data("waterLevel", tank_value, [])
    time.sleep(1)
```