/\*

--------------------------------------------------------------------

working code

sensors + fan (on and off) and iot dim

(best

15-11-24)

--------------------------------------------------------------------

+\*/

#include "arduino\_secrets.h"

/\*

  Sketch generated by the Arduino IoT Cloud Thing "Untitled"

  https://create.arduino.cc/cloud/things/74317b6c-7e57-4076-89e2-0ea3f0728145

  Arduino IoT Cloud Variables description

  The following variables are automatically generated and updated when changes are made to the Thing

  CloudLight led;

  CloudDimmedLight led2;  // Changed to CloudDimmedLight for brightness control

  int fan;

  Variables marked as READ/WRITE in the Cloud Thing will also have functions

  which are called when their values are changed from the Dashboard.

  These functions are generated with the Thing and added at the end of this sketch.

\*/

#include "thingProperties.h"

#define LED\_PIN 4               // Pin connected to the LED

#define LED2\_PIN 13             // Pin connected to the second LED for brightness control

#define IR\_SENSOR\_PIN 21        // Pin connected to the IR sensor

#define PHOTORESISTOR\_PIN 15    // Pin connected to the photoresistor

#define SOUND\_SENSOR\_PIN 18     // Pin connected to the sound sensor

#define FAN\_CONTROL\_PIN 22

#define FAN\_speed 23

#    // Pin connected to control the fan speed

bool ledState = LOW;            // Stores current LED state (on or off)

unsigned long lastClapTime = 0; // Stores the last time a clap was detected

const unsigned long clapInterval = 200;  // Minimum time between claps to avoid noise

bool irTriggered = false;       // Tracks if IR sensor has triggered the LED

void setup() {

  // Initialize serial and wait for port to open:

  Serial.begin(9600);

  pinMode(LED\_PIN, OUTPUT);              // Set LED pin as output

  pinMode(LED2\_PIN, OUTPUT);             // Set LED2 pin as output for brightness control

  pinMode(SOUND\_SENSOR\_PIN, INPUT);      // Set sound sensor pin as input

  pinMode(IR\_SENSOR\_PIN, INPUT);         // Set IR sensor pin as input

  pinMode(PHOTORESISTOR\_PIN, INPUT);     // Set photoresistor pin as input

  digitalWrite(LED\_PIN, ledState);       // Set initial LED state

pinMode(FAN\_CONTROL\_PIN, OUTPUT);

pinMode(FAN\_speed, OUTPUT);

digitalWrite(FAN\_CONTROL\_PIN, LOW);

//delay(5000);

  // This delay gives the chance to wait for Serial Monitor without blocking if none is found

  delay(1500);

  // Defined in thingProperties.h

  initProperties();

  // Connect to Arduino IoT Cloud

  ArduinoCloud.begin(ArduinoIoTPreferredConnection);

  setDebugMessageLevel(2);

  ArduinoCloud.printDebugInfo();

}

void loop() {

  // Update the IoT Cloud connection

  ArduinoCloud.update();

  int soundValue = digitalRead(SOUND\_SENSOR\_PIN);       // Read sound sensor value

  int irValue = digitalRead(IR\_SENSOR\_PIN);             // Read IR sensor value

  int photoresistorValue = digitalRead(PHOTORESISTOR\_PIN);  // Read photoresistor value

  // Only allow the IR sensor to toggle the LED when the photoresistor is not active

  if (photoresistorValue == HIGH) {

    if (irValue == HIGH && !irTriggered) {

      // Toggle LED state on IR detection and mark it as triggered

      ledState = !ledState;

      digitalWrite(LED\_PIN, ledState);

      irTriggered = true;

      Serial.println("LED toggled by IR sensor.");

    } else if (irValue == LOW) {

      // Reset IR trigger when IR sensor is no longer detecting

      irTriggered = false;

    }

  } else {

    // If the photoresistor is triggered, disable LED control through the sound sensor

    Serial.println("Photoresistor active: LED control disabled.");

  }

  // Sound sensor logic for toggling LED on clap, only if photoresistor is not active

  if (soundValue == HIGH && photoresistorValue == HIGH) {

    unsigned long currentTime = millis();

    if (currentTime - lastClapTime > clapInterval) {

      ledState = !ledState;          // Toggle LED state on clap

      digitalWrite(LED\_PIN, ledState);

      lastClapTime = currentTime;

      Serial.println("Clap detected! Toggling LED.");

    }

  }

}

/\*

  Since Led is a READ\_WRITE variable, onLedChange() is

  executed every time a new value is received from the IoT Cloud.

\*/

void onLedChange() {

  if (led) {

    digitalWrite(LED\_PIN, HIGH);

  } else {

    digitalWrite(LED\_PIN, LOW);

  }

}

/\*

  Since Led2 is a CloudDimmedLight, onLed2Change() is

  executed every time a new value is received from the IoT Cloud.

  It adjusts the brightness based on the value of led2.

\*/

void onLed2Change() {

  int led2Brightness = led2.getBrightness();  // Get the brightness value from CloudDimmedLight

  if (led2Brightness > 0) {

    analogWrite(LED2\_PIN, map(led2Brightness, 0, 100, 0, 255)); // Map brightness to PWM range2

  } else {

    digitalWrite(LED2\_PIN, LOW);  // Turn off the LED if brightness is 0

  }

}

/\*

  Since Fan is a READ\_WRITE variable, onFanChange() is

  executed every time a new value is received from the IoT Cloud.

\*/

void onFanChange() {

  analogWrite(FAN\_speed, fan);

  // Code to control fan speed based on dashboard input

}

//ecuted every time a new value is received from IoT Cloud.

void onLed3Change()  {

  if (led3) {

    digitalWrite(22, HIGH);

  } else {

    digitalWrite(22, LOW);

  }

  // Add your code here to act upon Led3 change

}

void onRgbLedChange() {}