Sistem de Iluminare pentru curte

Introducere

In viata de zi cu zi se intampla foarte des sa uiti lumina aprinsa undeva. Cel mai enervant este cand deja esti pregatit sa dormi si iti dai seama ca ai uitat lumina aprinsa afara si trebuie sa cobori, jumatate adormit, sa stingi luminile din toate curtea.

De aceea, dupa multe rugaminte primite de la tata, am decis sa creez un sistem de iluminare inteligent ce se aprinde automat in jurul orei apusului si se dezactiveaza dupa 4 ore, astfel neavand grija ca va uita vreo lumina aprinsa. Mai mult, pentru a salva din curentul folosit, senzorii de miscare conectati la aparat vor detecta cand cineva e in curte si, astfel, va aprinde lumina cand e nevoie. In functie de senzorul ce detecteaza miscarea, se vor activa doar luminile din zona acelui senzor.

Desigur, daca e nevoie de lumina in afara intervalului, sistemul poate fi activat prin trimiterea comenzii de aprindere catre Google Assistant, care va notifica microcontrolerul sa se activeze.

Descriere generală

Schema Bloc

- Telefon:- trimite comanda prin intermediul Google Assistant la ESP
- Senzor de miscare:- anunta cand e activitate ⇒ ESP32 porneste luminile
- Server Web:- ESP32 trimite cereri HTTPS Get pentru a face rost de ora la care se intampla rasaritul si apusul.
- ESP32:- aprinde si stinge luminile in functie de:
  - Ora actuala (culoarea luminii + daca se aprinde sau nu)
  - Activitatea in jurul unui senzor PIR
  - Comanda de pornire/oprire trimisa prin telefon

Hardware Design

Piese Necesare

- 1 x ESP32

- 1 x Banda Led RGB WS2812B
- 2 x Senzori PIR
- cabluri

Schema Electrica

Piesele

Software Design

Pentru dezvoltarea aplicației am folosit Arduino IDE cu plugin-urile instalate pentru ESP32.

Librarii externe folosite:

- Adafruit_MQTT → comunicarea cu servicile Adafruit ce primesc notificare de la IFTTT
- Adafruit_Neopixel → controlarea benzii de led
- HTTPClient → trimiterea de cereri GET pentru a face rost de ora apusului si rasaritului + pentru obtinerea de date pentru a configura ceasul intern al placutei
- ArduinoJson → parsarea obiectelor JSON primite din cererile GET
- Wifi → conectarea la reteaua wi-fi

Modul de functionare

La pornirea placutei, aceasta se conecteaza intai la wi-fi-ul selectat [1)]. Dupa aceea, va trimite cerere la serverul “pool.ntp.org” pentru a face rost de datele necesare pentru a configura ceasul intern al placutei. O data ce este gata, se va atasa la switch-ul digital creat in baza de date de la Adafruit. Va trimite o cerere get la api-ul Sunset and Sunrise Times si isi va salva ora la care este apusul si rasaritul in acea zi, apoi va calcula ora la care isi va incepe si termina automat activitatea.

In timpul activitatii sale, placuta va verifica daca e conectata la serverele IO de la Adafruit si va verifica daca se afla in intervalul de activitate automata sau nu. Apoi va verifica daca a primit vreo comanda de pornire sau de inchidere. Daca e “activa”, aceasta va verifica daca a fost depistata vreo miscare de catre un senzor PIR si va porni portiunea de banda de LED ce coresponde acelui senzor, culoarea LED-urilor depinzand de ora la care este aprinsa banda. Mai exact, daca e aprinsa cand e intuneric afara, vor fi colorate galben, iar daca e aprinsa cand e lumina afara, vor fi colorate albastru inchis. O data ce a trecut un minut, placuta va inchide portiunea de banda de led.

Functii din cod

Intreruperile pentru fiecare PIR sunt functii ce se executa direct de catre RAM-ul placutei si constau in punerea unui flag si afisarea mesajului corespunzator pe Serial Monitor. Mai exact, pentru senzorul PIR 1:

```
void IRAM_ATTR pir_1() {
  if (connected == 1) {
     motion_1 = 1;
     Serial.println("Object Detected_1 things conencted");
  } else {
    Serial.println("Object Detected_1 nothing conencted");
  }
}
```

Placuta va activa intreruperea cand va sesiza o coborare in tensiune la pinul la care este conectat senzorul.

Pentru a vedea culoarea led-urilor, ma folosesc de functia `mktime` pentru a converti o structura `tm` intr-una `time_t` ca apoi sa pot folosi functia `difftime`. Cu ajutorul acelei functii pot vedea daca timpul curent este pe timpul zilei (deci mai mare decat ora rasaritului si mai mica decat apusul) sau daca este pe timpul zilei (mai mare decat ora apusului SAU mai mica decat ora rasaritului). Acelasi mod de gandire este folosit si pentru a vedea daca placuta se afla in intervalul de activitate sau nu (+15 minute dupa apus fiind activa pentru maxim 4 ore). Mai jos e prezentat cum e folosit `difftime` pentru setarea culorii becurilor:

```
  double diffSecs = difftime(current_time, daylight_start);
  if (diffSecs > 0) {
    diffSecs = difftime(current_time, daylight_end);
    if (diffSecs < 0) {
      daylight = true;
      nightlight = false;
    } else {
      daylight = false;
      nightlight = true;
    }
  } else {
    daylight = false;
    nightlight = true;
  }
}
```

Verificarea aceasta se face o data la 10 minute, o data cu verificarea orei si zilei.

Pentru obtinerea orei apusului si a rasaritului, placuta va trimite o cerere de tip GET la api-ul `sunrise-sunset` cu latitudinea si longitudinea (aici sunt hardcodate drept cele ale Bucurestiului), data sa fie de azi si sa fie in timezone-ul Bucurestiului in acel moment [2)]. O data obtinut un raspuns, placuta va parsa datele primite drept un json si va salva intr-un `struct tm` datele necesare:

```
  DynamicJsonDocument doc(1024);
```

```
  DeserializationError error = deserializeJson(doc, payload);
  const char* time_data1 = doc["results"]["sunset"];
  int year, month, day, hour, minute, second;
  char timezone_sign;
  int timezone_hour, timezone_minute;

  // Parsează șirul de caractere
  sscanf(time_data1, "%d-%d-%dT%d:%d:%d%c%d:%d",
         &year, &month, &day, &hour, &minute, &second,
         &timezone_sign, &timezone_hour, &timezone_minute);

  sunset_time.tm_year = year - 1900;
  sunset_time.tm_mon = month - 1;
  sunset_time.tm_mday = day;
  sunset_time.tm_hour = hour;
  sunset_time.tm_min = minute;
  sunset_time.tm_sec = second;
  sunset_time.tm_isdst = -1;

  const char* time_data2 = doc["results"]["sunrise"];

  // Parsează șirul de caractere
  sscanf(time_data2, "%d-%d-%dT%d:%d:%d%c%d:%d",
         &year, &month, &day, &hour, &minute, &second,
         &timezone_sign, &timezone_hour, &timezone_minute);

  sunrise_time.tm_year = year - 1900;
  sunrise_time.tm_mon = month - 1;
  sunrise_time.tm_mday = day;
  sunrise_time.tm_hour = hour;
  sunrise_time.tm_min = minute;
  sunrise_time.tm_sec = second;
  sunrise_time.tm_isdst = -1;
```

Din acelasi json putem obtine atat ora de apus cat si ora de rasarit.

In afara orelor de functionare, placuta poate fi activata prin trimiterea unei comenzi prin Google Assistant, care, prin intermediul IFTTT, va schimba valoarea switch-ului de pe server-ul Adafruit in functie de comanda primita. Daca primeste comanda de activare, acesta va pune switch-ul pe 1. Daca primeste comanda de oprire, acesta va pune switch-ul pe 0. O data efectuata schimbarea, serverul IO va trimite o notificare la placuta cu noua valoare a variabilei, astfel stiind daca va porni luminile sau nu:

```
Adafruit_MQTT_Subscribe *subscription;
  if (subscription = mqtt.readSubscription(100)) {
    if (subscription == &Light1) {
      int Light1_State = atoi((char *)Light1.lastread);
      connected = Light1_State;
      if (connected == 1) {
        shutdown = false;
```

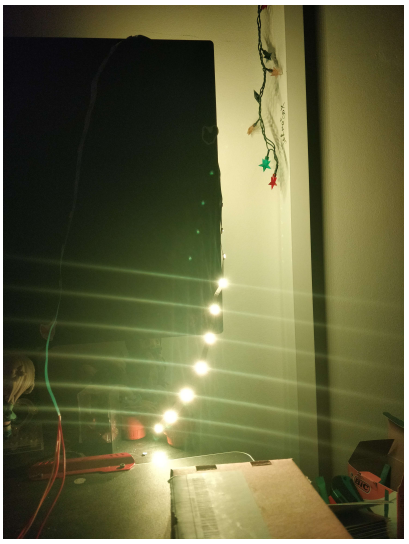
```
        Serial.println("Manual activation received");
      } else {
        shutdown = true;
        Serial.println("Forced shutdown received");
      }
    }
  }
```

Unde `connected` este variabila ce arata daca placuta e in starea “de activitate” sau nu. `shutdown` anunta placuta daca trebuie sa se inchida chiar daca este inca in intervalul de activitate.

Rezultate Obţinute

Video de prezentare





Imagini ce prezinta functionarea pe timp de noapte:

- Stanga: Cand ambii senzori detecteaza miscare
- Mijloc: Cand senzorul 2 detecteaza miscare
- Dreapta: Cand senzorul 1 detecteaza miscare

Concluzii

Proiectul poate fi expandat prin adaugarea unei benzi de led mai lunga si a mai multor senzori PIR (in limita pinilor). Totodata, se poate adauga o sursa de alimentare separata pentru senzorii PIR si banda de led.

Chiar si asa, proiectul reprezinta un prototip foarte bun pentru ideea la care voiam sa ajung si il voi instala la casa cat de curand.

Download

drugeadianaioana_333ca_proiect.zip

Jurnal

- 15:54 04.05.2024: Crearea paginii de documentatie si adaugarea informatiilor generale si lista hardware de inceput
- 16:45 23.05.2024: Update Lista Hardware + descriere generala
- 17:35 23.05.2024: Adaugare schema electrica
- 18:49 26.05.2024: Explicare firmware
- 19:46 26.05.2024: Adaugare poze si filmulet de prezentare
- 19:57 26.05.2024: Adaugare link-uri in bibliografie

Bibliografie/Resurse

Resurse Software

- Tutorial folosire ESP32 pentru conectarea cu aplicatii
- Tutorial Led Strip
- Tutorial conectare Adafruit cu Google asssistant
- Sunrise-Sunset API
- Tutorial configurare timp local
- Tutorial http pe ESP32
- Json Parser exemplu
- Tutorial PIR

Resurse Hardware

- ESP32 Datasheet
- Pini PIR

Export to PDF

[1] datele despre reteaua wi-fi sunt hardcodate in cod
[2] fie GMT+2, fie GMT+3 in functie de Daylight Savings