PRAKTIKUM INTERNET OF THINGS
Smart Home Prototype

Anggota Kelompok 2 :

Cahyo Arissabarno (1223800006)
Zacky Maulana Achmad (1223800008)

A. DASAR TEORI

Pada projek ini akan dibuat prototipe dari sebuah system smart home yang akan dibangun menggunakan mikrokontroler ESP 32. ESP32 adalah mikrokontroler yang dikenalkan oleh Espressif System dan merupakan penerus dari mikrokontroler ESP8266. Pada mikrokontroler ini sudah tersedia modul WiFi dan Bluetooth dalam chip. ESP32 memiliki fitur yang cukup lengkap karena mendukung input/output Analog dan Digital, PWM, SPI, I2C, dll.

Pada system smart home ini akan menerapkan beberapa fitur, diantaranya adalah :

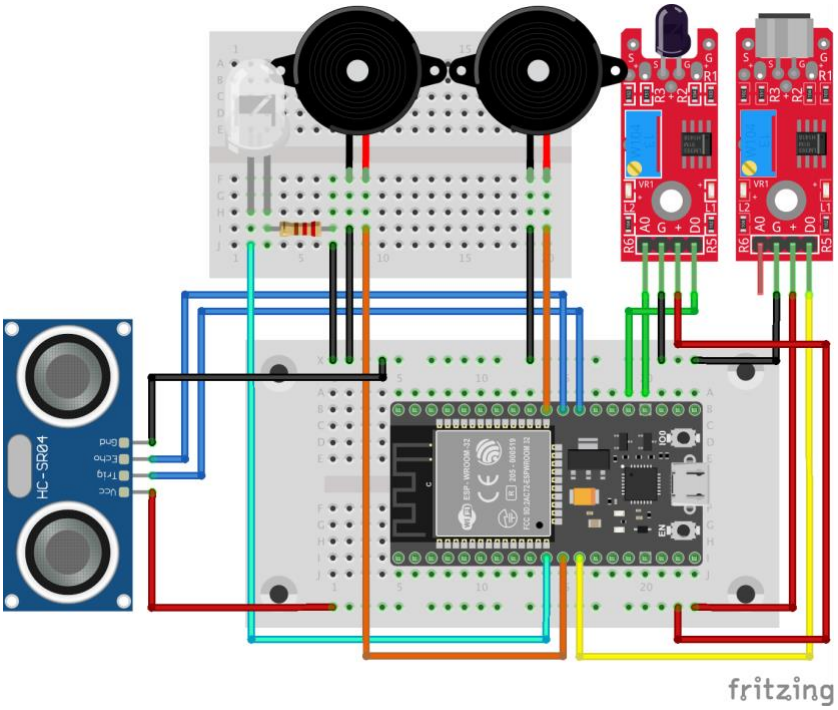
a. Sistem Smart Lamp
   Pada sistem ini, akan dibuat sebuah mekanisme untuk menyalakan lampu dengan menggunakan suara yang diucapkan. Untuk membuat prototipe dari system tersebut, digunakan modul sensor suara untuk menangkap gelombang suara dengan tingkat intensitas tertentu. Untuk aktuator nya, digunakan LED sebagai representasi dari lampu.

b. Sistem Alarm Kebakaran
   Pada sistem ini, akan dibuat sebuah mekanisme untuk membuat sebuah alarm yang akan berbunyi jika terdapat kebakaran. Untuk membuat prototipe dari system tersebut, digunakan modul sensor api untuk mengetahui adanya api di sekitar. Untuk aktuator nya, digunakan Buzzer sebagai representasi dari alarm.

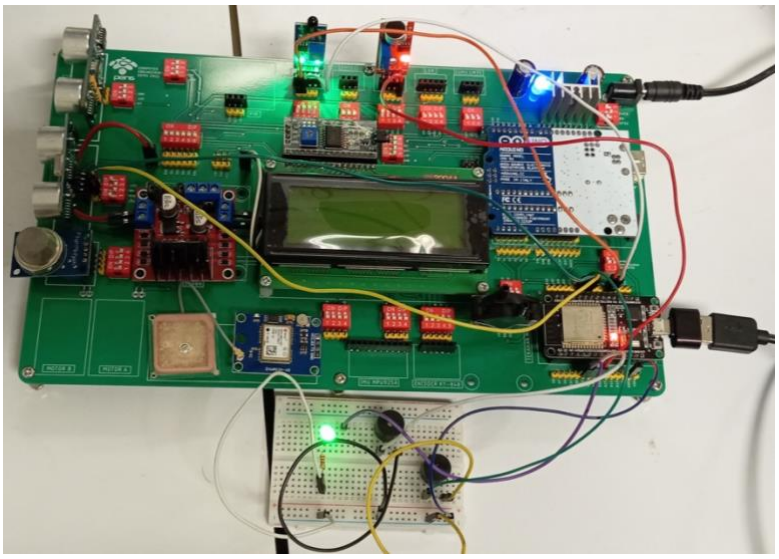
c. Sistem Alarm Parkir
   Pada sistem ini, akan dibuat sebuah mekanisme untuk membuat sebuah alat bantu parkir dimana akan mengeluarkan bunyi jika posisi mobil berada pada jarak tertentu mendekati tembok . Untuk membuat prototipe dari system tersebut, digunakan sensor Ultrasonik untuk mendeteksi jarak antara tembok dengan mobil di depan nya. Untuk aktuator, digunakan Buzzer untuk mengeluarkan tanda bunyi jika mobil sudah mendekati batas tertentu.

Semua data pembacaan dari sensor suara, sensor api, dan sensor ultrasonik akan dikirimkan menuju Blink Cloud untuk dapat dimonitor setiap data dari setiap sensor yang digunakan pada projek ini.

B. SCHEMATIC



Implementasi pada modul :



C. SOURCE CODE

```
#define BLYNK_TEMPLATE_ID           "....."
#define BLYNK_TEMPLATE_NAME         "..."
#define BLYNK_AUTH_TOKEN            "..."

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

char ssid[] = "..... ";
char pass[] = "....";
```

```
BlynkTimer timer;

// Variabel Ultrasonik Sensor
const int trigPin = 5;
const int echoPin = 18;
const int buzzerUltra = 19;

// Variabel Sound Sensor
const int ledPin = 32;
const int sensorPin = 35;

// Variabel Flame Sensor
const int buzzerFlame = 33;
const int Fire_analog = 34;

// Define sound speed in cm/uS
const float SOUND_SPEED = 0.034;
const float CM_TO_INCH = 0.393701;

long duration;
float distanceCm;
int val;
bool val_h = false;
int firesensorAnalog;

BLYNK_WRITE(V0)
{
  int value = param.asInt();
  Blynk.virtualWrite(V1, value);
}

BLYNK_CONNECTED()
{
  Blynk.setProperty(V3, "offImageUrl", "https://static-
image.nyc3.cdn.digitaloceanspaces.com/general/fte/congratulations.png");
  Blynk.setProperty(V3, "onImageUrl",  "https://static-
image.nyc3.cdn.digitaloceanspaces.com/general/fte/congratulations_pressed.png");
  Blynk.setProperty(V3, "url", "https://docs.blynk.io/en/getting-started/what-do-i-need-to-blynk/how-quickstart-device-
was-made");
}

void myTimerEvent()
{
  Blynk.virtualWrite(V2, millis() / 1000);
}

void setup() {
  Serial.begin(9600);
  // pinMode untuk Ultrasonik Sensor
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(buzzerUltra, OUTPUT);

  // pinMode untuk Sound Sensor
  pinMode(sensorPin, INPUT);
  pinMode(ledPin, OUTPUT);

  // pinMode untuk flame sensor
  pinMode(Fire_analog, INPUT);
  pinMode(buzzerFlame, OUTPUT);
```

```
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  timer.setInterval(1000L, myTimerEvent);

}
void loop() {
  Blynk.run();
  timer.run();

  data_Sound();
  data_Ultra();
  data_Flame();

  // PRINT ALL DATA
  Serial.print("ULTRASONIC: ");
  Serial.print(distanceCm);
  Blynk.virtualWrite(V4, distanceCm);
  Blynk.virtualWrite(V5, distanceCm);
  Serial.print(" | SOUND: ");
  Serial.print(val);
  Blynk.virtualWrite(V2, val);
  Blynk.virtualWrite(V3, val);
  Serial.print(" | FLAME: ");
  Serial.println(firesensorAnalog);
  Blynk.virtualWrite(V0, firesensorAnalog);
  Blynk.virtualWrite(V1, firesensorAnalog);
  delay(100);

}
void data_Ultra(){
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distanceCm = duration * SOUND_SPEED / 2;
  if (distanceCm <= 5) {
    digitalWrite(buzzerUltra, HIGH);
    delay(100);
    digitalWrite(buzzerUltra, LOW);
    delay(100);
  }
}
void data_Sound(){
  val = analogRead(sensorPin);
  if (val >= 30) {
    val_h = !val_h; // Toggle the value
    digitalWrite(ledPin, val_h);
    delay(100);
  }
}
void data_Flame(){
  firesensorAnalog = analogRead(Fire_analog);
  if (firesensorAnalog < 1000) {
    digitalWrite(buzzerFlame, HIGH);
    delay(100);
    digitalWrite(buzzerFlame, LOW);
  }
}
```

D. ANALISA

Setelah rangkaian prototipe dapat berjalan, dapat diamati bagaimana cara kerja dan respon dari setiap implementasi sistem. Pada sistem smart lamp, input analog menuju ESP 32 diperlukan untuk mendapatkan nilai intensitas suara di sekitar. Pada algoritma yang digunakan, terdapat perkondisian dimana jika nilai analog dari sensor suara lebih besar dari 30, maka pin yang terhubung dengan LED akan diberikan nilai HIGH jika kondisi LED mati dan diberikan nilai LOW jika kondisi LED menyala. Sistem ini diuji dengan menggunakan suara manusia dengan kata “Hallo” dan dengan tepukan tangan pada jarak yang dekat dan jauh. Hasil yang didapatkan adalah suara manusia dengan jarak yang dekat dengan sensor berhasil membuat LED menyala dengan nilai analog sekitar 40-60. Sedangkan suara manusai dengan jarak yang jauh dari sensor menghasilkan nilai kurang dari 30 sehingga tidak berhasil membuat LED menyala. Hal tersebut juga terjadi pada tepukan tangan. Dengan jarak yang dekat dan jauh tepukan tangan menghasilkan nilai kurang dari 30 sehingga tidak berhasil membuat LED menyala.

Pada implementasi sistem alarm kebakaran , input analog dari sensor api juga diperlukan untuk mendapatkan nilai kondisi panas api di sekitar sensor. Pada algoritma yang digunakan, terdapat perkondisian dimana jika nilai analog dari sensor api lebih kecil dari 1000, maka pin yang terhubung dengan Buzzer akan diberikan nilai HIGH sehingga Buzzer dapat menghasilkan bunyi sebagai tanda adanya kebakaran. Sistem ini diuji dengan menggunakan korek api yang dinyalakan di dekat sensor. Hasil yang didapatkan adalah sensor ini cukup sensitif untuk mendeteksi api yang dihasilkan oleh korek api dengan didapatkan nilai analog 0 saat api dinyalakan. Buzzer dapat langsung menyala setelah terdapat api yang dinyalakan di sekitar sensor.

Pada implementasi sistem alarm parkir, input yang digunakan adalah pulsa dari gelombang yang didapatkan oleh sensor ultrasonic dan perlu dikonversi menuju nilai ukur centimeter (cm) untuk mengetahui nilai jarak dengan pasti. Pada algoritma yang digunakan, terdapat perkondisian dimana jika nilai jarak dalam cm yang didapatkan dari sensor ultrasonik lebih kecil dari 5 cm, maka Buzzer akan menyala dengan periodik HIGH dan LOW dengan jeda waktu 100 ms. Sistem ini diuji dengan menempatkan objek didepan sensor ultrasonic. Hasil yang didapatkan adalah sensor ultrasonic telah dapat mendeteksi objek yang ada di depannya dengan cukup akurat dan Buzzer dapat menyala sesuai dengan perkondisian jarak yang telah ditetapkan.

Selain implementasi diatas, setiap data pembacaan sensor telah berhasil terkirim pada Blink Cloud dan dapat divisualisasikan dalam bentuk grafik pada dashboard Blink Cloud. Namun, terdapat delay sekitar 1-2 detik dari pembacaan sensor hingga tertampilkan pada dashboard Blink Cloud.

E. KESIMPULAN

Dari hasil yang didapatkan, dapat disimpulkan bahwa prototipe smart home telah dapat terimplementasi dengan cukup baik dari hasil pembacaan sensor yang cukup sesuai dan aktuator telah dapat merespon sesuai kondisi pada algoritma. Monitoring data melalui

Blink Cloud juga telah dapat berjalan. Namun, sensifitas dari sensor suara masih sedikit kurang untuk dapat mendeteksi suara dengan jarak yang lebih besar.