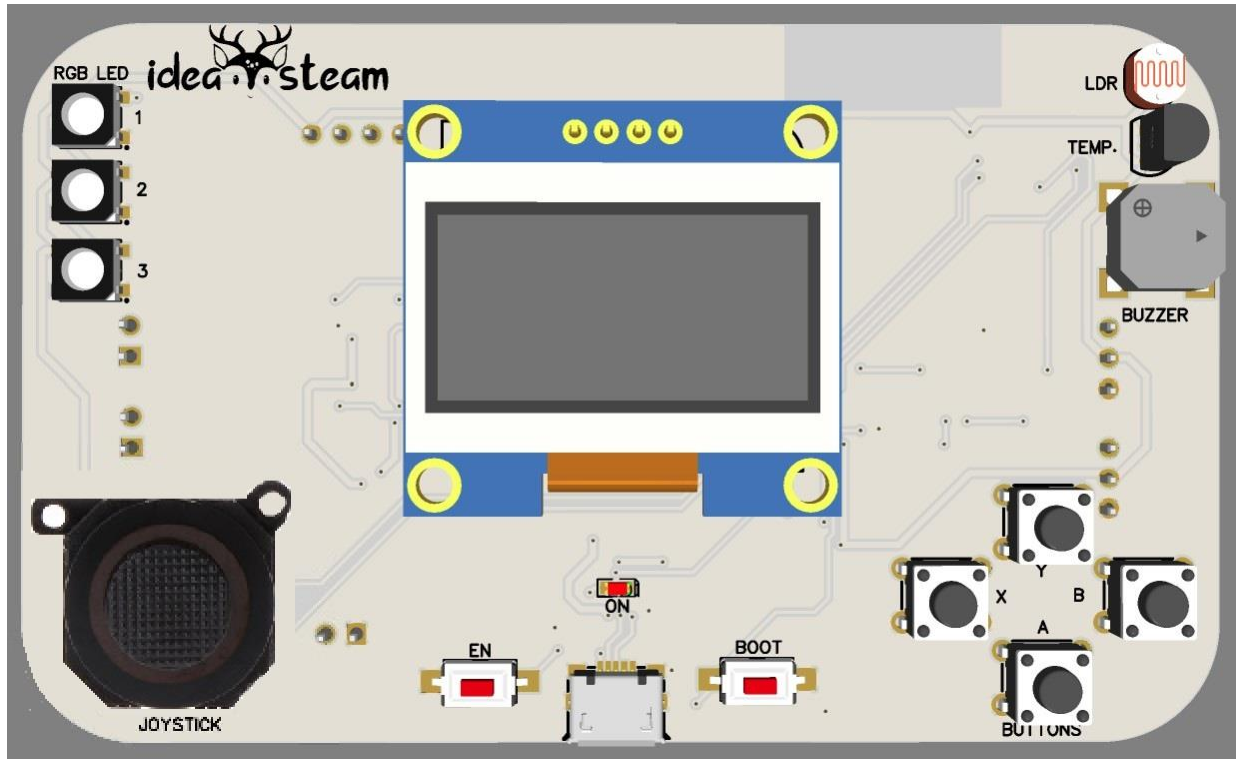


Panduan Penggunaan Papan Pengembangan ideaSTEAM XB1



TIM ideaSTEAM

MEDAN

2022

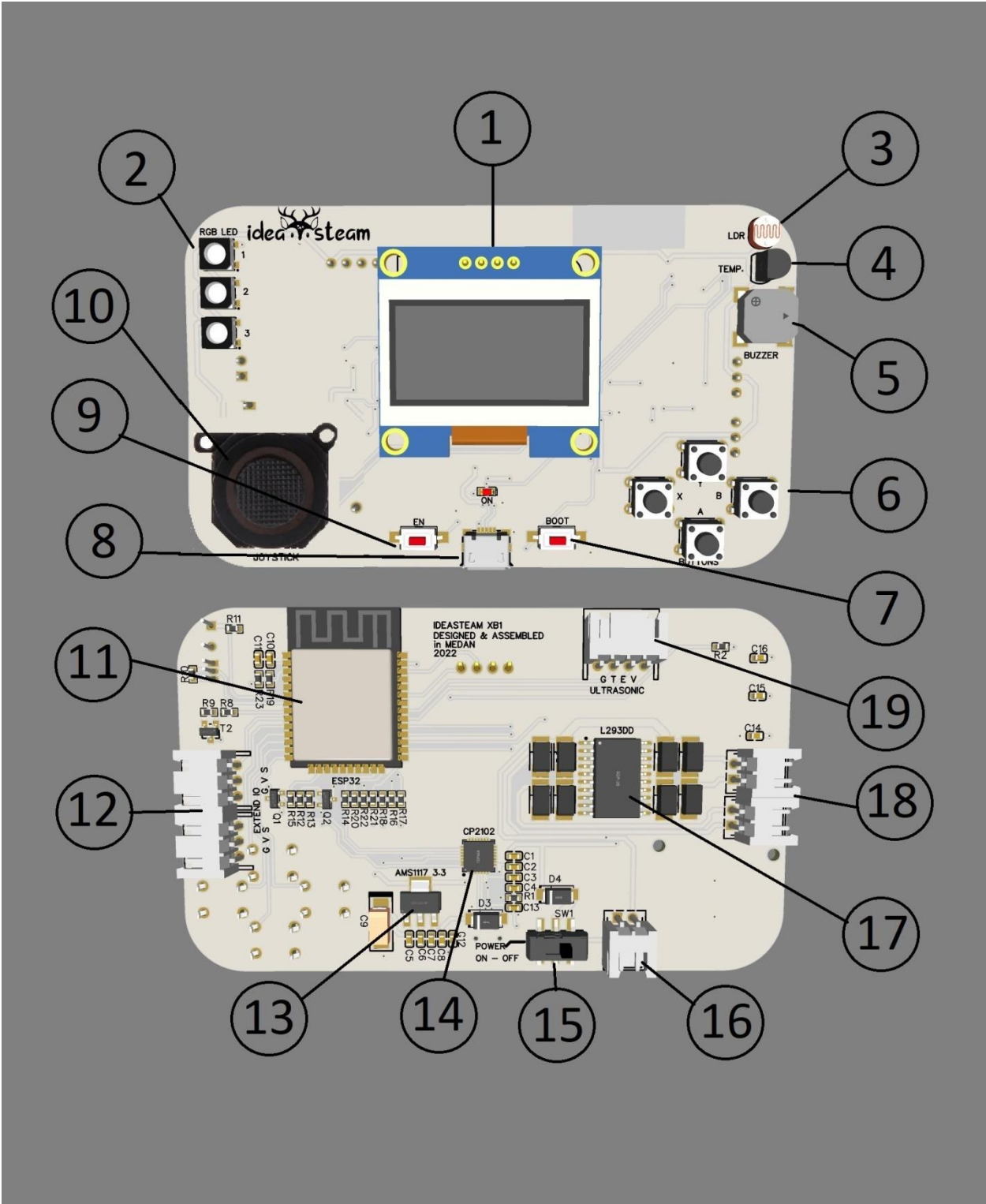
IdeaSTEAM XB1

IdeaSTEAM eXperiment Board Versi 1 atau disingkat IdeaSTEAM XB1, merupakan papan eksperimen rangkaian elektronika yang dapat digunakan untuk berbagai percobaan komputasi elektronik dengan lingkungan sekitar.

SPESIFIKASI:

- Mikrokontroler ESP32
- Konektifitas: WiFi dan Bluetooth
- CP2102 USB-to-UART
- Layar OLED 0.96"
- 3 buah LED RGB WS2812B
- Sensor Light Dependant Resistor
- Sensor Suhu DS18B20
- Buzzer
- Analog Joystick
- 4 buah Push Button
- Driver Motor L293DD
- USB Micro Type-B
- Konektor I/O 4-Pin
- 2 Buah Konektor I/O 3-Pin
- 2 Buah Konektor Motor DC 2-Pin
- Konektor Battery / DC-Input 2-Pin, Max 5.5V
- Regulator Tegangan AMS1117 3.3V

Tata Letak Komponen



Penjelasan Komponen :

1. Layar OLED 0.96”

Layar OLED atau *Organic Light Emitting Diodes* merupakan sebuah layar yang terbentuk dari LED bersifat organik. Pada board ini, Layar OLED yang terpasang adalah OLED SSD1306 0.96” yang memiliki resolusi 128x64 Pixel. Layar ini dihubungkan dengan mikrokontroler melalui antarmuka **I2C** pada pin **GPIO21** dan **GPIO22**.

2. LED RGB WS2812B

LED RGB WS2812 merupakan sumber cahaya yang dapat dikontrol warnanya dengan memprogramnya. Pada Board ini, LED WS2812B yang terpasang sebanyak 3 buah. LED ini dihubungkan dengan mikrokontroler melalui antarmuka **digital** pada pin **GPIO 23**.

3. Sensor Light Dependant Resistor (LDR)

Sensor LDR merupakan sensor yang dapat mendeteksi perubahan cahaya. Sensor ini dihubungkan dengan mikrokontroler melalui antarmuka **analog** pada pin **GPIO34**.

4. Sensor Suhu DS18B20

Sensor Suhu DS18B20 merupakan sensor yang dapat mendeteksi perubahan suhu. Sensor ini dihubungkan dengan mikrokontroler melalui antarmuka **One Wire** pada pin **GPIO35**.

5. Buzzer

Buzzer merupakan sebuah perangkat yang dapat mengeluarkan suara, mirip *speaker*. Sensor ini dihubungkan dengan mikrokontroler melalui antarmuka Analog pada pin **GPIO32**.

6. Push Button

Push Button (Tombol) merupakan perangkat masukan sinyal digital, dapat juga disebut saklar. Pada board ini, push button yang dipasang adalah sejumlah 4 buah, dan terhubung pada mikrokontroler melalui antarmuka digital pada pin GPIO26, GPIO27, GPIO14, dan GPIO12.

7. BOOT – Push Button

Tombol BOOT digunakan untuk memasuki mode meng-upload program. Tombol ini ditekan saat kita memasukan program dari komputer ke board ideaSTEAM XB1.

8. USB Micro Type-B

USB pada board ini digunakan sebagai koneksi dengan komputer, dan dapat pula sebagai sumber daya untuk board ini.

9. EN/Reset Botton

Tombol EN (Enable) merupakan tombol yang dapat digunakan untuk me-reset kerja dari mikrokontroler. Jika tombol ini ditekan, program akan berpindah ke awal program.

10. Analog Joystick

Analog Joystick merupakan tombol yang dapat digerakkan ke segala arah(atas-bawah-kiri-kanan). Joystick dapat menjadi masukan sinyal analog. Pada board ini Joystick dihubungkan ke mikrokontroler melalui antarmuka analog pada pin **GPIO36** dan **GPIO39**.

11. ESP32 WROOM32-E

ESP32 merupakan mikrokontroler yang bertugas sebagai otak utama pada board ini, yang menangani berbagai masukan, keluaran dan komputasi yang diprogram pada board ini.

12. Konektor I/O 3-Pin

Konektor I/O 3-pin pada board ini dapat digunakan sebagai masukan maupun keluaran. Konektor ini terdiri dari 3-pin yang ditandai dengan tulisan SVG(Signal-VCC-Ground). Ada dua konektor I/O 3-pin pada Board ini. Pin yang dapat digunakan sebagai masukan maupun keluaran adalah pin (S)ignal yang dihubungkan ke mikrokontroler pada pin GPIO25 dan GPIO33.

13. Regulator Tegangan

Regulator tegangan pada board ini berfungsi untuk menurunkan tegangan dari 5 Volt (dari baterai maupun usb) ke 3.3 Volt, hal ini karena sebagian besar komponen pada board ini bekerja dengan tegangan 3.3 Volt.

14. USB-to-UART CP2102

USB-to-UART CP2102 merupakan sebuah chip yang bertugas menangani komunikasi antara komputer dan board IdeaSTEAM XB1.

15. Saklar ON-OFF

Saklar ON-OFF pada board ini berfungsi untuk menghubungkan – memutuskan aliran daya dari sumber daya baterai/input dc ke board ini. Jika menggunakan sumber daya dari USB, saklar ini tidak difungsikan.

16. Konektor Battery

Konektor *Battery* digunakan untuk menghubungkan sumber daya DC dari luar board, dapat berupa baterai maupun catu daya DC. Pada konektor ini memiliki 2 penanda, yaitu positif (+) dan Negatif (-). Tanda (+) dapat dihubungkan dengan kutub baterai positif(+) atau VCC pada catu daya DC. Sementara tanda (-) dapat dihubungkan dengan kutub baterai (-) atau GND pada catu daya DC. Sebaiknya tegangan yang diberikan pada board ini berkisar antara 4.5 volt hingga 5.5

Volt dan perlu diingat bahwa masukan DC untuk board ini **tidak boleh lebih dari 5.5 Volt.**

17. Driver Motor DC

Driver Motor DC merupakan komponen yang mengatur kerja motor DC, baik kecepatan maupun arah putar dari Motor DC. Driver motor ini bekerja dengan masukan sinyal digital maupun analog (PWM). Pada board ini, driver motor yang dipasang adalah L293DD dengan tambahan rangkaian H-Bridge yang dapat mengontrol motor DC hingga 12Volt, 7.6Watt.

18. Konektor Motor DC

Konektor Motor DC, pada board ini berfungsi untuk menghubungkan motor dc dengan board ini.

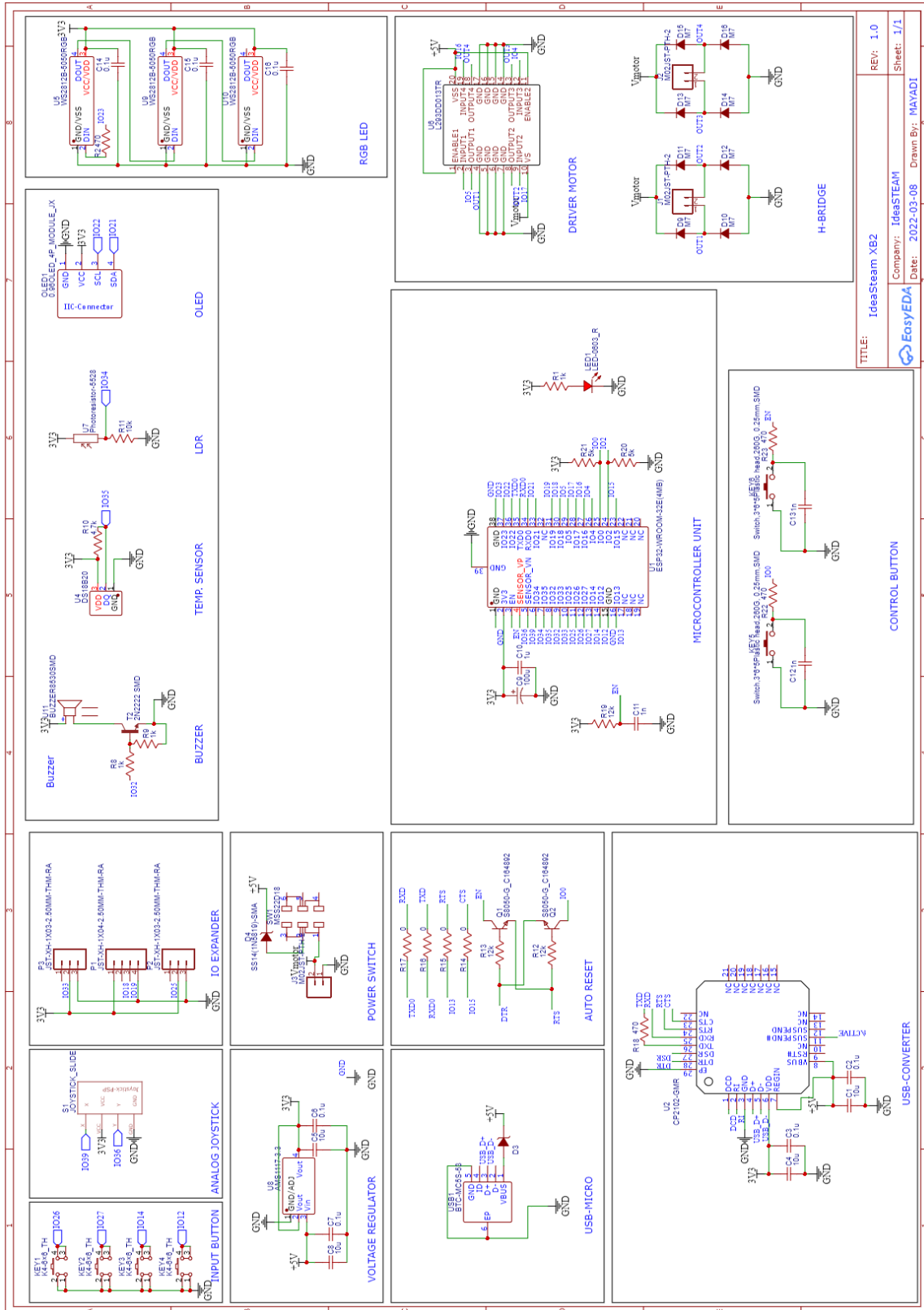
19. Konektor I/O 4-Pin

Konektor I/O 4-pin pada board ini dapat digunakan sebagai masukan maupun keluaran, namun konektor ini lebih dikhususkan untuk menghubungkan board dengan sensor ultrasonik. Konektor ini terdiri dari 4-pin yang ditandai dengan tulisan GTEV (Ground-Triger-Echo-VCC). Konektor ini dihubungkan ke mikrokontroler pada board ini pada pin GPIO19(Trigger) dan GPIO18 (Echo).

PIN-MAP INPUT-OUTPUT

NAMA KOMPONEN	ANTARMUKA	KETERANGAN	PIN GPIO
OLED	I2C	SDA	21
		SCL	22
LED RGB	DIGITAL: OUTPUT		23
LDR	ANALOG: INPUT		34
DS18B20	DIGITAL: INPUT/OUTPUT		35
BUZZER	ANALOG: OUTPUT		32
BUTTON	DIGITAL: INPUT	Y	26
		A	14
		B	27
		X	12
JOYSTICK	ANALOG: INPUT	X-DATAR	39
		Y-TEGAK	36
KONEKTOR 3-PIN	DIGITAL/ANALOG: INPUT-OUTPUT	ATAS	33
		BAWAH	25
MOTOR DC	DIGITAL/ANALOG: OUTPUT	MOTOR 1-A	5
		MOTOR 1-B	17
		MOTOR 2-A	4
		MOTOR 2-B	16
KONEKTOR 4-PIN	DIGITAL: INPUT/OUTPUT	INPUT((E)CHO)	19
		OUTPUT((T)RIGGER)	18

Rangkaian Skematik



CONTOH PROYEK:

1. Tampilan Layar

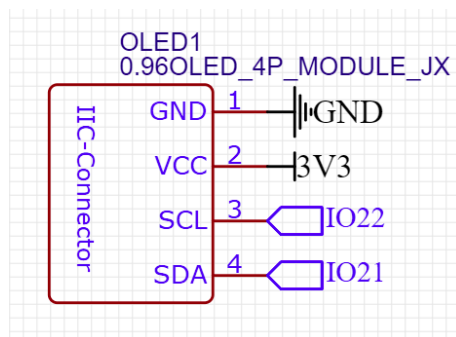
Berikut adalah contoh proyek untuk menampilkan tulisan berupa huruf dan angka pada layar OLED.

Komponen:

-ESP32

-OLED SSD1306 0.96"

Rangkaian:



GND OLED ke GND ESP32

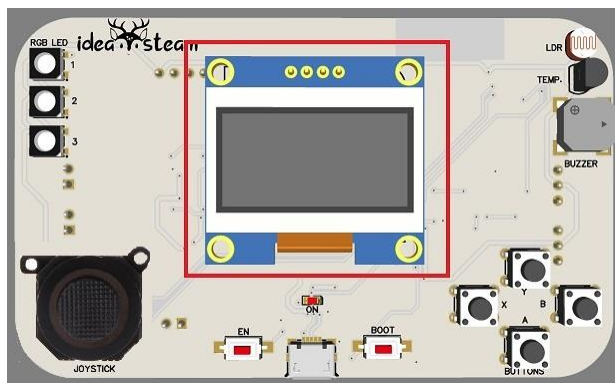
VCC OLED ke 3.3V ESP32

SDA OLED ke SDA/IO21 ESP32

SCL OLED ke SCL/IO22 ESP32

K

Komponen yang digunakan pada Board:



Program:

Program untuk menampilkan tulisan “Hello World!” pada layar.

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

Adafruit_SSD1306 display(128, 64, &Wire, -1);

void setup() {
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);

  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 30);

  display.print("Hello World!");

  display.display();
  delay(2000);
}

void loop() {
}
```

Program untuk menampilkan variasi tulisan di Layar OLED.

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // lebar layar, dalam piksel
#define SCREEN_HEIGHT 64 // tinggi layar, dalam piksel

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT,
&Wire, -1);

void setup() {
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);

  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 30);
  display.print("Hello World!");
  display.display();
  delay(2000);
}

void loop() {
  display.clearDisplay();
  display.setCursor(0, 10);
  display.print("Hallo !!!");
  display.display();
  delay(1000);

  display.setCursor(0, 20);
  display.print("Nama saya adalah IdeaSTEAM");
  display.display();
  delay(1000);

  display.setCursor(0, 40);
  display.print("Saya senang belajar koding dan STEAM");
  display.display();
  delay(5000);
}
```

2. Digital Output : LED

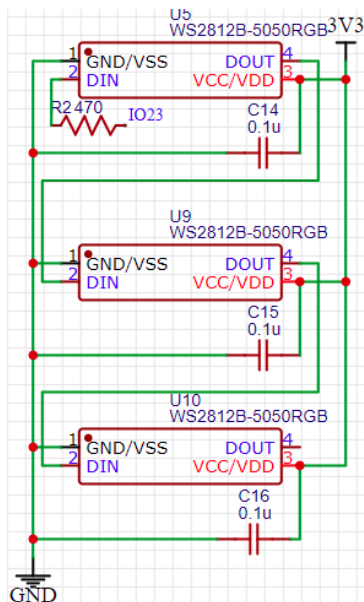
Berikut adalah contoh proyek untuk mengontrol LED RGB pada board ideaSTEAM XB1.

Komponen:

-ESP32

-3 buah LED RGB WS2812B

Rangkaian:



VCC LED ke 3.3V ESP32

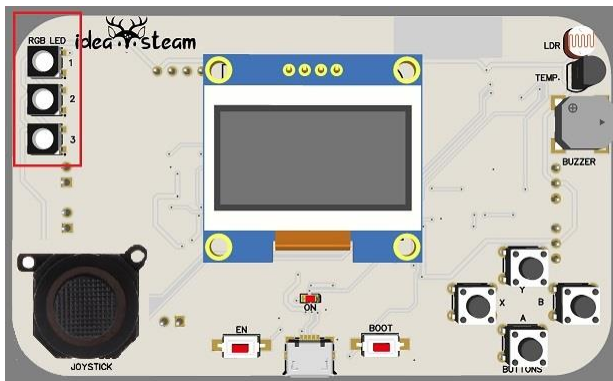
GND LED ke GND ESP32

DIN LED 1 ke IO23 ESP32

DOUT LED 1 ke DIN LED 2

DOUT LED 2 ke DIN LED 3

Komponen yang digunakan pada board:



Program:

Program untuk mengontrol LED 1, Merubah warna pada LED 1 berurutan mulai dari Merah(Red), Hijau(Green) dan Biru(Blue).

```
#include <Adafruit_NeoPixel.h>

#define PIN 23
#define jumlah_LED 3

Adafruit_NeoPixel pixels(jumlah_LED, PIN, NEO_GRB +
NEO_KHZ800);

void setup() {
  pixels.begin();
  pixels.clear();
}

void loop() {

  pixels.setPixelColor(0, pixels.Color(100, 0, 0)); //Red
  pixels.show();
  delay(1000);

  pixels.setPixelColor(0, pixels.Color(0, 100, 0)); //Green
  pixels.show();
  delay(1000);

  pixels.setPixelColor(0, pixels.Color(0, 0, 100)); //Blue
  pixels.show();
  delay(1000);

}
```

Program untuk mengontrol 3 buah LED. Menampilkan susunan dari LED 1 ke LED 3 dengan susunan Merah, Hijau, dan Biru. Kemudian melakukan simulasi perubahan Lampu Lalu Lintas, Merah, Kuning dan Hijau.

```
#include <Adafruit_NeoPixel.h>

#define PIN 23
#define jumlah_LED 3
Adafruit_NeoPixel pixels(jumlah_LED, PIN, NEO_GRB +
NEO_KHZ800);

void setup() {
  pixels.begin();
  pixels.clear();

  pixels.setPixelColor(0, pixels.Color(100, 0, 0));
  pixels.setPixelColor(1, pixels.Color(0, 100, 0));
  pixels.setPixelColor(2, pixels.Color(0, 0, 100));

  pixels.show();
  delay(2000);
}

void loop() {
  pixels.setPixelColor(0, pixels.Color(100, 0, 0));
  pixels.setPixelColor(1, pixels.Color(0, 0, 0));
  pixels.setPixelColor(2, pixels.Color(0, 0, 0));
  pixels.show();
  delay(1000);

  pixels.setPixelColor(0, pixels.Color(0, 0, 0));
  pixels.setPixelColor(1, pixels.Color(100, 10, 0));
  pixels.setPixelColor(2, pixels.Color(0, 0, 0));
  pixels.show();
  delay(1000);

  pixels.setPixelColor(0, pixels.Color(0, 0, 0));
  pixels.setPixelColor(1, pixels.Color(0, 0, 0));
  pixels.setPixelColor(2, pixels.Color(0, 100, 0));
  pixels.show();
  delay(1000);
}
```

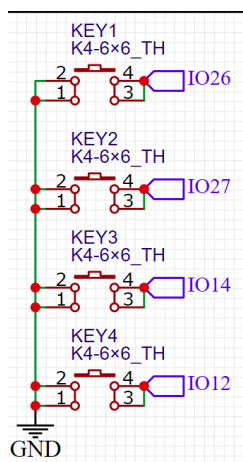

3. Digital Input : Button

Berikut adalah contoh proyek untuk membaca masukan Button dan menampilkan kondisinya ke Layar OLED.

Komponen:

- ESP32
- 4 buah Push Button
- Layar OLED

Rangkaian:



Pin 3 atau 4 Button A ke IO14 ESP32

Pin 3 atau 4 Button B ke IO27 ESP32

Pin 3 atau 4 Button X ke IO12 ESP32

Pin 3 atau 4 Button Y ke IO26 ESP32

Pin 1 atau 2 Semua Button ke GND ESP32

Komponen yang digunakan pada board:



Program:

Pada program ini, akan dilakukan pembacaan satu buah button yaitu button A.

```
#define button_A 14

void setup() {
  pinMode(button_A, INPUT_PULLUP);

  Serial.begin(115200);
}

void loop() {
  if (digitalRead(button_A) == LOW) {
    Serial.println("button A ditekan!")
  }
  delay(1000);
}
```

Pada program berikut, akan dilakukan pembacaan kondisi terhadap 4 buah Button, yaitu Button A, B, X, dan Y.

```
#define button_A 14
#define button_B 27
#define button_X 12
#define button_Y 26

void setup() {
  pinMode(button_A, INPUT_PULLUP);
  pinMode(button_B, INPUT_PULLUP);
  pinMode(button_X, INPUT_PULLUP);
  pinMode(button_Y, INPUT_PULLUP);

  Serial.begin(115200);
}

void loop() {
  if(digitalRead(button_A) == LOW){
    Serial.println("button A ditekan!");
  }
  if(digitalRead(button_B) == LOW){
    Serial.println("button B ditekan!");
  }
  if(digitalRead(button_X) == LOW){
    Serial.println("button X ditekan!");
  }
  if(digitalRead(button_Y) == LOW){
    Serial.println("button Y ditekan!");
  }
  delay(1000);
}
```

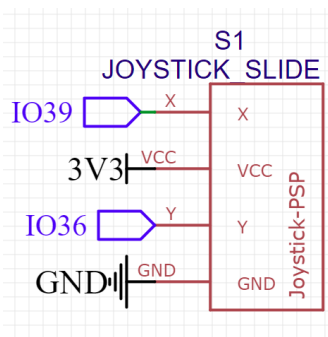
4. Analog Input : JoyStick

Berikut adalah contoh proyek untuk menggunakan joystick sebagai masukan analog.

Komponen:

- ESP32
- Joystick

Rangkaian:



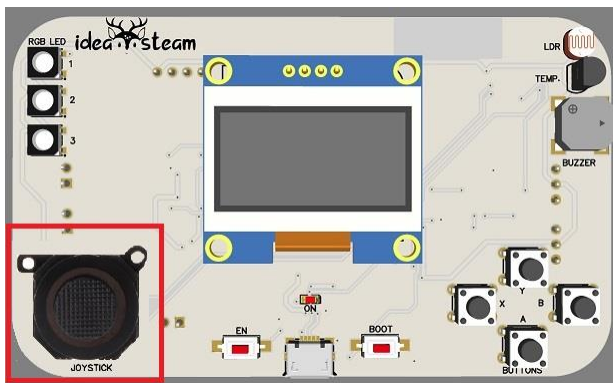
VCC Joystick ke 3.3V ESP32

GND Joystick ke GND ESP32

X pada Joystick ke IO39 ESP32

Y pada Joystick ke IO36 ESP32

Komponen yang digunakan pada board:



Program:

```
#define joystick_x 36
#define joystick_y 39

int x = 0;
int y = 0;

void setup() {
  Serial.begin(115200);
}

void loop() {

  x = analogRead(joystick_x);
  y = analogRead(joystick_y);

  x = abs(map(x, 3600, 610, 0, 100));
  y = abs(map(y, 510, 3400, 0, 100));

  Serial.print("Nilai X : ");
  Serial.println(x);

  Serial.print("Nilai Y : ");
  Serial.println(y);
  delay(1000);
}
```

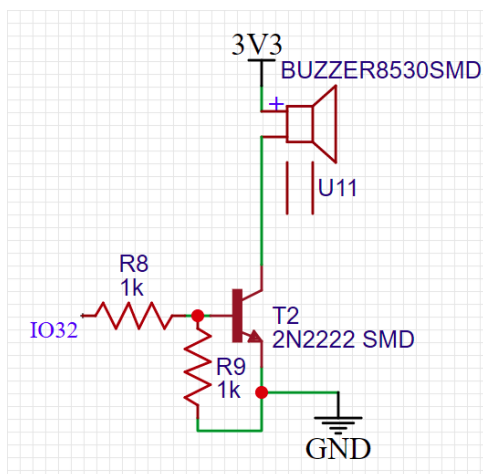
5. Buzzer

Berikut adalah contoh proyek untuk mengontrol keluaran suara menggunakan buzzer.

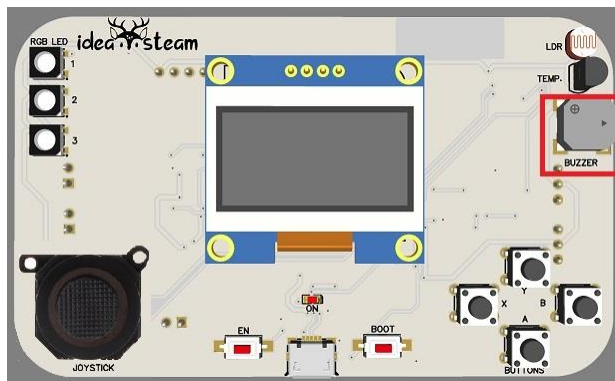
Komponen:

- ESP32
- Buzzer
- Transistor NPN
- 2 buah Resistor 1K

Rangkaian:



Komponen yang digunakan pada board:



Program:

```
#define buzzer 32

void setup() {
  pinMode(buzzer, OUTPUT);
}

void loop() {
  for (int i = 0; i < 100; i++) {
    digitalWrite(buzzer, HIGH);
    delayMicroseconds(250);
    digitalWrite(buzzer, LOW);
    delayMicroseconds(250);
  }
  delay(990);
}
```

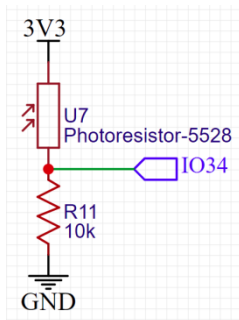
6. Sensor Cahaya

Berikut adalah contoh proyek untuk membaca sensor cahaya Light Dependant Resistor (LDR). Hasil dari pembacaan nilai sensor, dapat dilihat melalui Serial Monitor, Layar OLED, maupun diolah untuk mengatur cahaya pada LED.

Komponen:

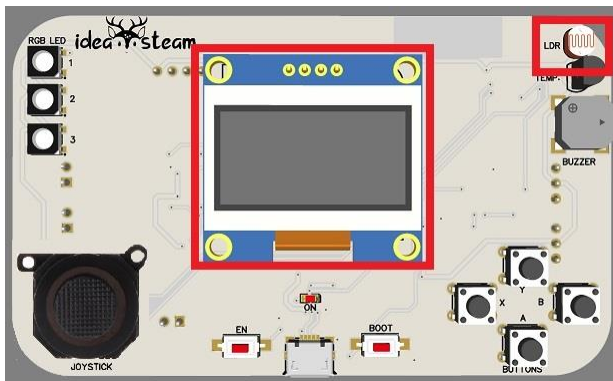
- ESP32
- OLED 0.96"
- Sensor LDR
- Resistor 10k

Rangkaian:



Rangkaian dibuat seperti pada gambar disamping, dengan menghubungkan IO34 ke cabang antara sensor LDR dan resistor 10k.

Komponen yang digunakan pada board:



Program:

Berikut adalah program untuk membaca nilai perubahan cahaya menggunakan sensor ldr, dan hasil pembacaannya dapat dilihat pada serial monitor.

```
#define ldr 34

int cahaya = 0;

void setup() {
  Serial.begin(115200);
}

void loop() {
  cahaya = analogRead(ldr);
  Serial.println(cahaya);
  delay(1000);
}
```

Kemudian, berikut adalah program untuk menampilkan hasil pembacaan sensor cahaya ke layar OLED.

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

Adafruit_SSD1306 display(128, 64, &Wire, -1);

#define sensor_ldr 34

int cahaya = 0;

void setup() {
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
}

void loop() {
  cahaya = analogRead(sensor_ldr);

  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 32);
  display.print("Cahaya : ");
  display.print(cahaya);
  display.display();
  delay(1000);
}
```

Berikut adalah pengembangan program, dimana hasil pembacaan sensor cahaya dapat dipakai untuk mengontrol hidup atau matinya lampu (LED).

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <Adafruit_NeoPixel.h>

Adafruit_SSD1306 display(128, 64, &Wire, -1);

#define PIN 23
#define jumlah_LED 3

Adafruit_NeoPixel pixels(jumlah_LED, PIN, NEO_GRB +
NEO_KHZ800);

#define sensor_ldr 34

int cahaya = 0;

void setup() {
  pixels.begin();
  pixels.clear();

  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
}

void loop() {
  cahaya = analogRead(sensor_ldr);
  display.clearDisplay();
  display.setCursor(0, 0);
  display.print("Cahaya : ");
  display.print(cahaya);
  display.setCursor(0, 32);
  if (cahaya < 1000) {
    pixels.setPixelColor(0, pixels.Color(100, 100, 100));
    display.print("Gelap : hidupkan LED");
  } else {
    pixels.setPixelColor(0, pixels.Color(0, 0, 0));
    display.print("Terang : matikan LED");
  }
  pixels.show();
  display.display();
  delay(1000);
}
```

7. Sensor Suhu

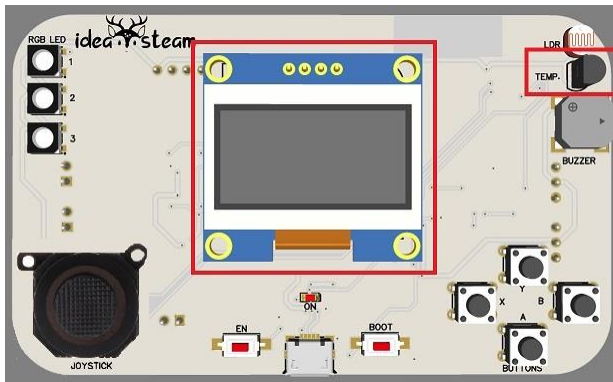
Berikut adalah contoh proyek menggunakan sensor suhu dan layar.

Komponen:

- ESP32
- Sensor Suhu
- Layar OLED

Rangkaian:

Komponen yang digunakan pada board:



Program:

```
#include <OneWire.h>
#include <DallasTemperature.h>

const int oneWireBus = 35;      //Pin ke sensor

OneWire oneWire(oneWireBus);
DallasTemperature sensors(&oneWire);

void setup() {
    Serial.begin(115200);
    sensors.begin();
}

void loop() {
    sensors.requestTemperatures();
    float temperatureC = sensors.getTempCByIndex(0);
    float temperatureF = sensors.getTempFByIndex(0);
    Serial.print(temperatureC);
    Serial.println("°C");
    Serial.print(temperatureF);
    Serial.println("°F");
    delay(5000);
}
```

8. Sensor Jarak

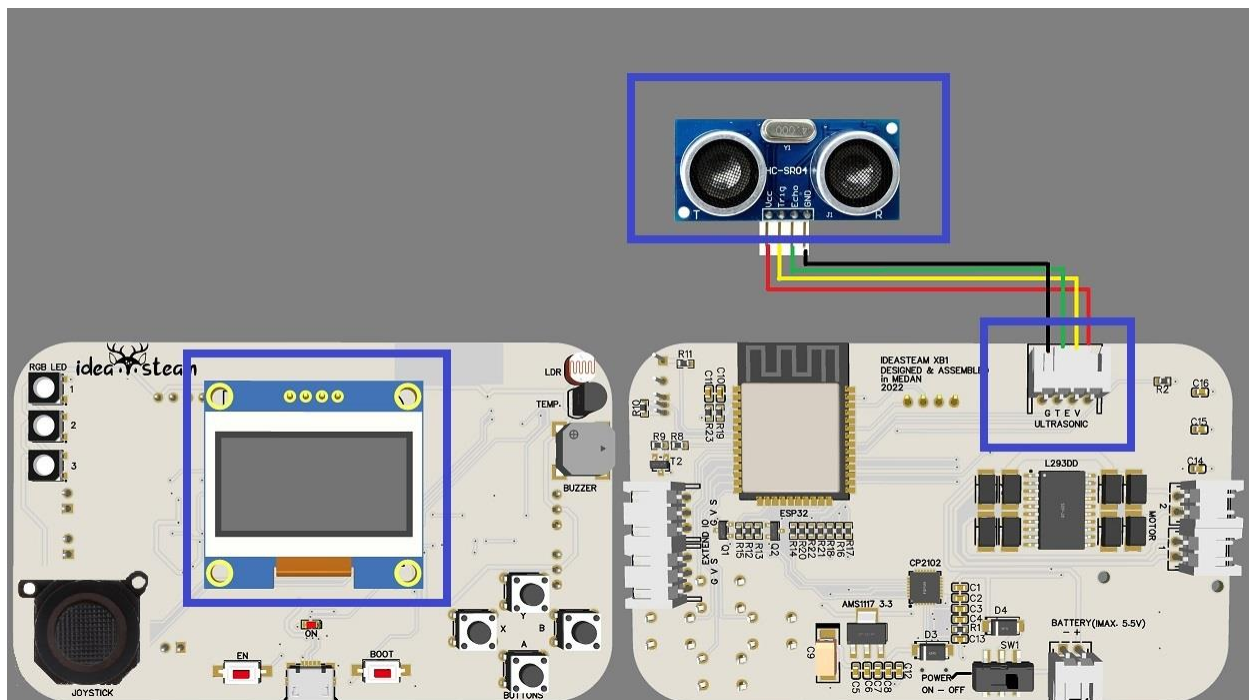
Berikut adalah contoh proyek menggunakan sensor jarak HC-SR04 dan menampilkan hasilnya ke Serial Monitor.

Komponen:

- ESP32
- Sensor Jarak Ultrasonik
- Layar OLED

Rangkaian:

Komponen yang digunakan pada board:



Program:

Berikut adalah program untuk menampilkan hasil pembacaan sensor jarak ke serial monitor.

```
const int trigPin = 18;
const int echoPin = 19;

#define SOUND_SPEED 0.034

long duration;
float distanceCm;

void setup() {
  Serial.begin(115200);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);

  distanceCm = duration * SOUND_SPEED/2;

  Serial.print("Distance (cm): ");
  Serial.println(distanceCm);

  delay(1000);
}
```

Berikut adalah program untk menampilkan hasil pembacaan sensor jarak ke layar OLED.

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

Adafruit_SSD1306 display(128, 64, &Wire, -1);
const int trigPin = 18;
const int echoPin = 19;

#define SOUND_SPEED 0.034

long duration;
float jarak;

void setup() {
  Serial.begin(115200);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
}

void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  jarak = duration * SOUND_SPEED / 2;

  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 30);
  display.print("Jarak : ");
  display.print(jarak);
  display.print(" CM");
  display.display();
  delay(2000);
}
```