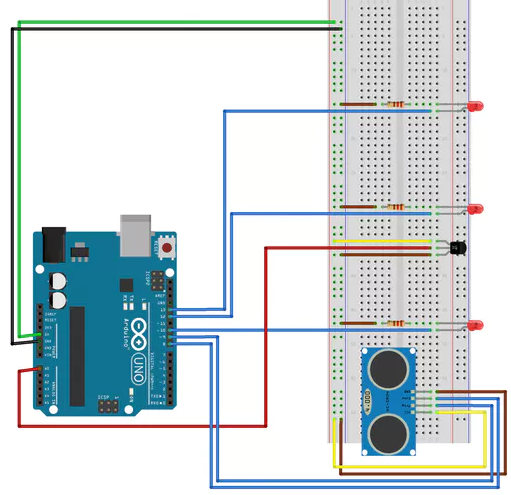
|  |  |  |
| --- | --- | --- |
| **KELOMPOK** | 1 | **ACC** |
| **Tanggal Praktikum** | 21 Juli 2018 |  |
|  | Khairunnisa Sekar R. (D400150005) |  |
|  | Dian Arieska (D400150141) | **Tanggal ACC :** |
|  | Nindya Kaloka (D400170019) | **Revisi Tanggal :** |
|  | Saiful Arifin Wahyu W. (D400160065) |  |

**FINAL PROJECT SISTEM OPERASI MIKROPROSESOR**

**Multitasking dan Real-Time Arduino Sistem**

**Menggunakan Ultrasonic Sensor HC-SR04 dan Temperatur Sensor LM35**

1. **ALAT DAN BAHAN**
2. Laptop / PC yang telah diinstal software Arduino
3. Arduino Uno with library FreeRTOS
4. LED (3 buah)
5. Sensor Ultrasonik HC-SR04
6. Sensor Temperatur LM 35
7. Resistor 220 ohm (3 buah)
8. Kabel jumper
9. **GAMBAR RANGKAIAN**



1. **HASIL PERCOBAAN**

#include <Arduino\_FreeRTOS.h>

//Pin definition

static int trigger = 9; // Pin HC-SR04

static int echo = 8; // Pin HC-SR04

static int led\_HC = 10; // Pin HC-SR04

static int led\_pulsed = 13; // Timed pulsed led

static int led\_temperature = 12; // Led to temperature overflow

static int LM35\_analogPin = 0; // analog pin for temperature sensor

// Variables for calculating the distance with sensor HC-SR04

long duration; // Flight time of signal

long distance; // Calculated distance

float temp\_C; // temperature variable (Celsius degrees)

void Task1(void \*pvParameters);

void Task2(void \*pvParameters);

void Task3(void \*pvParameters);

void setup() {

//Serial communication

Serial.begin(9600);

//Pin setup

pinMode(trigger, OUTPUT);

pinMode(echo, INPUT);

pinMode(led\_HC, OUTPUT);

pinMode(led\_pulsed, OUTPUT);

pinMode(led\_temperature, OUTPUT);

xTaskCreate(

Task1

, (const portCHAR \*)"Task1"

, 128

, NULL

, 1

, NULL );

xTaskCreate(

Task2

, (const portCHAR \*)"Task2"

, 128

, NULL

, 1

, NULL );

xTaskCreate(

Task3

, (const portCHAR \*)"Task3"

, 128

, NULL

, 1

, NULL );

digitalWrite(echo, LOW);

digitalWrite(trigger, LOW);

digitalWrite(led\_HC, LOW);

digitalWrite(led\_pulsed, LOW);

digitalWrite(led\_temperature, LOW);

//Analog input

analogReference(INTERNAL); // prendo la misura analogica di riferimento di Arduino a 1.1V

//Variable initialization of the distance

duration = 0;

distance = 0;

}

void loop() {

}

void Task1(void \*pvParameters) {

(void) pvParameters;

//Send a HIGH pulse to the trigger pin

digitalWrite(trigger, HIGH);

//I leave it to the HIGH value for 10 microseconds

delayMicroseconds(10);

//I carry it back to the LOW state

digitalWrite(trigger, LOW);

//I get the number of microseconds for which the echo PIN is left in HIGH state

duration = pulseIn(echo, HIGH);

/\*Sound velocity is 340 meters per second, or 29 microseconds per cent.

Our impulse travels back and forth, so to calculate the distance

Between the sensor and our obstacle we need to do:\*/

distance = duration / 29 / 2;

// Turn on the led if there is an obstacle at a distance of less than 10 cm

if (distance < 10) {

digitalWrite(led\_HC, HIGH);

}

else {

digitalWrite(led\_HC, LOW);

}

// Print on the serial buffer

Serial.print("duration : ");

Serial.print(duration);

Serial.print(" - distance : ");

Serial.println(distance);

delay(3000);

}

void Task2(void \*pvParameters) {

(void) pvParameters;

int i;

Serial.println("Pulsed LED");

// Code for the flashing of the led

for (i = 0; i < 5; i++)

{

digitalWrite(led\_pulsed, HIGH);

delay(500);

digitalWrite(led\_pulsed, LOW);

delay(500);

}

delay(7000);

}

void Task3(void \*pvParameters) {

(void) pvParameters;

float sum\_temp = 0;

float average\_temp = 0;

/\* Acquire 3 values from the sensor every 500 milliseconds.

Through an appropriate conversion I get the measure in degrees Celsius.

Calculate the average of subsequent measurements.\*/

for (int i = 0; i < 3; i++){

delay(500);

temp\_C = (1.1 \* analogRead(LM35\_analogPin) \* 100.0) / 1024;

sum\_temp += temp\_C;

}

average\_temp = sum\_temp / 3;

// If the temperature is greater than 23 degrees, turn on the led

if (average\_temp > 23) {

digitalWrite(led\_temperature, HIGH);

}

else {

digitalWrite(led\_temperature, LOW);

}

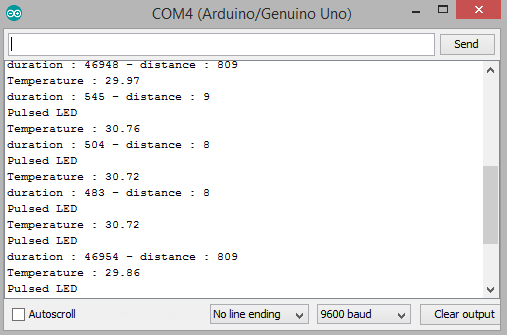
Serial.print("Temperature : ");

Serial.println(average\_temp);

delay(11000);

}

1. **DATA HASIL**



1. **PRINSIP KERJA**

Pada final project Multitasking dan Real-Time Arduino Sistem Menggunakan Ultrasonic Sensor HC-SR04 dan Temperatur Sensor LM35 menggunakan beberapa perangkat hardware diantaranya Kit Arduino Uno dengan library FreeRTOS, LED, Resistor 220 ohm, Sensor Ultrasonic, dan sensor temperatur. Setelah semua terpasang pada project board dan run program pada arduino kemuadian pada serial monitor muncul Temperatur, Durasi dan Jarak yang terukur. LED hijau berfungsi sebagai indikator Pengukuran jarak. Jika pengukuran jarak kurang dari 10cm maka LED hijau akan menyala. LED kuning berfungsi sebagai penampilan data pada serial monitor yang diatur setiap 7ms. LED merah berfungsi sebagai indikator pengukuran suhu. LED akan berkedip ketika suhu melebihi 23ºC. Semua proses dikelola secara real time.