REPORT

PROJECT INTERNSHIP
at ATMDUINO
ROBOTIC

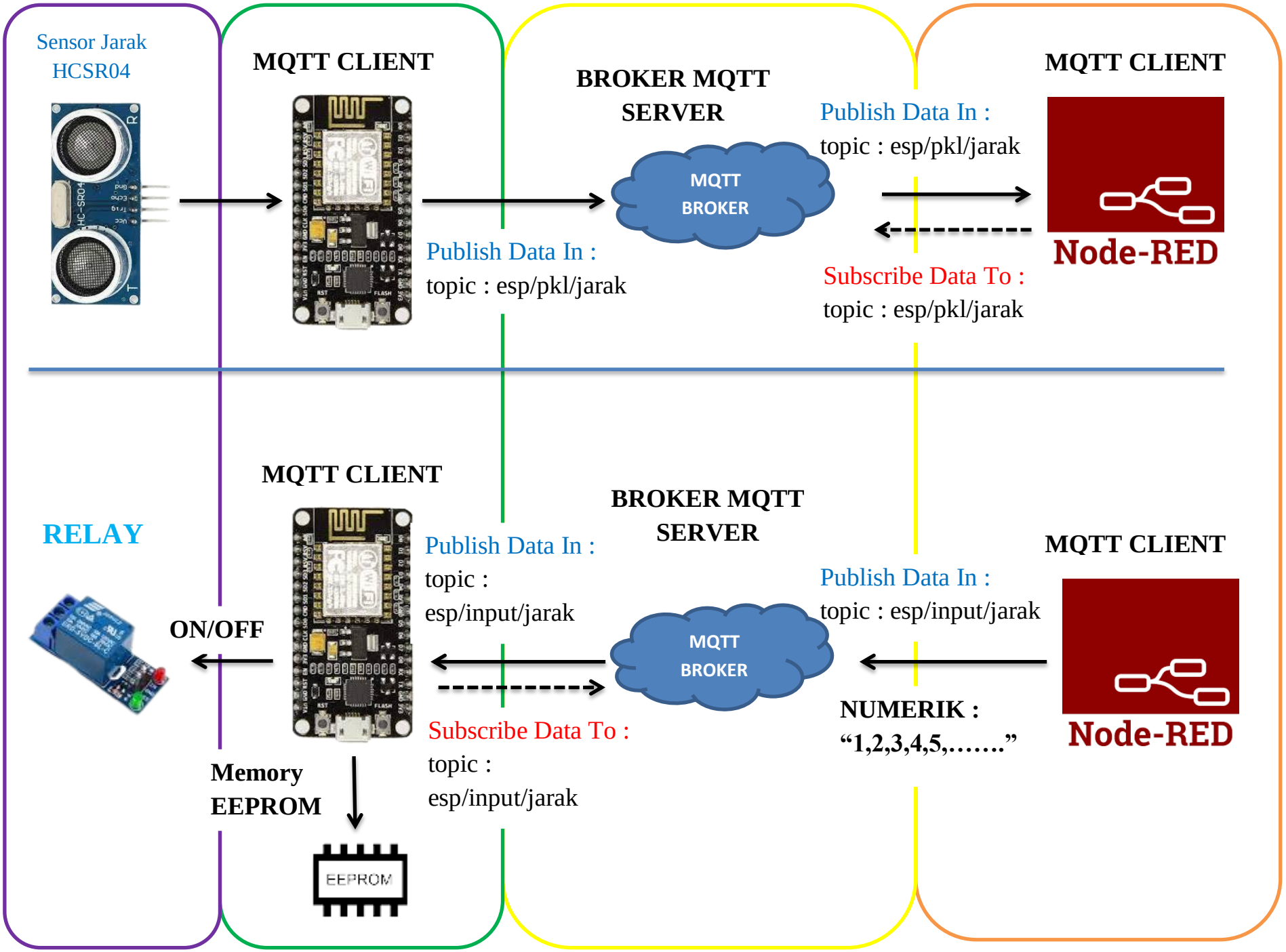
MOH. ALI MAKHRUS 20202152
SISTEM KOMPUTER
ITB ASIA MALANG

Controlling hcsr04 proximity sensor max value from esp8266 & node-red with mqtt (Publish & Subscribe) and stored in eeprom

In this report I will explain about how to control the value inputted by node-red to esp8266 via the mqtt protocol as the maximum parameter for the distance sensor value and output via relay.

below will show the schema that will be used in this report.

SATURASA


Sensor Jarak
HCSR04
MQTT CLIENT
BROKER MQTT
SERVER
MQTT CLIENT
Publish Data In :
topic : esp/pkl/jarak
MQTT
BROKER
Node-RED
Publish Data In :
topic : esp/pkl/jarak
Subscribe Data To :
topic : esp/pkl/jarak
MQTT CLIENT
BROKER MQTT
SERVER
MQTT CLIENT
RELAY
Publish Data In :
topic :
esp/input/jarak
Publish Data In :
topic : esp/input/jarak
ON/OFF
MQTT
BROKER
Node-RED
NUMERIK :
"1,2,3,4,5,……."
Subscribe Data To :
topic :
esp/input/jarak
Memory
EEPROM
EEPROM

MQTT protocol

MQTT (MQ Telemetry Transport) is a messaging protocol built on TCP/IP based on the publish-subscribe messaging model. Publishers send messages, subscribers receive messages they like, and brokers pass messages from sender to recipient. This protocol is very supportive for WAN networks, because WANs cover a wide area. It is a simple messaging protocol, designed for limited devices and with low bandwidth has a very small header size (2 bytes) and saves device resource requirements. So it is suitable for Internet of Things applications.

MQTT makes it easy to encrypt messages using TLS and authenticate clients using modern authentication protocols, such as OAuth. High Scalability. MQTT can support millions of devices that can be connected at the same time because of its vertical and horizontal scalability.

What is the difference between MQTT and HTTP protocols?

The MQTT protocol can send data to the database, which is 152.2 data, while the HTTP protocol can send 58.4 data within one minute. So the MQTT protocol can send more data than the HTTP protocol.

in this report I use a free public broker server is **broker.mqtt-dashboard.com**

SATURASA

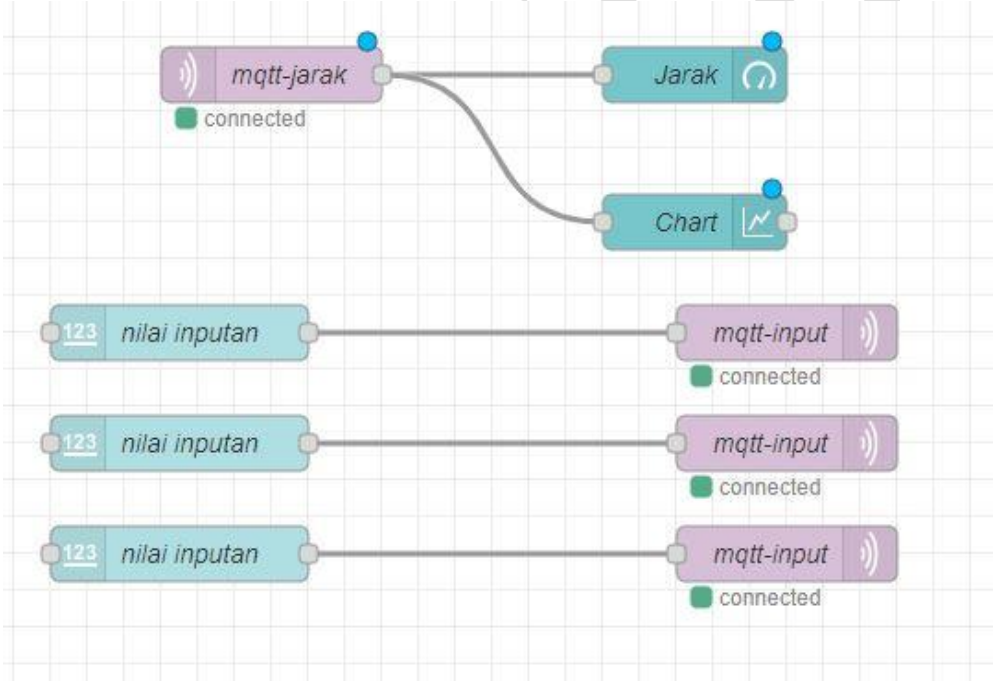
NODE-RED

Node-Red is one of the programming tools that can connect hardware, APIs in a very easy way just by drag and drop. Flow Editor on node-red based on web browser application and Created based on nodejs.

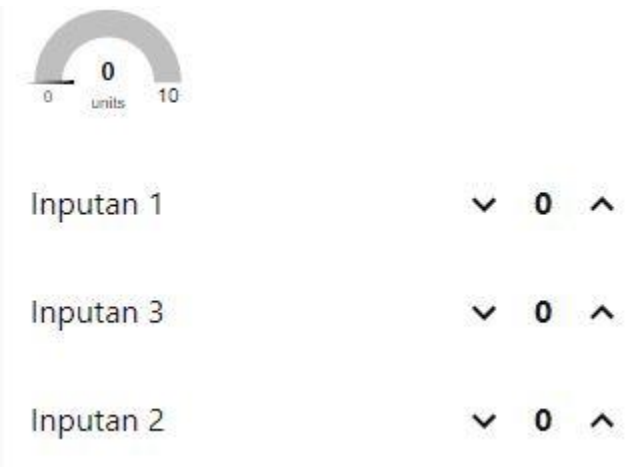
Node-Red can also collect data from various sensor nodes. This sensor node can be a sensor installed with a controller and a network communication module (either WiFi or Ethernet). For example the sensor that comes with the ESP8266.

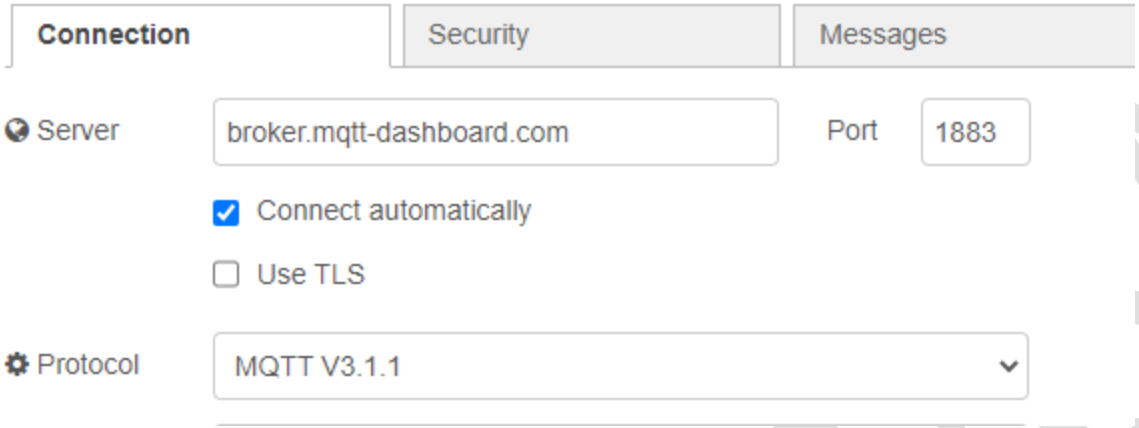
Node-Red-ui can also present the data collected in various visual data, it can be in the form of a line chart, a meter (gauge).

Dashboard Layout in this project.



Display on node-red-ui





In this mqtt node for server broker I use free public **broker.mqtt-dashboard.com** port 1883

flow description above :

1. numeric node – this will input a numeric value
2. 3x mqtt output node – this will publish numeric value messages to ESP8266 according to the topics esp/input/jarak, esp/input/2jarak and esp/input/3jarak

3. mqtt input node – this node will subscribe to the topic esp/pkl/jarak to receive proximity sensor data from ESP

4. gauge – will display the proximity sensor reading

SATURAJA

Program Code in Arduino IDE

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <HCSR04.h>
#include <EEPROM.h>
const char *ssid = "PKL";
const char *password = "12345678";
const char *MQTT_SERVER = "broker.mqtt-dashboard.com";
const char *RELAY = "esp/input/jarak";
#define MQTT_JARAK "esp/pkl/jarak"
#define relayPin D0
#define triggerPin D1
#define echoPin D2
long duration;
int jarak1;
unsigned long startMillis=0;
int a,b,c;
WiFiClient espClient;
PubSubClient client(espClient);
void setup_wifi() {
  delay(10);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  randomSeed(micros());
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}
```

SATURASA

```
void jarak_sensor(){
  long duration;
  int jarak1;
  unsigned long currentMillis = millis();
  if (currentMillis - startMillis >= 2000) {
    startMillis = currentMillis;
    digitalWrite(triggerPin, LOW);
    delayMicroseconds(2);
    digitalWrite(triggerPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(triggerPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    jarak1 = (duration/2) / 29.1;
     Serial.print("Jarak :");
    Serial.print(jarak1);
    Serial.println(" cm");
    client.publish(MQTT_JARAK, String(jarak1).c_str());
    client.publish(MQTT_DURASI, String(duration).c_str());
    if (jarak1 >= EEPROM.read(0)){
       digitalWrite(relayPin, HIGH);
       Serial.println("Relay Mati");
     }
     else if (jarak1 < EEPROM.read(0)){
       digitalWrite(relayPin, LOW);
       Serial.println("Relay Hidup");
     }
     else {
       Serial.println("");
     }
   }
}
```

SATURASA

```
void reconnect() {
 while (!client.connected()) {
Serial.print("Attempting MQTT connection...");
    if (client.connect("espClient"))
    {
      Serial.println("connected");
      client.subscribe("esp/input/jarak");
      client.subscribe("esp/input/2jarak");
      client.subscribe("esp/input/3jarak");
    }
    else
    {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      delay(5000);
    }
  }
}
```

```
void setup() {
 pinMode(relayPin, OUTPUT);
 pinMode(triggerPin, OUTPUT);
 pinMode(echoPin, INPUT);
 setup_wifi();
 Serial.begin(115200);
 EEPROM.begin(512);
 client.setServer(MQTT_SERVER, 1883);
 client.setCallback(callback);
 Serial.print("IP address: ");
 Serial.println(WiFi.localIP());
 Serial.print("MAC Address: ");
 Serial.println(WiFi.macAddress());
 WiFi.setAutoReconnect(true);
 WiFi.persistent(true);
```

SATURASA

```
EEPROM.read(0);
 Serial.print("Nilai Awal A : ");
 Serial.println(EEPROM.read(0));
 EEPROM.read(5);
 Serial.print("Nilai Awal B : ");
 Serial.println(EEPROM.read(5));
 EEPROM.read(9);
 Serial.print("Nilai Awal C : ");
 Serial.println(EEPROM.read(9));
 delay(1000);
 }
void callback(char *topic, byte *message, unsigned int length)
{
  Serial.print("Message arrived in topic: ");
  Serial.println(topic);
Serial.print("Message:");
  String payload;
  for (int i = 0; i < length; i++)
  {
   Serial.print((char)message[i]);
   Serial.println();
   if ((char)message[i] != "")
    payload += (char)message[i];
  }
   if (String(topic)=="esp/input/jarak") {
     a = payload.toInt();
      Serial.print("Nilai A = ");
    Serial.println(a);
    EEPROM.write(0, a);
    EEPROM.commit();
    }
   if (String(topic)=="esp/input/2jarak") {
```

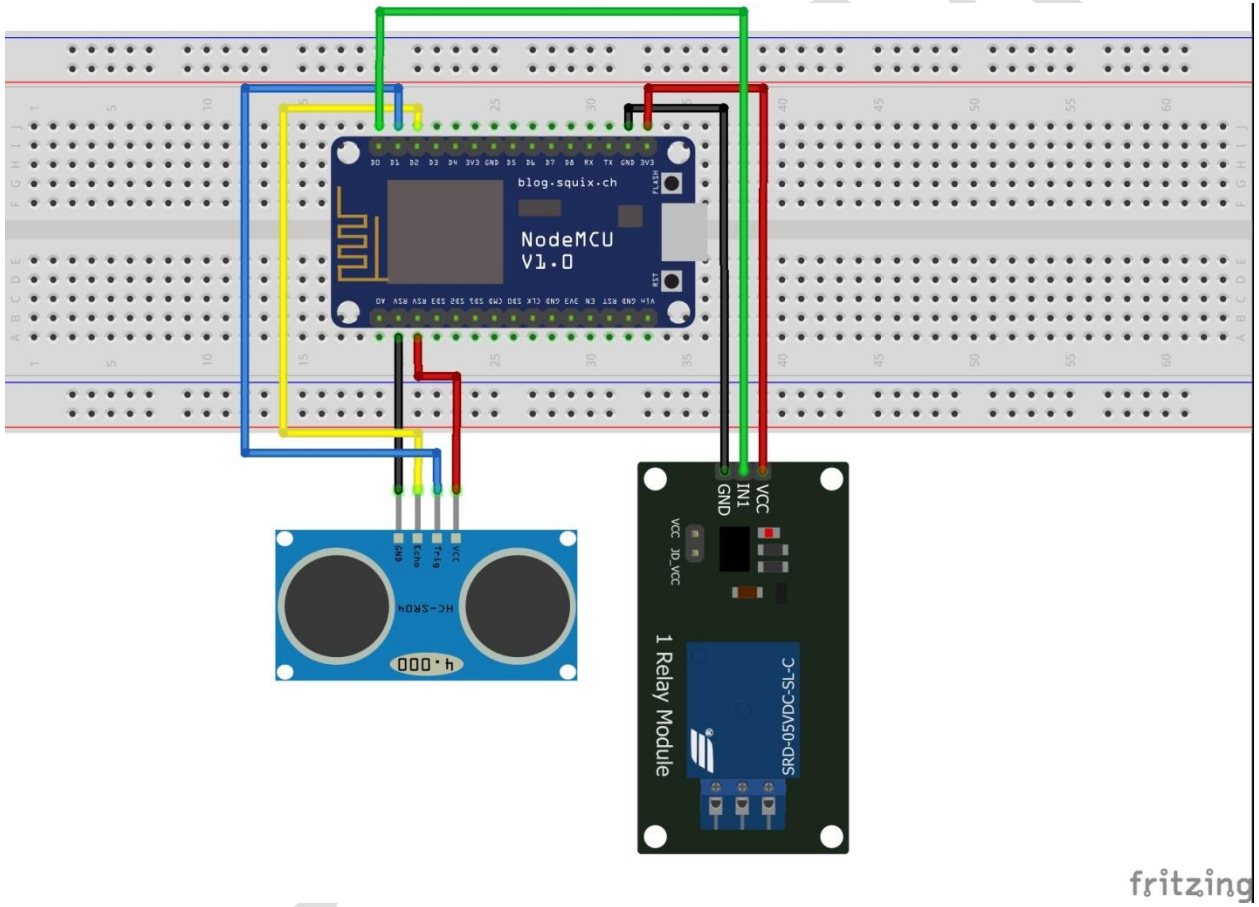
SATURASA

```
    b = payload.toInt();
    Serial.print("Nilai B = ");
    Serial.println(b);
    EEPROM.write(5, b);
    EEPROM.commit();
    }
  if (String(topic)=="esp/input/3jarak") {
      c = payload.toInt();
      Serial.print("Nilai C = ");
      Serial.println(c);
       EEPROM.write(9, c);
       EEPROM.commit();
    }
 }
void loop() {
jarak_sensor();
   if (!client.connected()){
    reconnect();
  }
   client.loop();
}
```

SATURASA

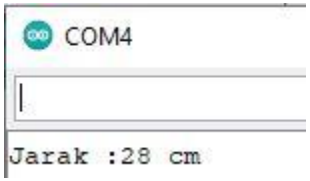
SCHEMATICS

Here are the schematics for this project's circuit.



DEMONSTRATION

1. Upload program code to controller esp8266



2. Deploy flow Node-red

3. Input value(publish) in node-red

SAKURASA

5. Display in Serial Monitor Arduino IDE (Subscribe)

```
Nilai A = 11
Message arrived in topic: esp/input/2jarak
Message:9
Nilai B = 9
Jarak :27 cm
Relay Mati
Message arrived in topic: esp/input/3jarak
Message:1
0
Nilai C = 10
Jarak :28 cm
```

6. Nilai A as the maximum parameter of the distance value

   If jarak>Nilai A

   Relay OFF

```
Relay Mati
Jarak :27 cm
Relay Mati
Jarak :28 cm
Relay Mati
```

   If jarak<Nilai A

   Relay ON

```
Relay Hidup
Jarak :3 cm
Relay Hidup
Jarak :3 cm
Relay Hidup
```

SATURASA

7. With EEPROM it is possible if the controller is turned off the last value of the input node-red is still stored

```
rll§r□$□§n♠♠♠§♠l§♠b|§§□□□§r□b§♠b□§nn□
MAC Address: 40:91:51:51:1A:D7
Nilai Awal A : 11
Nilai Awal B : 9
Nilai Awal C : 10
Jarak :27 cm
Relay Mati
```

SATURASA