

TK2402

ANTARMUKA PENGGUNA DAN PERIFERAL

Hanya dipergunakan di lingkungan Fakultas Ilmu Terapan

| | |
|---------|--|
| Nama : | |
| NIM : | |
| Kelas : | |



Program Studi D3 Teknik Komputer
Fakultas Ilmu Terapan
2015

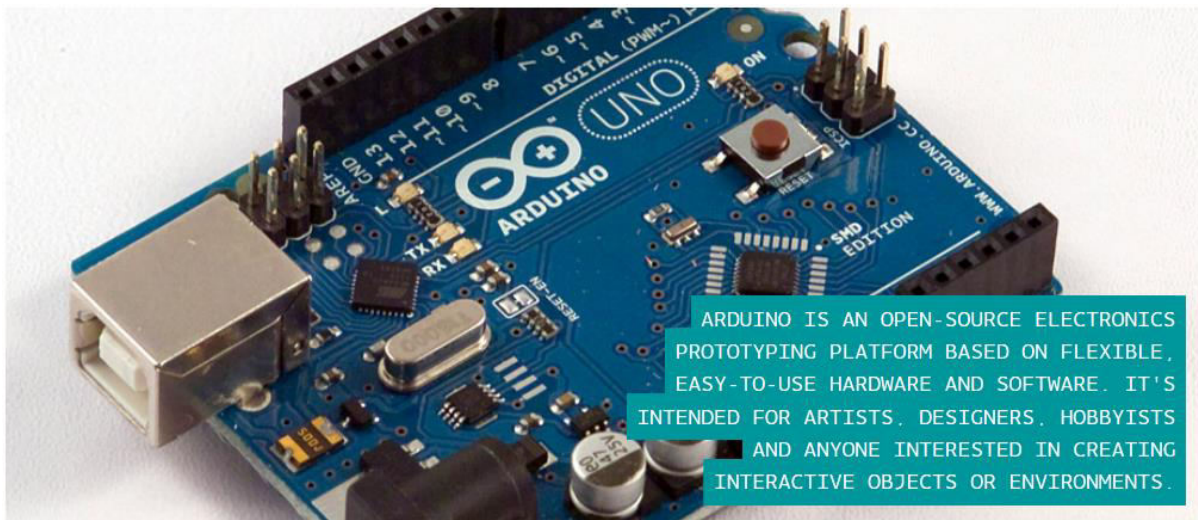
DAFTAR PENYUSUN

1. Versi 1 : 2015 Juli : Simon Siregar, S.Si, M.T.

MODUL PENGENALAN

PRAKTIKUM INTERFACE, PEMOGRAMAN MIKROKONTROLER, DAN SISTEM KENDALI

What is Arduino?



Arduino dikatakan sebagai sebuah *platform* dari *physical computing* yang bersifat *open source*. Pertama-tama perlu dipahami bahwa kata “platform” di sini adalah sebuah pilihan kata yang tepat. Arduino tidak hanya sekedar sebuah alat pengembangan, tetapi ia adalah kombinasi dari hardware, bahasa pemrograman dan **Integrated Development Environment (IDE)** yang canggih.

IDE adalah sebuah software yang sangat berperan untuk menulis program, meng-*compile* menjadi kode biner dan meng-*upload* ke dalam *memory* microcontroller. Ada banyak proyek dan alat-alat dikembangkan oleh akademisi dan profesional dengan menggunakan Arduino, selain itu juga ada banyak modul-modul pendukung (sensor, tampilan, penggerak dan sebagainya) yang dibuat oleh pihak lain untuk bisa disambungkan dengan Arduino. Arduino berevolusi menjadi sebuah platform karena ia menjadi pilihan dan acuan bagi banyak praktisi.

Salah satu yang membuat Arduino memikat hati banyak orang adalah karena sifatnya yang open source, baik untuk hardware maupun software-nya. Diagram rangkaian elektronik Arduino digratiskan kepada semua orang. Anda bisa bebas men-download gambarnya, membeli komponen-komponennya, membuat PCB-nya dan merangkainya sendiri tanpa harus membayar kepada para pembuat Arduino. Sama halnya dengan IDE Arduino yang bisa di-download dan diinstal pada komputer secara gratis. Kita patut berterima kasih kepada tim Arduino yang sangat dermawan membagi-bagikan kemewahan hasil kerja keras mereka kepada semua orang. Saya pribadi betul-betul kagum dengan desain hardware, bahasa pemrograman dan IDE Arduino yang berkualitas tinggi dan sangat berkelas.

Arduino dikembangkan oleh sebuah tim yang beranggotakan orang-orang dari berbagai belahan dunia. Anggota inti dari tim ini adalah:

- Massimo Banzi Milano, Italy
- David Cuartielles Malmoe, Sweden
- Tom Igoe New York, US
- Gianluca Martino Torino, Italy
- David A. Mellis Boston, MA, USA

Profil mengenai anggota tim tersebut dan kontribusinya bisa diakses pada situs web <http://www.arduino.cc/playground/Main/People>.

Saat ini komunitas Arduino berkembang dengan pesat dan dinamis di berbagai belahan dunia. Berbagai macam kegiatan yang berkaitan dengan proyek-proyek Arduino bermunculan dimana-mana, termasuk di Indonesia. Yang membuat Arduino dengan cepat diterima oleh orang-orang adalah :

- Murah, dibandingkan platform yang lain.
- Lintas platform, software Arduino dapat dijalankan pada system operasi Windows, Macintosh OSX dan Linux.
- Sangat mudah dipelajari dan digunakan. **Processing** adalah bahasa pemrograman yang digunakan untuk menulis program di dalam Arduino. Untuk mengenal Processing lebih lanjut, silakan mengunjungi situs web-nya di <http://www.processing.org>.
- Sistem yang terbuka, baik dari sisi hardware maupun software-nya.
- Arduino diperuntukan bagi seniman, perancang dan penemu.

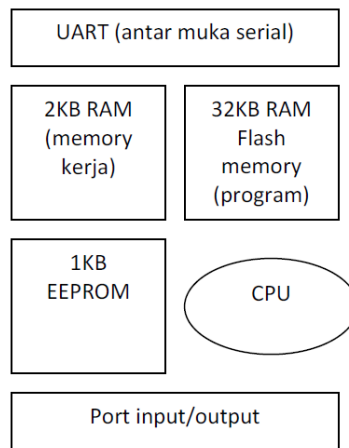
Arduino UNO R3



Secara umum Arduino terdiri dari dua bagian, yaitu:

1. Hardware - papan input/output (I/O)
2. Software - Software Arduino meliputi IDE untuk menulis program, *driver* untuk koneksi dengan komputer, contoh program dan *library* untuk pengembangan program.

Komponen utama di dalam papan Arduino UNO R3 adalah sebuah microcontroller 8 bit dengan merk **ATmega** yang dibuat oleh perusahaan **Atmel Corporation**. Berikut adalah blok sederhana dari Arduino UNO R3 :

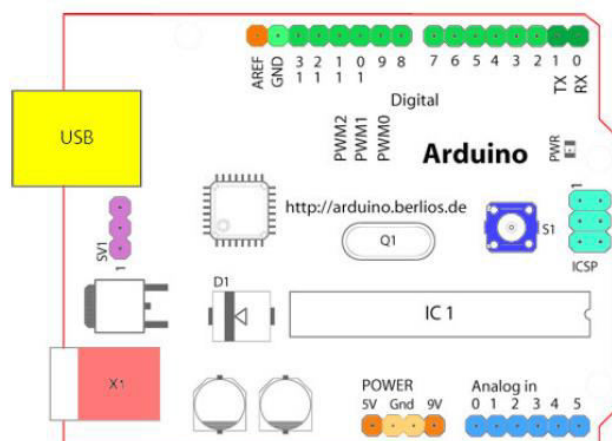


Blok-blok didalam arduino UNO R3 adalah sebagai berikut:

- *Universal Asynchronous Receiver/Transmitter (UART)* adalah antar muka yang digunakan untuk komunikasi serial seperti pada RS-232, RS-422 dan RS-485.
- 2KB RAM pada memory kerja bersifat *volatile* (hilang saat daya dimatikan), digunakan oleh variable-variabel di dalam program.
- 32KB RAM flash memory bersifat *non-volatile*, digunakan untuk menyimpan program yang dimuat dari komputer. Selain program, flash memory juga menyimpan *bootloader*. **Bootloader** adalah program inisiasi yang ukurannya kecil, dijalankan oleh CPU saat daya dihidupkan. Setelah bootloader selesai dijalankan, berikutnya program di dalam RAM akan dieksekusi.
- 1KB EEPROM bersifat non-volatile, digunakan untuk menyimpan data yang tidak boleh hilang saat daya dimatikan. Tidak digunakan pada papan Arduino.
- *Central Processing Unit (CPU)*, bagian dari microcontroller untuk menjalankan setiap instruksi dari program.
- Port input/output, pin-pin untuk menerima data (input) digital atau analog, dan mengeluarkan data (output) digital atau analog.
- Setelah mengenal bagian-bagian utama dari microcontroller ATmega sebagai komponen utama, selanjutnya kita akan mengenal bagian-bagian dari papan Arduino itu sendiri.

BAGIAN-BAGIAN PAPAN ARDUINO

Dengan mengambil contoh sebuah papan Arduino tipe USB, bagian-bagiannya dapat dijelaskan sebagai berikut.



14 PIN INPUT/OUTPUT DIGITAL (0-13)

Berfungsi sebagai input atau output, dapat diatur oleh program.

Khusus untuk 6 buah pin 3, 5, 6, 9, 10 dan 11, dapat juga berfungsi sebagai pin analog output dimana tegangan output-nya dapat diatur. Nilai sebuah pin output analog dapat diprogram antara 0 – 255, dimana hal itu mewakili nilai tegangan 0 – 5V.

6 PIN ANALOG INPUT (A0-A5)

Pin Analog berfungsi sebagai input dan output. Jika menjadi Input, pin analog berfungsi dengan menggunakan ADC. Jika menjadi output, pin analog, berfungsi sebagai digital output

USB

Berfungsi untuk:

1. Memuat program dari komputer ke dalam papan
2. Komunikasi serial antara papan dan komputer
3. Memberi daya listrik kepada papan

SAMBUNGAN SV1

Sambungan atau *jumper* untuk memilih sumber daya papan, apakah dari sumber eksternal atau

menggunakan USB. Sambungan ini tidak diperlukan lagi pada papan Arduino versi terakhir karena pemilihan sumber daya eksternal atau USB dilakukan secara otomatis.

Q1 – KRISTAL (*quartz crystal oscillator*)

Jika microcontroller dianggap sebagai sebuah otak, maka kristal adalah jantung-nya karena komponen ini menghasilkan detak-detak yang dikirim kepada microcontroller agar melakukan

sebuah operasi untuk setiap detak-nya. Kristal ini dipilih yang berdetak 16 juta kali per detik (16MHz).

TOMBOL RESET S1

Untuk me-reset papan sehingga program akan mulai lagi dari awal. Perhatikan bahwa tombol reset ini bukan untuk menghapus program atau mengosongkan microcontroller.

In-Circuit Serial Programming (ICSP)

Port ICSP memungkinkan pengguna untuk memprogram microcontroller secara langsung, tanpa melalui bootloader. Umumnya pengguna Arduino tidak melakukan ini sehingga ICSP tidak terlalu dipakai walaupun disediakan.

IC 1 – Microcontroller Atmega

Komponen utama dari papan Arduino, di dalamnya terdapat CPU, ROM dan RAM. Jika hendak disuplai dengan sumber daya eksternal, papan Arduino dapat diberikan tegangan

DC ANTARA 9-12V.

Pin ini sangat berguna untuk membaca tegangan yang dihasilkan oleh sensor analog, seperti sensor suhu. Program dapat membaca nilai sebuah pin input antara 0 – 1023, dimana hal itu mewakili nilai tegangan 0 – 5V.

4. SOFTWARE ARDUINO

Sehubungan dengan pembahasan untuk saat ini software Arduino yang akan digunakan adalah

driver dan IDE, walaupun masih ada beberapa software lain yang sangat berguna selama pengembangan Arduino.

IDE Arduino adalah software yang sangat canggih ditulis dengan menggunakan Java. IDE Arduino

terdiri dari:

1. *Editor* program, sebuah window yang memungkinkan pengguna menulis dan mengedit program dalam bahasa Processing.
2. *Compiler*, sebuah modul yang mengubah kode program (bahasa Processing) menjadi kode biner. Bagaimanapun sebuah microcontroller tidak akan bisa memahami bahasa Processing. Yang bisa dipahami oleh microcontroller adalah kode biner. Itulah sebabnya compiler diperlukan dalam hal ini.
3. *Uploader*, sebuah modul yang memuat kode biner dari Jomputer ke dalam memory di dalam papan Arduino.
4. Menginstall driver Arduino
5. Menguji koneksi komputer dan papan Arduino

Urutan penulisan software Arduino beserta contohnya :

INCLUDE LIBRARY

```
#include <Servo.h>
#include <EEPROM.h>
```

VARIABEL

```
int led = 13;
char key, keyR, keyUR;
```

INISIALISASI

```
void setup() {
    // set up the LCD's number of columns and rows:
    lcd.begin(16, 2);
    // Print a message to the LCD.
    lcd.print("hello, world!");
}
```

PROSES BERULANG HINGGA AKHIR

```
void loop() {
    // Turn off the blinking cursor:
    lcd.noBlink();
    delay(3000);
    // Turn on the blinking cursor:
    lcd.blink();
    delay(3000);
}
```


MODUL 1

PRAKTIKUM INTERFACE, PEMOGRAMAN MIKROKONTROLER, DAN SISTEM KENDALI, FAKULTAS ILMU TERAPAN, UNIVERSITAS TELKOM

1. JUDUL PRAKTIKUM

Pengenalan PORT Digital dan Analog, LED, Push Button, Potensiometer

2. MAKSUD DAN TUJUAN

Maksud dan tujuan dari praktikum ini adalah :

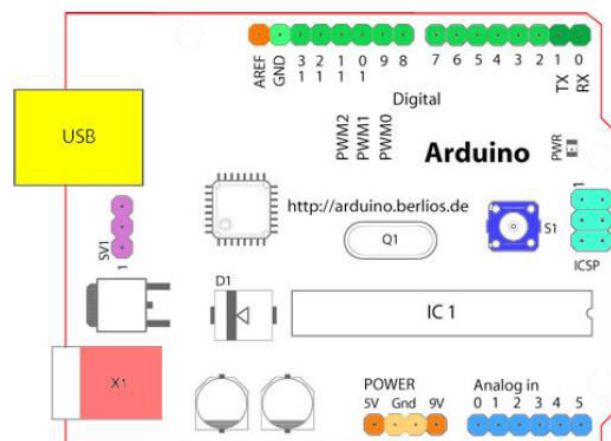
1. Mahasiswa mengenal jenis-jenis dan fungsi pin pada mikrokontroler berbasis Atmega 328 (Arduino Uno)
2. Mahasiswa mampu menggunakan pin-pin pada mikrokontroler sesuai dengan fungsinya
3. Mahasiswa mampu menyelesaikan kasus tertentu dengan menggunakan setiap fungsi pin-pin dalam mikrokontroler

3. PERALATAN DAN BAHAN

Peralatan yang dibutuhkan dalam praktikum ini adalah :

1. 1 buah Arduino Uno R3 + Kabel USB
2. Jumper Secukupnya
3. 4 buah Resistor 330 Ohm
4. 3 buah LED
5. 1 buah potensioresistor
6. 1 buah Protoboard

4. TEORI DASAR



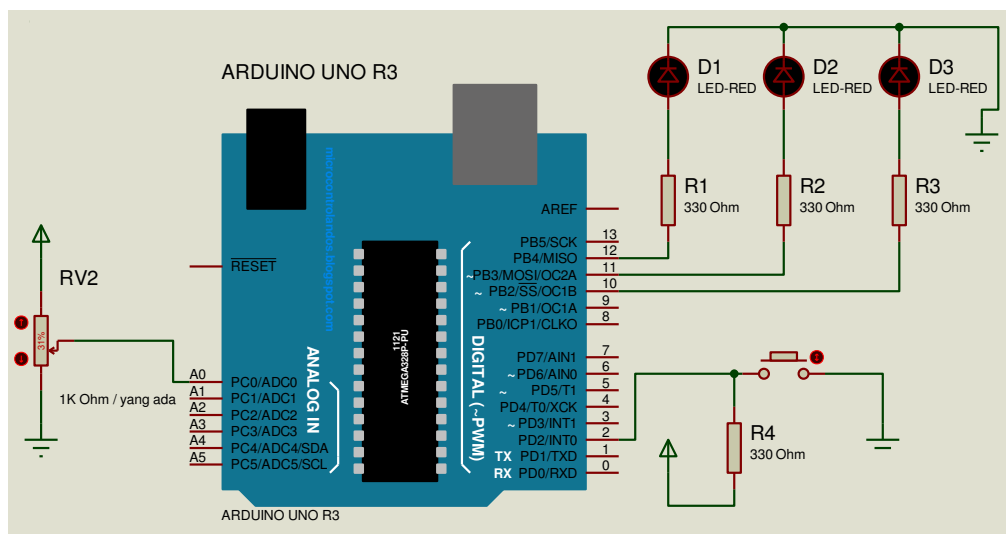
Arduino secara umum memiliki :

- 14 PIN INPUT/OUTPUT DIGITAL (0-13)
Berfungsi sebagai input atau output, dapat diatur oleh program.

- 6 PIN PWM (dalam 14 Pin I/O Digital)
Khusus untuk 6 buah pin 3, 5, 6, 9, 10 dan 11, dapat juga berfungsi sebagai pin analog output dimana tegangan output-nya dapat diatur. Nilai sebuah pin output analog dapat diprogram antara 0 – 255, dimana hal itu mewakili nilai tegangan 0 – 5V.
- 6 PIN ANALOG INPUT (A0-A5)
Pin Analog berfungsi sebagai input dan output. Jika menjadi Input, pin analog berfungsi dengan menggunakan ADC. Jika menjadi output, pin analog, berfungsi sebagai digital output

5. PROSEDUR PRAKTIKUM

A. Buat rangkaian sesuai dengan skematik berikut :



B. Percobaan dalam praktikum

1. Digital Port / On-Off LED

a. Tuliskan program dibawah ini pada software *Arduino* dan upload keboard *Arduino Uno R3* :

```
int led=11;

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

void loop() {
  digitalWrite(led, HIGH);
  delay(1000);
  digitalWrite(led, LOW);
  delay(1000);
}
```

b. Tuliskan komentar pada list program tersebut dengan mencoba menggantikan nilai A, B, C dan D

```
int led=A; // .....

void setup() { // .....
  pinMode(led, B); // .....
}
```

```

void loop(){ // .....
    digitalWrite(led,HIGH); // .....
    delay(C); // .....
    digitalWrite(led,LOW); // .....
    delay(D); // .....
}

```

- c. Buat program untuk menyalakan ketiga LED dengan aturan :
 - Secara bersama-sama dengan durasi on selama 2 detik dan off selama 1 detik.
 - Setiap Led dapat menyala dan mati secara bergantian dengan durasi 500 milidetik menyala dan mati. (*Running LED*)
- d. Catat program yang dibuat pada percobaan c dan perlihatkan pada asisten.
 - Jelaskan perubahan dalam program untuk menjalankan prosedur c.

2. Digital Port / PWM - Fading LED

- a. Tuliskan program dibawah ini pada software *Arduino* dan upload keyboard Arduino Uno R3 :

```

int led=11;
int i;
void setup(){
    pinMode(led,OUTPUT);
}

void loop(){
    for (i=0; i<=255; i+=5){
        analogWrite(led,i);
        delay(100);
    }
    for (i=255; i>=0; i-=5){
        analogWrite(led,i);
        delay(100);
    }
}

```

- b. Tuliskan komentar pada list program tersebut dengan mencoba menggantikan nilai A, B, C dan D

```

int led=11;
int A; // .....
void setup(){
    pinMode(led,OUTPUT);
}

void loop(){
    for (i=B; i<=C; i+=D){ // .....
        analogWrite(led,i); // .....
        delay(100);
    }
    for (i=255; i>=0; i-=5){ // .....
        analogWrite(led,i); // .....
        delay(100);
    }
}

```

- c. Buat program untuk menyalakan ketiga LED dengan aturan :
 - Secara bersama-sama dengan durasi fading on selama 2 detik dan fading off selama 1 detik.
 - Setiap Led dapat fading menyala dan mati secara bergantian dengan durasi 500 milidetik menyala dan mati.
- d. Catat program yang dibuat pada percobaan c dan perlihatkan pada asisten.
 - Jelaskan apakah percobaan dapat berjalan sesuai dengan prosedur?
 - Adakah yang perlu dirubah dalam rangkaian?
 - Berapa nilai maksimum dari nilai analog write yang bisa digunakan.

3. Analog Port / Input – Potensiometer LED

- a. Tuliskan program dibawah ini pada software *Arduino* dan upload keyboard Arduino Uno R3 :

```
int led=11;
int pot=A0;
int data;

void setup(){
  pinMode(led,OUTPUT);
  pinMode(pot,INPUT);
}

void loop(){
  data=analogRead(pot);
  data=data/4;
  analogWrite(led,data);
}
```

- b. Tuliskan komentar pada list program tersebut dengan mencoba menggantikan nilai A, B, C dan D

```
int led=A; //.....
int pot=B; //.....
int data;

void setup(){
  pinMode(led,OUTPUT);
  pinMode(pot,C); //.....
}

void loop(){
  data=analogRead(pot);
  data=D; //.....
  analogWrite(led,data);
}
```

- c. Buat program untuk menyalakan ketiga LED dengan aturan :
 - Secara bersama-sama dengan menggunakan potensio.
 - Setiap Led dapat fading menyala dan mati secara bergantian dengan durasi pengaturan milidetik menyala dan mati menggunakan potensiometer.
- d. Catat program yang dibuat pada percobaan c dan perlihatkan pada asisten.
 - Jelaskan apakah percobaan dapat berjalan sesuai dengan prosedur?
 - Adakah yang perlu dirubah dalam rangkaian?
 - Jelaskan perubahan dalam program untuk menjalankan prosedur c.

4. Digital Port / Input – Push Button LED

- a. Tuliskan program dibawah ini pada software *Arduino* dan upload keyboard *Arduino Uno R3* :

```
int led=11;
int tombol=2;
void setup(){
  pinMode(led,OUTPUT);
  pinMode(tombol,INPUT);
}

void loop(){

  if(digitalRead(tombol)==LOW){
    digitalWrite(led,HIGH);
  }
  else{
    digitalWrite(led,LOW);
  }
}
```

- b. Tuliskan komentar pada list program tersebut dengan mencoba menggantikan nilai A, B, C dan D

```
int led=A; //.....
int tombol=B; //.....
void setup(){
  pinMode(led,OUTPUT);
  pinMode(tombol,INPUT);
}

void loop(){

  if(digitalRead(C)==D){ //.....
    digitalWrite(led,HIGH);
  }
  else{
    digitalWrite(led,LOW);
  }
}
```

- c. Buat program untuk menyalakan ketiga LED dengan aturan :
- Secara bersama-sama dengan menggunakan push button.
 - Setiap Led dapat fading menyala dan mati secara bergantian dengan durasi pengaturan milidetik menyala dan mati ketika push button ditekan.
- d. Catat program yang dibuat pada percobaan c dan perlihatkan pada asisten.
- Jelaskan apakah percobaan dapat berjalan sesuai dengan prosedur?
 - Adakah yang perlu dirubah dalam rangkaian?
 - Jelaskan perubahan dalam program untuk menjalankan prosedur c.

6. KASUS PERCOBAAN

- a. Buat sebuah aplikasi dengan menggunakan 3 LED, 1 Potensio, 1 push button
- b. Terdapat kendali on/off, PWM dan delay
- c. Terdapat interface analog dan digital
- d. Catat skematik beserta pin/port yang digunakan, dan program yang dibuat pada kasus diatas dan perlihatkan pada asisten

7. LAPORAN AKHIR

- a. Sesuaikan dengan aturan praktikum sebelumnya
- b. Lampirkan hasil percobaan, skematik dan program yang diuji beserta komentar
- c. Jelaskan kasus percobaan 6 dari sisi interface dan peripheral, sistem kendali, dan pemrograman mikrokontroler.
- d. Jelaskan fungsi berikut ini dalam laporan
 - i. pinMode
 - ii. analogRead
 - iii. analogWrite
 - iv. digitalWrite
 - v. void Setup ()
 - vi. void loop ()
 - vii. for ()
 - viii. if ()

MODUL 2

PRAKTIKUM INTERFACE, PEMOGRAMAN MIKROKONTROLER, DAN SISTEM KENDALI, FAKULTAS ILMU TERAPAN, UNIVERSITAS TELKOM

1. JUDUL PRAKTIKUM

Modul LCD dan Shift Register LCD

2. MAKSUD DAN TUJUAN

Maksud dan tujuan dari praktikum ini adalah :

1. Mahasiswa mampu menggunakan pin-pin pada mikrokontroler dalam mengendalikan modul LCD
2. Mahasiswa mampu menyelesaikan kasus tertentu dengan menggunakan LCD dalam mikrokontroler.

3. PERALATAN DAN BAHAN

Peralatan yang dibutuhkan dalam praktikum ini adalah :

1. 1 buah Arduino Uno R3 + Kabel USB
2. Jumper Secukupnya
3. 7 buah Resistor 330 Ohm
4. 3 buah LED
5. 1 buah potensio
6. 1 buah Protoboard
7. 1 buah LCD 16x2
8. 1 buah pin header 16x1
9. 1 buah IC Shift register 4094

4. TEORI DASAR

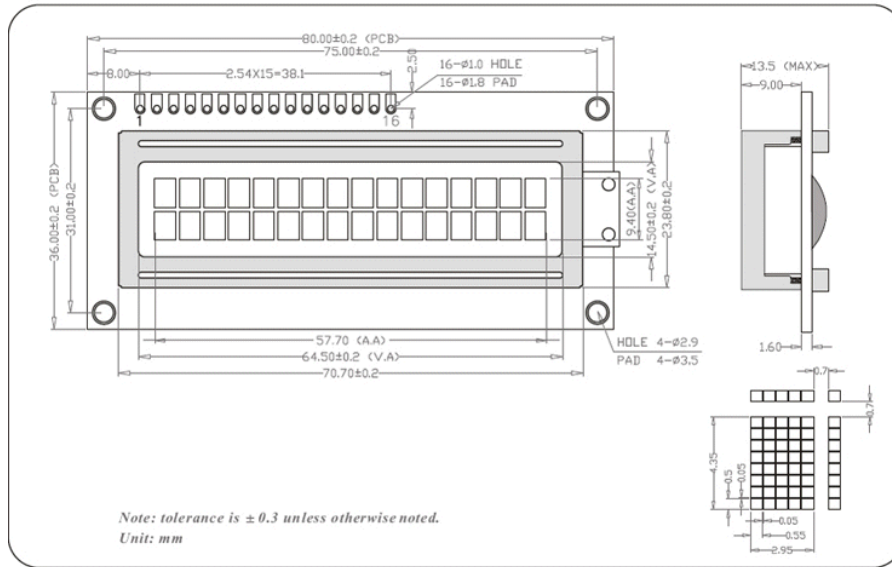
LCD (*Liquid Crystal Display*) merupakan salah satu perangkat untuk menampilkan (display) informasi dari suatu system kontroler.

- | | |
|-----------------|--|
| A. Power Supply | : 4.7 – 5.3 Volt |
| B. Input | : Data dari mikrokontroler |
| C. Output | : Cahaya yang dapat menampilkan karakter atau tulisan. |
| D. Batasan | : Tegangan maksimal 5.3 volt |

LCD secara umum memiliki 16 PIN dengan urutan sebagai berikut :

1. Pin 1 VSS, sebagai input tegangan – LCD (GND)
2. Pin 2 VDD, sebagai input tegangan + LCD
3. Pin 3 Vo, sebagai pengatur intensitas tulisan LCD
4. Pin 4 RS, sebagai pengatur clock LCD
5. Pin 5 R/W, sebagai pengatur aliran data LCD
6. Pin 6 E, sebagai enabler LCD, Active Low
7. Pin 7 – Pin 14 : Data 0 – Data 7
8. Pin 15, Tegangan Backlight LED +
9. Pin 16, Ground Backlight LED -

■ DIMENSIONS/DISPLAY CONTENT

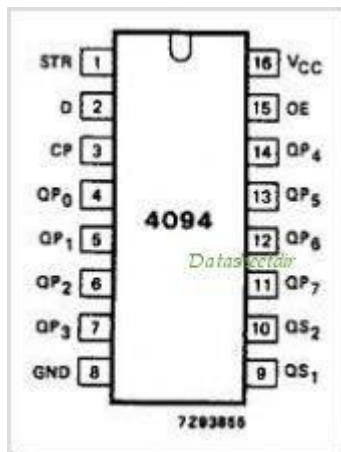


■ PIN CONFIGURATION

| | | | | | | | | | | | | | | | |
|-----|-----|----|----|-----|---|-----|-----|-----|-----|-----|-----|-----|-----|------|------|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| Vss | VDD | Vo | RS | R/W | E | DB0 | DB1 | DB2 | DB3 | DB4 | DB5 | DB6 | DB7 | LEDA | LEDK |

Shift Register 4094 merupakan shift register tipe serial in, paralel out. IC shift register adalah sebuah komponen elektronik (IC) yang digunakan untuk memasukkan data secara serial dan mengeluarkan data secara paralel.

Konfigurasi Pin 4094

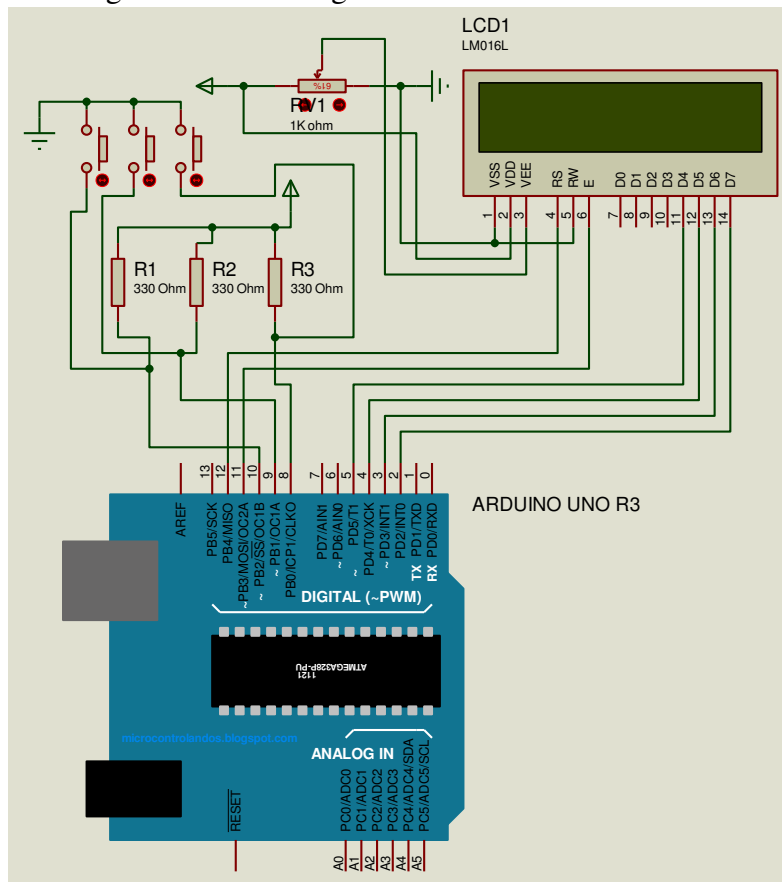


Keterangan Pin

- OE : Output enable
- QP0-QP7 : Output Paralel 0 -
- D : Input Data Serial
- CP : Clock Input
- QS1-QS2 : Output Serial1 -
- STR : Strobe Input
- VCC : V+
- GND : Ground

5. PROSEDUR PRAKTIKUM

A. Buat rangkaian sesuai dengan skematik berikut :



Catatan : ditambahkan hambatan 330 Ohm pada input LED + pada kaki 15 LCD

B. Percobaan dalam praktikum

1. Hello world LCD

a. Tuliskan program dibawah ini pada software *Arduino* dan upload keyboard Arduino Uno R3 :

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

```
void setup() {
  lcd.begin(16, 2);
  lcd.print("hello, world!");
}
```

```
void loop() {
  lcd.setCursor(0, 1);
  lcd.print(millis()/1000);
}
```

b. Tuliskan komentar pada list program tersebut dengan mencoba menggantikan nilai A, B, C dan D

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(A, A, A, A, A, A); //.....
```

```
void setup() {
  lcd.begin(16, B); //.....
```

```

    lcd.print(C); //.....
}

void loop() {
    lcd.setCursor(D); //.....
    lcd.print(millis()/1000); //.....
}

```

- c. Buat program untuk memunculkan tulisan di LCD dengan aturan :
- Ketika tombol 1 ditekan tulisan akan muncul di baris pertama LCD : tombol 1 ditekan
 - Ketika tombol 2 ditekan tulisan akan muncul di baris kedua LCD : tombol 2 ditekan
 - Fungsi `lcd.clear()`; untuk menghapus semua tulisan yang ada di LCD. Tambahkan didalam program sehingga ketika tombol 3 ditekan tulisan akan hilang.
- d. Catat program yang dibuat pada percobaan c dan perlihatkan pada asisten.
- Jelaskan perubahan dalam program untuk menjalankan prosedur c.

2. Running Text LCD

- a. Tuliskan program dibawah ini pada software *Arduino* dan upload keyboard

Arduino Uno R3 :

```

#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup() {
    lcd.begin(16, 2);
    lcd.print("hello, world!");
    delay(1000);
}
void loop() {
    for (int positionCounter = 0; positionCounter < 13;
        positionCounter++) {
        lcd.scrollDisplayLeft();
        delay(150);
    }
    for (int positionCounter = 0; positionCounter < 29;
        positionCounter++) {
        lcd.scrollDisplayRight();
        delay(150);
    }
    for (int positionCounter = 0; positionCounter < 16;
        positionCounter++) {
        lcd.scrollDisplayLeft();
        delay(150);
    }
    delay(1000);
}

```

- b. Tuliskan komentar pada list program pada bagian yang ditentukan

```

#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup() {
    lcd.begin(16, 2);
    lcd.print("hello, world!");
    delay(1000);
}
void loop() { //.....

```

```

for (int positionCounter = 0; positionCounter < 13;
    positionCounter++) { //.....
    lcd.scrollDisplayLeft(); //.....
    delay(150);
}
for (int positionCounter = 0; positionCounter < 29;
    positionCounter++) { //.....
    lcd.scrollDisplayRight(); //.....
    delay(150);
}
for (int positionCounter = 0; positionCounter < 16;
    positionCounter++) { //.....
    lcd.scrollDisplayLeft(); //.....
    delay(150);
}
delay(1000);
}

```

- c. Buat program untuk memunculkan tulisan di LCD dengan aturan :
 - Gerakan Running Text dengan arah gerakan text diatur oleh tombol 1 sebagai penentu arah kanan dan tombol 2 sebagai penentu arah kiri
- d. Catat program yang dibuat pada percobaan c dan perlihatkan pada asisten.
 - Jelaskan perubahan dalam program untuk menjalankan prosedur c.

3. Serial text - LCD

- a. Tuliskan program dibawah ini pada software *Arduino* dan upload keyboard **Arduino Uno R3** :

```

#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup(){
    lcd.begin(16, 2);
    Serial.begin(9600);
}
void loop()
{
    if (Serial.available()) {
        delay(100);
        lcd.clear();
        while (Serial.available() > 0) {
            lcd.write(Serial.read());
        }
    }
}

```

- b. Tuliskan komentar pada list program pada bagian yang ditentukan

```

#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup(){
    lcd.begin(16, 2);
    Serial.begin(9600);
}
void loop()
{
    if (Serial.available()) { //.....
        delay(100);
        lcd.clear();
    }
}

```

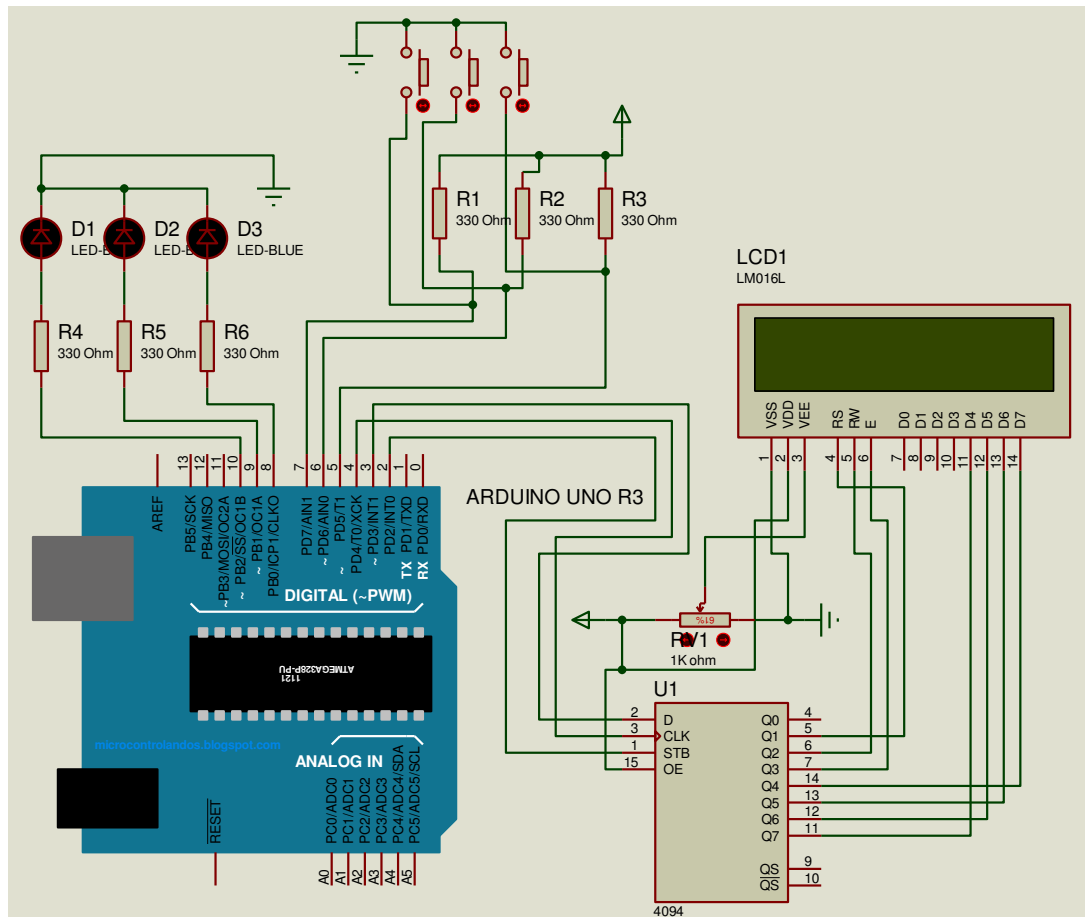
```

while (Serial.available() > 0) { //.....
  lcd.write(Serial.read()); //.....
}
}
}
}

```

- c. Buat program untuk memunculkan tulisan di LCD dengan aturan :
 - Start program dengan memunculkan tulisan “Hallo”, kemudian input text dari serial akan menghapus tulisan sebelumnya dan memunculkan tulisan baru, tulisan tersebut akan bergerak (*running text*) jika tombol 1 ditekan.
- d. Catat program yang dibuat pada percobaan c dan perlihatkan pada asisten.
 - Jelaskan perubahan dalam program untuk menjalankan prosedur c.

C. Buat rangkaian sesuai dengan skematik berikut :



Catatan : ditambahkan hambatan 330 Ohm pada input LED + pada kaki 15 LCD

D. Percobaan dalam praktikum

1. Hello World, Running Text, Serial Shift Register LCD

- a. Tuliskan program dibawah ini pada software *Arduino* dan upload keyboard *Arduino Uno R3* :

```

#include <LiquidCrystal_SR_LCD3.h>
const int PIN_LCD_STROBE = 2;
const int PIN_LCD_DATA = 3;
const int PIN_LCD_CLOCK = 4;

```

```

LiquidCrystal_SR_LCD3 lcd(PIN_LCD_DATA, PIN_LCD_CLOCK,
PIN_LCD_STROBE);

void setup(){
  lcd.begin(16,2);
  lcd.home ();
  lcd.setCursor (0, 0);
  lcd.print("HALLO WORLD");
}

void loop()
{
  lcd.setCursor(0,1);
  lcd.print(millis()/1000);
}

```

b. Tuliskan komentar pada list program pada bagian yang ditentukan

```

#include <LiquidCrystal_SR_LCD3.h> //.....
const int PIN_LCD_STROBE      = 2; //.....
const int PIN_LCD_DATA        = 3; //.....
const int PIN_LCD_CLOCK       = 4; //.....
LiquidCrystal_SR_LCD3 lcd(PIN_LCD_DATA, PIN_LCD_CLOCK,
PIN_LCD_STROBE); //.....

void setup(){
  lcd.begin(16,2);
  lcd.home ();
  lcd.setCursor (0, 0);
  lcd.print("HALLO WORLD");
}

void loop()
{
  lcd.setCursor(0,1);
  lcd.print(millis()/1000);
}

```

c. Buat program untuk menyalakan ketiga LED dengan aturan :

- Hello world Shift register LCD :
 - i. Ketika tombol 1 ditekan tulisan akan muncul di baris pertama LCD :
tombol 1 ditekan
 - ii. Ketika tombol 2 ditekan tulisan akan muncul dibaris kedua LCD : tombol
2 ditekan
- Running Text Shift register LCD :
 - i. Gerakan Running Text dengan arah gerakan text diatur oleh tombol 1 sebagai penentu arah kanan dan tombol 2 sebagai penentu arah kiri
- Serial Shift register LCD :
 - i. Start program dengan memunculkan tulisan “Hallo”, kemudian input text dari serial akan menghapus tulisan sebelumnya dan memunculkan tulisan baru, tulisan tersebut akan bergerak (*running text*) jika tombol 1 ditekan.

d. Catat program yang dibuat pada percobaan c dan perlihatkan pada asisten.

- Jelaskan perubahan dalam program untuk menjalankan prosedur c.
- Jelaskan apa keuntungan menggunakan shift register?

2. Menu

- a. Tuliskan program dibawah ini pada software *Arduino* dan upload keyboard *Arduino Uno R3* :

```
#include <LiquidCrystal_SR_LCD3.h>
const int PIN_LCD_STROBE      = 2;
const int PIN_LCD_DATA        = 3;
const int PIN_LCD_CLOCK       = 4;
int sw_ok=7;
int sw_up=6;
int sw_back=5;
int batas;
LiquidCrystal_SR_LCD3 lcd(PIN_LCD_DATA, PIN_LCD_CLOCK,
PIN_LCD_STROBE);
void setup() {
    lcd.begin(16,2);
    lcd.home ();
    lcd.setCursor (4, 0);           //.....
    lcd.print("ES 12-01");
    delay(3000);
    batas=0;
    menu();
}
void loop()
{
    for(int i=0; i<=batas; i++){
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("BATAS : ");
        lcd.setCursor(7,0);
        lcd.print(batas);
        lcd.setCursor(0,1);
        lcd.print("Data : ");
        lcd.setCursor(7,1);
        lcd.print(i);
        delay(1000);
    }
}
void menu() {
    set:           //.....
    lcd.clear();
    lcd.setCursor(5,0);
    lcd.print("SETTING");
    lcd.setCursor(0,1);
    lcd.print("SET");
    lcd.setCursor(11,1);
    lcd.print("START");
    if(digitalRead(sw_ok)==LOW) {           //.....
        while(digitalRead(sw_ok)==LOW) {} //.....
        goto set1;
    }
    if(digitalRead(sw_back)==LOW) {
        while(digitalRead(sw_back)==LOW) {} //.....
        goto start;
    }
    else{
        goto set;           //.....
    }
}
set1:           //.....
```

```

lcd.clear();
lcd.setCursor(0,0);
lcd.print("BATAS :");
lcd.setCursor(7,0);
lcd.print(batas);
lcd.setCursor(0,1);
lcd.print("SET");
lcd.setCursor(7,1);
lcd.print("UP");
lcd.setCursor(12,1);
lcd.print("DOWN");
if(digitalRead(sw_ok)==LOW) {
    while(digitalRead(sw_ok)==LOW) {} //.....
    goto set; //.....
}
if(digitalRead(sw_back)==LOW) { //.....
    batas=batas-1;
    if (batas<0){
        batas=0;
    }
    delay(100);
    goto set1;
}
if(digitalRead(sw_up)==LOW) {
    batas=batas+1; //.....
    delay(100);
    goto set1;
}
else{
    goto set1; //.....
}
start:
lcd.clear();
lcd.setCursor(8,0);
lcd.print("DONE");
delay(3000);
}

```

- b. Tuliskan komentar pada list program tersebut

6. KASUS PERCOBAAN

- Buat sebuah aplikasi dengan menggunakan 3 LED, 1 Potensio, 3 push button, 1 shift register 4094, 1 buah LCD,
- Terdapat kendali on/off, PWM dan delay dan shift register,
- Terdapat interface analog dan digital
- Catat skematik beserta pin/port yang digunakan, dan program yang dibuat pada kasus diatas dan perlihatkan pada asisten
- Kasusnya : membuat sebuah menu dengan pilihan
 - Blink-blink LED
 - Running LED
 - Fading LED
- Tombol 1, 2 dan 3 berfungsi sebagai pemilih.

7. LAPORAN AKHIR

- Sesuaikan dengan aturan praktikum sebelumnya

- b. Lampirkan hasil percobaan, skematik dan program yang diuji beserta komentar
- c. Jelaskan kasus percobaan 6 dari sisi interface dan peripheral, sistem kendali, dan pemrograman mikrokontroler.
- d. Jelaskan fungsi berikut ini dalam laporan
 - i. while
 - ii. delay
 - iii. goto
 - iv. if (Serial.available())
 - v. lcd.write ()
 - vi. lcd.clear ()
 - vii. lcd.scrollDisplayLeft();
 - viii. lcd.scrollDisplayRight();

MODUL 3

PRAKTIKUM INTERFACE, PEMOGRAMAN MIKROKONTROLER, DAN SISTEM KENDALI, FAKULTAS ILMU TERAPAN, UNIVERSITAS TELKOM

1. JUDUL PRAKTIKUM

Modul Keypad, Seven Segmen, dan Buzzer

2. MAKSUD DAN TUJUAN

Maksud dan tujuan dari praktikum ini adalah :

1. Mahasiswa mampu menggunakan pin-pin pada mikrokontroler dalam mengendalikan modul Keypad, Seven Segmen dan Buzzer
2. Mahasiswa mampu menyelesaikan kasus tertentu dengan menggunakan Keypad, Seven Segmen dan Buzzer dalam mikrokontroler.

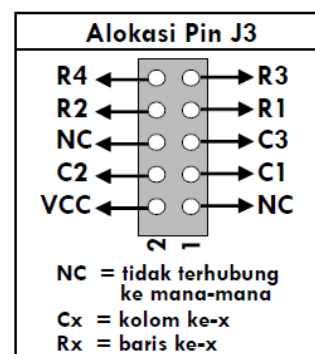
3. PERALATAN DAN BAHAN

Peralatan yang dibutuhkan dalam praktikum ini adalah :

1. 1 buah Arduino Uno R3 + Kabel USB
2. Jumper + header Secukupnya
3. 7 buah Resistor 330 Ohm (optional)
4. 3 buah LED (optional)
5. 1 buah potensio
6. 1 buah Protoboard
7. 1 buah LCD 16x2
8. 1 buah pin header 16x1
9. 1 buah IC Shift register 4094
10. 1 keypad 3x4
11. 1 seven segmen katoda
12. 1 buzzer
13. 1 push button

4. TEORI DASAR

Keypad 3x4

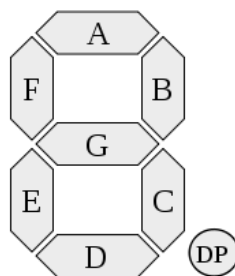


3 x 4 Keypad Module merupakan suatu modul keypad berukuran 3 kolom x 4 baris. Modul ini dapat difungsikan sebagai input dalam aplikasi seperti pengaman digital, datalogger, absensi, pengendali kecepatan motor, robotik, dan sebagainya. Karakteristik dari Keypad :

- Power Supply : 4 V – 18 V.
- Input : Tegangan (Prinsip kerja seperti tombol)
- Output : Tegangan.
- Batasan : Tegangan maksimalnya 18V. Jika menggunakan pada tegangan yang besar dapat merusak komponen lain

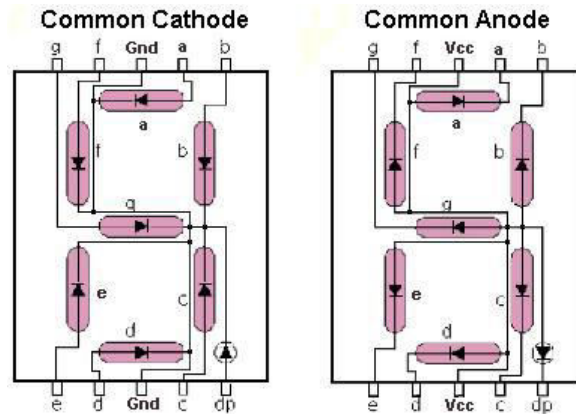
Display Seven Segmen

Display 7 segment merupakan komponen yang berfungsi sebagai penampil karakter angka dan karakter huruf. Display 7 segment sering juga disebut sebagai penampil 7 ruas. Pada display 7 segment juga dilengkapi karakter titik (dot) yang sering dibutuhkan untuk karakter koma atau titik pada saat menampilkan suatu bilangan. Display 7 segment terdiri dari 7 penampil karakter yang disusun dalam sebuah kemasan sehingga dapat menampilkan karakter angka dan karakter huruf. Terdapat 7 buah penampil dasar dari LED (Light Emiting Diode) yang dinamakan karakter A-F dan karakter dot. Bentuk susunan karakter penampil karakter A-F pada display 7 segmen dapat dilihat pada gambar berikut.

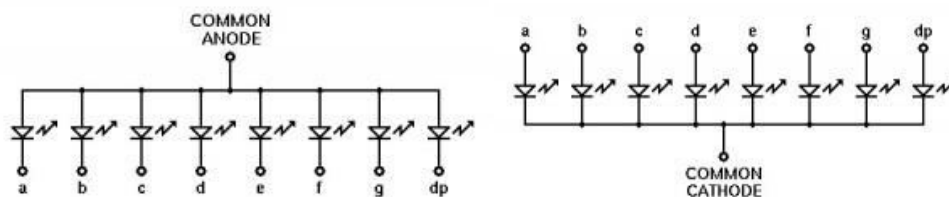


Bentuk Susunan Karakter Display 7 Segment 7 segment, penampil 7 ruas, teori 7 segment, teori penampil 7 ruas, susunan display 7 segment, pengertian display 7 segment, definisi display 7 ruas Pada dasarnya penampil 7 segment merupakan rangkaian 7 buah dioda LED (Light Emiting Diode).

Terdapat 2 (dua) jenis rangkaian dasar dari display 7 segment yang dikenal sebagai display 7 segment common anoda (CA) dan common cathoda (CC). Pada display common anoda untuk mengaktifkan karakter display 7 segment diperlukan logika low (0) pada jalur A-F dan DP dan sebaliknya untuk display 7 segment common cathoda (CA).



Rangkaian internal display 7 segment common anoda dan common cathoda (CC) dapat dilihat pada gambar berikut.



Buzzer

Buzzer adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Pada dasarnya prinsip kerja buzzer hampir sama dengan loud speaker, jadi buzzer juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara. Buzzer biasa digunakan sebagai indikator bahwa proses telah selesai atau terjadi suatu kesalahan pada sebuah alat (alarm).

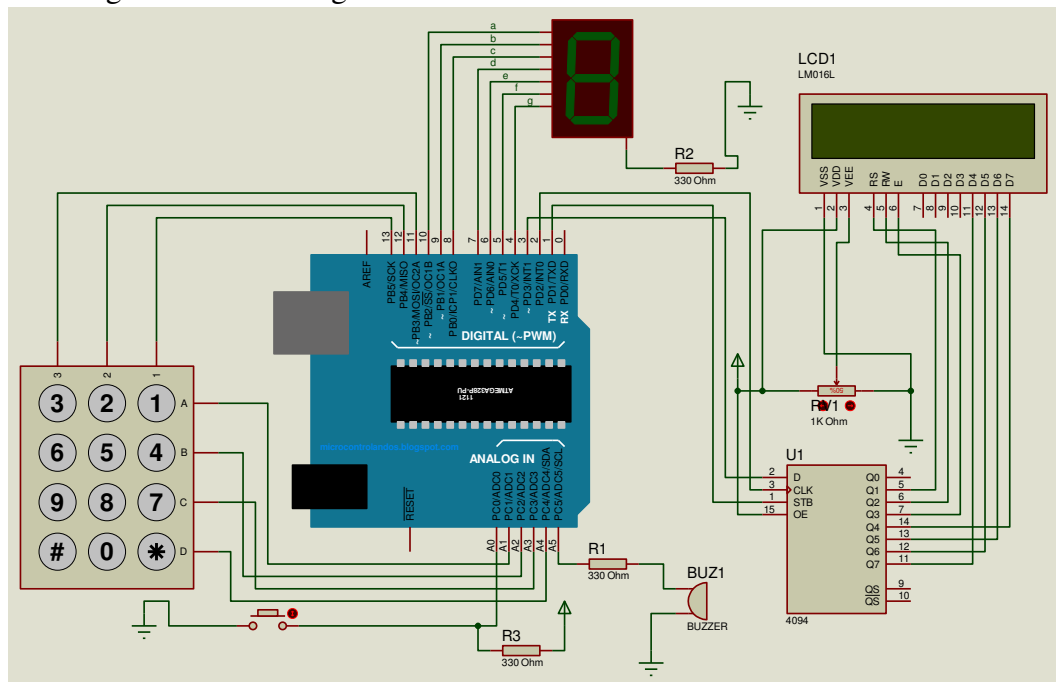


Karakteristik dari Buzzer :

- A. Power Supply : 3-6V DC 25mA.
- B. Input : Tegangan.
- C. Output : Suara 87dB dari PWM.
- D. Batasan : Tegangan Max hanya 6 volt.

5. PROSEDUR PRAKTIKUM

A. Buat rangkaian sesuai dengan skematik berikut :



Catatan : ditambahkan hambatan 330 Ohm pada input LED + pada kaki 15 LCD, dan kaki 16 diground

B. Percobaan dalam praktikum

1. Keypad – Serial

a. Tuliskan program dibawah ini pada software *Arduino* dan upload keyboard

Arduino Uno R3 :

```
int C[]={
  13,12,11};
int R[]={
  A1,A2,A3,A4};
char keymap[4][3]={
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}};
};
char key;

void setup(){
  for (int i=0; i<3; i++){
    pinMode(C[i],INPUT);
  }
  for (int i=0; i<4; i++){
    pinMode(R[i],OUTPUT);
  }
  Serial.begin(9600);
}
void loop(){
  scankeypad();
  Serial.println(key);
}
```

```

    key=' ';
}
void scankeypad(){
    for(int i=0; i<4; i++){
        digitalWrite(R[i],LOW);
        for(int j=0; j<3; j++){
            if(digitalRead(C[j])==LOW){
                key=keymap[i][j];
                delay(500);
            }
        }
        digitalWrite(R[i],HIGH);
    }
}
}

```

- b. Tuliskan komentar pada list program tersebut dengan mencoba menggantikan nilai A, B, C dan D

```

int C[]={
    13,12,11}; // .....
int R[]={
    A1,A2,A3,A4}; // .....
char keymap[4][3]={
    {'1','2','3'}, // .....
    {'4','5','6'},
    {'7','8','9'},
    {'*','0','#'}};
};
char key; // .....

void setup(){
    for (int i=0; i<3; i++){ // .....
        pinMode(C[i],INPUT);
    }
    for (int i=0; i<4; i++){ // .....
        pinMode(R[i],OUTPUT);
    }
    Serial.begin(9600);
}
void loop(){
    scankeypad(); // .....
    Serial.println(key); // .....
    key=' ';
}
void scankeypad(){ // .....
    for(int i=0; i<4; i++){
        digitalWrite(R[i],LOW);
        for(int j=0; j<3; j++){
            if(digitalRead(C[j])==LOW){
                key=keymap[i][j]; // .....
                delay(500);
            }
        }
        digitalWrite(R[i],HIGH); // .....
    }
}
}

```

- c. Buat program untuk memunculkan tulisan di LCD dengan aturan :
 - Tampilkan angka keypad pada LCD dan serial monitor ketika tombol keypad ditekan
- d. Catat program yang dibuat pada percobaan c dan perlihatkan pada asisten.
 - Jelaskan perubahan dalam program untuk menjalankan prosedur c.

2. Seven Segmen

- a. Tuliskan program dibawah ini pada software *Arduino* dan upload keyboard **Arduino Uno R3** :

```

int a=10;
int b=9;
int c=8;
int d=7;
int e=6;
int f=5;
int g=4;
void setup(){
  for(int i=10; i>=4; i--){
    pinMode(i,OUTPUT);
  }
}
void loop(){
  for(int i=0; i<=2; i++){
    segment(i);
    delay(1000);
  }
}

void segment(int angka){
  if (angka==0){
    digitalWrite(a,HIGH);
    digitalWrite(b,HIGH);
    digitalWrite(c,HIGH);
    digitalWrite(d,HIGH);
    digitalWrite(e,HIGH);
    digitalWrite(f,HIGH);
    digitalWrite(g,LOW);
  }
  if (angka==1){
    digitalWrite(a,LOW);
    digitalWrite(b,HIGH);
    digitalWrite(c,HIGH);
    digitalWrite(d,LOW);
    digitalWrite(e,LOW);
    digitalWrite(f,LOW);
    digitalWrite(g,LOW);
  }
  if (angka==2){
    digitalWrite(a,HIGH);
    digitalWrite(b,HIGH);
    digitalWrite(c,LOW);
    digitalWrite(d,HIGH);
    digitalWrite(e,HIGH);
    digitalWrite(f,LOW);
    digitalWrite(g,HIGH);
  }
}

```

b. Tuliskan komentar pada list program pada bagian yang ditentukan

```
int a=10;           // .....
int b=9;           // .....
int c=8;           // .....
int d=7;           // .....
int e=6;           // .....
int f=5;           // .....
int g=4;           // .....
void setup(){      // .....
    for(int i=10; i>=4; i--){
        pinMode(i,OUTPUT);
    }
}
void loop(){        // .....
    for(int i=0; i<=2; i++){
        segment(i);
        delay(1000);
    }
}

void segment(int angka){ // .....
    if (angka==0){      // .....
        digitalWrite(a,HIGH);
        digitalWrite(b,HIGH);
        digitalWrite(c,HIGH);
        digitalWrite(d,HIGH);
        digitalWrite(e,HIGH);
        digitalWrite(f,HIGH);
        digitalWrite(g,LOW);
    }
    if (angka==1){      // .....
        digitalWrite(a,LOW);
        digitalWrite(b,HIGH);
        digitalWrite(c,HIGH);
        digitalWrite(d,LOW);
        digitalWrite(e,LOW);
        digitalWrite(f,LOW);
        digitalWrite(g,LOW);
    }
    if (angka==2){      // .....
        digitalWrite(a,HIGH);
        digitalWrite(b,HIGH);
        digitalWrite(c,LOW);
        digitalWrite(d,HIGH);
        digitalWrite(e,HIGH);
        digitalWrite(f,LOW);
        digitalWrite(g,HIGH);
    }
}
```

c. Buat program berikut dengan aturan :

- Kasus 1 : Lengkapi program diatas hingga angka 9 dan kembali berulang ke angka 0
- Kasus 2 : Ketika angka keypad ditekan, maka angka keypad akan muncul di seven segmen (kecuali * dan #)

d. Catat program yang dibuat pada percobaan c dan perlihatkan pada asisten.

- Jelaskan perubahan dalam program untuk menjalankan prosedur c.

3. Buzzer

- a. Tuliskan program dibawah ini pada software *Arduino* dan upload keboard *Arduino Uno R3* :

```
int buzzer=A0;

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

void loop(){
  digitalWrite(buzzer,HIGH);
  delay(1000);
  digitalWrite(buzzer,LOW);
  delay(1000);
}
```

- b. Tuliskan komentar pada list program pada bagian yang ditentukan

```
int buzzer=A0;          // .....

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

void loop(){
  digitalWrite(buzzer,HIGH); // .....
  delay(1000);
  digitalWrite(buzzer,LOW);  // .....
  delay(1000);
}
```

- c. Buat program dengan aturan :

- Kasus 1 : Seven Segmen hitung mundur dan ketika menyentuh angka 0, buzzer berbunyi.
- Kasus 2 : Push button ditekan, suara buzzer fading meningkat dari paling kecil ke paling tinggi. (rangkaiannya sedikit berubah... ingat fading ~ PWM)

- d. Catat program yang dibuat pada percobaan c dan perlihatkan pada asisten.

- Jelaskan perubahan dalam program untuk menjalankan prosedur c.

6. KASUS PERCOBAAN

- Buat sebuah aplikasi dengan menggunakan 1 Potensio, 3 push button, 1 shift register 4094, 1 buah LCD, Seven Segmen, buzzer
- Terdapat kendali on/off, PWM dan delay dan shift register,
- Terdapat interface analog dan digital
- Catat skematik beserta pin/port yang digunakan, dan program yang dibuat pada kasus diatas dan perlihatkan pada asisten
- Kasusnya : input no handphone menggunakan keypad ditampilkan ke LCD, kemudian button sebagai tombol start, dan akan menampilkan di seven segmen setiap angka satu-satu, dan ketika angka terakhir muncul, maka buzzer berbunyi, end.

7. LAPORAN AKHIR

- a. Sesuaikan dengan aturan praktikum sebelumnya
- b. Lampirkan hasil percobaan, skematik dan program yang diuji beserta komentar
- c. Jelaskan kasus percobaan 6 dari sisi interface dan peripheral, sistem kendali, dan pemrograman mikrokontroler.
- d. Jelaskan cara kerja dan jelaskan juga port mana saja yang bisa digunakan.
 - i. Keypad
 - ii. Seven Segmen
 - iii. Buzzer
- e. Jelaskan fungsi berikut ini dalam laporan
 - i. Function dalam arduino

MODUL 4

PRAKTIKUM INTERFACE, PEMOGRAMAN MIKROKONTROLER, DAN SISTEM KENDALI, FAKULTAS ILMU TERAPAN, UNIVERSITAS TELKOM

1. JUDUL PRAKTIKUM

Modul Dot Matrix dan EEPROM

2. MAKSUD DAN TUJUAN

Maksud dan tujuan dari praktikum ini adalah :

1. Mahasiswa mampu menggunakan pin-pin pada mikrokontroler dalam mengendalikan EEPROM dan dot matrix
2. Mahasiswa mampu menyelesaikan kasus tertentu dengan EEPROM dan dot matrix dalam mikrokontroler.

3. PERALATAN DAN BAHAN

Peralatan yang dibutuhkan dalam praktikum ini adalah :

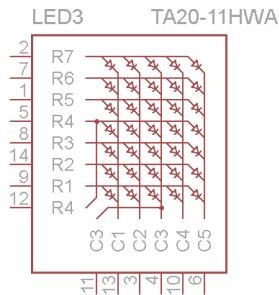
1. 1 buah Arduino Uno R3 + Kabel USB
2. Jumper + header Secukupnya
3. 7 buah Resistor 330 Ohm (optional)
4. 3 buah LED (optional)
5. 1 buah potensio
6. 1 buah Protoboard
7. 1 buah LCD 16x2
8. 1 buah pin header 16x1
9. 1 buah IC Shift register 4094
10. 1 keypad 3x4
11. 1 seven segmen katoda
12. 1 buzzer
13. 1 push button
14. 1 dot matrix

4. TEORI DASAR

EEPROM

EPROM kependekan dari *Erasable Programmable Read Only Memory*. EPROM berbeda dengan PROM. EPROM adalah jenis chip memori yang dapat ditulisi program secara elektris. Program atau informasi yang tersimpan di dalam EPROM dapat dihapus bila terkena sinar ultraviolet dan dapat ditulisi kembali. Kesamaannya dengan PROM adalah keduanya merupakan jenis ROM, termasuk memori non-volatile, data yang tersimpan di dalamnya tidak bisa hilang walaupun komputer dimatikan, tidak membutuhkan daya listrik untuk mempertahankan atau menjaga informasi atau program yang tersimpan di dalamnya.

DOT Matri 7x5



| | | | | | | | |
|----|----|----|----|----|----|----|----|
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 7F | 88 | 88 | 88 | 7F | 00 | 00 | 00 |

Pada dasarnya DOT matrix adalah Display LED yg disusun sedemikian rupa sehingga untuk menghidupkan led ke (x,x) dibutuhkan kombinasi tegangan antara Pin baris dan kolom. Kemudian untuk mempermudah kontrol dan menghemat pin maka diperlukan proses scanning (biasanya kolom) sedangkan pada baris diberikan bit sesuai huruf/karakter yang akan ditampilkan yang bersesuaian dengan posisi scanning.

Scanning untuk kolom dimaksud memberikan "1" high (untuk common katoda) atau "0" untuk common anoda, untuk kolom lainnya diberi nilai negasi dari kolom yang diberi nilai 1 tadi dan seterusnya untuk kolom berikutnya sampai kolom terakhir dari rangkaian led matrix, kemudian berulang dari depan lagi. Proses ini dilakukan sangat cepat sehingga mata kita melihatnya tetap sebagai suatu karakter yang diam. Misalnya ingin menampilkan huruf A, secara umum digambarkan sebagai berikut:

```

0000000
0000000
0000000
0000000
0000000
0000000
0000000

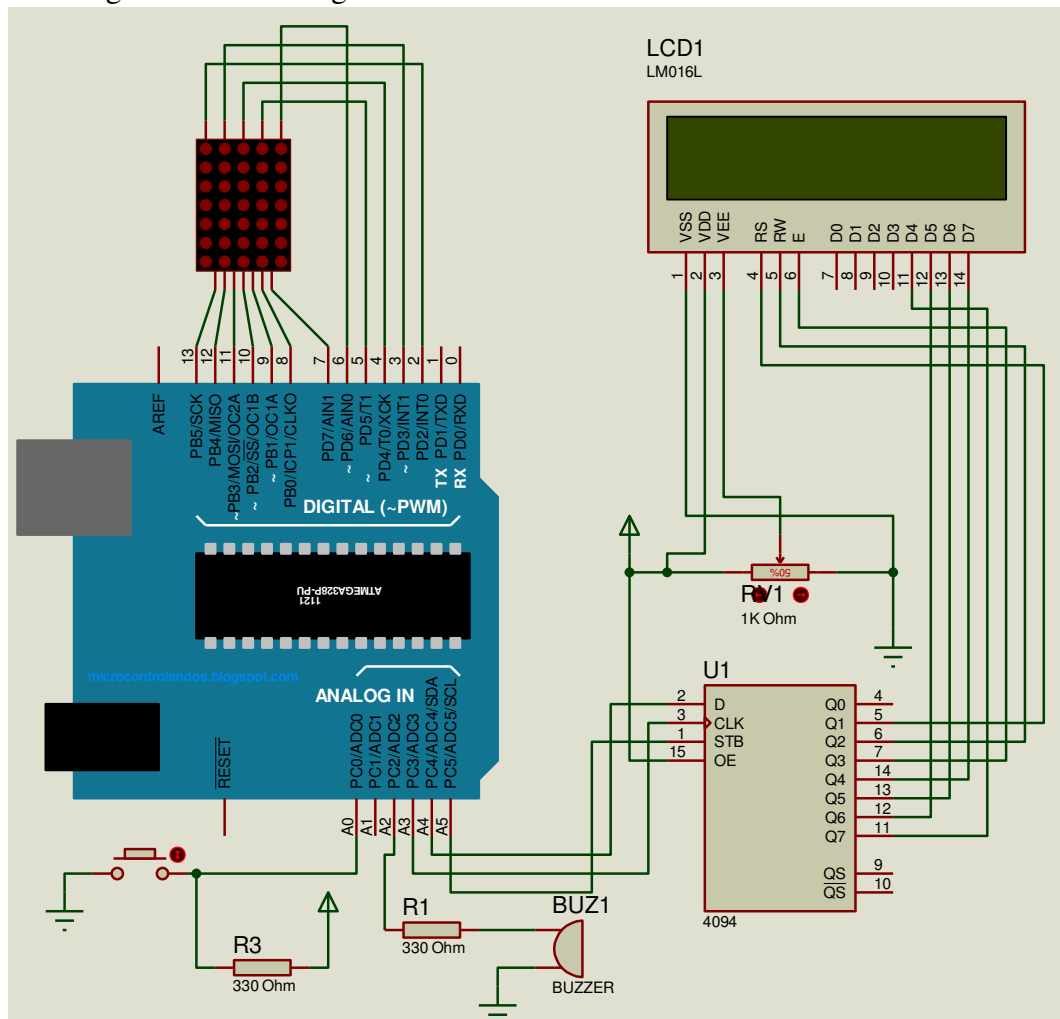
```

dari gambar diatas logikanya seperti ini :

- Saat kolom pertama (scan kolom 1 = 1 / high/2.4 v) maka bit yg diberi pada baris berupa "1110000", sehingga jika ketemu 1 vs 1 ga ada arus mengalir, jadi LED mati, jika 1 ketemu 0 maka arus mengalir, led akan menyala
- Untuk kolom kedua, ketika kolom kedua diberi tegangan maka pada baris bit yg diberikan adalah "1101011"
- Dan seterusnya, ketika kolom nya mencapai ujung maka akan diulang ke kolom 1 lagi

5. PROSEDUR PRAKTIKUM

A. Buat rangkaian sesuai dengan skematik berikut :



Catatan : ditambahkan hambatan 330 Ohm pada input LED + pada kaki 15 LCD, dan kaki 16 diground (jika menggunakan 8x8 dot matri)

B. Percobaan dalam praktikum

1. EEPROM write

a. Tuliskan program dibawah ini pada software *Arduino* dan upload keboard

Arduino Uno R3 :

```
#include <EEPROM.h>
int addr = 0;
int potensio = A0;
void setup()
{
  Serial.begin(9600);
}
```

```
void loop()
{
  int val = analogRead(potensio) / 4;
  if (addr <= 512) {
    EEPROM.write(addr, val);
    Serial.print(val);
  }
```

```

        Serial.print("\t");
        Serial.println(addr);

        addr = addr + 1;
    }
    delay(100);
}

```

b. Tuliskan komentar pada list program pada bagian yang ditentukan

```

#include <EEPROM.h> //.....
int addr = 0; //.....
int potensio = A0; //.....
void setup()
{
    Serial.begin(9600); //.....
}

void loop()
{
    int val = analogRead(potensio) / 4; //.....
    if (addr <= 512) { //.....
        EEPROM.write(addr, val); //.....
        Serial.println(addr); //.....
        addr = addr + 1; //.....
    }
    delay(100);
}

```

2. EEPROM read

a. Tuliskan program dibawah ini pada software *Arduino* dan upload keyboard **Arduino Uno R3** :

```

#include <EEPROM.h>
int address = 0;
byte value;

void setup()
{
    Serial.begin(9600);
}

void loop()
{
    value = EEPROM.read(address); //.....
    Serial.print(address); //.....
    Serial.print("\t"); //.....
    Serial.print(value, DEC); //.....
    Serial.println(); //.....
    address = address + 1; //.....
    if (address == 512) //.....
        address = 0; //.....
    delay(500);
}

```

3. EEPROM clear

- a. Tuliskan program dibawah ini pada software *Arduino* dan upload keyboard

Arduino Uno R3 :

```
#include <EEPROM.h>

void setup()
{
  for (int i = 0; i < 512; i++)
    EEPROM.write(i, 0);
  digitalWrite(13, HIGH);
}

void loop()
{
}
```

- b. Tuliskan komentar pada list program pada bagian yang ditentukan

```
#include <EEPROM.h>

void setup()
{
  for (int i = 0; i < 512; i++)      //.....
    EEPROM.write(i, 0);           //.....
  digitalWrite(13, HIGH);         //.....
}

void loop()
{
}
```

4. Kasus EEPROM

- a. Buat aplikasi dengan aturan berikut : terdapat 1 push button, 1 shift register, 1 LCD, 1 potensiometer, dan 1 dot matrix 7x5

- Push button ditekan pertama kali untuk merekam data dengan merubah potensio(data rekaman ditunjukkan pada serial monitor dan LCD).
- Push button ditekan kedua kali untuk menunjukkan hasil rekaman dalam bentuk keluaran serial monitor dan LED di pin PWM (silahkan pilih)
- Push button ditekan ketiga kali untuk menghapus semua rekaman dalam EEPROM diakhiri dengan bunyi buzzer pada akhir penghapusan
- Push button ditekan keempat kali untuk menunjukkan data rekaman EEPROM di serial monitor telah terhapus

5. Dot Matrix 7x5

- a. Tuliskan program dibawah ini pada software *Arduino* dan upload keyboard **Arduino Uno R3 :**

```
#include <FrequencyTimer2.h>

#define SPACE { \
{0, 0, 0, 0, 0}, \
{0, 0, 0, 0, 0}, \
{0, 0, 0, 0, 0}, \
{0, 0, 0, 0, 0}, \
{0, 0, 0, 0, 0}, \
{0, 0, 0, 0, 0}, \
{0, 0, 0, 0, 0} \
}
```

```

#define H { \
{1, 0, 0, 0, 1}, \
{1, 0, 0, 0, 1}, \
{1, 0, 0, 0, 1}, \
{1, 1, 1, 1, 1}, \
{1, 0, 0, 0, 1}, \
{1, 0, 0, 0, 1}, \
{1, 0, 0, 0, 1} \
}

#define E { \
{1, 1, 1, 1, 1}, \
{1, 0, 0, 0, 0}, \
{1, 0, 0, 0, 0}, \
{1, 1, 1, 1, 0}, \
{1, 0, 0, 0, 0}, \
{1, 0, 0, 0, 0}, \
{1, 1, 1, 1, 1} \
}

#define small_E { \
{0, 0, 0, 0, 0}, \
{0, 0, 0, 0, 0}, \
{0, 1, 1, 1, 0}, \
{1, 0, 0, 0, 1}, \
{1, 1, 1, 1, 0}, \
{1, 0, 0, 0, 0}, \
{0, 1, 1, 1, 0} \
}

#define L { \
{1, 0, 0, 0, 0}, \
{1, 0, 0, 0, 0}, \
{1, 0, 0, 0, 0}, \
{1, 0, 0, 0, 0}, \
{1, 0, 0, 0, 0}, \
{1, 0, 0, 0, 0}, \
{1, 1, 1, 1, 1} \
}

#define small_L { \
{0, 1, 1, 0, 0}, \
{0, 0, 1, 0, 0}, \
{0, 0, 1, 0, 0}, \
{0, 0, 1, 0, 0}, \
{0, 0, 1, 0, 0}, \
{0, 0, 1, 0, 0}, \
{0, 1, 1, 1, 0} \
}

#define O { \
{0, 1, 1, 1, 0}, \
{1, 0, 0, 0, 1}, \
{1, 0, 0, 0, 1}, \
{1, 0, 0, 0, 1}, \
{1, 0, 0, 0, 1}, \
{1, 0, 0, 0, 1}, \
}

```

```

{0, 1, 1, 1, 0} \
}

#define small_O { \
{0, 0, 0, 0, 0}, \
{0, 0, 0, 0, 0}, \
{0, 1, 1, 1, 0}, \
{1, 0, 0, 0, 1}, \
{1, 0, 0, 0, 1}, \
{1, 0, 0, 0, 1}, \
{0, 1, 1, 1, 0} \
}

#define small_W { \
{0, 0, 0, 0, 0}, \
{0, 0, 0, 0, 0}, \
{1, 0, 0, 0, 1}, \
{1, 0, 0, 0, 1}, \
{1, 0, 1, 0, 1}, \
{1, 0, 1, 0, 1}, \
{0, 1, 0, 1, 0} \
}

#define small_R { \
{0, 0, 0, 0, 0}, \
{0, 0, 0, 0, 0}, \
{0, 1, 0, 1, 1}, \
{0, 1, 1, 0, 0}, \
{0, 1, 0, 0, 0}, \
{0, 1, 0, 0, 0}, \
{0, 1, 0, 0, 0} \
}

#define small_D { \
{0, 0, 0, 0, 1}, \
{0, 0, 0, 0, 1}, \
{0, 1, 1, 0, 1}, \
{1, 0, 0, 1, 1}, \
{1, 0, 0, 0, 1}, \
{1, 0, 0, 0, 1}, \
{0, 1, 1, 1, 1} \
}

byte col = 0;
byte leds[5][7];
int pins[13]= {-1, 2, 3, 4, 5, 6, 13, 12, 11, 10, 9, 8, 7};

int cols[5] = {pins[1], pins[2], pins[3], pins[4], pins[5]};

int rows[7] = {pins[6], pins[7], pins[8], pins[9],
pins[10], pins[11], pins[12]};

const int numPatterns = 12;
byte patterns[numPatterns][7][5] = {
SPACE, H, small_E, small_L, small_L, small_O,
SPACE, small_W, small_O, small_R, small_L, small_D
};

```



```

int pattern = 0;

void setup()
{
    for (int i = 1; i <= 12; i++) {
        pinMode(pins[i], OUTPUT);
    }

    for (int i = 1; i <= 5; i++) {
        digitalWrite(cols[i - 1], LOW);
    }

    for (int i = 1; i <= 7; i++) {
        digitalWrite(rows[i - 1], LOW);
    }

    clearLeds();

    FrequencyTimer2::disable();
    FrequencyTimer2::setPeriod(2000);
    FrequencyTimer2::setOnOverflow(display);
    setPattern(pattern);
}

void loop()
{
    pattern = ++pattern % numPatterns;
    slidePattern(pattern, 100);
}

void clearLeds()
{
    for (int i = 0; i < 5; i++) {
        for (int j = 0; j < 7; j++) {
            leds[i][j] = 0;
        }
    }
}

void setPattern(int pattern)
{
    for (int i = 0; i < 5; i++) {
        for (int j = 0; j < 7; j++) {
            leds[i][j] = patterns[pattern][j][i];
        }
    }
}

void slidePattern(int pattern, int del)
{
    for (int newcol = 0; newcol <= 4; newcol++) {

        // shift the first 4 columns left
        for (int row = 0; row <= 6; row++)

```

```

        for (int col = 0; col <= 3; col++)
            leds[col][row] = leds[col+1][row];

    for (int row = 0; row <= 6; row++)
        leds[4][row] = patterns[pattern][row][newcol];

    delay(del);
}
}

void display()
{
    digitalWrite(cols[col], LOW);
    col++;
    if (col == 5) {
        col = 0;
    }
    for (int row = 0; row < 7; row++) {
        if (leds[col][row] == 1) {
            digitalWrite(rows[row], LOW);
        }
        else {
            digitalWrite(rows[row], HIGH);
        }
    }
    digitalWrite(cols[col], HIGH);
}
}

```

- b. Lakukan modifikasi program diatas kemudian artikan menurut kamu bagaimana cara kerja koding diatas.

6. KASUS PERCOBAAN

- a. Buat sebuah aplikasi dengan menggunakan 1 Potensio, 1 push button, 1 shift register 4094, 1 buah LCD, buzzer dan dot matrix.
- b. Terdapat kendali on/off, PWM dan delay dan shift register,
- c. Terdapat interface analog dan digital
- d. Catat skematik beserta pin/port yang digunakan, dan program yang dibuat pada kasus diatas dan perlihatkan pada asisten
- e. Kasusnya : jika tombol start ditekan maka berfungsi untuk merekam inputan satu huruf dari serial monitor, kemudian setelah input dari serial monitor dilakukan, kemudian penekanan pada tombol yang sama berfungsi menjalankan running text pada dot matrix dari inputan serial tersebut.

7. LAPORAN AKHIR

- a. Sesuaikan dengan aturan praktikum sebelumnya
- b. Lampirkan hasil percobaan, skematik dan program yang diuji beserta komentar
- c. Jelaskan kasus percobaan 6 dari sisi interface dan peripheral, sistem kendali, dan pemrograman mikrokontroler.

- d. Jelaskan cara kerja dan jelaskan juga port mana saja yang bisa digunakan.
 - i. Dot matrix
 - ii. EEPROM

MODUL 5

PRAKTIKUM INTERFACE, PEMOGRAMAN MIKROKONTROLER, DAN SISTEM KENDALI, FAKULTAS ILMU TERAPAN, UNIVERSITAS TELKOM

1. JUDUL PRAKTIKUM

Photodiode, LDR dan Relay

2. MAKSUD DAN TUJUAN

Maksud dan tujuan dari praktikum ini adalah :

1. Mahasiswa mampu menggunakan pin-pin pada mikrokontroler dalam mengendalikan Photodiode, Relay dan LDR
2. Mahasiswa mampu menyelesaikan kasus tertentu dengan Photodiode, Relay dan LDR dalam mikrokontroler.

3. PERALATAN DAN BAHAN

Peralatan yang dibutuhkan dalam praktikum ini adalah :

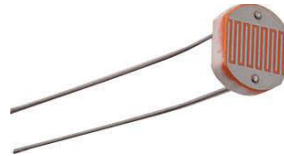
1. 1 buah Arduino Uno R3 + Kabel USB
2. Jumper + header Secukupnya
3. 7 buah Resistor 330 Ohm (optional)
4. 3 buah LED (optional)
5. 1 buah potensio
6. 1 buah Protoboard
7. 1 buah LCD 16x2
8. 1 buah pin header 16x1
9. 1 buah IC Shift register 4094
10. 1 keypad 3x4
11. 1 seven segmen katoda
12. 1 buzzer
13. 1 push button
14. 1 dot matrix
15. 1 Photodiode
16. 1 LDR
17. 1 Relay
18. 2 Resistor 33 K

4. TEORI DASAR PHOTODIODE



Photodiode merupakan sensor cahaya. Dimana resistansi yang dimiliki sensor ini akan berubah-ubah sesuai intensitas cahaya yang mengenai sensor ini.

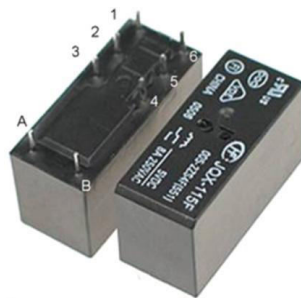
- A. Power supply : 5 – 12 Volt
- B. Input : Intensitas Cahaya (Analog)
- C. Output : Tegangan (Analog)
- D. Batasan : Dapat bekerja sampai tegangan 12 Volt. Dengan menggunakan resistor yang besar akan membuat sensor ini lebih efisien.



LDR

LDR (*Light Dependent Resistor*) dapat juga digunakan sebagai sensor cahaya. LDR merupakan resistor yang nilai resistansinya akan berubah-ubah sesuai intensitas cahaya yang mengenai sensor ini.

- A. Power supply : 5-320 volt
- B. Input : Intensitas Cahaya (Analog)
- C. Output : Tegangan (Analog)
- D. Batasan : Tegangan maksimal 320 Volt

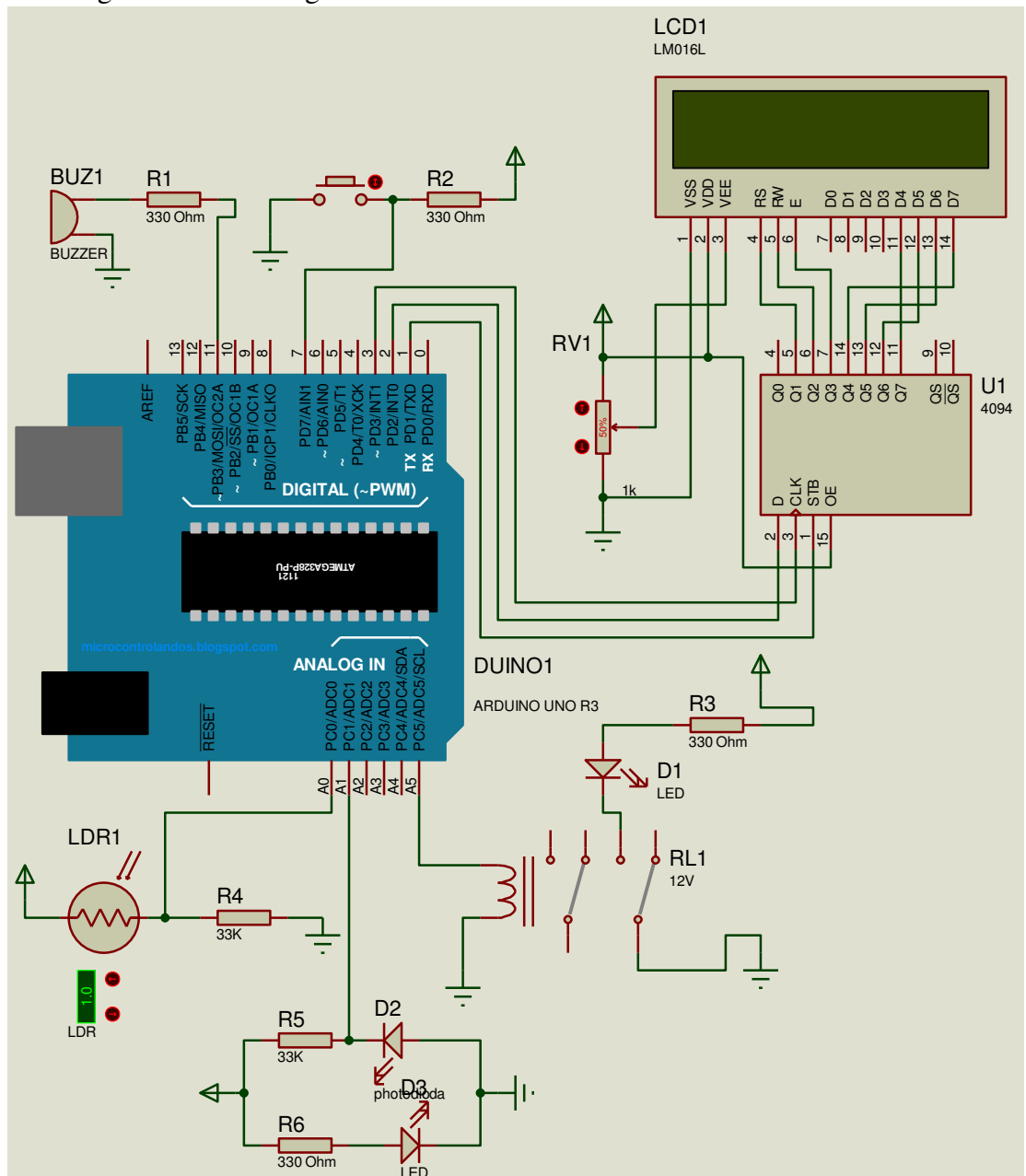


RELAY

- A. Power Supply : 5-12 Volt
- B. Input : Tegangan
- C. Output : Arus (Sebagai sakelar)
- D. Batasan : Tegangan minimum untuk trigger = 5Volt

5. PROSEDUR PRAKTIKUM

A. Buat rangkaian sesuai dengan skematik berikut :



Catatan : ditambahkan hambatan 330 Ohm pada input LED + pada kaki 15 LCD, dan kaki 16 diground

B. Percobaan dalam praktikum

1. LDR - Serial

a. Tuliskan program dibawah ini pada software *Arduino* dan upload keboard *Arduino Uno R3* :

```
int lightPin = A0;
int ledPin = 13;

void setup()
{
  Serial.begin(9600);
```

```

        pinMode( ledPin, OUTPUT );
    }

    void loop()
    {
        Serial.println(analogRead(lightPin));
        analogWrite(ledPin, analogRead(lightPin)/4);
        delay(10);
    }

```

- b. Lakukan modifikasi pada rangkaian diatas dan berikan komentar pada setiap line program diatas.

2. Photodiode - Buzzer

- a. Tuliskan program dibawah ini pada software *Arduino* dan upload keboard *Arduino Uno R3* :

```

int buzz = 10;
int senRead = 0;
int limit = 850;
void setup()
{
    pinMode(pd,OUTPUT);
    pinMode(buzz,OUTPUT);
    digitalWrite(buzz,LOW);
    Serial.begin(9600);
}
void loop()
{
    int val=analogRead(senRead);
    Serial.println(val);
    if(val <= limit)
    {
        digitalWrite(buzz,HIGH);
        delay(20);
    }
    else if(val > limit)
    {
        digitalWrite(buzz,LOW);
        delay(20);
    }
}

```

- b. Lakukan modifikasi pada rangkaian diatas dan berikan komentar pada setiap line program diatas.

3. Relay

- a. Tuliskan program dibawah ini pada software *Arduino* dan upload keboard *Arduino Uno R3* :

```

int relay = 13;
void setup() {
    pinMode(relay, OUTPUT);
}

void loop() {
    digitalWrite(relay, HIGH);
    delay(1000);
}

```

```
digitalWrite(relay, LOW);  
delay(1000);  
}
```

- b. Lakukan modifikasi pada rangkaian diatas dan berikan komentar pada setiap line program diatas.

6. KASUS PERCOBAAN

- a. Buat sebuah aplikasi dengan menggunakan sensor photodiode, LDR, push button, LCD, Relay, dan LED
- b. Terdapat kendali on/off, PWM dan delay dan shift register,
- c. Terdapat interface analog dan digital
- d. Catat skematik beserta pin/port yang digunakan, dan program yang dibuat pada kasus diatas dan perlihatkan pada asisten
- e. Kasusnya :
 - i. Fungsi push button 1 untuk menunjukkan nilai LDR dan ditunjukkan dalam LED (PWM, rangkaian ditambah LED 1 + 330 Ohm)
 - ii. Fungsi push button 2 untuk menunjukkan nilai Photodiode dan ditunjukkan dalam LED (PWM, rangkaian ditambah LED 1 + 330 Ohm)
 - iii. Fungsi push button 3 untuk menunjukkan jika photodiode memiliki batas nilai tertentu agar dapat membunyikan buzzer
 - iv. Fungsi push button 4 untuk menunjukkan jika LDR memiliki batas nilai tertentu agar dapat membunyikan buzzer

7. LAPORAN AKHIR

- a. Sesuaikan dengan aturan praktikum sebelumnya
- b. Lampirkan hasil percobaan, skematik dan program yang diuji beserta komentar
- c. Jelaskan apa saja yang dimaksud dalam jenis-jenis interface.
- d. Jelaskan kasus percobaan 6 dari sisi interface dan pheripheral, sistem kendali, dan pemograman mikrokontroler.
- e. Jelaskan cara kerja dan jelaskan juga port mana saja yang bisa digunakan.
 - i. Photodiode
 - ii. LDR
 - iii. Relay

MODUL 6

PRAKTIKUM INTERFACE, PEMOGRAMAN MIKROKONTROLER, DAN SISTEM KENDALI, FAKULTAS ILMU TERAPAN, UNIVERSITAS TELKOM

1. JUDUL PRAKTIKUM

LM35 dan 4N35

2. MAKSUD DAN TUJUAN

Maksud dan tujuan dari praktikum ini adalah :

1. Mahasiswa mampu menggunakan pin-pin pada mikrokontroler dalam mengendalikan LM35 dan 4N35
2. Mahasiswa mampu menyelesaikan kasus tertentu dengan LM35 dan 4N35 dalam mikrokontroler.

3. PERALATAN DAN BAHAN

Peralatan yang dibutuhkan dalam praktikum ini adalah :

1. 1 buah Arduino Uno R3 + Kabel USB
2. Jumper + header Secukupnya
3. 7 buah Resistor 330 Ohm (optional)
4. 3 buah LED (optional)
5. 1 buah potensio
6. 1 buah Protoboard
7. 1 buah LCD 16x2
8. 1 buah pin header 16x1
9. 1 buah IC Shift register 4094
10. 1 keypad 3x4
11. 1 seven segmen katoda
12. 1 buzzer
13. 1 push button
14. 1 dot matrix
15. 2 Resistor 33 K
16. 1 LM35
17. 1 4N35

4. TEORI DASAR

A. LM35



- a. Power Supply : -0,2 – 5 volt.
- b. Input : Tegangan.
- c. Output : Tegangan.
- d. Tegangan Max : 5 V.

```
temp = (5.0 * analogRead(tempPin) * 100.0) / 1024;
```

Namun, hal ini tidak menghasilkan resolusi tinggi. Hal ini dapat dengan mudah dihindari, LM35 hanya menghasilkan tegangan 0-1 V. ADC menggunakan 5V sebagai nilai tertinggi. Ini membuang-buang 80% dari kisaran mungkin. Jika mengubah Aref ke 1.1V, sehingga mendapatkan resolusi tertinggi.

Persamaan asli berasal dari mengambil pembacaan, menemukan apa yang persentase kisaran (1024) itu, mengalikan bahwa dengan kisaran sendiri (Aref, atau 5000 mV), dan membaginya dengan sepuluh (10 mV per derajat Celcius, sesuai datasheet : <http://www.ti.com/lit/ds/symlink/lm35.pdf>)

Namun, jika Anda menggunakan 1.1V sebagai Aref, persamaan akan berubah. Jika anda membagi 1.1V lebih 1024, masing-masing langkah dalam pembacaan analog adalah sama dengan sekitar $0.001074V = 1,0742 \text{ mV}$. Jika 10mV sama dengan 1 derajat Celcius, $10 / 1,0742 = \sim 9.31$. Jadi, untuk setiap perubahan dari 9.31 dalam pembacaan analog, ada satu derajat perubahan suhu.

Untuk mengubah Aref ke 1.1V, gunakan perintah "analogReference (INTERNAL) ;"

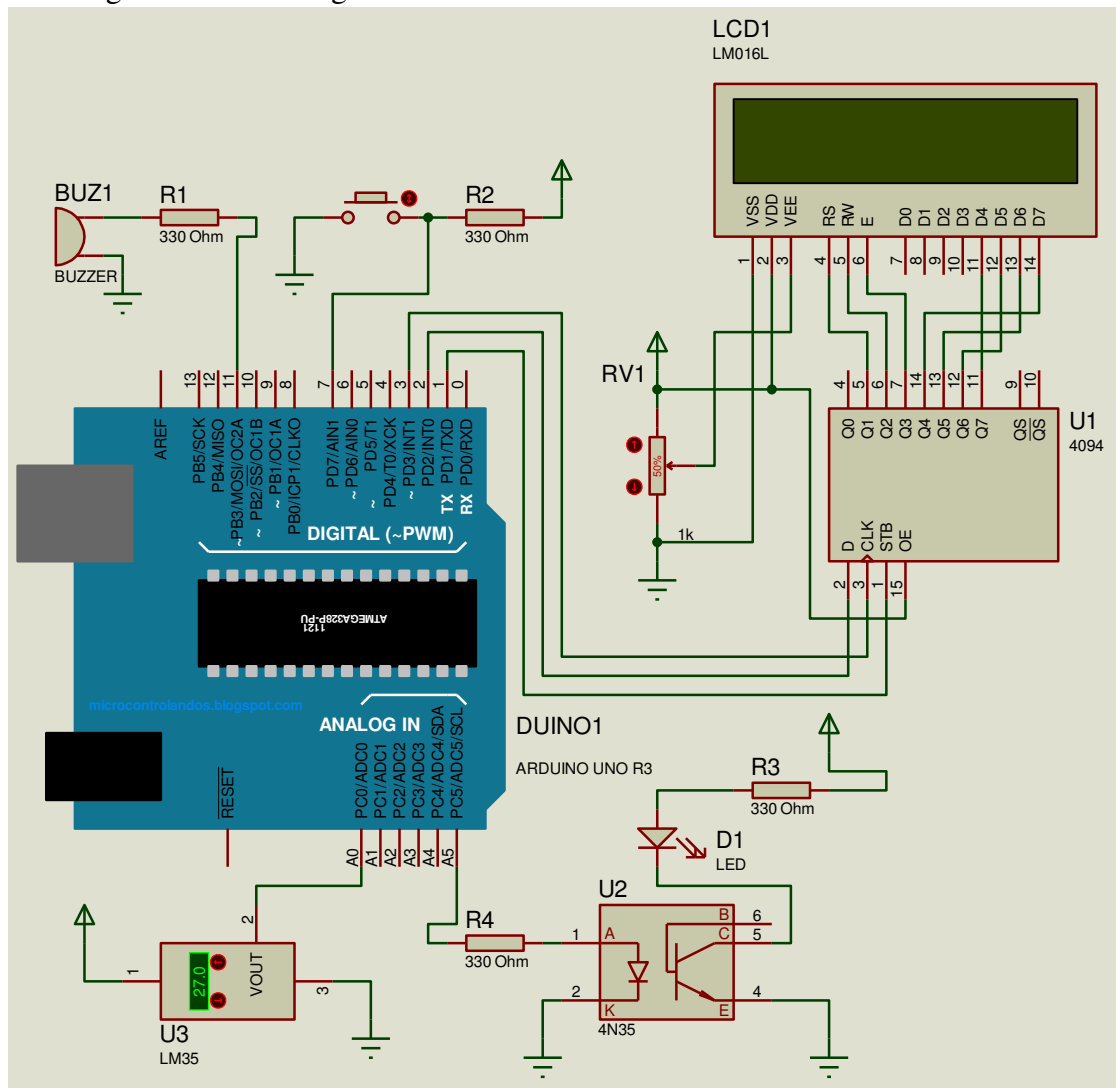
B. 4N35



- a. Power Supply : 400W 35VDC.
- b. Input : Analog dan tegangan.
- c. Output : Tegangan.
- d. Tegangan bernilai 1,5 volt dan nilai inputnya mempunyai tegangan yang tinggi.

5. PROSEDUR PRAKTIKUM

A. Buat rangkaian sesuai dengan skematik berikut :



Catatan : ditambahkan hambatan 330 Ohm pada input LED + pada kaki 15 LCD, dan kaki 16 diground

B. Percobaan dalam praktikum

1. LM35 - Serial

a. Tuliskan program dibawah ini pada software *Arduino* dan upload keboard *Arduino Uno R3* :

```
float tempC;
int reading;
int tempPin = 0;

void setup()
{
  analogReference (INTERNAL);
}

void loop()
{
  reading = analogRead(tempPin);
```

```
tempC = reading / 9.31;  
}
```

- b. Lakukan modifikasi pada rangkaian diatas dan berikan komentar pada setiap line program diatas.

2. 4N35

- a. Tuliskan program dibawah ini pada software *Arduino* dan upload keboard

Arduino Uno R3 :

```
int led = A5;  
  
void setup() {  
  pinMode(led, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(led, HIGH);  
  delay(1000);  
  digitalWrite(led, LOW);  
  delay(1000);  
}
```

- b. Lakukan modifikasi pada rangkaian diatas dan berikan komentar pada setiap line program diatas.

6. KASUS PERCOBAAN

- a. Buat sebuah aplikasi dengan menggunakan 1 push button, 1 LM35, 4N35, LED (posisi port dapat dirubah sesuai dengan kebutuhan)
- b. Terdapat kendali on/off, PWM dan delay dan shift register,
- c. Terdapat interface analog dan digital
- d. Catat skematik beserta pin/port yang digunakan, dan program yang dibuat pada kasus diatas dan perlihatkan pada asisten
- e. Kasusnya :
 - i. Push button ditekan pertama kali untuk merekam data suhu yang dihasilkan LM35 dan ditunjukkan dalam LCD dan serial monitor.
 - ii. Push button ditekan kedua kali untuk menunjukkan hasil rekaman dalam bentuk keluaran serial monitor dan LCD.
 - iii. Push button ditekan ketiga kali untuk menghapus semua rekaman dalam EEPROM diakhiri dengan bunyi buzzer pada akhir penghapusan
 - iv. Push button ditekan keempat kali untuk melaksanakan LED fading dan variasi (bisa kelap-kelip)
 - v. Push button ditekan kelima kali kembali ke kasus 1

7. LAPORAN AKHIR

- a. Sesuaikan dengan aturan praktikum sebelumnya
- b. Lampirkan hasil percobaan, skematik dan program yang diuji beserta komentar
- c. Jelaskan apa saja yang dimaksud dalam jenis-jenis sistem kendali.
- d. Jelaskan kasus percobaan 6 dari sisi interface dan peripheral, sistem kendali, dan pemograman mikrokontroler.
- e. Jelaskan cara kerja dan jelaskan juga port mana saja yang bisa digunakan.
 - i. LM35

ii. 4N35

MODUL 7

PRAKTIKUM INTERFACE, PEMOGRAMAN MIKROKONTROLER, DAN SISTEM KENDALI, FAKULTAS ILMU TERAPAN, UNIVERSITAS TELKOM

1. JUDUL PRAKTIKUM

PIR (Passive Infra Red)

2. MAKSUD DAN TUJUAN

Maksud dan tujuan dari praktikum ini adalah :

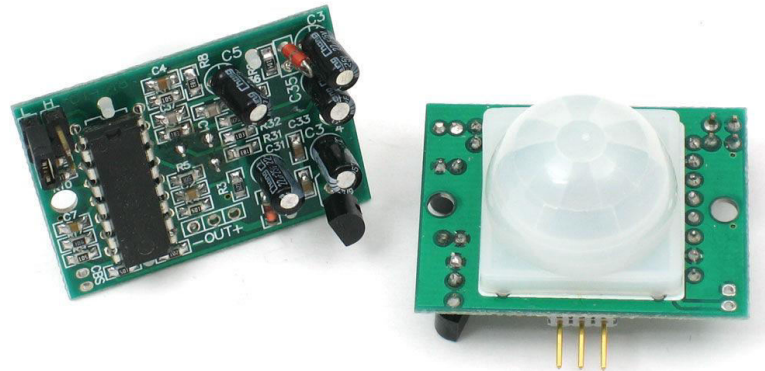
1. Mahasiswa mampu menggunakan pin-pin pada mikrokontroler dalam mengendalikan PIR
2. Mahasiswa mampu menyelesaikan kasus tertentu dengan PIR dalam mikrokontroler.

3. PERALATAN DAN BAHAN

Peralatan yang dibutuhkan dalam praktikum ini adalah :

1. 1 buah Arduino Uno R3 + Kabel USB
2. Jumper + header Secukupnya
3. 7 buah Resistor 330 Ohm (optional)
4. 3 buah LED (optional)
5. 1 buah potensio
6. 1 buah Protoboard
7. 1 buah LCD 16x2
8. 1 buah pin header 16x1
9. 1 buah IC Shift register 4094
10. 1 keypad 3x4
11. 1 seven segmen katoda
12. 1 buzzer
13. 1 push button
14. 1 dot matrix
15. 2 Resistor 33 K
16. 1 4N35
17. PIR

4. TEORI DASAR PASSIVE INFRA RED (PIR)



- a. Power Supply : 5V.
- b. Input : Perubahan Gerakan (thermal).
- c. Output : Digital.
- d. Tegangan Max hanya 5 V.

PIR (*Passive Infrared Receiver*) merupakan sebuah sensor berbasis infrared. Akan tetapi, tidak seperti sensor infrared kebanyakan yang terdiri dari IR LED dan fototransistor. PIR tidak memancarkan apapun seperti IR LED. Sesuai dengan namanya '*Passive*', sensor ini hanya merespon energi dari pancaran sinar inframerah pasif yang dimiliki oleh setiap benda yang terdeteksi olehnya. Benda yang bisa dideteksi oleh sensor ini biasanya adalah tubuh manusia.

Di dalam sensor PIR ini terdapat bagian-bagian yang mempunyai perannya masing-masing, yaitu Fresnel Lens, IR Filter, Pyroelectric sensor, amplifier, dan comparator.

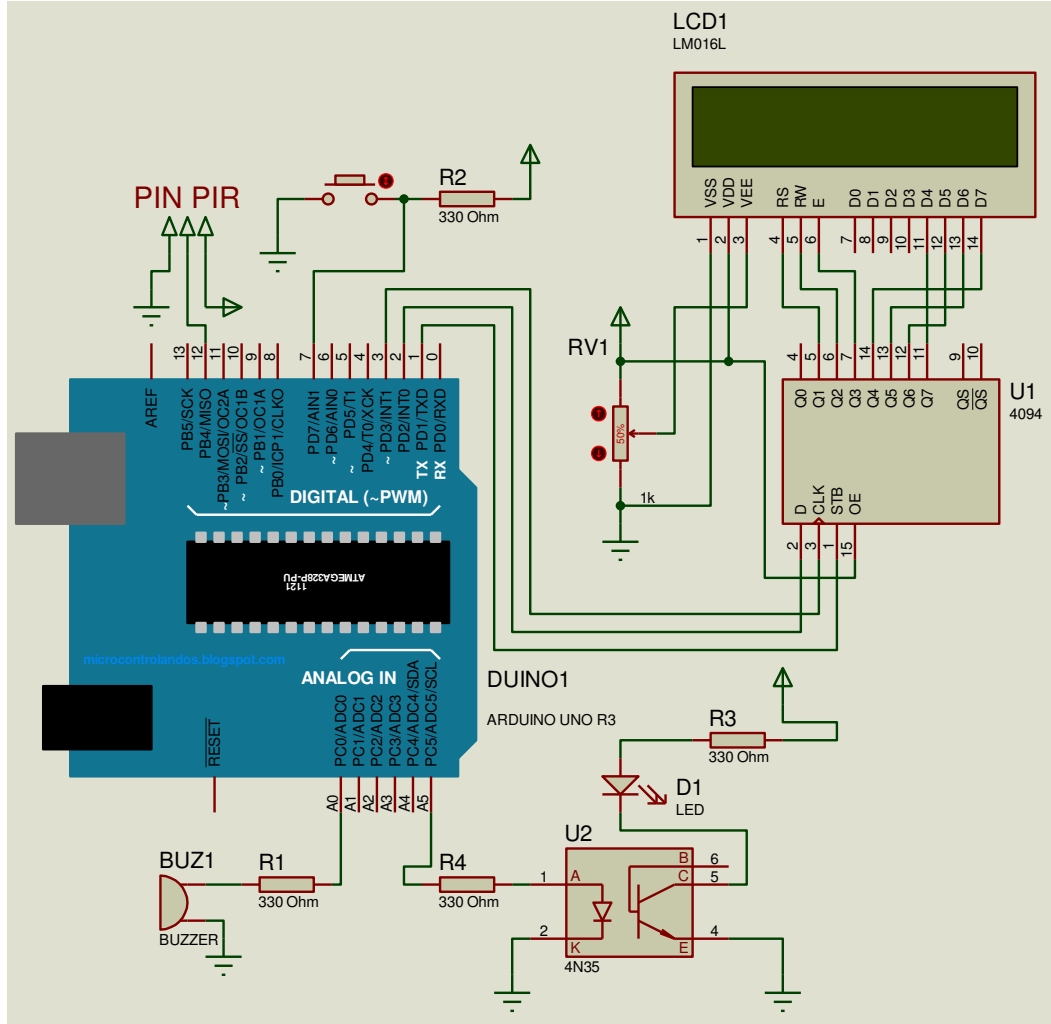
Sensor PIR ini bekerja dengan menangkap energi panas yang dihasilkan dari pancaran sinar inframerah pasif yang dimiliki setiap benda dengan suhu benda diatas nol mutlak. Seperti tubuh manusia yang memiliki suhu tubuh kira-kira 32 derajat celcius, yang merupakan suhu panas yang khas yang terdapat pada lingkungan. Pancaran sinar inframerah inilah yang kemudian ditangkap oleh Pyroelectric sensor yang merupakan inti dari sensor PIR ini sehingga menyebabkan Pyroelectric sensor yang terdiri dari galium nitrida, caesium nitrat dan litium tantalate menghasilkan arus listrik. Mengapa bisa menghasilkan arus listrik? Karena pancaran sinar inframerah pasif ini membawa energi panas. Prosesnya hampir sama seperti arus listrik yang terbentuk ketika sinar matahari mengenai solar cell.

Mengapa sensor PIR hanya bereaksi pada tubuh manusia saja? Hal ini disebabkan karena adanya IR Filter yang menyaring panjang gelombang sinar inframerah pasif. IR Filter dimodul sensor PIR ini mampu menyaring panjang gelombang sinar inframerah pasif antara 8 sampai 14 mikrometer, sehingga

panjang gelombang yang dihasilkan dari tubuh manusia yang berkisar antara 9 sampai 10 mikrometer ini saja yang dapat dideteksi oleh sensor.

5. PROSEDUR PRAKTIKUM

A. Buat rangkaian sesuai dengan skematik berikut :



Catatan : ditambahkan hambatan 330 Ohm pada input LED + pada kaki 15 LCD, dan kaki 16 diground

B. Percobaan dalam praktikum

1. PIR - Serial

a. Tuliskan program dibawah ini pada software *Arduino* dan upload keyboard Arduino Uno R3 :

```
int calibrationTime = 30;
long unsigned int lowIn;
long unsigned int pause = 5000;
boolean lockLow = true;
boolean takeLowTime;
int pirPin = 12;
int ledPin = A5;

void setup() {
  Serial.begin(9600);
```



```

pinMode(pirPin, INPUT);
pinMode(ledPin, OUTPUT);
digitalWrite(pirPin, LOW);

Serial.print("calibrating sensor ");
for(int i = 0; i < calibrationTime; i++){
  Serial.print(".");
  delay(1000);
}
Serial.println(" done");
Serial.println("SENSOR ACTIVE");
delay(50);
}

void loop(){

  if(digitalRead(pirPin) == HIGH){
    digitalWrite(ledPin, HIGH);
    if(lockLow){
      lockLow = false;
      Serial.println("---");
      Serial.print("motion detected at ");
      Serial.print(millis()/1000);
      Serial.println(" sec");
      delay(50);
    }
    takeLowTime = true;
  }

  if(digitalRead(pirPin) == LOW){
    digitalWrite(ledPin, LOW);
    if(takeLowTime){
      lowIn = millis();
      takeLowTime = false;
    }

    if(!lockLow && millis() - lowIn > pause){
      lockLow = true;
      Serial.print("motion ended at ");
      Serial.print((millis() - pause)/1000);
      Serial.println(" sec");
      delay(50);
    }
  }
}

```

- b. Lakukan modifikasi pada rangkaian diatas dan berikan komentar pada setiap line program diatas.

6. KASUS PERCOBAAN

- a. Buat sebuah aplikasi dengan menggunakan 1 push button, 4N35, LED, dan PIR (posisi port dapat dirubah sesuai dengan kebutuhan)
- b. Terdapat kendali on/off, PWM dan delay dan shift register,
- c. Terdapat interface analog dan digital
- d. Catat skematik beserta pin/port yang digunakan, dan program yang dibuat pada kasus diatas dan perlihatkan pada asisten
- e. Kasusnya :

- i. Push button ditekan pertama kali untuk menjalankan PIR dan menunjukkan fungsi kerja PIR dalam LCD dan serial monitor
- ii. Push button ditekan kedua kali untuk menunjukkan kerja PIR jika ada orang dalam bentuk buzzer, LCD dan serial monitor
- iii. Push button ditekan ketiga kali menunjukkan kerja PIR dengan keluaran LCD dan serial monitor dan jika ada orang dalam bentuk LED kelap-kelip jika ada orang dan LED akan mati jika tidak ada orang.
- iv. Push button ditekan keempat kali kembali ke kasus 1

7. LAPORAN AKHIR

- a. Sesuaikan dengan aturan praktikum sebelumnya
- b. Lampirkan hasil percobaan, skematik dan program yang diuji beserta komentar
- c. Jelaskan kasus percobaan 6 dari sisi interface dan peripheral, sistem kendali, dan pemrograman mikrokontroler.
- d. Jelaskan cara kerja dan jelaskan juga port mana saja yang bisa digunakan.
 - i. PIR

MODUL 8

PRAKTIKUM INTERFACE, PEMOGRAMAN MIKROKONTROLER, DAN SISTEM KENDALI, FAKULTAS ILMU TERAPAN, UNIVERSITAS TELKOM

1. JUDUL PRAKTIKUM

PING (Ultrasonic)

2. MAKSUD DAN TUJUAN

Maksud dan tujuan dari praktikum ini adalah :

1. Mahasiswa mampu menggunakan pin-pin pada mikrokontroler dalam mengendalikan PING
2. Mahasiswa mampu menyelesaikan kasus tertentu dengan PING dalam mikrokontroler.

3. PERALATAN DAN BAHAN

Peralatan yang dibutuhkan dalam praktikum ini adalah :

1. 1 buah Arduino Uno R3 + Kabel USB
2. Jumper + header Secukupnya
3. 7 buah Resistor 330 Ohm (optional)
4. 3 buah LED (optional)
5. 1 buah potensio
6. 1 buah Protoboard
7. 1 buah LCD 16x2
8. 1 buah pin header 16x1
9. 1 buah IC Shift register 4094
10. 1 keypad 3x4
11. 1 seven segmen katoda
12. 1 buzzer
13. 1 push button
14. 1 dot matrix
15. 2 Resistor 33 K
16. 1 4N35
17. 1 PING

4. TEORI DASAR PING

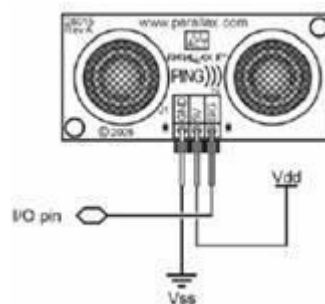
Sensor PING merupakan sensor ultrasonik yang dapat mendeteksi jarak obyek dengan cara memancarkan gelombang ultrasonik dengan frekuensi 40 KHz dan kemudian mendeteksi pantulannya. Tampilan sensor jarak PING ditunjukkan pada Gambar berikut:



Sensor ini dapat mengukur jarak antara 3 cm sampai 300 cm. keluaran dari sensor ini berupa pulsa yang lebarnya merepresentasikan jarak. Lebar pulsanya bervariasi dari 115 μ S sampai 18,5 mS. Pada dasarnya, PING terdiri dari sebuah chip pembangkit sinyal 40KHz, sebuah speaker ultrasonik dan sebuah mikropon ultrasonik. Speaker ultrasonik mengubah sinyal 40 KHz menjadi suara sementara mikropon ultrasonik berfungsi untuk mendeteksi pantulan suaranya.

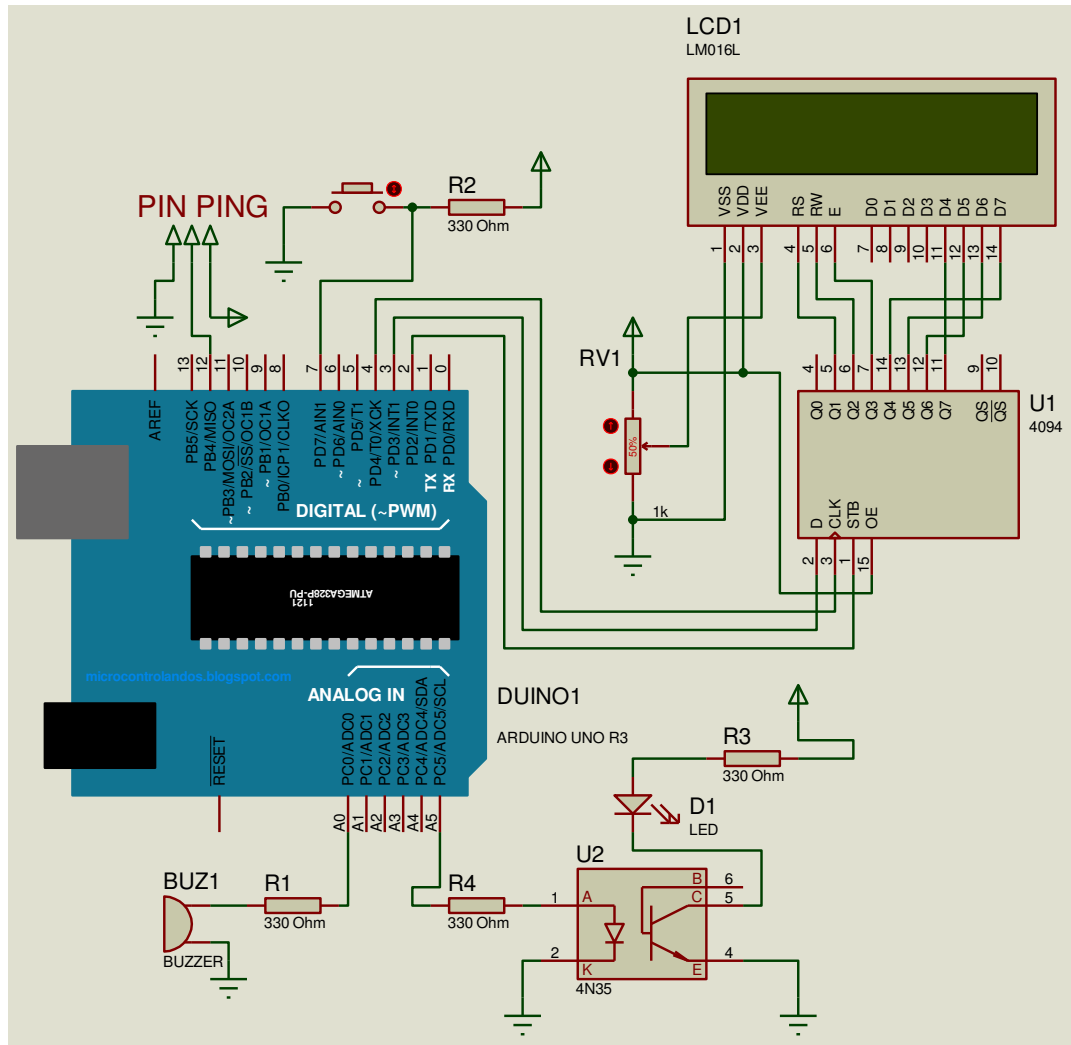
Pin signal dapat langsung dihubungkan dengan mikrokontroler tanpa tambahan komponen apapun. Ping hanya akan mengirimkan suara ultrasonik ketika ada pulsa trigger dari mikrokontroler (Pulsa high selama 5 μ S). Suara ultrasonik dengan frekuensi sebesar 40KHz akan dipancarkan selama 200 μ S. Suara ini akan merambat di udara dengan kecepatan 344.424m/detik (atau 1cm setiap 29.034 μ S), mengenai objek untuk kemudian terpantul kembali ke Ping. Selama menunggu pantulan, Ping akan menghasilkan sebuah pulsa. Pulsa ini akan berhenti (low) ketika suara pantulan terdeteksi oleh Ping. Oleh karena itulah lebar pulsa tersebut dapat merepresentasikan jarak antara Ping dengan objek.

Konfigurasi pin sensor PING sbagai berikut:



5. TEORI DASAR PING

Buat rangkaian sesuai dengan skematik berikut :



Catatan : ditambahkan hambatan 330 Ohm pada input LED + pada kaki 15 LCD, dan kaki 16 diground

A. Percobaan dalam praktikum

1. PING - Serial

- Tuliskan program dibawah ini pada software *Arduino* dan upload keyboard Arduino Uno R3 :

```
const int pingPin = 12;
void setup() {
  Serial.begin(9600);
}
void loop()
{
  long duration, inches, cm;
  pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
```

```

delayMicroseconds(5);
digitalWrite(pingPin, LOW);
pinMode(pingPin, INPUT);
duration = pulseIn(pingPin, HIGH);
inches = microsecondsToInches(duration);
cm = microsecondsToCentimeters(duration);
Serial.print(inches);
Serial.print("in, ");
Serial.print(cm);
Serial.print("cm");
Serial.println();
delay(100);
}
long microsecondsToInches(long microseconds)
{
    return microseconds / 74 / 2;
}
long microsecondsToCentimeters(long microseconds)
{
    return microseconds / 29 / 2;
}

```

- b. Lakukan modifikasi pada rangkaian diatas dan berikan komentar pada setiap line program diatas.

6. KASUS PERCOBAAN

- a. Buat sebuah aplikasi dengan menggunakan 1 Potensio, 1 push button, 1 shift register 4094, 1 buah LCD, buzzer dan PING.
- b. Terdapat kendali on/off, PWM dan delay dan shift register,
- c. Terdapat interface analog dan digital
- d. Catat skematik beserta pin/port yang digunakan, dan program yang dibuat pada kasus diatas dan perlihatkan pada asisten
- e. Kasusnya :
 - i. Push button ditekan pertama kali untuk menjalankan PING dan menunjukkan fungsi kerja PING dalam LCD dan serial monitor
 - ii. Push button ditekan kedua kali untuk menunjukkan kerja PING jika nilai PING melewati batas tertentu ada peringatan dalam bentuk buzzer, LCD dan serial monitor
 - iii. Push button ditekan ketiga kali menunjukkan kerja PING dengan keluaran LCD, serial monitor dan bentuk LED fading PWM sesuai dengan keluaran PING.
 - iv. Push button ditekan keempat kali kembali ke kasus 1

7. LAPORAN AKHIR

- a. Sesuaikan dengan aturan praktikum sebelumnya
- b. Lampirkan hasil percobaan, skematik dan program yang diuji beserta komentar
- c. Jelaskan kasus percobaan 6 dari sisi interface dan pheripheral, sistem kendali, dan pemograman mikrokontroler.
- d. Jelaskan cara kerja dan jelaskan juga port mana saja yang bisa digunakan.
 - i. PING

MODUL 9

PRAKTIKUM INTERFACE, PEMOGRAMAN MIKROKONTROLER, DAN SISTEM KENDALI, FAKULTAS ILMU TERAPAN, UNIVERSITAS TELKOM

1. JUDUL PRAKTIKUM

Sensor Proximity

2. MAKSUD DAN TUJUAN

Maksud dan tujuan dari praktikum ini adalah :

1. Mahasiswa mampu menggunakan pin-pin pada mikrokontroler dalam mengendalikan sensor proximity
2. Mahasiswa mampu menyelesaikan kasus tertentu dengan sensor proximity dalam mikrokontroler.

3. PERALATAN DAN BAHAN

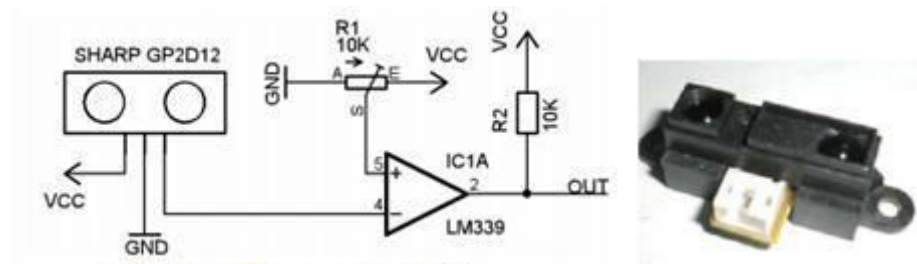
Peralatan yang dibutuhkan dalam praktikum ini adalah :

1. 1 buah Arduino Uno R3 + Kabel USB
2. Jumper + header Secukupnya
3. 7 buah Resistor 330 Ohm (optional)
4. 3 buah LED (optional)
5. 1 buah potensio
6. 1 buah Protoboard
7. 1 buah LCD 16x2
8. 1 buah pin header 16x1
9. 1 buah IC Shift register 4094
10. 1 keypad 3x4
11. 1 seven segmen katoda
12. 1 buzzer
13. 1 push button
14. 1 dot matrix
15. 2 Resistor 33 K
16. 1 4N35
17. 1 Sensor Proximity

4. TEORI DASAR PROXIMITY SENSOR

Sensor sharp digunakan untuk membaca jarak. Sensor ini menggunakan prinsip pantulan sinar infra merah. Dalam aplikasi ini nilai tegangan keluran dari sensor yang berbanding terbalik dengan hasil pembacaan jarak dikomparasi dengan tegangan referensi komparator.

Rangkaian sistem komparator pembacaan jarak dengan sensor sharp ini disajikan pada Gambar berikut:



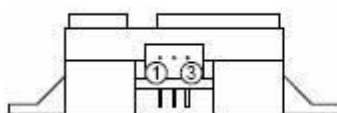
Prinsip kerja dari rangkaian komparator sensor sharp pada Gambar diatas adalah jika sensor mengeluarkan tegangan melebihi tegangan referensi, maka keluaran dari komparator akan berlogika rendah. Jika tegangan referensi lebih besar dari tegangan sensor maka keluaran dari komparator akan berlogika tinggi.

Selain menggunakan komparator, untuk mengakases sensor jarak sharp dapat dengan menggunakan prinsip ADC, atau dengan kata lain mengolah sinyal analog dari pembacaan sensor sharp ke bentuk digital dengan bantuan pemrograman.

Spesifikasi Sensor Infra Merah Sharp

- Power Supply : 4.5 to 5.5 volt
- Voutput : 0,25 to 0,55 volt
- Distance Range : 20 to 150 cm

Pin-pin pada sensor proximity dengan kemampuannya :

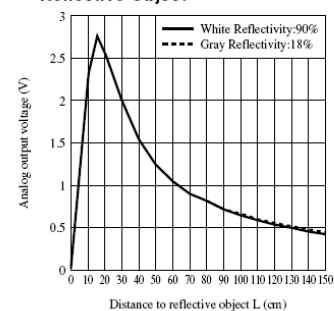


Terminal connection

- ① V_O
- ② GND
- ③ V_{CC}

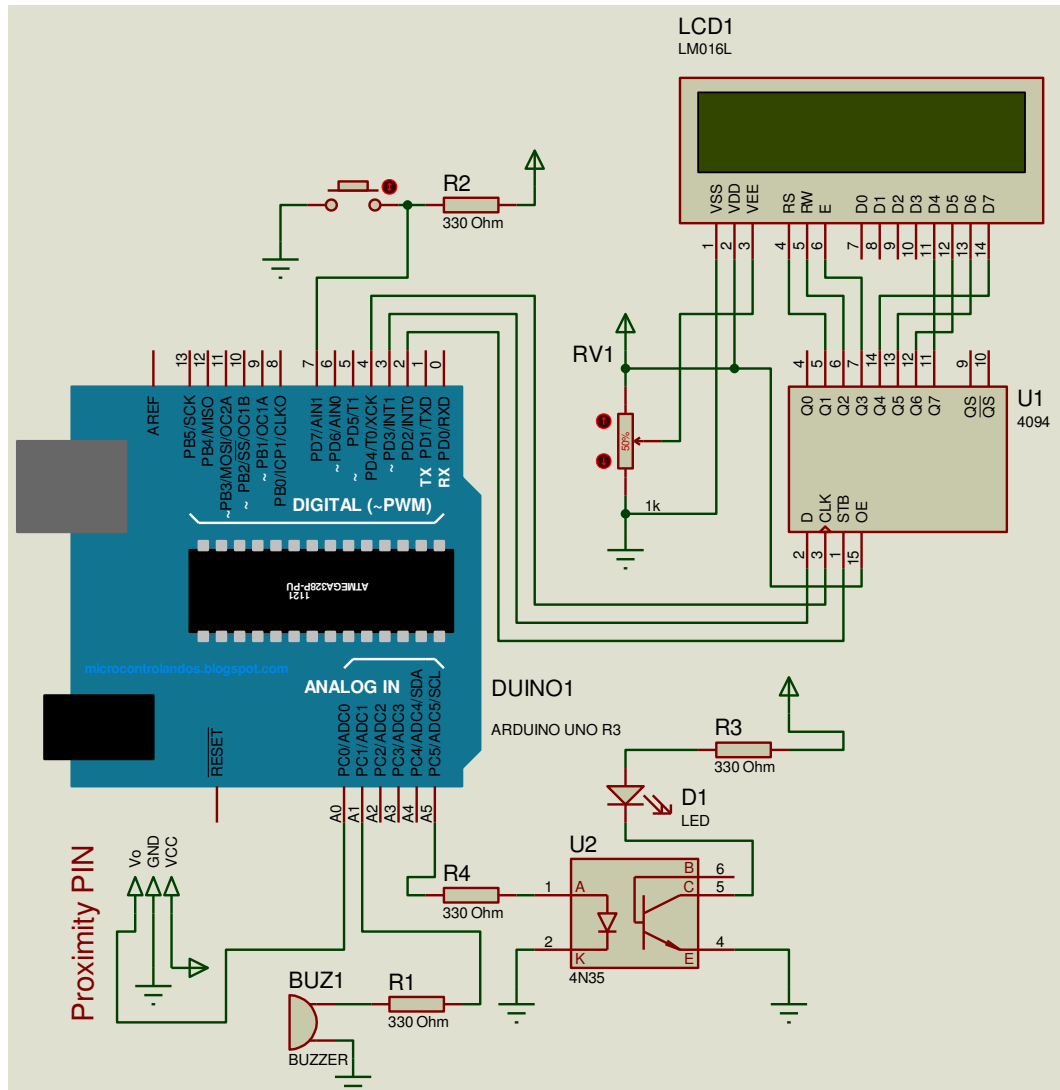
SHARP

Fig.3 Analog Output Voltage vs. Distance to Reflective Object



5. PROSEDUR PRAKTIKUM

Buat rangkaian sesuai dengan skematik berikut :



Catatan : ditambahkan hambatan 330 Ohm pada input LED + pada kaki 15 LCD, dan kaki 16 diground

A. Percobaan dalam praktikum

1. Sharp - Serial

- a. Tuliskan program dibawah ini pada software *Arduino* dan upload keboard Arduino Uno R3 :

```
int IRpin = A0;
void setup() {
  Serial.begin(9600);
}

void loop() {
  float volts = analogRead(IRpin)*0.0048828125;
  float distance = 65*pow(volts, -1.10);
  Serial.println(distance);
  delay(100);
}
```

- b. Lakukan modifikasi pada rangkaian diatas dan berikan komentar pada setiap line program diatas.

6. KASUS PERCOBAAN

- a. Buat sebuah aplikasi dengan menggunakan 1 Potensio, 1 push button, 1 shift register 4094, 1 buah LCD, buzzer dan sensor proximity.
- b. Terdapat kendali on/off, PWM dan delay dan shift register,
- c. Terdapat interface analog dan digital
- d. Catat skematik beserta pin/port yang digunakan, dan program yang dibuat pada kasus diatas dan perlihatkan pada asisten
- e. Kasusnya :
 - i. Push button ditekan pertama kali untuk menjalankan sensor proximity dan menunjukkan fungsi kerja sensor proximity dalam LCD dan serial monitor
 - ii. Push button ditekan kedua kali untuk menunjukkan kerja sensor proximity jika nilai sensor proximity melewati batas tertentu ada peringatan dalam bentuk buzzer, LCD dan serial monitor
 - iii. Push button ditekan ketiga kali menunjukkan kerja sensor proximity dengan keluaran LCD, serial monitor dan bentuk LED fading PWM sesuai dengan keluaran sensor proximity.
 - iv. Push button ditekan keempat kali kembali ke kasus 1

7. LAPORAN AKHIR

- a. Sesuaikan dengan aturan praktikum sebelumnya
- b. Lampirkan hasil percobaan, skematik dan program yang diuji beserta komentar
- c. Jelaskan kasus percobaan 6 dari sisi interface dan pheripheral, sistem kendali, dan pemograman mikrokontroler.
- d. Jelaskan cara kerja dan jelaskan juga port mana saja yang bisa digunakan.
 - i. sensor proximity

MODUL 10

PRAKTIKUM INTERFACE, PEMOGRAMAN MIKROKONTROLER, DAN SISTEM KENDALI, FAKULTAS ILMU TERAPAN, UNIVERSITAS TELKOM

1. JUDUL PRAKTIKUM

Sensor Gyroscope

2. MAKSUD DAN TUJUAN

Maksud dan tujuan dari praktikum ini adalah :

1. Mahasiswa mampu menggunakan pin-pin pada mikrokontroler dalam mengendalikan sensor gyroscope
2. Mahasiswa mampu menyelesaikan kasus tertentu dengan sensor gyroscope dalam mikrokontroler.

3. PERALATAN DAN BAHAN

Peralatan yang dibutuhkan dalam praktikum ini adalah :

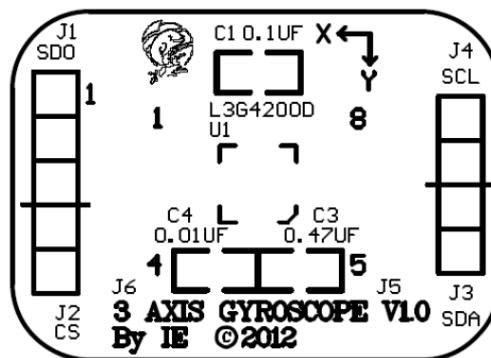
1. 1 buah Arduino Uno R3 + Kabel USB
2. Jumper + header Secukupnya
3. 7 buah Resistor 330 Ohm (optional)
4. 3 buah LED (optional)
5. 1 buah potensio
6. 1 buah Protoboard
7. 1 buah LCD 16x2
8. 1 buah pin header 16x1
9. 1 buah IC Shift register 4094
10. 1 keypad 3x4
11. 1 seven segmen katoda
12. 1 buzzer
13. 1 push button
14. 1 dot matrix
15. 2 Resistor 33 K
16. 1 4N35
17. 1 Gyroscope

4. TEORI DASAR GYROSCOPE

DT-Sense 3 Axis Gyroscope merupakan suatu modul sensor kecepatan sudut (*gyroscope*) yang menggunakan IC L3G4200D produksi STMicroelectronics (www.st.com). Modul ini mampu membaca kecepatan sudut pada 3 sumbu dengan jangkauan pengukuran yang bisa dipilih. Selain itu, modul ini memiliki kemampuan untuk memilih *bandwidth* atau kecepatan output data, memiliki FIFO *buffer*, serta memiliki pin dan sistem interupsi. Modul ini cocok untuk aplikasi seperti alat kendali game atau *virtual reality*, bagian dari sistem navigasi GPS dan robot, alat kendali untuk MMI (*Man Machine Interface*) atau aplikasi-aplikasi lain yang memerlukan data kecepatan sudut.

Spesifikasi :

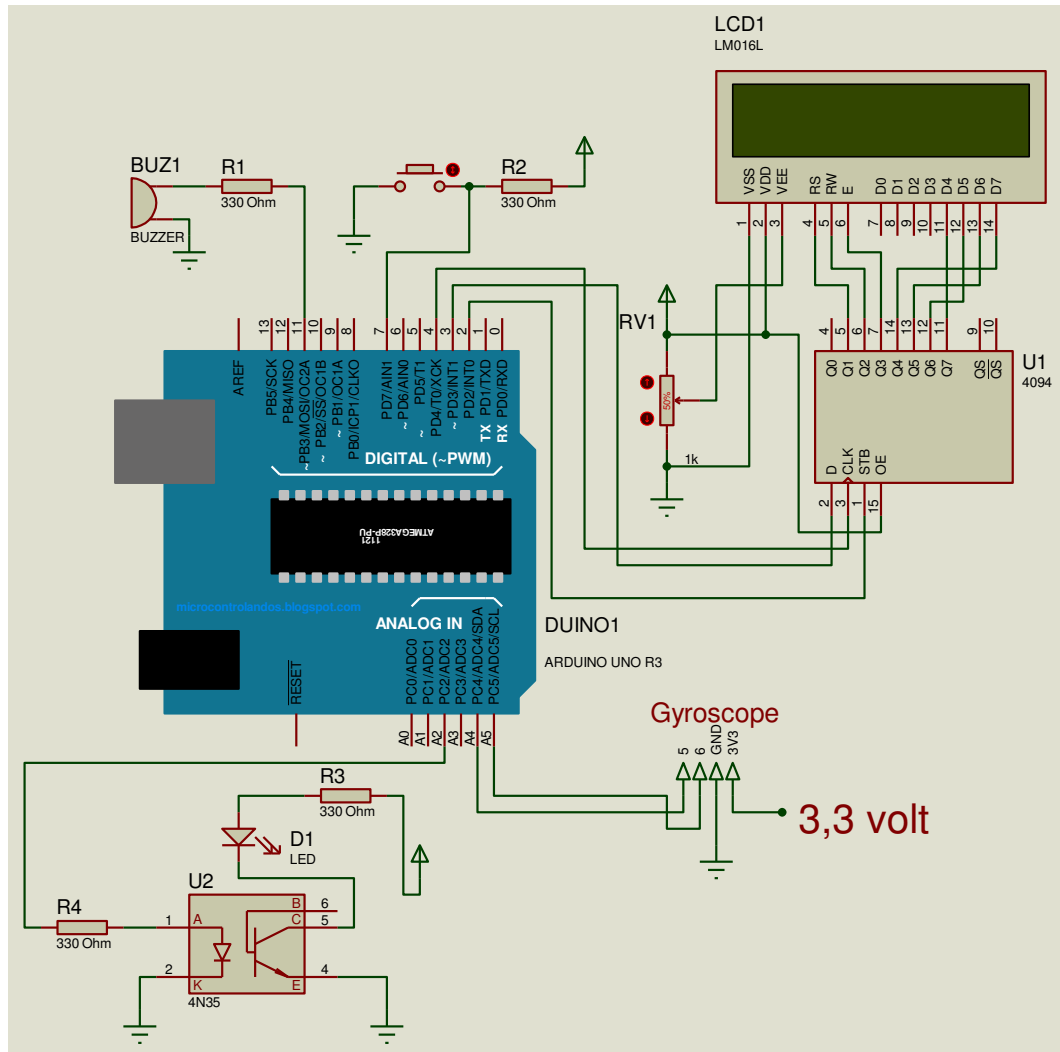
1. Tegangan kerja **3,3 VDC**.
2. Mampu membaca kecepatan sudut pada 3 sumbu.
3. Jangkauan pengukuran yang bisa dipilih yaitu $\pm 250/\pm 500/\pm 2000$ dps (*degree per second*).
4. *Data output* dengan format 16 bit.
5. Dilengkapi dengan sensor temperatur 8 bit.
6. Pin *interrupt* dengan fungsi yang dapat dipilih.
7. Tersedia 2 pilihan antarmuka untuk pembacaan sensor dan konfigurasi modul: I2C dan SPI.



| Pin | Nama | Fungsi | |
|-----|---------|--------------|--|
| 1 | INT1 | Output | Programmable Interrupt |
| 2 | INT2 | Output | Data Ready / FIFO Interrupt |
| 3 | SDO | Input/Output | SPI Serial Data Output / I ² C Address Selector (0 : D0h/D1h ; 1 : D2h/D3h) |
| 4 | CS | Input | Chip Select & SPI Enable (0) atau I ² C Enable (1) |
| 5 | SDA/SDI | Input/Output | I ² C Serial Data (SDA) / SPI Serial Data Input (SDI) |
| 6 | SCL/SPC | Input | I ² C Serial Clock (SCL) / SPI Serial Clock (SPC) |
| 7 | Ground | - | Titik referensi ground |
| 8 | V33 | Input | Tegangan 3,3 VDC |

5. PROSEDUR PRAKTIKUM

Buat rangkaian sesuai dengan skematik berikut :



Catatan : ditambahkan hambatan 330 Ohm pada input LED + pada kaki 15 LCD, dan kaki 16 diground

A. Percobaan dalam praktikum

1. Gyroscope - Serial

- a. Tuliskan program dibawah ini pada software *Arduino* dan upload keboard *Arduino Uno R3* :

```
#include <Wire.h>

#define CTRL_REG1 0x20
#define CTRL_REG2 0x21
#define CTRL_REG3 0x22
#define CTRL_REG4 0x23
#define CTRL_REG5 0x24

int L3G4200D_Address = 105; //I2C address of the L3G4200D

int x;
int y;
```

```

int z;

void setup(){

    Wire.begin();
    Serial.begin(9600);

    Serial.println("starting up L3G4200D");
    setupL3G4200D(2000);

    delay(1500);
}

void loop(){
    getGyroValues();
    Serial.print("X:");
    Serial.print(x);

    Serial.print(" Y:");
    Serial.print(y);

    Serial.print(" Z:");
    Serial.println(z);

    delay(100);
}

void getGyroValues(){

    byte xMSB = readRegister(L3G4200D_Address, 0x29);
    byte xLSB = readRegister(L3G4200D_Address, 0x28);
    x = ((xMSB << 8) | xLSB);

    byte yMSB = readRegister(L3G4200D_Address, 0x2B);
    byte yLSB = readRegister(L3G4200D_Address, 0x2A);
    y = ((yMSB << 8) | yLSB);

    byte zMSB = readRegister(L3G4200D_Address, 0x2D);
    byte zLSB = readRegister(L3G4200D_Address, 0x2C);
    z = ((zMSB << 8) | zLSB);
}

int setupL3G4200D(int scale){
    writeRegister(L3G4200D_Address, CTRL_REG1, 0b00001111);
    writeRegister(L3G4200D_Address, CTRL_REG2, 0b00000000);
    writeRegister(L3G4200D_Address, CTRL_REG3, 0b00001000);
    if(scale == 250){
        writeRegister(L3G4200D_Address, CTRL_REG4, 0b00000000);
    }else if(scale == 500){
        writeRegister(L3G4200D_Address, CTRL_REG4, 0b00010000);
    }else{
        writeRegister(L3G4200D_Address, CTRL_REG4, 0b00110000);
    }
    writeRegister(L3G4200D_Address, CTRL_REG5, 0b00000000);
}

void writeRegister(int deviceAddress, byte address, byte val) {
    Wire.beginTransaction(deviceAddress);
    Wire.write(address);
    Wire.write(val);
    Wire.endTransmission();
}

```

```

}

int readRegister(int deviceAddress, byte address){

    int v;
    Wire.beginTransmission(deviceAddress);
    Wire.write(address);
    Wire.endTransmission();

    Wire.requestFrom(deviceAddress, 1);
    while(!Wire.available()) {

    }

    v = Wire.read();
    return v;
}

```

- b. Lakukan modifikasi pada rangkaian diatas dan berikan komentar pada setiap line program diatas.

6. KASUS PERCOBAAN

- a. Buat sebuah aplikasi dengan menggunakan 1 Potensio, 1 push button, 1 shift register 4094, 1 buah LCD, buzzer dan Gyroscope.
- b. Terdapat kendali on/off, PWM dan delay dan shift register,
- c. Terdapat interface analog dan digital
- d. Catat skematik beserta pin/port yang digunakan, dan program yang dibuat pada kasus diatas dan perlihatkan pada asisten
- e. Kasusnya :
 - i. Push button ditekan pertama kali untuk menjalankan sensor proximity dan menunjukkan fungsi kerja sensor gyroscope dalam LCD dan serial monitor
 - ii. Push button ditekan kedua kali untuk menunjukkan kerja sensor gyroscope jika nilai sensor gyroscope melewati batas tertentu ada peringatan dalam bentuk buzzer, LCD dan serial monitor
 - iii. Push button ditekan ketiga kali menunjukkan kerja sensor gyroscope dengan keluaran LCD, serial monitor dan bentuk LED fading PWM sesuai dengan keluaran sensor gyroscope.
 - iv. Push button ditekan keempat kali kembali ke kasus 1

7. LAPORAN AKHIR

- a. Sesuaikan dengan aturan praktikum sebelumnya
- b. Lampirkan hasil percobaan, skematik dan program yang diuji beserta komentar
- c. Jelaskan kasus percobaan 6 dari sisi interface dan pheripheral, sistem kendali, dan pemograman mikrokontroler.
- d. Jelaskan cara kerja dan jelaskan juga port mana saja yang bisa digunakan.
 - i. I₂C
 - ii. sensor gyroscope

MODUL 11

PRAKTIKUM INTERFACE, PEMOGRAMAN MIKROKONTROLER, DAN SISTEM KENDALI, FAKULTAS ILMU TERAPAN, UNIVERSITAS TELKOM

1. JUDUL PRAKTIKUM

RFID

2. MAKSUD DAN TUJUAN

Maksud dan tujuan dari praktikum ini adalah :

1. Mahasiswa mampu menggunakan pin-pin pada mikrokontroler dalam mengendalikan RFID
2. Mahasiswa mampu menyelesaikan kasus tertentu dengan RFID dalam mikrokontroler.

3. PERALATAN DAN BAHAN

Peralatan yang dibutuhkan dalam praktikum ini adalah :

1. 1 buah Arduino Uno R3 + Kabel USB
2. Jumper + header Secukupnya
3. 7 buah Resistor 330 Ohm (optional)
4. 3 buah LED (optional)
5. 1 buah potensio
6. 1 buah Protoboard
7. 1 buah LCD 16x2
8. 1 buah pin header 16x1
9. 1 buah IC Shift register 4094
10. 1 keypad 3x4
11. 1 seven segmen katoda
12. 1 dot matrix
13. 1 buzzer
14. 1 push button
15. 2 Resistor 33 K
16. 1 4N35
17. 1 RFID

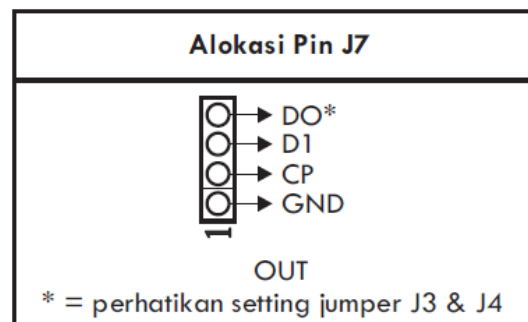
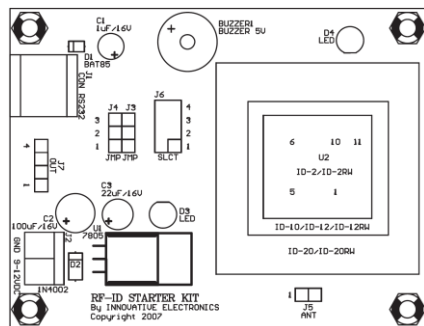
4. TEORI DASAR GYROSCOPE

RFID Starter Kit merupakan suatu sarana pengembangan RFID berbasis reader tipe ID-12 yang telah dilengkapi dengan jalur komunikasi RS-232 serta indikator buzzer dan LED. Modul ini dapat digunakan dalam aplikasi mesin absensi RFID, RFID access controller, dsb.

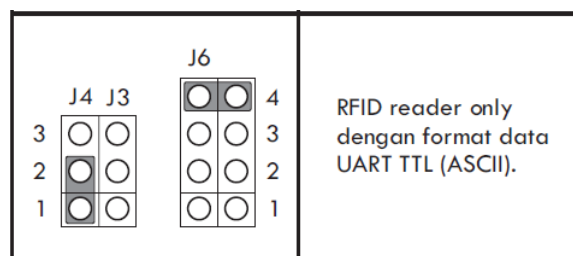
Spesifikasi :

1. Berbasis RFID reader ID-12 dengan frekuensi kerja 125 kHz untuk kartu berformat EM4001/sejenis dan memiliki jarak baca maksimal 12 cm.
2. Kompatibel dengan varian RFID reader lainnya, antara lain: ID-2, ID-10, dan ID-20.
3. Mendukung varian RFID reader/writer, antara lain: ID-2RW, ID-12RW, dan ID-20RW.
4. Mendukung format data ASCII (UART TTL/RS-232), Wiegand26, maupun Magnetic ABA Track2 (Magnet Emulation).
5. Dilengkapi dengan buzzer sebagai indikator baca, serta LED sebagai indikator tulis.
6. Tersedia jalur komunikasi serial UART RS-232 dengan konektor RJ11.
7. Tegangan input catu daya 9 - 12 VDC (J2).
8. Antena dapat dihubungkan ke J5 untuk RFID reader only atau reader/writer yang memerlukan antena eksternal, seperti ID-2 dan ID-2RW.

Tata Letak dan posisi jumper :



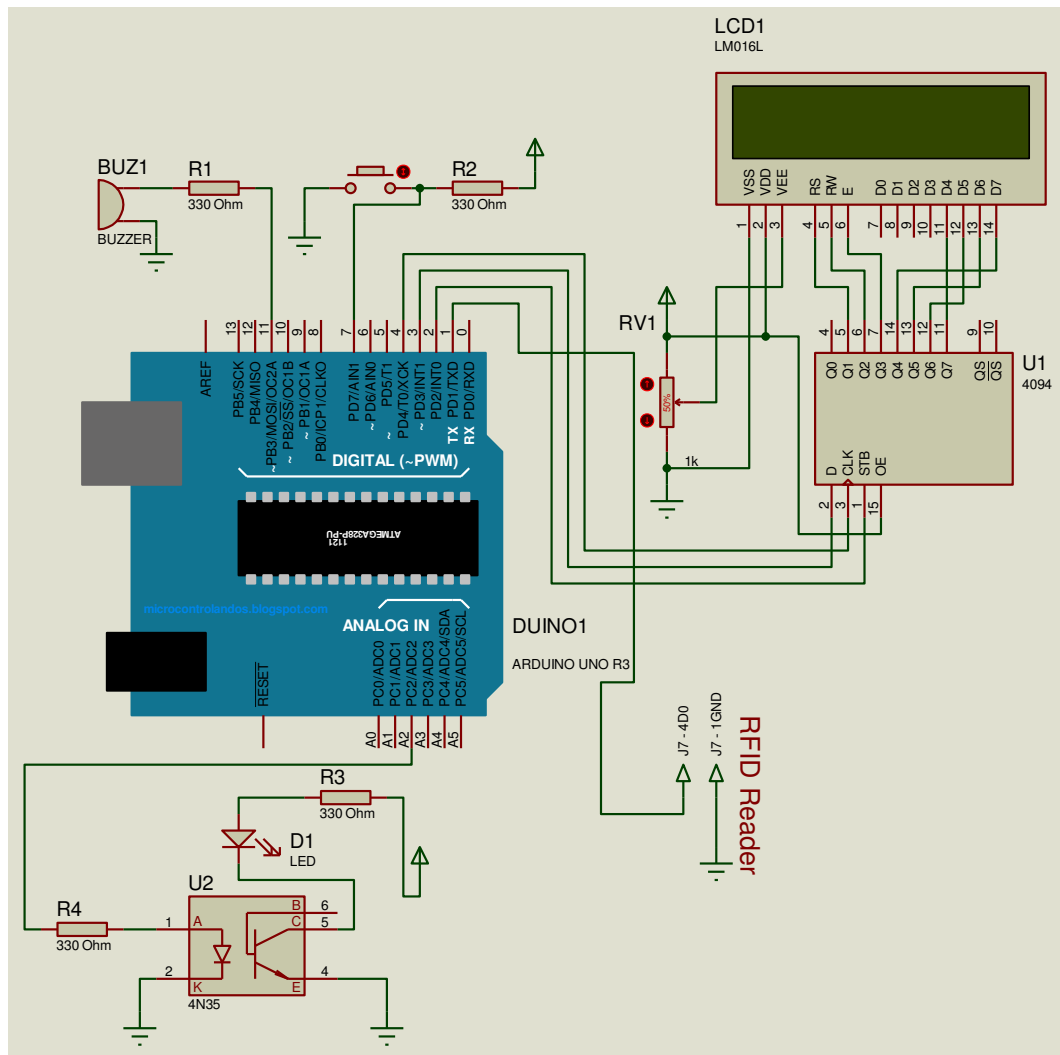
Setting jumper J4, J3 dan J6 :



J7 hanya digunakan untuk RFID reader only dalam mode UART TTL (ASCII), Wiegand26, dan Magnet Emulation. J7 tidak boleh digunakan pada mode lain. Pengaturan jumper J3, J4, dan J6 harus disesuaikan dengan jenis RFID (reader only atau reader/writer) serta format data RFID reader yang akan digunakan.

5. PROSEDUR PRAKTIKUM

Buat rangkaian sesuai dengan skematik berikut :



Catatan : ditambahkan hambatan 330 Ohm pada input LED + pada kaki 15 LCD, dan kaki 16 diground

A. Percobaan dalam praktikum

1. RFID - Serial

- a. Tuliskan program dibawah ini pada software *Arduino* dan upload keboard Arduino Uno R3 : (cabut pin D0 sebelum upload ke Arduino)

```
char val = 0;
void setup() {
  Serial.begin(9600);
}
void loop () {
}
```

- b. Lakukan modifikasi pada rangkaian diatas dan berikan komentar pada setiap line program diatas.

2. Parsing data

- a. Tuliskan program dibawah ini pada software *Arduino* dan upload keyboard *Arduino Uno R3* :

```
#include <SoftwareSerial.h>
#define rxPin 4
#define txPin 5
SoftwareSerial rfid = SoftwareSerial( rxPin, txPin );
#define futureOutput 12
#define ledPin 13
#define unlockSeconds 2

char* allowedTags[] = {
    "[INSERT IDENTIFIER HERE]",          // Tag 1
    "2900940E95",                        // Tag 2
    "ABC123DE45",                        // Tag 3
};
char* tagName[] = {
    "[ADD YOUR NAME HERE]",              // Tag 1
    "Mark Trussell",                     // Tag 2
    "NAME 3",                             // Tag 3
};

int numberOfTags = sizeof(allowedTags)/sizeof(allowedTags[0]);
int incomingByte = 0;
void setup() {
    pinMode(ledPin, OUTPUT);
    digitalWrite(ledPin, LOW);
    pinMode(futureOutput, OUTPUT);
    digitalWrite(futureOutput, LOW);
    Serial.begin(9600);
    rfid.begin(9600);
    Serial.println("RFID Reader Initialized");
}

void loop() {
    byte i          = 0;
    byte val        = 0;
    byte checksum   = 0;
    byte bytesRead  = 0;
    byte tempByte   = 0;
    byte tagBytes[6];
    char tagValue[10];
    if((val = rfid.read()) == 2) {
        bytesRead = 0;
        while (bytesRead < 12) {
            val = rfid.read();
            if (bytesRead < 10)
            {
                tagValue[bytesRead] = val;
            }
            if((val == 0x0D)|| (val == 0x0A)|| (val == 0x03)|| (val ==
0x02)) {
                break;
            }
            if ((val >= '0') && (val <= '9')) {
                val = val - '0';
            }
            else if ((val >= 'A') && (val <= 'F')) {
                val = 10 + val - 'A';
            }
        }
    }
}
```

```

    }
    if (bytesRead & 1 == 1) {
        tagBytes[bytesRead >> 1] = (val | (tempByte << 4));
        if (bytesRead >> 1 != 5) {
            checksum ^= tagBytes[bytesRead >> 1];
        };
    }
    else {
        tempByte = val;
    };
    bytesRead++;
}

if (bytesRead == 12) {
    tagValue[10] = '\0';
    Serial.print("Tag read: ");
    for (i=0; i<5; i++) {
        if (tagBytes[i] < 16) {
            Serial.print("0");
        }
        Serial.print(tagBytes[i], HEX);
    }
    Serial.println();
    Serial.print("Checksum: ");
    Serial.print(tagBytes[5], HEX);
    Serial.println(tagBytes[5] == checksum ? " -- passed." :
" -- error.");
    Serial.print("VALUE: ");
    Serial.println(tagValue);
    int tagId = findTag( tagValue );
    if( tagId > 0 )
    {
        Serial.print("Authorized tag ID ");
        Serial.print(tagId);
        Serial.print(": unlocking for ");
        Serial.println(tagName[tagId - 1]);
        unlock();
    }
    else {
        Serial.println("Tag not authorized");
    }
    Serial.println();
}
bytesRead = 0;
}

void unlock() {
    digitalWrite(ledPin, HIGH);
    digitalWrite(futureOutput, HIGH);
    delay(unlockSeconds * 1000);
    digitalWrite(futureOutput, LOW);
    digitalWrite(ledPin, LOW);
}

int findTag( char tagValue[10] ) {
    for (int thisCard = 0; thisCard < numberOfTags; thisCard++) {
        if(strcmp(tagValue, allowedTags[thisCard]) == 0)
        {
            return(thisCard + 1);
        }
    }
}

```

```
    }  
    return(0);  
}
```

- b. Lakukan modifikasi pada rangkaian diatas dan berikan komentar pada setiap line program diatas.

6. KASUS PERCOBAAN

- a. Buat sebuah aplikasi dengan menggunakan 1 Potensio, 1 push button, 1 shift register 4094, 1 buah LCD, buzzer dan RFID.
- b. Terdapat kendali on/off, PWM dan delay dan shift register,
- c. Terdapat interface analog dan digital
- d. Catat skematik beserta pin/port yang digunakan, dan program yang dibuat pada kasus diatas dan perlihatkan pada asisten
- e. Kasusnya :
 - i. Push button ditekan pertama kali untuk menjalankan RFID (siap untuk membaca RFID)
 - ii. Membaca RFID, jika sesuai dengan data yang disimpan, maka buzzer akan berbunyi sekali dan pelan dan LED fading PWM kelap-kelip sebanyak 3 kali. Nilai RFID ditampilkan di LCD dan Serial Monitor.
 - iii. Membaca RFID, jika tidak sesuai dengan data yang disimpan, maka buzzer akan berbunyi sekali dan panjang dan LED kelap-kelip selama buzzer berbunyi 10 kali. Nilai RFID ditampilkan di LCD dan Serial Monitor.
 - iv. Push button ditekan kedua kali mematikan RFID dan LED berkelap-kelip hingga push button ditekan.
 - v. Push button ditekan sekali lagi kembali ke 1.

7. LAPORAN AKHIR

- a. Sesuaikan dengan aturan praktikum sebelumnya
- b. Lampirkan hasil percobaan, skematik dan program yang diuji beserta komentar
- c. Jelaskan kasus percobaan 6 dari sisi interface dan pheripheral, sistem kendali, dan pemograman mikrokontroler.
- d. Jelaskan cara kerja dan jelaskan juga port mana saja yang bisa digunakan.
 - i. UART
 - ii. TTL
 - iii. RFID