

## MODUL 4

### PRAKTIKUM INTERFACE, PERIPHERAL, DAN KOMUNIKASI

#### 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 :

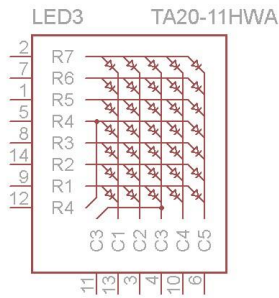
1. 1 buah Arduino Uno R3 + Kabel USB
2. Jumper + header Secukupnya
3. 7 buah Resistor 330 Ohm (optional)
4. 3 buah LED (optional)
5. 1 buah potensio
6. 1 buah Protoboard
7. 1 buah LCD 16x2
8. 1 buah pin header 16x1
9. 1 buah IC Shift register 4094
10. 1 keypad 3x4
11. 1 seven segmen katoda
12. 1 buzzer
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 elektrik. 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
0000000

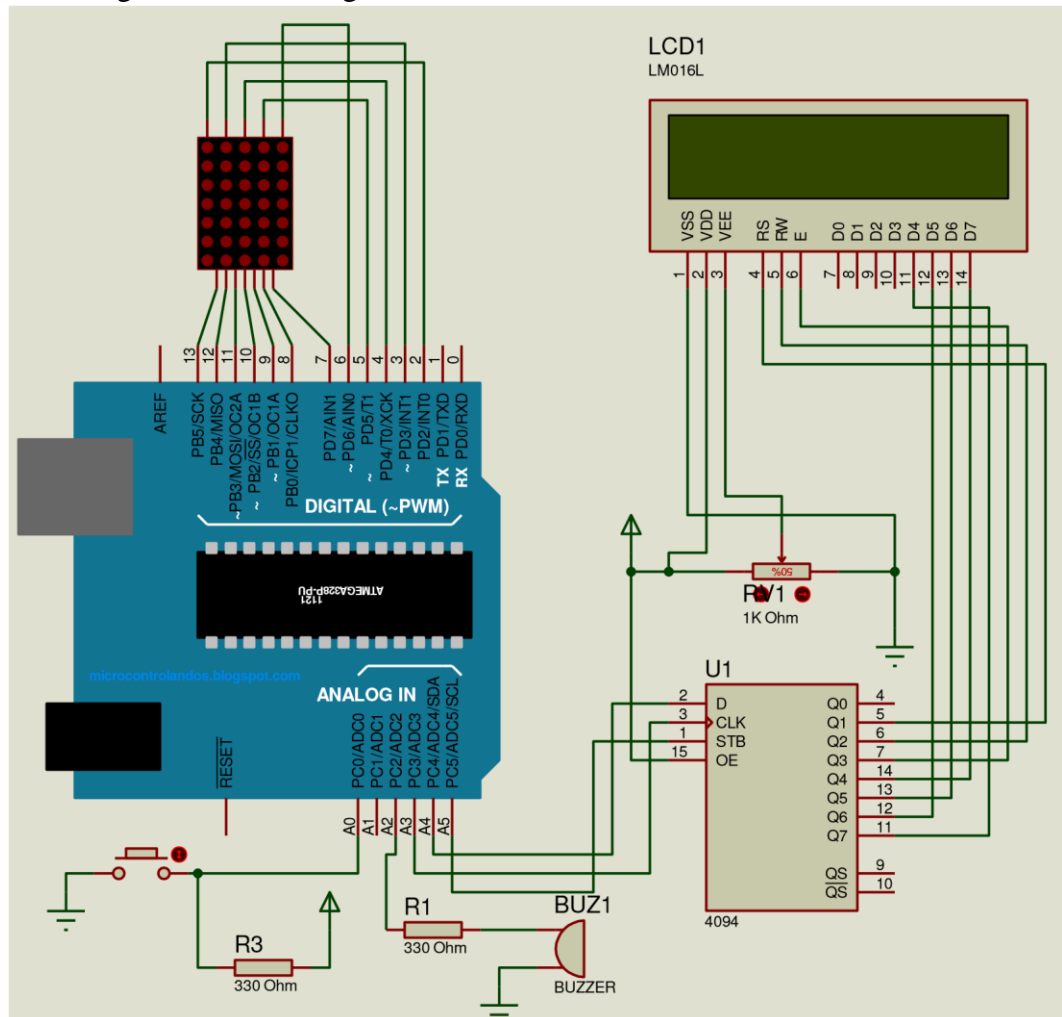
```

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, 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}, \
{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}, \
{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