MIKROPROSESOR & Teknik Antarmuka 1

ARDUINO

PROGRAM STUDI

TEKNIK TELKOMUNIKASI

Semester 4

Akuwan Saleh, MT

PENILAIAN

UTS	= 35%
UAS	= 50%
Tugas-1	= 5%
Tugas-2	= 10%

REFERENSI

- Julien Bayle, “C Programming for Arduino”, Packt Publishing Ltd, Birmingham, May 2013.
- James Floyd K & Harold T , “Arduino Adventure Escape from Gemini Station”, Apress, 2013.
- Famosa Studio Arduino Starter Kit Manual – V1.0, Famosa Studio, 2013.
- Martin E, Joshua N, & Jordan H, “Arduino in Action“,Maniing Publications.Co, USA, 2013.
- Jack Purdum, “Beginning C for Arduino, Learn C Programming for the Arduino and Compatible Microcontrollers”, Apress, 2012.
- John-David Warren, Josh Adams, and Harald Molle, “Arduino Robotics”, Springe, New York, 2011.
- ________, Sistem minimum Arduino Uno/ATmega328, Instruction Manual, 2010.

MATERI

1. PENDAHULUAN
2. ANTARMUKA MIKROKONTROLER DENGAN LIGHT EMITTING DIODE (LED)
3. ANTARMUKA MIKROKONTROLER DENGAN SAKLAR
4. ANTARMUKA MIKROKONTROLER DENGAN LED DOT MATRIK
5. ANTARMUKA MIKROKONTROLER DENGAN KEYPAD
6. ANTARMUKA MIKROKONTROLER DENGAN SEVEN SEGMEN (7-S)
7. ANTARMUKA MIKROKONTROLER DENGAN LCD 2x16
8. PEMROGRAMAN MELODY
9. ANALOG INPUT (ADC)
10. KOMUNIKASI SERIAL
11. ANTARMUKA MIKROKONTROLER DENGAN LM 35
12. ANTARMUKA MIKROKONTROLER DENGAN LDR
13. ANTARMUKA MIKROKONTROLER DENGAN LAMPU AC 220V
14. ANTARMUKA MIKROKONTROLER DENGAN MOTOR DC
15. ANTARMUKA MIKROKONTROLER DENGAN SENSOR ULTRASONIC
16. Demo Tugas Proyek Semester

4. ANTARMUKA MIKROKONTROLER DENGAN LED DOT MATRIK

1. TUJUAN
2. Umum
3. Dot Matrik LED Display
4. Driver/Pengendali
5. Rangkaian interface
6. Program

TUJUAN

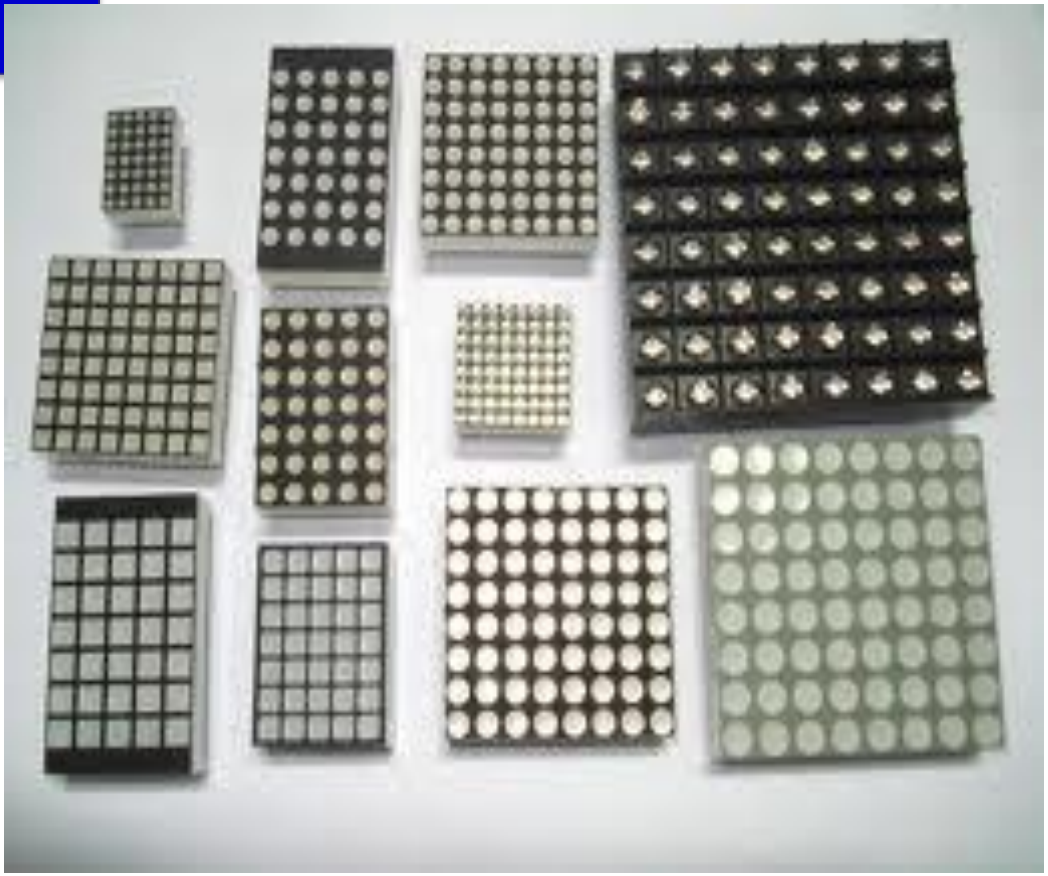
- Membuat aplikasi system tampilan karakter yang berupa LED Dot Matrik dengan menggunakan mikrokontroler Arduino
- Membuat program untuk menampilkan karekter pada media LED Dot Matrik

DASAR TEORI

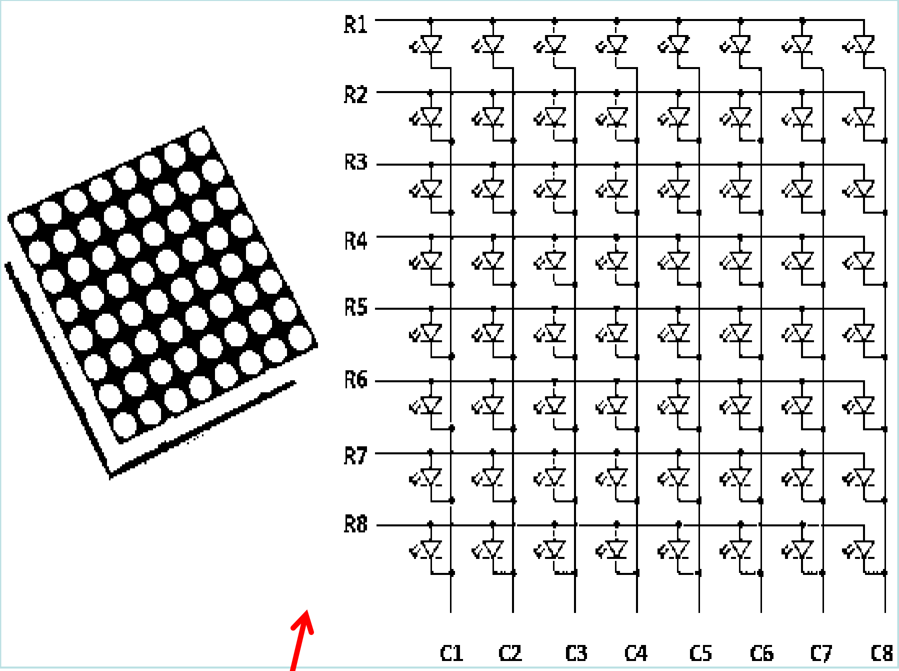
Umum :

- **Display dot matrik** : Beberapa LED yang disusun membentuk matrik baris dan kolom yang bervariasi sesuai dengan tipenya.
- **Tipe**: 5x7 atau 8x8, berupa modul jadi atau dibuat sendiri dengan menyusun beberapa LED.
- **Ukuran**: ada yang besar, sedang maupun yang kecil.

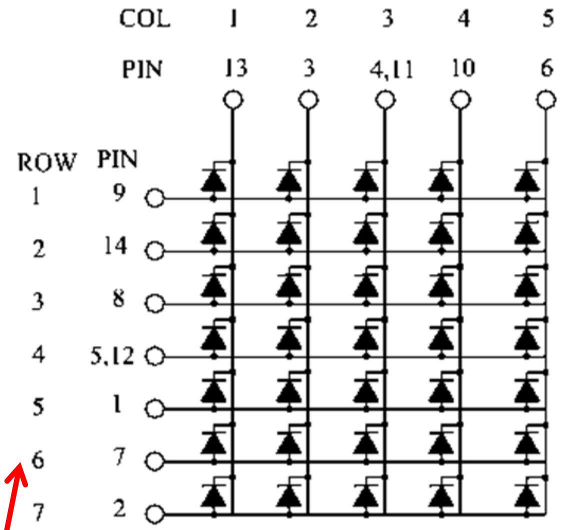
Secara fisik, bentuknya

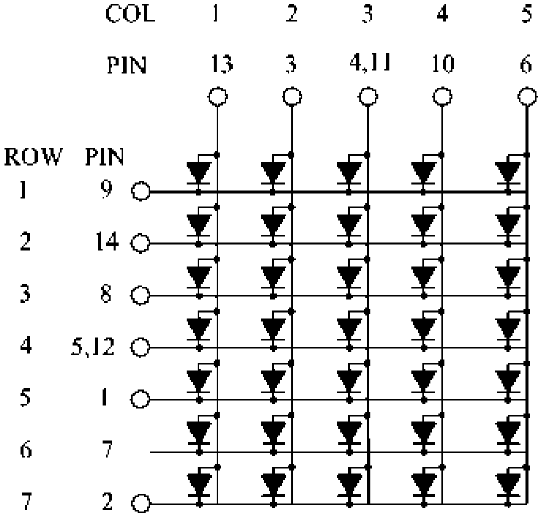


Konfigurasi:

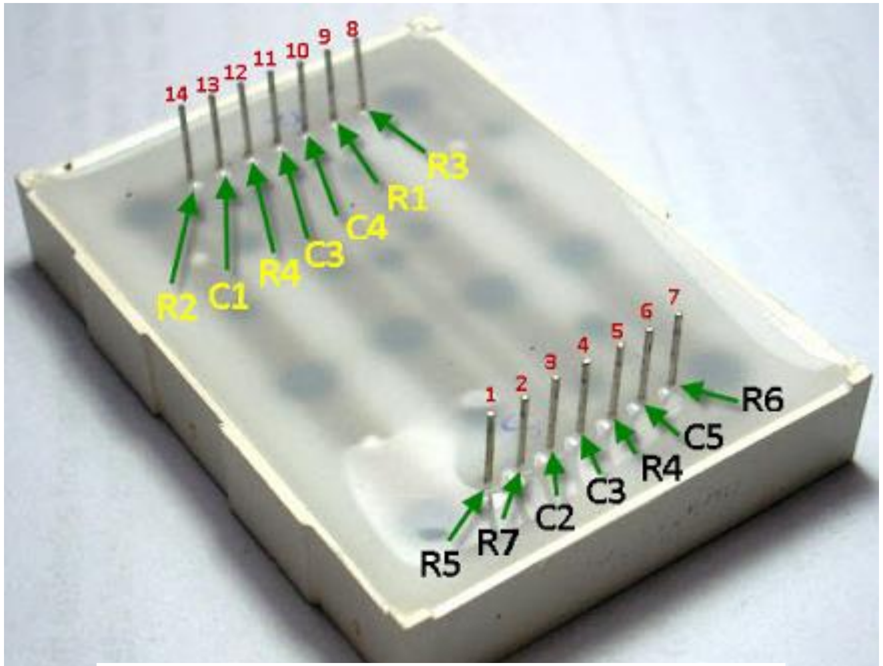


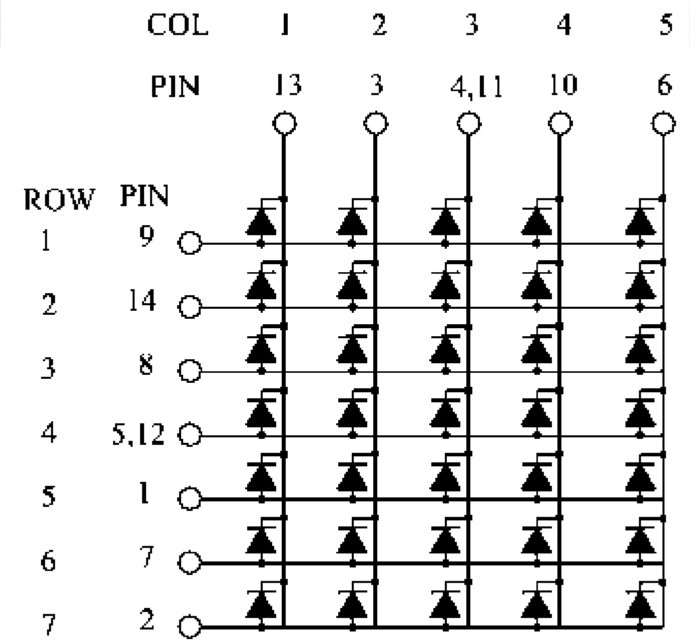
Baris = Anoda → +3,3V / +5V
Kolom = Katoda → GND





Kolom = Anoda → +3,3V / +5V
Baris = Katoda → GND





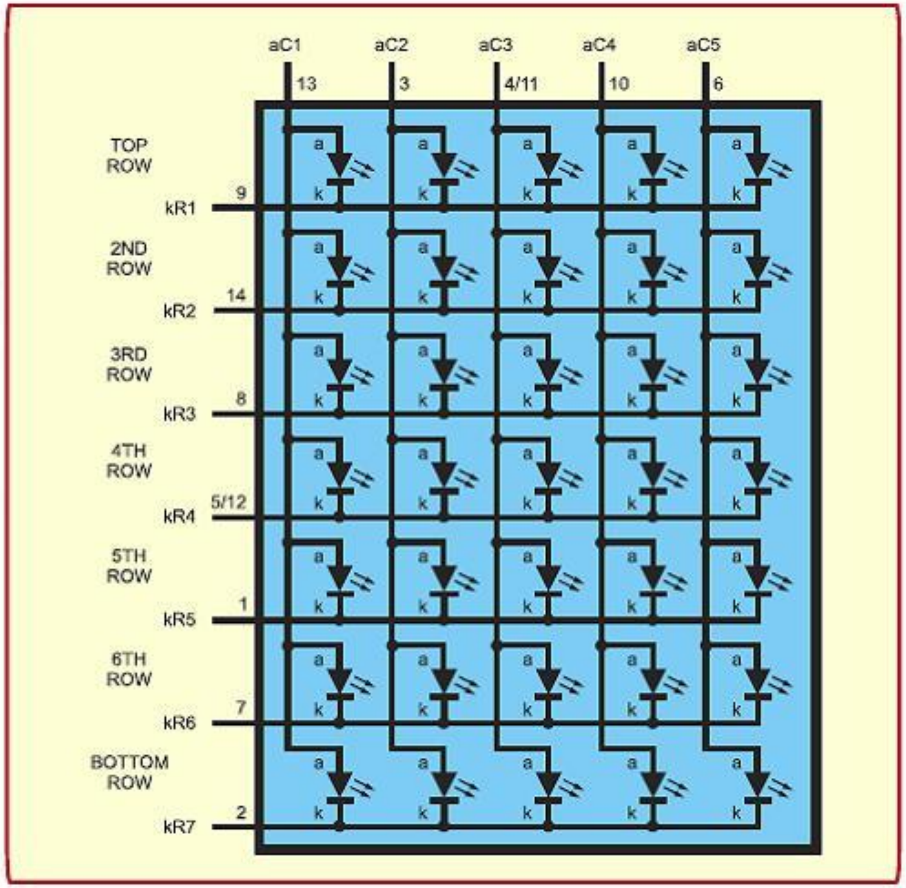
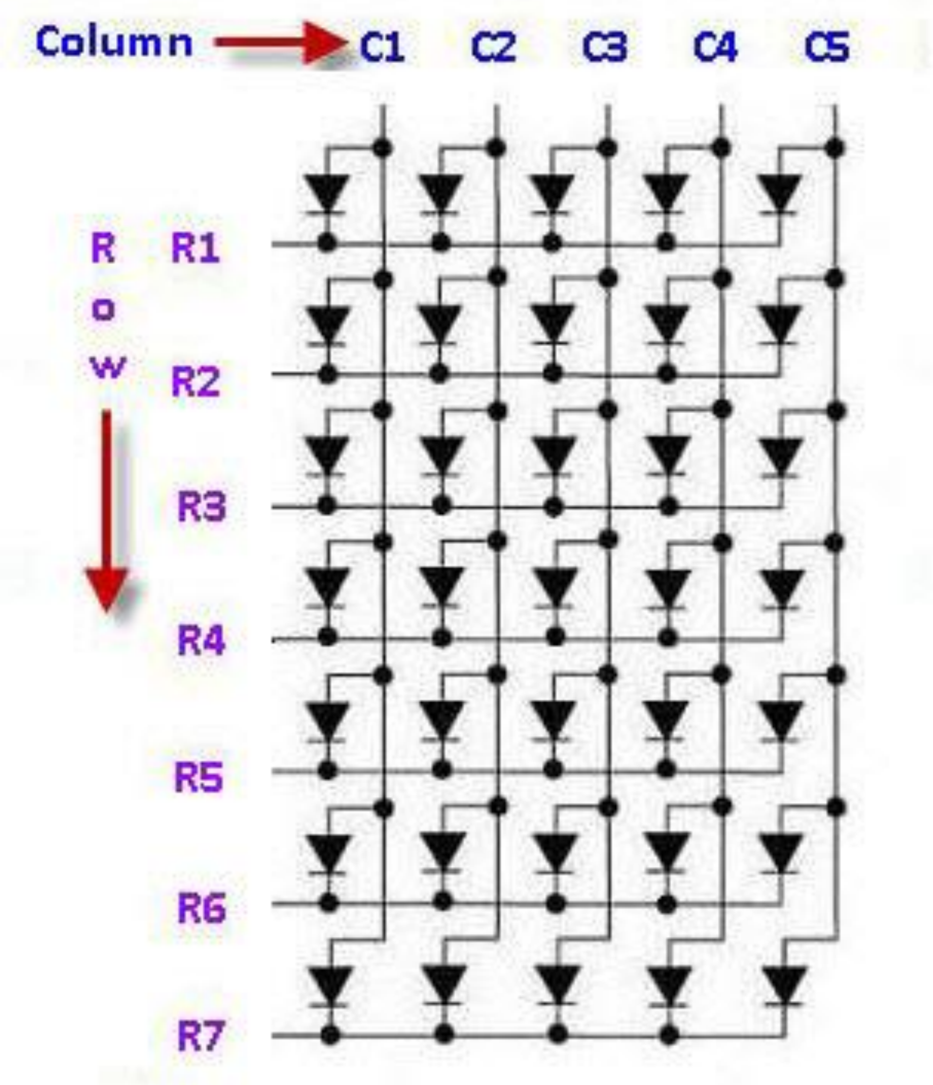


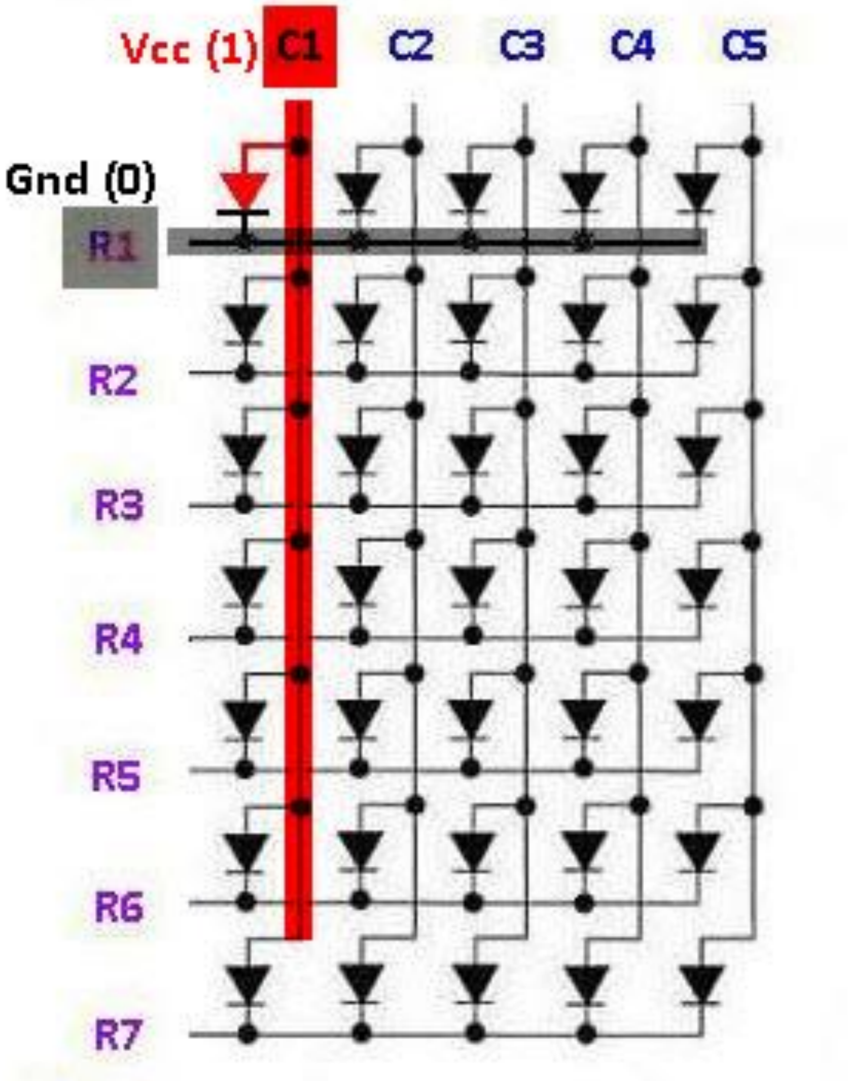
Fig.1. Internal structure of a 7 × 5 matrixed LED module

Kolom = Anoda → +3,3V / +5V
Baris = Katoda → GND

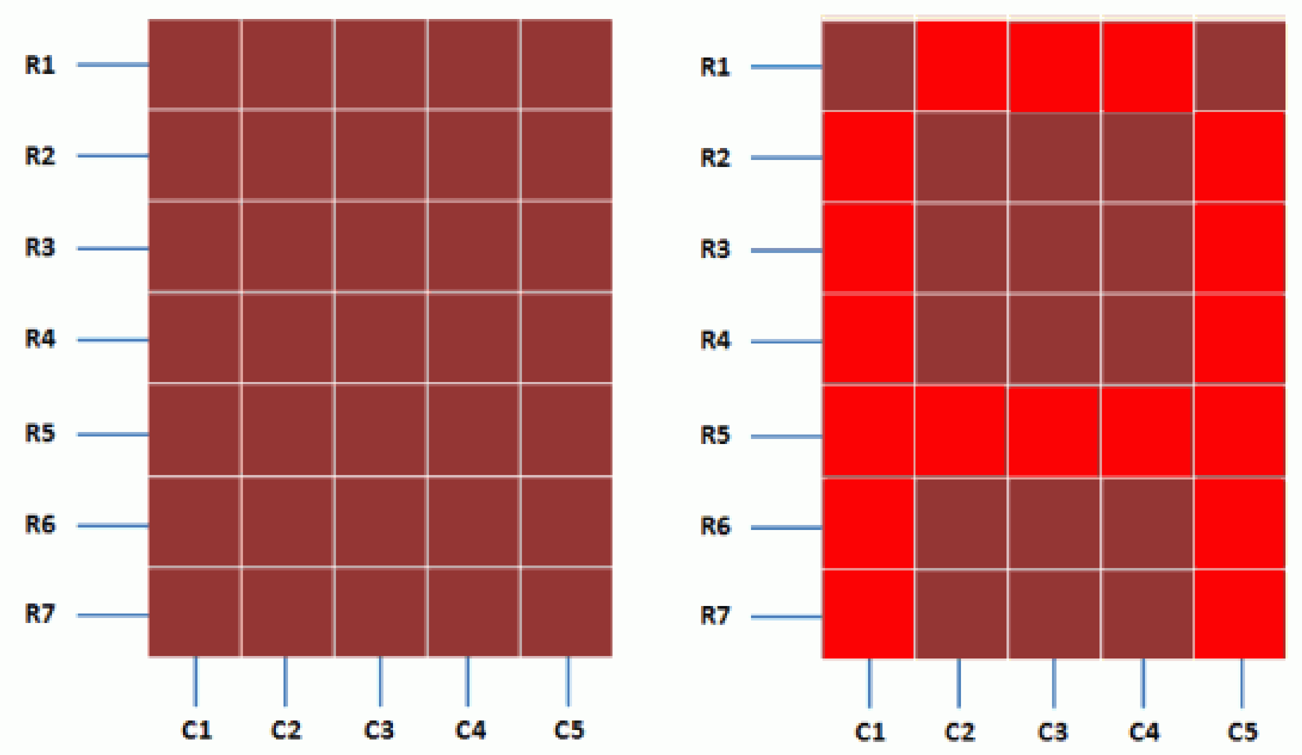
Baris = Anoda → +3,3V / +5V
Kolom = Katoda → GND

Cara Pengujian:





5x7 matrix of LEDs



All LEDs OFF

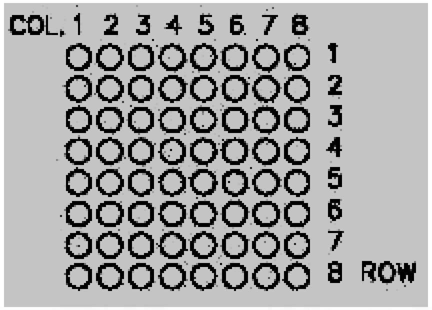
Displaying 'A'

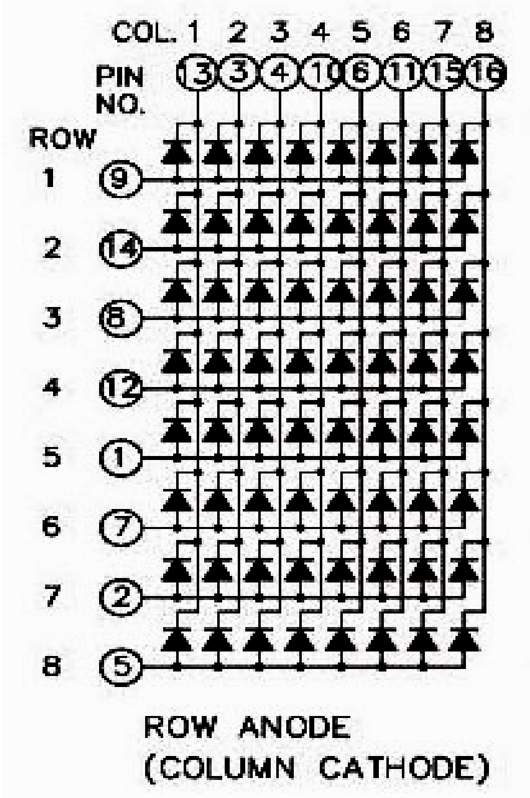
Row\Col	C1	C2	C3	C4	C5
R1	0	1	1	1	0
R2	1	0	0	0	1
R3	1	0	0	0	1
R4	1	0	0	0	1
R5	1	1	1	1	1
R6	1	0	0	0	1
R7	1	0	0	0	1

Dot matrix led display

- Menampilkan *(display)* dari sebuah program yang dikirim secara paralel.
- Misalnya, 8 × 8 matriks LED

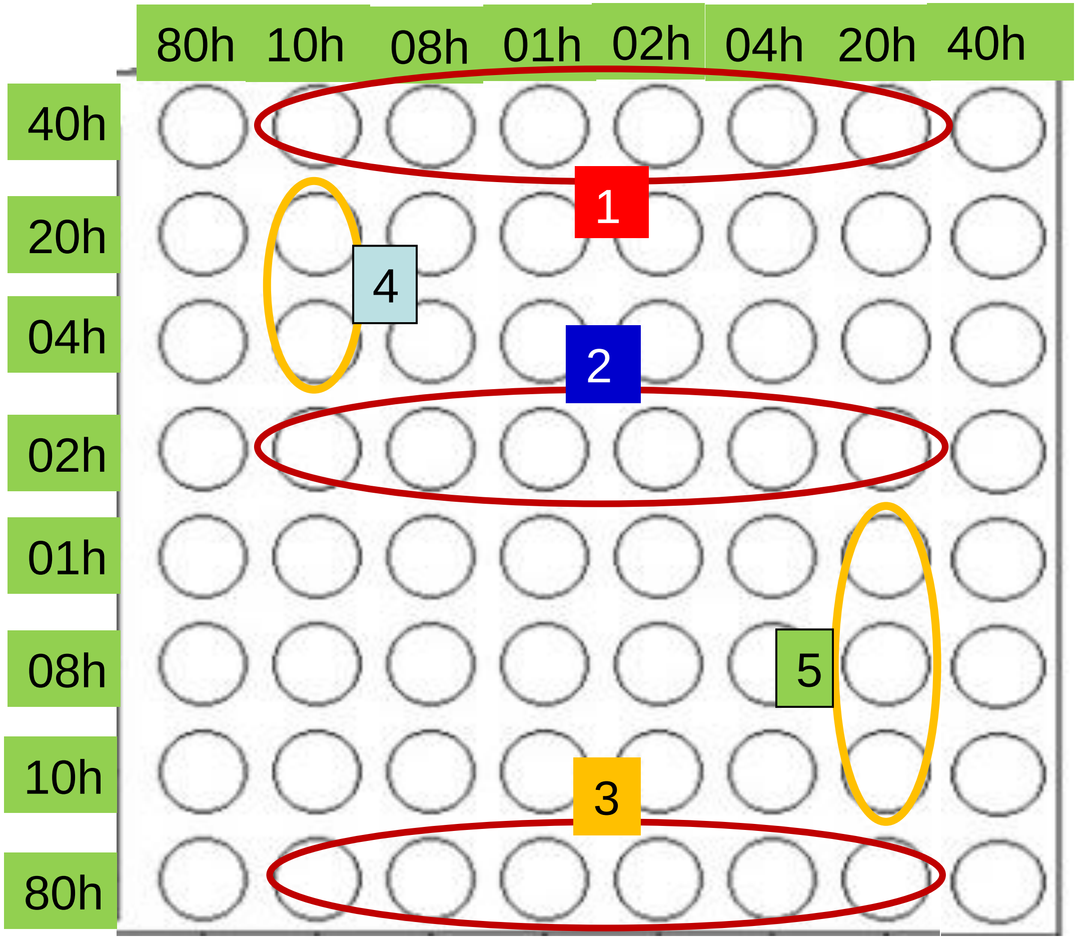
  Anoda bersama-sama dalam baris (R1 - R8), dan katoda dalam kolom (C1 - C8), jumlah yang diperlukan ***pin I /O*** **=** 16.

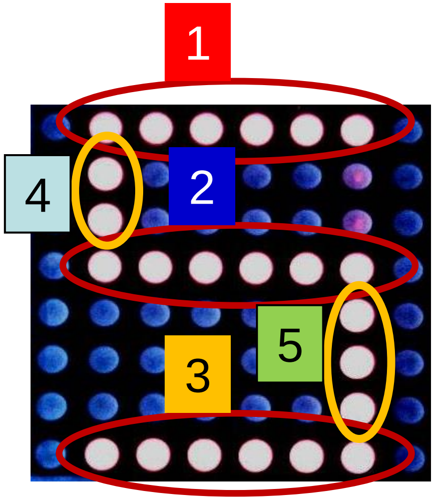




❑ Scanning

- LED dot matrix 8x8 (**Data Sheet**)





1

Col = 10+08+01+02+04+20=3Fh
Row = 40h

2

Col = 10+08+01+02+04+20=3Fh
Row = 02h

3

Col = 10+08+01+02+04+20=3Fh
Row = 80h

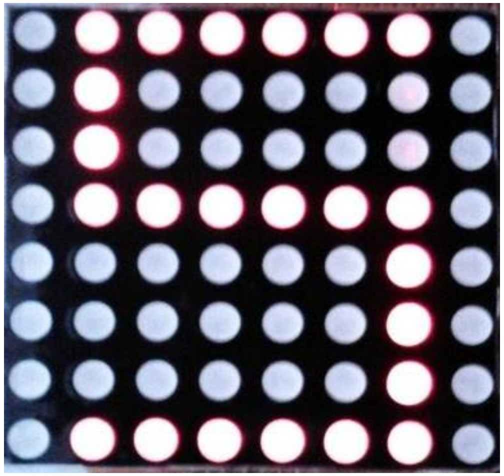
5

Col = 20h
Row = 01+08+10=19h

4

Col = 10h
Row = 20+04=24h

Contoh Program:

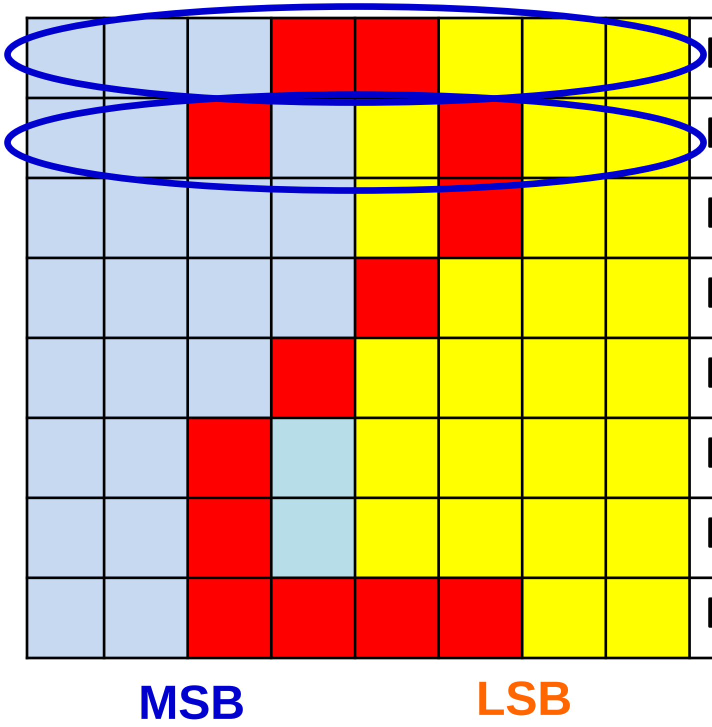


```
CEK4:   CJNE A,#10H,CEK5
LIMA:   MOV DPTR,#2000H
        MOV A,#3FH
        MOVX @DPTR,A
        MOV DPTR,#2001H
        MOV A,#40H
        MOVX @DPTR,A
        CALL  DELAY
MOV DPTR,#2000H
        MOV A,#3FH
        MOVX @DPTR,A
        MOV DPTR,#2001H
        MOV A,#02H
        MOVX @DPTR,A
        CALL  DELAY
MOV DPTR,#2000H
        MOV A,#3FH
        MOVX @DPTR,A
        MOV DPTR,#2001H
```

```
        MOV A,#80H
        MOVX @DPTR,A
        CALL  DELAY
MOV DPTR,#2000H
        MOV A,#10H
        MOVX @DPTR,A
        MOV DPTR,#2001H
        MOV A,#24H
        MOVX @DPTR,A
        CALL  DELAY
MOV DPTR,#2000H
        MOV A,#20H
        MOVX @DPTR,A
        MOV DPTR,#2001H
        MOV A,#19H
        MOVX @DPTR,A
        CALL  DELAY
        LJMP START
```

❑ Scanning

- LED dot matrix 8x8



Baris pertama menampilkan data pertama 18h

Baris kedua menampilkan data kedua 24h

Baris ketiga menampilkan data ketiga 04h

Baris keempat menampilkan data keempat 08h

Baris kelima menampilkan data kelima 20h

Baris keenam menampilkan data keenam 20h

Baris ketujuh menampilkan data ketujuh 20h

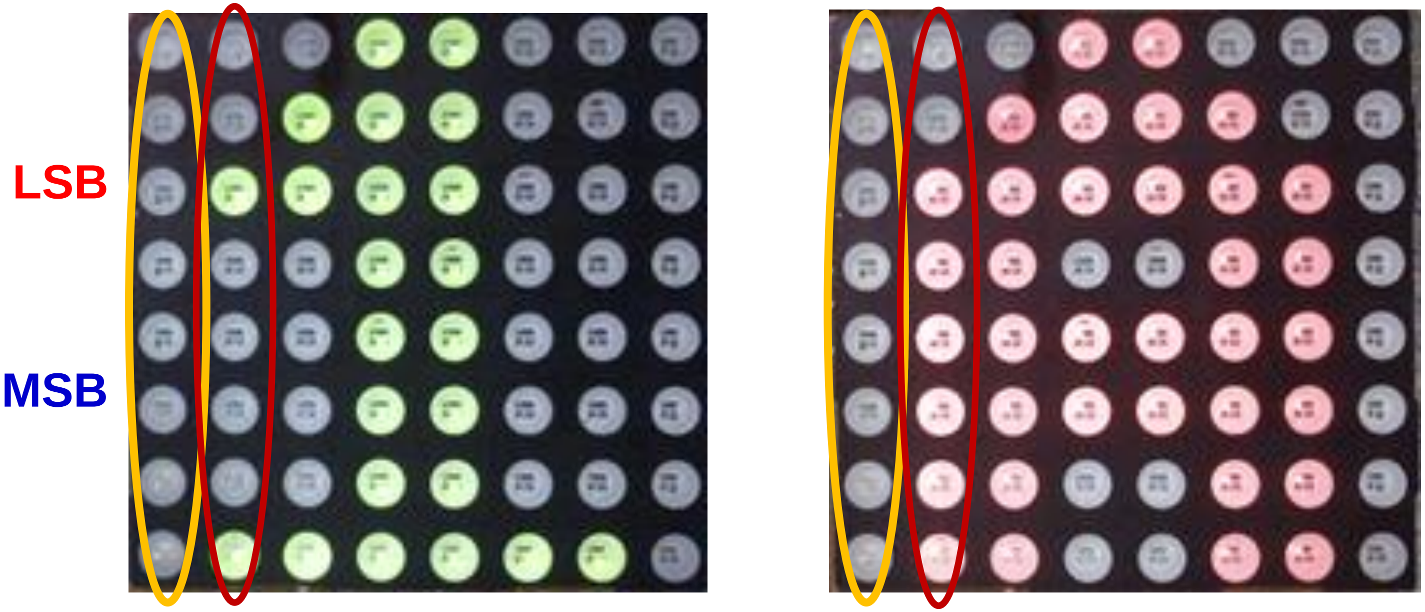
Baris kedelapan menampilkan data kedelapan 3Ch

Nilai data karakter dalam variabel array :

```
{{0x18,0x24,0x24,0x24,0x24,0x24,0x24,0x18}, // 0
 {0x08,0x18,0x08,0x08,0x08,0x08,0x08,0x1C}, // 1
 {0x18,0x24,0x04,0x08,0x10,0x20,0x20,0x3C}, // 2
 {0x18,0x24,0x04,0x18,0x04,0x04,0x24,0x18}, // 3
  dst}
```

❑ Scanning

- LED dot matrix 8x8 dua warna

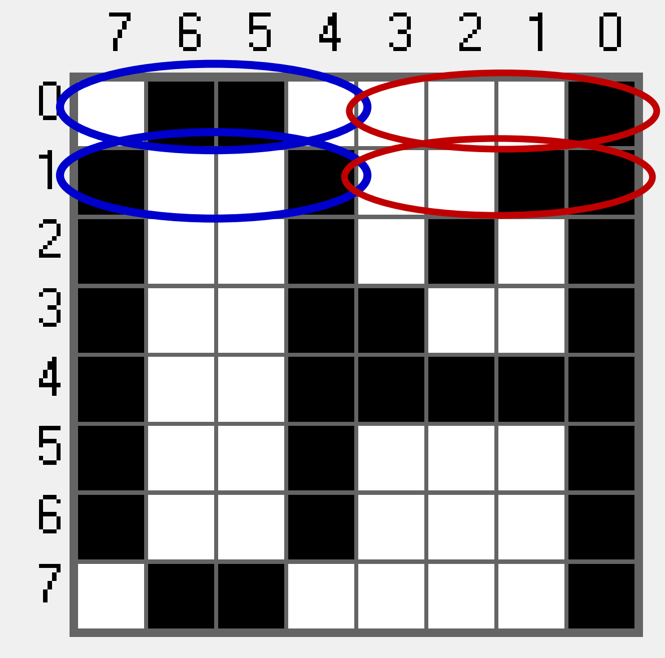


Nilai data karakter dalam variabel array :

```
{
   0x00, 0x84, 0x86, 0xFF, 0xFF, 0x80, 0x80, 0x00,// 1
   0x00, 0xFC, 0xFE, 0x37, 0x37, 0xFE, 0xFC, 0x0, // A
}
```

❑ Scanning

- LED dot matrix 8x8 (dua data)

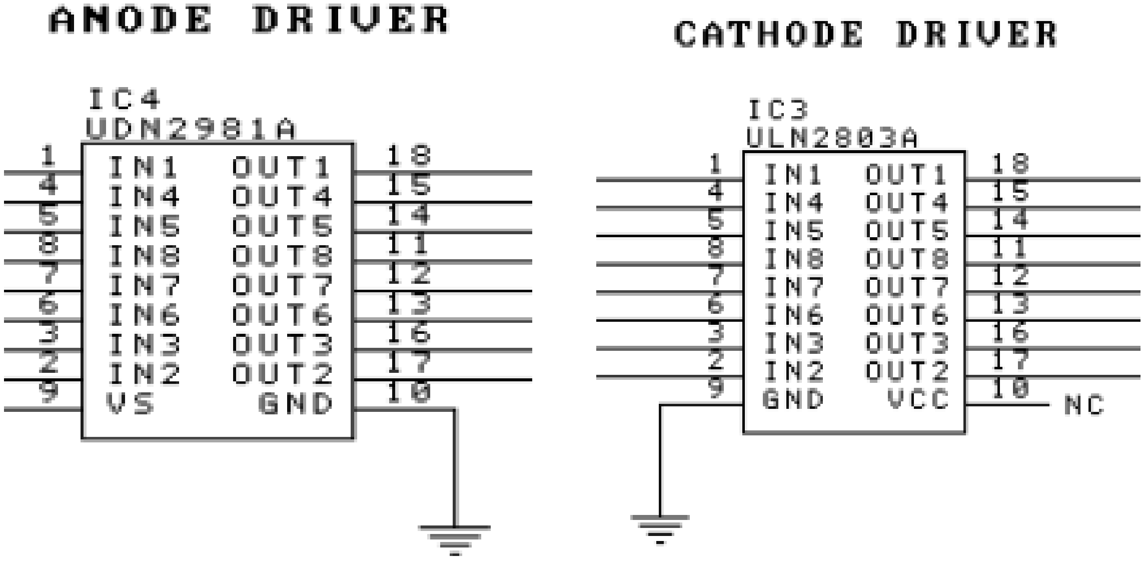


Nilai data karakter dalam variabel array :

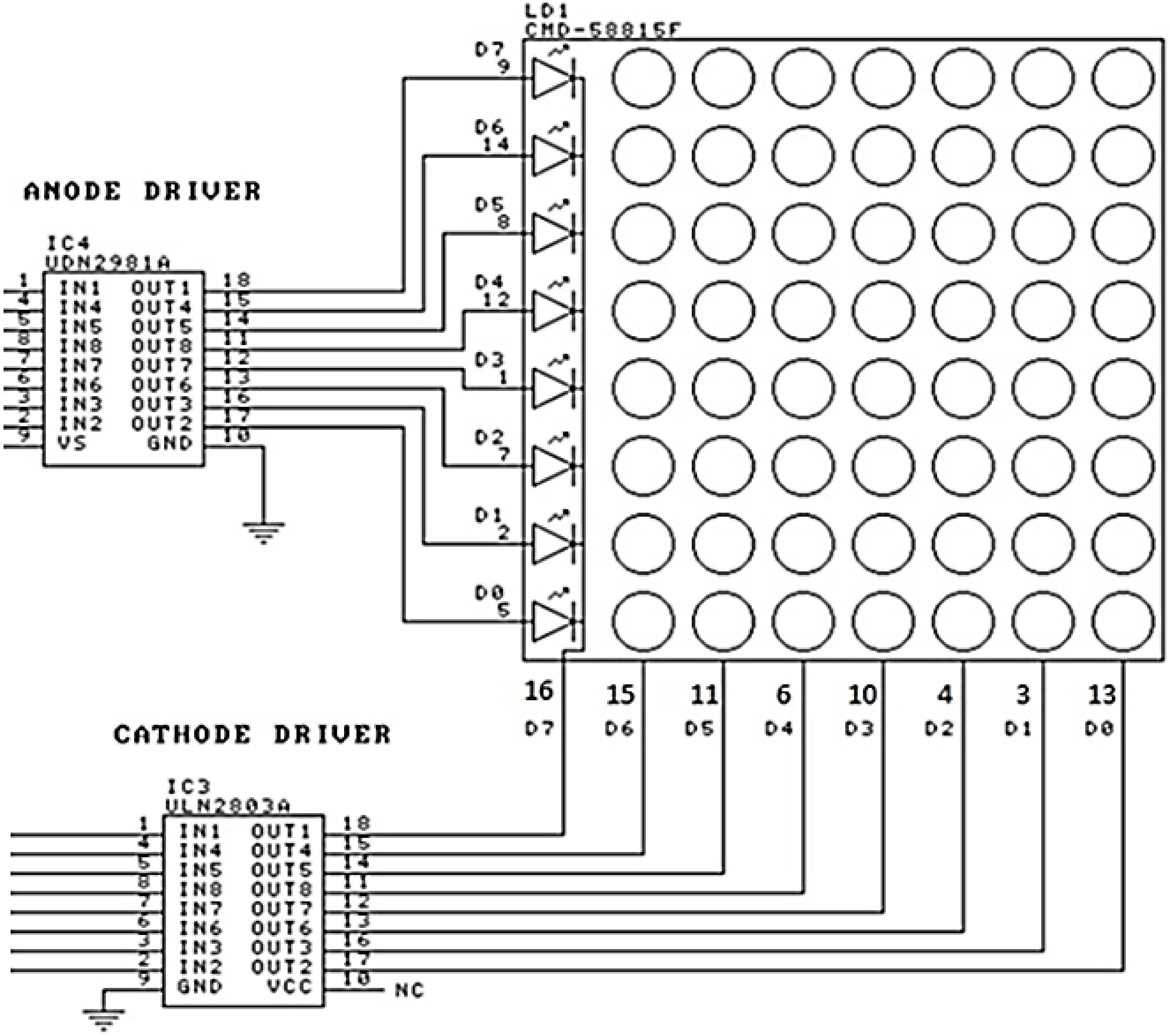
```
{
   0x06, 0x09, 0x09, 0x09, 0x09, 0x09, 0x09, 0x06, // 0
   0x01, 0x03, 0x05, 0x09, 0x0f, 0x01, 0x01, 0x01, // 4
}
```

Driver

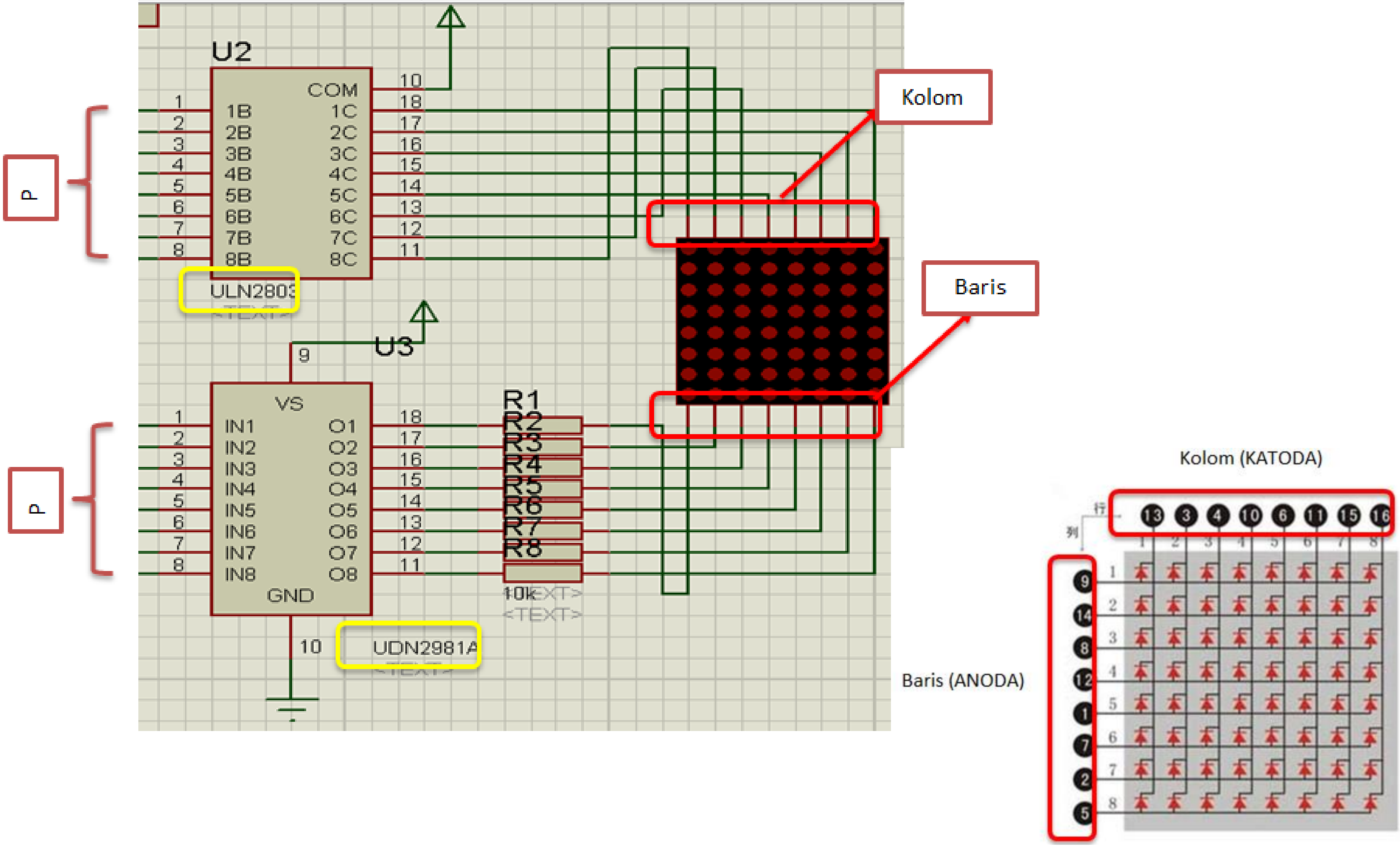
- IC driver : UDN2981A dan ULN2803A
- UDN2981A : driver ke transistor sekaligus mengatur anoda LED.
- ULN2803A : driver ke transistor sekaligus mengatur katoda LED.



Rangkaian Interface



Rangkaian Interface



P

P

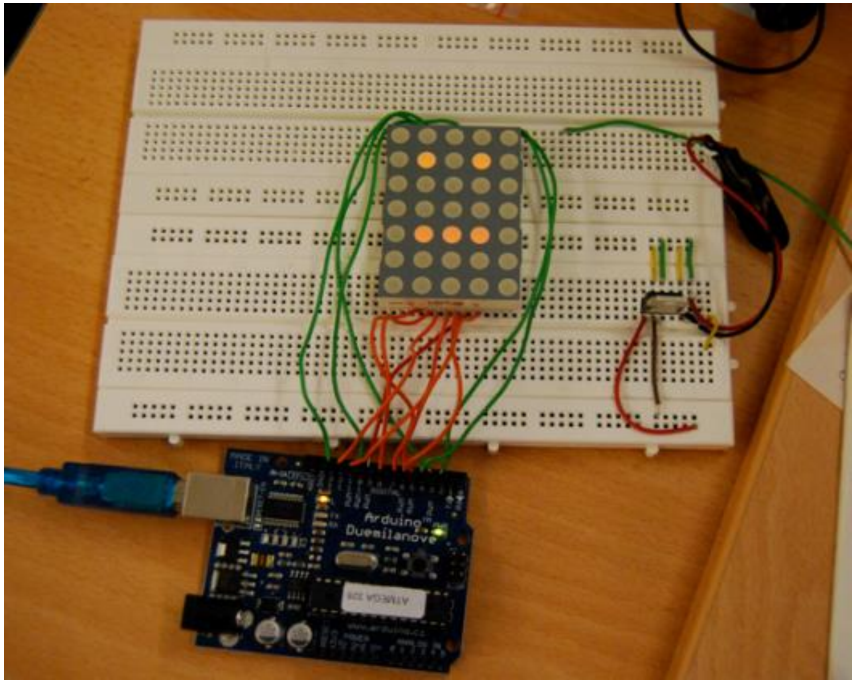
Kolom

Baris

Kolom (KATODA)

Baris (ANODA)

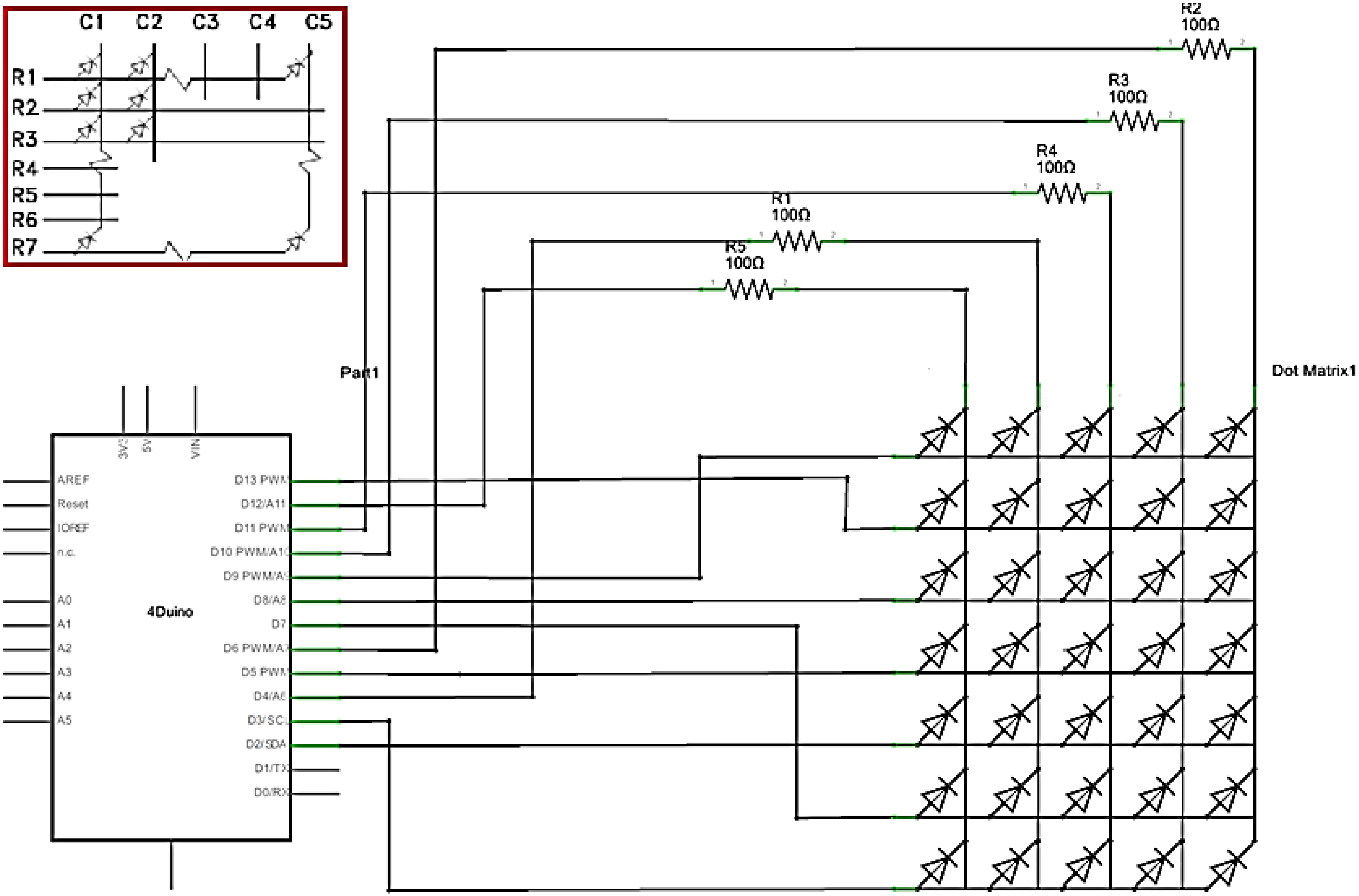
Rangkaian Interface Dot Matrik 5x7 dengan Arduino



Hardware :

- Arduino Uno Board
- 1x Breadboard
- 1x LED Dot Matrik 5x7
- Kabel jumper

Rangkaian Interface Dot Matrik 5x7 dengan Arduino



PROGRAM 1:

```
int idx = 0; unsigned long last;
void setup() {
  last = millis();
//Kolom
  pinMode( 9, OUTPUT );
  pinMode( 10, OUTPUT );
  pinMode( 11, OUTPUT );
  pinMode( 12, OUTPUT );
  pinMode( 13, OUTPUT );
//Baris
  pinMode( 2, OUTPUT );
  pinMode( 3, OUTPUT );
  pinMode( 4, OUTPUT );
  pinMode( 5, OUTPUT );
  pinMode( 6, OUTPUT );
  pinMode( 7, OUTPUT );
  pinMode( 8, OUTPUT );
for( int r = 0; r < 7; r++) {
     digitalWrite( r + 2, LOW );
   }
   for( int c = 0; c < 5; c++) {
     digitalWrite( c + 9, HIGH);
   }
}

byte leds[7][5];
```

PROGRAM 1:

```
void setPattern( byte pattern[20][5], int idx ) {
 for( int r =0; r < 7; r++) {
  for( int c = 0; c < 5; c++) {
   leds[r][c] = pattern[r + idx][c];
   }
  }
}

void draw() {

 for( int r =0; r < 7; r ++ ) {
  digitalWrite( r + 2, HIGH );
  for( int c=0; c < 5; c ++ ) {
    digitalWrite( 13 - c, ( leds[r][c] == 1 ? LOW : HIGH ));
}
   delayMicroseconds(900);
   digitalWrite( r + 2, LOW );
 }
}

void loop() {
 if ( millis() - last > 400 ) {
  idx = (idx == 0 ?  7 : 0);
  last = millis();
 }
```

PROGRAM 1:

```
byte tmp[14][5] = {
  { 0,0,0,0,0},
  { 0,1,0,1,0},
  { 0,0,0,0,0},
  { 0,1,1,1,0},
  { 0,1,0,1,0},
  { 0,1,1,1,0},
  { 0,0,0,0,0},
  { 0,0,0,0,0},
  { 0,1,0,1,0},
  { 0,0,0,0,0},
  { 0,0,0,0,0},
  { 0,1,1,1,0},
  { 0,0,0,0,0},
  { 0,0,0,0,0},
  };

  setPattern( tmp, idx );
  draw();
}
```

PROGRAM 2:

```
#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} \
}
```

PROGRAM 2:

```
#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]; // columns x rows

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

int rows[7] = {pins[12], pins[11], pins[2], pins[9], pins[4],
pins[5], pins[6]};
```

Ctt: program-2 tidak sama dengan rangkaian di program-1

PROGRAM 2:

```
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);
}
```

PROGRAM 2:

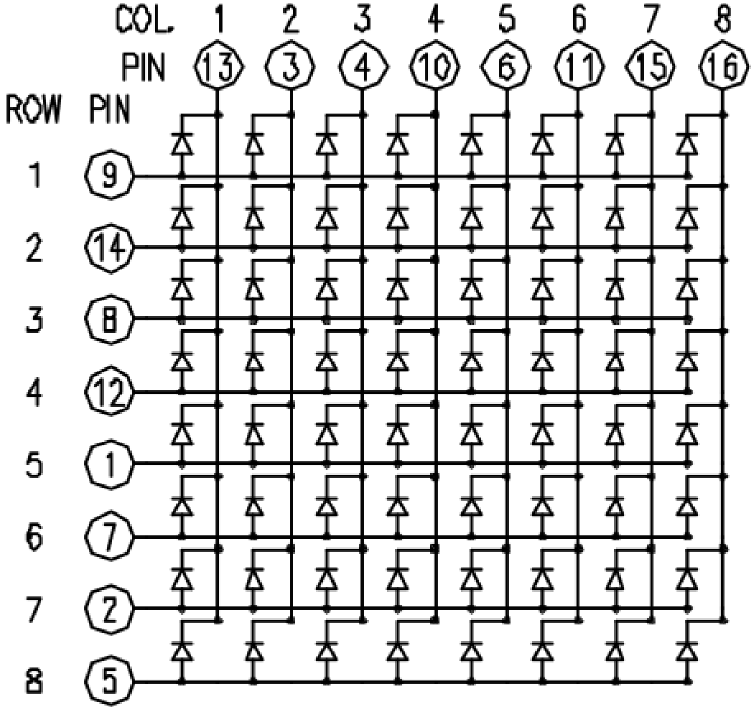
```
void clearLeds()
{
  // Clear display array
  for (int i = 0; i < 5; i++) {
    for (int j = 0; j < 7; j++) {
      leds[i][j] = 0;
    }
  }
}

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];
    }
  }
}
```

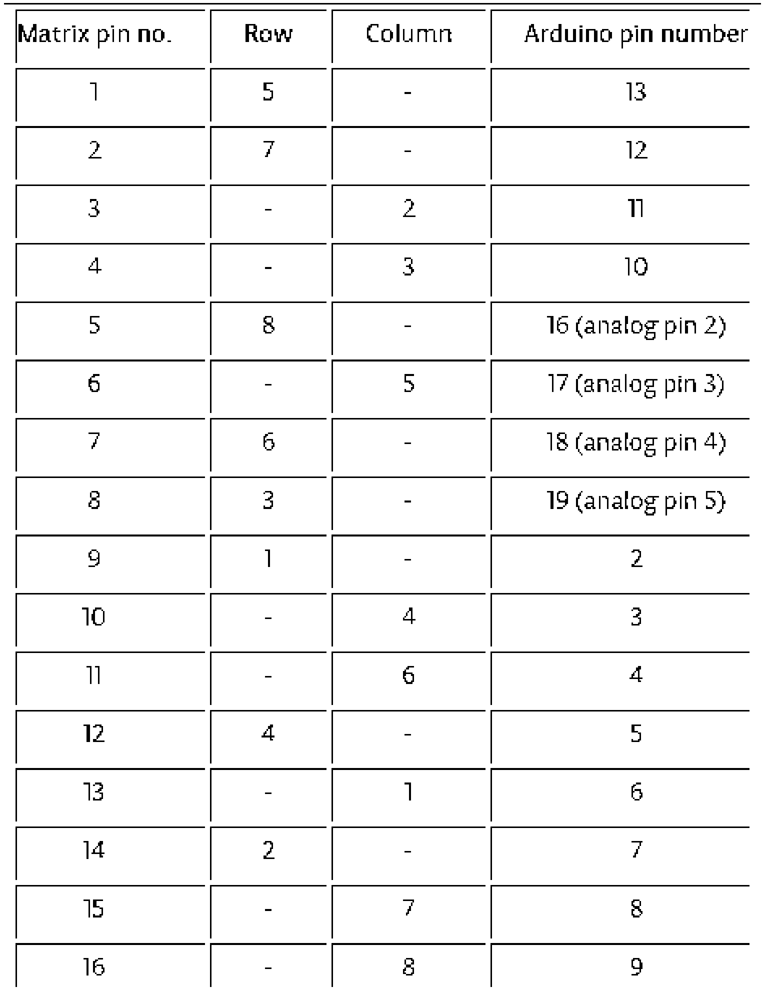
```
void slidePattern(int pattern, int del)
{
  for (int newcol = 0; newcol <= 4; newcol++)
{
    // shift the first 4 columns left
    for (int row = 0; row <= 6; row++)
      for (int col = 0; col <= 3; col++)
        leds[col][row] = leds[col+1][row];

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

    delay(del);
  }
}
```

PROGRAM 2:

```
void display()
{
  // Turn whole previous column off:
  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); //
Turn on this led
    }
    else {
      digitalWrite(rows[row], HIGH); //
Turn off this led
    }
  }
digitalWrite(cols[col], HIGH);
}
```

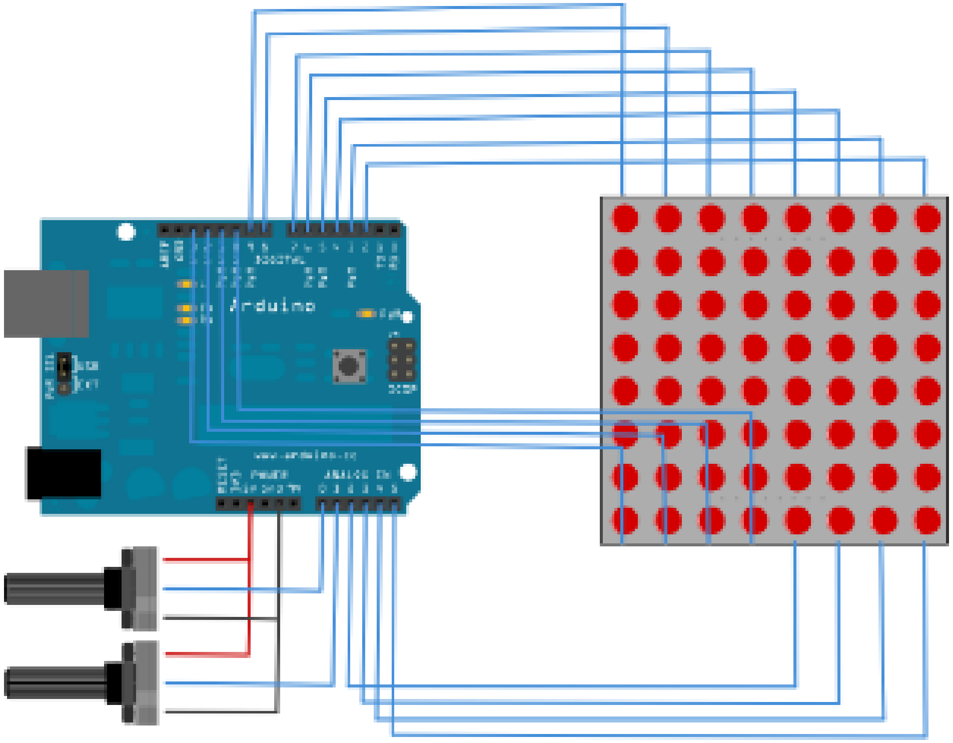
Rangkaian Interface Dot Matrik 8x8 dengan Arduino





Matrix pin no.	Row	Column	Arduino pin number
1	5	-	13
2	7	-	12
3	-	2	11
4	-	3	10
5	8	-	16 (analog pin 2)
6	-	5	17 (analog pin 3)
7	6	-	18 (analog pin 4)
8	3	-	19 (analog pin 5)
9	1	-	2
10	-	4	3
11	-	6	4
12	4	-	5
13	-	1	6
14	2	-	7
15	-	7	8
16	-	8	9

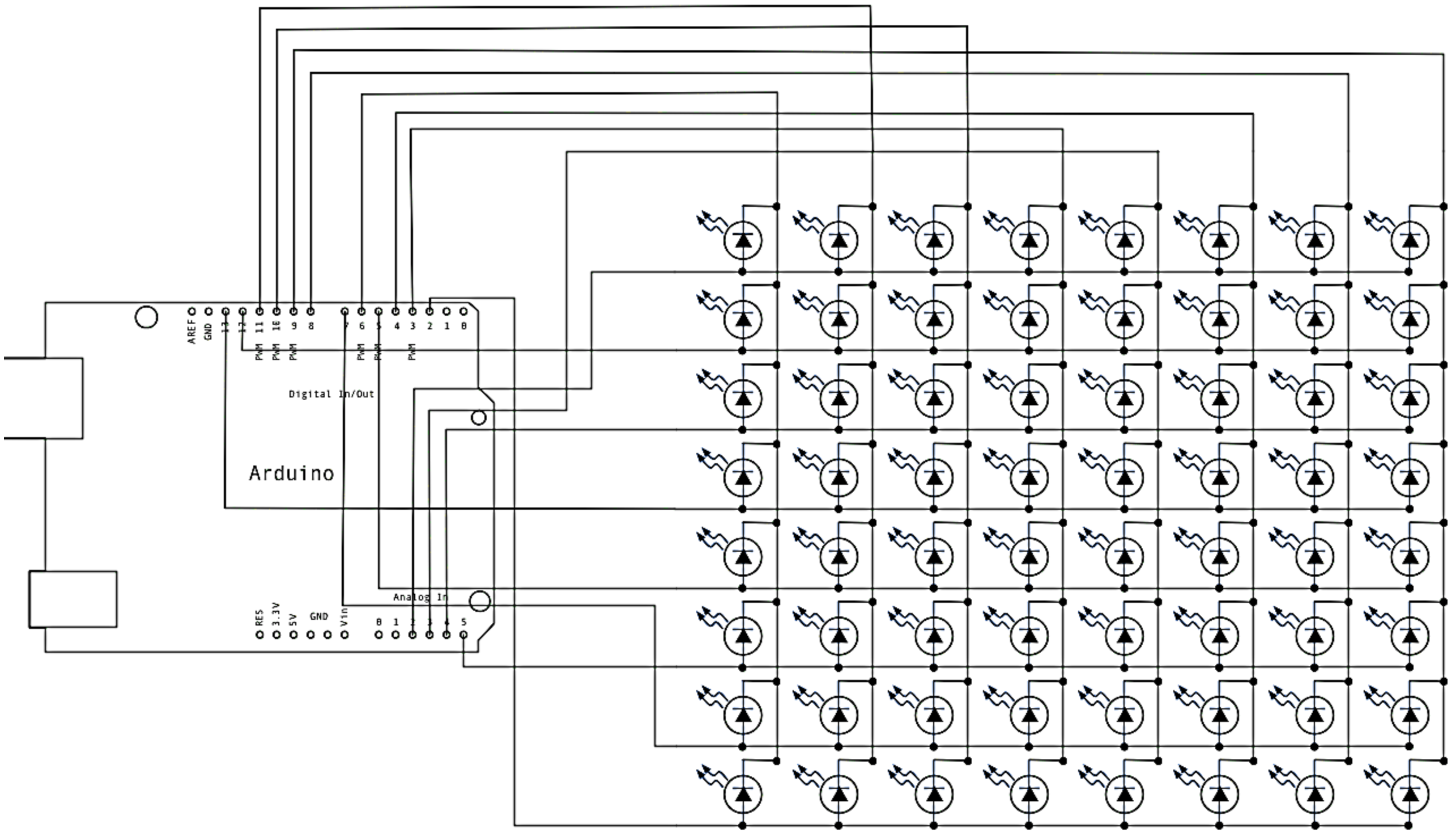
Rangkaian Interface Dot Matrik 8x8 dengan Arduino



Hardware :

- Arduino Uno Board
- 1x Breadboard
- 1x LED Dot Matrik 8x8
- 2x potensiometer 10K
- Kabel jumper

Rangkaian Interface Dot Matrik 8x8 dengan Arduino



PROGRAM : Metode scanning

```
const int row[8] = {  2, 7, 19, 5, 13, 18, 12, 16 };
const int col[8] = {  6, 11, 10, 3, 17, 4, 8, 9 };

int pixels[8][8];
int x = 5;
int y = 5;
void setup() {
    for (int thisPin = 0; thisPin < 8; thisPin++) {
    pinMode(col[thisPin], OUTPUT);
    pinMode(row[thisPin], OUTPUT);
    digitalWrite(col[thisPin], HIGH);
  }
  for (int x = 0; x < 8; x++) {
    for (int y = 0; y < 8; y++) {
      pixels[x][y] = HIGH;
    }
  }
}
void loop() {
 readSensors();
 refreshScreen();
}
```

PROGRAM : *Metode scanning*

```
void readSensors() {
 pixels[x][y] = HIGH;
 x = 7 - map(analogRead(A0), 0, 1023, 0, 7);
 y = map(analogRead(A1), 0, 1023, 0, 7);
 pixels[x][y] = LOW;
}
void refreshScreen() {
  for (int thisRow = 0; thisRow < 8; thisRow++) {
  digitalWrite(row[thisRow], HIGH);
  for (int thisCol = 0; thisCol < 8; thisCol++) {
  int thisPixel = pixels[thisRow][thisCol];
  digitalWrite(col[thisCol], thisPixel);
  if (thisPixel == LOW) {
        digitalWrite(col[thisCol], HIGH);
      }
    }
  digitalWrite(row[thisRow], LOW);
  }
}
```

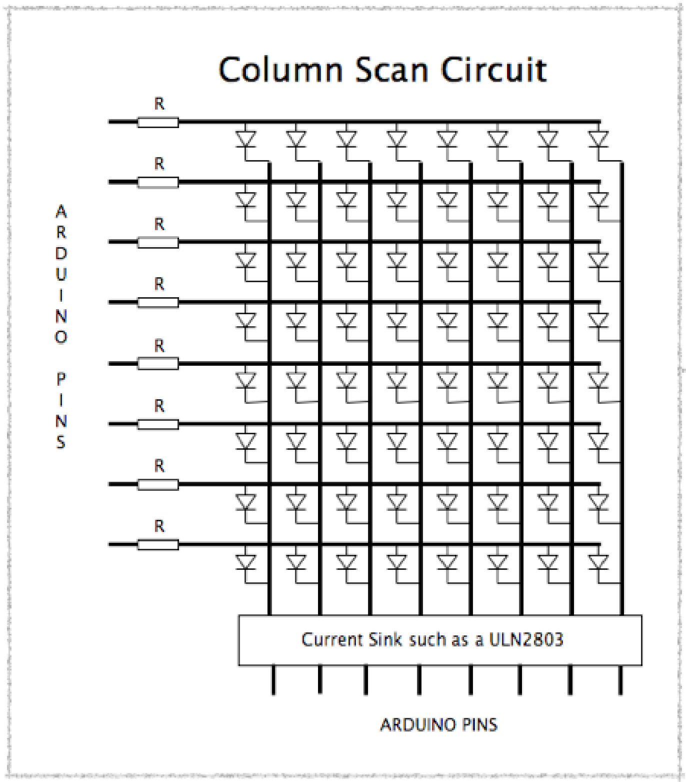
Hasil :

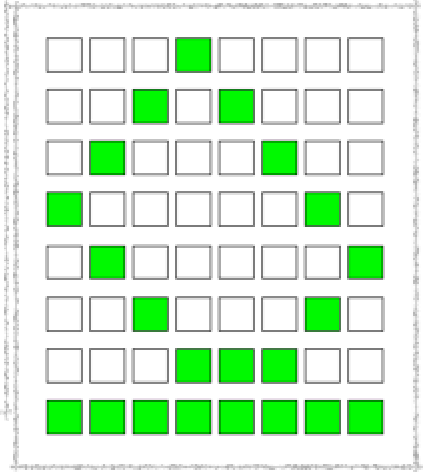
- Catat hasil data yang tampil di LED Dot Matrik
- Buat laporan hasil dari percobaan

Latihan :

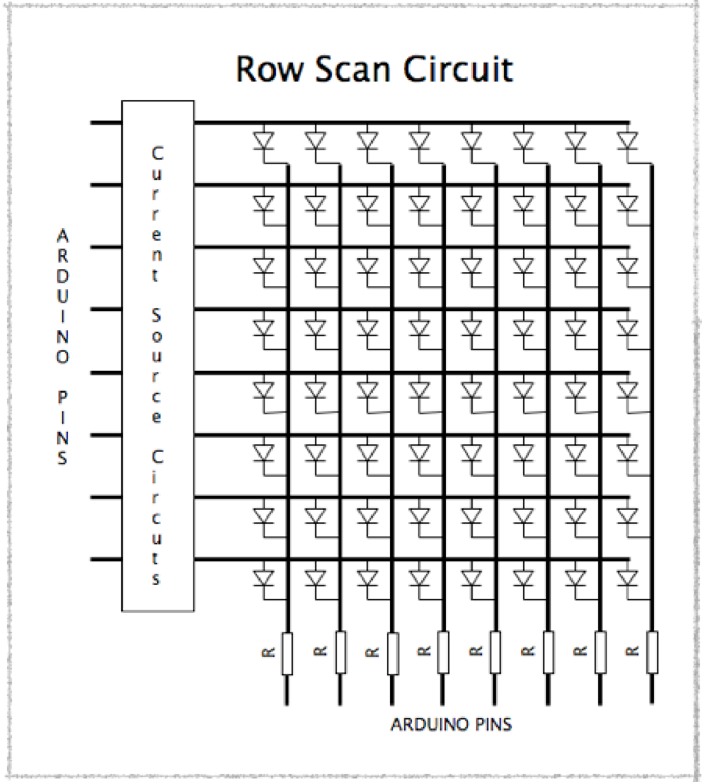
1. Buatlah program untuk tulisan (huruf A) berjalan dari kiri ke kanan.
2. Buatlah program untuk menampilkan 2,T,A,B
3. Buatlah program satu angka paling belakang dari NRP anda menggunakan metode scanning
4. Buatlah program untuk menampilkan karakter menggunakan IC Driver

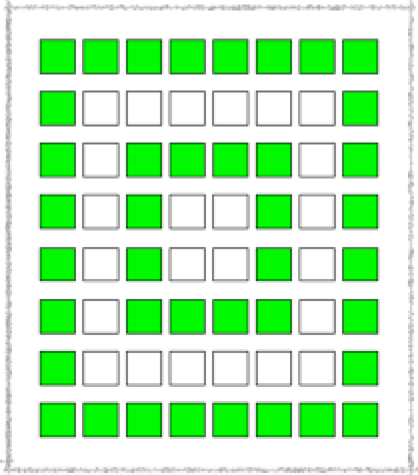
Latihan :


Column Scan Circuit
R
R
R
R
R
R
R
R
ARDUINO PINS
Current Sink such as a ULN2803
ARDUINO PINS



□□□🟩□□□□
□□🟩□🟩□□□
□🟩□□□🟩□□
🟩□□□□□🟩□
□🟩□□□□□🟩
□□🟩□□□🟩□
□□□🟩🟩🟩□□
🟩🟩🟩🟩🟩🟩🟩🟩


Row Scan Circuit
ARDUINO PINS
Current Source Circuits
R
R
R
R
R
R
R
R
ARDUINO PINS



🟩🟩🟩🟩🟩🟩🟩🟩
🟩□□□□□□🟩
🟩□🟩🟩🟩🟩□🟩
🟩□🟩□□🟩□🟩
🟩□🟩□□🟩□🟩
🟩□🟩🟩🟩🟩□🟩
🟩□□□□□□🟩
🟩🟩🟩🟩🟩🟩🟩🟩

Latihan :

```
int rowCount = 0;
byte rowPin[ ] = { 2, 3, 4, 5, 6, 7, 8, 9};
byte colPin[ ] = { 10, 11, 12, 13, 14, 15, 16, 17};
byte displayData [ ] = { B11111111, B10000001, B10111101, B10100101, B10100101, B10111101
B10000001, B11111111 }; // display pattern

void setup(){
for(int i = 0;  i<8; i++){
   pinMode(rowPin[i], OUTPUT);
   pinMode(colPin[i], OUTPUT);
  }
} // end of setup

void refresh(){
byte tempData;
digitalWrite(colPin[rowCount], LOW);
rowCount ++;
if (rowCount == 8) rowCount = 0;
tempData = displayData[rowCount];
for(int i=0 ; i<8; i++){
   if((tempData & 1) == 0) digitalWrite(rowPin[i], LOW); else digitalWrite(rowPin[i], HIGH);
   }
digitalWrite(colPin[rowCount], HIGH);
}
```