

به نام خدا

عنوان آزمایش: شکلک (دات ماتریکس)

وسایل مورد نیاز: برد Arduino UNO R3 ، کابل رابط، محیط توسعه آردینو (Arduino IDE)، سیم رابط، یک عدد دات ماتریکس، برد بورد

هدف آزمایش: نمایش شکلک خنده و گریه و... در دات ماتریکس

شرح آزمایش: به ترتیب پایه‌های زمین، 5 ولت، DIN ، CLK و CS دات ماتریکس را به پایه‌های زمین، 5 ولت، 7، 6 و 5 آردینو وصل می‌کنیم.

سپس آردینو را با کابل رابط مناسب به کامپیوتر متصل می‌کنیم. محیط توسعه آردینو را اجرا می‌کنیم. بعد از آن تنظیمات اولیه و لازم نرم‌افزار را به شرح زیر اعمال می‌کنیم:

۱- ابتدا از نوار ناوبری بالا گزینه tools را انتخاب می‌کنیم. از منوی ظاهر شده، board را روی گزینه Arduino/Genuino Uno قرار می‌دهیم.

۲- سپس در همان منو، port مناسب را انتخاب می‌کنیم (پورتهی که مدار را به آن متصل کردیم).

۳- بعد از آن در همان منو گزینه programmer را روی Arduino as ISP قرار می‌دهیم.

حالا برنامه قابل تست و اجرا است.

کد اجرای آن، که در ادامه می‌آید، را نوشته و روی گزینه compile کلیک می‌کنیم.

بعد از کامپایل شدن برنامه، روی گزینه upload کلیک می‌کنیم تا برنامه روی بورد بارگذاری شود.

عملکرد مدار به این صورت است که شکلک smile و sad و poker و kiss به ترتیب و با فاصله زمانی روی دات ماتریکس به نمایش در می‌آید و همزمان از سمت راست به سمت چپ حرکت می‌کند.

کد ابتدایی آن به صورت زیر است:

```
#include <MaxMatrix.h>
#include <avr/pgmspace.h>

int DIN = 7;
int CLK = 6;
int CS = 5;
int maxInUse = 1;

MaxMatrix m(DIN, CS, CLK, maxInUse);
byte buffer[10];

const byte smile[] = {8,8,
    B00111100,
    B01000010,
    B10100101,
    B10000001,
    B10100101,
    B10011001,
    B01000010,
    B00111100
};

const byte sad[] = {8,8,
    B00111100,
    B01000010,
    B10100101,
    B10000001,
    B10011001,
    B10100101,
    B01000010,
    B00111100
};

const byte poker[] = {8,8,
    B00111100,
    B01000010,
    B10100101,
    B10000001,
    B10000001,
    B10100101,
    B01000010,
    B00111100
};
```

```
};

const byte kiss[] = {8,8,
    B00111100,
    B01000010,
    B10100101,
    B10011001,
    B10111101,
    B10011001,
    B01000010,
    B00111100
};

void setup() {
    m.init();
    m.setIntensity(5);
}

void scrollSprite(const byte* sprite) {
    m.writeSprite(8, 0, sprite);
    for (int i = 0; i < 8 + sprite[0]; i++) {
        m.shiftLeft(false, false);
        delay(100);
    }
    m.clear();
}

void loop() {
    scrollSprite(smile);
    scrollSprite(sad);
    scrollSprite(poker);
    scrollSprite(kiss);
}
```