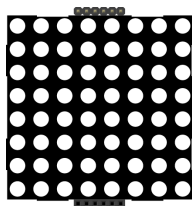


Maticový LED displej - III

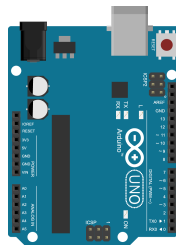
PŮVODNÍ ZAPOJENÍ MATICOVÉHO LED DISPLEJE ROZŠÍŘÍTE O PŘIPOJENÍ AKCELEROMETRU. V ZÁVISLOSTI NA POLOZE BUDE POSKYTOVAT DATA PRO POZICI ROZSVÍCENÉ DIODY NA MATICOVÉM DISPLEJI.

Sestavení obvodu

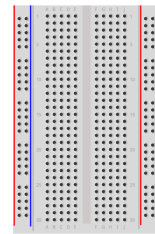
Co budeme potřebovat?



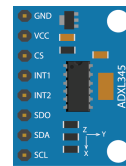
Maticový displej 8x8



Deska Arduino



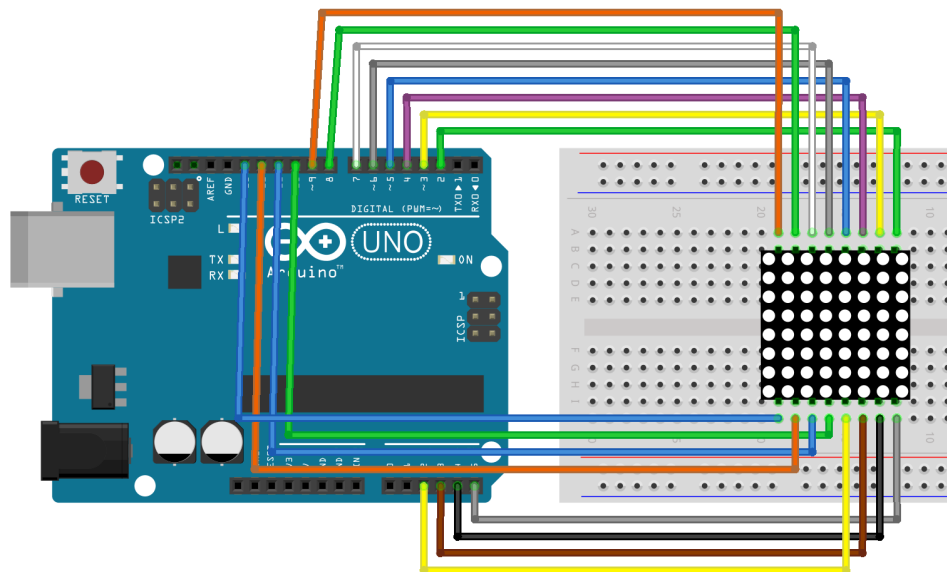
Kontaktní pole



Akcelerometr

Elektronický obvod

Schéma zapojení



Otázky pro vás

Víte kde se můžete setkat se zařízením
akcelerometr?

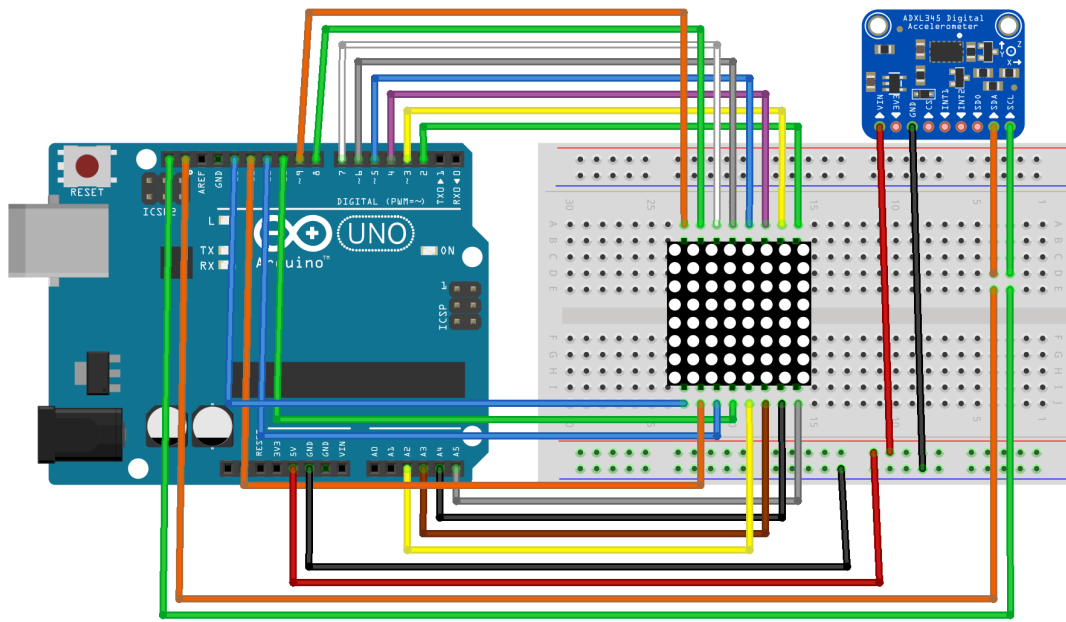
V dnešní době má akcelerometr takřka každý
mobilní telefon. Dále jej nalezneme
v automobilech, letadlech apod.

Víte, co akcelerometr měří?

Měří pohybové zrychlení, a to nejlépe ve všech
třech osách.



Zapojení akcelerometru



Krátce o akcelerometru

Akcelerometr je malé pohybové čidlo, které měří pohybové zrychlení a to nejlépe ve všech třech osách.

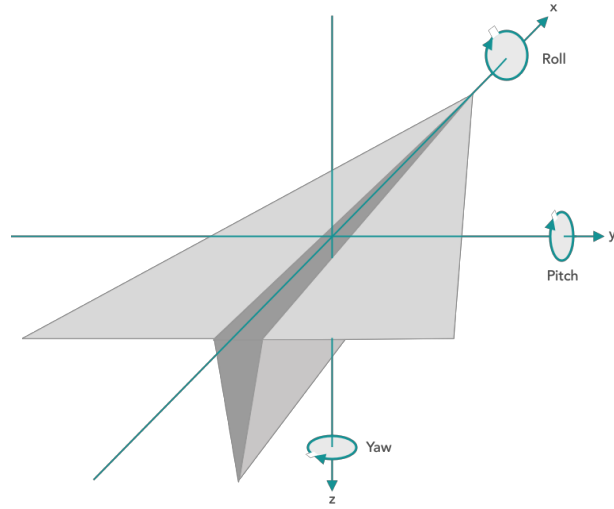
Ze znalosti zrychlení a hmotnosti lze zjistit sílu působící na těleso.

Akcelerometry jsou vhodné nejen pro měření odstředivých a setrvačných sil, ale i pro určování pozice tělesa, jeho náklon nebo vibrace.

Akcelerometry jsou dnes i v mobilních telefonech a využívají se v leteckém a automobilovém průmyslu.



Krátce o akcelerometru



$$pitch = \arctan\left(\frac{G_y}{\sqrt{G_x^2 + G_z^2}}\right)$$

$$roll = \arctan\left(\frac{-G_x}{G_z}\right)$$



Arduino a akcelerometr

Vzorce lze v Arduino kódu přepsat v následujícím tvaru:

```
roll  = (atan2(-Yg, Zg)*180.0)/M_PI;  
pitch = (atan2(Xg, sqrt(Yg*Yg + Zg*Zg))*180.0)/M_PI;
```

Připojení knihoven pro práci s akcelerometrem je následující:

```
#include <Wire.h>  
#include <ADXL345.h>
```

```
acc.read(&Xg, &Yg, &Zg);
```



Úkol pro vás

Inovujte programový kód tak, abyste aplikovali vzorec pro výpočet úhlů roll a pitch. Nezapomeňte definovat všechny proměnné.

```
#include <Wire.h>
#include <ADXL345.h>

ADXL345 acc;

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(){
    acc.begin();

    for(int i = 0; i < 8; i++){
        pinMode(col[i], OUTPUT);
        pinMode(row[i], OUTPUT);
        digitalWrite(row[i], LOW);
    }

    for(int x = 0; x < 8; x++) {
        for(int y = 0; y < 8; y++) {
            pixels[x][y] = HIGH;
        }
    }
}
```

```
void loop(){
    readSensors();
    refreshScreen();
}

void readSensors(){
    double pitch, roll, Xg, Yg, Zg;
    acc.read(&Xg, &Yg, &Zg);

    roll = (atan2(-Yg, Zg)*180.0)/M_PI;
    pitch = (atan2(Xg, sqrt(Yg*Yg + Zg*Zg))*180.0)/M_PI;

    pixels[x][y] = HIGH;
    x = 7 - map(roll, -20, 20, 0, 7);
    y = map(pitch, -20, 20, 0, 7);
    pixels[x][y] = LOW;
}

void refreshScreen(){
    for(int j = 0; j<8;j++){
        digitalWrite(row[j], HIGH);
        for(int k = 0; k<8; k++){
            int thisPixel = pixels[j][k];
            digitalWrite(col[k], thisPixel);
            if (thisPixel == LOW) {
                digitalWrite(col[k], HIGH);
            }
        }
        digitalWrite(row[j], LOW);
    }
}
```

