

1. Harjoitus osa 1

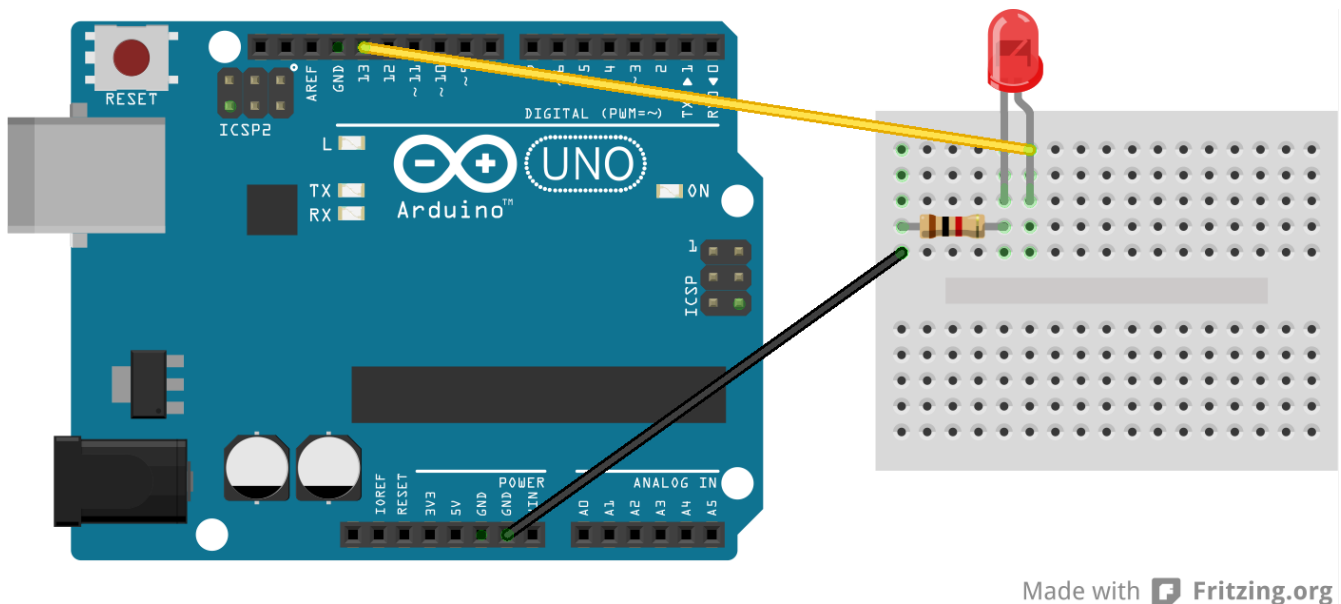
Harjoituksen tavoitteena tutustua ledin ohjaamiseen arduinolla.

Tavoitteet:

- Kytke ledi arduinoon
- Ohjelmoimaan Arduinoa käyttäen Arduino kehitystyökalua.
- Ohjataan ledin syttymistä/sammumista ohjelmallisesti

Toteutus

Ledin virransyöttö tulee ohjausjännitteestä, joka tässä tapauksessa on DIGITAL port 13. Ledi maadoitetaan vastuksen kautta maahan (GROUND=GND) kuvan 1 osoittamalla tavalla.



Kuva 1.

Koodi

// the setup function runs once when you press reset or power the board

```
void setup() {
```

```
  // initialize digital pin 13 as an output.
```

```
  pinMode(13, OUTPUT);
```

```
}
```

// the loop function runs over and over again forever

```
void loop() {
```

```
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
```

```
  delay(1000);           // wait for a second
```

```
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
```

```
  delay(1000);           // wait for a second
```

```
}
```

Ideoita

- Lisätään nappula jolla ohjataan ledin syttymistä
- Lisätään useita ledejä ja tehdään esimerkiksi Binäärilaskin
- Muita?

Oheismateriaali

Viralliset Arduino -esimerkit:

<http://arduino.cc/en/tutorial/blink>

Apua ensimmäiseen kokeiluun

- Asenna Arduino työkalut tietokoneeseen
 - <http://arduino.cc/en/Main/Software>
- Avaa laatikko (Ota arduino levy poydalle)
- Kytke usb -johto arduinon ja PC:n valiin
- Kaynnista Arduino sovellus PC:lla

Ensimmäinen sovellus

- Valitse File/Examples/01.Basics/Blink
- Ohjelmakoodi avautuu uuteen Arduino ikunaan.
- Lataa koodi Arduinoon valitsemalla File/Upload
- Arduino levyllä alka ledi vilkkua(, jos kaikki meni hyvin)

1. Harjoitus osa 2

Harjoituksen tavoitteena tutustua Kirjastoihin ja Arduinon ohjelmointiin Morsetus-luokan avulla.

Tavoitteet:

- Luodaan Arduino ohjelma jossa dash()- ja dot() -funktiot. Dash()- funktio esittää ledillä morsen viivan. Ja dot()-funktio morsetuksen pisteen.
- Luodaan Morse luokka, joka sijoitetaan libraries -hakemistoon. Ja tehdään arduino koodi, jolla kyseistä luokkaa käytetään.

Toteutus – 1. vaihe

Sovelluksen testaamiseen voidaan käyttää kuvan 1 mukaista kytkentää. (Ledin virransyöttö tulee ohjausjännitteestä, joka tässä tapauksessa on DIGITAL port 13. Ledi maadoitetaan vastuksen kautta maahan (GROUND=GND) kuvan 1 osoittamalla tavalla.)

Luo Arduinoon koodi, jossa funktiot dash() ja dot(). Lisäksi tarvitaan tavanomaiset setup() ja loop()-funktiot. Ohjelman pitäisi morsettaa lediä vilkuttamalla viestiö SOS.

```
int pin = 13;
```

```
void setup()
{
  pinMode(pin, OUTPUT);
}
```

```
void loop()
{
  dot(); dot(); dot();
  dash(); dash(); dash();
  dot(); dot(); dot();
  delay(3000);
}
```

```
void dot()
{
  digitalWrite(pin, HIGH);
  delay(250);
  digitalWrite(pin, LOW);
  delay(250);
}
```

```
void dash()
{
  digitalWrite(pin, HIGH);
  delay(1000);
  digitalWrite(pin, LOW);
}
```

```
    delay(250);  
}
```

Toteutus – 2. Vaihe

Toteutetaan edellinen sovellus käyttäen luokkaa Morse, siten että dash() ja dot() -funktiot sijoitetaan edellä mainittuun luokkaan.

- Luo hakemisto **Morse** hakemistoon arduino/libraries/
- Luo Morse -hakemistoon tiedosto **Morse.h**, joka on luokan toiminnot määrittelevä otsikkotiedosto, eli header.
- Luo Morse -hakemistoon toinen tiedosto **Morse.cpp**, joka toteuttaa otsikkotiedoston määrittelemät ominaisuudet.
- Kun tiedostot on luotu, niin Arduino-työkalusta valitaan Sketch/ Import Library... / Add Library , jolla valitaa edellä luotu Morse-hakemisto. Valinnan jälkeen Morse näkyy Sketch / Import Library / -valikon alimmaisena.
- Alla Tiedostojen koodit:

```
//Morse.h  
  
#ifndef Morse_h  
#define Morse_h  
  
#include "Arduino.h"  
  
class Morse  
{  
public:  
    Morse(int pin);  
    void dot();  
    void dash();  
private:  
    int _pin;  
};  
  
#endif
```

```
//Morse.cpp  
#include "Arduino.h"  
#include "Morse.h"
```

```
Morse::Morse(int pin)
{
  pinMode(pin, OUTPUT);
  _pin = pin;
}

void Morse::dot()
{
  digitalWrite(_pin, HIGH);
  delay(250);
  digitalWrite(_pin, LOW);
  delay(250);
}

void Morse::dash()
{
  digitalWrite(_pin, HIGH);
  delay(1000);
  digitalWrite(_pin, LOW);
  delay(250);
}

//Arduino -sovellus
#include <Morse.h>
Morse morse(13);
void setup()
{
}

void loop()
{
  morse.dot(); morse.dot(); morse.dot();
  morse.dash(); morse.dash(); morse.dash();
  morse.dot(); morse.dot(); morse.dot();
  delay(3000);
}
```

Ideoita

- SOS() -funktio
- Toteuta morse -merkistö kirjastona.
- Tee funktio, jolle voi antaa lähetettävän viestin parametrina.
- Tee sovellus, jolla terminaalin kautta annettu viesti lähetetään ledivilkutusena eteenpäin
- Tässä yhteydessä viestin voi näyttää myös lcd:llä

Oheismateriaali

Viralliset Arduino -esimerkit:

Mika.saari@tuni.fi

3.9.2019

<http://arduino.cc/en/Hacking/LibraryTutorial>

<https://www.arduino.cc/en/Guide/Libraries>