7-szegmenses kijelző

Agócs Norbert és Nagy Dániel

2020. szeptember 8.

Tartalom

Tartalom

Mit tanultunk eddig?

Hardware:

- Arduino alaplap
- Breadboard
- LED
- Ellenállás
- Gomb
- Ultrahang szenzor
- Szervomotor
- Joystick
- 4-szegmens kijelző

Software

- Változó létrehozása
- Különféle változótípusok + specifierek
- Alapműveletek (=, +, -, /, *)
- Alapvető parancsok
- Elágazás: `If()...else`
- Iteráció: `For()`
- Szervomotor irányítása
- Függvényírás
- Tömbök

Tartalom

Mit ír ki a kód?

```
void setup() {
  Serial.begin(9600);

  int szum=0;
  int szamok[4] = {1,2,3,4};

  for(int i =0; i<4; i++) {
    szum = szum+szamok[i];
  }

  Serial.println(szum);
}

void loop() {}
```

Mit ír ki a kód?

```
void setup() {
  Serial.begin(9600);

  int szum=0;
  int szamok[4] = {1,2,3,4};

  for(int i =0; i<4; i++) {
    szum = szum+szamok[i];
  }

  Serial.println(szum);
}

void loop() {}
```

Megoldás: Létrehoztunk egy tömböt, amiben 1-től 4-ig egész számok vannak. Utána a For ciklussal ezen végig megyünk és az aktuális indexű elemet hozzáadjuk a szum változóhoz, amit a végén kiiratunk. A megoldás így: 10.

Mit ír ki a kód?

```
double terulet (double sugar) {
  return (sugar*sugar*3.14);
}
void setup(){
  Serial.begin(9600);

  double r = 10;
  double T = 0;

  T = terulet(r);

  Serial.println(T);
}
void loop(){}
```

Mit ír ki a kód?

```
double terulet (double sugar) {
  return (sugar*sugar*3.14);
}
void setup(){
  Serial.begin(9600);

  double r = 10;
  double T = 0;

  T = terulet(r);

  Serial.println(T);
}
void loop(){}
```

Megoldás: A létrehozott függvény kiszámolja a sugár alapján a kör területét. A kódban megadunk egy sugarat és meghívjuk a függvényt. A megoldást elmentjük a T változóban, majd kiíratjuk, ahol 314.00 jelenik meg.

Ismétlő teszt: Hány állítás igaz?

Az állítások:

- A tömböm első elemére a 0 indexszel tudok hivatkozni.
- Egy általunk írt függvénynek csak 2 argumentuma lehet.
- A `void` típusú függvények nem rendelkeznek visszatérési értékkel.
- Egy tömbben egyszerre tárolhatok `int` és `double` típusú számokat.
- Egy `String` típusú függvény csak `String` értéket tud visszaadni.

Ismétlő teszt: Hány állítás igaz?

Az állítások:

- ✓ A tömböm első elemére a 0 indexszel tudok hivatkozni.
- X Egy általunk írt függvénynek ~~csak 2~~ bármennyi argumentuma lehet.
- ✓ A `void` típusú függvények nem rendelkeznek visszatérési értékkel.
- X Egy tömbben egyszerre nem tárolhatok `int` és `double` típusú számokat.
- ✓ Egy `String` típusú függvény csak `String` értéket tud visszaadni.

Tartalom

Mik azok a külső könyvtárak?

A külső könyvtárak olyan előre megírt kódokat tartalamznak, amik segítik az bonyolultabban használható komponensek programozását.

- Servo könyvtár
- LCD könyvtár
- ...

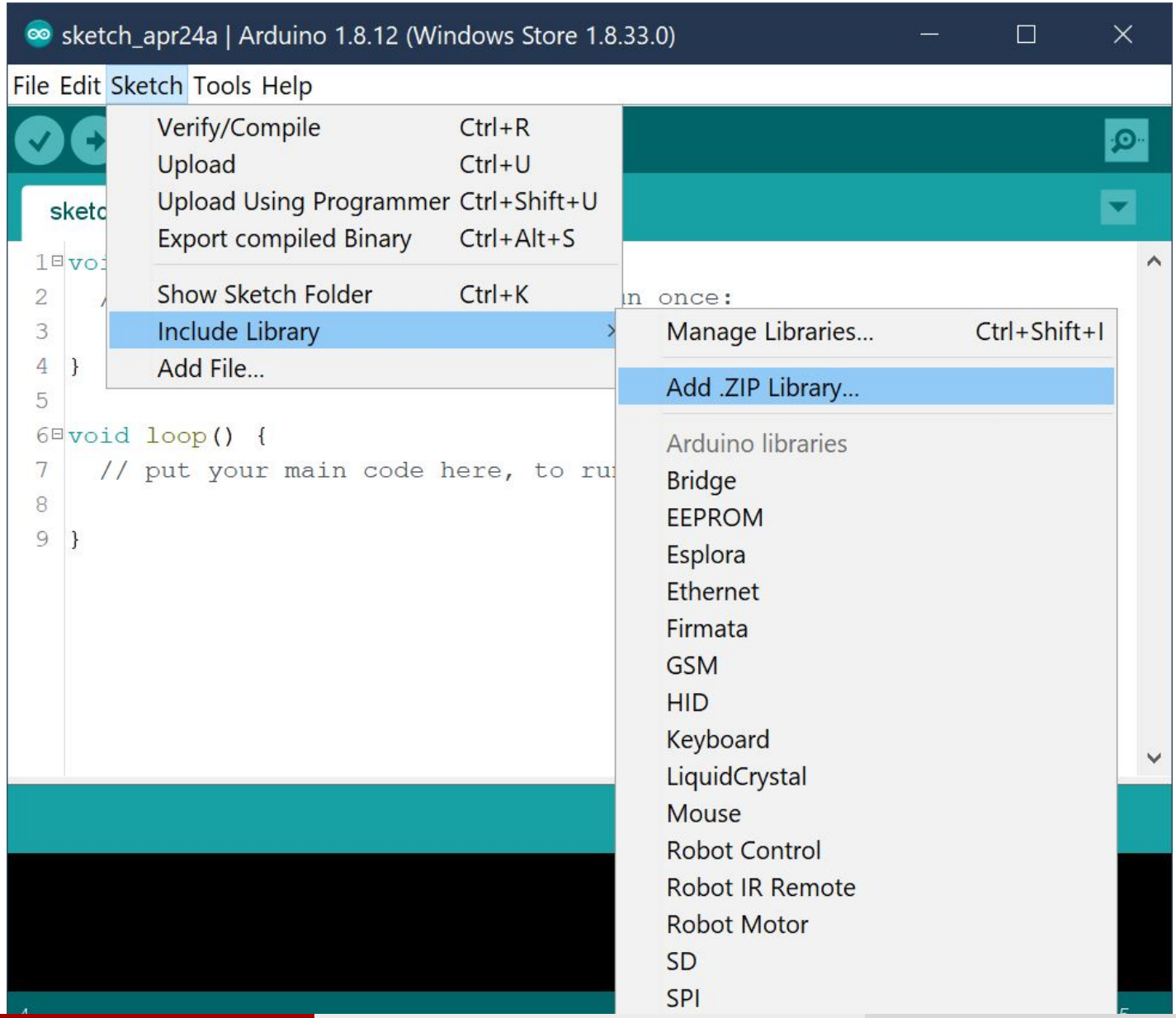
Az arduinoban a `Sketch/Include library` menüpont alatt találhatóak meg az előre telepített könyvtárak.

Külső könyvtár telepítése I

A drive-ban a segédanyagok mappában találhattok egy `segmens4.zip` fájlt. Ez egy könyvtár, amit mi írtunk nektek, hogy könnyebben tudjátok kezelni a 7-szegmenses kijelzőt. Ezt azonban telepítenetek kell az arduinohoz, a következő lépsek szerint:

- A `Sketch/Include library/Add .ZIP library...` menüpontot válasszátok ki.
- A felugró ablakban keressétek meg a letöltött `segmens4.zip` fájlt és válasszátok ki.

Külső könyvtár telepítése II



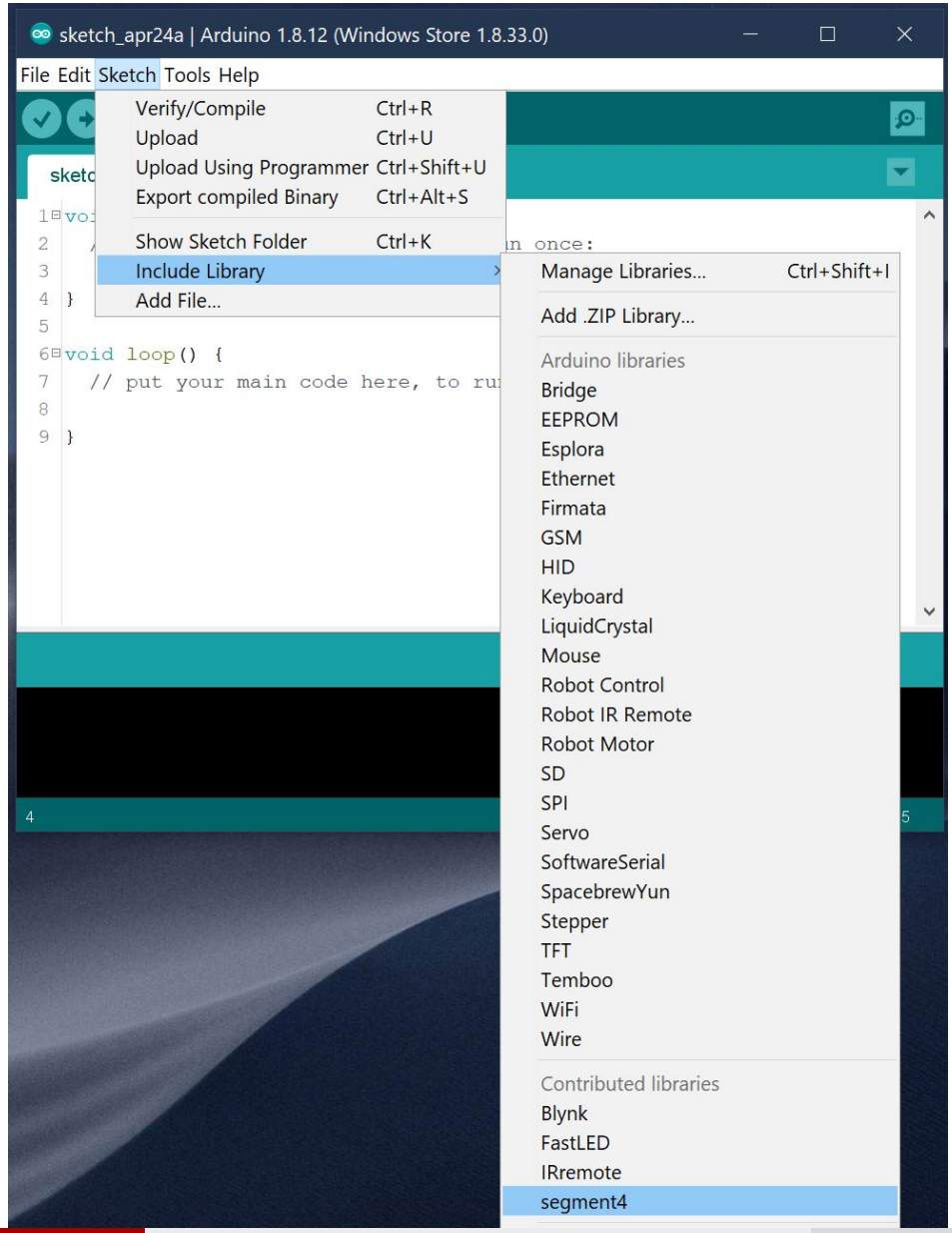
A könyvtárak használata I

Ahhoz, hogy egy könyvtárat használni tudjunk, az `#include` paranccsal meg kell hívnunk a programunk elején.

- Így az összes függvényt tudjuk majd használni, ami az adott könyvtárban benne van.
- Nem szükséges nekünk megírni bizonyos függvényeket.

A könyvtárak berakását a menüsorban is megtehetjük, ha a `Sketch/Include library` menüpontban kiválasztjuk a megfelelő könyvtárat. Most válasszuk ki a segment4 könyvtárat.

A könyvtárak használata II

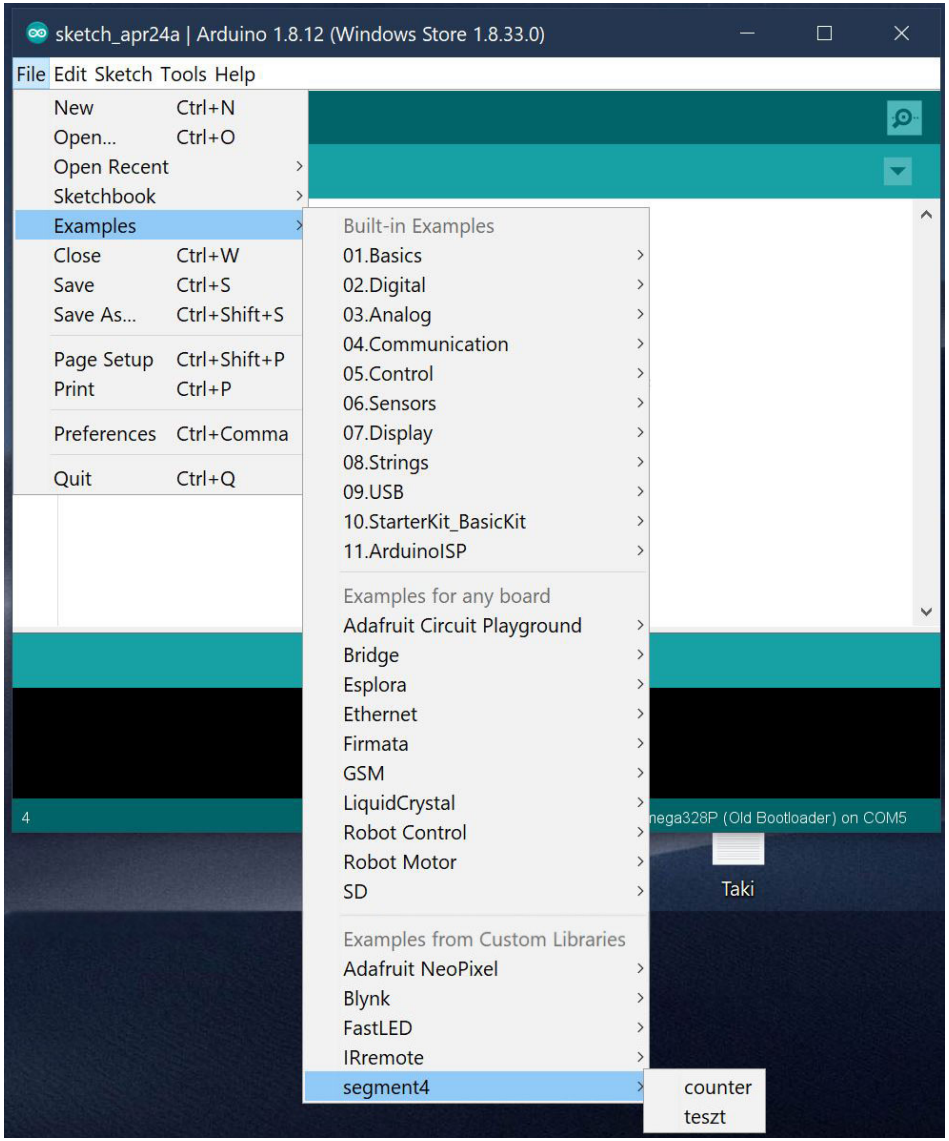


A beépített példa kódok I

A legtöbb könyvtárban vannak előre megírt példa kódok is, amiket bárki kipróbálhat. Mi is elkészítettünk nektek ilyen kódokat, hogy az otthon összekötött szegmens kijelzőt le tudjátok ellenőrizni vele.

- Elérési út: `File/Examples/...`
- Az általunk írt példa kódok a segmens4 alatt találhatóak.
  - `teszt:` Az összekötés tesztelésére.
  - `counter:` Egy számláló megvalósításához.

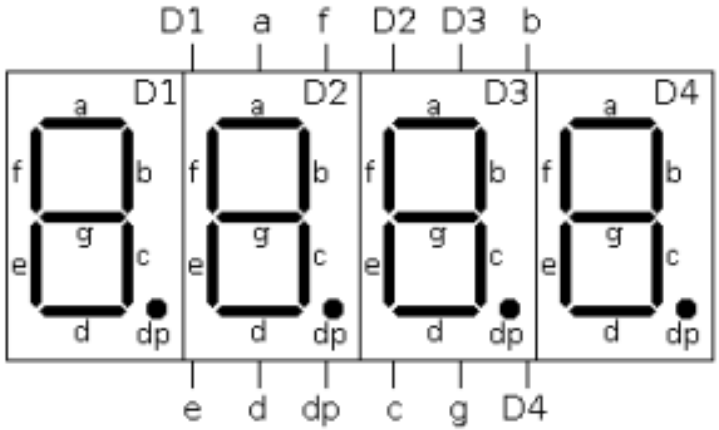
A beépített példa kódok II



Tartalom

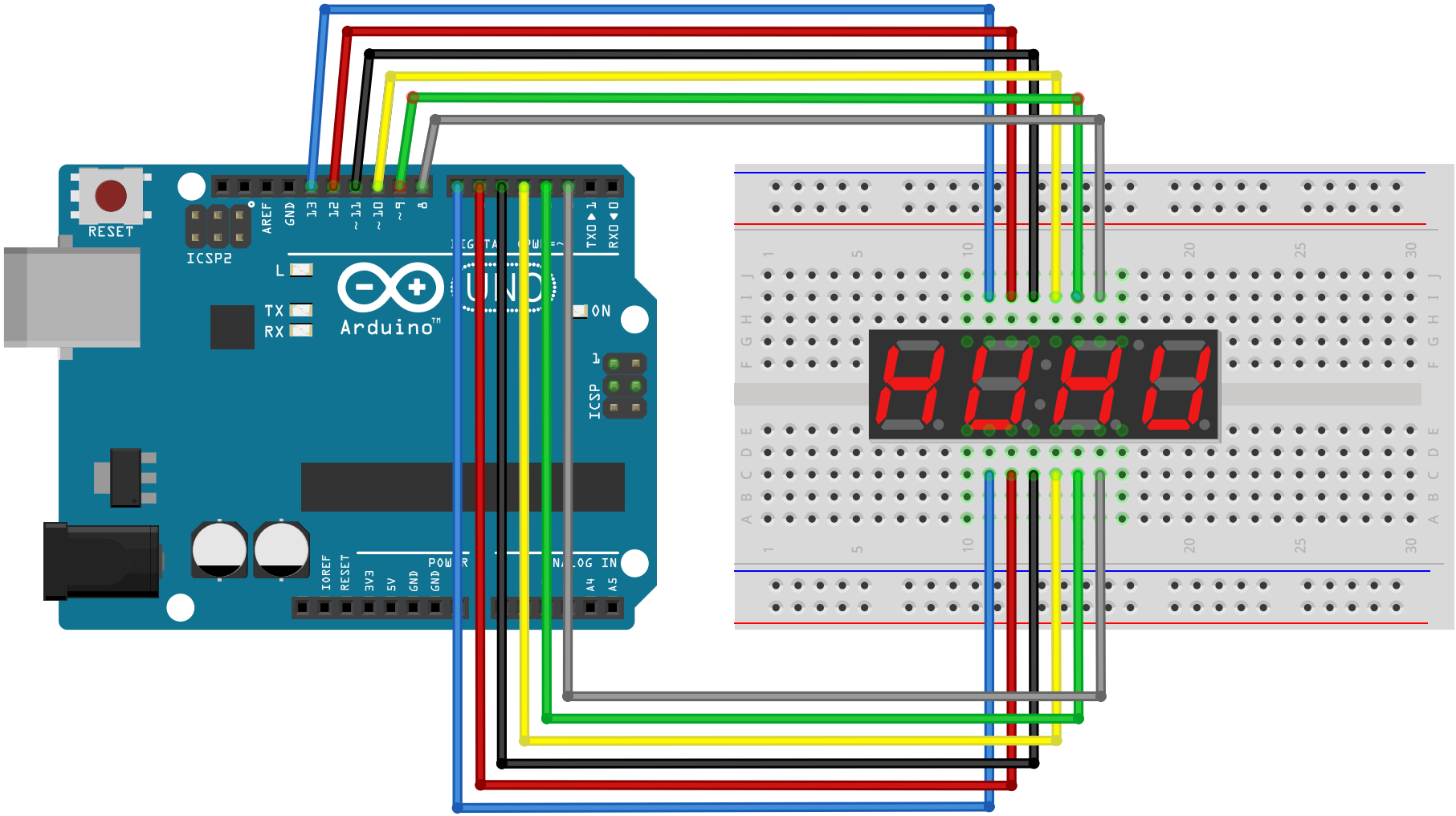
4 Szegmens kijelző

- 4 számjegy megjelenítésére alkalmas
- 12 bemenete van, de 12 kimenet mindössze $2^{12} = 4096$ lehetőséget kódolhat, de 10000 szám van. Tehát trükközni kell.
- A programozása nehéz, ezért készítettünk nektek egy könyvtárat!
- A kijelző egyszerre egy számot képes csak kiírni. Azonban a számok nagyon gyorsan váltanak, ezért a szemünk úgy érzékeli mindtha folytonosan világítana az összes szám.
- Nézzétek meg a segédletek mappában a lassított felvételt!



7 szegmens kijelző használata

Csináljuk meg a következő kapcsolást



Kód megírása

Ha a könyvtárat megfelelően telepítettétek akkor a **segment4** és printNumber függvény is narancssárgára változik

```
#include "segment4.h"
segment4 disp = segment4(13,12,11,10,9,8,7,6,5,4,3,2);

void setup() {
}

void loop()
{
  disp.printNumber(1234,1000);
  disp.printNumber(5678,1000);
}
```

Magyarázat

`segment4()` a második sorban:

- Ez egy osztálynak a konstruktora, itt meg megmondani mit hova kötöttünk
- `segment4(int pin1,int pin2...)` a pineket kell beírni sorrendben

pin1	pin2	pin3	pin4	pin5	pin6
felső sor balról 1	2	3	4	5	6 (felső sor jobb oldali)
pin7	pin8	pin9	pin10	pin11	pin12
alsó sor balról 1	2	3	4	5	6 (felső sor jobb oldali)

`printNumber(int szam, int ido` függvény

- Ezt használjuk egy szám kiiratásához
- `szam`: a szám amit ki akarunk iratni
- `ido`: mennyi ideig legyen kiiratva szám

Feladat

Készítsünk egy másodperc számlálót!

```
#include "segment4.h"
segment4 disp = segment4(13,12,11,10,9,8,7,6,5,4,3,2);
int masodpercek = 0;

void setup() {
}

void loop()
{
  //az aktualis masodperc kiiratása 1000 ms-re
  disp.printNumber(masodpercek,1000);
  //a másodperc számláló növelése 1-el
  masodpercek++;
}
```

Tartalom

Házi feladatok

1. Kössetek be egy gombot a A0 pinre (14-es PIN-ként tudtok rá hivatkozni majd) amivel nullázni lehet a másodpercszámlálót
2. Írjatok egy új kódot, ami azt számolja hányszor nyomjátok le a gombot.
3. Csináljátok meg hogy a számláló visszafelé számoljon
4. Kössetek be több gombot, az egyik növelje a számlálót egy másik csökkentse azt.