

Arduino programozás, hardverépítés

WS2812 LED mátrix óra programfrissítés

Kapcsolódó korábbi előadások:

- 2025.10.02.: Nyomógombok kezelése Arduino programokban
https://megtestesules.info/hobbielektronika/2025/h2025_01.pdf
- 2025.11.13.: Arduino programozás - a DS3231 óramodul beállítása és kezelése nyomógombokkal https://megtestesules.info/hobbielektronika/2025/h2025_04.pdf
- 2026.02.19.: Arduino programozás - DS3231 RTC óramodul színes LED-mátrix kijelzővel https://megtestesules.info/hobbielektronika/2025/h2025_15.pdf

Felhasznált internetes forrás:

Adafruit_NeoPixel: https://github.com/adafruit/Adafruit_NeoPixel

Mechanikai kialakításhoz 3D-s nyomtatófájl: <https://www.bastelbunker.de/pixel-it/>

WS2812_matrix_ora.ino

Módosítás szükségessége:

A korábbi változatnál problémát jelentett a változó memória terület szűkössége, emiatt a program bővítésére nem volt lehetőség.

A millis() függvény több helyről történő elhagyása, helyettesítése időzítési problémákat okozott.

Adafruit_NeoPixel.h library 1.1.7 verzió frissítése 1.15.4 verzióra a színbeállítások egyszerűsítéséhez.

Program tulajdonságai:

A változó memória terület felszabadítása érdekében az egyes digiteket tartalmazó tömbök a programmemóriába kerültek.

```
#include <avr/pgmspace.h> //PROGMEM használatához
```

```
Írás: const byte PROGMEM Digit1_1[9]={.....};
```

```
Olvasás: byte x = pgm_read_byte(&Digit1_1[i]);
```

A színbeállítás új funkciókkal bővült (auto, rainbow, white, manual)

Auto – folyamatosan változó, egyszínű kijelzések

Rainbow – Egyszerre jeleníti meg a szivárvány színeit, melyek balról jobbra mozognak

White – Fehér szín beállítása

Manual – R-G-B színek egyedi beállítása

Felhasznált új parancsok:

getPixelColor() - az aktív pixeleket lehet lekérdezni.

ColorHSV() – a szivárvány színeinek pontos beállítására szolgál

HSV (Hue-Saturation-Value)

Paraméter	Jelentés
Hue	színárnyalat (szivárvány pozíció)
Saturation	telítettség
Value	fényerő

Hue körben halad:

- 0° → piros
- 120° → zöld
- 240° → kék
- 360° → újra piros

ColorHSV() használata

```
uint32_t c = strip.ColorHSV(hue, sat, val);
```

Paraméterek:

```
hue   : uint16_t (0-65535!)
sat   : uint8_t  (0-255)
val   : uint8_t  (0-255)
```

Fontos:

A hue itt **0–65535** tartományban van, nem 0–360!

Teljes szivárvány rajzolása:

```
for(uint16_t i=0; i<strip.numPixels(); i++) {
    uint16_t hue = (i * 65536L) / strip.numPixels();
    strip.setPixelColor(i, strip.ColorHSV(hue));
}
strip.show();
```

A szín és fényerő beállítások EEPROM-ba kerülnek elmentésre, bekapcsoláskor a korábbi beállításokkal indul a kijelzés. (Az EEPROM-ba írás csak a teljes menü végighaladva történik meg)

A kijelző mechanikai kialakítása:

A mátrix kijelző folyamatos és hibamentes működtetéséhez stabil kialakítás szükséges.

Az összeszereléshez szükséges:

- 2mm vastag plexi lap (vagy polikarbonát)
- 3D nyomtatott fekete rács (két db-ból áll, össze kell ragasztani)
- Matt fólia
- 10x10-es műanyag kábelcsatorna (kb. 1m)

A plexi lapot és a fóliát a ráccsal megegyező méretre kell vágni.

A plexi lapot ki kell fúrni a kijelző elektromos csatlakozásához.

Összeszereléshez a plexire helyezük a kijelzőt, majd a rácsot, végül a matt fóliát.

A kábelcsatornát a fedél eltávolítása után méretre kell vágni úgy, hogy a kijelzőre körbe rá lehessen helyezni. Ezzel biztosítható a kijelző egyben tartása.

Az így kialakított keretet célszerű a kijelző hátlapján ragasztószalaggal rögzíteni az elmozdulás ellen.

A vezérlő elektronikát külön, egy megfelelő méretű dobozban célszerű elhelyezni. A méret csökkentése céljából az UNO helyett Pro-Mini, vagy Nano használata javasolt. A kijelzővel az elhelyezéshez szükséges hosszúságú vezetékkel kössük össze. Asztalon, vagy falon is elhelyezhető.

Fontos: DS3231 óramodulnál a CR2032 lítium elem használata esetén, a panelen a töltőáramkört meg kell szakítani (lásd: kapcsolódó korábbi előadások).

3D nyomtatott rács:



Méretre vágott keret:



Összeszerelve:



Asztali tartóval:



Bekötési vázlat:

