

Hálózatba csatolt social network számláló egység

File Péter
Informatika Kar
Debrecen, Magyarország
filepeter98@gmail.com

Beatrix Papp
London South Bank
University
School of Law and Social
Sciences,
London, United Kingdom
pappb@lsbu.ac.uk

Erdei Timotei István
Mechatronikai Tanszék
Debreceni Egyetem, Műszaki Kar
Debrecen, Magyarország
timoteierdei@eng.unideb.hu

Absztrakt— Modern világunkban az információ gyors áramlása elengedhetetlen a vállalatok számára. Egyre több embernek jelent sokat a népszerűség. Számos cég és vállalat indított web lapot vagy csatornát a közösségi oldalakon, ismeretterjesztő, oktatási, kutatási vagy jobb kommunikáció céljából. A fentiek figyelembe vétele mellett került megtervezésre, egy hálózatba csatolt „Social Network” számláló, melynek révén aktuális értéket kaphatunk az éppen feliratkozott követők számáról. Az eszköz automatikusan frissíti az aktuális követők, feliratkozók számát, így mindig a friss adatokat mutatja.

Kulcsszavak—számláló; ESP8266; Wifi; USB; kijelző; social network.

I. BEVEZETŐ

Modern világunkban a közösségi média egyre nagyobb szerepet kap. Egyesek arra használják, hogy egyszerűen népszerűbbek legyenek ismerőseik körében, mások termékeiket szeretnék népszerűsíteni általa.

Egy dolog közös bennük: mindannyian a követők, feliratkozók számával mérik sikerüket a versenytársakkal szemben. Mivel több közösségi média oldalról van szó, ezek napi ellenőrzése egyesével komoly időkiesést von maga után, így praktikus egy helyen megtekinteni az adatokat. Erre nyújt megoldást a soron következő projektben megalkotott eszköz (1. ábra), amely a kiválasztott 3 oldal- az Instructables, az Instagram, és a YouTube feliratkozásokat jeleníti meg egy 7 szegmenses kijelzőn [1]. A kutatás/fejlesztésnek a Debreceni Egyetem adott otthont [12].

II. TERVEZÉSI SZEMPONTOK

2018 -ban általánosan elmondható, hogy a szociális kapcsolatok teljesen átalakultak. Ennek vannak előnyei és hátrányai. Előnyei közé sorolandó, hogy gyakorlatilag bárkivel a másodperc törtresze alatt kapcsolatba lehet lépni, Live Chat -et indítani, konferencia beszélgetéseket, vagy éppen adatot küldeni. A távolság és az idő pedig fontos tényező, költség szempontjából.

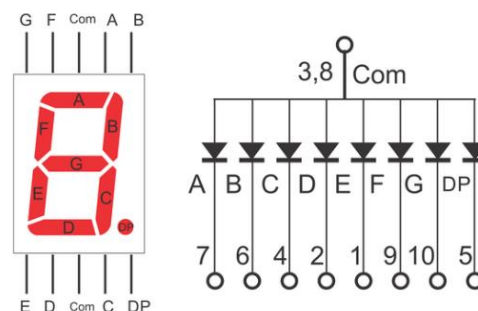


1. ábra Social Networks [7]

Viszont mindezek ellenére komoly negatívummal is rendelkezik, mivel a jelen generáció nagy részt ezen „Social Network” rendszerekre napi életterükként tekintenek, ami romboló hatással van személyes kontaktusokra.

Az eszköz tervezésénél, az elsődleges szempont a kijelző típusának kiválasztása volt. A választás a 4 számjegyű, 7 szegmenses kijelzőre esett, amellyel egyszerűen megvalósítható az adatok megjelenítése a kijelző működésének áttanulmányozása után.

A 7 szegmenses kijelző (SSD) egy elektronikus kijelző eszköz egy olyan számjegy megjelenítéséhez, amely a bonyolultabb pontmátrix kijelzőinek alternatívája. 7 szegmenses kijelzőket előszeretettel alkalmaznak digitális órákban, kalkulátorokban, és egyéb elektronikus eszközökben, amelyek numerikus információt tartalmaznak.



2. ábra Social Networks [8]

A kijelző 7 eleme különböző kombinációkban világít az arab számok megjelenítéséhez. A kijelző szegmenseit A-tól G-ig terjedő betűkkel jelöljük, a valós számok megjelenítéséhez pedig a tizedespontot (DP) használjuk.

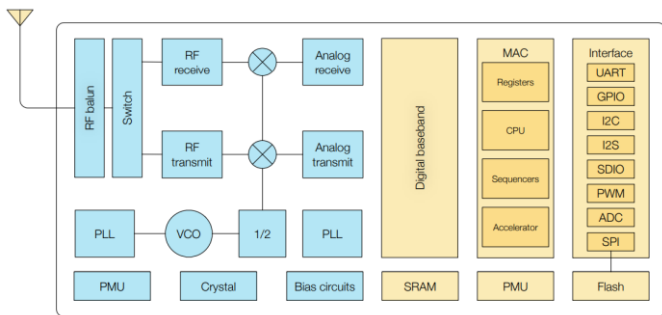
Figyelembe kellett venni azt is, hogy az adatokat egyszerre 3 kijelzőn kell megjeleníteni, és mindezt tudatosan irányítva, így meg kellett tervezni a kijelzők címeinek elosztását [8].

A tervezési szempontok között szerepelt a vezérlő kiválasztása is. A választás az ESP8266 Wifi modulra esett, amely olcsó és megbízható megoldást biztosít a Wifi kapcsolat megteremtésére, és az adatok kezelésére.

Az ESP8266 egy alacsony költségű Wifi mikrochip module, amely teljes TCP/IP verem és mikrokontroller képességgel rendelkezik. Gyártója egy sanghaji székhelyű kínai gyártó, az Espressif Systems [9].

Az ESP module egyik legnagyobb előnye, hogy könnyedén lehet hálózati rendszereket kialakítani a segítségével, és nincs szükség más közbülső egységre, pl. Arduino, PIC, FPGA –ra.

Továbbá ingyenes SDK –val is rendelkeznek, ami révén lehetőség van saját firmware készítésre is.



3. ábra ESP8266 block diagramm [9]

Felépítését tekintve chip rendelkezik, RF receiver/transmitter-el, GPIO és I2C –vel is többek között. Logikai feszültsége pedig + 3.3V – 3.6V –ig terjed.

ESP module kiosztása:

- VCC, Voltage (+ 3.3 V - 3.6)
- GND
- RX, Receive data bit X
- TX, Transmit data bit X
- CH_PD, Chip Power Down
- RST, Reset
- GPIO 0
- GPIO 2 [9].

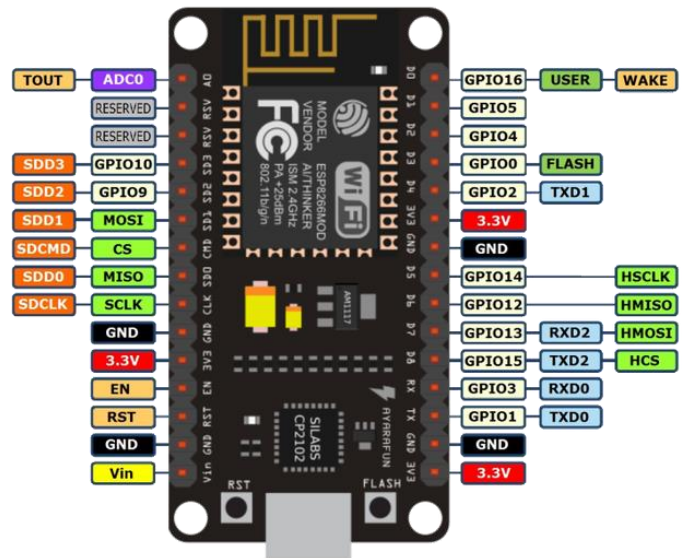
A chip első verziója 2014 augusztusában jelent meg, ESP-01 névvel. A modul már akkor lehetővé tette a mikrokontrollerek számára, hogy Wifi hálózathoz kapcsolódjanak, és egyszerű TCP/IP [2] kapcsolatot valósítsanak meg.

Azért esett a választás a Wifi-re a kapcsolat megteremtéséhez, mivel jóval biztonságosabb és gyorsabb, mint a Bluetooth vezeték nélküli szabvány kapcsolat, és

emellett számos más előnnyel is rendelkezik: a Wifi ugyan magasabb költségű, de jóval nagyobb sávszélességgel rendelkezik mint a Bluetooth vezeték nélküli szabvány. Emellett a késleltetési idő is rövidebb a Wifi kapcsolat esetén.

III. MEGÉPÍTETT ÁRAMKÖR

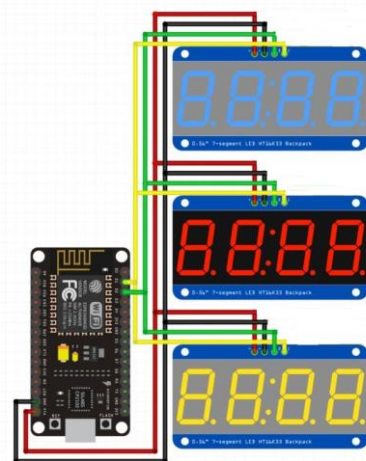
Az áramkör megépítéséhez mindössze egy ESP8266-os Wifi modulra és 3 darab 4 számjegyű, 7 szegmenses kijelzőre volt szükségem és 3 darab I2C [4] hátlapra, valamint egy USB-micro USB kábelre, amivel az áramforráshoz lehet kapcsolni a Wifi modult.



4. ábra ESP8266 block diagramm [10]

Mindenek előtt az ESP8266 module PIN kiosztását kellett tanulmányozni a kapcsolás megtervezése érdekében, mivel rendkívül sokoldalú eszköz, számtalan felhasználási lehetőséggel.

A felhasznált kijelzők lábai (14 darab) bele lettek helyezve az I2C hátlapba, majd a forrasztásuk történt meg a hátlaphoz, A megépített áramkör kapcsolási rajza a 2.ábrán látható.



5. ábra Kapcsolási rajz

Az áramkör tervezésénél az I2C a 4 pin-je került használatra (pozitív, negatív, Clock, Data). Az ESP8266 modul Vin és Gnd pinjét az I2C hátlap pozitív és negatív pin-jeivel lettek összekötve, a D1 és D2 lábak pedig a Clock és Data lábakkal [10].

Ezután következett a kijelzők összekapcsolása, a kijelzők ugyanazon pin-jei is össze lettek kötve. A végleges megépítés előtt egy breadboard próbapanelen lett tesztelve az áramkör. Ezáltal forrasztás mentesen, „dugaszolással” lett megépítve, így egyszerűbb volt a hibák kijavítása, és a tesztelés.

IV. ARDUINO SKETCH

Az Arduino sketch elkészítéséhez, az Arduino IDE fejlesztői környezet került alkalmazásra [3]. Maga az Arduino IDE egy kereszt platformos Java nyelvet alkalmaz a programok írása során. A sketch megalkotása során fontos volt az átlátható kód megalkotása, a további fejlesztési lehetőségek érdekében. Az Arduino sketch az 4. ábrán látható.

```
1 #include <Wire.h>
2 #include <Adafruit_GFX.h>
3 #include "Adafruit_LEDBackpack.h"
4 #include <YoutubeApi.h>
5 #include <ArduinoJson.h>
6 #include <InstagramStats.h>
7 #include <JsonListener.h>
8 #include <JsonStreamingParser.h>
9 #include <InstructablesApi.h>
10 #include <ESP8266WiFi.h>
11 #include <WiFiClientSecure.h>
12 char ssid[] = "Peter";
13 char password[] = "2014augusztus";
14 String userName = "peter.file.5";
15 #define SCREEN_NAME "bekathwia"
16 #define API_KEY "AIzaSyD6F_5BhvkV690qx3Uc5jCOPCRT21UPMhg"
17 #define CHANNEL_ID "UCTNyfXj028is-PN8WCL4eyw"
18
19 Adafruit_7segment matrix = Adafruit_7segment();
20 Adafruit_7segment matrix2 = Adafruit_7segment();
21 Adafruit_7segment matrix3 = Adafruit_7segment();
22 WiFiClientSecure client;
23 YoutubeApi api(API_KEY, client);
24 WiFiClientSecure client2;
25 InstagramStats instaStats(client2);
26 WiFiClient client3;
27 InstructablesApi api2(client3);
28 void setup() {
29
30     Serial.begin(115200);
```

6. ábra. Arduino sketch részlet

A setup rész előtt be lettek importálva azok az előre megírt könyvtárak, amelyek szükségesek az adatok megjelenítéséhez. Az első közülük az Adafruit_GFX [5] könyvtár. Ez egy olyan grafikus könyvtár, amely számos egyszerű alakzat (pont, vonal, kör) megjelenítésére szolgál. Az Adafruit_GFX könyvtára közös szintaxist és grafikus funkciókat kínál az LCD és OLED kijelzők számára. Ez lehetővé teszi, hogy az Arduino vázlatok könnyen beállíthatók legyenek a megjelenítési típusok között. Az Adafruit_LedBackpack könyvtár a 7 szegmensű kijelző használatát teszi lehetővé.

A YoutubeApi, az InstagramStats, és az InstructablesApi könyvtárak az egyes közösségi média adatok kinyerését teszik lehetővé. Ezek olyan függvényeket tartalmaznak, mint például

a YoutubeApi esetében a „channelStats.subscriberCount”, amely visszaadja a megadott csatornához tartozó követők számát. Az importálások után meg kell adni egy elérhető Wifi hálózat azonosítóját és jelszavát, majd ezután következnek az egyes közösségi média azonosítók deklarálása. Az Instructables és az Instagram esetén elegendő megadni a felhasználónevet az azonosításhoz, a Youtube esetében meg kell adni a felhasználó által generált Google API Keyt[6], és a csatorna azonosítóját is a biztonságosság érdekében. A setup rész a címkiostással kezdődik. Az általam használt HT16K33 vezérlő chip-el rendelkező I2C hátlap alapértelmezett címe a 0x70. Fontos, hogy minden eszköznek egyedi címmel kell rendelkeznie, hogy elkerüljük az értelmetlen kimeneteket. Egy I2C hátlap címe könnyen megváltoztatható. A hátlap hátoldalán található 3 forrasztási jumper, A0, A1 és A2. Ha egy jumper-t forrasztással rövidre zárunk, azzal egy címet állítunk be. A0 a legalacsonyabb bitet 1-esre állítja be, A1 a középső bitet 2-esre, A2 pedig a 4-es értéket állítja be. A végső cím 0x70+A0+A1+A2 lesz.

A projektben használt kijelzők közül az egyik az alapértelmezett 0x70-es címet kapta, a másik a 0x71-est, a harmadik a 0x72-est. Az Adafruit_7segment [11] típusú mátrix váltókkal 3 címen kezdjük meg a kommunikációt, így lehetővé téve, hogy egyszerre 3 különböző eredményt tudjunk megjeleníteni rajtuk.

Miután megtörtént a címkiostás, a következő lépés a csatlakozás a Wifi hálózathoz. Ha sikeres volt a csatlakozás, megjelenítjük a soros monitoron a felhasználó IP címét.

A setup részt követi a loop rész, amelyben az ismétlődő utasítások szerepelnek. Ezt követően, amennyiben érvényes azonosítókat adtunk meg az egyes közösségi média oldalakhoz, megjelenítjük a soros monitoron a megjeleníthető adatokat, mint például a Youtube esetében a nézettség számát, kommentek számát, feltöltött videók számát. Mivel a kijelzőn csak egyet lehet ezek közül megjeleníteni, ezért egy kiválasztott statisztikát ki íratunk a kijelzőre. A kiírás úgy történik, hogy az Adafruit_7segment típusú változóra meghívjuk a print függvényt, aminek az első paraméterként megadjuk, hogy mit íratunk ki, másodikként pedig hogy milyen formában. Esetünkben decimális formában történik a kiírás, ezért a második paraméter a „DEC” kulcsszó lesz. Ezután a writeDisplay() függvény segítségével jelenítjük meg az értékeket a kijelzőkön.

Mivel a loop részben helyezkednek el az adatok lekérését végrehajtó utasítások, ezért folyamatosan végrehajthatódnak, és automatikusan frissítik az értékeket. Így érhetjük el, hogy mindig az aktuális adatokat lássuk az eszközünkön.

Mivel a kijelzőink 4 számjegyűek, ezért a maximális érték, amit megjeleníthető a 9999. Lehetőség van arra is, hogy nagyobb számokat is ki tudjunk írni, ez esetben 2 kijelzőt kell összekapcsolnunk, egymás mellé helyeznünk, és megjelenítendő számot 2 részletben tudjuk kiírni maradékos osztás segítségével (az egyik kijelzőn a maradékos osztás egész részét írjuk, a másik kijelzőn pedig a maradékot, decimális formában).

V. DIZÁJN KIALAKÍTÁSA

Az utolsó lépés az eszköz dizájnjának a megtervezése volt, ahol ki lett választva a megjelenés típusa, a keret méretei stb. Kialakítását tekintve, egymás alatt helyezkednek el, a közösségi média prominensebb képviselői, mellettük jobb oldalt pedig a 7 szegmense kijelzők.

A dizájn kialakításának az első lépése a képkeretszerű forma megalkotása volt, ami ragasztással lett megoldva. Fontos volt figyelembe venni a méretek tervezésénél a belmagasságot, hiszen elegendő helyet kellett hagyni az ESP8266-os modulnak, a kijelzőknek, és a jumper kábeleknek. Valamint helyet kellett csinálni a micro USB kábelnek is a keret hátlapján. Így a keret a következő méreteket kapta: 13cm x 18cm x 5cm. Az 5cm-es belmagasság elegendőnek bizonyult, a 13cm x 18cm területen pedig szintén elfértek a kijelzők és a logók. A következő lépés a keret festése volt, színnek a fekete lett kiválasztva.

Ezután következett az üveglap behelyezése, amely a keret nagyságához lett igazítva. A kijelzők összekapcsolásánál figyelni kellett, hogy olyan sorrendben legyenek, amilyen sorrendben vannak a logók feltüntetve a helyes eredmények elérése érdekében. Miután a papír oda lett rögzítve, már csak a kijelzőket és az ESP8266 modult kellett odarögzíteni, ami ragasztószalaggal történt.



7. ábra Social Network számláló

Utolsó lépésként a micro USB kábel lett csatlakoztatva az ESP8266 modulba, a másik vége pedig a számítógép egyik USB bemenetébe. Ezután az Arduino sketch fel lett töltve a modulra, a fordítás és a feltöltés sikeresnek bizonyult. Futtatás után az eredmények ellenőrzésre kerültek a soros monitoron, pár másodperc várakozás után az ESP8266 Wifi modul sikeresen csatlakozott a megadott Wifi hálózathoz, lekérte és megjelenítette a szükséges adatokat a soros monitorra, majd a kijelzőkön is megjelentek a statisztikák. A sebességet befolyásolta a Wifi kapcsolat minősége és sebessége, erősebb hálózat esetén természetesen gyorsabban meg tudjuk kapni az adatokat.

Végül ráhelyezésre került a keret hátlapja, így elrejtve a kábelezt és az eszközöket. Az eszköz egy íróasztalon került elhelyezésre.

VI. ÖSSZEZÉS

A projekt során egy olyan eszköz került elkészítésre, amely előre megadott Közösségi média oldalak statisztikáit jeleníti meg 4 számjegyű, 7 szegmenses kijelzőkön. Vezérlő eszköznek az ESP8266-os Wifi modul lett kiválasztva, amely Wifi-n keresztül kapja meg az adatokat. Az eszköz áramforrásául a számítógép szolgál, amely egy micro USB-USB kábellel van összekötve az ESP8266-os Wifi modullal.

Az eszköz megalkotása során számos tapasztalatot sikerült szerezni a 7 szegmenses kijelzők működését illetően, illetve hasznos tapasztalatokat szereztem az Arduino programozásban is.

A fejlesztési lehetőségek között szerepel további social network rendszerek hozzáadása, és hogy az adatok Android Linux Os rendszeren keresztül is lekérdezhetőek legyenek.

VII. KÖSZÖNETNYÍLVÁNÍTÁS

A publikáció elkészítését az EFOP-3.6.1-16-2016-00022 számú projekt támogatta. A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósult meg. Szeretném a köszönetem kifejezni a tervezéshez szükséges hasznos tanácsokért Erdei Timotei Istvánnak, a Debreceni Egyetem oktatójának.

VIII. HIVATKOZÁSOK

- [1] 7 segment display, (2018.06.03). [Online]. Available: <https://www.electronics-tutorials.ws/blog/7-segment-display-tutorial.html>
- [2] D.Sec, P.Mikulecky, L.Mercl, „Low Cost Architecture of Autonomous Subsystems for Internet of Things”, Trends in Cyber-Physical Multi-Agent Systems. The PAAMS Collection - 15th International Conference, PAAMS (pp.39-49), 2017
- [3] Arduino, (2017, May 14). [Online]. Available: <https://www.arduino.cc/en/main/arduinoBoardUno>
- [4] Z.W.Hu, „I2C Protocol Design for Reusability”, 17 Oct, 2010.
- [5] Adafruit, (2018, 06 03). [Online]. Available: <https://www.adafruit.com>
- [6] Get API Key, (2018.06.03). [Online]. Available: <https://developers.google.com/maps/documentation/javascript/get-api-key>
- [7] Social Networks, (2018, 06 03). [Online]. Available: <https://www.allthingsweb.co.uk/alternative-social-media-platforms-your-business-should-consider-in-2016/>
- [8] Circuits Today, (2018, 06 03). [Online]. Available: <http://www.circuitstoday.com/7-segment-display-to-avr-atmega32-using-cd4511b>
- [9] ESP8266 Datasheet, (2018, 06 03). [Online]. Available: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf
- [10] ESP8266 Pin Definition, (2018, 06 03). [Online]. Available: <https://iotbytes.wordpress.com/nodemcu-pinout/>
- [11] Adafruit_7segment, (2018, 06 03). [Online]. Available: <https://www.adafruit.com/product/879>
- [12] N. C. Obinna, T. I. Erdei, Zs. Molnár, G. Husi, „Investigating Configuration Space Singularities of KUKA KR5,” International Mechatronical Student micro-Conference IMSμC'20