|  |  |
| --- | --- |
| **Óbudai Egyetem**  **Kandó Kálmán Villamosmérnöki Kar** |  |

**TDK DOLGOZAT**

**AVR mikrovezérlők programozása saját eszközökkel**

**OE-KVK Széles Péter**

**2023 YYHITZ**

**Tartalomjegyzék**

[1 Bevezetés 2](#_Toc53041222)

[2 A feladat részletes leírása 3](#_Toc53041223)

[2.1 Fizikailag magvalósított programozó 3](#_Toc53041223)

[2.1 Számítógépes alkalmazás 3](#_Toc53041223)

[3 Logikai Tervezés 4](#_Toc53041224)

[3.1 Logikai architektúra 4](#_Toc53041225)

[3.2 Lehetséges kommunikációs protokollok 4](#_Toc53041226)

[3.2.1 AVR Mikrokontroller 4](#_Toc53041227)

[3.2.1 Számítógép 6](#_Toc53041227)

[4 Eszközök kiválasztása 7](#_Toc53041228)

[4.2 Kiválasztási kritériumok 7](#_Toc53041230)

[4.3 Technológiák/termékek/szolgáltatások összehasonlítása és kiválasztása a kritériumok mentén 7](#_Toc53041231)

[5 Megvalósítás 9](#_Toc53041232)

[5.1 Programozó összeállítása 9](#_Toc53041233)

[5.1 Programozó programozása 10](#_Toc53041233)

[5.1 Számítógépes alkalmazás készítése 10](#_Toc53041233)

[6 Tesztelés 13](#_Toc53041235)

[6.1 Tesztforgatókönyvek 13](#_Toc53041236)

[6.2 Tesztforgatókönyvek végrehajtása 13](#_Toc53041238)

[7 Összegzés 14](#_Toc53041239)

[8 Irodalomjegyzék 15](#_Toc53041240)

[9 Ábrajegyzék 16](#_Toc53041241)

# Bevezetés

Az alábbi dolgozatban AVR mikrokontrollerek programozását fogom bemutatni több saját megoldással. A project célja egy olcsó automatizálható egyköz, és hozzá tartozó szoftver elkészítése, amivel felprogramozhatók az ATmega és ATthiny termékcsalád mikrovezérlői. A projektem csak mikrokontroller konfigurálására és memóriájának felöltésére tér ki.

Az AVR mikrokontrollerek eredetileg az Atmel cég termékcsládja, amit 2008-ban a Microchip Technology nevű cég felvásárolt. Az AVR termékcslád tartalmaz 8 és 16-bites architektúrájú mikrokontrollereket, de projectem során csak a 8 bites AVR-ekkel volt lehetőségem foglalkozni. Tartalmaznak belső flash és EEPROM memóriát, melynek tartalmának kiolvasása és módosítása projectünk célja. Jelenleg a AVR mikrokontrollerek nem számítanak a legmodernebb sem a legolcsóbbak közé, de szerepük még mindig fontos az iparban.

# A feladat részletes leírása

## Fizikailag magvalósított programozó

Az AVR mikrokontrollerek nem programozhatók közvetlenül asztali PC-vel, vagy laptoppal, ezért szükségünk van egy eszközre ami kapcsolatot teremt a kettő között. Kétirányú kapcsolatot teremt a PC-vel ésa mikrokontrollerrel egyidőben. A PC vel történő kapcsolat USB-buszon keresztül alakítjuk ki. A mikrokontrollerrel SPI soros kommunikációs protokollal kommunikálunk. Emellett eszközünknek el kell látnija a mikrokontrollert a számára megfelelő tápfeszültséggel, ami lehet 3,3V vagy 5V. Mikrokontroller „Reset” állapotb kell tudni helyeznie a programozás idejére. Opcionálisan visszajelzést adhat ledeken a kommunikáció meglétéről, és a esetlegesen fellépő kommunikációs hibákról.

## Számítógépes alkalmazás

Alkalmazásunknak kommunikálnia kell USB buszon keresztül a programozóval, kiválasztható legyen a programozó típusa. HEX fájlok betöltése és mentés a PC-n. Programozandó mikrokontroller kiválasztása, vagy automata felismerése. Mikrokontroller flash és EEPROM memóriájának írása, olvasása és validálása. „Fuse” bitek írása, olvasása. Mocrocontroller memóriájának törlése.

# Logikai Tervezés

## Logikai architektúra

###### Rendszerünk három fő részből áll. A számítógép tárolja a feltöltendő és kiolvasott fájlokat, vezérli az információ áramlását, és kapcsolatot tart a felhasználóval. Mivel nincs olyan kommunikációs protokoll amin a számítógép ás az AVR mikrokontrollerek is tudnának kommunikálni, ezért be kell iktatni kettő közé egy programozót, ami kapcsolatot teremt a két különböző kommunikációs protokoll között. A két eltérő protokoll közötti átalakítást egy mikrokontrollerrel érdemes megoldani. A számítógép és programozó között USB-busz az egyetlen napjainkban szélesen elterjedt megoldás. Megemlíthető a soros port és a párhuzamos port, de használatuk nem lenne praktikus, mivel az új számítógépeket már csak USB portal látják el.



ábra Logikai architektúra (saját készítés)

A programozó és a mikrokontroller között lehetőségünk van UART, párhuzamos programozás, SPI és JTAG kommunikációs protokollok közül választani.

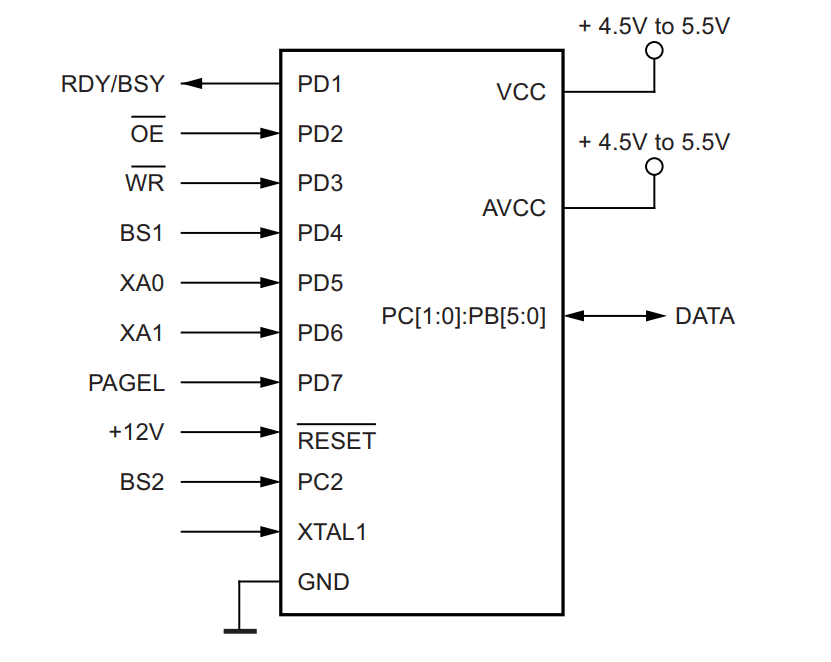
## Lehetséges kommunikációs protokollok

### AVR Mikrokontroller

Az AVR mikrokontrollercsalád programozására 4 lehetőség áll rendelkezésünkre, de ne minden típus támogatja mindhárom programozási módot. Az alábbiakban a 4 programozási módot fogom részletezni.

„Self Programing” az az önprogramozás. A mikrokontroller UART(universal asynchronous receiver-transmitter) soros kommunikációval kap és küld adatokat. Az adatok küldését és fogadását a flash program memória „Boot” szekciójában található, úgynevezett „Boot Loader” program kezeli. Abban az esetben ha a „Boot Loader” program nem található meg a programmemóriában ezt a programozási eljárást nem lehet alkalmazni. A „Bootloader program helyet foglal el a programmemóriából, így a feltölthető kód hossza ennél a programozási módszernél kisebb mint a többi módszernél.[1]”

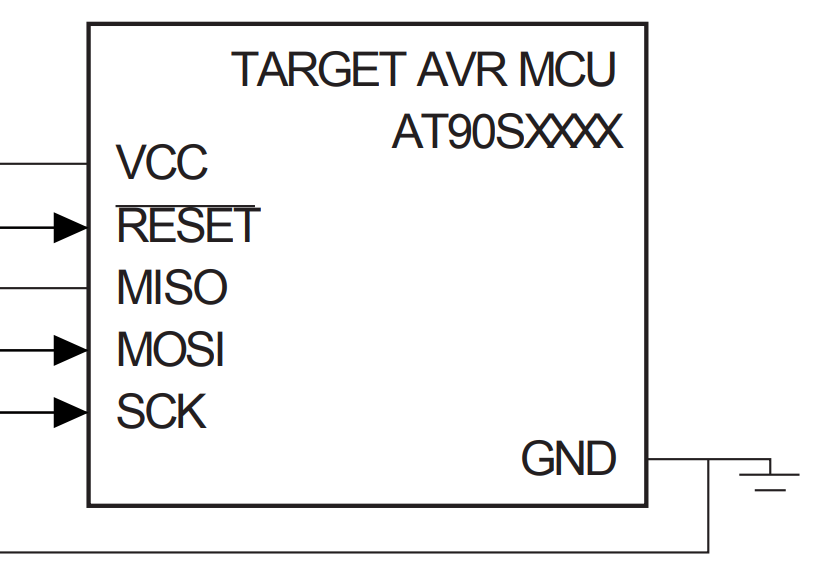
„Parallel Programing”, soros programozási módszer az ATmega328P mikrokontrollertípuson mutatnám be.



ábra 2Párhuzamos programozás[2]

A felhasznált vezetékek száma húsz, ami jóval meghaladja az összes többi programozási megoldás hoz szükséges vezetékek számát. 5V-os tápfeszültség mellet szükségünk van a „RESET” benetre kapcsolt 12V-ra.

„Serial Programin” az az soros programozás, vagy másnéven „In-System Programing” rendszerben történő programozás. A mikrokontrollerrel SPI(Serial Peripheral Interface) buszán keresztül történik a programozás. Erre akkor is van lehetőség ha az SPI buszra más eszközök is csatlakoztatva vannak. Program feltöltéséhez nincs szükség „Boot Loader” programra.

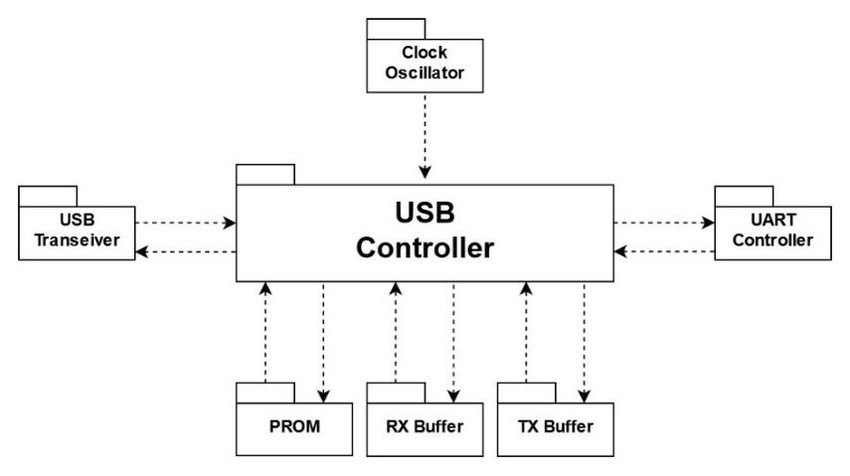


ábra 3 Soros programozás[3]

Az utolsó bemutatott programozási mód a JTAG(Joint Test Action Group)-al való hozzáférésen keresztül történő programozás. A JTAG egy csipbe épített tesztáramkör. Elsődleges feladata a kimenetek bemenetek vizsgálata, de programozásra is lehet használni. Ez a módszer nem használja a mikrokontroller „RESET” bemenetét.

### Számítógép

Az USB kommunikáció a programozó oldaláról megoldható hardveres és szoftveres módon. Hardveres USB esetén szükségünk van egy UART-USB átalakító áramkörre.



ábra 4 USB-UART átalakító blokk diagram [5]

Szoftvere USB esetén az USB adat vezetékei (D+ és D-) közvetlenül csatlakozik a programozó mikrokontrollerébe, és azok jeleit a rajta futó kód dolgozza fel. Ennél a megoldásnál számolni kell azzal, hogy az USB kommunikáció lekezelés, számítási kapacitást von el a programozó mikrokontrollerétől.

# Eszközök kiválasztása

## Kiválasztási kritériumok

A programozó és AVR mikrokontroller közötti kommunikációs protokoll kiválasztásánál elsődleges szempont, hogy az minél több AVR-típussal legyen kompatibilis. Fontos hogy lehet-e üres memóriájú mikrokontrollert programozni az adott módszerrel. Hány vezetéken kell csatlakozni az eszközhöz, mert a vezetékek száma megszabja a csatlakozók méretét, és így ezek befolyásolással vannak a termék árára és méretére. Hány feszültségszint szükséges a megoldáshoz. Az 5V-tól eltérő feszültségszinteket a programozóunknak kell létrehozni, így növelik annak összetettségét, alkarészszámát és méretét. A számítógép és programozó között szempont az alkatrészek száma, mert szintén hozzájárul a termék árához és méretéhez. Meg kell vizsgálni, használ-e fel a programozó számítási kapacitásából a megoldás. Emelet eltérő lehet a szükséges becsült fejlesztési idő.

## Technológiák/termékek/szolgáltatások összehasonlítása és kiválasztása a kritériumok mentén

Programozó és mikrokontroller közötti programozási módszerek összehasonlítása:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | „Self Programing” | „Parallel Programing” | „Serial Programin” | JTAG |
| Kompatibilis AVR típusok száma. | Sok | Sok | Sok | Kevés |
| Tud-e üres memóriájú mikrokontrollert programozni. | Nem | Igen | Igen | Igen |
| Vezetékek száma. | 4 | 20 | 6 | 6 |
| 5V-tól eltérő feszültségszintek száma. | 0 | 1 | 0 | 0 |

táblázat 1 Programozó-mikrokontroller protokollok összehasonlítása

A kevés kompatibilis mikrokontrollerek száma kizárja számunkra a JTAG-al történő programozási eljárás használatát. A „Self Programing” mivel nem alkalmas olyan mikrokontrollerek programozására amikbe nincs előre feltöltve „Boot Loader” kód, így a mi megoldásunkra nem megfelelő. A „Parallel Programing” eljárást a vezetékek nagy száma, és a szükséges 12V-os tápforrás miatt vettettem el. Így az egyenetlen fennmaradó lehetőség a „Serial Programin” eljárás,

Programozó és mikrokontroller közötti programozási módszerek összehasonlítása:

|  |  |  |
| --- | --- | --- |
|  | Hardveres USB | Szoftvere USB |
| Alkatrészek száma. | Nagyobb | Kisebb |
| Becsült fejlesztési idő. | Kisebb | Nagyobb |
| Elvont számítási kapacitás. | Nincs | Van |

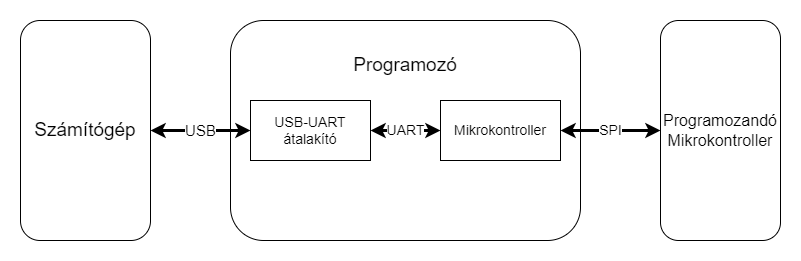
táblázat 2 Számítógép-programozó protokollok összehasonlítása

A két lehetséges USB kommunikációs megoldás közül a hardveres megoldást választottam mert az a projekt megkezdésekor rendelkezésemre ált, de hozzáadott tervezési munkával egy szoftveres megoldás optimálisabb lenne, a kevesebb alkatrészszám miatt. Mivel esetünken az programozó által elvégzendő számítási feladat egyszerű, így nem jelentene gondot a szoftveres USB által elvont számítási kapacitás, így ez az összehasonlítási szempont nem számottevő.

# Megvalósítás

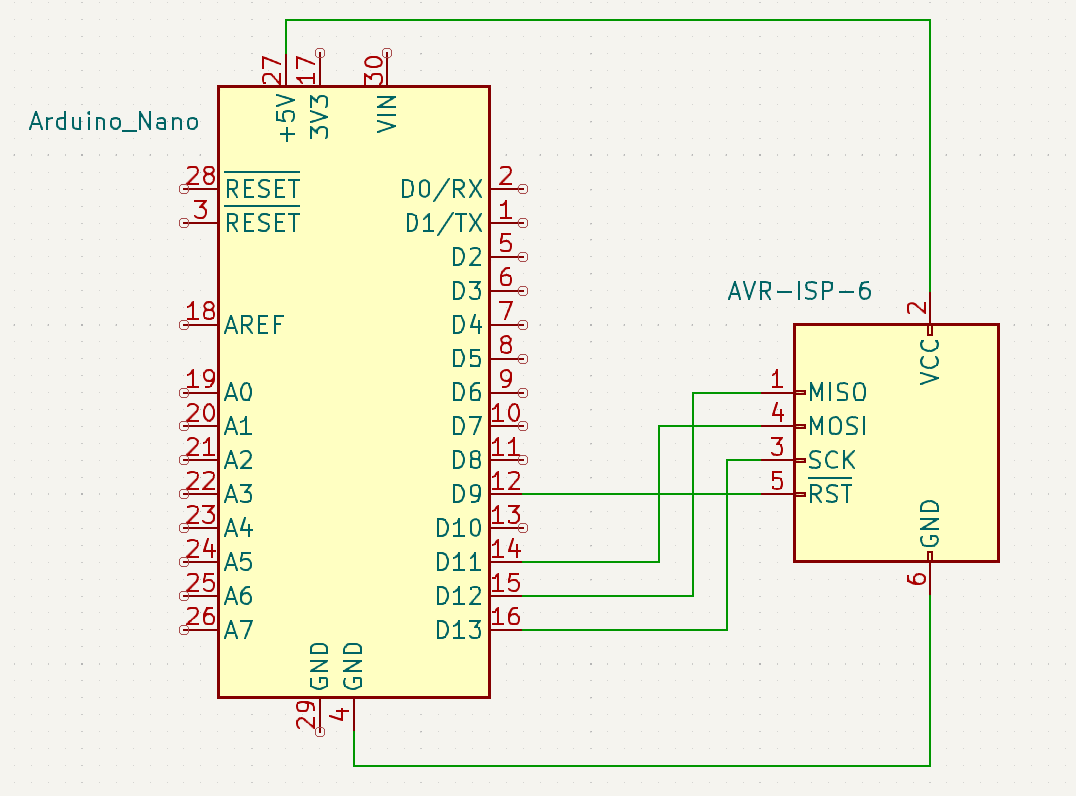
## Programozó összeállítása

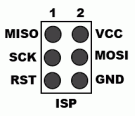
A fentiekben kiválasztott kommunikációs protokollok alapján a rendszerünk részletes blokkvázlata alapján megtörténhet az alkatrészek kiválasztása.



ábra 5 Rendszer részletes blokkvázlata

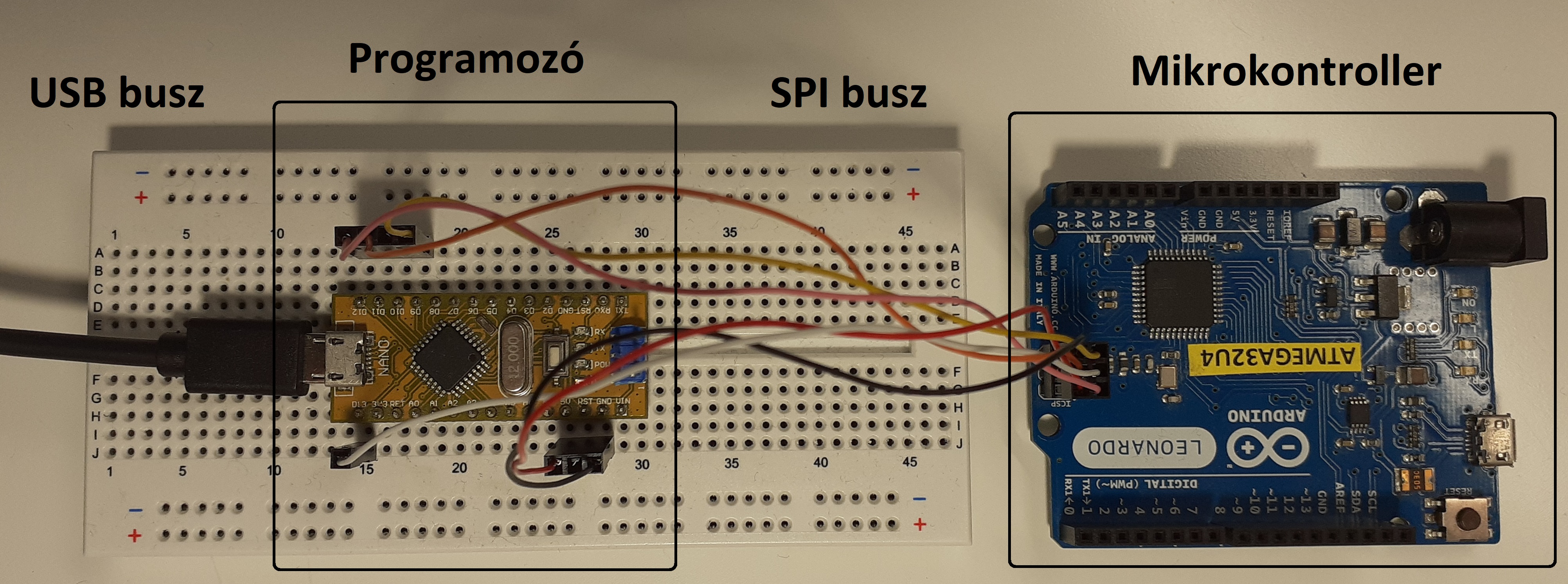
Minden a programozó által támasztott követelésünknek meg tud felelni egy Arduino NANO. Mind az USB es az SPI kommunikációs protokollt támogatja, emelet könnyen beszerezhető. USB vezeték csatlakoztatható az eszközhöz, az általam beszerzett verzió esetében Micro USB csatlakozón keresztül. Beszerzésre került tehát egy USB A – Micro USB vezeték a számítógép és programozó csatlakoztatásához. Az SPI busz vezetékét egyedileg kellett megtervezni. Az Arduino Nano hardveres SPI csatlakozásai adottak, a „REST” bementet a Nano D9 es kimenetével kötöttem össze. A vezeték programozott mikrokontrollerhez csatlakozó végére az ATMEL által meghatározott ASP csatlakozó lábkiosztásával megegyező csatlakozót készítettem.





ábra 6 ISP vezeték kapcsolási rajza ábra 7 AVR ISP csatlakozó lábkiosztása

Megtörténhet a rendszer összeszerelés és a programozó programozása.

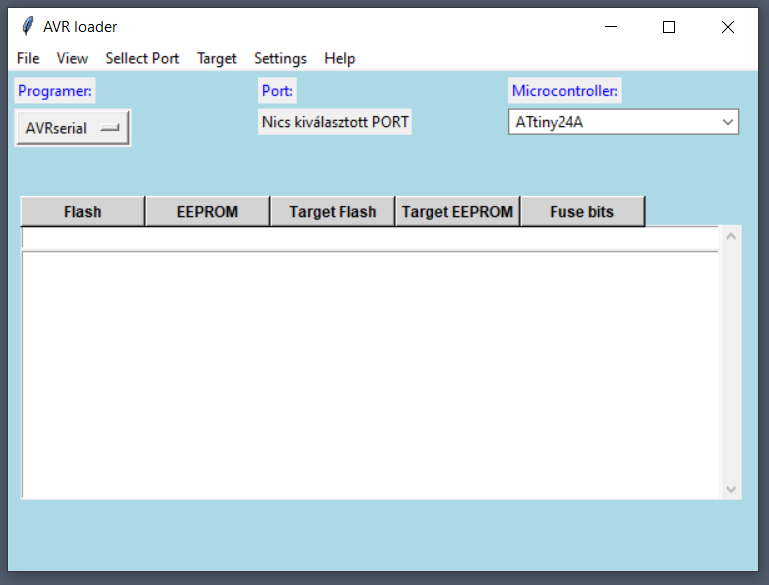


ábra 8 Fénykép a programozóról

## Programozó programozása

A programozón futó program írása C nyelven történt. Feladata az UART-on beérkező adatok továbbküldése SPI buszon, az SPI buszon érkező adatok továbbküldése UART-on, és a programozás idejére a programozott mikrokontroller „RESET” bemenetét a programozás időtartamára alacsony logikai szinten tartsa. Az USART kommunikáció sebessége 57600baud/s,

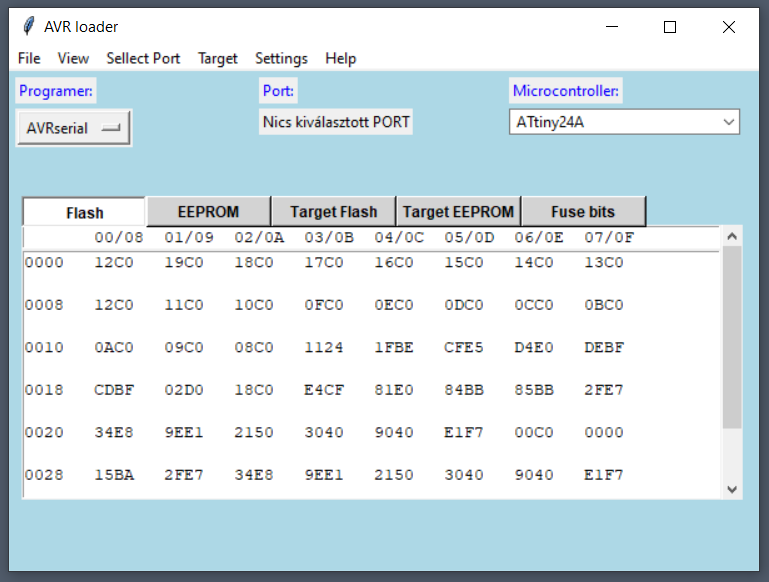
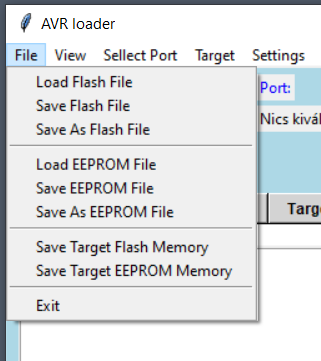
## Számítógépes alkalmazás készítése

**

ábra 9 Alkalmazás kezdőképernyője

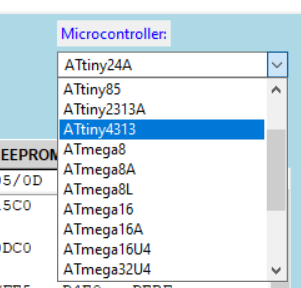
Az alkalmazás elkészítése Python programozási nyelven történt. A grafikus felület a Tkinter segítségével készült. Grafikus felületen be tudunk tölteni és menteni HEX fájlokat, és meg is tudjuk tekinteni tartalmukat. Lehetőség van Programozó kiválasztására. Menüpontban megtalálok az aktív COM portókat, és kiválaszthatjuk meikre van csatlakoztatva programozónkat. A portkezelést a Pyserial könyvtár valósítja meg. Kiválaszthatjuk a programozott mikrokontroller típusát. Jelenleg az alkalmazás az alábbi mikrokontroller típusokat támogatja: ATtiny24A; ATtiny44A; ATtiny84A; ATtiny25; ATtiny45; ATtiny85; ATtiny88; ATtiny2313A; ATtiny4313; ATmega8; ATmega8A; ATmega8L; ATmega16; ATmega16A; ATmega16U4; ATmega32U4; ATmega128; ATmega128A; ATmega128L; ATmega48V; ATmega88V; ATmega168V; ATmega328P.

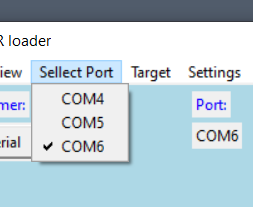
Egy program feltöltéséhez első lépés a HEX fájl betöltése.



ábra 10-11 Hex fájl betöltése

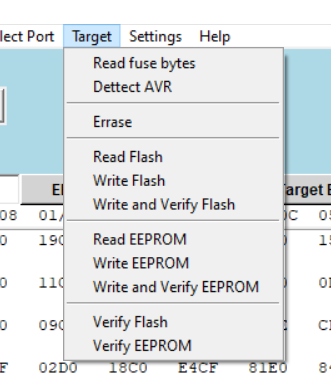
Második lépés a Port kiválasztása, amihez a programozót csatlakoztattuk. Harmadik lépés a Mikrokontroller kiválasztása amit szeretnénk programozni.





ábra 12 Port kiválasztása ábra 13 Port kiválasztása

A feltöltést a „Target” menüpontból tudjuk kiválasztani.



ábra 14 "Target" menüpont

# Tesztelés

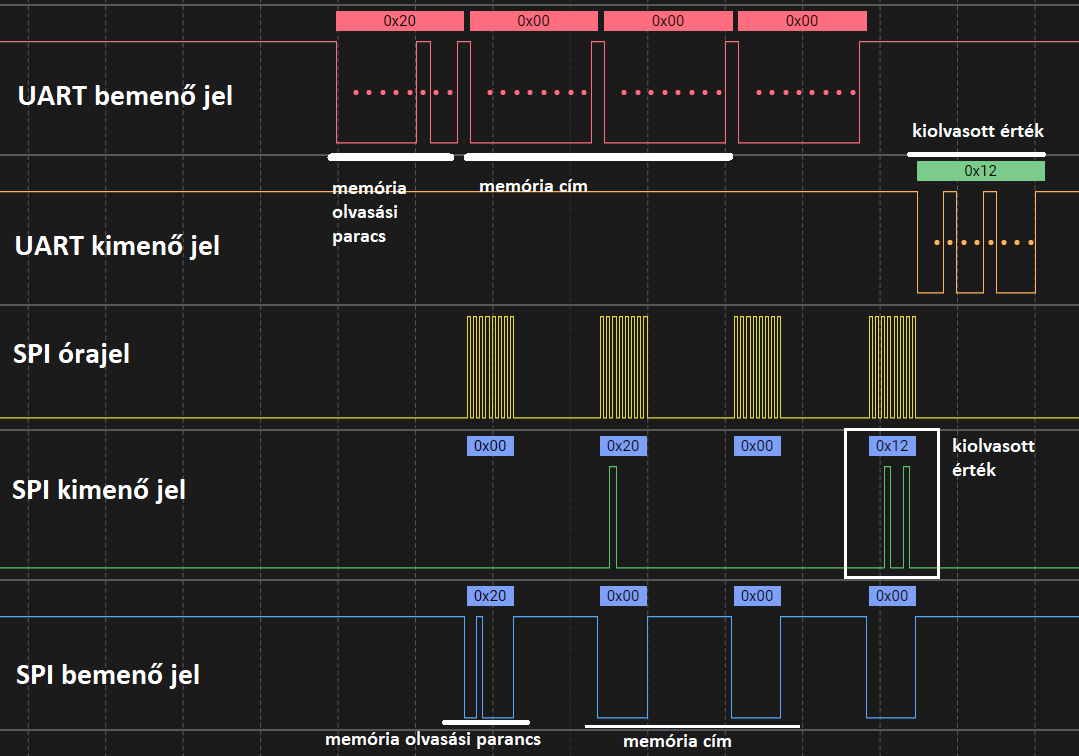
## Tesztforgatókönyv

A programozó jeleit logikai analizátorral vizsgáljuk. Példámban egy nullás címről fogjuk az alsó bájtot kiolvasni, melynek értékét 12 hexadecimális számmal töltöttük fel. A kiolvasási parancs hexadecimális 20.[2]

Második tesztfeladatként feltöltjük az eszközünk memóriáját majd kiolvassuk és alkalmazásunk segítségével a kettőt öszehasonlítjuk.

## Tesztforgatókönyvek végrehajtása

A mért jelalakok elvártnak megfelelők.



ábra 15 Mért jelalakok

Második tesztként teljes feltöltést és kiolvasást végeztem. A kezdet HEX fájl és a kiolvasott HEX fájl megegyezett. A feltöltés és kiolvasás hiba nélkül lezajlott.

# Összegzés

A projekt során létrehoztam egy olcsó és kisméretű programozót az AVR termékcsalád mikrokontrolleréihez, aminek a használata egyszerű és felhasználóbarát. A projektet folytatásaként a hardveresen megvalósított USB kommunikációt szoftveresre cserélném ezzel is tovább egyszerűsítve a programozót, és csökkentvé annak méretét.

# Irodalomjegyzék

|  |  |
| --- | --- |
| [1] | „AVR109: Self Programing" letöltve: 2023.05.29  [AVR109 Self-programming (microchip.com)](https://ww1.microchip.com/downloads/en/AppNotes/doc1644.pdf) |
| [2] | ,,ATmega328P DATASHEET" letöltve: 2023.05.29  [ATmega328P (microchip.com)](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf) |
| [3] | ,,AVR910: In-System Programing" letöltve: 2023.05.28  [AVR910: In-System Programming (microchip.com)](https://ww1.microchip.com/downloads/en/Appnotes/Atmel-0943-In-System-Programming_ApplicationNote_AVR910.pdf) |
| [4] | ,,ATmega16U4/ATmega32U4 DATASHEET" letöltve: 2023.05.29  [ATmega16U4/32U4 Datasheet (microchip.com)](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7766-8-bit-AVR-ATmega16U4-32U4_Datasheet.pdf) |
| [5] | S E Khalzev, A I Vlasov and V A Shakhnov ,,Visual methods of high-level system design for digital hardware components" letöltve: 2023.05.29  https://www.researchgate.net/publication/341373143\_Visual\_methods\_of\_high-level\_system\_design\_for\_digital\_hardware\_components#pf5 |
| [6] | ,,AVRISP mkII" letöltve: 2023.05.29  [Atmel AVRISP MkII (microchip.com)](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42093-AVR-ISP-mkII_UserGuide.pdf) |

# Ábrajegyzék

ábra 1 Logikai architektúra (saját készítés)

ábra 2 Párhuzamos programozás[2]

ábra 3 Soros programozás[3]

ábra 4 USB-UART átalakító blokk diagram [5]

ábra 5 Rendszer részletes blokkvázlata

ábra 6 ISP vezeték kapcsolási rajza

ábra 7 AVR ISP csatlakozó lábkiosztása

ábra 8 Fénykép a programozóról

ábra 10 Hex fájl betöltése

ábra 11 Hex fájl betöltése

ábra 12 Port kiválasztása

ábra 13 Port kiválasztása

ábra 14 "Target" menüpont

ábra 15 Mért jelalakok

táblázat 1 Programozó-mikrokontroller protokollok összehasonlítása

táblázat 2 Számítógép-programozó protokollok összehasonlítása