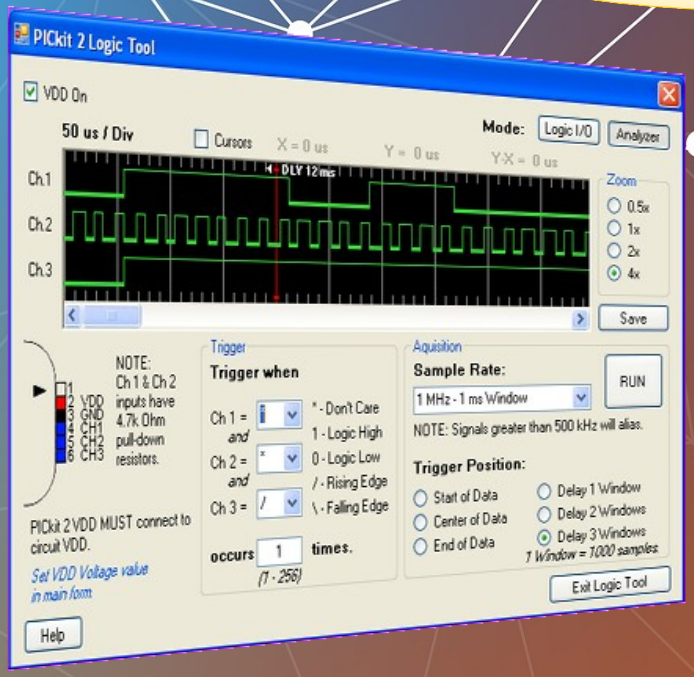
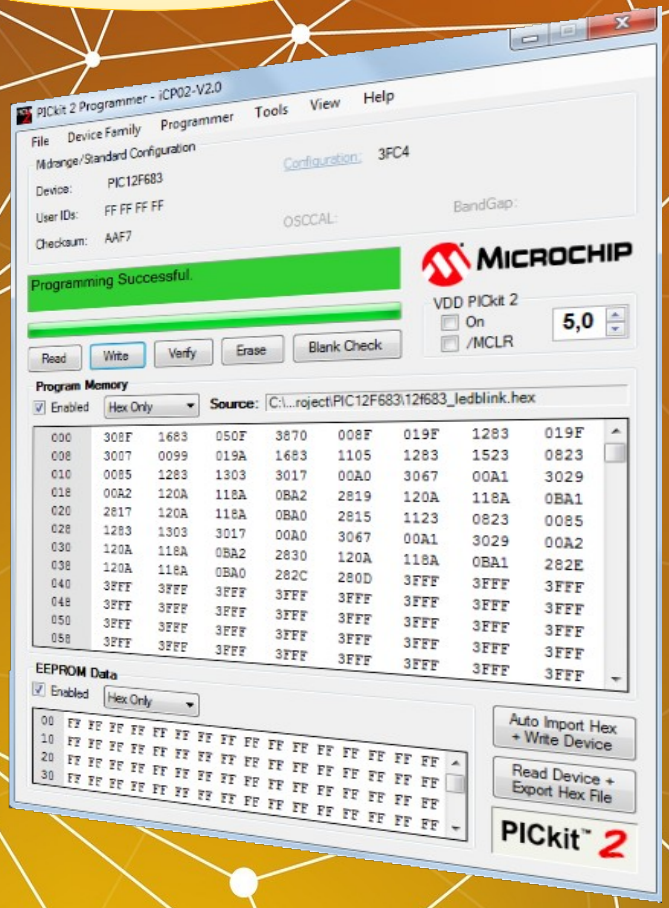
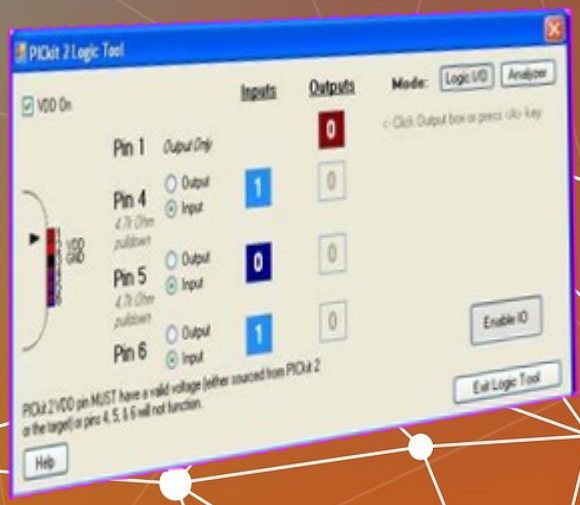
PICkit2, a sokoldalú segéd



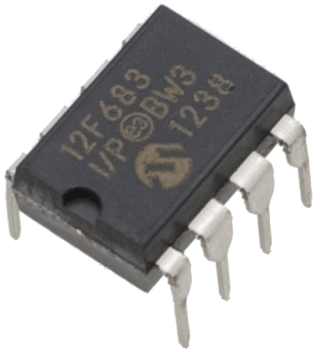




Felhasznált és ajánlott irodalom

- **Varga László:** A PICkit 2 programozó és hibakereső működésének ismertetése
- **Milan Verle:** PIC MicrocontrollersProgramming in Assembly
- **Microchip Technology:** PICmicro Mid-Range MCU Family Reference Manual
- **T&T:** Közepes teljesítményű PIC mikrovezérlők Felhasználói Kézikönyv
- **The Jallib Team:**
  - Have fun with PIC microcontrollers, Jal v2 and Jallib
  - Jal v2 Compiler Documentation

Adatlapok:



- **PIC12F683** adatlap és termékinfo
- **Microchip Technology:** PICkit2 programmer User's Guide
- **Microchip Technology:** PICKIT™2 LOGIC TOOL USER GUIDE
- **Icircuit Technologies:** iCP02v2 USB PIC/EEPROM programmer manual

Miért foglalkozunk a PICkit2-vel?

- A **PICkit2** a Microchip Technology Inc. olcsó, programozásra és hibakeresésre is alkalmas, USB interfésszel ellátott eszköze
- Sajnos, a gyártó már nem támogatja a **PICkit2** programozót (az **MPLAB X** IDE-ben már nem használható), pedig a *„továbbfejlesztett”* utódai nem ennyire jól sikerült, sokoldalúan használható eszközök
- **Mit tud a PICkit2?**
  - Programozhatjuk vele a 2010. előtti PIC mikrovezérlők többségét (PIC10, 12, 16, 18, 24, dsPIC30, dsPIC33, PIC32)
  - I2C és SPI memóriák írása/olvasása
  - Hardveres nyomkövetés (pl. PIC16F887)
  - Logikai vizsgálóeszköz (3 ki/bemenet + 1 kimenet)
  - Háromcsatornás logikai analizátor (5 kHz – 1 MHz mintavételezés)
  - UART eszköz (kétirányú kommunikáció)
- **Mire lesz szükségünk?**
  - Eredeti vagy utángyártott **PICkit2** (esetleg ICP02v2) programozó
  - **PICkit v2.61** alkalmazói program a Microchip Archívumból

PICkit2

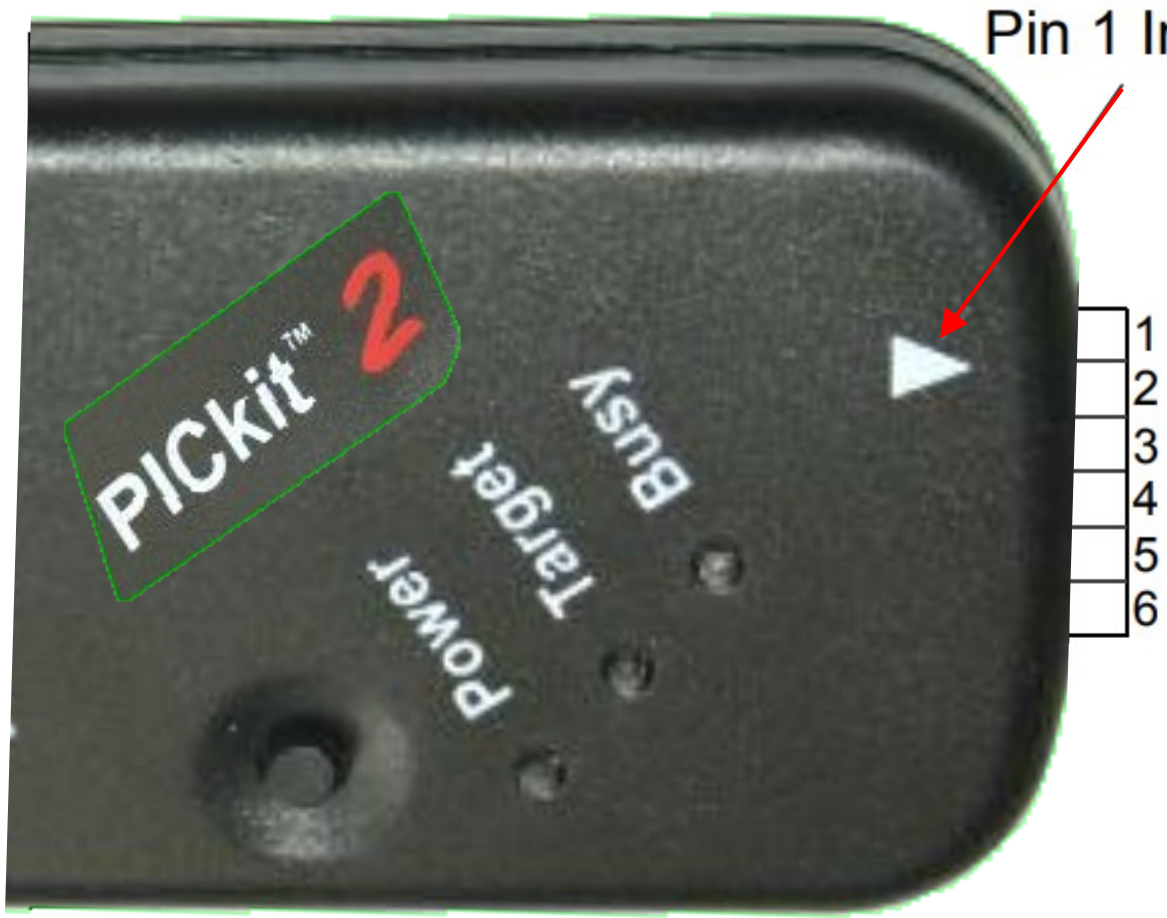
- Kezelőszervek és csatlakozók

A PICkit2 felépítése

- A **PICkit2** programozó egy **PIC18f2550** mikrovezérlőre épült, ami egy Full Speed (12 MHz) USB perifériával rendelkező mikrovezérlő
- A kép bal alsó sarkában látható memória IC-k csak a Programmer-To-Go funkció biztosításához kellenek
- A **PICkit2** kapcsolási rajza és szofvere nyilvános, ezért sokféle utángyártott, illetve átalakított változata készült

A programozó csatlakozó lábkiosztása

- A programozó csatlakozó hatpólusú, 0,1” (2.54 mm) lábtávolságú tüskesor fogadó hüvelysor. Az 1. kivezetést a háromszög alakú jelzés jelöli
- **Figyelmeztetés:** a **Vpp** programozó feszültség felkapcsolásakor akár 13 V is megjelenhet az 1. kivezetésen



Pin Description*

1 = $V_{PP}/\overline{MCLR}$

2 = V_{DD} Target

3 = V_{SS} (ground)

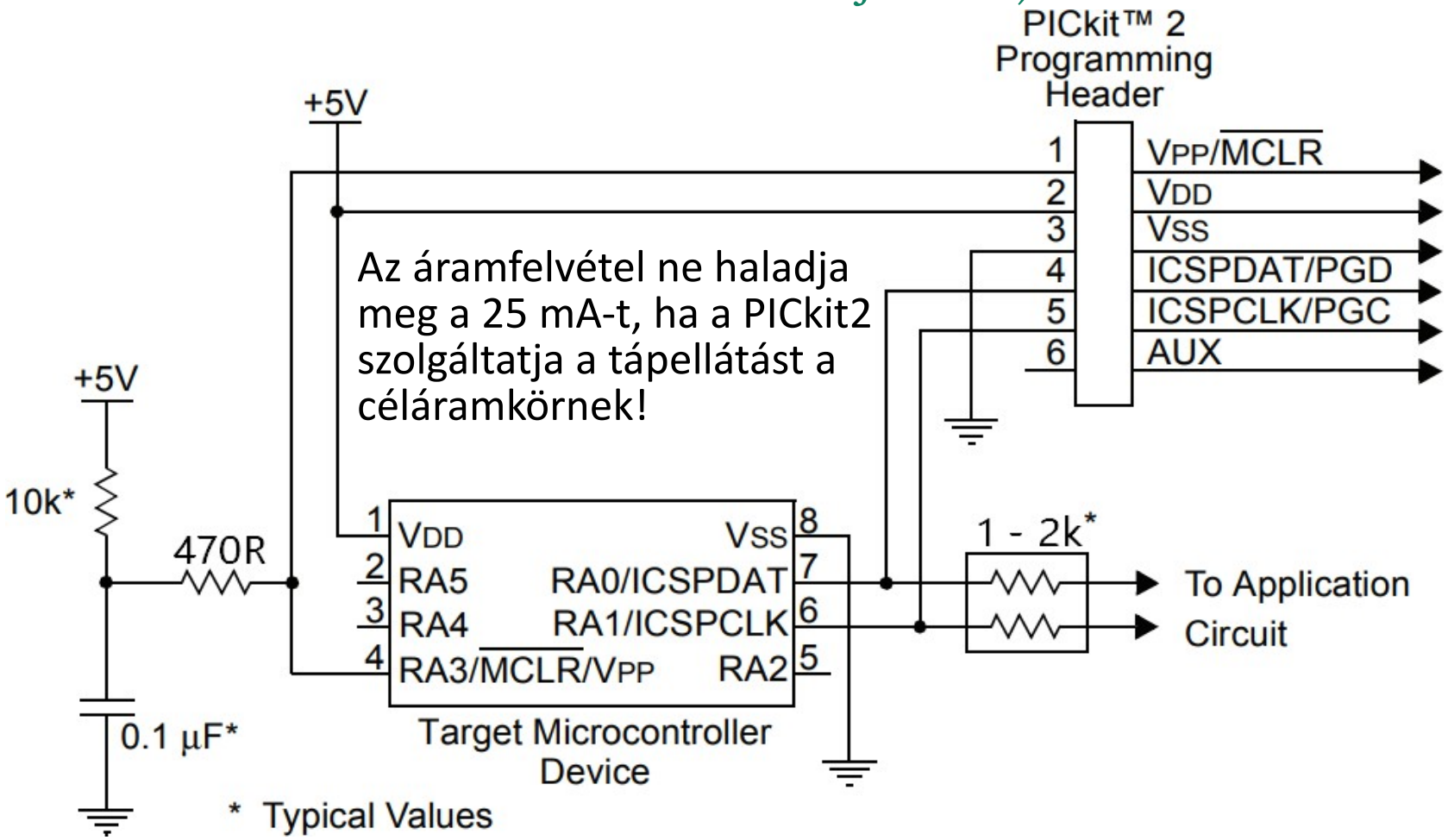
4 = ICSPDAT/PGD

5 = ICSPCLK/PGC

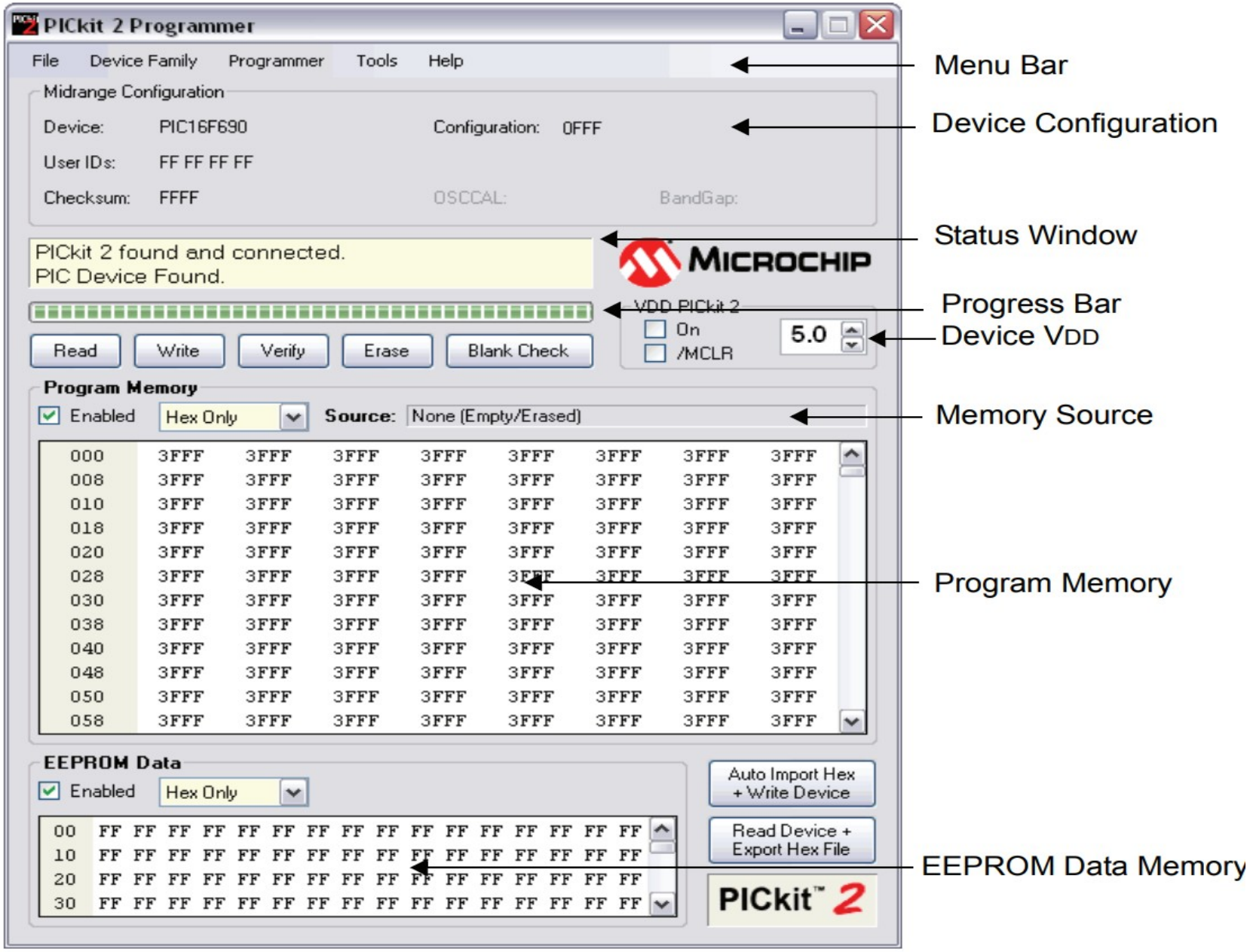
6 = Auxiliary

A PICkit2 javasolt bekötése

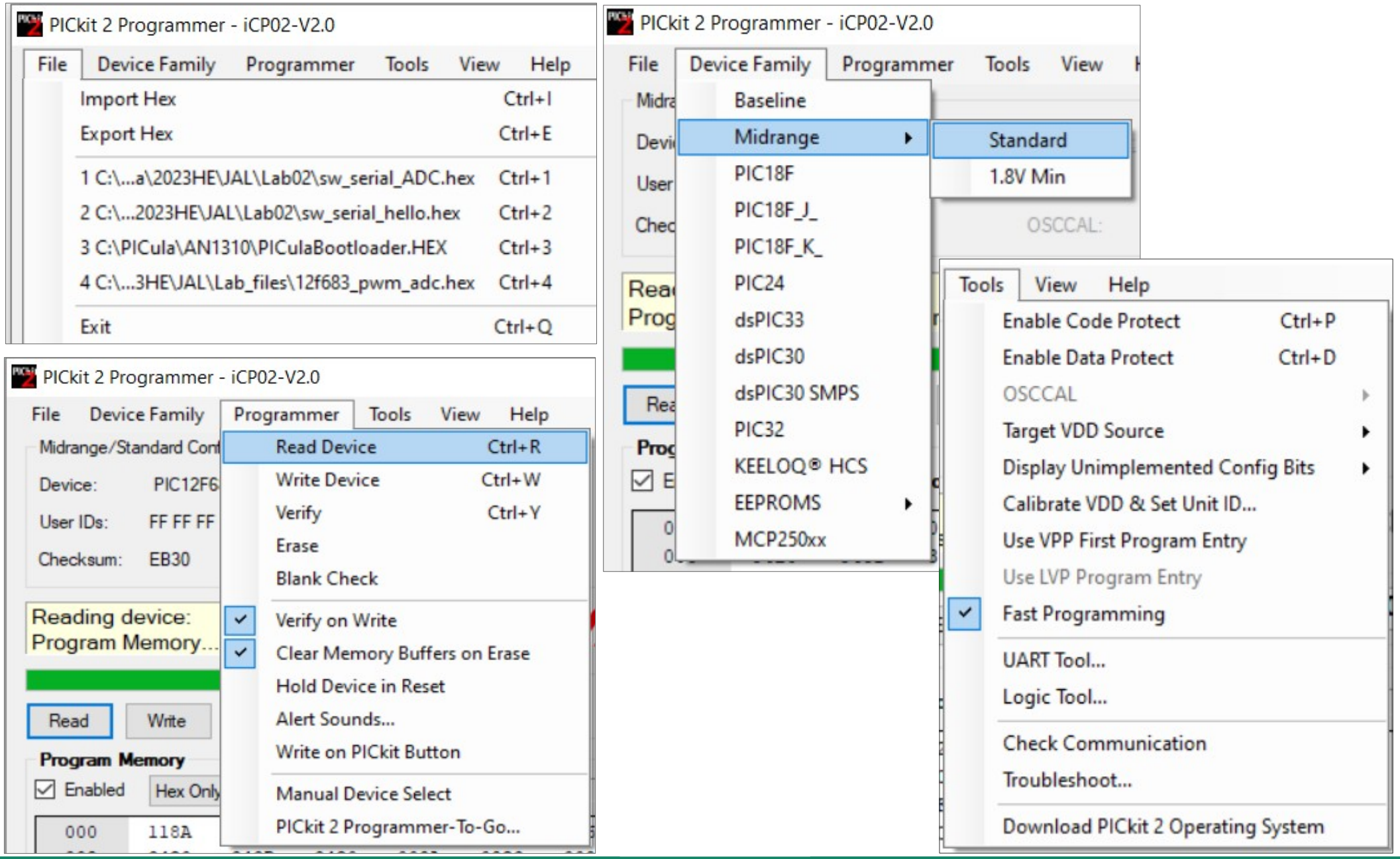
- A **PICkit2** az alkalmazói áramkörbe már beépített mikrovezérlők felprogramozására is képes (**ICSP** = In-Circuit Serial Programming), de az áramkört úgy kell kialakítani, hogy ne zavarja a programozást (ne torzítsa az **ICSPDAT/ICSPCLK** jeleket)



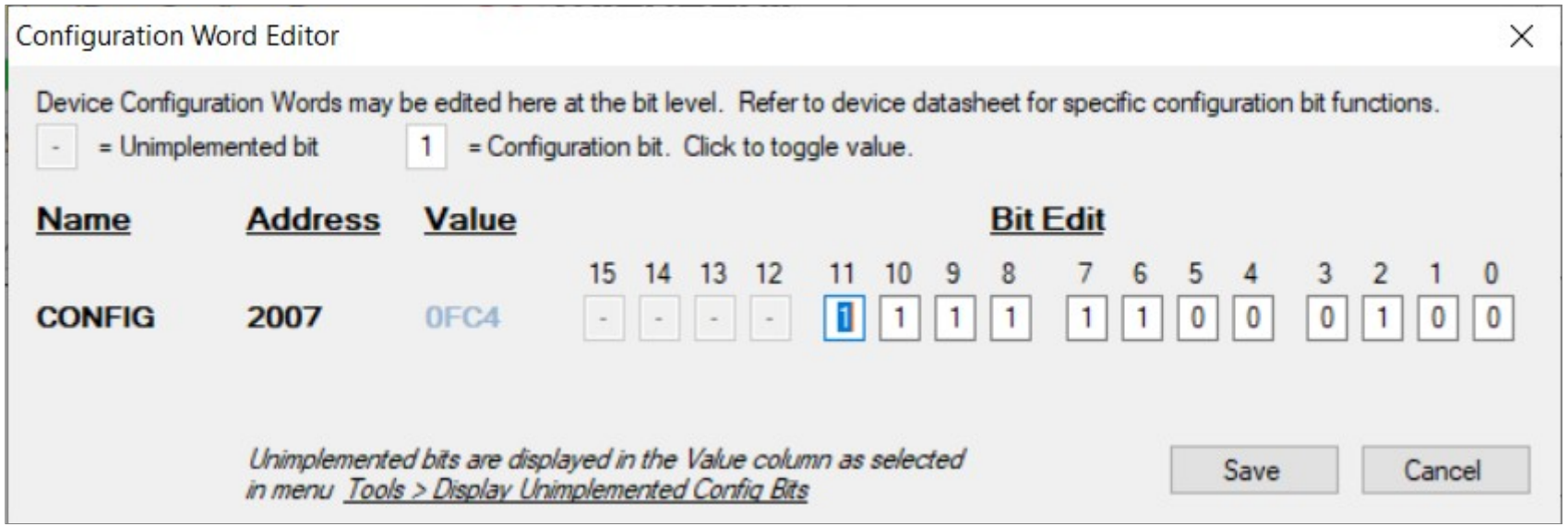
A PICkit2 programletöltő alkalmazás



A menürendszer



Konfigurációs bitek szerkesztése



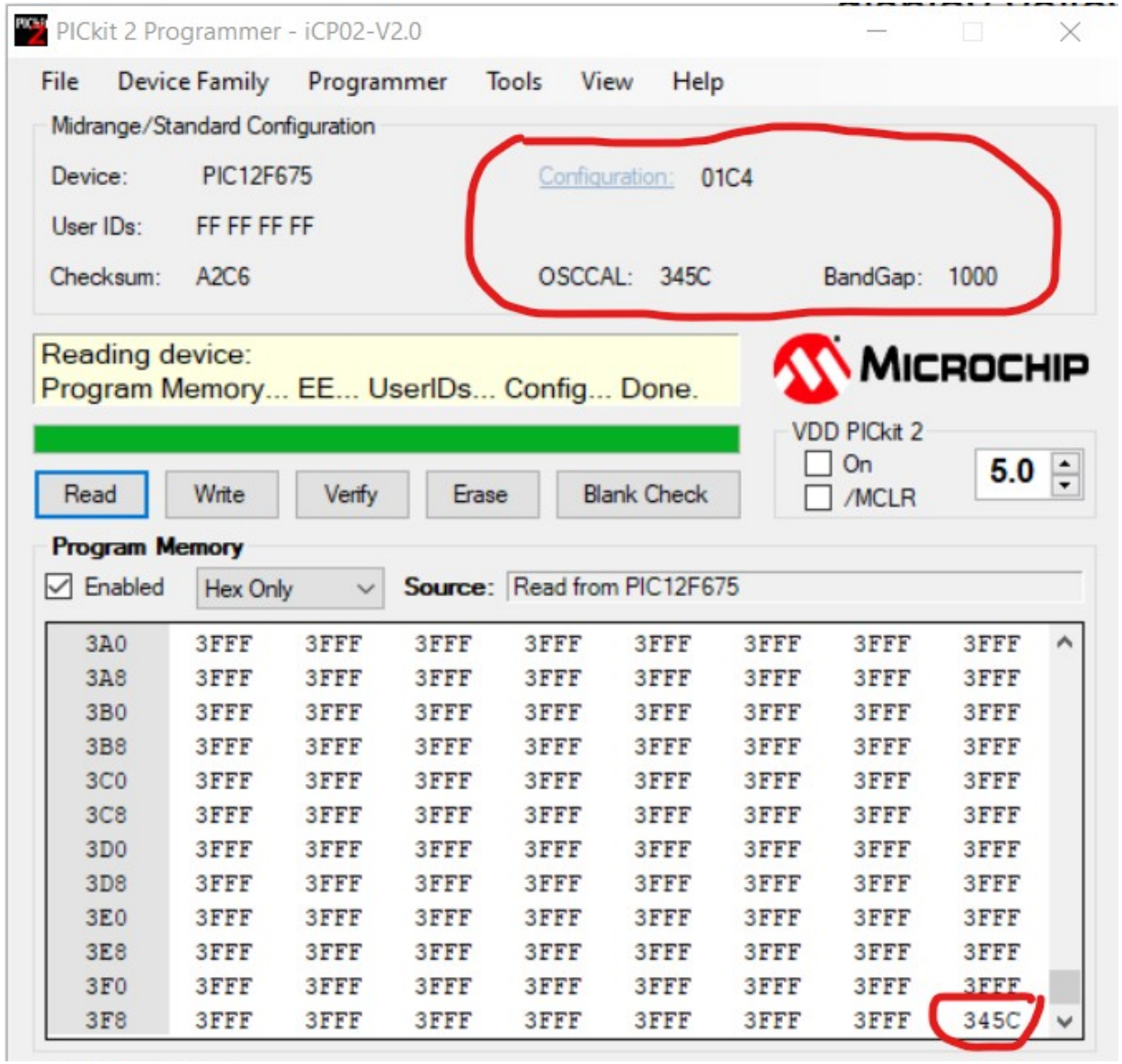
—	—	—	—	FCMEN	IESO	BOREN1	BOREN0
bit 15							bit 8

$\overline{CPD}$	$\overline{CP}$	MCLRE	$\overline{PWRTE}$	WDTE	FOSC2	FOSC1	FOSC0
bit 7							bit 0

FCMEN: Fail-Safe Clock Monitor Enabled bit, **IESO:** Internal External Switchover enable
BOREN<1:0>: Brown-out Reset Selection, **CPD:** Data Code Protection bit
CP: Code Protection bit, **MCLRE:** MCLR pin function select bit
PWRTE: Power-up Timer Enable bit, **WDTE:** Watchdog Timer Enable bit
FOSC<2:0>: Oscillator Selection bits (órajelforrás választása)

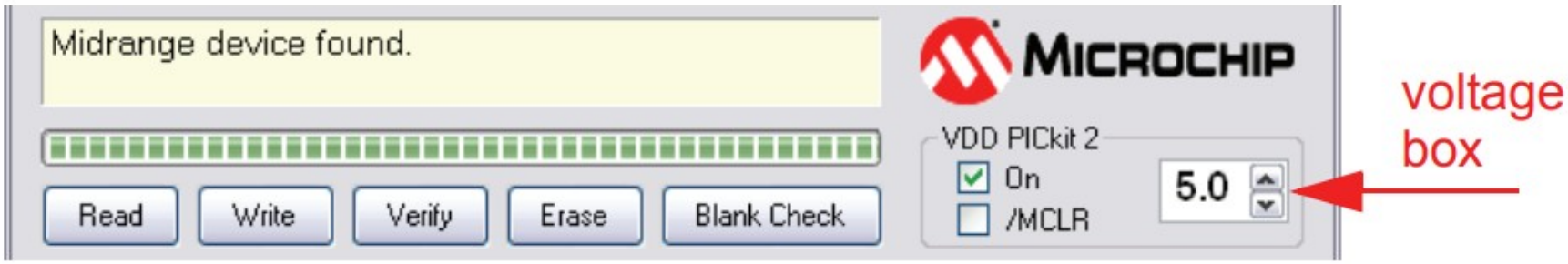
OSCCAL és BandGap

- A konfigurációs bitek beállításai mellett az OSCCAL és BandGap értékek is megjelennek, ha az adott mikrovezérlő rendelkezik ilyen paraméterekkel
- Az **OSCCAL** regiszterbe írandó érték egy **`RETLW k`** utasítás alakjában van kódolva a memória legvégén, itt **`RETLW 5Ch`** a gyárilag kalibrált érték



A céláramkör tápellátása

- A céláramkör számára a **PICkit2** is szolgáltathat tápfeszültséget Ekkor a **Vdd Target** felirat alatti jelölőnégyzet segítségével kapcsolhatjuk ki- és be a céláramkör tápellátását (ha nem detektál tápfeszültséget, a programozáshoz mindenképp nyújt tápellátást)
- A nyilakkal 0.1 V-os lépésekben állíthatjuk a tápfeszültség értékét

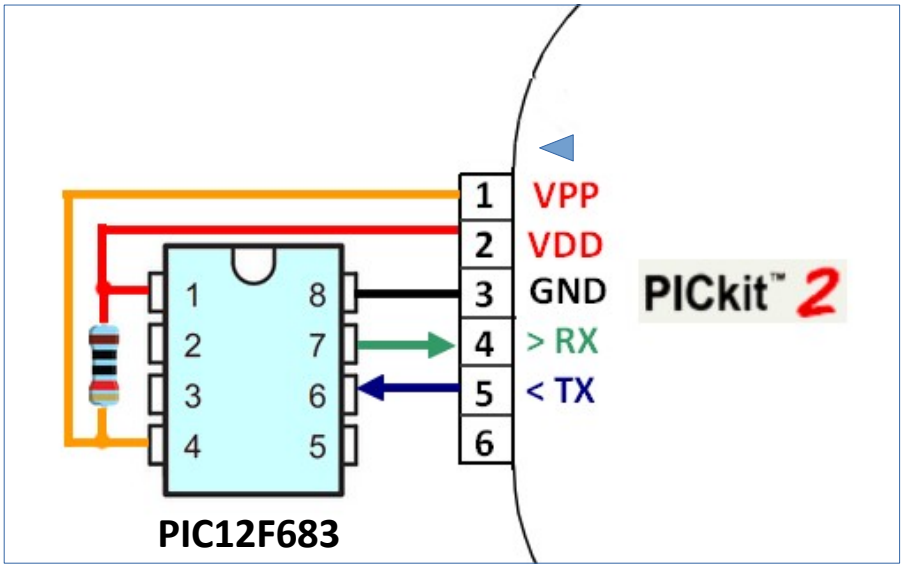


- A tápellátás a **Tools** menüből indítható **Logic Tool** és **UART Tool** paneljén is ki- és bekapcsolható
- A **PICkit2,** mint USB eszköz, 100 mA-re van konfigurálva, így a **PICkit2** és az általa táplált céláramkör együttes áramfelvétele nem haladhatja meg a 100 mA-t, mert a PC lekapcsolja a tápellátást
- Az **/MCLR** jelölőnégyzettel a **RESET**-ben tartást kapcsolhatjuk ki/be

PICkit2, mint UART terminál

- A **PICkit2** kezelőprogramja full duplex **UART** (univerzális aszinkron soros adó-vevő) kommunikációra is használható, 150 – 38 400 Baud közötti sebesség tartományban
- A **PICkit2** az ICSP csatlakozón közvetlenül, logikai jelszinten kapcsolódik a mikrovezérlő **TX, RX** kivezetéseihez, (nem kell **RS-232** transciever)
- **Bekötési vázlat :**

```
PIC12F683    PICkit2
   4 MCLR  →   VPP
   1 VDD   →   VDD
   8 GND   →   GND
   7 GP0   →   RX
   6 GP1   →   TX
```



- **Felhasználási lehetőségek:**
  - Nyomkövető üzenetek kiíratása
  - Adatgyűjtési eredmények naplózása
  - Mikrovezérlő UART kezelőprogram fejlesztése
  - A mikrovezérlő irányítása UART-on küldött parancsokkal

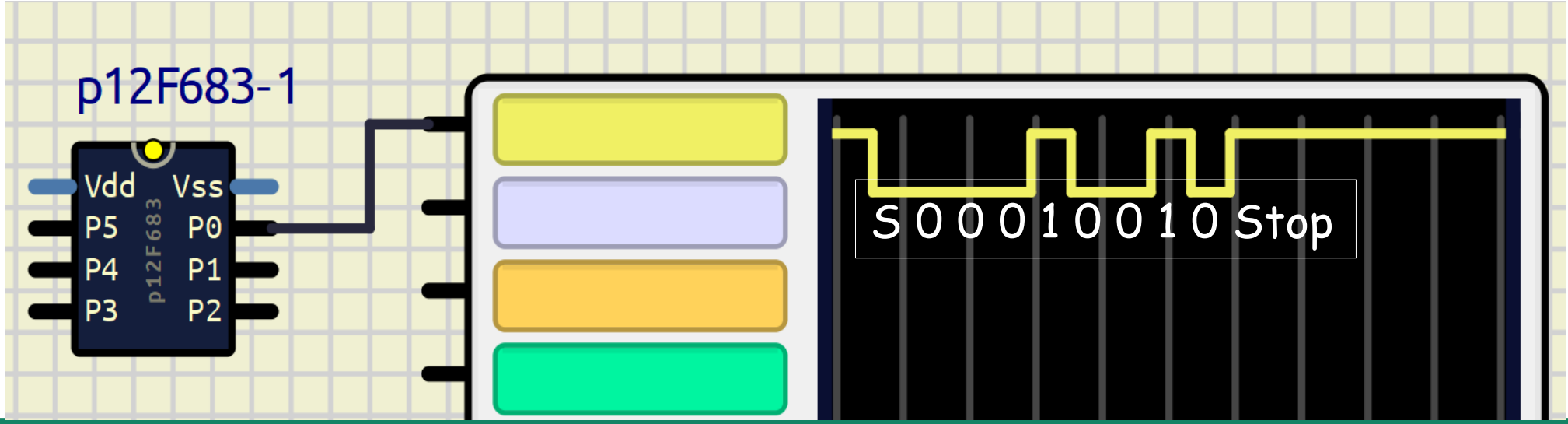
sw_serial_hello.jal

- Öt másodpercenként kiíratjuk a „Hello world!” üzenetet

```
1   include 12f683                              -- PIC céláramkör
2   pragma target CLOCK     8_000_000           -- oszcillátor frekvencia
3   pragma target OSC       INTOSC_NOCLKOUT     -- belső oszcillátor 8MHz-en
4   pragma target WDT       disabled            -- Watchdog letiltása
5   OSCCON_IRCF   = 0b_111                      -- Fosc = 8 MHz beállítása
6   include delay                               -- Késleltető függvények
7   include print                               -- Kiíratő függvények
8   enable_digital_io()
9   alias serial_sw_tx_pin   is pin_A0          -- GP0 lesz a TX láb
10  alias serial_sw_rx_pin   is pin_A1          -- GP1 lesz az RX láb
11  const serial_sw_baudrate = 9600             -- bitráta beállítása
12  pin_A0_direction = output                   -- TX az UART kimenet
13  pin_A1_direction = input                    -- RX az UART bemenet
14  include serial_software
15  serial_sw_init()                            -- SW UART inicializálása
16  const byte str1[] = "Hello world!\r\n"      -- string definiálás
17
18  forever loop
19    print_string(serial_sw_data, str1)        -- üzenet kiírása
20    delay_1ms(5000)
21  end loop
```

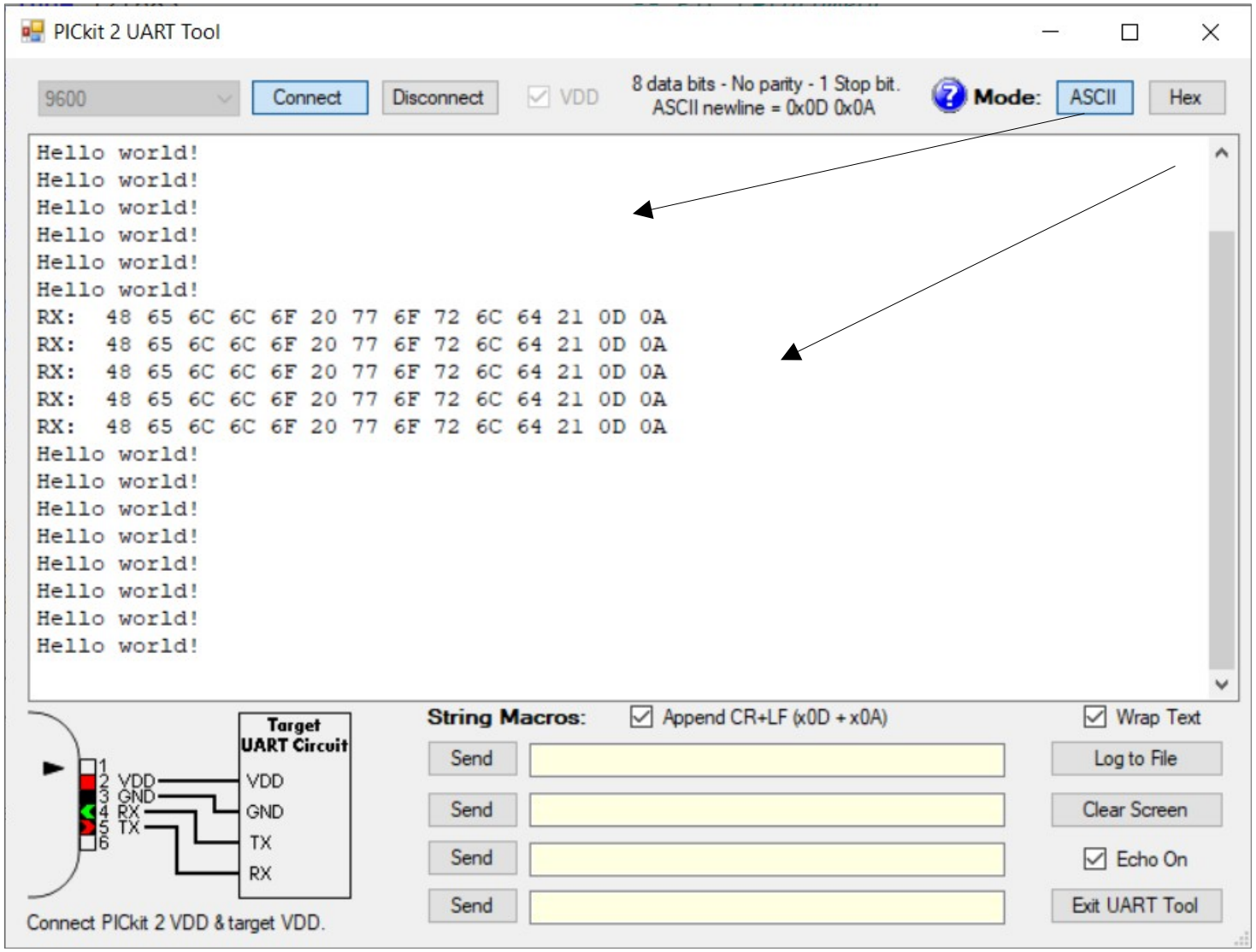
UART küldés próbája a szimulátorban

- A **SimulIDE 1.0.0** szimulátorban végzett próbafuttatásnál csak ’H’ karaktereket küldtünk ki, hogy könnyen ellenőrizhető legyen az eredmény
- A ’H’ karakter ASCII kódja $72_{10} = 01\,001\,000_2$
  Küldésnél a **Start bitet** követően az alacsonyabb helyiértékű bitek következnek, végül az utolsó adatbit után 1 vagy 2 **Stop bit** zárja az adást
- Logikai jelszinten a nyugalmi állapot, az ’1’ adatbit és a STOP bit magas, a START bit és a ’0’ adatbit alacsony



PICkit2 UART Tool

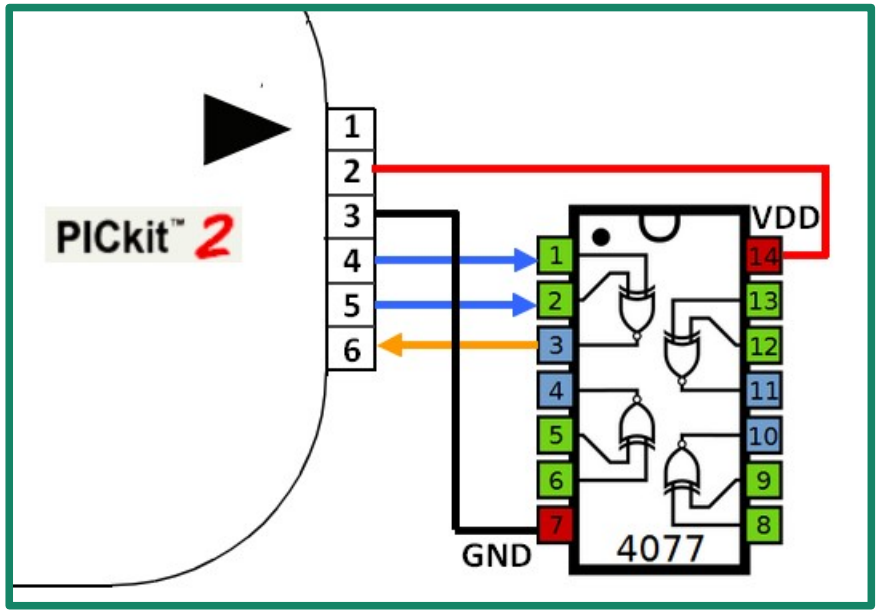
- A beérkező karaktereket ASCII, illetve hexadecimális formában (16-os számrendszerben) is kiírathatjuk



PICkit2, mint logikai vizsgálóeszköz

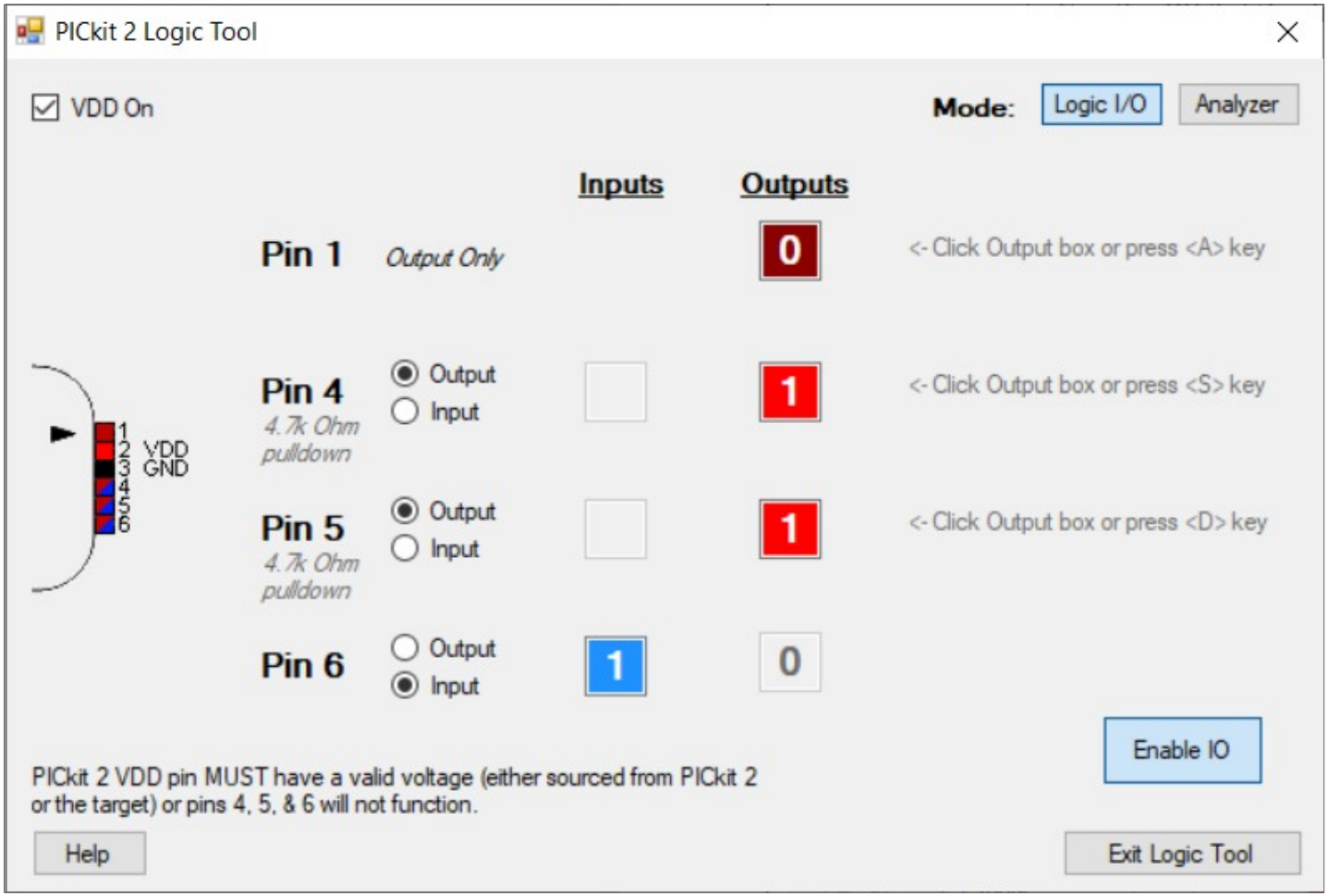
PICkit2, mint logikai vizsgálóeszköz

- A **PICkit2** alkalmazás **Tools** menüjében indíthatjuk a **Logic Tool** funkciót, aminek két üzemmódja van: **Logic I/O** és **Analyzer**
- A **Logic I/O** üzemmód jelszint vizsgáló, illetve jelszint állító funkciót ad
  A Pin 1 (Vpp) kimenet csak jelszintállításra használható, míg a Pin 4, 5 és 6 kivezetések válaszhatóan logikai bemenetek vagy kimenetek lehetnek
- Természetesen a logikai jelszintvizsgálat mikrovezérlő nélkül, általános célra is használható, mint az alábbi példában, ahol egy **XNOR** kapu működését vizsgáljuk (**CD4077** IC)
- A **PICkit2** 4. és 5. kivezetését kimenetre állítjuk és az első kapu bemeneteit vezéreljük velük
- A **PICkit2** 6. kivezetését bemenetre állítjuk és az **XNOR** kapu kimenetével kötjük össze
- A tápfeszültséget és a GND-t is be kell kötni (itt a **PICkit2** adja a tápfeszültséget)



PICkit2, mint logikai vizsgálóeszköz

- A **Vdd On** bejelölése után kattintsunk az **Enable I/O** gombra, majd állítsuk kimenetre a Pin 4 és Pin 5 kivezetéseket, ill. bemenetre a Pin 6 kivezetést! Általánosságban a Pin 1, 4, 5, 6 kimeneteket az egérkattintás helyett az *A, S, D, F* gombok lenyomásával is billegtethetjük

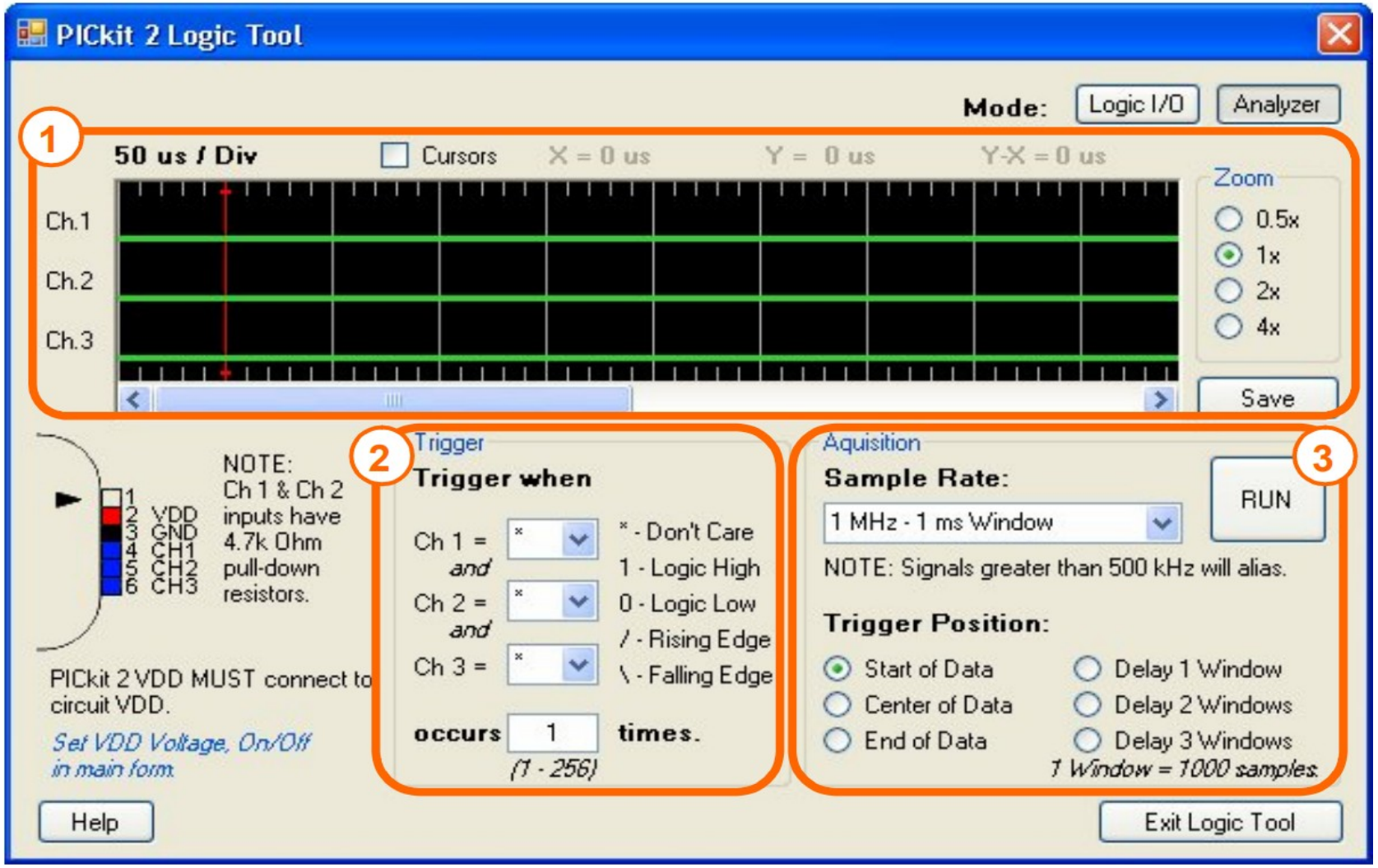


A kísérleteinkből kapott igazságtáblázat megfelel az **XNOR** kapuénak, tehát a vizsgált kapu jól működik

Pin 4	Pin 5	Pin 6
0	0	1
0	1	0
1	0	0
1	1	1

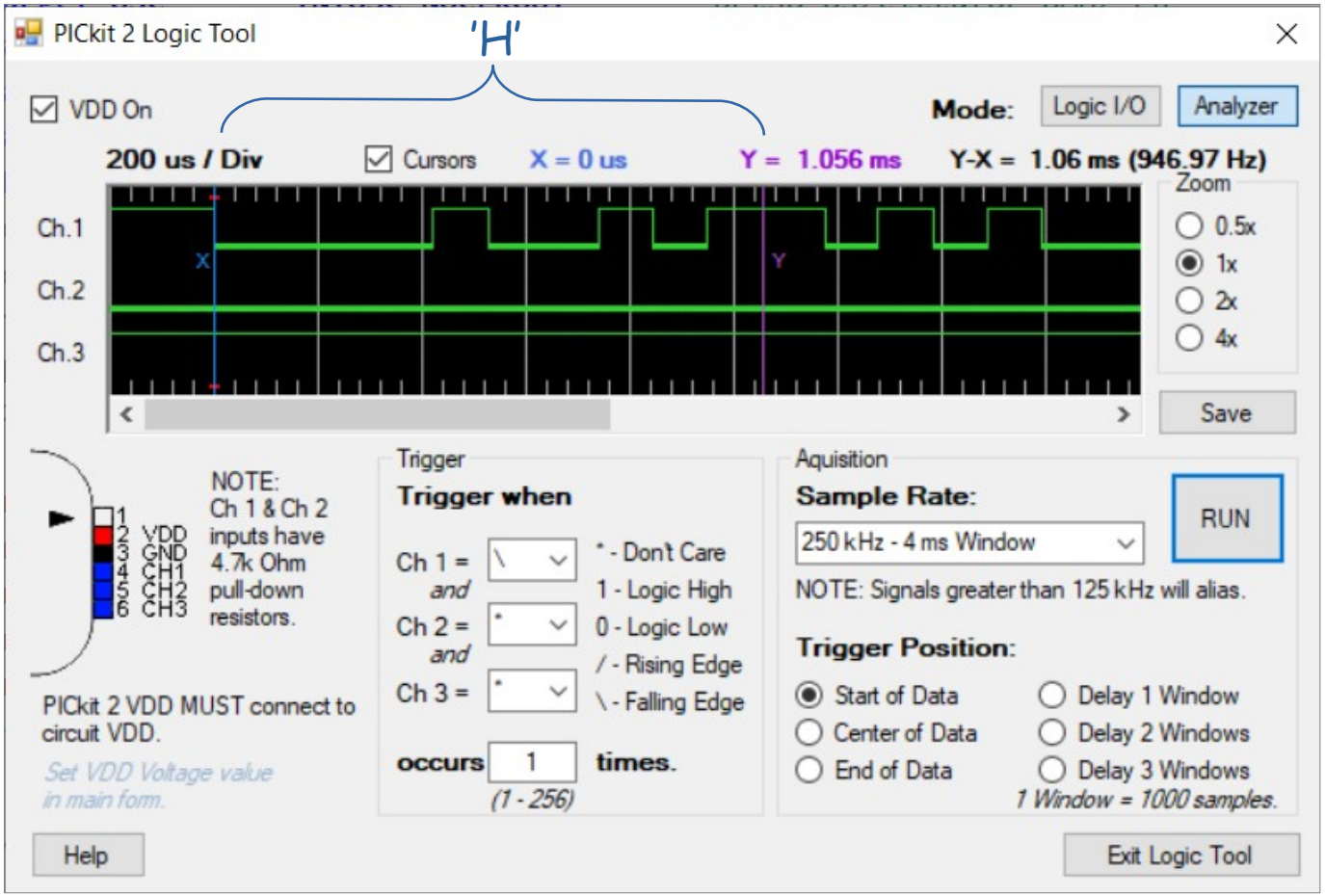
PICkit2, mint logikai analizátor

- A **Logic Tool Analyzer** módban háromcsatornás logikai analizátort biztosít (1 – kijelző, 2 – triggerelés beállítása, 3 – Mintavételezés konfigurálása)



sw_serial_hello.jal futtatása

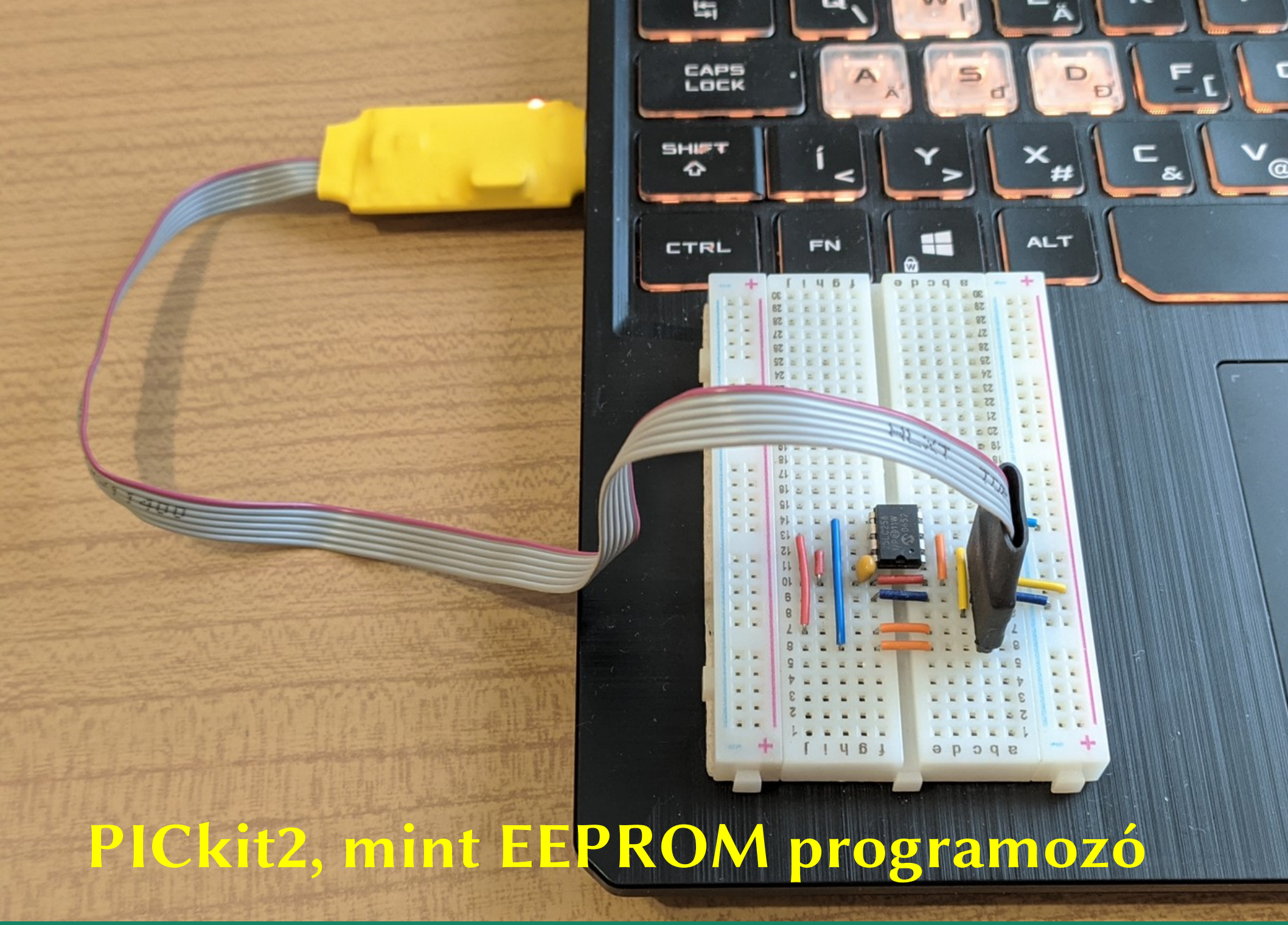
- A 13-14. oldalakon bemutatott **UART** próbát ismételtük meg, de az **UART** jelet itt a logikai analizátor fogadta. A **Cursors** box bejelölése után az egér bal és jobb gombjával helyezhetjük el az időmérő markereket
- A képen látható adatok szerint 10 bitidő = 1.056 ms ⟶ 9469 bit/s



A triggerelés a Pin 4-en (Ch1) észlelt lefutó élre történt, ami szerencsés esetben az első Start bitet jelenti

A mintavételezés gyakorisága 5 kHz-től 1 MHz-ig állítható

A **Run** gomb nyomásakor a trigger esemény 1000 db minta gyűjtését indítja el

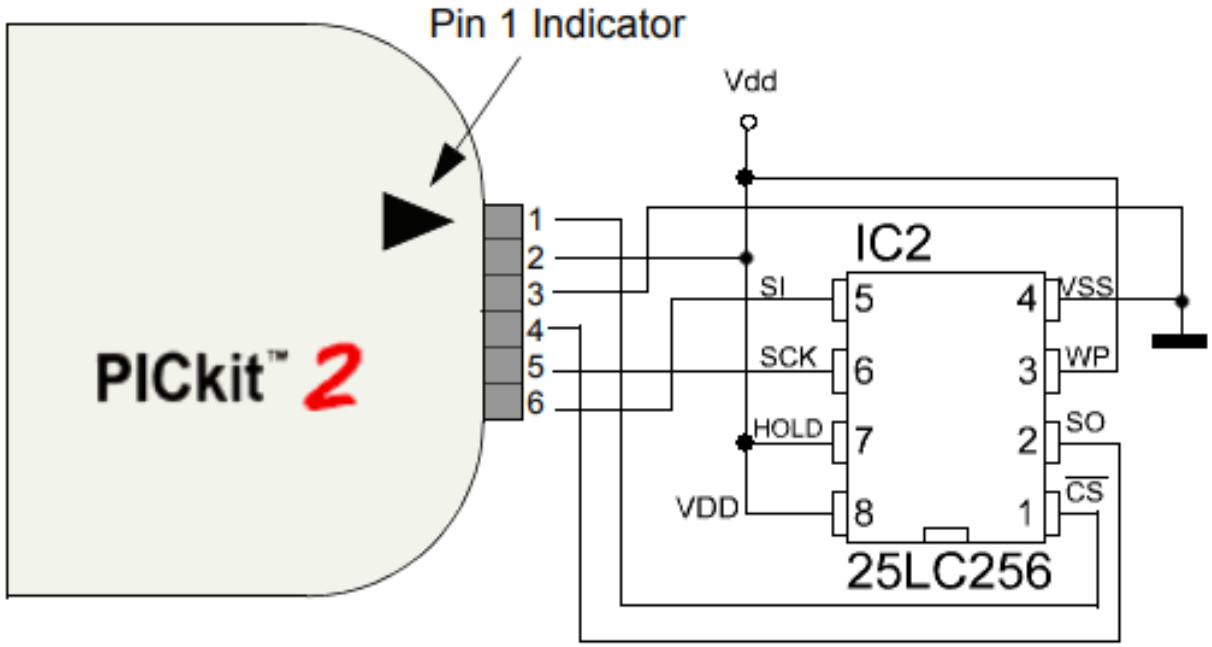


PICkit2, mint EEPROM programozó

EEPROM programozása PICkit2-vel

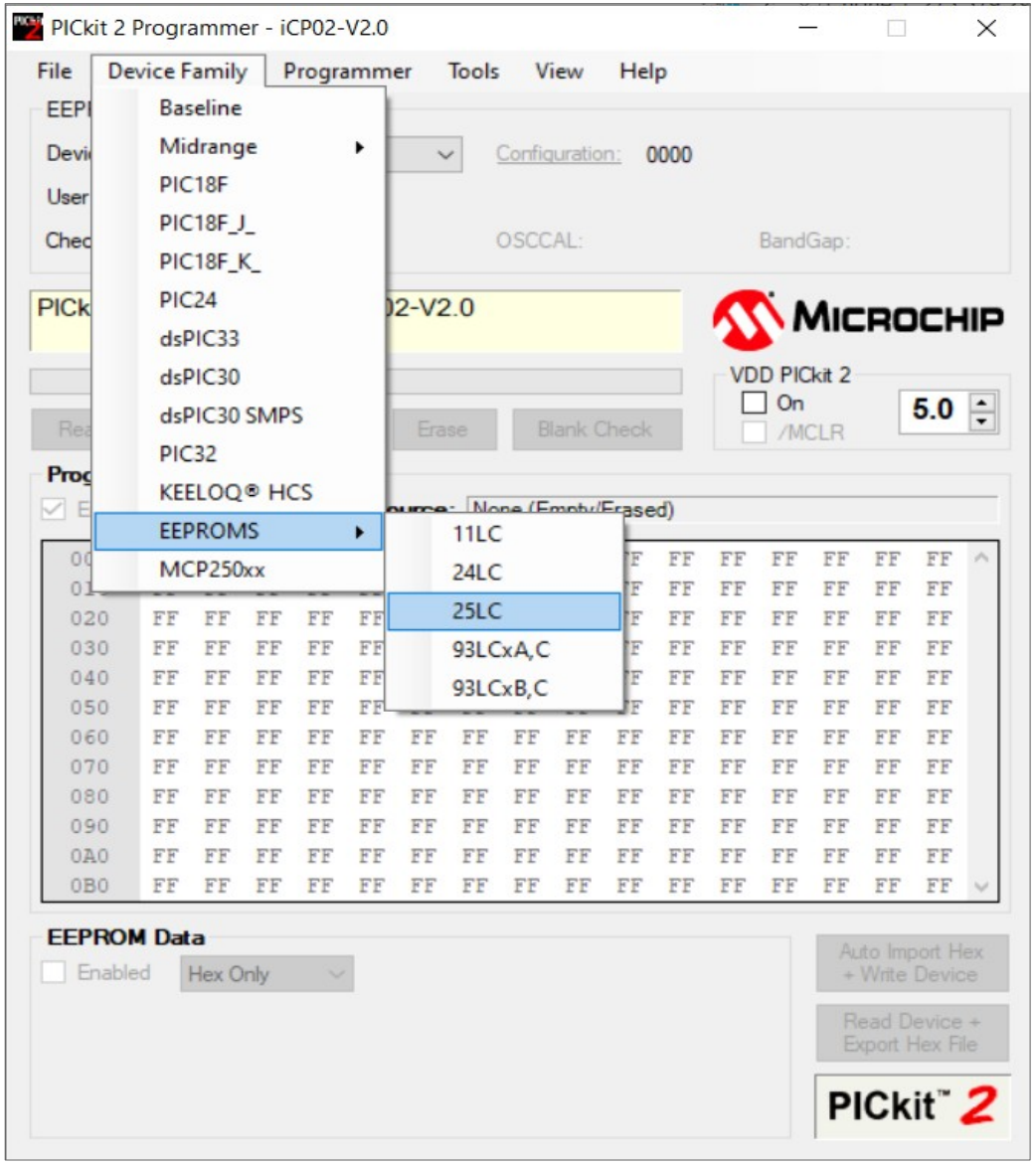
- A **PICkit2** az alkalmazói programjával együtt I2C, illetve SPI EEPROM memóriák kiolvasására és írására is használható
- A **PICkit2** telepítési könyvtárában található **PICkit 2 Readme.txt** állomány találjuk a támogatott eszközök listáját és a bekötési vázlatot
- Az alábbi példában a **25LC256** típusú, **SPI** interfésszel rendelkező IC bekötését, felprogramozását és kiolvasását mutatjuk be. A tápfeszültség a PICkit2-ből, vagy külső forrásból is vehető. A **WP** írásvédelmi és a **HOLD** kommunikációt felfüggesztő kivezetéseket magas szintre (*Vdd*) kell húzni!

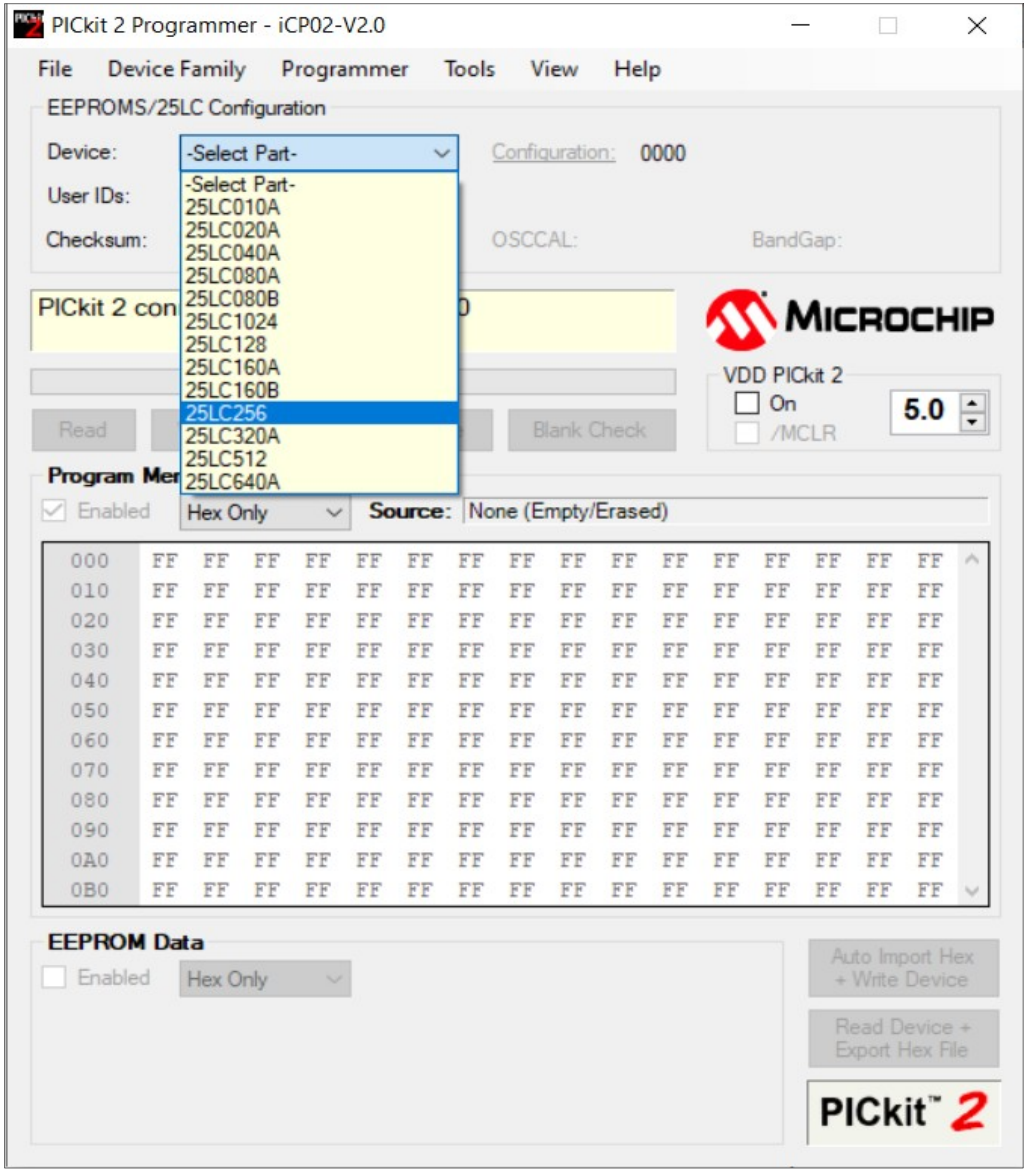
PICkit2	25LC256
1: Vpp	1: $\overline{CS}$
2: Vdd	8: Vdd
3: GND	4: Vss
4: PGD	2: SO
5: PGC	6: SCK
6: AUX	5: SI
to Vdd	3: WP
to Vdd	7: Hold



A PICkit2 beállítása

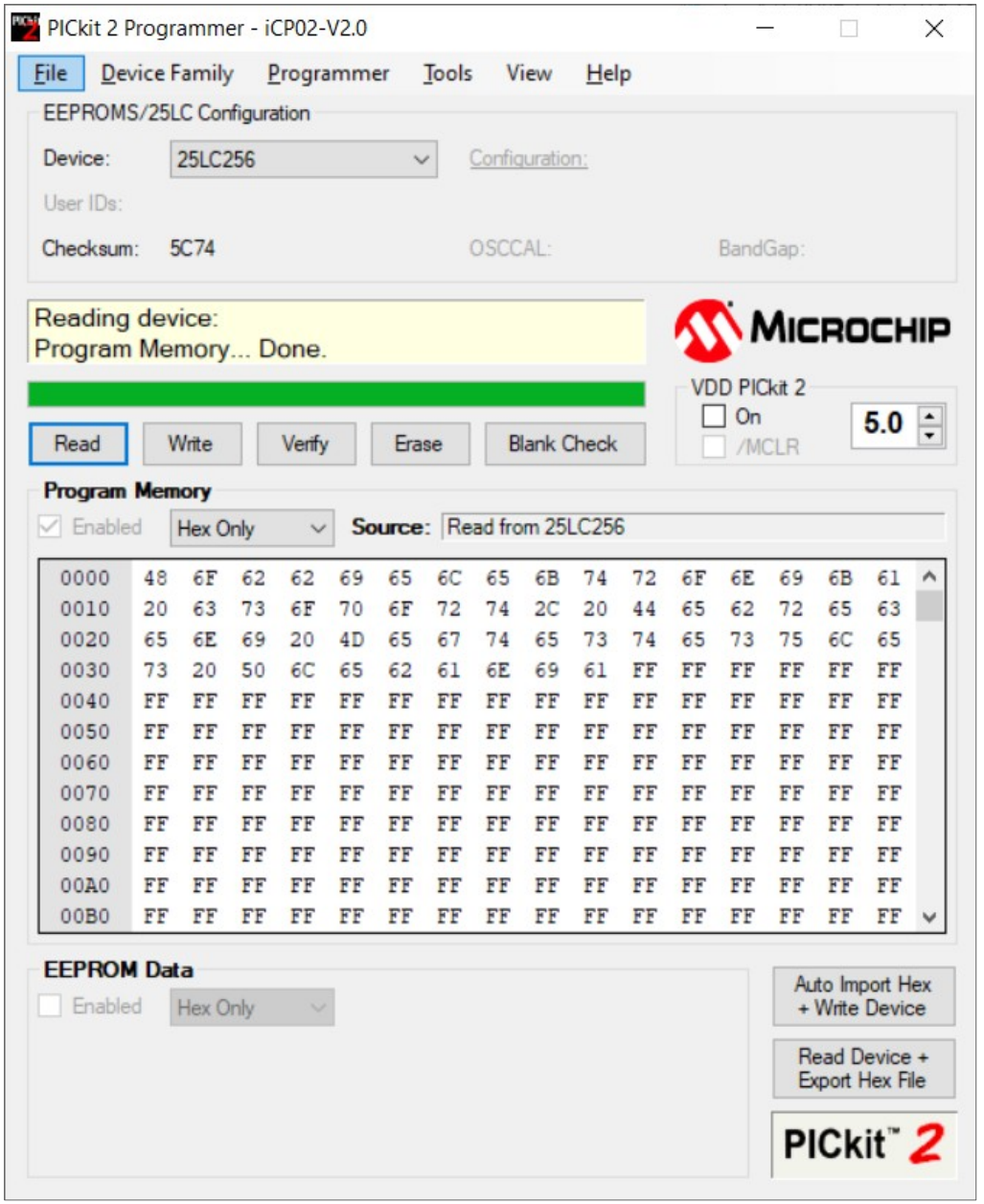
- Csatlakozás után előbb a 25LC eszközcsaládot, majd a típust kell beállítani





Az EEPROM írása/olvasása

- Az EEPROM-ok írása, olvasása, törlése, ellenőrzése ugyanúgy történik, mint a **PIC** mikrovezérlők esetében
- A letöltenő HEX fájlt a **File/Import** menüpontban kell kiválasztani, majd a **Write** gombbal beírni
- Az EEPROM tartalma a **Read** gombbal olvasható ki, s a **File/Export** menü-pontban menthetjük fájlba
- Az ábrán egy korábban kiírt tartalom visszaolvasása látható



make_hex.py

```python
def text_to_hex(text):
    hex_data = [ord(char) for char in text]
    return hex_data

def calculate_checksum(data):
    return (sum(data) & 0xFF) ^ 0xFF

def create_intel_hex_file(hex_data, filename):
    with open(filename, 'w') as file:
        address = 0x0000
        data_record_size = 16  # Maximum number of data bytes per record
        file.write(":020000040000FA\n")        # Extended Linear Address Record
        while hex_data:
            chunk = hex_data[:data_record_size]
            hex_data = hex_data[data_record_size:]
            record = [len(chunk), (address >> 8) & 0xFF, address & 0xFF, 0x00] + chunk
            checksum = calculate_checksum(record)
            record.append(checksum)
            line = ":" + "".join(f"{byte:02X}" for byte in record) + "\n"
            file.write(line)
            address += len(chunk)
        file.write(":00000001FF\n")            # Lezáró sor

text = "Hobbielektronika csoport, Debreceni Megtestesules Plebania"
hex_data = text_to_hex(text)
create_intel_hex_file(hex_data, "output.hex"
```

Ez a Python 3.10-ben írt program a megadott fix szöveget tárolja el nullával lezárt stringként
A HEX fájlkimenet: **output.hex**

```
:020000040000FA
:10000000486F626269656C656B74726F6E696B6172
:100010002063736F706F72742C2044656272656324
:10002000656E69204D6567746573746573756C657C
:0A0030007320506C6562616E696116
:00000001FF
```

A kódolandó szöveg