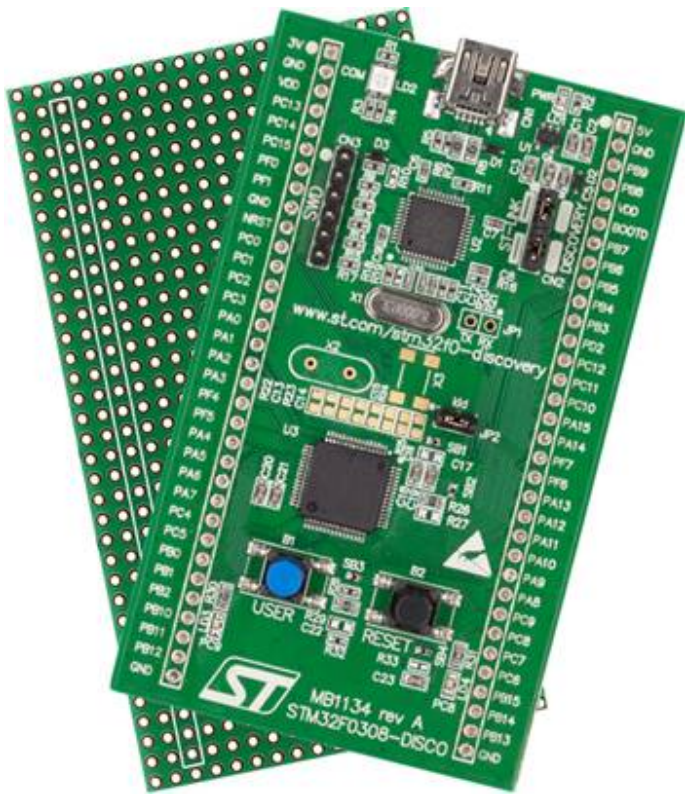


```
main.cpp x
1  #include "mbed.h"
2
3  DigitalOut led1(LED1);
4  DigitalOut led2(LED2);
5
6  int main()
7  {
8      while (1)
9      {
10         led1 = 1;
11         led2 = 0;
12         wait(1.0);
13
14         led1 = 0;
15         led2 = 1;
16         wait(1.0);
17     }
18 }
```



STM32F0-Discovery

STM32F0- Discovery



Az STM32F0DISCOVERY fejlesztőkészlet az **ARM Cortex-M0** magú, 32 bites, STM32F051R8T6 mikrovezérlőn alapul, melynek főbb jellemzői:

- 64KB Flash/8 KB RAM,
- max. 48MHz CPU frekvencia
- 2 UART, 2 SPI, 2 I2C
- 1 db 32-bites időzítő/számláló
- 6 db 16-bites időzítő/számláló
- 16 csatornás 12-bites ADC
- 12-bites DAC
- Real-time óra
- Tápfeszültség/jelszint: 3,3V

Órajel források

Gyárilag nincsenek beültetve a kvarc rezonátorok, így órajel a HSI RC oszcillátorból vagy a programozó és hibavadász egységtől kapható

A kártya a céláramkörön kívül programozó és hibavadász áramkört is tartalmaz, így a hatékony fejlesztés minden feltétele adott.

Nálam a hibavadász áramkör J-Link OB módra át van programozva!!!

A kártya elsősorban tanuláshoz, vagy szerényebb igényű alkalmazások fejlesztéséhez javasolt.

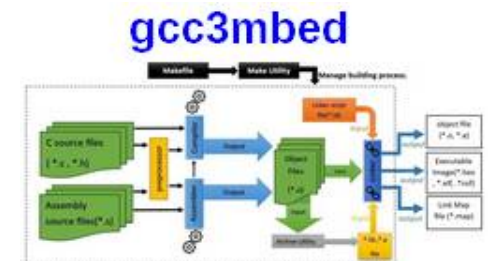
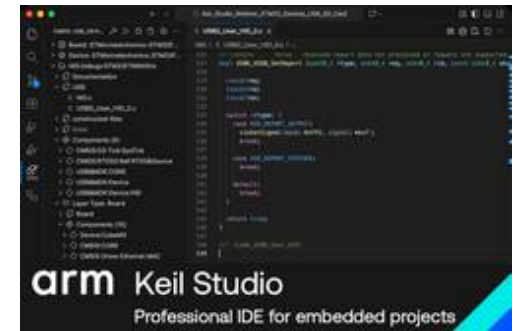
Ajánlott tankönyv: Joseph Yiu: The Definitive Guide to the ARM Cortex-M0

A termék honlapja:

<http://www.st.com/web/catalog/tools/FM116/SC959/SS1532/PF253215>

Ajánlott fejlesztői környezetek

- ❖ **STM32CubeIDE:** Az STMicroelectronics hivatalos, Eclipse-alapú grafikus környezete. Beépített kódgenerátorral (CubeMX), hardverkonfigurátorral és fejlett, natív hibavadászati funkciókkal rendelkezik
- ❖ **Keil Studio Community:** A klasszikus Keil μ Vision modern, VS Code-ba költözött utódja. Az Arm MDK 6 részeként professzionális Arm Compiler-t, teljes körű STM32 eszköz-támogatást, Git-integrációt és CI/CD-re felkészített parancssoros (CLI) vezérlést nyújt
- ❖ **GCC4MBED:** Egy offline, parancssoros (Makefile-alapú) build rendszer és eszközkészlet. **Egyszerű megoldás** az Mbed2 projektek fordításához



GCC4Mbed projekt: Blink2

main.cpp

```
// Két LED villog, ellenütemben
#include "mbed.h"

DigitalOut led1(LED1);
DigitalOut led2(LED2);

int main() {
    while (1) {
        led1 = 1;
        led2 = 0;
        wait(1.0);

        led1 = 0;
        led2 = 1;
        wait(1.0);
    }
}
```

Makefile

```
PROJECT      := Blink2
DEVICES      := DISCO_F051R8
GCC4MBED_DIR := ../..
GCC4MBED_TYPE := Release
MBED_OS_ENABLE := 0
NEWLIB_NANO  := 1
NO_FLOAT_SCANF := 1
NO_FLOAT_PRINTF := 1

include $(GCC4MBED_DIR)/build/gcc4mbed.mk
```

Programletöltés:

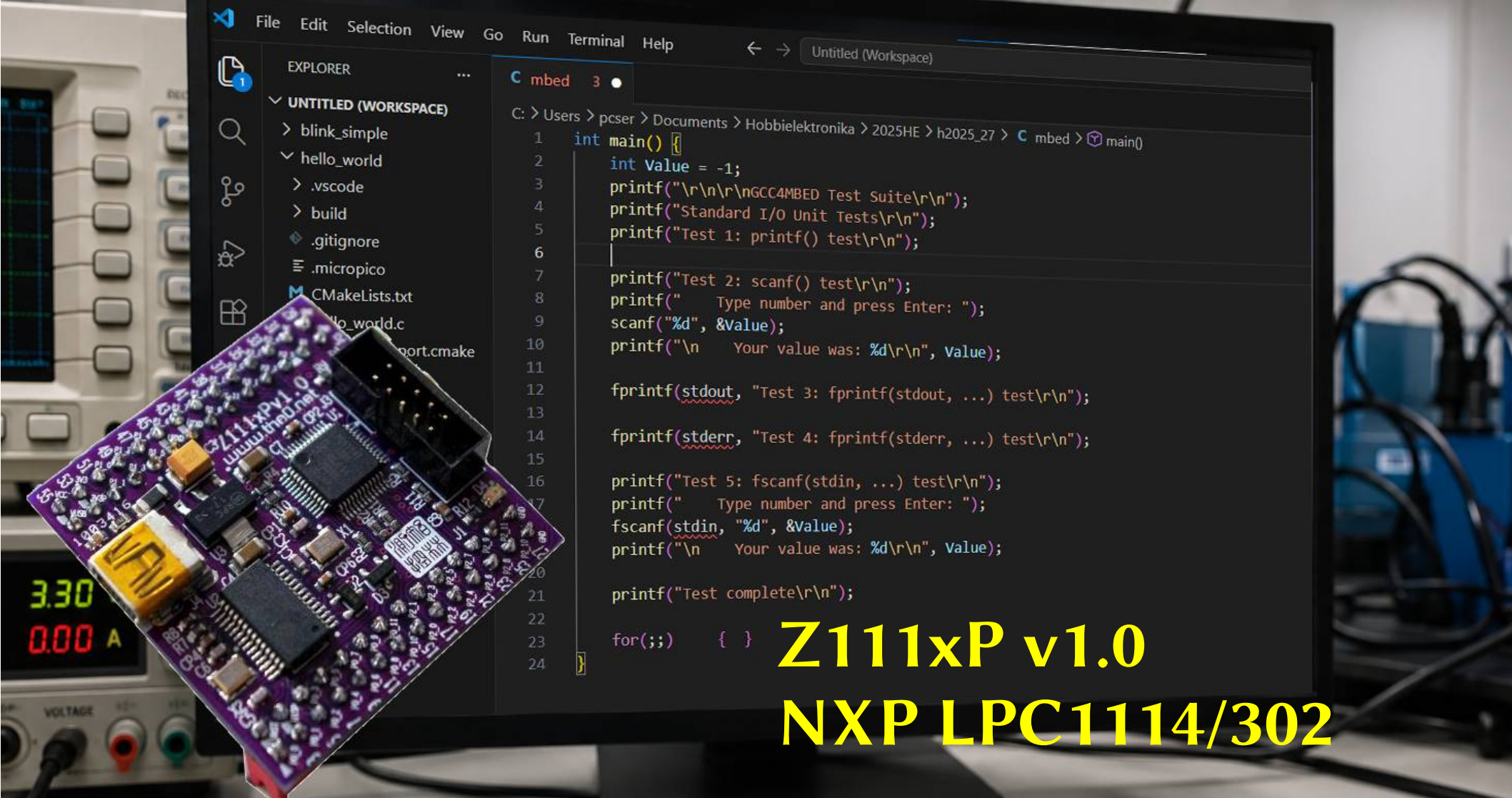
Segger **JFlash.exe** GUI programmal, vagy
Segger **Jlink.exe** programmal (lásd köv. oldal)

Programletöltés

flash.bat

```
@echo off
for %%F in (*.bin) do set BIN=%%F
if "%BIN%"==" " (
    echo Nincs .bin a jelenlegi könyvtárban!
    pause
    exit /b
)
echo erase> flash.jlink
echo loadbin %BIN%,0x08000000>> flash.jlink
echo reset>> flash.jlink
echo g>> flash.jlink
echo q>> flash.jlink
JLink.exe -device STM32F051R8 -if SWD -speed 4000 -AutoConnect 1 -CommandFile flash.jlink
del flash.jlink
pause
```

A kártyára épített és a **J-Link OB** firmware-rel átprogramozott hibavadással vagy a **JFlash.exe** programot, vagy a fenti batch fájlal indított **Jlink.exe** programot használhatjuk. A **flash.bat** állománynak és a Segger Jlink mappának az elérési útvonalban kell lennie. A **flash.bat** állományt abban a könyvtárban kell indítani, ahol a letölteni kívánt .bin állomány van

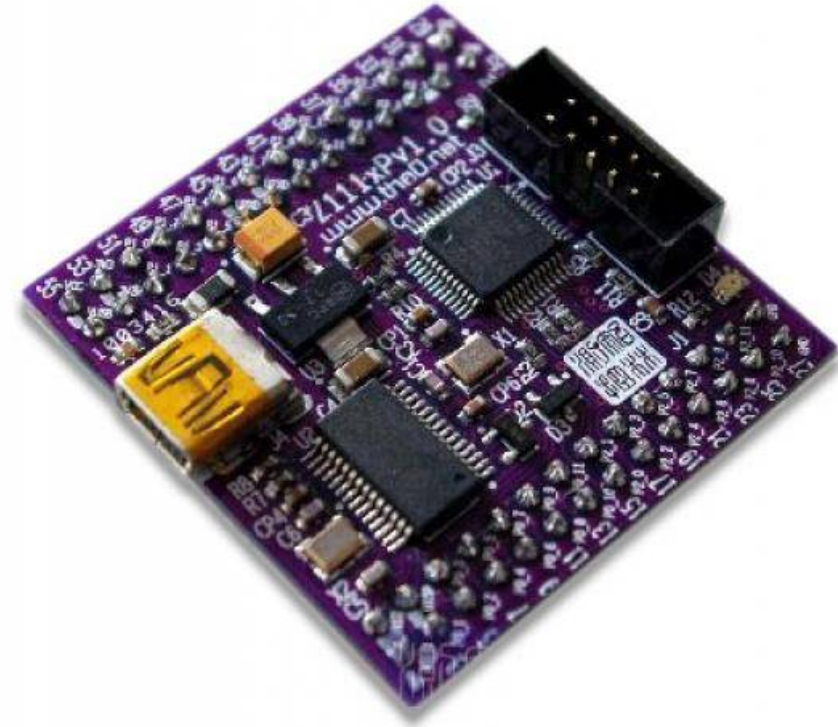


```
C mbed 3 ●
C: > Users > pcser > Documents > Hobbielektronika > 2025HE > h2025_27 > C mbed > main()
1 int main() {
2     int Value = -1;
3     printf("\r\n\r\nGCC4MBED Test Suite\r\n");
4     printf("Standard I/O Unit Tests\r\n");
5     printf("Test 1: printf() test\r\n");
6
7     printf("Test 2: scanf() test\r\n");
8     printf("    Type number and press Enter: ");
9     scanf("%d", &Value);
10    printf("\n    Your value was: %d\r\n", Value);
11
12    fprintf(stdout, "Test 3: fprintf(stdout, ...) test\r\n");
13
14    fprintf(stderr, "Test 4: fprintf(stderr, ...) test\r\n");
15
16    printf("Test 5: fscanf(stdin, ...) test\r\n");
17    printf("    Type number and press Enter: ");
18    fscanf(stdin, "%d", &Value);
19    printf("\n    Your value was: %d\r\n", Value);
20
21    printf("Test complete\r\n");
22
23    for(;;) { }
24 }
```

Z111xP v1.0

NXP LPC1114/302

Z111xP v1.0 kártya (NXP LPC1114/302)



A kártya **ARM Cortex-M0** magú, 32 bites, NXP **LPC1114** mikrovezérlőt tartalmaz, melynek főbb jellemzői:

- 32KB Flash/8 KB RAM,
- max. 50 MHz CPU frekvencia
- 1 UART, 2 SPI, 1 I2C
- 2 db 32-bites időzítő/számláló
- 2 db 16-bites időzítő/számláló
- 8 csatornás 10-bites ADC
- Real-time óra
- Tápfeszültség/jelszint: 3,3V

A kártya a mikrovezérlőn kívül egy USB/UART átalakítót tartalmaz, amely kommunikációra és a programok letöltésre (a mikrovezérlőbe gyárilag betöltött bootloaderrel) használható. A panel egy 10 pólusú mini JTAG (2 mm-es osztású) csatlakozón keresztül szabványos JTAG programozó és hibavadász készülékhez is csatlakoztatható (pl. JLink).

A kártya elsősorban tanuláshoz, vagy szerényebb igényű alkalmazások fejlesztéséhez javasolt.

Ajánlott tankönyv: Joseph Yiu: The Definitive Guide to the ARM Cortex-M0

GCC4Mbed projekt: HelloWorld

main.cpp

```
// A beépített LED1 villogtatása

#include "mbed.h"

DigitalOut myled(LED1); // P1_5

int main()
{
    while(1)
    {
        myled = 1;
        wait(0.2);
        myled = 0;
        wait(0.2);
    }
}
```

Makefile

```
PROJECT      := HelloWorld
DEVICES      := LPC1114
GCC4MBED_DIR := ../..
GCC4MBED_TYPE := Release
MBED_OS_ENABLE := 0
NEWLIB_NANO  := 1
NO_FLOAT_SCANF := 1
NO_FLOAT_PRINTF := 1
```

```
include $(GCC4MBED_DIR)/build/gcc4mbed.mk
```

Programletöltés:

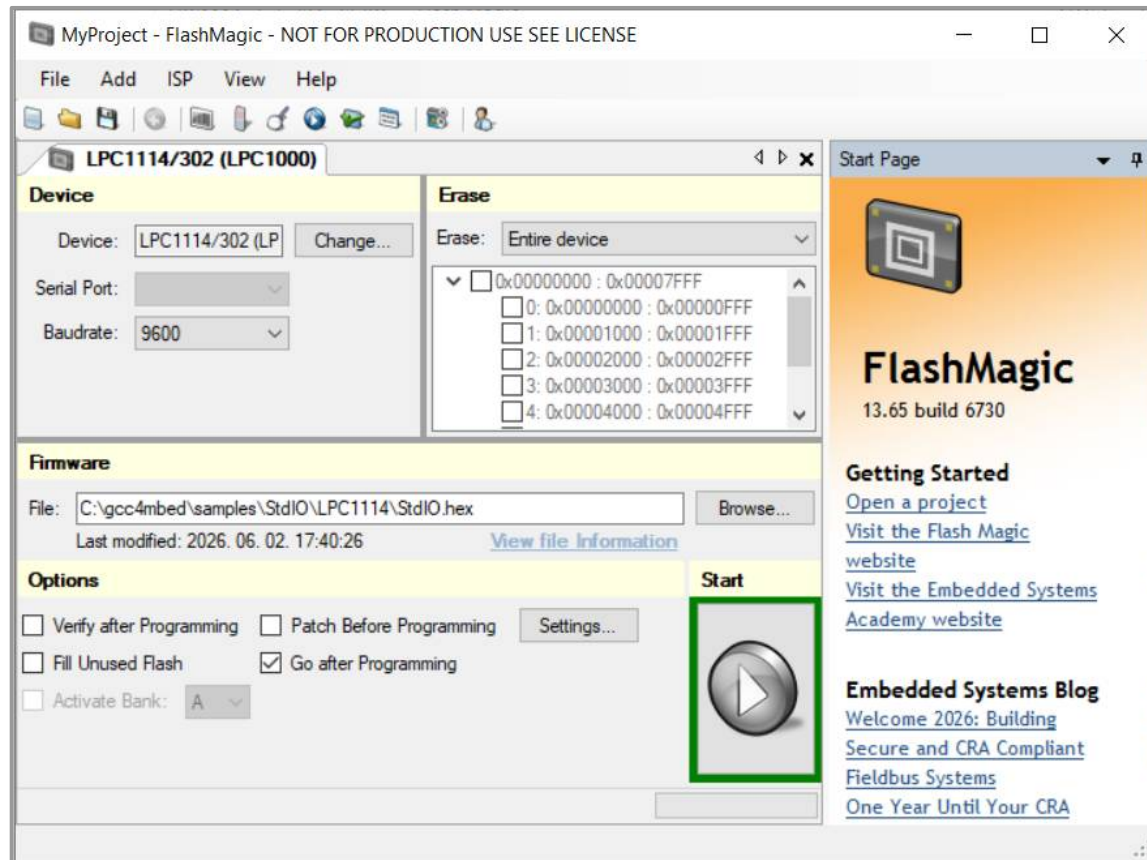
Segger **Jflash.exe** programmal (ha van Jlink)
FlashMagic programmal (bootloader módban)

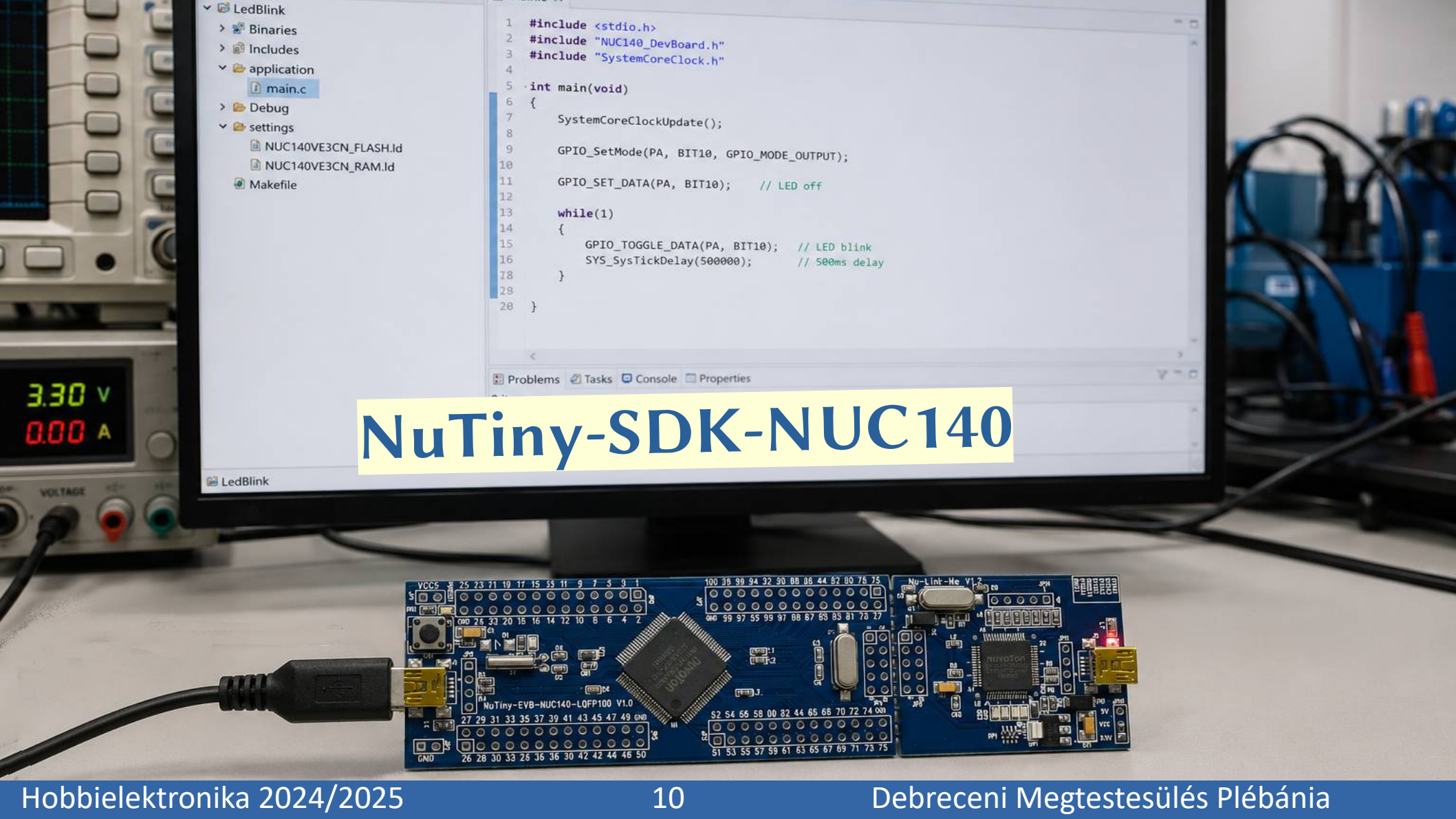
Programletöltés

A [Flash Magic program](#) az **LPC1114** mikrovezérlő gyárilag beépített ISP bootloadert használja, amely UART-on keresztül fogadja a programozási parancsokat

A **Z111xP v1.0** kártyán egy **PL2303** USB–soros átalakító biztosítja a PC és az MCU közti kommunikációt, és a hardver úgy van kialakítva, hogy a **PL2303** vezérelni tudja a mikrovezérlő RESET és ISP lábait is

Ennek köszönhetően a Flash Magic automatikusan bootloader módba tudja kapcsolni az MCU-t, majd az RX/TX vonalakon keresztül rátölti a kiválasztott fájl tartalmát





- ▼ LedBlink
 - ▼ Binaries
 - ▼ Includes
 - ▼ application
 - main.c
 - ▼ Debug
 - ▼ settings
 - NUC140VE3CN_FLASH.ld
 - NUC140VE3CN_RAM.ld
 - Makefile

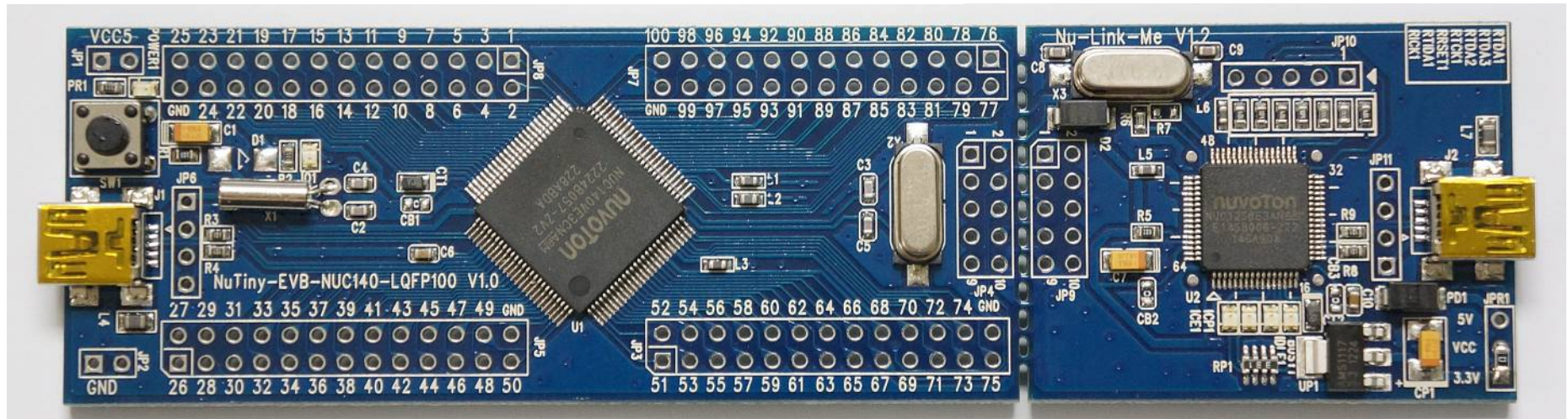
```
1 #include <stdio.h>
2 #include "NUC140_DevBoard.h"
3 #include "SystemCoreClock.h"
4
5 int main(void)
6 {
7     SystemCoreClockUpdate();
8
9     GPIO_SetMode(PA, BIT10, GPIO_MODE_OUTPUT);
10
11    GPIO_SET_DATA(PA, BIT10);    // LED off
12
13    while(1)
14    {
15        GPIO_TOGGLE_DATA(PA, BIT10);    // LED blink
16        SYS_SysTickDelay(500000);    // 500ms delay
17    }
18
19 }
20 }
```

NuTiny-SDK-NUC140

A kártya **ARM Cortex-M0** magú, USB eszköz és CAN kommunikációs képességekkel is rendelkező 32 bites **NUC140VE3CN** mikrovezérlőt tartalmaz, melynek főbb jellemzői:

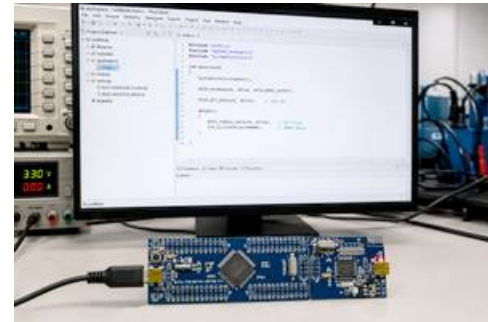
- 128KB Flash/16 KB RAM,
- max. 50 MHz CPU frekvencia
- CAN modul
- USB device kommunikáció (FSUSB)
- 3 UART, 4 SPI, 2 I2C
- 4 db 32-bites időzítő/számláló capture/compare/PWM csatornával
- 8 csatornás 12-bites ADC
- Real-time óra
- Tápfeszültség/jelszint: 3,3V – 5V

A kártya CAN, UART, SPI és USB 2.0 kommunikációt (USB HID, USB CDC, stb.) igénylő fejlesztésekhez ajánlható, s ahol lényeges előnyt jelent a beállítható digitális I/O jelszint (3.3 V – 5 V) értéke

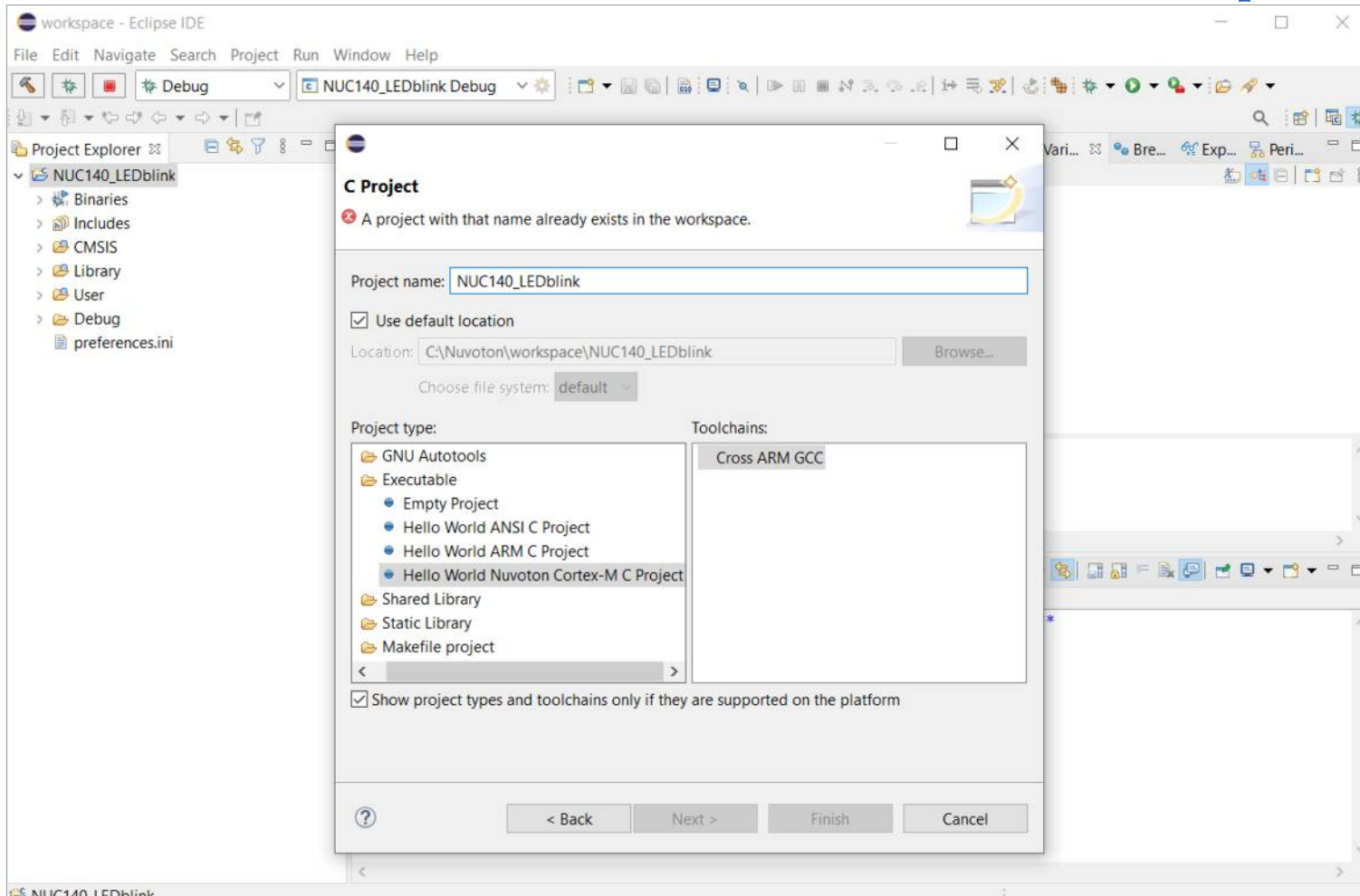


Ajánlott fejlesztői környezetek

- ❖ **NuEclipse:** A Nuvoton által karbantartott Eclipse-alapú fejlesztőkörnyezet, GCC toolchainnel és **NUC100BSP**-támogatással. Közvetlenül kezeli a Nu-Link debug eszközt, a NUC140 sorozathoz jelenleg ez biztosítja a legteljesebb, gyártó-specifikus támogatást
- ❖ **Keil Studio Community:** A klasszikus Keil μ Vision modern, VS Code-ba költözött utódja. Az Arm MDK 6 részeként professzionális Arm Compiler-t, Git-integrációt és CI/CD-re felkészített parancssoros (CLI) vezérlést nyújt



NUC140_LEDblink (NuEclipse projekt)



NUC140_LEDblink (NuEclipse projekt)

```
#define NUC140VE3CN
#include "NUC100Series.h"

void SYS_Init(void) {
    SYS_UnlockReg();
    CLK->PWRCON |= CLK_PWRCON_IRC22M_EN_Msk;
    CLK_WaitClockReady(CLK_CLKSTATUS_IRC22M_STB_Msk);
    CLK->CLKSEL0 = (CLK->CLKSEL0 & ~CLK_CLKSEL0_HCLK_S_Msk) | CLK_CLKSEL0_HCLK_S_HIRC;
    CLK_SetCoreClock(50000000);
    SYS_LockReg();
}

int main(void) {
    SYS_Init();
    PA10 = GPIO_PMD_OUTPUT;
    SysTick->CTRL = SysTick_CTRL_CLKSOURCE_Msk | SysTick_CTRL_ENABLE_Msk;
    While(1) {
        PA10 = 1;
        CLK_SysTickLongDelay(500000);
        PA10 = 0;
        CLK_SysTickLongDelay(500000);
    }
}
```