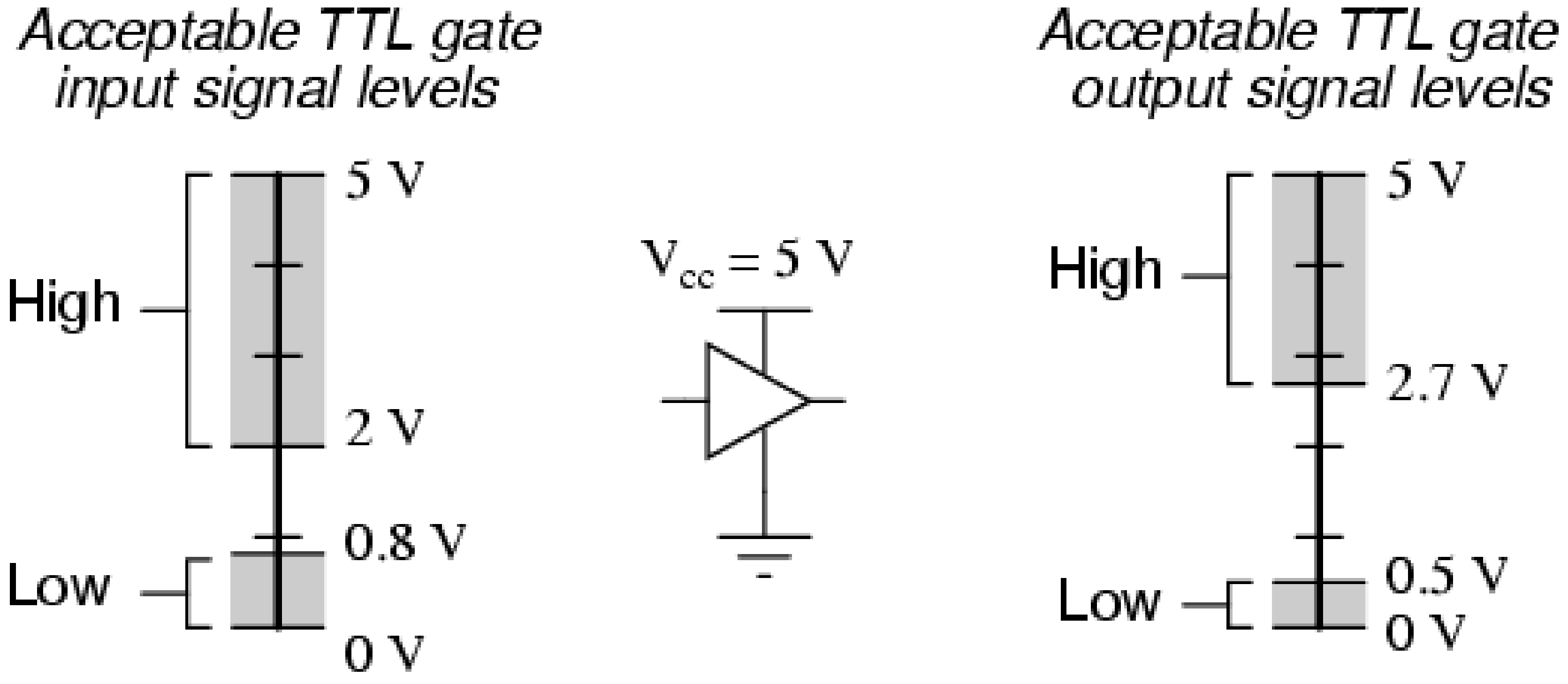
Beágyazott és érzékelő alapú rendszerek
STM32F4 Discovery 2.

GPIO

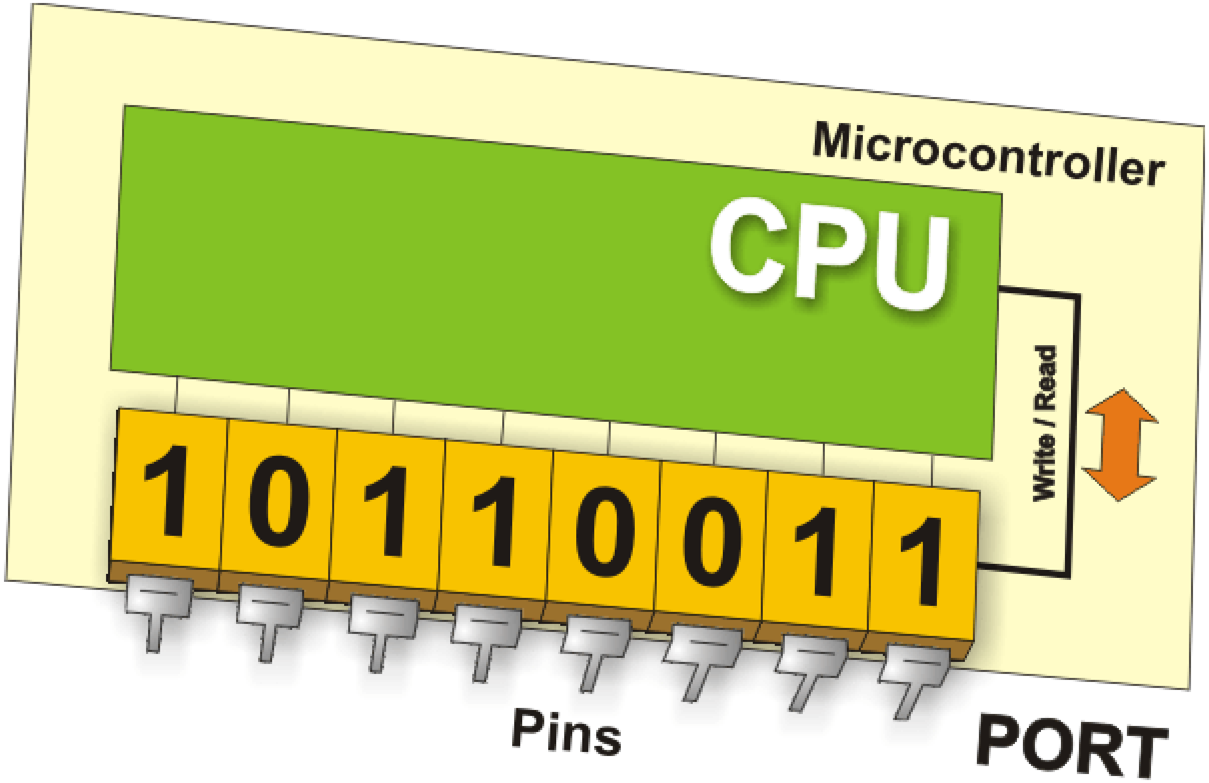
Logikai jelszintek



Acceptable TTL gate input signal levels

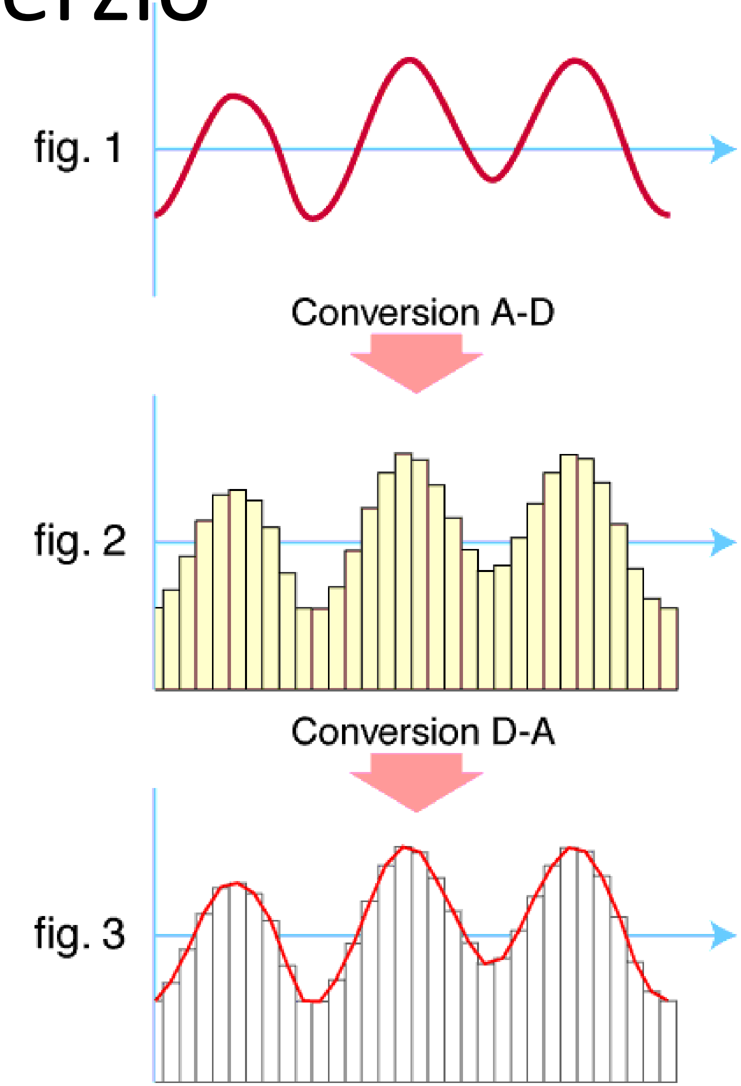
Acceptable TTL gate output signal levels

Digitális I/O

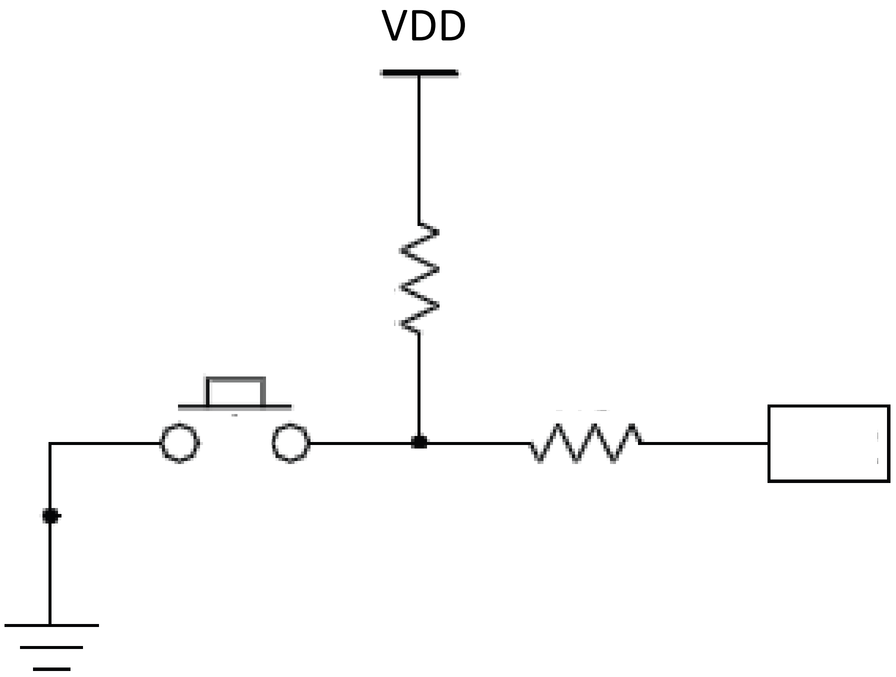


Analóg I/O:
A/D D/A konverzió

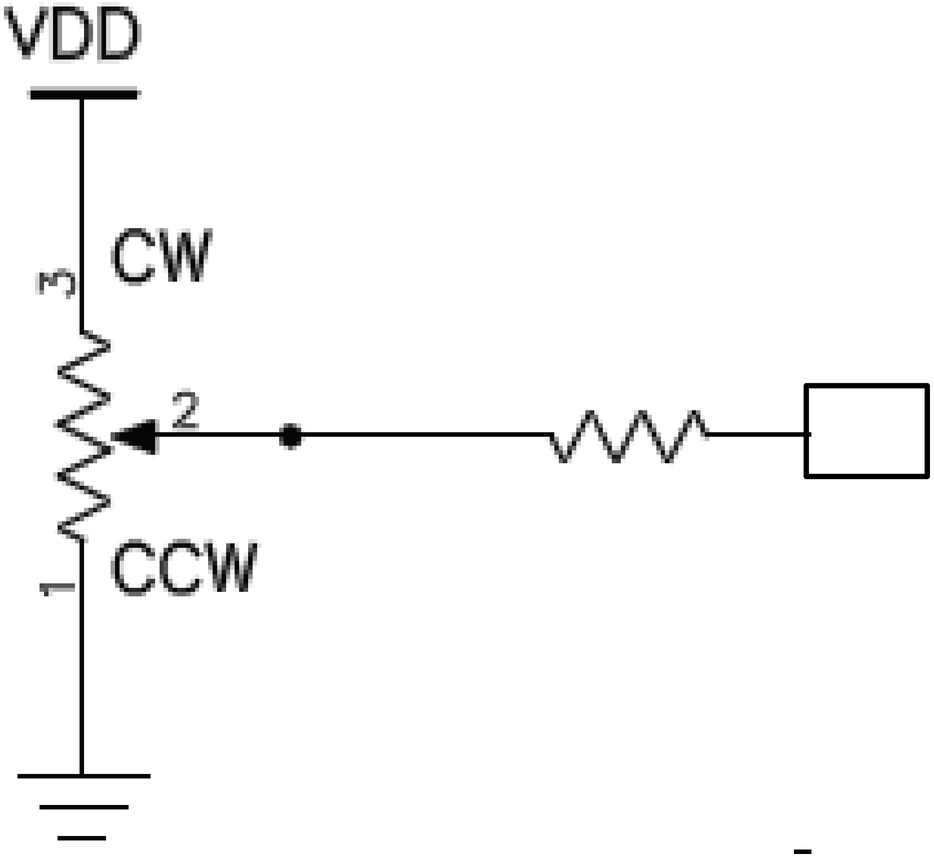
- Pl. 10 biten 1024 érték



Billentyű

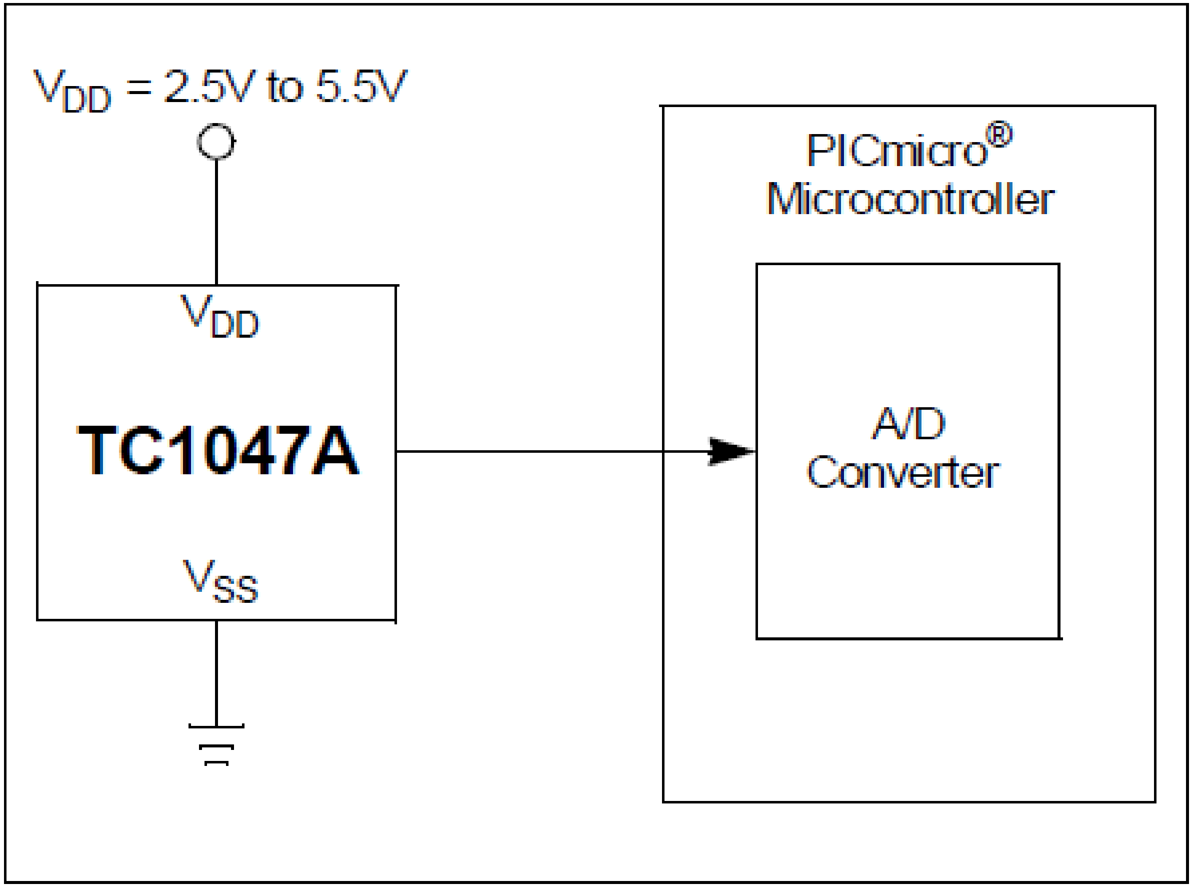


Potmeter



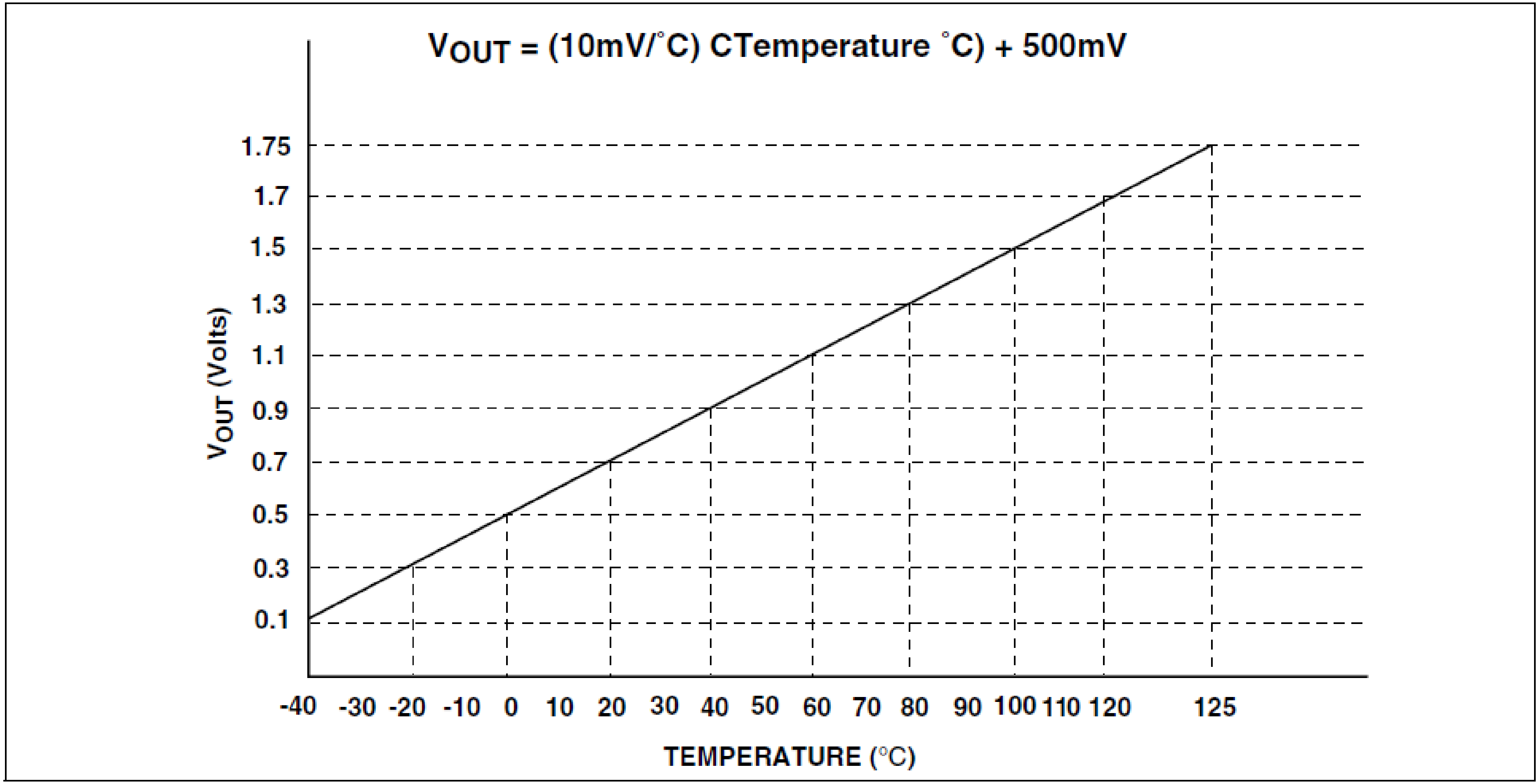
Hőszenzor

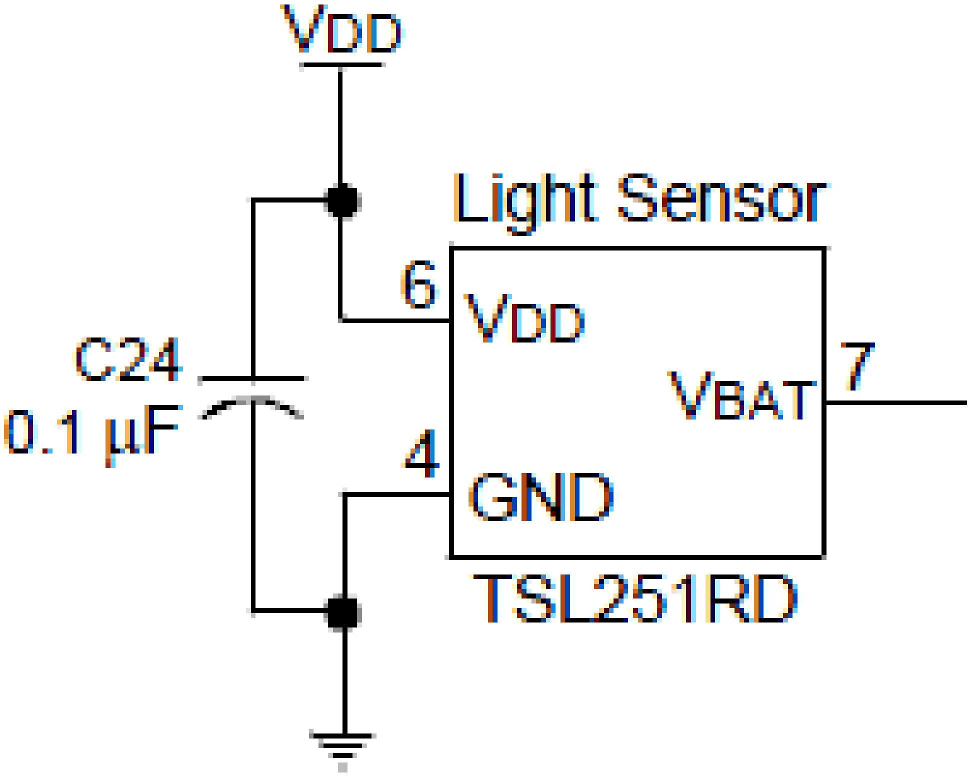
Block Diagram



Hőszenzor

FIGURE 3-1: OUTPUT VOLTAGE VS. TEMPERATURE

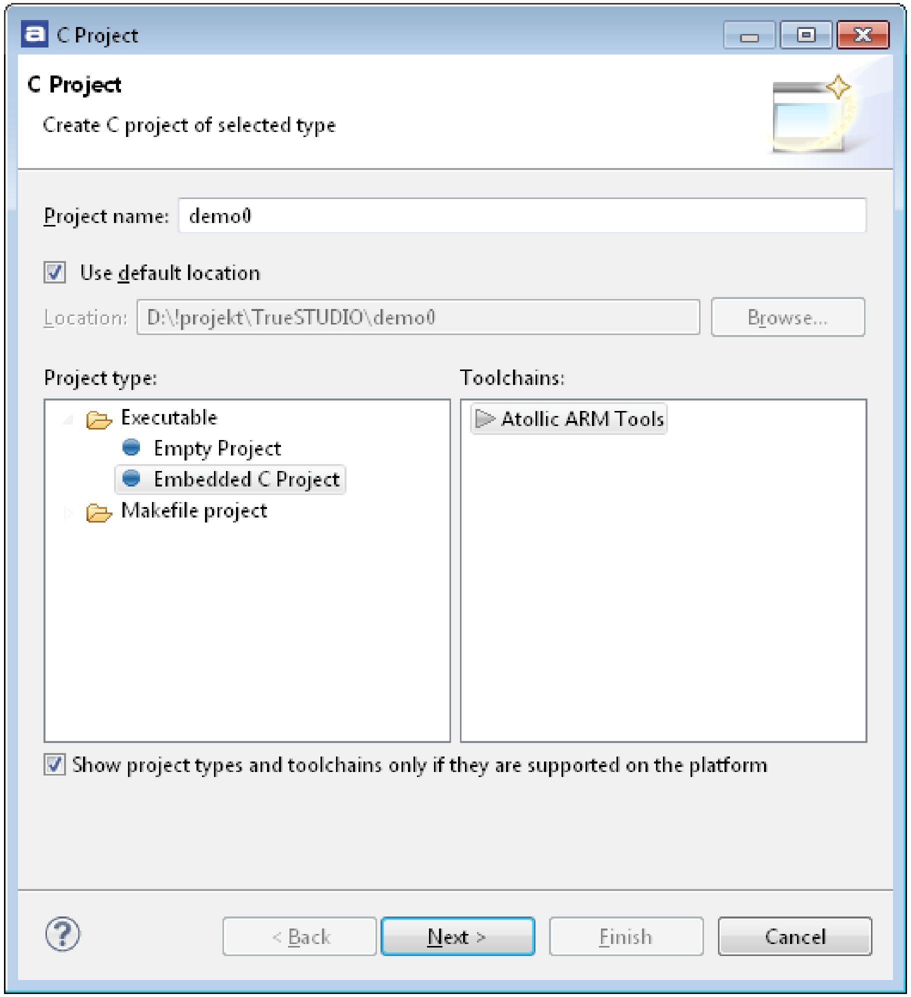



VDD
Light Sensor
6
VDD
C24
0.1 µF
7
VBAT
4
GND
TSL251RD

Atollic TrueStudio

Új projekt

- Atollic TrueSTUDIO
  - – File / New / C Project



C Project

C Project

Create C project of selected type

Project name: demo0

☑ Use default location

Location: D:\!projekt\TrueSTUDIO\demo0 Browse...

Project type:

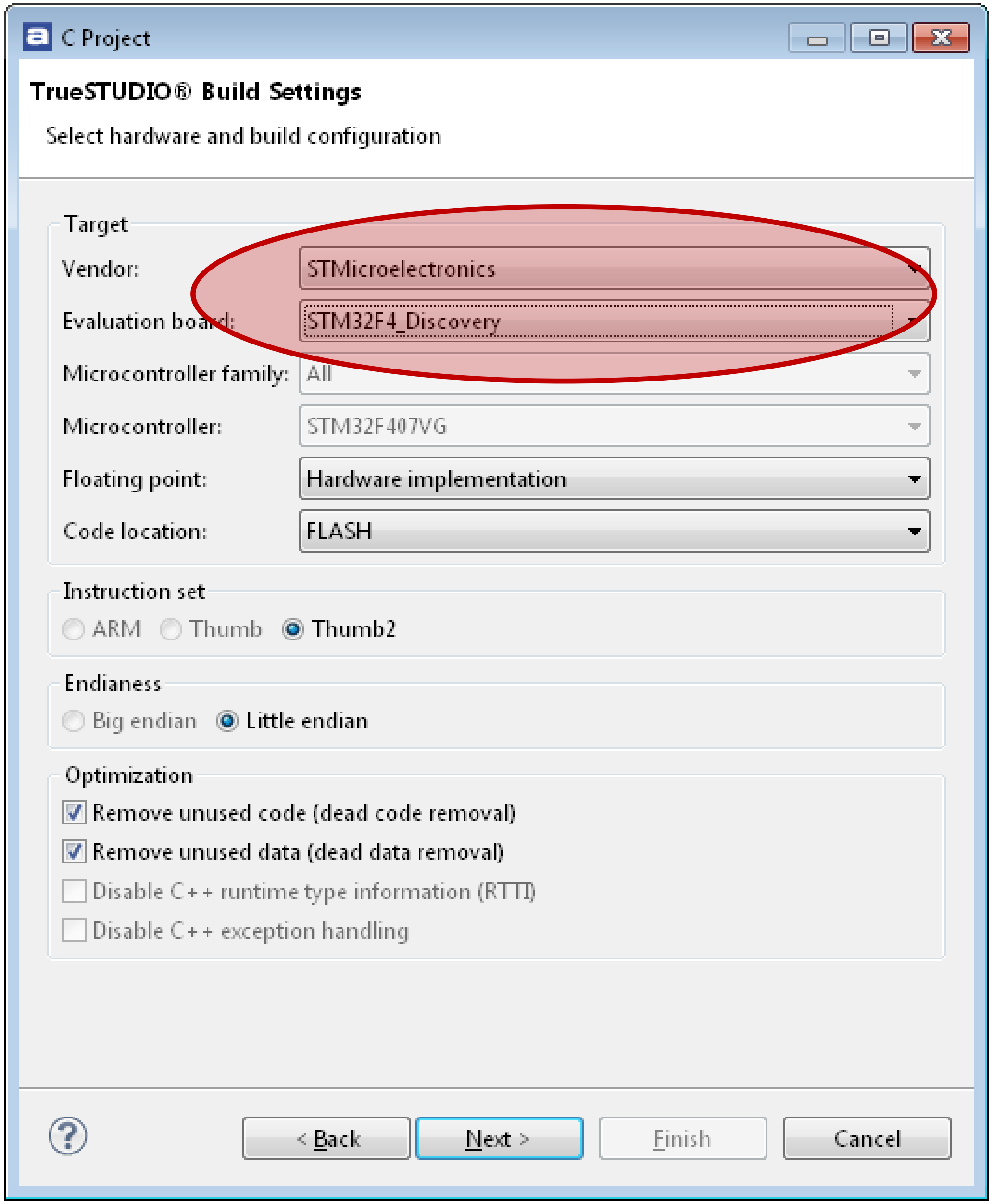
- Executable
  - Empty Project
  - Embedded C Project
- Makefile project

Toolchains:

- Atollic ARM Tools

☑ Show project types and toolchains only if they are supported on the platform

< Back | Next > | Finish | Cancel



TrueSTUDIO® Build Settings

Select hardware and build configuration

Target

Vendor: STMicroelectronics

Evaluation board: STM32F4_Discovery

Microcontroller family: All

Microcontroller: STM32F407VG

Floating point: Hardware implementation

Code location: FLASH

Instruction set

○ ARM ○ Thumb ● Thumb2

Endianess

○ Big endian ● Little endian

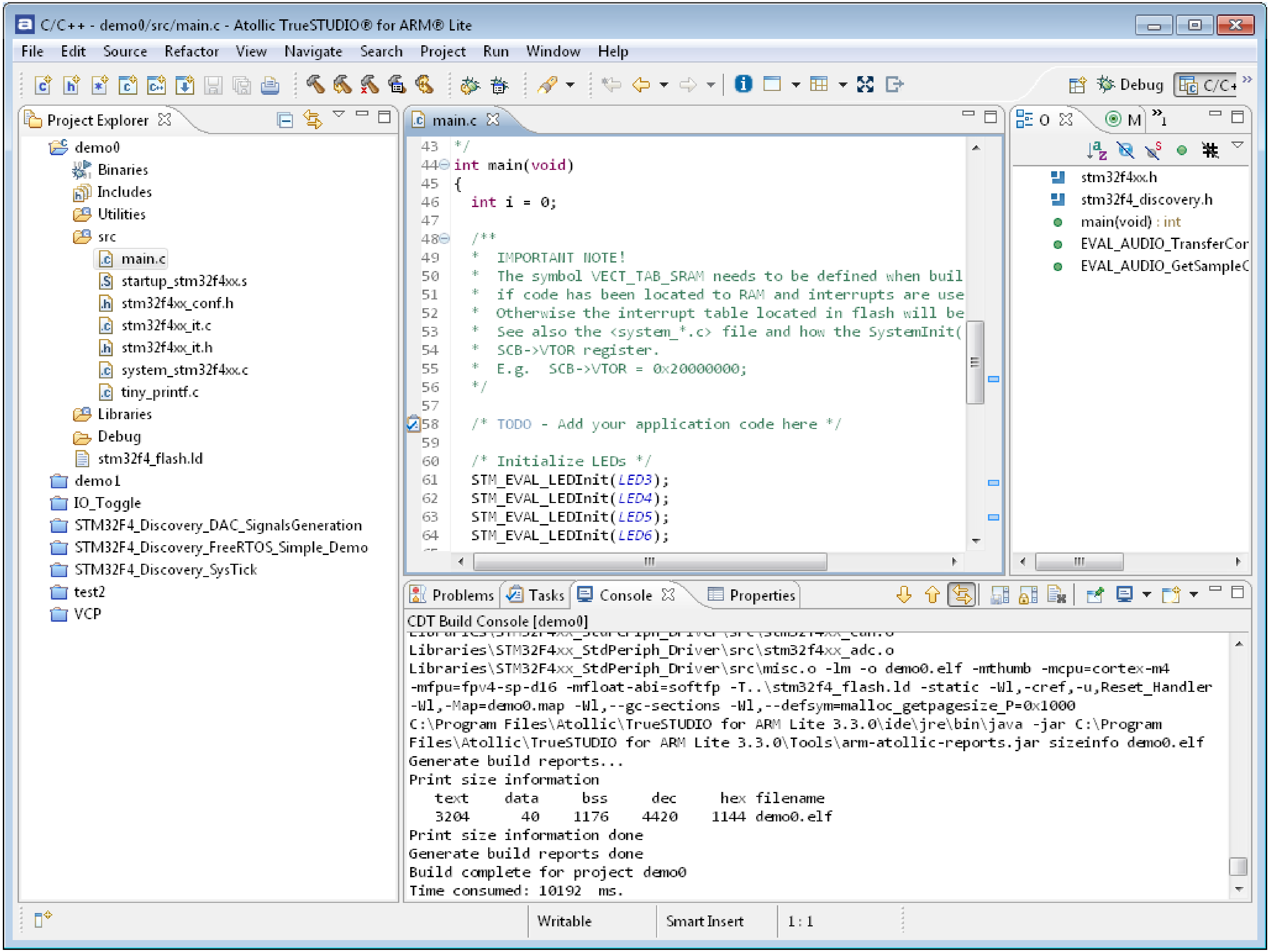
Optimization

☑ Remove unused code (dead code removal)

☑ Remove unused data (dead data removal)

□ Disable C++ runtime type information (RTTI)

□ Disable C++ exception handling



```
*/
int main(void)
{
  int i = 0;

  /**
  *  IMPORTANT NOTE!
  *  The symbol VECT_TAB_SRAM needs to be defined when buil
  *  if code has been located to RAM and interrupts are use
  *  Otherwise the interrupt table located in flash will be
  *  See also the <system_*.c> file and how the SystemInit(
  *  SCB->VTOR register.
  *  E.g.  SCB->VTOR = 0x20000000;
  */

  /* TODO - Add your application code here */

  /* Initialize LEDs */
  STM_EVAL_LEDInit(LED3);
  STM_EVAL_LEDInit(LED4);
  STM_EVAL_LEDInit(LED5);
  STM_EVAL_LEDInit(LED6);
```

```
Libraries\STM32F4xx_StdPeriph_Driver\src\stm32f4xx_can.o
Libraries\STM32F4xx_StdPeriph_Driver\src\stm32f4xx_adc.o
Libraries\STM32F4xx_StdPeriph_Driver\src\misc.o -lm -o demo0.elf -mthumb -mcpu=cortex-m4
-mfpu=fpv4-sp-d16 -mfloat-abi=softfp -T..\stm32f4_flash.ld -static -Wl,-cref,-u,Reset_Handler
-Wl,-Map=demo0.map -Wl,--gc-sections -Wl,--defsym=malloc_getpagesize_P=0x1000
C:\Program Files\Atollic\TrueSTUDIO for ARM Lite 3.3.0\ide\jre\bin\java -jar C:\Program
Files\Atollic\TrueSTUDIO for ARM Lite 3.3.0\Tools\arm-atollic-reports.jar sizeinfo demo0.elf
Generate build reports...
Print size information
   text    data     bss     dec     hex filename
   3204      40    1176    4420    1144 demo0.elf
Print size information done
Generate build reports done
Build complete for project demo0
Time consumed: 10192  ms.
```

Main.c

- /* Includes */
- #include "stm32f4xx.h"
- #include "stm32f4_discovery.h"

F3

stm32f4xx.h

```
* @brief   CMSIS Cortex-M4 Device Peripheral Access Layer Header File.
  *          This file contains all the peripheral register's definitions, bits
  *          definitions and memory mapping for STM32F4xx devices.
  *
  *          The file is the unique include file that the application programmer
  *          is using in the C source code, usually in main.c. This file contains:
  *           - Configuration section that allows to select:
  *              - The device used in the target application
  *              - To use or not the peripheral’s drivers in application code(i.e.
  *                code will be based on direct access to peripheral’s registers
  *                rather than drivers API), this option is controlled by
  *                "#define USE_STDPERIPH_DRIVER"
  *              - To change few application-specific parameters such as the HSE
  *                crystal frequency
  *           - Data structures and the address mapping for all peripherals
  *           - Peripheral's registers declarations and bits definition
  *           - Macros to access peripheral’s registers hardware
```

stm32f4_discovery.h

```
* @brief   This file contains definitions for STM32F4-Discovery Kit's Leds and
*          push-button hardware resources.
```

stm32f4_discovery.c

```
* @brief   This file provides set of firmware functions to manage Leds and
*          push-button available on STM32F4-Discovery Kit from STMicroelectronics.
```

stm32f4xx_it.h
stm32f4xx_it.c

Main Interrupt Service Routines.

```
void HardFault_Handler(void)
{
  /* Go to infinite loop when Hard
  Fault exception occurs */
  while (1)
  {
  }
}
```

```c
int main(void)
{
  int i = 0;

  /* TODO - Add your application code here */

  /* Initialize LEDs */
  STM_EVAL_LEDInit(LED3);
  STM_EVAL_LEDInit(LED4);
  STM_EVAL_LEDInit(LED5);
  STM_EVAL_LEDInit(LED6);

  /* Turn on LEDs */
  STM_EVAL_LEDOn(LED3);
  STM_EVAL_LEDOn(LED4);
  STM_EVAL_LEDOn(LED5);
  STM_EVAL_LEDOn(LED6);

  /* Infinite loop */
  while (1)
  {
    i++;
  }
}
```

F3

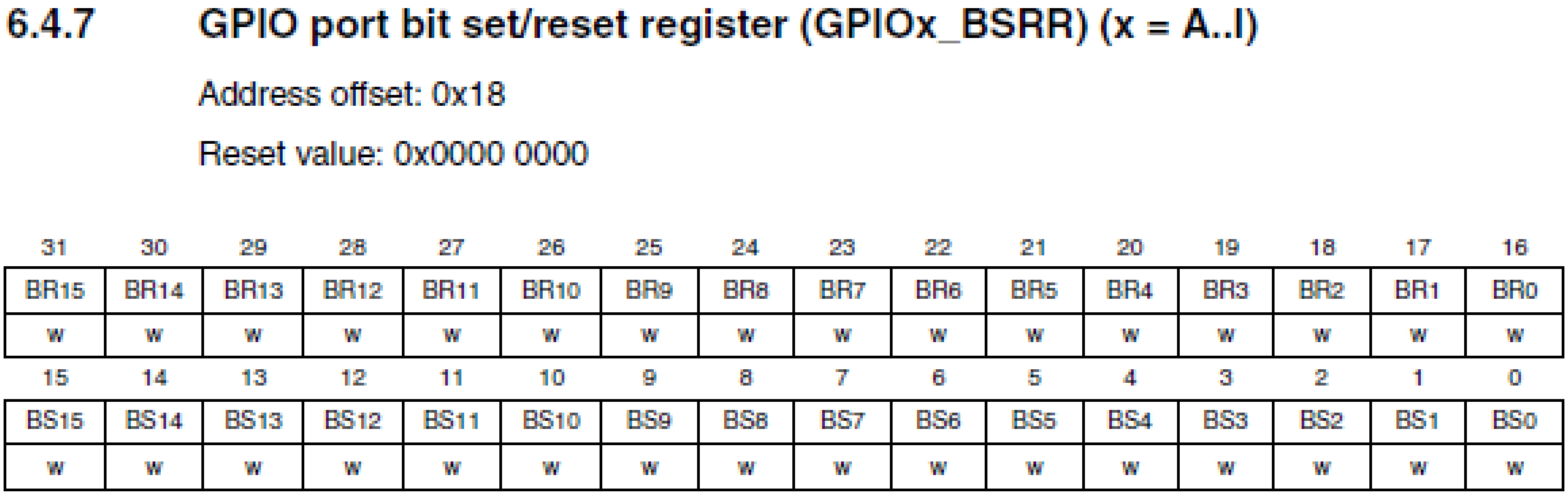
F3

```
void STM_EVAL_LEDInit(Led_TypeDef Led)
{
  GPIO_InitTypeDef  GPIO_InitStructure;

  /* Enable the GPIO_LED Clock */
  RCC_AHB1PeriphClockCmd(GPIO_CLK[Led], ENABLE);

  /* Configure the GPIO_LED pin */
  GPIO_InitStructure.GPIO_Pin = GPIO_PIN[Led];
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
  GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
  GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_UP;
  GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
  GPIO_Init(GPIO_PORT[Led], &GPIO_InitStructure);
}
```

```
/**
  * @brief  Turns selected LED On.
  * @param  Led: Specifies the Led to be set on.
  *   This parameter can be one of following parameters:
  *     @arg LED4
  *     @arg LED3
  *     @arg LED5
  *     @arg LED6
  * @retval None
  */
void STM_EVAL_LEDOn(Led_TypeDef Led)
{
  GPIO_PORT[Led]->BSRRL = GPIO_PIN[Led];
}
```



6.4.7 GPIO port bit set/reset register (GPIOx_BSRR) (x = A..I)

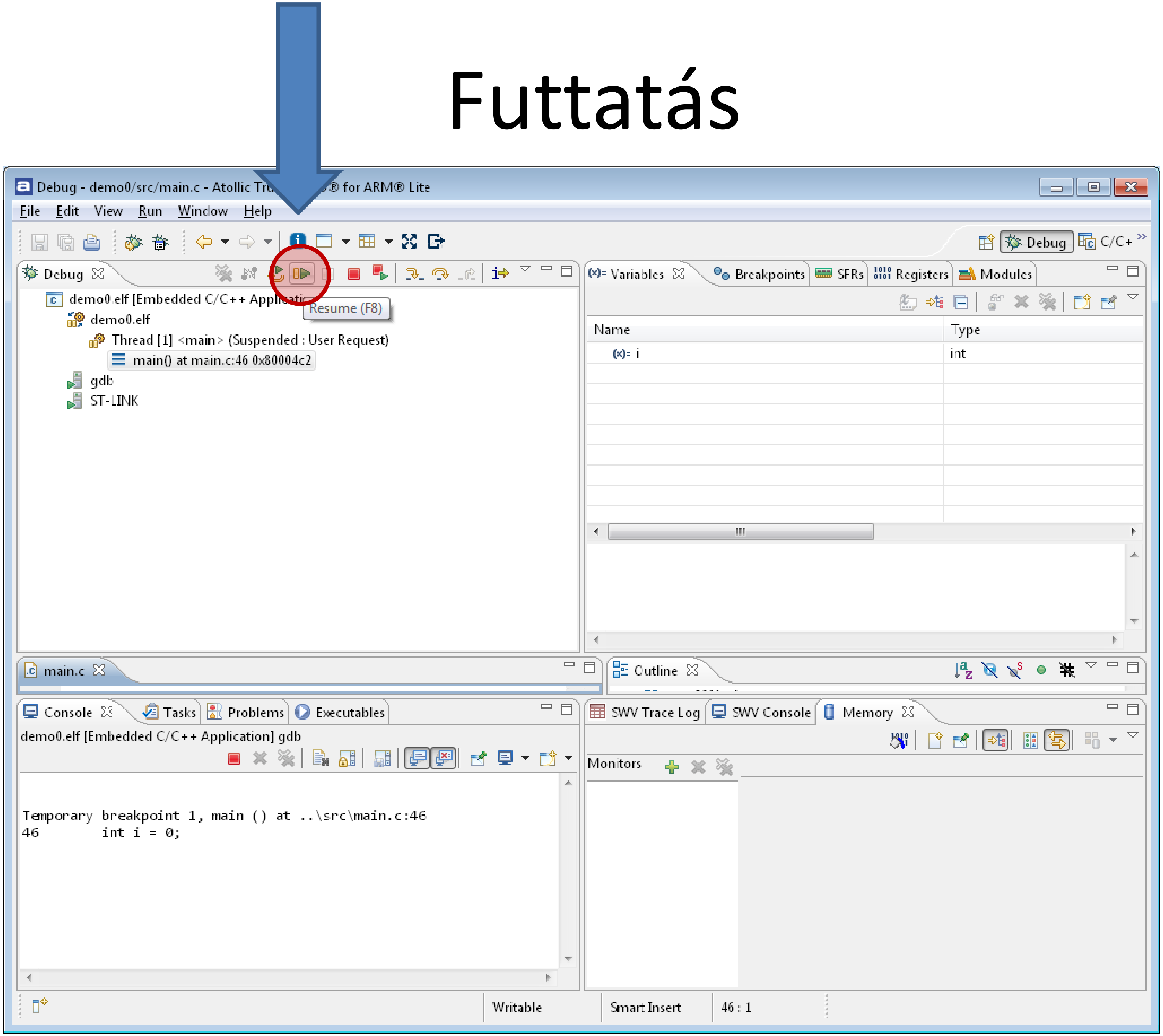
Address offset: 0x18

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Futtatás



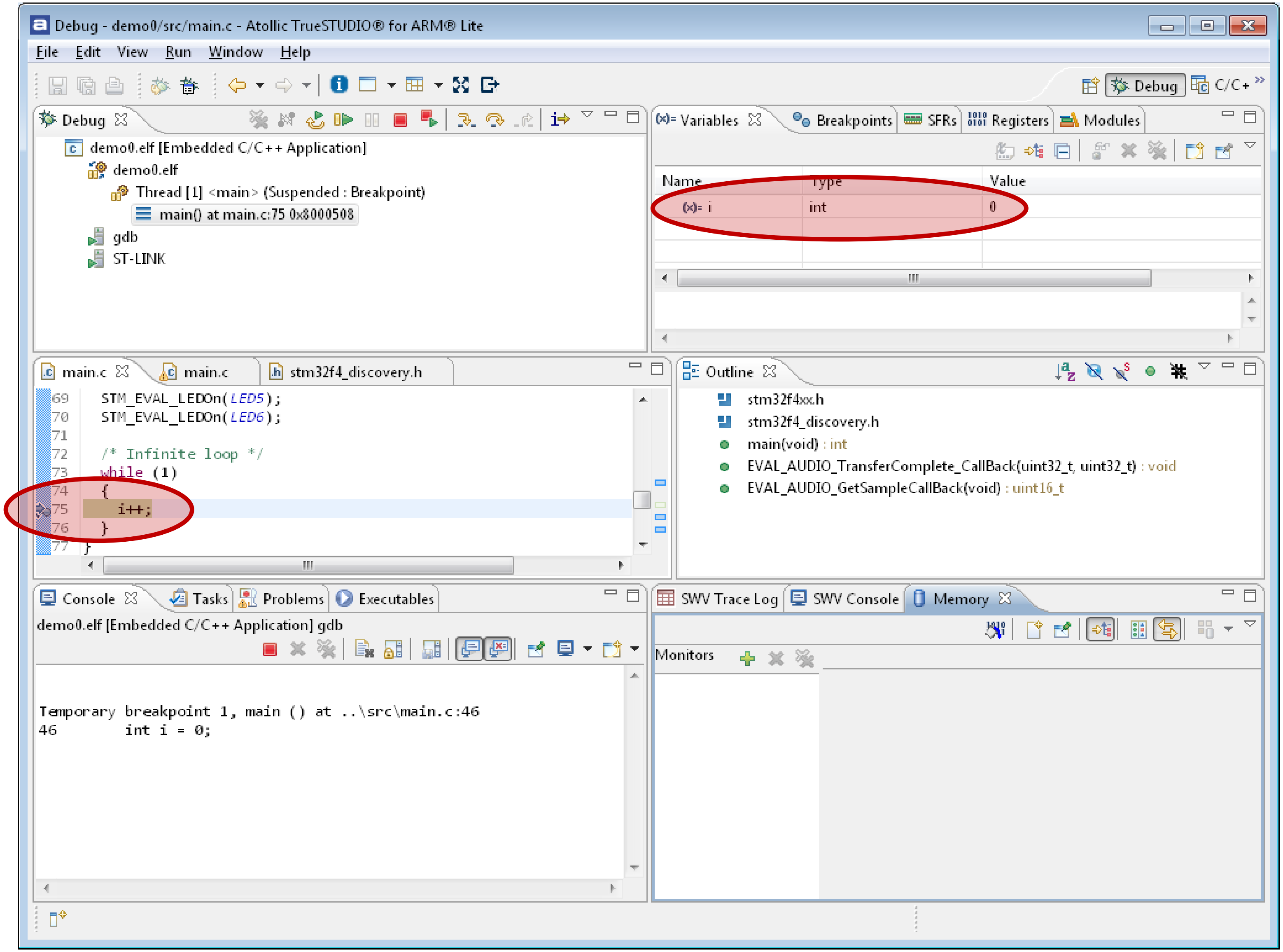
```
demo0.elf [Embedded C/C++ Appli
  demo0.elf
    Thread [1] <main> (Suspended : User Request)
      main() at main.c:46 0x80004c2
  gdb
  ST-LINK
```

Name	Type
(x)= i	int

demo0.elf [Embedded C/C++ Application] gdb

```
Temporary breakpoint 1, main () at ..\src\main.c:46
46        int i = 0;
```

Debug



Debug - demo0/src/main.c - Atollic TrueSTUDIO® for ARM® Lite

```
demo0.elf [Embedded C/C++ Application]
  demo0.elf
    Thread [1] <main> (Suspended : Breakpoint)
      main() at main.c:75 0x8000508
  gdb
  ST-LINK
```

Name	Type	Value
(x)= i	int	0

```
    STM_EVAL_LEDOn(LED5);
    STM_EVAL_LEDOn(LED6);

    /* Infinite loop */
    while (1)
    {
      i++;
    }
}
```

- stm32f4xx.h
- stm32f4_discovery.h
- main(void) : int
- EVAL_AUDIO_TransferComplete_CallBack(uint32_t, uint32_t) : void
- EVAL_AUDIO_GetSampleCallBack(void) : uint16_t

demo0.elf [Embedded C/C++ Application] gdb

```
Temporary breakpoint 1, main () at ..\src\main.c:46
46        int i = 0;
```

Gomb kezelés

GPIO

LED csak lenyomott gomb mellett világít

STM_EVAL_LEDInit(LED6); után:

```
  /*Initialize UserButton*/
  STM_EVAL_PBInit(BUTTON_USER, BUTTON_MODE_GPIO);
```

F3

```
  /* Infinite loop */
  while (1)
  {
  if (STM_EVAL_PBGetState(BUTTON_USER)!=0x00)
    {
    /* Turn on LEDs */
    STM_EVAL_LEDOn(LED3);
    STM_EVAL_LEDOn(LED4);
    STM_EVAL_LEDOn(LED5);
    STM_EVAL_LEDOn(LED6);
    }
    else
    {
    /* Turn off LEDs */
    STM_EVAL_LEDOff(LED3);
    STM_EVAL_LEDOff(LED4);
    STM_EVAL_LEDOff(LED5);
    STM_EVAL_LEDOff(LED6);
    }
  }
}
```

F3

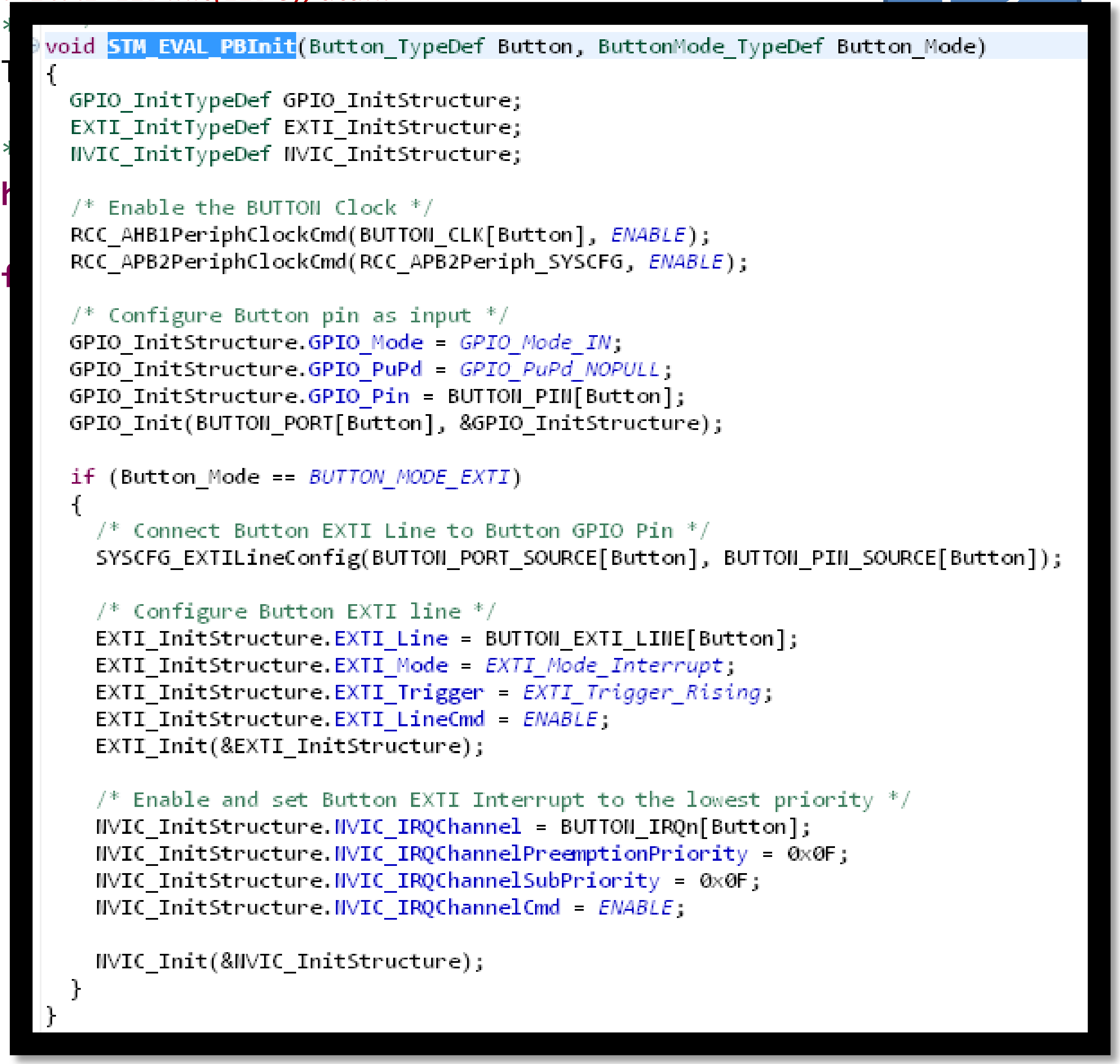
STM_EVAL_LEDInit(LED6); után:

```
/*
ST

/*
wh
{
if



  }
}
```



```
void STM_EVAL_PBInit(Button_TypeDef Button, ButtonMode_TypeDef Button_Mode)
{
  GPIO_InitTypeDef GPIO_InitStructure;
  EXTI_InitTypeDef EXTI_InitStructure;
  NVIC_InitTypeDef NVIC_InitStructure;

  /* Enable the BUTTON Clock */
  RCC_AHB1PeriphClockCmd(BUTTON_CLK[Button], ENABLE);
  RCC_APB2PeriphClockCmd(RCC_APB2Periph_SYSCFG, ENABLE);

  /* Configure Button pin as input */
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN;
  GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL;
  GPIO_InitStructure.GPIO_Pin = BUTTON_PIN[Button];
  GPIO_Init(BUTTON_PORT[Button], &GPIO_InitStructure);

  if (Button_Mode == BUTTON_MODE_EXTI)
  {
    /* Connect Button EXTI Line to Button GPIO Pin */
    SYSCFG_EXTILineConfig(BUTTON_PORT_SOURCE[Button], BUTTON_PIN_SOURCE[Button]);

    /* Configure Button EXTI line */
    EXTI_InitStructure.EXTI_Line = BUTTON_EXTI_LINE[Button];
    EXTI_InitStructure.EXTI_Mode = EXTI_Mode_Interrupt;
    EXTI_InitStructure.EXTI_Trigger = EXTI_Trigger_Rising;
    EXTI_InitStructure.EXTI_LineCmd = ENABLE;
    EXTI_Init(&EXTI_InitStructure);

    /* Enable and set Button EXTI Interrupt to the lowest priority */
    NVIC_InitStructure.NVIC_IRQChannel = BUTTON_IRQn[Button];
    NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0x0F;
    NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0x0F;
    NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;

    NVIC_Init(&NVIC_InitStructure);
  }
}
```

STM_EVAL_LEDInit(LED6); után:

F3

```
  /*Initialize UserButton*/
  STM_EVAL_PBInit(BUTTON_USER, BUTTON_MODE_GPIO);

  /* Infinite loop */
  while (1)
  {
  if (STM_EVAL_PBGetState(BUTTON_USER)!=0x00)
    {
    /* Turn on LEDs */
    STM_EVAL_LEDOn(LED3);
    STM_EVAL_LEDOn(LED4);
    STM_EVAL_LEDOn(LED5);
    STM_EVAL_LEDOn(LED6);
    }
    else
    {
    /* Turn off LEDs */
    STM_EVAL_LEDOff(LED3);
    STM_EVAL_LEDOff(LED4);
    STM_EVAL_LEDOff(LED5);
    STM_EVAL_LEDOff(LED6);
    }
  }
}
```

F3

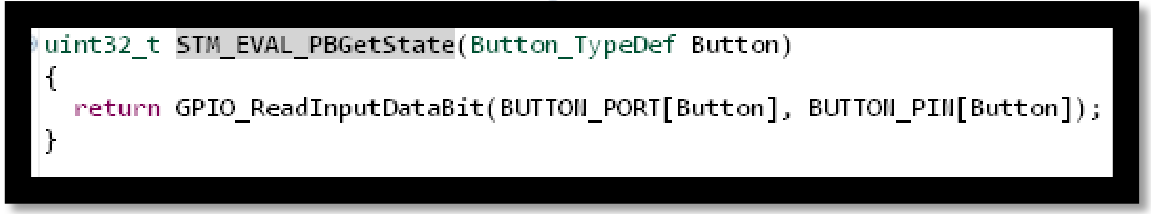
STM_EVAL_LEDInit(LED6); után:

```
  /*Initialize UserButton*/
  STM_EVAL_PBInit(BUTTON_USER, BUTTON_MODE_GPIO);

  /* Infinite loop */
  while (1)
  {
  if (STM_EVAL_PBGetState(BUTTON_USER)!=0x00)
    {
    /* Turn on LEDs */
    STM_EVAL_LEDOn(LED3);
    STM_EVAL_LEDOn(LED4);
    STM_EVAL_LEDOn(LED5);
    STM_EVAL_LEDOn(LED6);
    }
    else
    {
    /* Turn off LEDs */
    STM_EVAL_LEDOff(LED3);
    STM_EVAL_LEDOff(LED4);
    STM_EVAL_LEDOff(LED5);
    STM_EVAL_LEDOff(LED6);
    }
  }
}
```

F3

F3



```
uint32_t STM_EVAL_PBGetState(Button_TypeDef Button)
{
  return GPIO_ReadInputDataBit(BUTTON_PORT[Button], BUTTON_PIN[Button]);
}
```

Gomb kezelés

GPIO

Gombnyomásra LED állapota invertálódik

módosítás

```
while (1)
  {
     if (STM_EVAL_PBGetState(BUTTON_USER)!=0x00)
        {
        /* Toggle LEDs */
        STM_EVAL_LEDToggle(LED3);
        STM_EVAL_LEDToggle(LED4);
        STM_EVAL_LEDToggle(LED5);
        STM_EVAL_LEDToggle(LED6);
        }
  }
```

F3

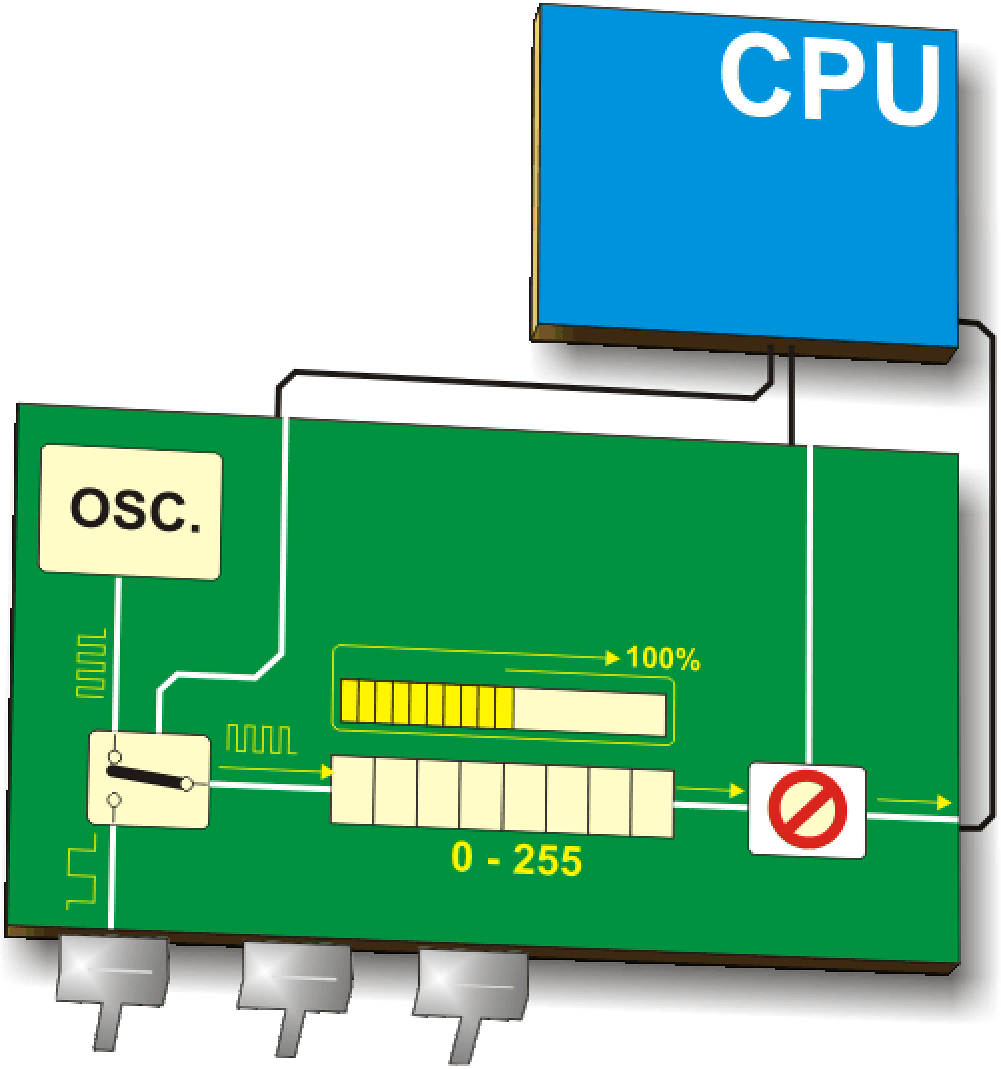
Hiba?

- Prell ☺
  - Megoldás?

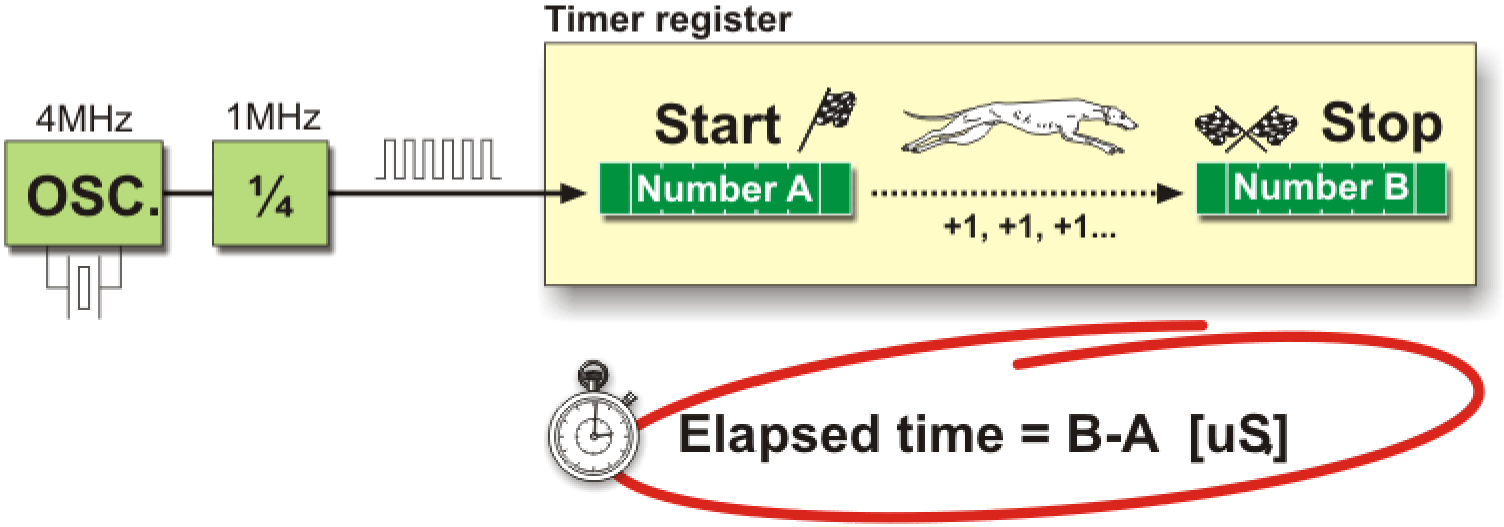
Hiba?

- Prell ☺
  - Megoldás:
    - Mintavételezés: 20…100ms
  - SYSTICK időzítő (1ms felbontás)

Időzítők / Számlálók



Időzítők / Számlálók



Timer register

4MHz

OSC.

1MHz

¼

Start

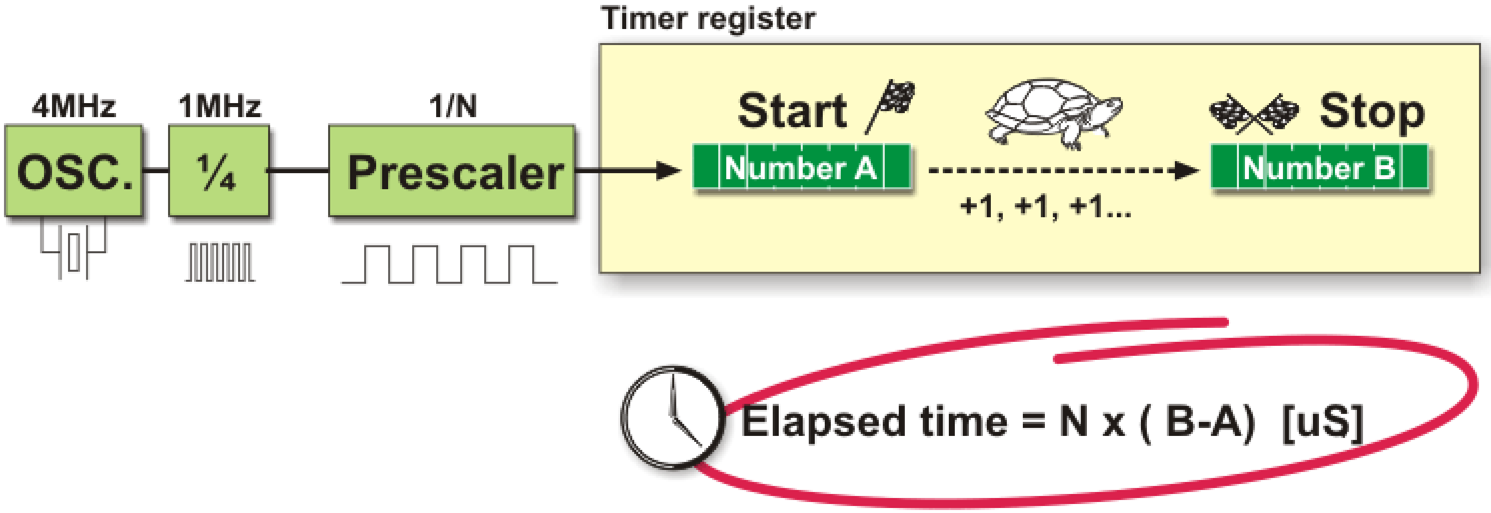
Number A

+1, +1, +1...

Stop

Number B

Elapsed time = B-A [uS]



Timer register

4MHz

OSC.

1MHz

¼

1/N

Prescaler

Start

Number A

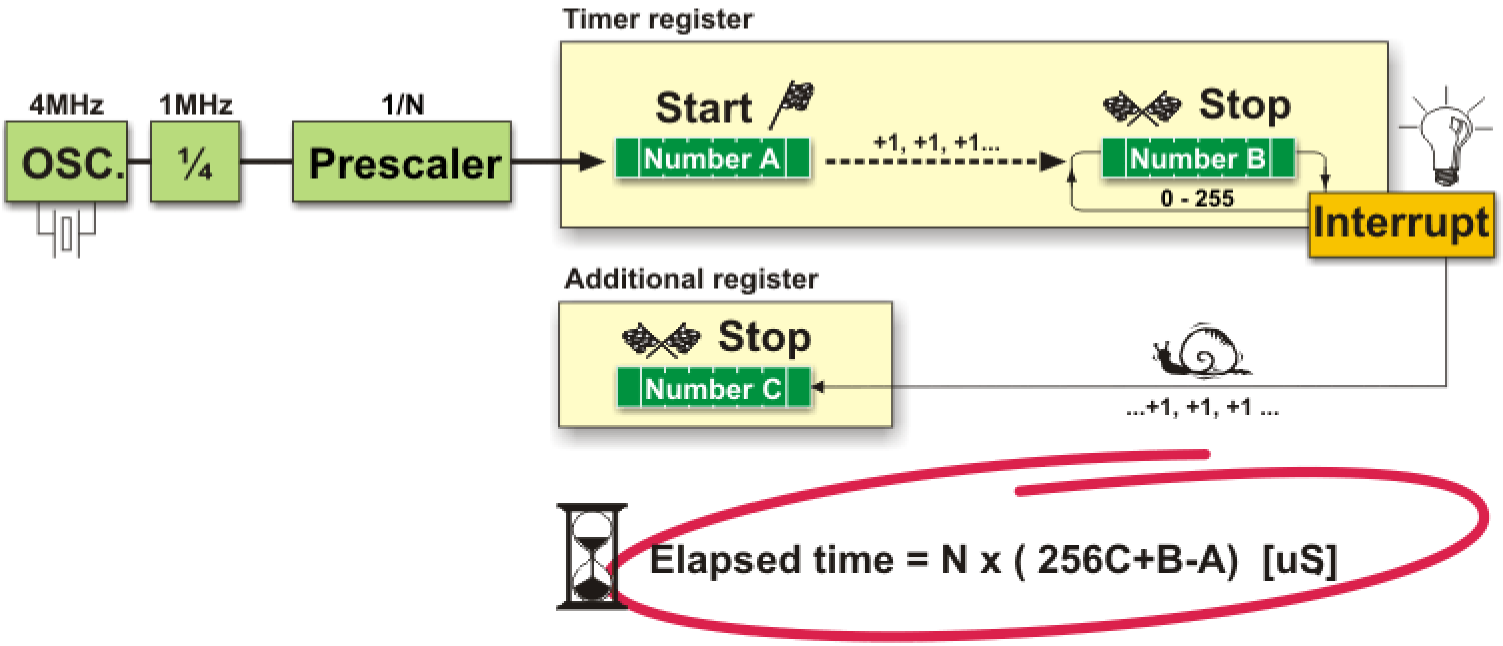
+1, +1, +1...

Stop

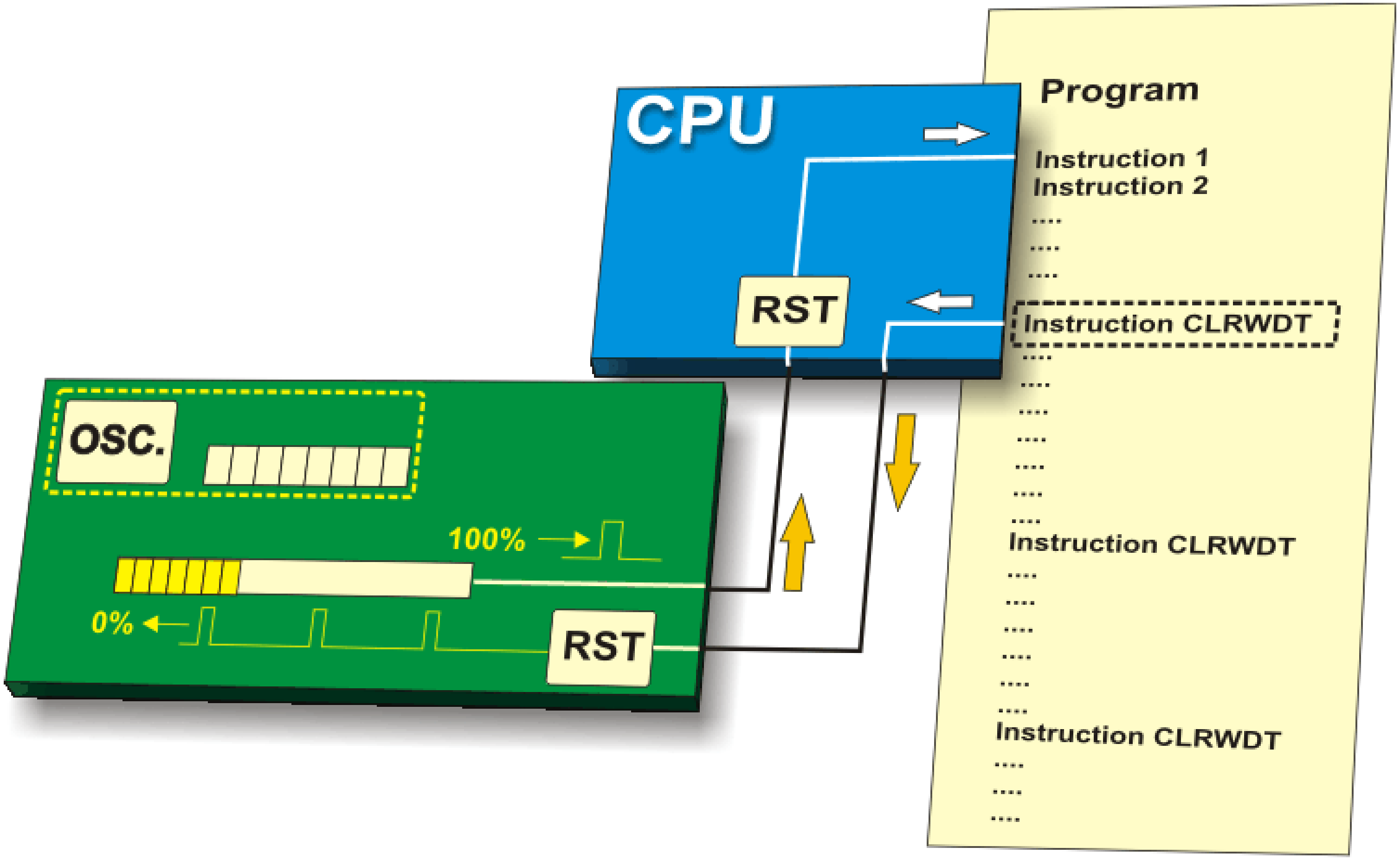
Number B

Elapsed time = N x (B-A) [uS]

Időzítők / Számlálók



Watchdog


CPU
RST
Program
Instruction 1
Instruction 2
....
....
....
Instruction CLRWDT
....
....
....
....
....
....
....
Instruction CLRWDT
....
....
....
....
....
....
Instruction CLRWDT
....
....
....
OSC.
100%
0%
RST

Systick timer

Systick

- Main.c, *~32. sor környéke*:

```
/* Private macro */
/* Private variables */
static __IO uint32_t TimingDelay;
/* Private function prototypes */
void Delay(__IO uint32_t nTime);
/* Private functions */
```

```
/* #define    __IO   volatile  //  defines 'read / write' permissions          */
```

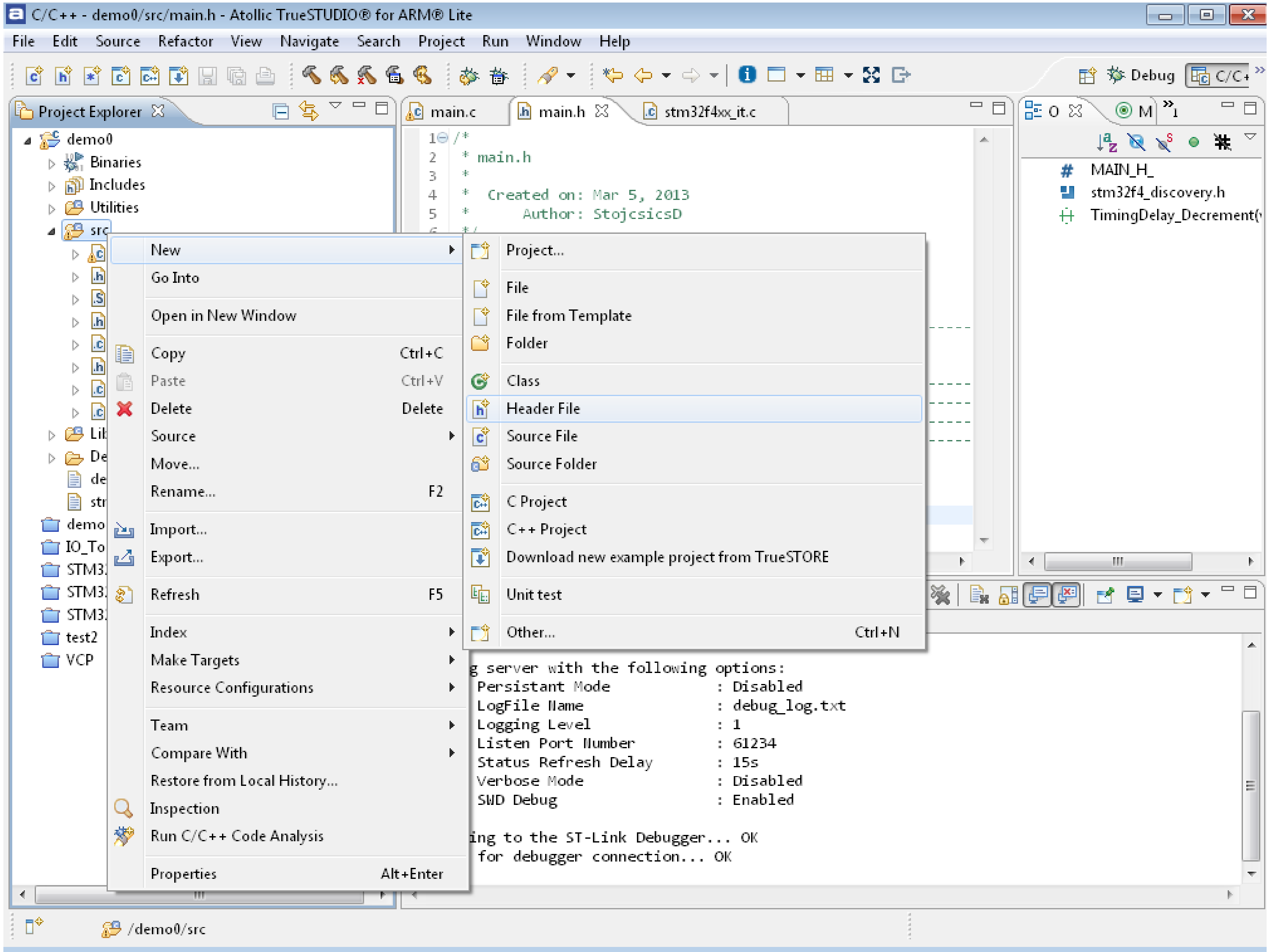
Main() alá

```
void Delay(__IO uint32_t nTime)
{
  TimingDelay = nTime;

  while(TimingDelay != 0);
}

void TimingDelay_Decrement(void)
{
  if (TimingDelay != 0x00)
  {
    TimingDelay--;
  }
}
```

Main.h

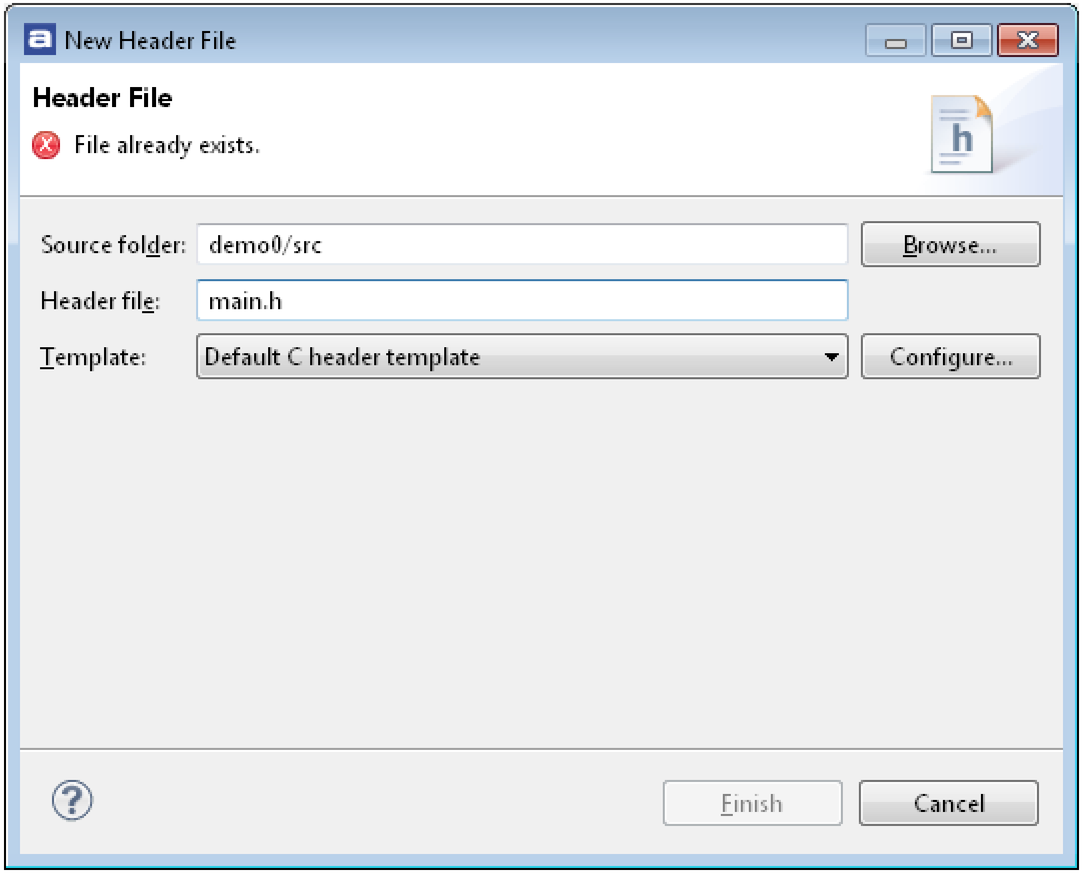


```
/*
 * main.h
 *
 *  Created on: Mar 5, 2013
 *      Author: StojcsicsD
 */
```

```
g server with the following options:
 Persistant Mode            : Disabled
 LogFile Name               : debug_log.txt
 Logging Level              : 1
 Listen Port Number         : 61234
 Status Refresh Delay       : 15s
 Verbose Mode               : Disabled
 SWD Debug                  : Enabled

ing to the ST-Link Debugger... OK
 for debugger connection... OK
```

Main.h



Header File

File already exists.

Source folder: demo0/src

Header file: main.h

Template: Default C header template

Main.h

```
/*
 * main.h
 *
 *  Created on: Mar 5, 2013
 *      Author: StojcsicsD
 */

#ifndef MAIN_H_
#define MAIN_H_

void TimingDelay_Decrement(void);

#endif /* MAIN_H_ */
```

Stm32f4xx_it.c

- SysTick_Handler interrupt *(135.sor)*

```c
void SysTick_Handler(void)
{
  TimingDelay_Decrement();
}
```

F3

Új main()

```
if (SysTick_Config(SystemCoreClock / 1000))
  {
    /* Capture error */
    while (1);
  }
  /* Infinite loop */
 while (1)
  {
   Delay(1000);
    //if (STM_EVAL_PBGetState(BUTTON_USER)!=0x00)
   {
      /* Toggle LEDs */
      STM_EVAL_LEDToggle(LED3);
      STM_EVAL_LEDToggle(LED4);
      STM_EVAL_LEDToggle(LED5);
      STM_EVAL_LEDToggle(LED6);
    }
}
```

Gomb, prell nélkül ☺

```
while (1)
 {
    Delay(100);
    if(STM_EVAL_PBGetState(BUTTON_USER)!=0x00)
    {
         STM_EVAL_LEDToggle(LED3);
         STM_EVAL_LEDToggle(LED4);
         STM_EVAL_LEDToggle(LED5);
         STM_EVAL_LEDToggle(LED6);
    }
 }
```

Feladat 1.:

- Gomb teljes értékű kezelése *(előző állapot bevezetése, le és felfutó él)*

```
int main(void)
{
 int prev_button_state = 0;
 int button_state = 0;
…
 while (1)
  {
  Delay(20);
  button_state=STM_EVAL_PBGetState(BUTTON_USER);
  if(button_state && !prev_button_state)
      {
      STM_EVAL_LEDToggle(LED3);
      STM_EVAL_LEDToggle(LED4);
      STM_EVAL_LEDToggle(LED5);
      STM_EVAL_LEDToggle(LED6);
      }
  prev_button_state=button_state;
  }
}
```

Feladat 2.:

- Körbe futó fény készítése

```
STM_EVAL_LEDOn(0);
…
while (1)
  {
    Delay(100);
    STM_EVAL_LEDOff(i);
    i=(i+1)%4;
    STM_EVAL_LEDOn(i);
  }
```