

LAB 2 GPIO – Alikace přerušení (INTERRUPS)

Zde si ověříme základní rysy programování přerušení pro rodinu procesorů na bázi STM32L4xx

Cíle:

1. Principy přerušení, jako základní nástroj programové logiky
2. Využití softwarového a hardwarového přerušení
3. Procvičení interního a externího přerušení
4. Nastavení registrů v systému přerušení N
5. Stanovení priority přerušení u STM32L4xx
6. Využití sériového kanálu pro zpětnou odezvu desky NUCLEO

Příprava před cvičením

1. Prostudujte architekturu maměťového prostoru ARM – CMSIS(CortexM) [ZDE](#)
2. Zopakujte si vlastnosti Callback funkce a jejich použití. Funkce Stack Pointru.
3. Vyzkoušejte si monitorování programů po sériovém portu. Přečtěte si příklad [ZDE](#).
4. Přečtěte LAB2.

Laboratorní úkol

Napište program v C podle vzoru z LAB2 za použití desky NUCLEO-L433RC.

1. Po spuštění se zelená LED dioda nebude svítit.
2. Modrým tlačítkem vyvoláme přerušení a rozsvítíme diodu LED nebo ji zhasneme svítí-li.
3. Po seriovém kanálu odešleme do terminálu hlášení, že proběhlo přerušení.

Kreativní nápady jsou vždy podporovány.

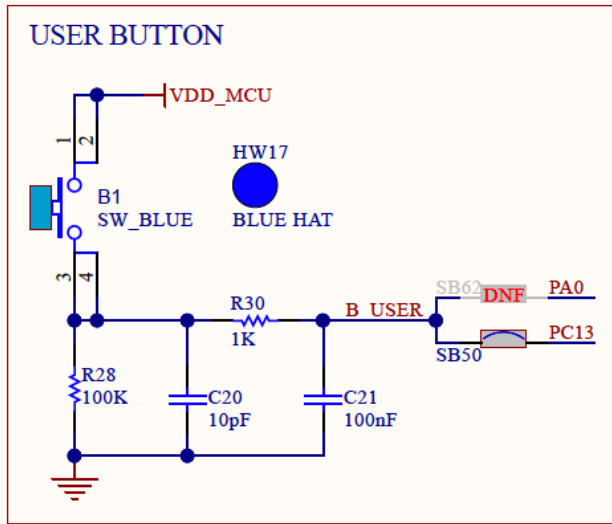
Další úkoly pro chytré hlavy:

Po stlačení tlačítka vyvolejte přerušení a pak zhasněte LED po 5s.

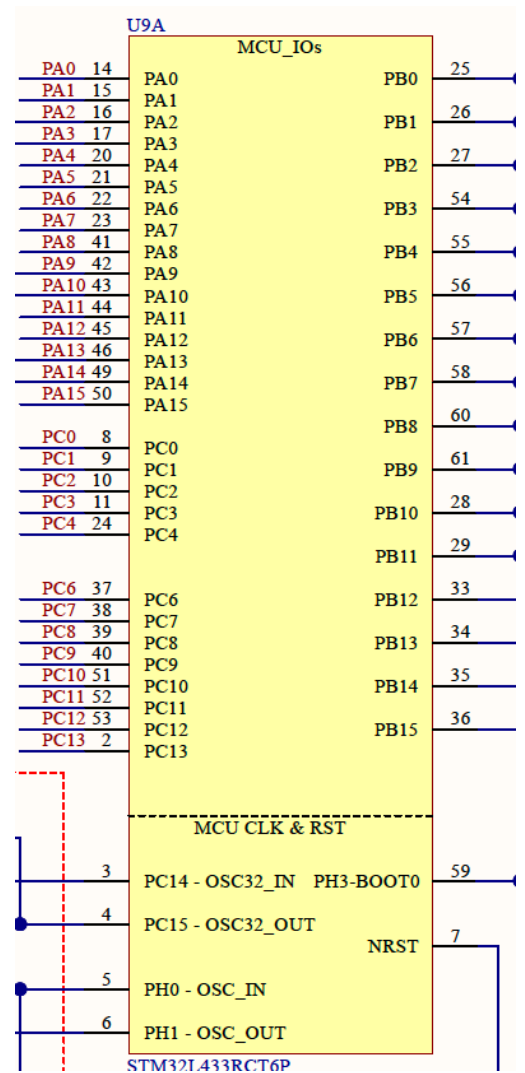
LAB 2 GPIO – Alikace přerušení (INTERRUPTS)

Pomocné nástroje

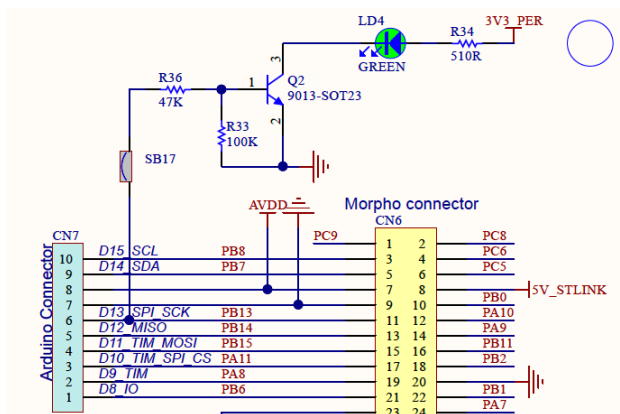
Zapojení modrého tlačítka



Piny STM32F433RC



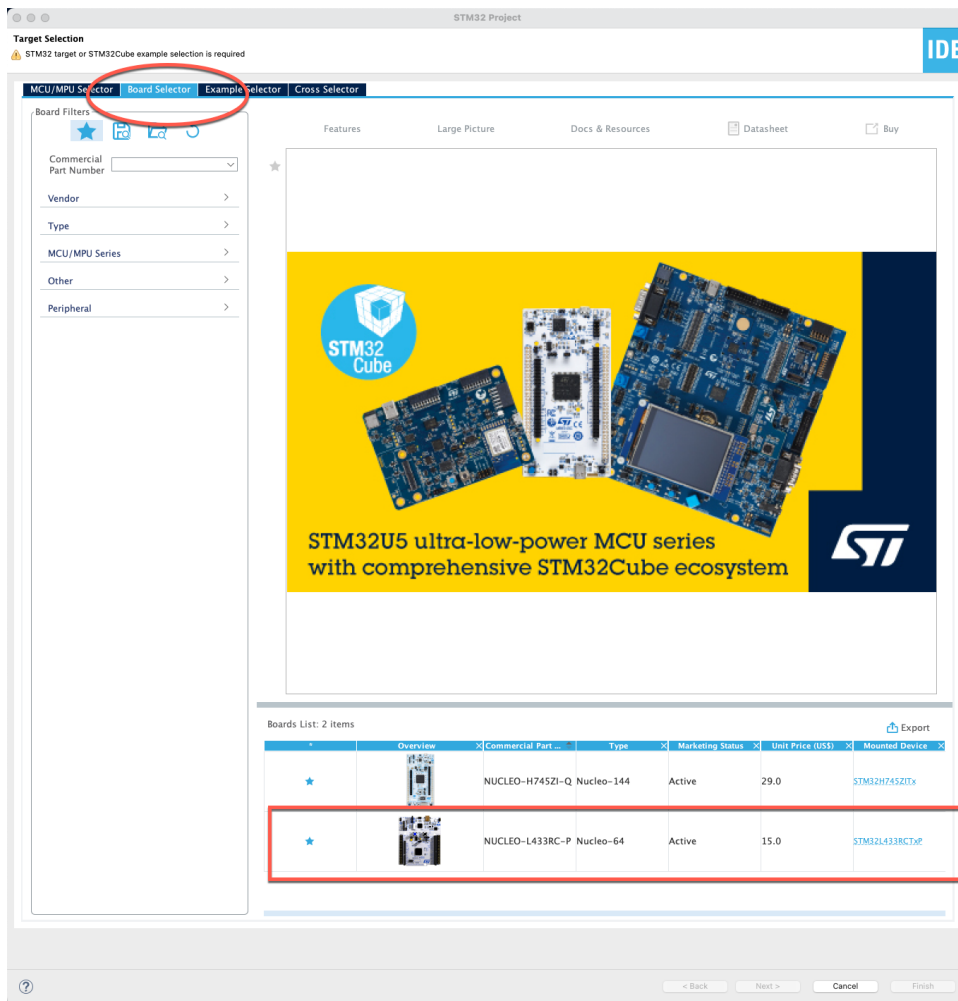
Zapojení zelené diody LED



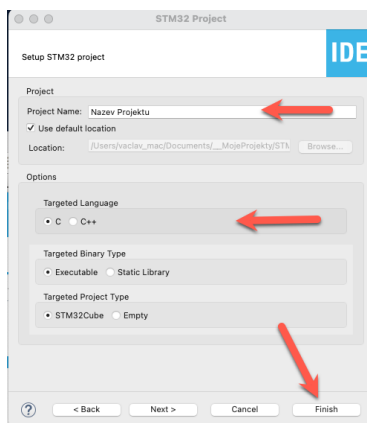
Podrobné schéma desky najdete [zde](#) (ve složce: MB1319-L433RC-P-C02 Board schematic)

LAB 2 GPIO – Alikace přerušení (INTERRUPS)

Výběr desky

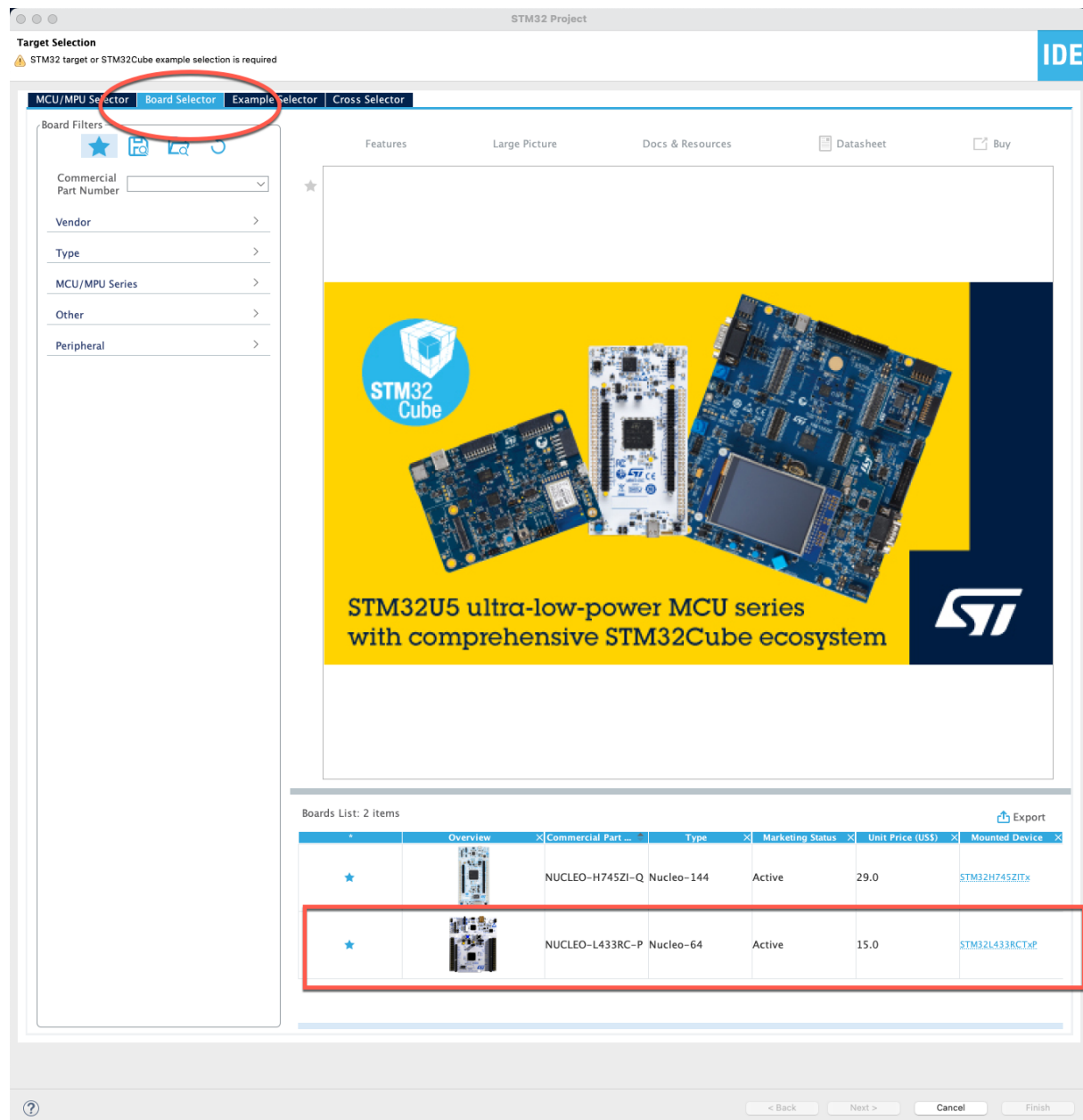


Po **Next >**

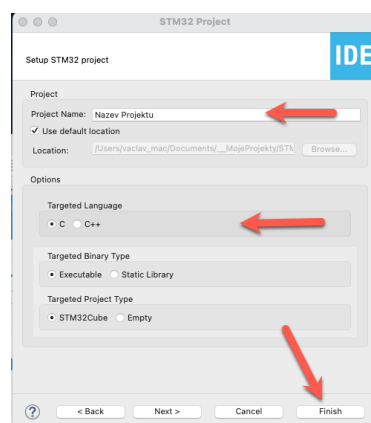


Teď se vygeneruje příslušný projekt. Pokračujeme v nastavení STM32CubeMX.

LAB 2 GPIO – Alikace přerušení (INTERRUPS)



Po Next >



Konfigurace STM32CubeMX pro externí přerušení

The screenshot displays the STM32CubeIDE interface. On the left, the 'NVIC Mode and Configuration' window is open, showing the 'NVIC - Code generation' tab. The 'NVIC Interrupt Table' is visible, with a red box highlighting the 'EXTI line[15:10] interrupts' and 'FPU global interrupt' rows. The 'Priority Group' is set to '4 bits for ...'. The 'System Core' is set to 'NVIC'. The 'NVIC Interrupt Table' shows the following interrupts:

Interrupt	Enabled	Preemption	Priority
Non maskable interrupt	☒	☒	0
Hard fault interrupt	☒	☒	0
Memory management fault	☒	☒	0
Prefetch fault, memory access fault	☒	☒	0
Undefined instruction or illegal state	☒	☒	0
System service call via SW interrupt	☒	☒	0
Debug monitor	☒	☒	0
Pendable request for system service	☒	☒	0
Time base: System tick timer	☒	☒	0
PVD/PVMI/PWMI/PWMI interrupts through EXTI lines 16/35/36/37/...	☒	☒	0
Flash global interrupt	☒	☒	0
RCC global interrupt	☒	☒	0
USART global interrupt	☒	☒	0
EXTI line[15:10] interrupts	☒	☒	0
FPU global interrupt	☒	☒	0

On the right, the 'Pinout view' shows the pin configuration for the STM32L433RCTxP LQFP64 package. The pins are color-coded: green for digital, yellow for analog, and grey for power. The pins are labeled as follows:

- VDD
- VSS
- VDD12
- PB9
- PB8
- PB3...
- PB7
- PB6
- PB5
- PB4 (...)
- PB3 (...)
- PC12
- PC11
- PC10
- PC9
- PC8
- PC7
- PC6
- PC5
- PC4
- PC3
- PC2
- PC1
- PC0
- NRST
- VSSA...
- VDDA...
- PA0
- PA1
- PA2
- PA3
- VSS
- VDD
- PA4
- PA5
- PA6
- PA7
- PC4
- PB0
- PB1
- PB2
- PB10
- PB11
- VDD12

The pinout diagram also shows the following connections:

- GPIO_EXTI_13
- RCC_OSC32_IN
- RCC_OSC32_OUT
- MCO
- RCC_OSC_OUT
- USART_TX

LAB 2 GPIO – Alikace přerušení (INTERRUPTS)

Nastavení sériové komunikace pro monitoring programu

```
static void MX_USART2_UART_Init(void)
{
    /* USER CODE BEGIN USART2_Init 0 */

    /* USER CODE END USART2_Init 0 */

    /* USER CODE BEGIN USART2_Init 1 */

    /* USER CODE END USART2_Init 1 */
    huart2.Instance = USART2;
    huart2.Init.BaudRate = 115200;
    huart2.Init.WordLength = UART_WORDLENGTH_8B;
    huart2.Init.StopBits = UART_STOPBITS_1;
    huart2.Init.Parity = UART_PARITY_NONE;
    huart2.Init.Mode = UART_MODE_TX_RX;
    huart2.Init.HwFlowCtl = UART_HWCONTROL_NONE;
    huart2.Init.OverSampling = UART_OVERSAMPLING_16;
    huart2.Init.OneBitSampling = UART_ONE_BIT_SAMPLE_DISABLE;
    huart2.AdvancedInit.AdvFeatureInit = UART_ADVFEATURE_NO_INIT;
    if (HAL_UART_Init(&huart2) != HAL_OK)
    {
        Error_Handler();
    }
    /* USER CODE BEGIN USART2_Init 2 */

    /* USER CODE END USART2_Init 2 */

}

}Nastavení Callback funkce

// deklarace
uint8_t MSG[35] = {"\0"};

/* Private user code -----
-----*/
/* USER CODE BEGIN 0 */
// EXTI Line9 External Interrupt ISR Handler CallbackFun
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    if(GPIO_Pin == GPIO_PIN_13) // If The INT Source Is EXTI Line9 (A9
Pin)
    {
        // Toggle The Output (LED) Pin

        HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_13);

        // hláška pro PC
        sprintf(MSG, "Hello clovece! Preruseni EXT 13 \r\n");

        // odeslání dat
        HAL_UART_Transmit(&huart2, MSG, sizeof(MSG), 100);
    }
}
```

LAB 2 GPIO – Alikace přerušení (INTERRUPTS)

Taty je vybrano přerušení – poslední dva řádky

```
static void MX_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStruct = {0};

    /* GPIO Ports Clock Enable */
    __HAL_RCC_GPIOC_CLK_ENABLE();
    __HAL_RCC_GPIOH_CLK_ENABLE();
    __HAL_RCC_GPIOA_CLK_ENABLE();
    __HAL_RCC_GPIOB_CLK_ENABLE();

    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(GPIOA, SMPS_EN_Pin|SMPS_V1_Pin|SMPS_SW_Pin, GPIO_PIN_RESET);

    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(LD4_GPIO_Port, LD4_Pin, GPIO_PIN_RESET);

    /*Configure GPIO pin : GPIO_EXTI_13_Pin */
    GPIO_InitStruct.Pin = GPIO_EXTI_13_Pin;
    GPIO_InitStruct.Mode = GPIO_MODE_IT_RISING;
    GPIO_InitStruct.Pull = GPIO_PULLDOWN;
    HAL_GPIO_Init(GPIO_EXTI_13_GPIO_Port, &GPIO_InitStruct);

    /*Configure GPIO pins : SMPS_EN_Pin SMPS_V1_Pin SMPS_SW_Pin */
    GPIO_InitStruct.Pin = SMPS_EN_Pin|SMPS_V1_Pin|SMPS_SW_Pin;
    GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

    /*Configure GPIO pin : SMPS_PG_Pin */
    GPIO_InitStruct.Pin = SMPS_PG_Pin;
    GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
    GPIO_InitStruct.Pull = GPIO_PULLUP;
    HAL_GPIO_Init(SMPS_PG_GPIO_Port, &GPIO_InitStruct);

    /*Configure GPIO pin : LD4_Pin */
    GPIO_InitStruct.Pin = LD4_Pin;
    GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(LD4_GPIO_Port, &GPIO_InitStruct);

    /* EXTI interrupt init*/
    HAL_NVIC_SetPriority(EXTI15_10_IRQn, 0, 0);
    HAL_NVIC_EnableIRQ(EXTI15_10_IRQn);
}
```

LAB 2 GPIO – Alikace přerušení (INTERRUPS)

Program main.c (pouze while) pro rozsvícení led

```
/* USER CODE BEGIN WHILE */
while (1)
{
    /* USER CODE END WHILE */
    /* HAL_GPIO_TogglePin(GPIOB, "LED");
    HAL_Delay(1000);
    */
    // IF Button Is Pressed
    if(HAL_GPIO_ReadPin (GPIOC, GPIO_PIN_13))
    {
        // Set The LED ON!
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_13, GPIO_PIN_SET);
    }
    else
    {
        // Else .. Turn LED OFF!
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_13, GPIO_PIN_RESET);
    }
}
/* USER CODE BEGIN 3 */
}
```