ARM Cortex Core Microcontrollers

4th Laboratory: External IT handling

Scherer Balázs

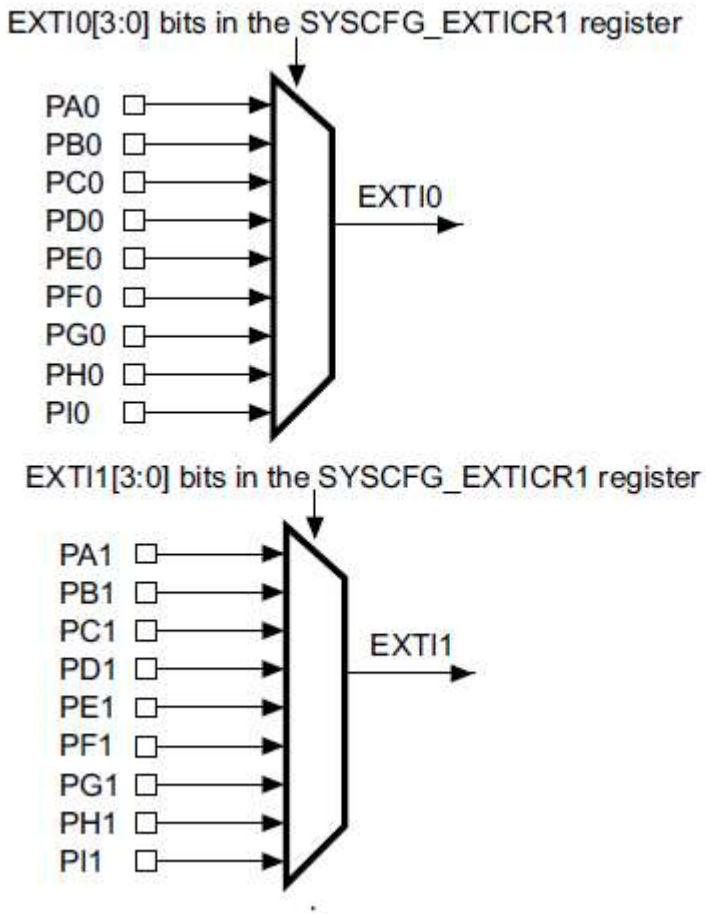
Budapest University of Technology and Economics
Department of Measurement and Information Systems

Push button on STM32F429 discovery

- B1 USER: PA0
- Should be configured with internal pull down to work

Exti in STM32F4 micro

- Every GPIO can be used as External interrupt, there are Exti: 0-15, but for every External interrupt line a GPIO port should be selected, so for Exti x, only GPIOA-I x line can be used
- Chapter 12.2.5 Exti mapping



Selecting Exti Source

- System module in firmware library

__STATIC_INLINE uint32_t LL_SYSCFG_GetEXTISource (uint32_t Line)

Get the configured defined for specific EXTI Line.

Reference Manual to LL API cross reference:

SYSCFG_EXTICR1 EXTIx LL_SYSCFG_GetEXTISource
SYSCFG_EXTICR2 EXTIx LL_SYSCFG_GetEXTISource
SYSCFG_EXTICR3 EXTIx LL_SYSCFG_GetEXTISource
SYSCFG_EXTICR4 EXTIx LL_SYSCFG_GetEXTISource

Parameters:

Line This parameter can be one of the following values:

- LL_SYSCFG_EXTI_LINE0
- LL_SYSCFG_EXTI_LINE1
- LL_SYSCFG_EXTI_LINE2
- LL_SYSCFG_EXTI_LINE3

Exti edge and operation configuration

- External interrupt operation parameters
- Firmware Library EXTI module

uint32_t LL_EXTI_Init (LL_EXTI_InitTypeDef * EXTI_InitStruct)

Initialize the EXTI registers according to the specified parameters in EXTI_InitStruct.

Parameters:

EXTI_InitStruct pointer to a **LL_EXTI_InitTypeDef** structure.

Return values:

An ErrorStatus enumeration value:

- SUCCESS: EXTI registers are initialized
- ERROR: not applicable

NVIC setup

- Not firmware library CMSIS!
- device.h the IRQ number list
- NVIC_EnableIRQ()

```
/******  Cortex-M4 Processor Exceptions Numbers ***********************************************************/
  NonMaskableInt_IRQn         = -14,    /*!< 2 Non Maskable Interrupt                                */
  MemoryManagement_IRQn       = -12,    /*!< 4 Cortex-M4 Memory Management Interrupt                 */
  BusFault_IRQn               = -11,    /*!< 5 Cortex-M4 Bus Fault Interrupt                         */
  UsageFault_IRQn             = -10,    /*!< 6 Cortex-M4 Usage Fault Interrupt                       */
  SVCall_IRQn                 = -5,     /*!< 11 Cortex-M4 SV Call Interrupt                          */
  DebugMonitor_IRQn           = -4,     /*!< 12 Cortex-M4 Debug Monitor Interrupt                    */
  PendSV_IRQn                 = -2,     /*!< 14 Cortex-M4 Pend SV Interrupt                          */
  SysTick_IRQn                = -1,     /*!< 15 Cortex-M4 System Tick Interrupt                      */
/******  STM32 specific Interrupt Numbers *****************************************************************/
  WWDG_IRQn                   = 0,      /*!< Window WatchDog Interrupt                               */
  PVD_IRQn                    = 1,      /*!< PVD through EXTI Line detection Interrupt               */
  TAMP_STAMP_IRQn             = 2,      /*!< Tamper and TimeStamp interrupts through the EXTI line   */
  RTC_WKUP_IRQn               = 3,      /*!< RTC Wakeup interrupt through the EXTI line              */
  FLASH_IRQn                  = 4,      /*!< FLASH global Interrupt                                  */
  RCC_IRQn                    = 5,      /*!< RCC global Interrupt                                    */
  EXTI0_IRQn                  = 6,      /*!< EXTI Line0 Interrupt                                    */
  EXTI1_IRQn                  = 7,      /*!< EXTI Line1 Interrupt                                    */
  EXTI2_IRQn                  = 8,      /*!< EXTI Line2 Interrupt                                    */
  EXTI3_IRQn                  = 9,      /*!< EXTI Line3 Interrupt                                    */
```

IRQ function

- CMSIS: startup_device.s

```
/* External Interrupts */
.word     WWDG_IRQHandler                   /* Window WatchDog                  */
.word     PVD_IRQHandler                    /* PVD through EXTI Line detection */
.word     TAMP_STAMP_IRQHandler             /* Tamper and TimeStamps through the EXTI line */
.word     RTC_WKUP_IRQHandler               /* RTC Wakeup through the EXTI line */
.word     FLASH_IRQHandler                  /* FLASH                            */
.word     RCC_IRQHandler                    /* RCC                              */
.word     EXTI0_IRQHandler                  /* EXTI Line0                       */
.word     EXTI1_IRQHandler                  /* EXTI Line1                       */
.word     EXTI2_IRQHandler                  /* EXTI Line2                       */
.word     EXTI3_IRQHandler                  /* EXTI Line3                       */
.word     EXTI4_IRQHandler                  /* EXTI Line4                       */
.word     DMA1_Stream0_IRQHandler           /* DMA1 Stream 0                    */
.word     DMA1_Stream1_IRQHandler           /* DMA1 Stream 1                    */
.word     DMA1_Stream2_IRQHandler           /* DMA1 Stream 2                    */
```

Flag management

- Firmware library EXTI module

__STATIC_INLINE void LL_EXTI_ClearFlag_0_31 (uint32_t ExtiLine)

Clear ExtLine Flags for Lines in range 0 to 31.

Note:
This bit is set when the selected edge event arrives on the interrupt line. This bit is cleared by writing a 1 to the bit.

Reference Manual to LL API cross reference:
PR PIFx LL_EXTI_ClearFlag_0_31

Parameters:
ExtiLine This parameter can be a combination of the following values:

- LL_EXTI_LINE_0
- LL_EXTI_LINE_1
- LL_EXTI_LINE_2