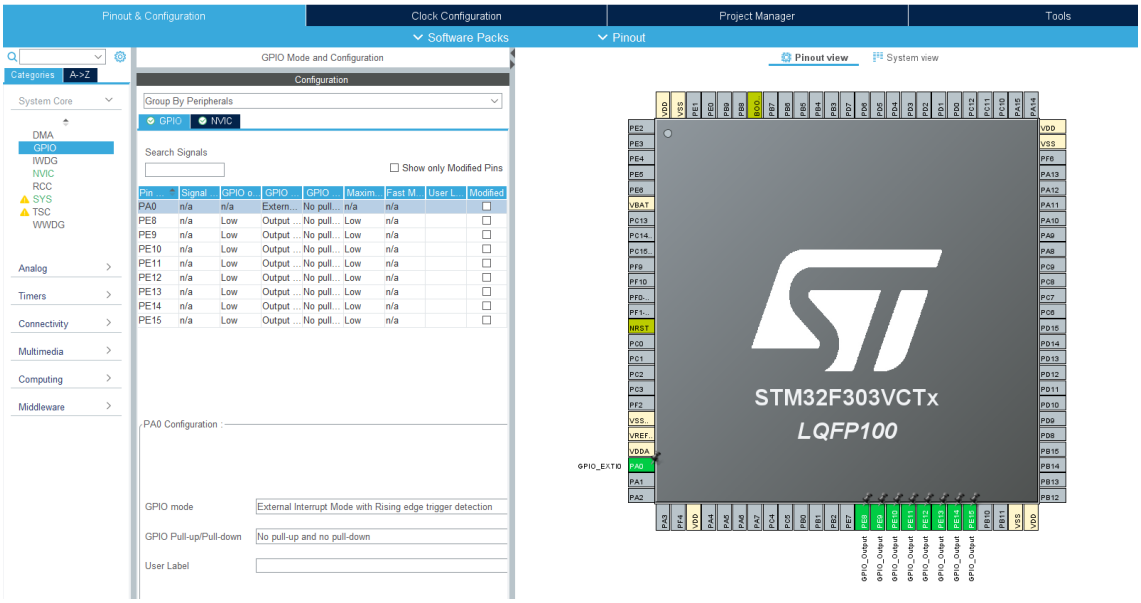
Lösung Vorlesungbegleitende Übungsaufgaben

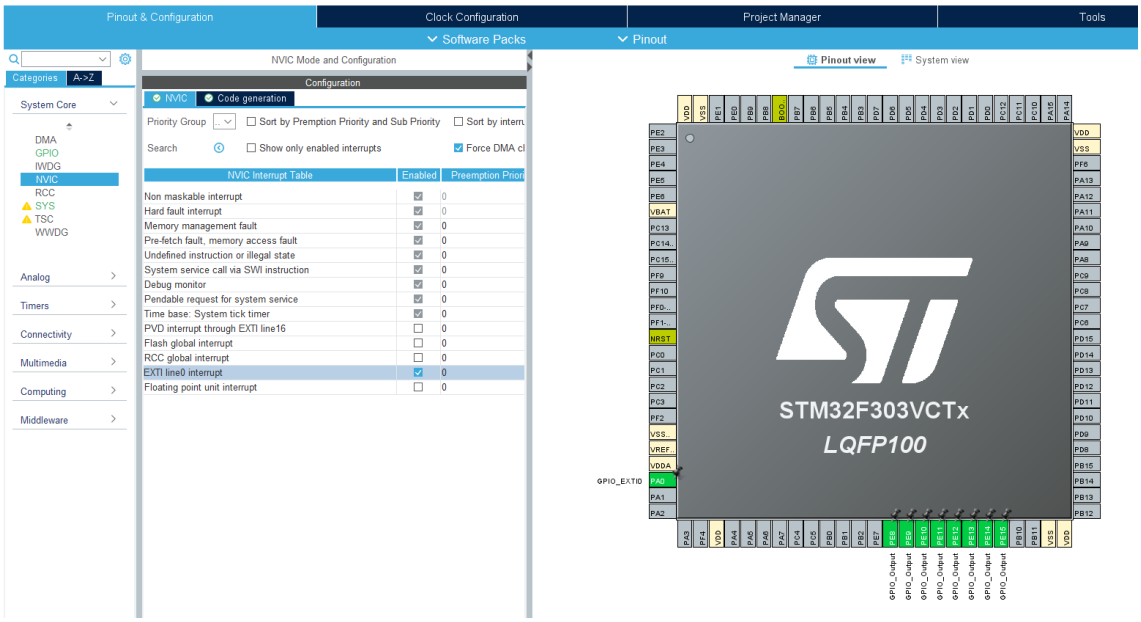
-Mikrocomputertechnik-

Aufgabe 3: Interrupt Counter

Schreiben Sie ein Programm, dass bei Betätigung des USER Tasters einen Interrupt auslöst und den Wert einer Variable inkrementell erhöht. Der Wert der Variable soll in binärer Codierung durch die LEDs PE8 bis PE15 dargestellt werden. Kann der Wert nicht mehr dargestellt werden soll er zurückgesetzt werden.

Lösung:






```c
volatile static int InterruptCounter=0;

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
        if(GPIO_Pin == GPIO_PIN_0) //Auswertung von PA0
        {
                if(InterruptCounter>=255)
                {
                        InterruptCounter=0;
                }
                else
                {
                        InterruptCounter++;
                }
        }
}


int main(void)
{
  HAL_Init();

  SystemClock_Config();

  MX_GPIO_Init();
  while (1)
  {
        //Reset aller LEDs
        HAL_GPIO_WritePin(GPIOE, GPIO_PIN_8|GPIO_PIN_9|GPIO_PIN_10|GPIO_PIN_11|GPIO_PIN_12|
                          GPIO_PIN_13|GPIO_PIN_14|GPIO_PIN_15, GPIO_PIN_RESET);

        //Setzen der jeweiligen LED
        if(InterruptCounter&1<<0)
    {
        HAL_GPIO_WritePin(GPIOE, GPIO_PIN_8, GPIO_PIN_SET);
    }
    if(InterruptCounter&1<<1)
    {
        HAL_GPIO_WritePin(GPIOE, GPIO_PIN_9, GPIO_PIN_SET);
        }
    if(InterruptCounter&1<<2)
    {
        HAL_GPIO_WritePin(GPIOE, GPIO_PIN_10, GPIO_PIN_SET);
        }
    if(InterruptCounter&1<<3)
    {
```

```
        HAL_GPIO_WritePin(GPIOE, GPIO_PIN_11, GPIO_PIN_SET);
        }
    if(InterruptCounter&1<<4)
    {
        HAL_GPIO_WritePin(GPIOE, GPIO_PIN_12, GPIO_PIN_SET);
    }
    if(InterruptCounter&1<<5)
    {
        HAL_GPIO_WritePin(GPIOE, GPIO_PIN_13, GPIO_PIN_SET);
    }
    if(InterruptCounter&1<<6)
    {
        HAL_GPIO_WritePin(GPIOE, GPIO_PIN_14, GPIO_PIN_SET);
    }
    if(InterruptCounter&1<<7)
    {
        HAL_GPIO_WritePin(GPIOE, GPIO_PIN_15, GPIO_PIN_SET);
    }

    //Kurzes Delay
    HAL_Delay(100);
  }
}



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

  /* GPIO Ports Clock Enable */
  __HAL_RCC_GPIOA_CLK_ENABLE();
  __HAL_RCC_GPIOE_CLK_ENABLE();

  /*Configure GPIO pin Output Level */
  HAL_GPIO_WritePin(GPIOE, GPIO_PIN_8|GPIO_PIN_9|GPIO_PIN_10|GPIO_PIN_11
                          |GPIO_PIN_12|GPIO_PIN_13|GPIO_PIN_14|GPIO_PIN_15, GPIO_PIN_RESET);

  /*Configure GPIO pin : PA0 */
  GPIO_InitStruct.Pin = GPIO_PIN_0;
  GPIO_InitStruct.Mode = GPIO_MODE_IT_RISING;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

  /*Configure GPIO pins : PE8 PE9 PE10 PE11
                           PE12 PE13 PE14 PE15 */
  GPIO_InitStruct.Pin = GPIO_PIN_8|GPIO_PIN_9|GPIO_PIN_10|GPIO_PIN_11
                          |GPIO_PIN_12|GPIO_PIN_13|GPIO_PIN_14|GPIO_PIN_15;
  GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
  HAL_GPIO_Init(GPIOE, &GPIO_InitStruct);

  /* EXTI interrupt init*/
  HAL_NVIC_SetPriority(EXTI0_IRQn, 0, 0);
  HAL_NVIC_EnableIRQ(EXTI0_IRQn);

}
```