Project_3.0_Interrupt

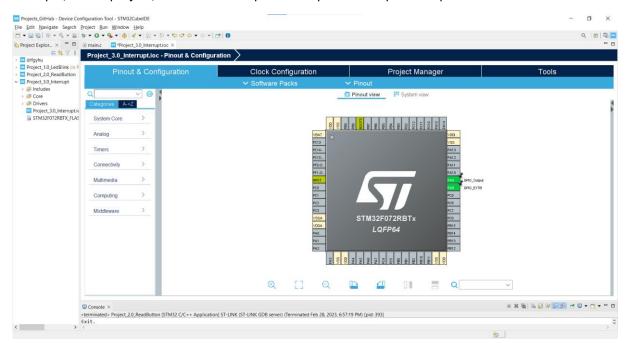
Bu projede, bir GPIO kesme pin'i ayarlayacağız. Ve bu kesmede bir çıkış pinini değiştireceğiz. GPIO kesme pinine bir buton bağlayacağız ve çıkış pinine ise yine bir led bağlayacağız. Kesme üretip ledi açma kapatma işlemi yapıcaz.

STEP 1: Open STM32CubeIDE and Create New Project.

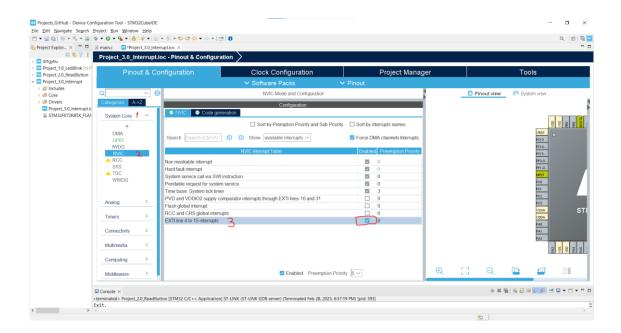
STEP 2: Select Target microcontroller and double Click. My MCU is STM32F072RBT6.

STEP3: Enter the project name and finish.

STEP 4: Click the pin you want to set as input for interrupt and select GPIO_EXTIX/ GPIO_output. For example, in this project, I choose the A9 pin for output and A8 pin for input.

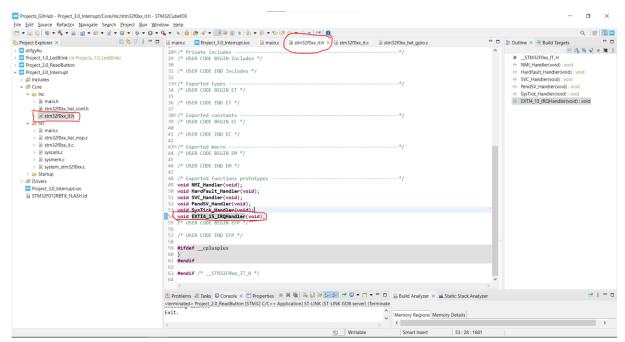


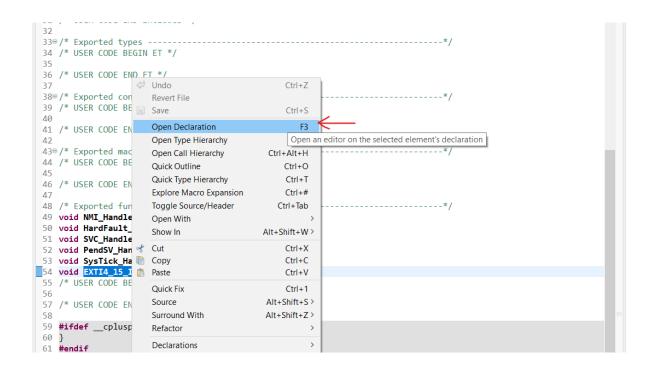
STEP 5: Open The NVIC Tab And Enable The EXTI line8 Interrupt.

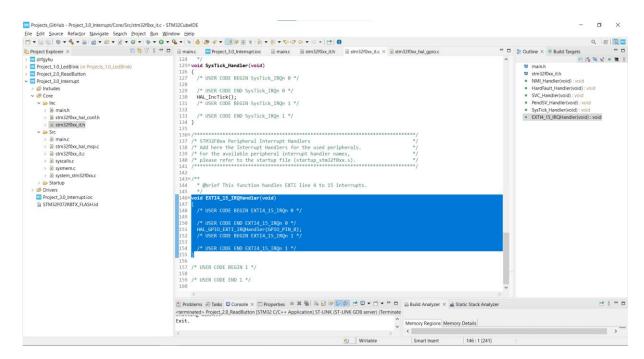


STEP 6: Set the RCC External Clock Source and then CTRL + S to generate the project code. And we open our main.c file in the project files.

When this interruption occurs, the called ISR (Interrupt Service Routine) is located in the stm32f1xx_it.h file.







```
124
125@ void SysTick_Handler(void)
126 {
127
       /* USER CODE BEGIN SysTick IRQn 0 */
128
129
       /* USER CODE END SysTick_IRQn 0 */
130
      HAL_IncTick();
       /* USER CODE BEGIN SysT: ♥ Undo
                                                                  Ctrl+7
131
                                     Revert File
132
       /* USER CODE END SysTick Save
                                                                  Ctrl+S
133
134 }
                                     Open Declaration
                                                                     F3
135
                                     Open Type Hierarchy Open an editor on the selected element's declaration
1369/****************
                                     Open Call Hierarchy
                                                              Ctrl+Alt+H
137 /* STM32F0xx Peripheral Ir
138 /* Add here the Interrupt
                                     Quick Outline
                                                                  Ctrl+O
139 /* For the available peri
                                     Quick Type Hierarchy
                                                                  Ctrl+T
140 /* please refer to the sta
                                     Explore Macro Expansion
                                                                  Ctrl+#
141 /************
                                     Toggle Source/Header
                                                                Ctrl+Tab
142
                                     Open With
1439 /**
      * @brief This function |
                                                             Alt+Shift+W >
                                     Show In
144
145
                                     Cut
                                                                  Ctrl+X
146 void EXTI4_15_IRQHandler(v
                                     Copy
                                                                  Ctrl+C
147 {
                                     Paste
                                                                  Ctrl+V
       /* USER CODE BEGIN EXTI
148
149
                                                                  Ctrl+1
150
        /* USER CODE END EXTI4
                                     Source
                                                              Alt+Shift+S >
 151 HAL_GPIO_EXTI_IRQHandler
                                     Surround With
                                                              Alt+Shift+Z>
152
       /* USER CODE BEGIN EXTI4
                                     Refactor
153
154
       /* USER CODE END EXTI4
                                                                       >
                                     Declarations
155 }
                                     References
                                                                       >
156
                                     Search Text
157 /* USER CODE BEGIN 1 */
158
                                     Build Selected File(s)
159 /* USER CODE END 1 */
                                     Clean Selected File(s)
160
                                     Resource Configurations
🖺 Problems 🧔 Tasks 📮 Console 🗴 🔲 🚺 Run As
                                                                       > 📑 🔻 🗖 🖟 Build Analyzer × 🛓 Static Stack Analyze
```

```
}
      else
487
      {
488
        return HAL_ERROR;
489
      }
490 }
491
4929 /**
      * @brief Handle EXTI interrupt request.
493
      * @param GPIO_Pin Specifies the port pin connected to corresponding EXTI line.
494
      * @retval None
495
496
497⊖<mark>void HAL_GPIO_EXTI_IRQHandler</mark>(uint16_t GPIO_Pin)
498
499
      if(_HAL_GPIO_EXTI_GET_IT(GPIO_Pin) != 0x00u)
500
501
502
           HAL GPTO FXTT CLEAR TT(GPTO Pin)
503
        HAL_GPIO_EXTI_Callback(GPIO_Pin);
504
505
507⊕ /**
508
      * @brief EXTI line detection callback.
509
      * @param GPIO_Pin Specifies the port pin connected to corresponding EXTI line.
510
      * @retval None
511
      week word HAI GOTA FYTT Callback/win+16 + GOTA Din
```

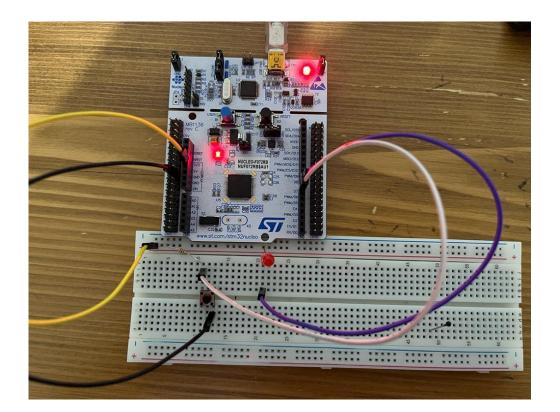
This function clears the interrupt source, then calls the ISR handler callback function. Now we will write code for this callback function (main.c file)

```
37 /* USER CODE BEGIN PM */
 38
 39 /* USER CODE END PM */
 41 /* Private variables -----*/
 43 /* USER CODE BEGIN PV */
 45 /* USER CODE END PV */
 47 /* Private function prototypes -
 48 void SystemClock_Config(void);
 49 static void MX_GPIO_Init(void);
 50 /* USER CODE BEGIN PFP */
 52 /* USER CODE END PFP */
 56@void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin) // EXTI Line[4:15] External Interrupt CallBackFunction
 57 {
       if(GPIO_Pin == GPIO_PIN_8)
                                          // If The Interrupt source Is EXTI Line8 (A8 Pin)
 59
 60
          HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_9); // Toggle The Output Pin
 61
 62
 63 }
 65
 669/**
 * @brief The application entry point.

8 * @retval int
69 */
 70⊖ int main(void)
 71 {
     /* USER CODE BEGIN 1 */
    /* USER CODE END 1 */
 74
```

STEP 7: We press RUN to compile the code and upload it to the board.

STEP 8: Now it's time to connect the led to the board.



That's it.

When we press the button, an interrupt will occur and the interrupt function will toggle the output pin. If the led is off it will turn on, if it is on it will turn off.