ใบงานครั้งที่ 06

Digital Input & External Interrupt Line

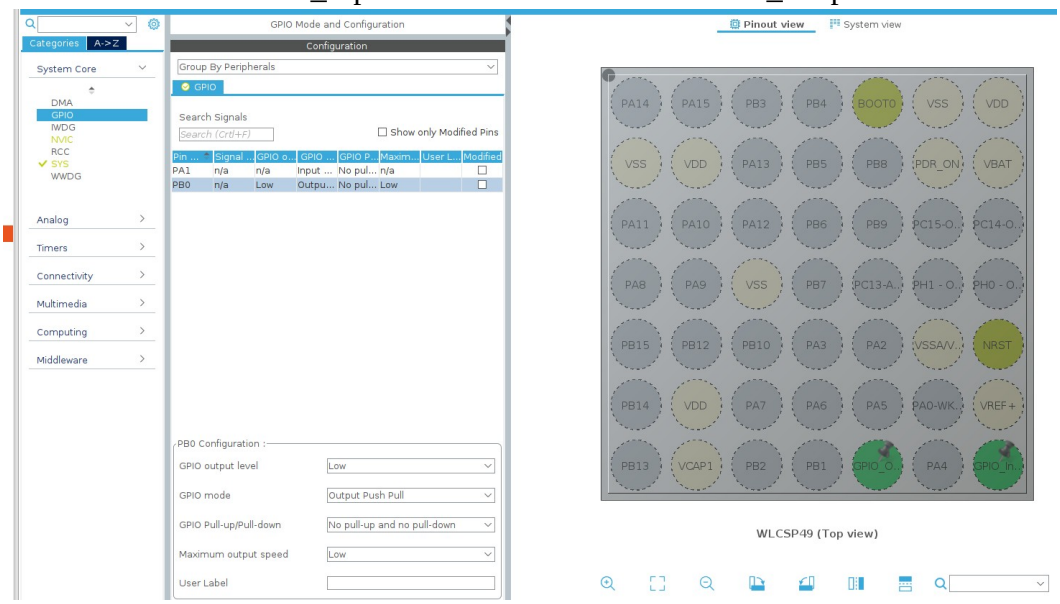
<u>อุปกรณ์ที่ใช้ในการทดลอง</u>

1. STM32CubeIDE
2. Proteus 8.9 SP 2

<u>ขั้นตอนการทดลอง</u>

<u>Digital Input</u>

1. ทำการกำหนดให้ PA1 เป็น GPIO_Input และกำหนดให้ PB0 เป็น GPIO_Output



2. ส่วนของโค้ดที่เกี่ยวข้อง

```
152 static void MX_GPIO_Init(void)
153 {
154   GPIO_InitTypeDef GPIO_InitStruct = {0};
155
156   /* GPIO Ports Clock Enable */
157   __HAL_RCC_GPIOB_CLK_ENABLE();
158   __HAL_RCC_GPIOA_CLK_ENABLE();
159
160   /*Configure GPIO pin Output Level */
161   HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, GPIO_PIN_RESET);
162
163   /*Configure GPIO pin : PB0 */
164   GPIO_InitStruct.Pin = GPIO_PIN_0;
165   GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
166   GPIO_InitStruct.Pull = GPIO_NOPULL;
167   GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
168   HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);
169
170   /*Configure GPIO pin : PA1 */
171   GPIO_InitStruct.Pin = GPIO_PIN_1;
172   GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
173   GPIO_InitStruct.Pull = GPIO_NOPULL;
174   HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
175
176 }
```

บรรทัดที่ 158 เป็นการเปิดสัญญาณนาฬิกาให้กับ GPIO Port A

บรรทัดที่ 171 เลือก Pin 1

บรรทัดที่ 172 กำหนดให้โหมดเป็น GPIO Input

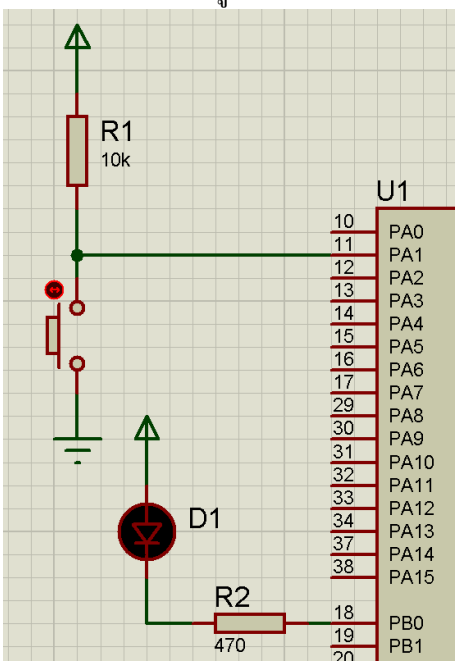
บรรทัดที่ 173 กำหนดให้เป็น No pull

3. โค้ดในส่วนของการอ่านค่าจาก Input

```
94   /* USER CODE BEGIN WHILE */
95   while (1)
96   {
97     /* USER CODE END WHILE */
98
99     /* USER CODE BEGIN 3 */
100    if(HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_1) == GPIO_PIN_RESET) HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, 1);
101    else HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, 0);
102  }
103  /* USER CODE END 3 */
```

บรรทัดที่ 100 อ่านค่าจาก PA1 ได้โดย function HAL_GPIO_ReadPin โดยต้องกำหนดพารามิเตอร์ 2 ค่า
- GPIOA คือ Port ที่ต้องการอ่านค่า
- GPIO_PIN_1 คือ Pin ที่ 1
เมื่อรวม 2 พารามิเตอร์เข้าด้วยกันจะหมายถึง PA1 เนื่องจากตัววงจรที่จะต่อใช้ให้ค่าเป็น 0(โลจิกต่ำ) เมื่อมีการกดปุ่ม หากไม่กดปุ่มจะมีค่าเป็น 1(โลจิกสูง) ดังนั้นจึงทำการเปรียบเทียบค่าหากที่อ่านเป็นโลจิกต่ำ (GPIO_PIN_RESET) ให้ทำการส่งโลจิกสูง ไปที่ PB0 หากไม่กดปุ่มจะส่งโลจิกต่ำที่ PB0

4. วงจรที่ใช้ในการทดลองสำหรับโค้ดด้านบนเป็นไปตามรูปด้านล่าง

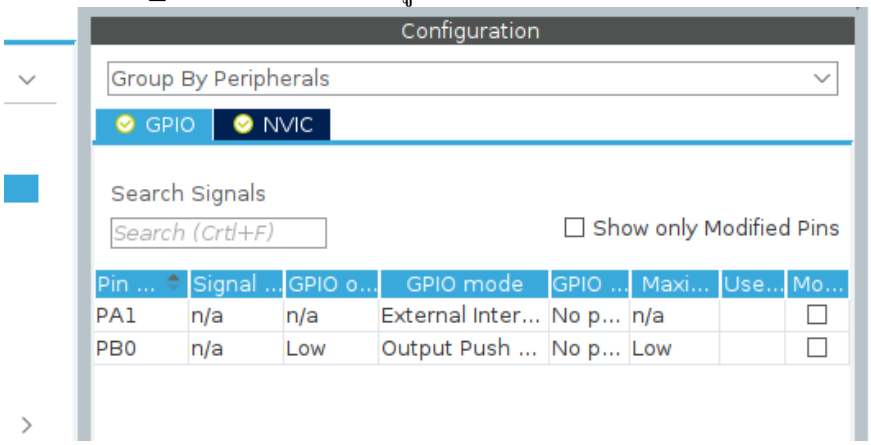


5. ให้ทดสอบโดยการกดปุ่มแล้วบันทึกผล

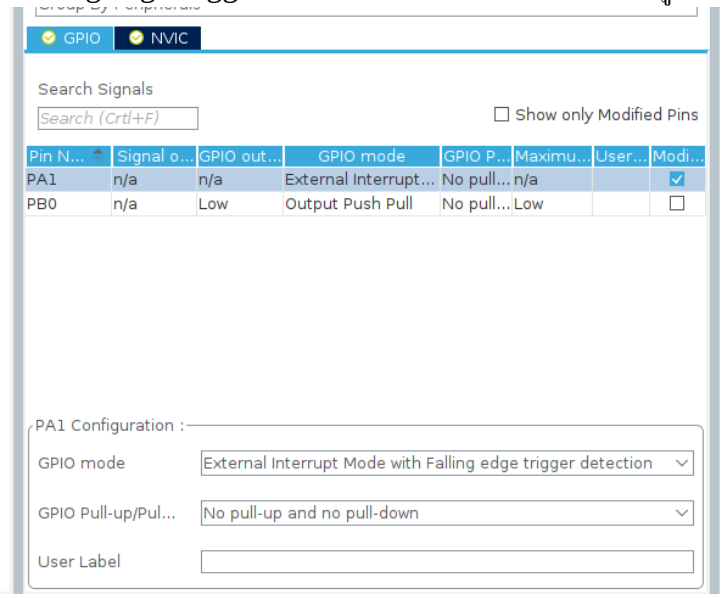
การกระทำกับปุ่ม	ผลที่ได้จาก LED (ติด/ดับ)
กดปุ่ม	
ไม่กดปุ่ม	

External Interrupt Line

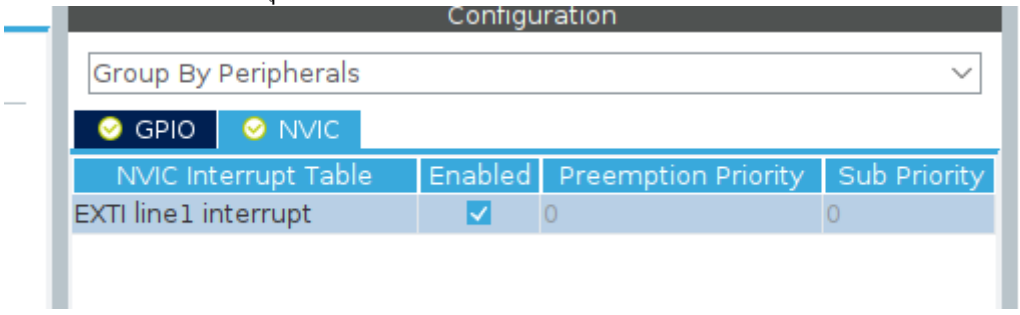
1. ให้ใช้วงจรเดิมในการทดลอง
2. ให้แก้ไข PA1 เป็น GPIO_EXTI1 จะทำให้มีเมนูในการ enable NVIC เพิ่มเข้ามา



3. ทำการกำหนดให้การเกิดอินเตอร์รัปท์ของ PA1 เป็นขอบขาลงได้โยการเลือกที่ PA1 และเลือก External Interrupt Mode with Falling edge trigger detection ตรง GPIO mode ตามรูปด้านล่าง



4. ไปที่ tab NVIC จากนั้น tick ไปที่ช่อง enable EXTI line1 Interrupt เพื่อเป็นการเปิดให้ PA1 เมื่อมีขอบของสัญญาณเกิดขึ้นจะไปทำการกระตุ้นให้เกิดการอินเตอร์รัปท์



5. กด save เพื่อ generate code

6. โค้ดในการกำหนดค่าที่เกี่ยวข้องกับ GPIO จะเปลี่ยนแปลงเพื่อให้สามารถทำงานแบบ EXTI Line Interrupt ได้

```
149 static void MX_GPIO_Init(void)
150 {
151   GPIO_InitTypeDef GPIO_InitStruct = {0};
152
153   /* GPIO Ports Clock Enable */
154   __HAL_RCC_GPIOB_CLK_ENABLE();
155   __HAL_RCC_GPIOA_CLK_ENABLE();
156
157   /*Configure GPIO pin Output Level */
158   HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, GPIO_PIN_RESET);
159
160   /*Configure GPIO pin : PB0 */
161   GPIO_InitStruct.Pin = GPIO_PIN_0;
162   GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
163   GPIO_InitStruct.Pull = GPIO_NOPULL;
164   GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
165   HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);
166
167   /*Configure GPIO pin : PA1 */
168   GPIO_InitStruct.Pin = GPIO_PIN_1;
169   GPIO_InitStruct.Mode = GPIO_MODE_IT_FALLING;
170   GPIO_InitStruct.Pull = GPIO_NOPULL;
171   HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
172
173   /* EXTI interrupt init */
174   HAL_NVIC_SetPriority(EXTI1_IRQn, 0, 0);
175   HAL_NVIC_EnableIRQ(EXTI1_IRQn);
176
177 }
```

บรรทัดที่ 169 เป็นการกำหนดให้ PA1 ทำงานในโหมด Interrupt ที่ขอบขาลง (Falling)
บรรทัดที่ 174 เป็นการกำหนด Priority ของ EXTI1_IRQ
บรรทัดที่ 175 เป็นการ enable EXTI1_IRQn (External Line 1 Interrupt ให้สามารถรับการเกิดอินเตอร์รัปท์

7. เพิ่มโค้ดในส่วนของไฟล์ main.h ตรงตามตำแหน่งตามรูป เพื่อให้ไฟล์ .c ต่างที่ include เข้าไปสามารถรู้จักตัวแปลที่ชื่อ speed และสามารถใช้งานตัวแปรนี้ข้ามไฟล์ได้ (extern)

```
48 /* Exported macro ------------------------
49 /* USER CODE BEGIN EM */
50 extern volatile unsigned int speed;
51
```

8. ประกาศตัวแปรที่ชื่อ speed ตามรูปด้านล่างในไฟล์ main.c เพื่อให้ตัวแปรดังกล่าวทำการจับจองหน่วยความจำ ทำให้สามารถเก็บค่าที่ต้องการได้

```
39 /* USER CODE BEGIN PM */
40 volatile unsigned int speed = 500;
41 /* USER CODE END PM */
42
```

9. ให้เพิ่มโค้ดในไฟล์ stm32f4xx_it.c ตรงส่วนของ function EXTI1_IRQHandler เมื่อเกิดการอินเตอร์รัปท์ ให้ปรับค่าของตัวแปร speed

```
202   */
203 void EXTI1_IRQHandler(void)
204 {
205   /* USER CODE BEGIN EXTI1_IRQn 0 */
206   speed = (speed>100)?100:500;
207
208   /* USER CODE END EXTI1_IRQn 0 */
209   HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_1);
210   /* USER CODE BEGIN EXTI1_IRQn 1 */
211
212   /* USER CODE END EXTI1_IRQn 1 */
213 }
214
```

10. ทำการแก้ไขโค้ดใน while ของไฟล์ main.c

```
94   /* USER CODE BEGIN WHILE */
95   while (1)
96   {
97     /* USER CODE END WHILE */
98
99     /* USER CODE BEGIN 3 */
100    HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_0);
101    HAL_Delay(speed);
102
103  }
104  /* USER CODE END 3 */
```

11. สังเกตุการกระพริบของหลอด LED เมื่อมีการกดปุ่มแล้วทำการบันทึกผลที่เกิดขึ้นด้านล่าง
HINT: ในไฟล์ main.c ไม่มีการเปลี่ยนค่าตัวแปร speed ใดๆ ทั้งสิ้น

……
……
……
……

<u>คำถามท้ายใบงาน</u>

1. จากคำถามท้ายการทดลองของใบงานที่ 5 ให้นิสิตเพิ่มปุ่มกดที่ PA1 จากนั้นให้โปรแกรมที่เขียนขึ้นให้ได้ผลการทำงานดังนี้
- ให้ step ที่ 1 – 8 ของใบงานที่ 5 เป็นรูปแบบการแสดงผลแบบที่ 1
- ให้ step ที่ 9 – 16 ของใบงานที่ 5 เป็นรูปแบบการแสดงผลแบบที่ 2 โดยเปลี่ยน step 9 เป็น step 1 ของการแสดงรูปแบบที่ 2 / step 10 ==> step 2 / …. step 16 ==> step 8
- เริ่มต้นให้แสดงผลรูปแบบที่ 1 ตั้งแต่ step 1 … 8 วนไปเรื่อย
- เมื่อมีการกดปุ่ม โปรแกรมจะทำการเปลี่ยนรูปแบบการแสดงผลสลับระหว่างรูปแบบที่ 1 และ 2 ในทุกๆ ครั้งกด หากไม่กดรูปแบบใดที่แสดงอยู่ก็ให้แสดงรูปแบบนั้นๆ ไปเรื่อย
- การเปลี่ยนรูปแบบการแสดงผล รูปแบบ 1 หรือ 2 จะสามารถขัดจังหวะได้เสมอ ไม่ต้องรอให้จบการแสดงรูปแบบนั้นๆ เช่น กำลังแสดงผล step ที่ 3 ของรูปแบบที่ 1 อยู่ หากมีการกดปุ่มจะเปลี่ยนไปแสดง step ที่ 4 ของรูปแบบที่ 2 ได้ต่อทันที
- การแสดงผล LED ในแต่ละ step จะห่างกันที่ 0.5 วินาที