

# ใบงานครั้งที่ 07

## Universal Asynchronous Receiver Transmitter (UART)

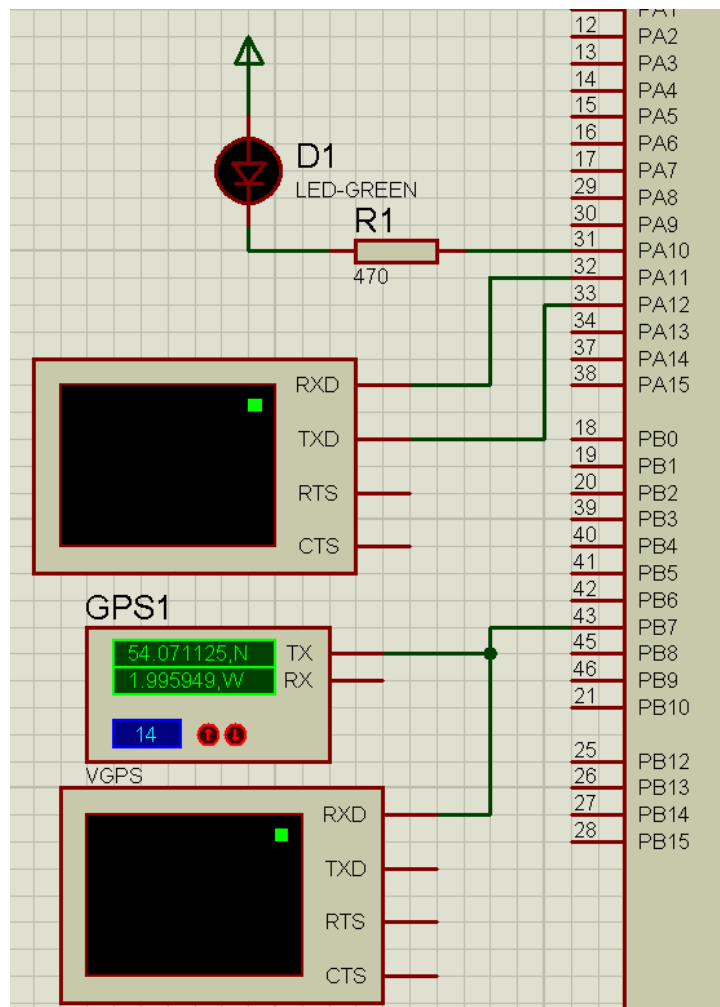
### อุปกรณ์ที่ใช้ในการทดลอง

1. STM32CubeIDE
2. Proteus 8.9 SP 2

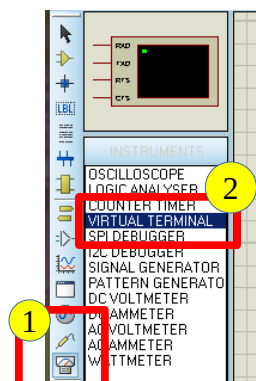
### ขั้นตอนการทดลอง

การทดลองเชื่อมต่อ MCU ไปที่ virtual terminal ผ่านทาง UART

1. ต่อดังตามรูปด้านล่าง



2. ตัว virtual terminal อยู่ใน tab ตามรูปด้านล่าง



3. เรียกการใช้ UART6 โดยไปที่ connectivity → USART6 → Mode เลือกเป็น Asynchronous และกำหนด

- Baud Rate เป็น 115200 Bits/S

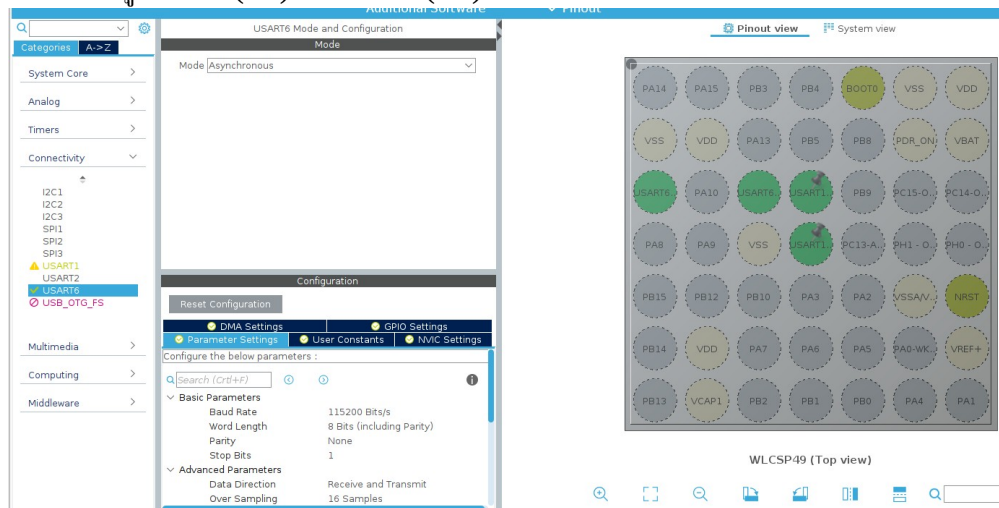
- Word Length เป็น 8 Bits

- Parity เป็น None

- Stop เป็น 1

(การกำหนดแบบนี้จะเรียกกันว่า 8-N-1)

- ขาของ UART6 จะอยู่ที่ PA11(Tx) และ PA12(Rx)



4. รายละเอียดของโค้ดที่ได้ configure ไว้ที่ UART6

```
203 static void MX_USART6_UART_Init(void)
204 {
205     /* USER CODE BEGIN USART6_Init 0 */
206
207     /* USER CODE END USART6_Init 0 */
208
209     /* USER CODE BEGIN USART6_Init 1 */
210
211     /* USER CODE END USART6_Init 1 */
212     huart6.Instance = USART6;
213     huart6.Init.BaudRate = 115200;
214     huart6.Init.WordLength = UART_WORDLENGTH_8B;
215     huart6.Init.StopBits = UART_STOPBITS_1;
216     huart6.Init.Parity = UART_PARITY_NONE;
217     huart6.Init.Mode = UART_MODE_TX_RX;
218     huart6.Init.HwFlowCtl = UART_HWCONTROL_NONE;
219     huart6.Init.OverSampling = UART_OVERSAMPLING_16;
220     if (HAL_UART_Init(&huart6) != HAL_OK)
221     {
222         Error_Handler();
223     }
224     /* USER CODE BEGIN USART6_Init 2 */
225
226     /* USER CODE END USART6_Init 2 */
227 }
228
229
```

5. เพิ่มโค้ดสำหรับรับ/ส่งข้อมูล ออกไปที่ UART6 ตามรูปด้านล่าง

```
77 /*
78 int main(void)
79 {
80     /* USER CODE BEGIN 1 */
81     unsigned char data[] = {'H', 'E', 'L', 'L', 'O', ' ', 'W', 'O', 'R', 'L', 'D', '!', 0x0A, 0x0D};
82     unsigned char ch;
83     /* USER CODE END 1 */
84
85     /* MCU Configuration-----*/

```

6. ทดลองได้รับข้อมูลจาก function HAL\_UART\_Receive มาเป็นไว้ที่ตัวแปร ch จำนวน 1 ตัวอักษร โดยมี timeout ที่ 100 mS

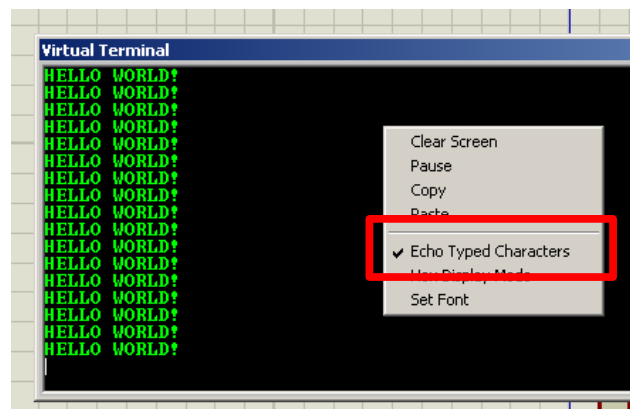
```

111  /* USER CODE BEGIN WHILE */
112  while (1)
113  {
114      /* USER CODE END WHILE */
115
116      /* USER CODE BEGIN 3 */
117      if(HAL_UART_Receive(&huart6, &ch, 1, 100)) {
118          if(ch=='1') HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, 0);
119          if(ch=='2') HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, 1);
120      }
121      HAL_UART_Transmit(&huart6, data, sizeof(data), 200);
122      HAL_Delay(500);
123  }
124  /* USER CODE END 3 */

```

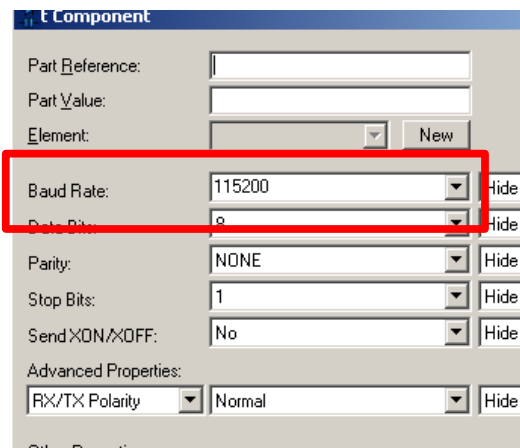
บรรทัดที่ 118 ตรวจสอบข้อมูลที่รับเข้ามาเป็นตัวอักษร '1' หรือไม่ ถ้าใช่ ให้ส่งลอจิกไปที่ LED ให้ติด  
 บรรทัดที่ 119 ตรวจสอบข้อมูลที่รับเข้ามาเป็นตัวอักษร '0' หรือไม่ ถ้าใช่ ให้ส่งลอจิกไปที่ LED ให้ดับ  
 บรรทัดที่ 121 เป็นการส่งข้อความใน array 'data' ทั้งหมด (sizeof)

7. ในช่วงที่ simulate อยู่ให้ click ขวาที่ virtual terminal แล้ว tick ที่ Echo Typed Characters เพื่อให้เวลาที่ทำการพิมพ์บนแป้นคีย์บอร์ด จะได้อ่านอักขระที่พิมพ์ ให้นิสิตทดลองพิมพ์เลข 1 และ 2 และบันทึกผล



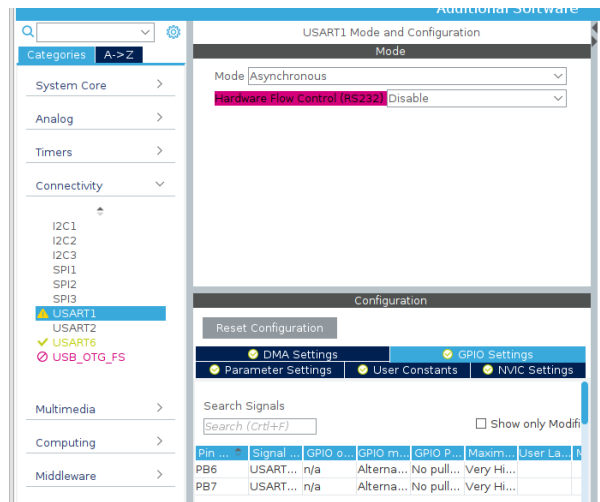
ทดลองพิมพ์	ผลการทดลองที่ LED (ติดหรือดับ และเวลาที่ LED เกิดการเปลี่ยนแปลงผล)
1	
2	

8. หากข้อความที่แสดงออกมาเป็นอักขระที่อ่านไม่ออก ให้นิสิตลองไปกำหนดค่า baud rate ของ virtual terminal ให้ตรงกับที่ได้ตั้งไว้ที่ MCU



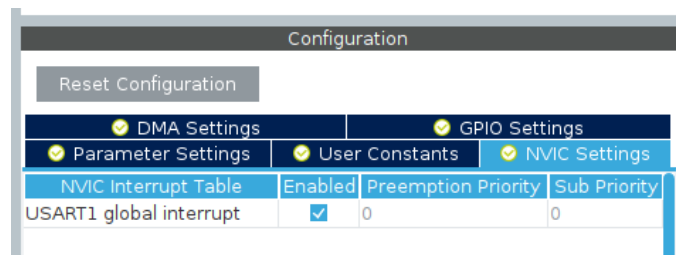
การทดลองเชื่อมต่อ MCU เข้ากับ GPS ผ่านทาง UART และทดลองการอินเทอร์รัปต์เมื่อรับข้อมูลเข้ามาอย่างต่อเนื่อง

1. ใน Proteus จะมีตัว GPS จำลองชื่อว่า VGPS อัตราการส่งข้อมูลจะอยู่ที่ 9600 บิต/วินาที รูปแบบ 8-N-1
2. set virtual terminal ที่เชื่อมกับ GPS ให้สามารถรับค่ามาแสดงผล (เพื่อ monitor)
3. ทำการกำหนดค่าให้กับ USART1 ตามนี้



- Baud rate: 9600
- 8-N-1
- PB6 (Tx), PB7 (Rx)

4. ทำการ enable ตัว USART1 ที่ NVIC



5. ทำการเพิ่มโค้ดที่จะทำงานเวลาเกิดอินเทอร์รัปต์ตอนรับข้อมูลด้วย function ตามรูปด้านล่าง

```
60 /* Private user code .....*/
61 /* USER CODE BEGIN 0 */
62 unsigned char msg[10];
63 void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
64 {
65     if(huart->Instance == USART1)
66     {
67         HAL_UART_Transmit(&huart6, msg, 10, 100);
68         HAL_UART_Receive_IT(&huart1, msg, 10);
69     }
70 }
71 /* USER CODE END 0 */
72
73 /**
74  * @brief The application entry point.
75  * @retval int
76  */
77 int main(void)
78 {
79 }
```

บรรทัดที่ 65 ทำการตรวจสอบตัว instance ที่ส่งเข้ามา หากเป็น USART1 แสดงว่า UART1 รับข้อมูลเสร็จสิ้นจึงอินเทอร์รัปต์เข้ามา

บรรทัดที่ 67 ทำการส่งข้อมูลที่รับเข้ามาออกไปที่ UART6

บรรทัดที่ 68 function สำหรับรับข้อมูล HAL\_UART\_Receive\_IT เมื่อข้อมูลรับเสร็จตามจำนวน (10 ตัว) จะร้องขออินเทอร์รัปต์รอบต่อไป

## 6. เพิ่มโค้ดในการรับข้อมูลเพื่อให้เกิดการอินเทอร์รัปต์ครั้งแรก

```
101  /* Initialize all configured peripherals */
102  MX_GPIO_Init();
103  MX_USART1_UART_Init();
104  MX_USART6_UART_Init();
105  /* USER CODE BEGIN 2 */
106  HAL_UART_Receive_IT(&huart1, msg, 10);
107
108  /* USER CODE END 2 */
```

## 7. แก้ไขโค้ดตรง while ตามรูป

```
111  /* USER CODE BEGIN WHILE */
112  while (1)
113  {
114      /* USER CODE END WHILE */
115
116      /* USER CODE BEGIN 3 */
117      if(HAL_UART_Receive(&huart6, &ch, 1, 100)) {
118          if(ch=='1') HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, 0);
119          if(ch=='2') HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, 1);
120      }
121      //HAL_UART_Transmit(&huart6, data, sizeof(data), 200);
122      //HAL_Delay(500);
123  }
```

## คำถามท้ายใบงาน

### 1. ต่อบอร์ดตามรูป จากนั้นให้เขียนโปรแกรมเพื่อให้ได้ผลตามนี้

- LED D2 ให้กะพริบทุกๆ 0.5 วินาที
- เมื่อส่งข้อมูล เป็น '1' ที่ virtual terminal ให้LED D1 ติด
- เมื่อส่งข้อมูล เป็น '2' ที่ virtual terminal ให้LED D1 ดับ
- การเปลี่ยนของ D1 จะกระทำทันที เมื่อส่งข้อมูลจาก virtual terminal; LED D1 ไม่มีการหน่วงเวลาใดๆ ทั้งสิ้น

