

9주차 보고서

월요일 분반 2조

조원)

201524566 정경호

201724539 이재욱

201724579 정현모

201814331 이해민

실험 목표

- Bluetooth 모듈 (FB755AC) 를 이용한 스마트폰과의 통신을 진행한다.
- 기판 납땜을 통해 보드와 모듈을 연결한다.
- Bluetooth의 Config를 설정한다.
- 안드로이드의 Serial Bluetooth Terminal 어플리케이션을 이용하여 PC의 putty와 통신한다. PC의 putty에서 입력 시 Bluetooth 모듈을 통해 스마트폰의 터미널에 출력하고, 스마트폰의 터미널에 입력 시 PC의 putty에 출력한다.

실험 과정

1. Enable the APB1 peripheral clock (RCC_Configure)

RCC_APB1PeriphClockCmd 함수를 이용해 USART2를 ENABLE 시켜주었다. 기존 USART1이 RCC_APB2를 사용한 것에 반해 APB1에 있다는 점을 주의해야 한다.

```
RCC_APB1PeriphClockCmd(RCC_APB1Periph_USART2, ENABLE); // USART2
```

2. Initialize the GPIO pins (GPIO_Configure)

USART2의 RX, TX GPIO를 설정해준다. USART2의 RX는 PA3, TX는 PA2에 위치하고 있다.

```
/* UART2 pin setting */
//TX a2
GPIO_InitTypeDef GPIO_InitStructure3;
GPIO_InitStructure3.GPIO_Pin = GPIO_Pin_2;
GPIO_InitStructure3.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_InitStructure3.GPIO_Mode = GPIO_Mode_AF_PP;
GPIO_Init(GPIOA, &GPIO_InitStructure3);

//RX a3
GPIO_InitTypeDef GPIO_InitStructure4;
GPIO_InitStructure4.GPIO_Pin = GPIO_Pin_3;
GPIO_InitStructure4.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_InitStructure4.GPIO_Mode = (GPIO_Mode_IPD) | (GPIO_Mode_IPU);
GPIO_Init(GPIOA, &GPIO_InitStructure4);
```

3. Initialize the USART & Enable the USART2 RX interrupts (USART2_Init)

이전 과제에서 진행했던 USART1과 동일하게 USART2를 설정해 준다.

```
// Enable the USART2 peripheral
USART_Cmd(USART2, ENABLE);

// TODO: Initialize the USART using the structure 'USART_InitTypeDef' and the function 'USART_Init'
USART_InitTypeDef USART2_InitStructure;
USART2_InitStructure.USART_BaudRate = 9600; // Baud Rate 9600
USART2_InitStructure.USART_WordLength = USART_WordLength_88; // word length 8bit
USART2_InitStructure.USART_StopBits = USART_StopBits_1; // stop bit 1bit
USART2_InitStructure.USART_Parity = USART_Parity_No ; // no parity bits
USART2_InitStructure.USART_Mode = USART_Mode_Rx | USART_Mode_Tx; // rx&tx mode
USART2_InitStructure.USART_HardwareFlowControl = USART_HardwareFlowControl_None;
USART2_Init(USART2,&USART2_InitStructure); // USART2

// TODO: Enable the USART2 RX interrupts using the function 'USART_ITConfig' and the argument value 'Receive Data register not empty i
USART_ITConfig(USART2, USART_IT_RXNE, ENABLE);
```

4. Initialize the NVIC (NVIC_Configure)

이전 과제와 동일하게 PriorityGroup 2를 사용하였고, USART2만 추가해 주었다.

```
// \ {\tt TODO: Initialize the NVIC using the structure 'NVIC_InitTypeDef' and the function 'NVIC_Init'} \\
NVIC_InitTypeDef NVIC_InitStructure;
// TODO: fill the arg you want
NVIC_PriorityGroupConfig(NVIC_PriorityGroup_2);
NVIC_EnableIRQ(USART1_IRQn);
NVIC InitStructure.NVIC IROChannel = USART1 IROn:
NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
NVIC InitStructure.NVIC IRQChannelSubPriority = 1;
NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
NVIC_Init(&NVIC_InitStructure);
// 'NVIC_EnableIRQ' is only required for USART setting
NVIC EnableIRQ(USART2_IRQn);
NVIC InitStructure.NVIC IRQChannel = USART2 IRQn;
NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 1;
NVIC_InitStructure.NVIC_IRQChannelSubPriority = 2;
NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
NVIC_Init(&NVIC_InitStructure);
```

5. : When the word is received (USART1_IRQHandler)

USART1에서 들어오는 입력은 USART2로 전송되게끔 함수를 작성했다. sendDataUART2를 이용해 USART1에서 받은 입력을 USART2로 전송한다.

```
uint16_t word;
if(USART_GetITStatus(USART1,USART_IT_RXNE)!=RESET){

// the most recent received data by the USART1 peripheral
word = USART_ReceiveData(USART1);
sendDataUART2(word); // send to USART2

// clear 'Read data register not empty' flag
USART_ClearITPendingBit(USART1,USART_IT_RXNE);
}
```

6. When the word is received (USART2 IRQHandler)

USART2에서 들어오는 입력은 USART1로 전송되게끔 함수를 작성했다. sendDataUART1를 이용해 USART2에서 받은 입력을 USART1로 전송한다.

```
uint16_t word;
if(USART_GetITStatus(USART2,USART_IT_RXNE)!=RESET){

// the most recent received data by the USART2 peripheral
word = USART_ReceiveData(USART2);
sendDataUART1(word); // send to USART1

// clear 'Read data register not empty' flag
USART_ClearITPendingBit(USART2,USART_IT_RXNE);
}
```

7. 블루투스 설정

아래 그림은 블루투스 config 초기 화면이다. Device name이 **기본값**이 설정되어 있고, PINCODE는 **1234**, CONNECTION MODE는 **CNT_MODE4**, ROLE은 **SLAVE**, UART CONFIG는 **(9600, 8, n, 1)**으로 설정되어 있다. Device name을 제외하고 모두 그대로 사용하면 되어서, Device name만 MON 02로 바꿔준다.

```
====== TOP MENU =====
0 => DEVICE NAME
                         : FB755v1.2.6
1 => AUTHENTICATION
                         : ENABLE PINCODE[1234]
2 => REMOTE BD ADDRESS
                         : 507705DA6AF1
    LOCAL BD ADDRESS
                         : 00189A24DD30
3 => CONNECTION MODE
                         : CNT MODE4
4 => OTHER PARAMETER
                         : E,D,5,2B,2,D
5 => UART CONFIG
6 => ROLE
                         : SLAVE
                         : OP MODEO
7 => OPERATION MODE
[ Back Spcae : Input data Cancel
[ t : Move top menu
Select(0 \sim 7) > 0
Change Device name
Within 12 character > MON 02
```

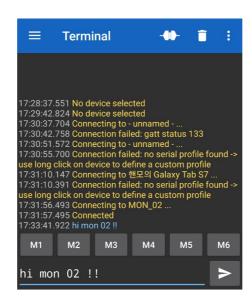
8. 블루투스 스마트폰 연결

블루투스 모듈의 Config선 연결을 해제하고 다시 시작하면 아래와 같이 Slave mode start가 뜨게 된다. 휴대폰에서 블루투스로 MON_02를 검색하여 연결하는 모습이다.

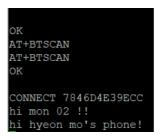


실험 결과

1. 스마트폰에서 "hi mon 02 !!"를 전송하는 모습.



2. hi mon 02 !!가 putty에 정상적으로 출력되는 모습. hi hyeon mo's phone!을 putty에서 입력하고 있는 모습.



3. hi hyeon mo's phone!이 스마트폰 terminal에 정상적으로 출력되는 모습.

