

# 2021년 2학기 임베디드 시스템 설계 및 실험

- 9주차 보고서 -

004분반 9조

## 개요

이번 실험에서는 Bluetooth 동작 및 기판 납땜 실험을 진행하였다.

## 목표

- Bluetooth 모듈 (FB755AC) 를 이용한 스마트폰과의 통신 및 기판 납땜을 통해 보드와 모듈 연결

## 세부실험내용

- 블루투스를 통하여 PC - 보드 - 스마트폰의 UART 통신 수행
- 기판 납땜을 통해 보드와 모듈 연결

## 개념

- 블루투스 : 근거리 무선통신기술
- 블루투스 프로파일 : 어플리케이션 관점에서 블루투스 기기의 기능별 성능을 정하는 사양
- SSID : 무선랜을 통해 클라이언트가 접속할 때 각 무선랜을 구별하기 위한 고유 식별자
- UUID : 네트워크 상에서 서로 다른 개체들을 구별하기 위한 128비트 고유 식별자

## 실험과정

### - 실습

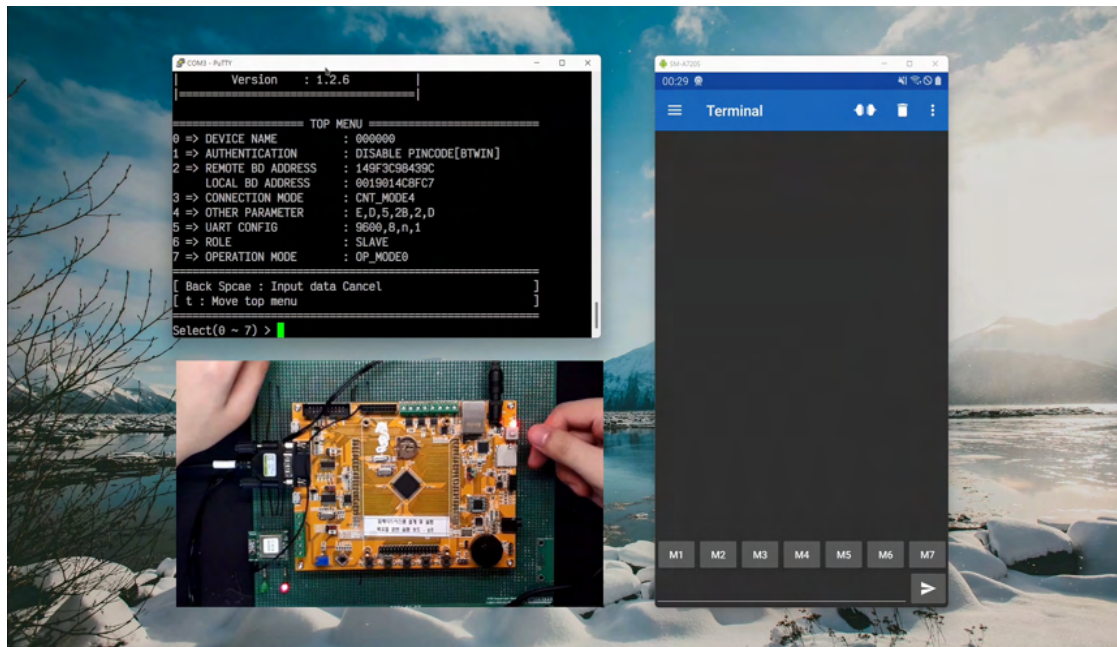


그림1) 버튼 입력시 PUTTY에 설정메뉴 출력

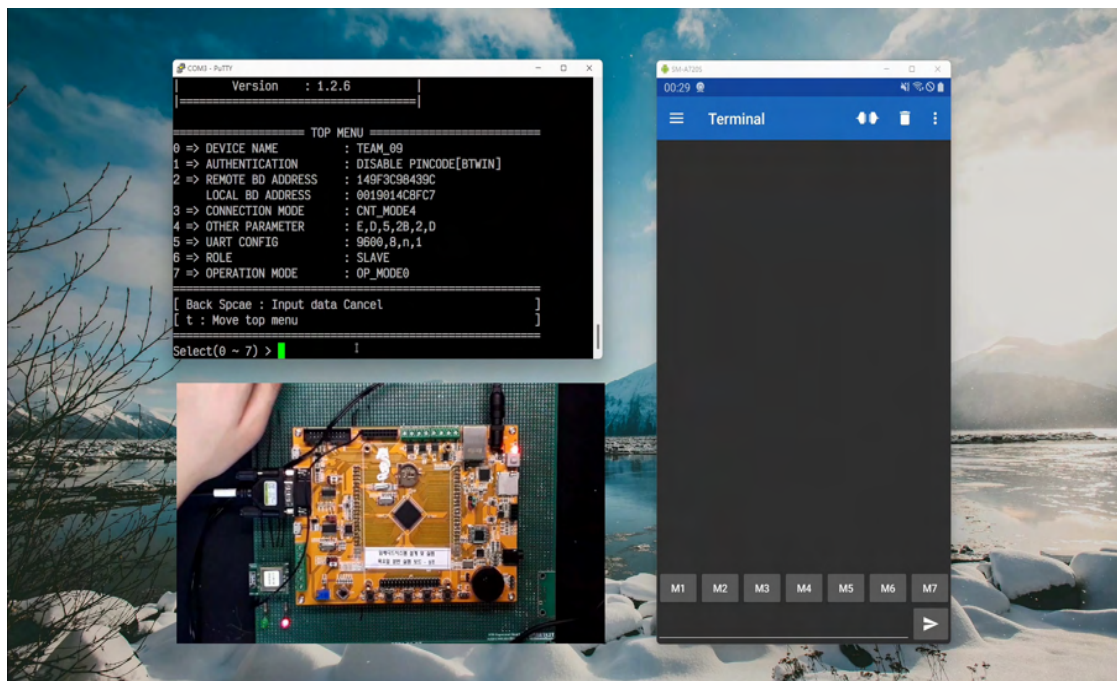


그림2) Putty에서 TEAM09 으로 디바이스명 변경

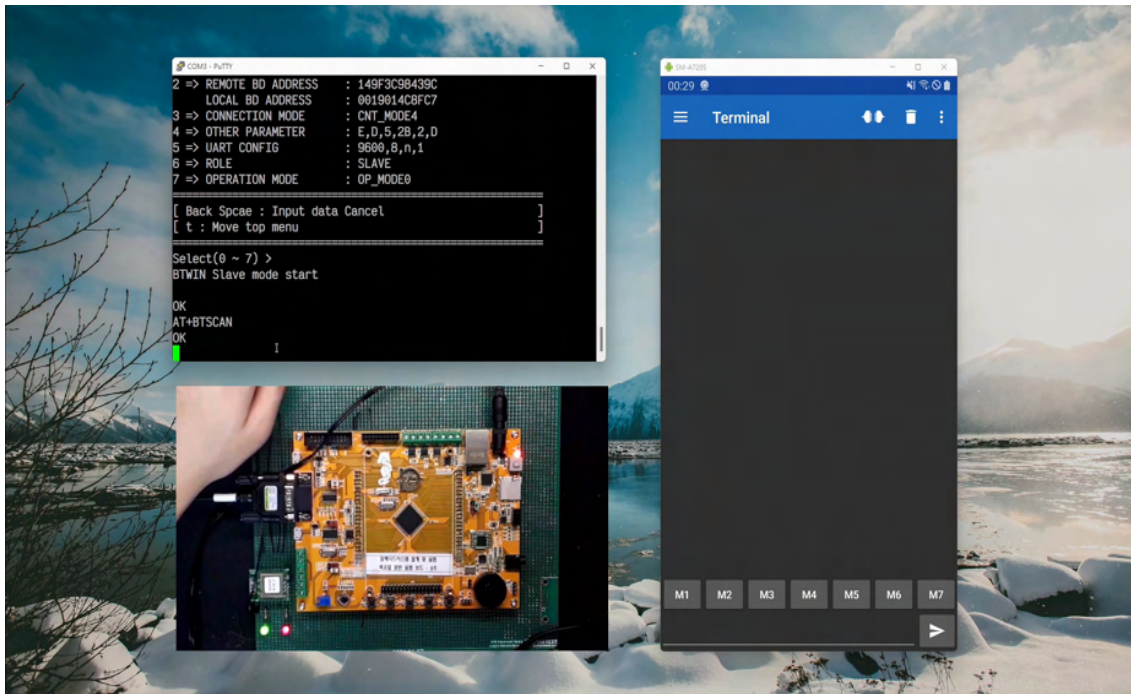


그림3) 명령어를 통하여 블루투스 신호 체크

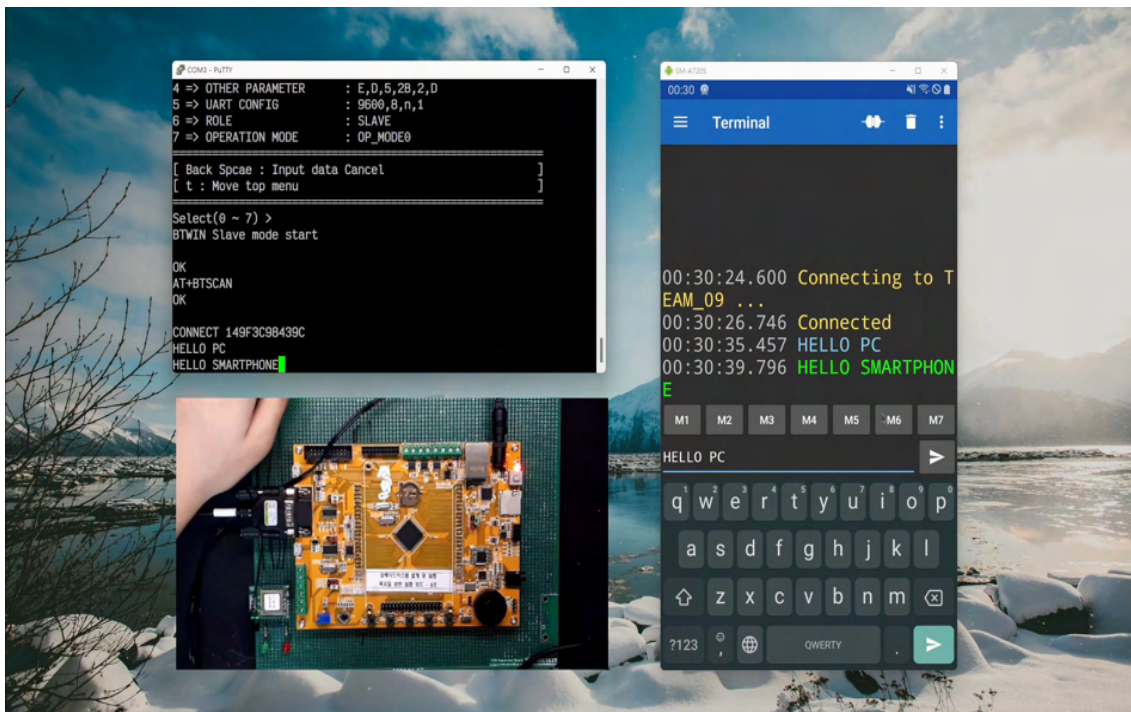


그림4) 스마트폰과 블루투스 통신



```
1  #include "stm32f10x.h"
2  #include "stm32f10x_gpio.h"
3  #include "stm32f10x_usart.h"
4  #include "stm32f10x_rcc.h"
5
6  #include "misc.h"
7
8  /* function prototype */
9  void RCC_Configure(void);
10 void GPIO_Configure(void);
11 void EXTI_Configure(void);
12 void USART_Init(void);
13 void NVIC_Configure(void);
14
15 void EXTI15_10_IRQHandler(void);
16 void EXTI9_5_IRQHandler(void);
17 void EXTI2_IRQHandler(void);
18
19 void Delay(void);
20
21 void sendDataUART1(uint16_t data);
22 void sendDataUART2(uint16_t data);
23
24 char dataBufferFromPC;
25 char flagPC = 0;
26
27 char dataBufferFromBT;
28 char flagBT = 0;
29
30
31 //-----
32
33 void RCC_Configure(void) // stm32f10x_rcc.h 참고
34 {
35     /* USART pin clock enable */
36     RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);
37
38     /* USART1 clock enable */
39     RCC_APB2PeriphClockCmd(RCC_APB2ENR_USART1EN, ENABLE);
40
41     /* USART2 clock enable */
42     RCC_APB1PeriphClockCmd(RCC_APB1ENR_USART2EN, ENABLE);
43
44     /* AFIO clock enable */
45     RCC_APB2PeriphClockCmd(RCC_APB2Periph_AFIO, ENABLE);
46 }
47
48
49 void GPIO_Configure(void) // stm32f10x_gpio.h 참고
50 {
51     GPIO_InitTypeDef GPIO_InitStructure;
52
53     /* UART between PC pin setting */
54     //TX
55     GPIO_InitStructure.GPIO_Pin = GPIO_Pin_9;
56     GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
57     GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
58     GPIO_Init(GPIOA, &GPIO_InitStructure);
59
60     //RX
61     GPIO_InitStructure.GPIO_Pin = GPIO_Pin_10;
62     GPIO_InitStructure.GPIO_Mode = GPIO_Mode_INP;
63     GPIO_Init(GPIOA, &GPIO_InitStructure);
64 }
```

```

65      /* UART between bluetooth pin setting */
66      //TX
67      GPIO_InitStructure.GPIO_Pin = GPIO_Pin_2;
68      GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
69      GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
70      GPIO_Init(GPIOA, &GPIO_InitStructure);
71
72      //RX
73      GPIO_InitStructure.GPIO_Pin = GPIO_Pin_3;
74      GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IPU;
75      GPIO_Init(GPIOA, &GPIO_InitStructure);
76  }
77
78
79
80  void USARTS_Init(void)
81  {
82      USART_InitTypeDef USART_InitStructure;
83
84      // Enable the USART1, USART2 peripheral
85      USART_Cmd(USART1, ENABLE);
86      USART_Cmd(USART2, ENABLE);
87
88      /*
89      BaudRate: 9600
90      WordLength: 8bits
91      Parity: None
92      StopBits: 1bit
93      Hardware Flow Control: None
94      */
95      USART_InitStructure.USART_BaudRate = 9600;
96      USART_InitStructure.USART_WordLength = (uint16_t) USART_WordLength_8b;
97      USART_InitStructure.USART_Mode = USART_Mode_Rx | USART_Mode_Tx;
98      USART_InitStructure.USART_Parity = (uint16_t) USART_Parity_No;
99      USART_InitStructure.USART_StopBits = (uint16_t) USART_StopBits_1;
100     USART_InitStructure.USART_HardwareFlowControl = (uint16_t) USART_HardwareFlowControl_None;
101
102     USART_Init(USART1, &USART_InitStructure);
103     USART_Init(USART2, &USART_InitStructure);
104
105
106     USART_ITConfig(USART1, USART_IT_RXNE, ENABLE);
107     USART_ITConfig(USART2, USART_IT_RXNE, ENABLE);
108 }
109
110 void NVIC_Configure(void) { // misc.h 참고
111
112     NVIC_InitTypeDef NVIC_InitStructure;
113
114     NVIC_PriorityGroupConfig(NVIC_PriorityGroup_1);
115
116     // UART1
117     NVIC_EnableIRQ(USART1_IRQn);
118     NVIC_InitStructure.NVIC_IRQChannel = USART1_IRQn;
119     NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0x0;
120     NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0x0;
121     NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
122     NVIC_Init(&NVIC_InitStructure);
123
124     // UART2
125     NVIC_EnableIRQ(USART2_IRQn);
126     NVIC_InitStructure.NVIC_IRQChannel = USART2_IRQn;
127     NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0x1;

```

```

128     NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0x0;
129     NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
130     NVIC_Init(&NVIC_InitStructure);
131 }
132
133 void USART1_IRQHandler() {
134     if(USART_GetITStatus(USART1,USART_IT_RXNE)!=RESET){
135         flagPC = 1;
136         dataBufferFromPC = USART_ReceiveData(USART1);
137
138         USART_ClearITPendingBit(USART1,USART_IT_RXNE);
139     }
140 }
141
142 void USART2_IRQHandler() {
143     if(USART_GetITStatus(USART2,USART_IT_RXNE)!=RESET){
144         flagBT = 1;
145         dataBufferFromBT = USART_ReceiveData(USART2);
146
147         USART_ClearITPendingBit(USART2,USART_IT_RXNE);
148     }
149 }
150
151 void sendDataUART1(uint16_t data) {
152     USART_SendData(USART1, data);
153 }
154
155 void sendDataUART2(uint16_t data) {
156     USART_SendData(USART2, data);
157 }
158
159 int main(void)
160 {
161     SystemInit();
162
163     RCC_Configure();
164
165     GPIO_Configure();
166
167     USARTS_Init();
168
169     NVIC_Configure();
170
171     while (1) {
172         if(flagPC == 1) {
173             sendDataUART2(dataBufferFromPC);
174             flagPC = 0;
175         } else if (flagBT == 1) {
176             sendDataUART1(dataBufferFromBT);
177             flagBT = 0;
178         }
179     }
180     return 0;
181 }
182

```

## 결론

이번 실험에서는 블루투스를 통한 보드와 PC간의 통신 및 납땜 실습을 진행하였다. 코드부분은 저번주 실습과 큰 차이점은 없었지만 블루투스와의 보드 통신에 NVIC을 이용한 것 처럼 블루투스의 LED 출력부분에서도 NVIC를 사용하면 동작의 신뢰성이 높아지지 않을까 생각하였지만 LED의 출력은 블루투스의 수신여부를 확인할 뿐이라는 점과 출력된 신호는 LED를 거친 후 소멸한다는 것을 고려하였을때 사용하지 않는 것이 좋겠다는 결론을 내게 되었다. 보드의 납땜은 조원 대부분이 경험이 없어 초반엔 힘들었지만 잘 진행되었고 결과물은 정상동작 함을 확인하였다.