



# Synchronization (KeyPad & FND)

조진성

경희대학교 컴퓨터공학과

Mobile & Embedded System Lab.



# Synchronization



## ❖ Semaphore in FreeRTOS

- SemaphoreHandle\_t xSemaphoreCreateBinary(void);
- SemaphoreHandle\_t xSemaphoreCreateCounting(  
UBaseType\_t uxMaxCount, UBaseType\_t uxInitialCount);

## ❖ Mutex in FreeRTOS

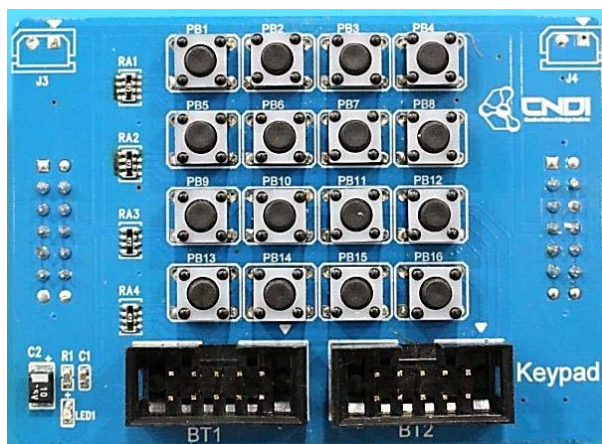
- SemaphoreHandle\_t xSemaphoreCreateMutex(void);
  - Support PIP(Priority Inheritance Protocol)

## ❖ Semaphore/Mutex operation in FreeRTOS

- int xSemaphoreTake(SemaphoreHandle\_t xSemaphore,  
TickType\_t xTicksToWait);
- int xSemaphoreGive(SemaphoreHandle\_t xSemaphore);

# 하드웨어 구성

- ❖ Keypad 버튼 4개(PB1, PB2, PB3, PB4)를 누르면 FND에 왼쪽으로 shift되며 표시



# 하드웨어 구성



## ❖ Keypad 버튼 4개(PB1, PB2, PB3, PB4)를 누르면 FND에 왼쪽으로 shift되며 표시

### ■ Keypad

- Arduino MEGA ADK의 DIGITAL 2(*Interrupt 0*)와 Keypad 모듈의 PB1을 연결
- Arduino MEGA ADK의 DIGITAL 3(*Interrupt 1*)와 Keypad 모듈의 PB2을 연결
- Arduino MEGA ADK의 DIGITAL 21(*Interrupt 2*)와 Keypad 모듈의 PB3을 연결
- Arduino MEGA ADK의 DIGITAL 20(*Interrupt 3*)와 Keypad 모듈의 PB4을 연결
- Arduino MEGA ADK의 3.3V와 Keypad 모듈의 J4 포트의 +3.3V와 연결
- Arduino MEGA ADK의 GND(0V)와 Keypad 모듈의 J4 포트의 좌측에 연결

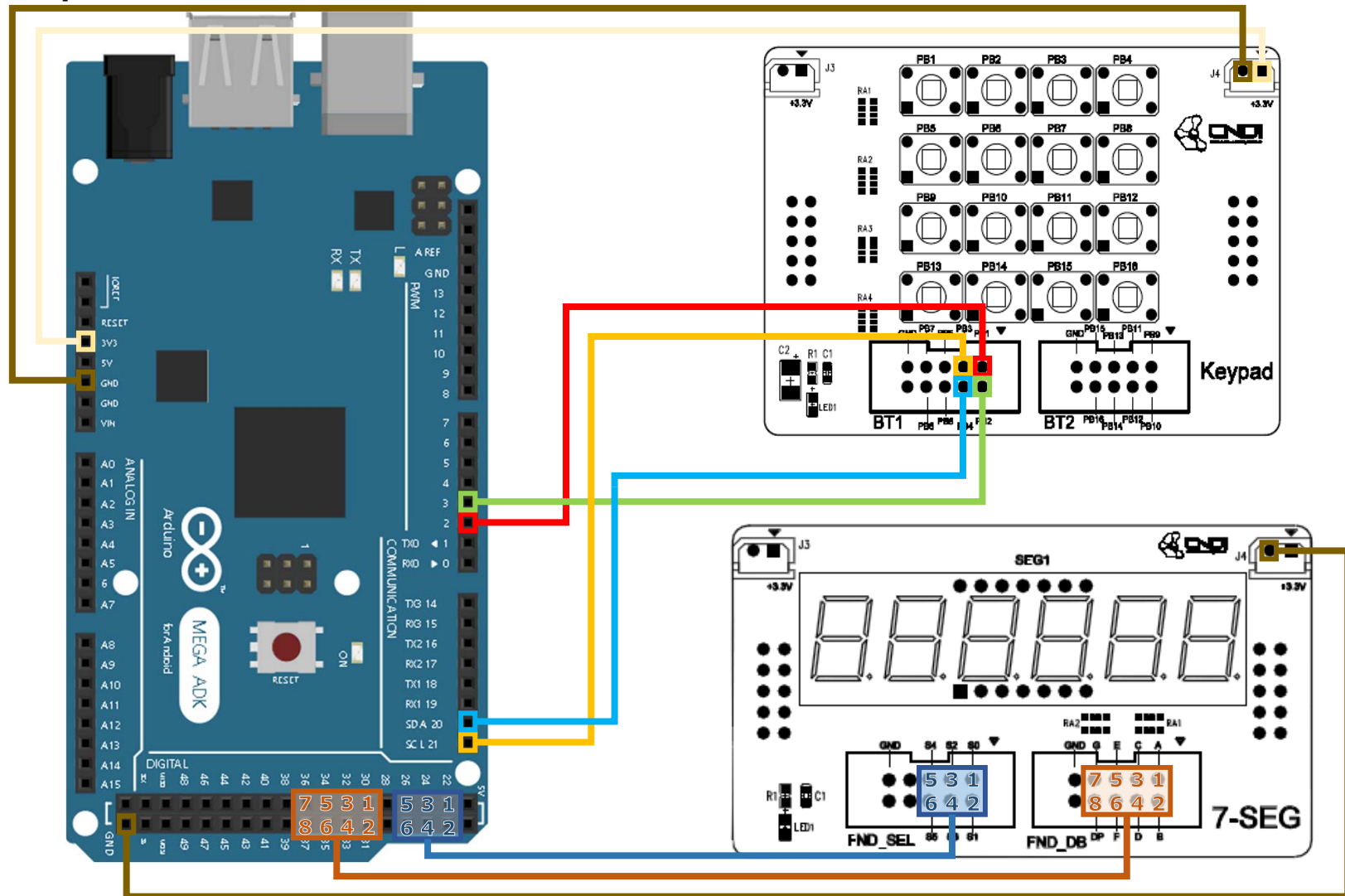
### ■ FND

- Arduino MEGA ADK의 DIGITAL 22~27과 FND 모듈의 S0~S5를 연결
- Arduino MEGA ADK의 DIGITAL 30~37과 FND 모듈의 A~DP를 연결
- Arduino MEGA ADK의 GND(0V)와 FND 모듈의 J4 포트의 좌측에 연결



# 하드웨어 구성

- ❖ Keypad 버튼 4개(PB1, PB2, PB3, PB4)를 누르면 FND에 왼쪽으로 shift되며 표시



# 실습 예제



## ❖ Keypad ISR과 Keypad Task의 동기화

- Keypad Task에서는 Keypad ISR로부터의 버튼 전달을 기다림
- Keypad ISR에서 눌린 버튼을 전역변수를 통해 Keypad에 전달
- 동기화를 위해 binary semaphore 사용

## ❖ Keypad Task와 FND Task

- FND Task에서는 Keypad Task로부터의 버튼 전달을 기다림
- Keypad Task는 눌린 버튼을 전역 변수를 통해 FND Task에 전달
- FND Task는 전달 받은 버튼을 FND에 왼쪽으로 shift하며 표시

# 실습 예제



- ❖ Keypad 버튼 4개(PB1, PB2, PB3, PB4)를 누르면 FND에 왼쪽으로 shift되며 표시 (lab3-4.ino)

```
#include <FreeRTOS_AVR.h>

#define MS2TICKS(ms) (ms / portTICK_PERIOD_MS)
#define FND_SIZE 6

const int Keypad[4] = { 2, 3, 21, 20 };
const int FndSelectPin[6] = { 22, 23, 24, 25, 26, 27 };
const int FndPin[8] = { 30, 31, 32, 33, 34, 35, 36, 37 };
const int FndFont[10] = { 0x3F, 0x06, 0x5B, 0x4F, 0x66,
                        0x60, 0x7D, 0x07, 0x7F, 0x67 };

SemaphoreHandle_t Sem;
Int SendValue = 0;
int Fnd[ FND_SIZE ] = { 0, };

// FND의 값을 왼쪽으로 Shift하고
// 파라미터로 들어온 새로운 값(data)를 0번지에 넣는 함수
void ShiftInsert( int data ) {
    int i;

    for( i = 1; i < FND_SIZE; i++ ) {
        Fnd[ FND_SIZE - i ] = Fnd[ FND_SIZE - i - 1 ];
    }
    Fnd[ 0 ] = data;
}
```

# 실습 예제



- ❖ Keypad 버튼 4개(PB1, PB2, PB3, PB4)를 누르면 FND에 왼쪽으로 shift되며 표시

```
// Keypad ISR
// 전역 변수(sendValue)로 값을 전달하기 때문에 세마포로 보호
void KeypadControl1() {
    delay(50);
    SendValue = 1;
    xSemaphoreGive(Sem);
}

void KeypadControl2() {
    delay(50);
    SendValue = 2;
    xSemaphoreGive(Sem);
}

void KeypadControl3() {
    delay(50);
    SendValue = 3;
    xSemaphoreGive(Sem);
}

void KeypadControl4() {
    delay(50);
    SendValue = 4;
    xSemaphoreGive(Sem);
}
```

```
void KeypadTask( void* arg ) {
    int i;
    int keypad;

    while(1) {
        // 세마포어를 통해 키패드가 눌렸음을 Keypad ISR로부터
        // 전달 받음
        // portMAX_DELAY: 세마포를 받기 전까지 Block
        // 입력된 번호는 ShiftInsert를 통해 Fnd[] 에 삽입
        if( xSemaphoreTake(Sem, portMAX_DELAY) ) {
            keypad = SendValue;
            ShiftInsert(keypad);
        }
    }
}
```



# 실습 예제



- ❖ Keypad 버튼 4개(PB1, PB2, PB3, PB4)를 누르면 FND에 왼쪽으로 shift되며 표시

```
void FndSelect( int pos ) {
    int i;

    for( i = 0; i < 6; i++ ) {
        if( i == pos ) {
            digitalWrite(FndSelectPin[ i ], LOW);
        }
        else {
            digitalWrite(FndSelectPin[ i ], HIGH);
        }
    }
}

void FndDisplay( int pos, int num ) {
    int i;
    int flag = 0;
    int shift = 0x01;

    FndSelect( pos );

    for( i = 0; i < 8; i++ ) {
        flag = ( FndFont[ num ] & shift );
        digitalWrite( FndPin[ i ], flag );
        shift <<= 1;
    }
}
```

```
void FndTask( void* arg ) {
    int i;

    // 지속적으로 Fnd[ ] 데이터를 FND에 출력
    while(1) {
        for( i = 0; i < FND_SIZE; i++ ) {
            delay(3);
            FndDisplay( i, Fnd[ i ] );
        }
    }
}
```

# 실습 예제

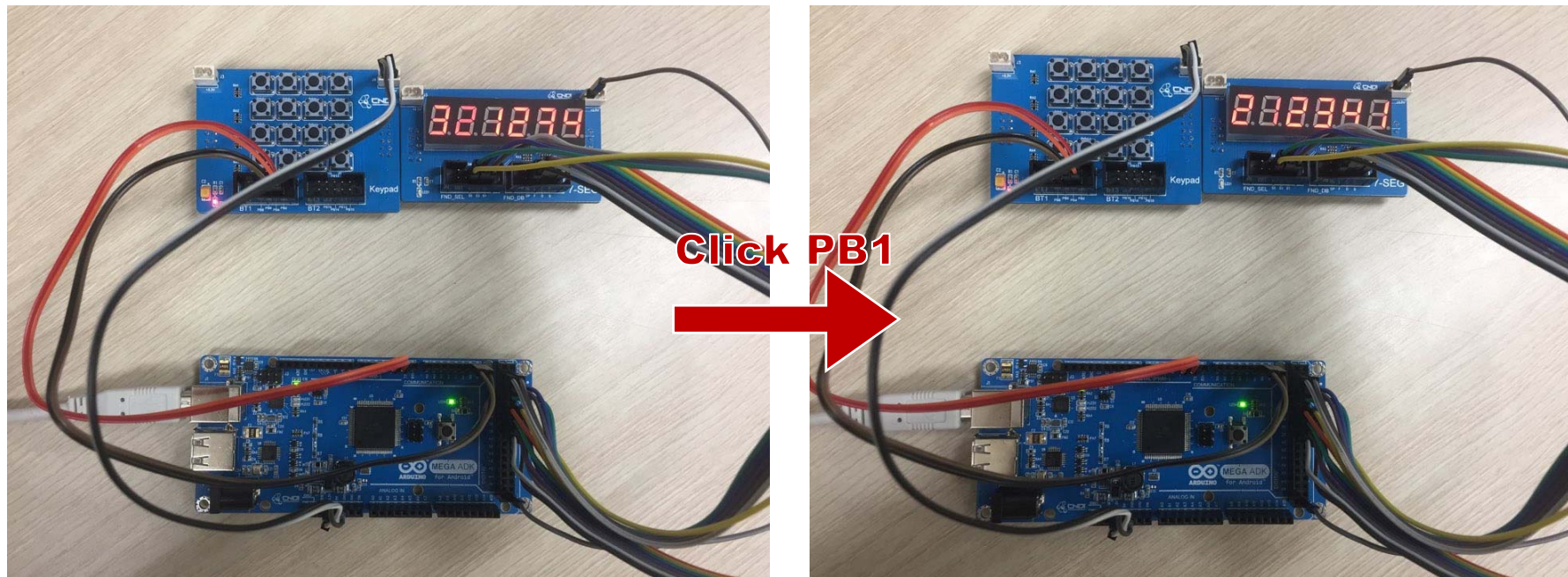


- ❖ Keypad 버튼 4개(PB1, PB2, PB3, PB4)를 누르면 FND에 왼쪽으로 shift되며 표시

```
void setup() {  
    int i;  
  
    for( i = 0; i < 6; i++ ) {  
        pinMode( FndSelectPin[ i ], OUTPUT );  
    }  
    for( i = 0; i < 8; i++ ) {  
        pinMode( FndPin[ i ], OUTPUT );  
    }  
    for( i = 0; i < 4; i++ ) {  
        pinMode( Keypad[ i ], INPUT );  
    }  
  
    attachInterrupt( 0, KeypadControl1, RISING );  
    attachInterrupt( 1, KeypadControl2, RISING );  
    attachInterrupt( 2, KeypadControl3, RISING );  
    attachInterrupt( 3, KeypadControl4, RISING );  
  
    vSemaphoreCreateBinary( Sem1 );  
  
    xTaskCreate( KeypadTask, NULL, 200, NULL, 2, NULL );  
    xTaskCreate( FndTask, NULL, 200, NULL, 1, NULL );  
    vTaskStartScheduler();  
}  
  
void loop() {  
  
}
```

# 실습 예제

- ❖ Keypad 버튼 4개(PB1, PB2, PB3, PB4)를 누르면 FND에 왼쪽으로 shift되며 표시



# 실습 과제 1)



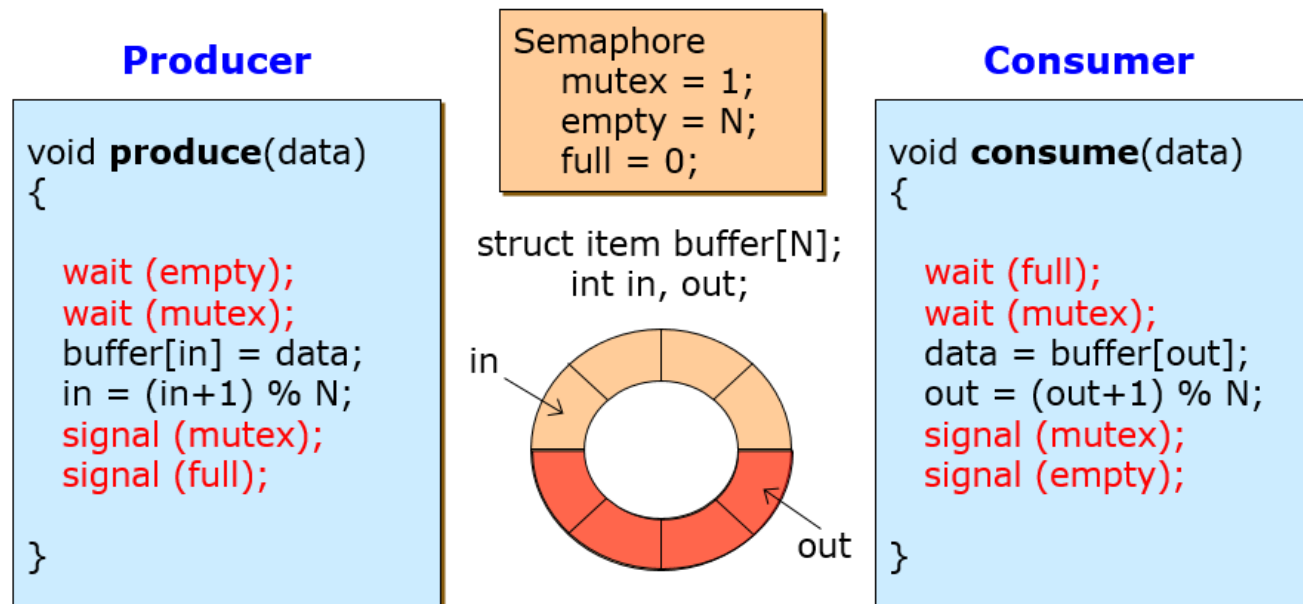
## ❖ Keypad를 통해 Motor 제어

- 왼쪽 회전, 정지, 오른쪽 회전 3개 키에 의해 모터 동작
- Keypad ISR과 Motor Task간 전역변수와 semaphore를 이용해 데이터 전달
- Lab. 3-3) 실습 예제 참조

## 실습 과제 2)

### ❖ Keypad를 통해 Motor 제어

- 실습과제 1)을 다음과 같이 확장
- Keypad ISR과 Keypad Task간 전역변수와 semaphore를 이용해 데이터 전달
- Keypad Task에서는 Circular Queue에 버튼을 입력
  - 이때 debounce 해결 (같은 값이면 무시)
- Motor Task에서는 Circular Queue로부터 버튼을 입력받아 모터 제어
- Keypad Task(producer)와 Motor Task(consumer)간 critical section 보호 및 동기화를 위해 semaphore를 사용
- 운영체제 Producer & Consumer의 bounded buffer problem 참조





# Q & A



<http://mesl.khu.ac.kr>