

```
##### 1번 보드 #####
#include <SPI.h>

void setup() {
  SPI.begin();
  // SCK, MOSI, SS를 출력으로 설정하고
  // SS를 high로 초기화
  SPI.setClockDivider(SPI_CLOCK_DIV16);
  Serial.begin(9600);
}

byte x = 0;
byte rx1, rx2;
char buf[80];

void loop() {
  digitalWrite(SS, LOW); // SPI 시작
  rx1 = SPI.transfer(x); // send one byte, get one byte
  x++;
  rx2 = SPI.transfer(x); // send one byte, get one byte
  x++;
  digitalWrite(SS, HIGH); // SPI 종료
  sprintf(buf, "Master x=%d, rx1=%d, rx2=%d", x, rx1, rx2);
  Serial.println(buf);
  delay(1000);
}

##### 2번 보드 #####

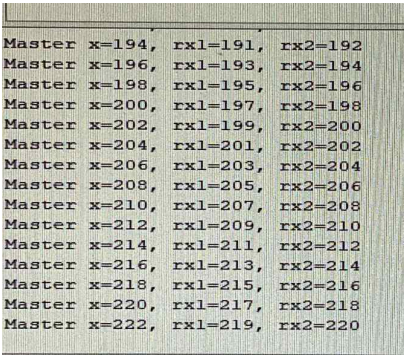
#include <SPI.h>

byte queue[100]; // receive queue
byte head, tail; // queue pointers
byte rxdata;
byte rxcount = 0;
char buf[80]; // print data to serial mointer

void setup() {
  pinMode(SCK, INPUT); // 받는 입장이라 입력모드
  pinMode(MOSI, INPUT); // 받는 입장이라 입력모드
  pinMode(MISO, OUTPUT); // 양방향 소통이므로 출력모드
  pinMode(SS, INPUT); // 받는 입장이라 입력모드

  SPI.setClockDivider(SPI_CLOCK_DIV16);
  SPCR &= ~(1 << MSTR); // Slave
  SPCR |= (1 << SPIE); // Enable SPI Interrupt
  SPCR |= (1 << SPE);
  Serial.begin(9600);
  head = 0;
  tail = 0;
}

ISR(SPI_STC_vect) { // SPI Interrupt
  rxdata = SPDR; // rx data
  SPDR = rxdata; // echo back
  queue[head] = rxdata;
  head = (head + 1) % 100;
  rxcount += 1;
}
```



<pre>void loop() { int rxdata; if (head != tail) { rxdata = queue[tail]; tail = (tail + 1) % 100; sprintf(buf, "SPI RX count = %d, data = %d", rxcount, rxdata); Serial.println(buf); } }</pre>	
<pre>##### 1번 보드 ##### // 마스터 Master #include <Wire.h> void setup(){ Wire.begin(); // I2C 버스에 연결하고 초기화 } byte x = 0; void loop(){ Wire.beginTransmission(4); // 4번 슬레이브로 주소 전송 시작 Wire.write("x is "); // 명령에 따라 전달할 데이터를 클럭에 저장 Wire.write(x); // 명령에 따라 전달할 데이터를 클럭에 저장 Wire.endTransmission(); // 통신의 세션을 종료 x++; delay(500); } ##### 2번 보드 ##### // 슬레이브 #include <Wire.h> void setup(){ Wire.begin(4); // 출력기기에서 4번 기기에 연결했으므로 // 마스터에서 송신한 데이터를 받을 때 선언 Wire.onReceive(receiveEvent); Serial.begin(9600); } void loop() { delay(100); } void receiveEvent(int howMany) { // 슬레이브 기기에서 Wire.onReceive 함수 안에서 선언 while(1 < Wire.available()) { // 전달받은 값(char형)을 저장한다. char c = Wire.read(); Serial.print(c); } // 전달받은 값(int형)을 저장한다. int x = Wire.read(); Serial.println(x); } }</pre>	<pre>## 중요 Wire.onReceive(함수이름);</pre>
#include <Wire.h>	LiquidCrystal_I2C : 클래스 객체 생성

```
#include <LiquidCrystal_I2C.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);

void setup() {
  lcd.init(); // Initialize I2C LCD module
  lcd.backlight(); // Turn backlight ON
  lcd.setCursor(0, 0); // Go to column 0, row 0
  lcd.print("Hello, world!");
  lcd.setCursor(0, 1); // Go to column 0, row 1
  lcd.print("Arduino I2C LCD");
}

void loop() {

  lcd.init();
  lcd.clear();
  //Backlight on/off
  lcd.backlight();
  lcd.print("BACKLIGHT ON");
  lcd.setCursor(0,1);
  lcd.print("AFTER 5SEC OFF");
  delay(5000);
  lcd.noBacklight();
  delay(3000);
  lcd.clear();

  //display on/off
  lcd.backlight();
  lcd.display();
  lcd.print("DISPLAY ON!");
  lcd.setCursor(0,1);
  lcd.print("AFTER 5SEC OFF");
  delay(5000);
  lcd.noDisplay();
  delay(3000);
  lcd.display();
  lcd.clear();

  //cursor
  lcd.print("CURSOR APPEAR");
  lcd.setCursor(0,1);
  lcd.cursor();
  delay(5000);
  lcd.clear();

  //corsor move
  lcd.print("CURSOR MOVE");
  for (int i=0; i < 10 ; i++) {
    lcd.setCursor(i,1);
    delay(500);
  }
  lcd.clear();

  //blink cursor
  lcd.print("CURSOR BLINK");
  lcd.setCursor(5,1);
  lcd.blink();
  delay(5000);
  lcd.noCursor();
```

```
name.begin() : 설정 초기화
clear() : 화면 초기화
backlight() : 백라이트 켜기
noblacklight() : 백라이트 끄기
cursor() : 커서 밑줄 깜빡임 켜기
noCursor() : 커서 밑줄 깜빡임 끄기
blink() : 커서 위치 깜빡임 켜기
noBlink() : 커서 위치 깜빡임 끄기
autoscroll() : 오토스크롤 켜기
noAutoscroll() : 오토스크롤 끄기
setCursor(column, row)
                : 커서 위치 이동
print("String") : 텍스트 쓰기
```

<pre>//write function lcd.print("ASCII CODE WRITE"); lcd.setCursor(0,1); lcd.write(36); lcd.write(37); lcd.write(38); delay(5000); lcd.clear(); //scroll right, left lcd.print("SCROLL RIGHT"); for (int positionCounter = 0; positionCounter < 16; positionCounter++) { lcd.scrollDisplayRight(); delay(500); } lcd.clear(); lcd.print(" SCROLL LEFT"); for (int positionCounter = 0; positionCounter < 16; positionCounter++) { lcd.scrollDisplayLeft(); delay(500); } lcd.clear(); //autoscroll lcd.print(" AUTO SCROLL"); lcd.setCursor(15,1); lcd.autoscroll(); for (int i=0; i < 10 ; i++) { lcd.print(i); delay(500); } lcd.clear(); lcd.init(); // put your main code here, to run repeatedly: }</pre>	