과제명 : 미로 탈출 게임기

|  |  |  |
| --- | --- | --- |
| **연구노트 1**   1. 제목 : ATmega4809 내부 레지스터 접근을 통한 UART 통신 실습 2. 목적 : 미로 탈출 게임기에서 각 부품과 데이터를 송수신 할 때 사용할 UART 통신에 대해 사용법을 익힌다. 3. 내용 : UART1을 9600/n/8/1로 초기화하고 Arduino IDE와 연결하여 문자를 송수신한다. 4. 결과 :           회로는 실습 3의 Blink 예제와 같이 구성하고 RX LED를 추가했다. 코드는 먼저, 각 레지스터 주소를 선언하고 Baud Rate를 9600으로 설정해주었다. CTRL C 레지스터에서 UART를 초기화하는데, 기본 값이 n/8/1이므로 따로 값을 지정하지는 않았다. CTRL B 레지스터로 RX와 TX를 enable 설정해주고 입출력 핀도 DIRC 레지스터를 통해 설정해주었다. 데이터를 송수신하는 함수 RX, TX를 선언하고 main 함수의 while 반복문에서 TX 함수를 통해 문자 ‘a’를 송신하였다. 시리얼 모니터로 확인해보니 설정한 만큼 딜레이가 생기며 a가 출력되었다.   1. 토의 : Baud rate를 9600으로 설정하기 위해서 ATmega4809 데이터시트에 나온 것 처럼 ((F\_CPU \* 64 / (16 \* BAUD\_RATE)) + 0.5)와 같이 선언해주었지만 시리얼 모니터에는 값이 제대로 출력되지 않았다. 그래서 공식의 계수를 여러 값으로 바꾸며 제대로 출력되는 식을 찾아보니 ((F\_CPU \* 20 / (30 \* BAUD\_RATE)) + 0.5) 일 때 제대로 a가 모니터에 나타나는 것을 알 수 있었다. | | |
| 기록자 이민정 | 점검자 | 점검자 |
| 일자 2020/05/25 | 일자 | 일자 |

|  |  |  |
| --- | --- | --- |
| **연구노트 2**   1. 제목 : 미로 탈출 게임기 부품 테스트와 회로 구성 2. 목적 : 미로 탈출 게임기에서 사용할 부품(조이스틱, OLED)의 사용법을 익히고 회로를 완성한다. 3. 내용 :    1. ATmega4809에 조이스틱을 연결하고 ADC를 통해 조이스틱의 X축, Y축 값을 읽어 UART로 전송한다.    2. ATmega4809에 OLED를 연결하고 OLED 화면에 글자를 출력한다. 4. 결과 :    1. ATmega4809와 조이스틱 모듈     조이스틱 모듈의 GND와 +5V 핀은 ATmega4809의 GND와 VCC에 연결하고 SW 핀은 PC5에 , X와 Y는 각각 PD0과 PD1에 연결해 주었다.        ADC 관련 레지스터 주소를 선언해주고 adc\_init 함수로 CTRL A와 B 레지스터를 통해 ADC 초기화를 해주었다. get\_value 함수를 통해 변환된 값이 저장된 레지스터 RES를 return하여 main 함수의 while 반복문에서 MUXPOS 레지스터를 0x00으로 설정하여 먼저 X축의 값을 받고 x\_i\_to\_str함수로 변환된 값을 편하게 볼 수 있게 4자씩 잘라서 출력시키도록 변환해주었다. 같은 과정을 Y축 값으로도 반복하여 X, Y의 순서로 계속 값을 변환시키도록 작성하였다. 변환된 값은 TX\_str함수를 통해 UART1으로 전송되어 조이스틱을 상하좌우로 움직일 때 모두 값이 제대로 전송된 것을 시리얼 모니터를 통해 확인하였다.   * 1. ATmega4809와 OLED 모듈     OLED 모듈의 RES 핀은 PA4, SDA 핀은 ATmega4809의 SDA 핀인 PA2, SCL 핀 ATmega4809의 SCL 핀인 PA3에 연결하고 GND와 VCC 핀도 연결해주었다.    디스플레이 테스트를 위해 U8g2 라이브러리를 다운받아 예제를 실행해보았다. 다양한 드라이버와 통신 방식 중 SSD1306, I2C를 선택하여 주석을 해제하고 PA3, PA2, PA4에 해당하는 아두이노 핀인 3, 2, 4를 추가하여 핀 설정을 완료하였다. u8g2.print 함수로 문자열을 출력하였으며 회로 사진과 같이 글자가 제대로 출력되었다.   1. 토의 :    1. ATmega4809와 조이스틱 모듈   시리얼 모니터를 통해 확인한 값이 계속 물음표로만 보여 해결 방법을 찾아보다int형으로 받은 RES 값을 string으로 변환하는 함수를 추가하였더니 값이 제대로 출력되었다.   * 1. ATmega4809와 OLED 모듈   구매한 OLED 모듈이 SSD1309 드라이버를 지원하는 제품이기 때문에 U8glib 라이브러리를 다운받아 시도해보았지만 ATmega4809와 호환되지 않았고 SSD1309는 SSD1306 드라이버와 호환되므로 Adafruit SSD1306 라이브러리로 시도해보았지만 OLED에 값이 출력되지 않았다. 마지막으로 U8g2 라이브러리의 SSD1306드라이버를 사용해보았더니 값이 화면에 출력되어 U8g2 라이브러리를 사용하기로 하였다. | | |
| 기록자 이민정 | 점검자 | 점검자 |
| 일자 2020/06/01 | 일자 | 일자 |

|  |  |  |
| --- | --- | --- |
| **연구노트 3**   1. 제목 : ATmega4809 내부 레지스터 접근을 통한 TWI 통신 실습 2. 목적 : OLED 디스플레이에 그래픽을 출력할 때 사용할 TWI 통신의 사용법을 익힌다. 3. 내용 : SSD1306 드라이버를 통해 ATmega4809에서 값을 받아와 OLED로 출력한다. 4. 결과 :           위의 표 3개와 OLED 부품 판매 사이트에 기재된 내용을 참고하여 SSD1306드라이버를 I2C 통신으로 연결하기 위한 최소한의 코드를 작성하였다. (코드가 매우 길어 주소를 define한 부분만 캡쳐 하였습니다.) I2C Address(0x3C)를 송신한 뒤에 쓰여질 데이터가 Command인지 GDDRAM에 쓰여질 데이터인지 구별하기 위하여 0x00과 0x40으로 구분하였고 ENTIRE\_ON, DISPLAY\_ON 주소를 설정하여 위의 코드 대로라면, Display가 모두 켜져야 한다.      위와 같이 ATmega4809의 TWI와 OLED 내의 SSD1306을 모두 설정 하였지만 OLED 디스플레이는 켜지지 않았다.   1. 토의 : ATmega4809와 SSD1306 데이터시트를 읽어보며 설정을 여러 방법으로 수정해보았지만, 문제는 파악하지 못했다. UART 통신 실습에서 BAUD 레지스터 설정이 문제가 있었기 때문에 TWI의 MBAUD에 문제가 있을 가능성이 높다고 생각하였다. 따라서 Fclk\_per과 Fscl에 각각 16MHz, 100000 대신 넓은 범위의 다른 값을 넣어보았지만 화면은 켜지지 않았다. 다시 시도해볼 때에는, 정확히 어느 부분이 잘못되었는지 파악할 수 있도록 실습실에서 여러 값을 확인해봐야겠다. | | |
| 기록자 이민정 | 점검자 | 점검자 |
| 일자 2020/06/08 | 일자 | 일자 |