|  |  |  |  |
| --- | --- | --- | --- |
| **4주차 결과보고서** | | | |
| **과목명** | 전자회로실험2 | **분반** | 005 |
| **학과** | 전자공학과 | **이름** | **이진호** |
| **제목** | ATmega128 USART LCD | | |

**1. 코드 작성**

(1)

|  |
| --- |
| 과제 1  void USART\_clear(){      USART\_Transmit("$C"); //lcd의 화면을 클리어      USART\_Transmit\_String("$G,1,1\r");  //lcd의 커서를 1,1로 지정      USART\_Transmit\_String("$T,");   //문자열 출력  } |
| 과제 2  #define F\_CPU 16000000UL //16MHz 설정  #define BAUD 9600 //9600bfs 설정  #define MYUBRR F\_CPU/16/BAUD-1 //UBRR 계산  #include <util/delay.h>  #include <avr/io.h>  // USART 초기화 함수  void USART\_Init(unsigned int ubrr){      UBRR0H = (unsigned char)(ubrr >> 8); //0으로 초기화      UBRR0L = (unsigned char)ubrr; //그대로 출력      UCSR0B =(1<<RXEN0) | (1<<TXEN0); //송수신 가능      //0000 0000 0000 0011 -> 0000 0000 0000 1100 char size 12로 설정      UCSR0C = (3<< UCSZ0);    }  // USART 데이터 전송 함수  void USART\_Transmit(char data){      while(!((UCSR0A)&(1<<UDRE0))); //입력이 안들어오면 반복      UDR0 = data; //UDR0에 데이터 저장  }  // USART 데이터 수신 함수  char USART\_Receive(){      while(!(UCSR0A &(1<<RXC0))); //입력이 안들어오면 반복      return UDR0; //UDR0 반환  }  //LCD에 char 문자 출력 하는 함수  void LCD\_Transmit(int x, int y, char data) {      USART\_Transmit\_String("$G,"); //커서 지정      USART\_Transmit((char)(x+48)); //x 번째 줄      USART\_Transmit(',');      USART\_Transmit((char)(y+48)); //y 번째 칸      USART\_Transmit('\r'); //종료      USART\_Transmit\_String("$T,"); //문자열 출력      USART\_Transmit(data); //data 문자 출력    }  //LCD에 String 문자열 출력 하는 함수  void LCD\_Transmit\_String(int x, int y, char\* String){      USART\_Transmit\_String("$G,"); //커서 지정      USART\_Transmit((char)(x+48)); //x 번째 줄      USART\_Transmit(',');      USART\_Transmit((char)(y+48)); //y 번째 칸      USART\_Transmit('\r'); //종료      USART\_Transmit\_String("$T,"); //문자열 출력      while(\*String) //문자열의 주소값이 null이 아닐때 까지      {          USART\_Transmit(\*String++); //char 문자 출력하는 함수 호출      }  }  int main(void)  {        int x=1;      int y=2;      int a=2;      int b=2;      char data = 'a';      char string[] = "hello world";      USART\_Init(MYUBRR); //초기화      USART\_Transmit\_String("$I\r"); //LCD 초기화      LCD\_Transmit(x,y,data); // x=1,y=2 인 위치에 char 문자 출력      USART\_Transmit('\r'); //종료      LCD\_Transmit\_String(a,b,string); //a=2, b=2인 위치에 string 문자열 출력      USART\_Transmit('\r'); //종료      /\* Replace with your application code \*/      while (1)      {          \_delay\_ms(100); //딜레이 0.1초      }  } |
| 과제 3  #define F\_CPU 16000000UL //16MHz 설정  #define BAUD 9600 //9600bfs 설정  #define MYUBRR F\_CPU/16/BAUD-1 //UBRR 계산  #include <util/delay.h>  #include <avr/io.h>  #include <string.h> //문자열 라이브러리 추가  // USART 초기화 함수  void USART\_Init(unsigned int ubrr){      UBRR0H = (unsigned char)(ubrr >> 8); //0으로 초기화      UBRR0L = (unsigned char)ubrr; //그대로 출력      UCSR0B =(1<<RXEN0) | (1<<TXEN0); //송수신 가능      //0000 0000 0000 0011 -> 0000 0000 0000 1100 char size 12로 설정      UCSR0C = (3<< UCSZ0);    }  // USART 데이터 전송 함수  void USART\_Transmit(char data){      while(!((UCSR0A)&(1<<UDRE0))); //입력이 안들어오면 반복      UDR0 = data; //UDR0에 데이터 저장  }  // USART 데이터 수신 함수  char USART\_Receive(){      while(!(UCSR0A &(1<<RXC0))); //입력이 안들어오면 반복      return UDR0; //UDR0 반환  }  //LCD에 char 문자 출력 하는 함수  void LCD\_Transmit(int x, int y, char data) {      USART\_Transmit\_String("$G,"); //커서 지정      USART\_Transmit((char)(x+48)); //x 번째 줄      USART\_Transmit(',');      USART\_Transmit((char)(y+48)); //y 번째 칸      USART\_Transmit('\r'); //종료      USART\_Transmit\_String("$T,"); //문자열 출력      USART\_Transmit(data); //data 문자 출력    }  //LCD에 String 문자열 출력 하는 함수  void LCD\_Transmit\_String(int x, int y, char\* String){      USART\_Transmit\_String("$G,"); //커서 지정      USART\_Transmit((char)(x+48)); //x 번째 줄      USART\_Transmit(',');      USART\_Transmit((char)(y+48)); //y 번째 칸      USART\_Transmit('\r'); //종료      USART\_Transmit\_String("$T,"); //문자열 출력      while(\*String) //문자열의 주소값이 null이 아닐때 까지      {          USART\_Transmit(\*String++); //char 문자 출력하는 함수 호출      }  }  int main(void)  {      USART\_Init(MYUBRR); //초기화      USART\_Transmit\_String("$I\r"); //LCD 초기화        char string[] = "Leejinho. Department of Electronic Engineering"; //문자열 저장      int size = strlen(string); //문자열 길이 저장      \_delay\_ms(100); //딜레이 0.1초      while (1)      {          //문자열의 길이 만큼 반복          for(int i=0;i<size;i++){              USART\_Transmit\_String("$C\r"); //LCD 화면 클리어              LCD\_Transmit\_String(1,1,string+i); // 1,1의 위치에 문자열+i의 주소로 시작하는 문자열 출력              USART\_Transmit('\r'); //종료              \_delay\_ms(500); //딜레이 0.5초          }          \_delay\_ms(100); //딜레이 0.1초      }  } |

**2. 고찰**

이번 실험은 아트메가128을 이용하여 LCD에 출력하는 실험을 하였다.실습에 앞서 LCD에 hello를 출력하는 방법을 배웠고, 그뒤 총 3가지의 실험을 진행하였다. 첫번째는 LCD를 초기화 하는 함수를 작성하는 것이었다. 이는 간단하게 AM-SLCD 커맨드 표의 화면 클리어 커맨드를 이용하여 LCD를 클리어 하는 명령어와, 커서를 1,1에 두는 명령어를 이용하여 함수를 작성하였다. 두번째는 x,y의 위치에 문자 출력하는 함수, x,y의 위치에 문자열을 출력하는 함수를 작성하는 것이었는데, x,y가 명령어 G에 바로 적용이 되지 않기 때문에, /r을 이용해 명령어가 끝난다는 것을 이용하여, 함수로 부터 전달 받은 x,y의 값을 아스키코드로 변환하여 문자로 만들었고, 이를 G명령어에 함께 사용하여 만들었다. 문자열 또한 마찬가지 방법으로 구현하였다. 마지막으로는 LCD에 긴 문자열을 출력하고, 글자가 한글자씩 흘러가도록 출력하는 프로그램을 만드는 것이었는데, 이는 문자 배열의 크기를 가져와, 문자 배열의 주소값을 1씩 증가시켜 1,1의 위치에 출력하도록 구현하였다. 마지막 3번째가 구현하기 까다로웠지만, 포인터개념을 이해하면 쉽게 구현이 가능할 수 있다는 것을 알게 되었다.