**전자회로실험 보고서**



학과 전자공학과

학번 1218165

이름 김재욱

제출일 2020-10-08

**실습 이론**

1. LCD 컨트롤러 기능

IR (명령 레지스터)

LCD 화면 초기화 ,커서 시프트, 문자표시 등등 LCD제어에 필요항 여러 명령을 저장한다.

표시데이터 RAM인 DDRAM의 위치 주소와 문자발생기 CGRAM의 위치를 설정하기 위한 주소를 저장한다.

DR (데이터 레지스터)

해당 레지스터에 데이터를 쓰면 LCD의 동작에의해 IR에 지정된 DDRAM이나CGRAM의 주소로 데이터를 전달한다.

해당 레지스터에서 데이터를 읽으면 IR이 설정한 DDRAM 이나 CGRAM의 주소 데이터가 마이크로컨트롤러로 전달된다.

DDRAM

80\*8 비트 용량으로 80개의 8비트 아스키코드를 저장 할 수 있다.

0x00-0F 주 소가 LCD 1행의 1-16번째, 0x40-4F 주소가 LCD 2행의 1-16번째 문자로 표시된다.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0x00 | 0x01 | 0x02 | 0x03 | 0x04 | 0x05 | 0x06 | 0x07 | 0x08 | 0x09 | 0x0A | 0x0B | 0x0C | 0x0D | 0x0E | 0x0F |
| 0x40 | 0x41 | 0x42 | 0x43 | 0x44 | 0x45 | 0x46 | 0x47 | 0x48 | 0x49 | 0x4A | 0x4B | 0x4C | 0x4D | 0x4E | 0x4F |

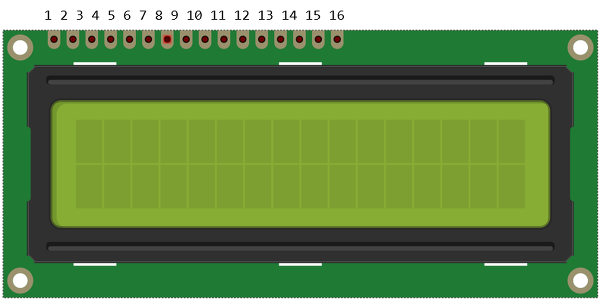
CGRAM

5\*7 5\*10도트 문자를 내장하고 있으며 문자들은 아스키코드와 일치한다.

CGROM

사용자가 원하는 문자를 만들때 사용한다.

2. LCD 핀의 기능





▶  RS(Register Select)

  입력 신호를 받는 핀이며 액정 표시 모듈에 데이터 혹은 명령을 write/read할 때 사용한다

▶ R /  / W(Read/Write)

  입력 신호로서 액정 표시 모듈에 데이터 혹은 명령을 write/read할 때 사용한다

▶ E1(Enable)

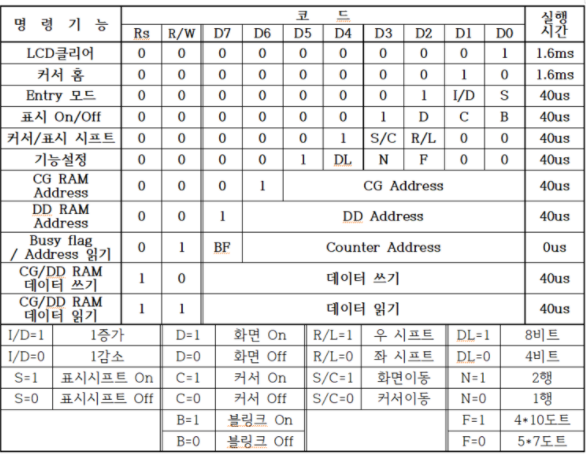
  이 신호는 Active High이다. 이것은 GAL 16V8의 입력 어드레스 라인을 조합하여 생성해낸 출력 어드레스 라인을 이용하여 LCD 인에이블 신호를 선택한다

▶ D0~D7(Data Bus)

  CPU와 LCD 사이에 데이터를 주고받기 위한 데이터 버스이다. 만약 4비트를 사용할 경우에는 D4~D7만 사용한다(D0~D3 사용 불가)

3. 초기 세팅

LCD에 데이터를 전송하여 LCD에 띄우기전 초기 세팅이 필요하다



위의 표에 명령기능 에 기능설정부터 위로 올라가는 순서대로 초기세팅이 이루어져야 한다.

실습 1

**실습내용 및 결과**

|  |
| --- |
|  |
| /\*  \* week3-1.c  \*  \* Created: 2020-10-02 오후 8:48:17  \* Author : lobgd  \*/  #define F\_CPU 16000000  #include <avr/io.h>  #include <util/delay.h>  void cmd(int command){  PORTE = command;  PORTB = 0x04; // 데이터 통신 enable, lcd 위치 설정 명령  \_delay\_ms(1);  PORTB = 0x00; // 데이터 통신 disable  }  void data(char str){  PORTE = str;  PORTB = 0x05; // 데이터 통신 enable, 레지스터에 저장해놨던 위치에 값을 쓰는 명령  \_delay\_ms(1);  PORTB = 0x01; // 데이터 통신 disable  }  void display1(char\*str){  cmd(0x80); // lcd 첫 위치 (0,0)  int i = 0;  while (str[i] != '\0' )  {  data(str[i++]);  \_delay\_ms(1000);  }  }  volatile char str[1][16] = {{"ABCDEFGHIJKLMNOPQRSTUVWXYZ"}}; // Initial LCD display  int main(){  DDRB=0xff;  DDRE=0xff;  cmd(0x38); // Set 8bit 2Line 5x7 dots  cmd(0x01); // Display clear  cmd(0x80); // Set DDRAM address or cursor position on display  cmd(0x0c); // Display on Cursor Off  cmd(0x06); // Entry Mode  display1(str[0]);  while(1){  }  } |
| 실습 1 결과 사진 |

실습 2

|  |
| --- |
|  |
| /\*  \* week3-2.c  \*  \* Created: 2020-10-02 오후 9:09:35  \* Author : lobgd  \*/  #define F\_CPU 16000000  #include <avr/io.h>  #include <util/delay.h>  volatile int count =0;  volatile char str[1][30] = {{"ABCDEFGHIJKLMNOPQRSTUVWXYZ"}}; // Initial LCD display  void cmd(int command){  PORTE = command;  PORTB = 0x04; // 데이터 통신 enable, lcd 위치 설정 명령  \_delay\_ms(1);  PORTB = 0x00; // 데이터 통신 disable  }  void data(char str){  PORTE = str;  PORTB = 0x05; // 데이터 통신 enable, 레지스터에 저장해놨던 위치에 값을 쓰는 명령  \_delay\_ms(1);  PORTB = 0x01; // 데이터 통신 disable  }  void display1(char\*str){  cmd(0x80); // lcd 첫 위치 (0,0)  int i = 0;  while (1)  {  if(count==16){ //lcd 1 line에 문자가 꽉차면 line의 처음으로 돌아감  cmd(0x80);  count=0;  }  if(i==26){ //모든 알파벳이 출력되면 다시 처음부터 출력  i=0;  }  data(str[i++]);  \_delay\_ms(1000);  count++;  }  }  int main(){  DDRB=0xff;  DDRE=0xff;  cmd(0x38); // Set 8bit 2Line 5x7 dots  cmd(0x01); // Display clear  cmd(0x80); // Set DDRAM address or cursor position on display  cmd(0x0c); // Display on Cursor Off  cmd(0x06); // Entry Mode  display1(str[0]);  while(1){  }  } |
| 실습 2 결과 동영상 첨부 |

실습 3

|  |
| --- |
|  |
| /\*  \* week3-3.c  \*  \* Created: 2020-10-02 오후 10:03:16  \* Author : lobgd  \*/  #define F\_CPU 16000000  #include <avr/io.h>  #include <util/delay.h>  volatile int count =1;  volatile int row =0;  volatile int column =0;  volatile char str[10][40] = {{"Hels forgotten all about the people in c"},  {"locks until he passed a group of them ne"},{"xt to the bakerls. He eyed them angrily "},  {"as he passed. He didlnt know why. but th"},{"Hels forgotten all about the people in c"},  {"locks until he passed a group of them ne"},{"xt to the bakerls. He eyed them angrily "},  {"as he passed. He didlnt know why. but th"},{"Hels forgotten all about the people in c"},  {"locks until he passed a group of them ne"}}; // Initial LCD display  void cmd(int command){  PORTE = command;  PORTB = 0x04; // 데이터 통신 enable, lcd 위치 설정 명령  \_delay\_ms(1);  PORTB = 0x00; // 데이터 통신 disable  }  void data(char str){  PORTE = str;  PORTB = 0x05; // 데이터 통신 enable, 레지스터에 저장해놨던 위치에 값을 쓰는 명령  \_delay\_ms(1);  PORTB = 0x01; // 데이터 통신 disable  }  void display1(char\*str){  cmd(0x80); // lcd 첫 위치 (0,0)  int i = column;  while (1)  {  data(str[i++]);  count++;  if((count%17)==0){ //lcd 1 line에 문자가 꽉차면 line의 처음으로 돌아감  count=0;  break;  }  }  }  void display2(char\*str){  cmd(0x80 | 0x40); //두번째줄 시작  int i=column;  while (1)  {  data(str[i++]);  count++;  if((count%17)==0){ //lcd 1 line에 문자가 꽉차면 line의 처음으로 돌아감  count=0;  break;  }  }  }  int main(){  unsigned char up;  unsigned char down;  unsigned char right;  unsigned char left;  DDRB=0xff;  DDRE=0xff;  DDRD=0b00000000; //하우좌상(4321)  PORTD=0x00; //풀업저항 설정    cmd(0x38); // Set 8bit 2Line 5x7 dots  cmd(0x01); // Display clear  cmd(0x80); // Set DDRAM address or cursor position on display  cmd(0x0c); // Display on Cursor Off  cmd(0x06); // Entry Mode  while(1){  up =(PIND&0b0000001);  left =(PIND&0b00000010);  right =(PIND&0b00000100);  down =(PIND&0b00001000);  if((up==0b00000001) && (row>0)){  row-=1;  }  if((down==0b00001000) && (row<10)){  row+=1;  }  if((left==0b00000010) && (column>0)){  column-=1;  }  if((right==0b00000100) && (column<31)){  column+=1;  }  display1(str[row]);  display2(str[row+1]);  \_delay\_ms(1000);  }  } |
| 실습 3 결과동영상 첨부  '가 출력되지 않아 ' -> l로 텍스트를 변환하였다.    위의 텍스트 내용을 스위치로 옮겨다니면서 2\*16(row\*col) 범위로 출력한다.   |  |  |  | | --- | --- | --- | | 첫 좌표 위치 | 텍스트 | 실제 결과 | | (0,0) |  |  | | (0,10) |  |  | | (2,4) |  |  | |

**고찰**

이번 실습에서 atmega에 LCD를 연결해 데이터를 LCD 화면에 출력하는 것을 배웠다.

LCD에서 데이터를 담고 이를 출력 하기 까지의 과정에서 2개의 레지스터와 3개의 메모리가 존재한다는 사실을 알게되었다 LCD를 세팅하고 데이터를 주고받는 등의 통신을 시작하거나 끝마칠때 인에이블에 신호를 인해줘야 한다.

LCD의 좌표를 설정할때는 DDRAM Address에서 D7에 1이 세팅 되어 있으므로 추가 연산이 필요하다. 즉 위의 표에서 각 주소마다 | 0x80 연산을 해주어야 해당 좌표로 접근이 가능하다.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0x00 | 0x01 | 0x02 | 0x03 | 0x04 | 0x05 | 0x06 | 0x07 | 0x08 | 0x09 | 0x0A | 0x0B | 0x0C | 0x0D | 0x0E | 0x0F |
| 0x40 | 0x41 | 0x42 | 0x43 | 0x44 | 0x45 | 0x46 | 0x47 | 0x48 | 0x49 | 0x4A | 0x4B | 0x4C | 0x4D | 0x4E | 0x4F |