P5 CLCD 문자 출력 결과보고서

3조 20162581 손승표

1. 소스코드를 완성하고, 아래의 질문에 답해보도록 하자

main.c

checker 함수

```
error_t checker(int argc, char* argv[]) {
 int S=0;
 if( argc <= 2 ) {
  return ERROR;
 }
 else if( argc >= 3 ) {
  if( strlen(argv[1]) > 1 ) { return ERROR; }
  if( argv[1][0] == '0') {
   clcd_entry_mode_set(0, S);
   clcd_set_DDRAM( strlen(argv[2]) - 1 );
   clcd_write_string(argv[2]);

   if( argc > 3 ) {
    clcd_set_DDRAM( 0x40 + strlen(argv[3]) - 1 );
    clcd_write_string(argv[3]);
   }
  }
  else if( argv[1][0] == '1' ) {
   clcd_entry_mode_set(1, S);

   clcd_set_DDRAM( strlen(argv[2]) - 1 );
   clcd_write_string(argv[2]);

   if( argc > 3 ) {
    clcd_set_DDRAM( 0x40 + strlen(argv[3]) - 1 );
    clcd_write_string(argv[3]);
   }
  }
  else { return ERROR; }
 }
 return SUCCESS;
}
```

inputter 함수

```
truth_t inputter() {
 int input_int;
 printf("\n");
 printf("*****************************************\n");
 printf("*     Please type a number              *\n");
 printf("*     1 : Display shift (left)          *\n");
 printf("*     2 : Display shift (right)         *\n");
 printf("*     3 : Display On                    *\n");
 printf("*     4 : Display Off                   *\n");
 printf("*     5 : clcd_up_shift                *\n");
 printf("*     0 : EXIT                          *\n");
 printf("*****************************************\n\n");
 scanf("%d", &input_int);

 switch(input_int) {
  case 1 :
   clcd_shift(1,0);
   break;
  case 2 :
   clcd_shift(1,1);
   break;
  case 3 :
   clcd_display_control(1, 1, 1);
   break;
  case 4 :
   clcd_display_control(0, 0, 0);
   break;
  case 5 :
   clcd_up_shift() ;
   break;
  case 0 :
   return FALSE;
  default :
   break;
 }
 return TRUE;
}
```

clcd.c

```
static short * clcd_cmd, * clcd_data;

void init_clcd(short * address_cmd, short * address_data) {
 int D = 1, C = 0, B = 0;
 int DL = 1, N = 1, F = 0;

 clcd_cmd  = address_cmd;
 clcd_data = address_data;

 clcd_clear_display();
```

```
 clcd_return_home();
 clcd_display_control(D, C, B);
 clcd_function_set(DL, N, F);
}

void clcd_write_cmd(int cmd) {
 *clcd_cmd = (short)cmd; usleep(CLCD_CMD_US);
}
void clcd_write_data(int data) {
 *clcd_data = (short)data; usleep(CLCD_CMD_US);
}

void clcd_clear_display() {
 clcd_write_cmd(1); usleep(CLCD_RETURN_US);
}
void clcd_return_home() {
 clcd_write_cmd(1 << 1); usleep(CLCD_RETURN_US);
}
void clcd_entry_mode_set(int ID, int S) {
 int cmd = 1 << 2;
 if( ID != 0 ) { cmd |= (1 << 1); }
 if( S != 0 ) { cmd |= (1); }
 clcd_write_cmd(cmd); usleep(CLCD_RETURN_US);
}
void clcd_display_control(int D, int C, int B) {
 int cmd = 1 << 3;
 if( D != 0 ) { cmd |= (1 << 2); }
 if( C != 0 ) { cmd |= (1 << 1); }
 if( B  != 0 ) { cmd |= (1); }
 clcd_write_cmd(cmd); usleep(CLCD_RETURN_US);
}
void clcd_shift(int SC, int RL) {
 int cmd = 1 << 4;
 if( SC != 0 ) { cmd |= (1 << 3); }
 if( RL != 0 ) { cmd |= (1 << 2); }
 clcd_write_cmd(cmd); usleep(CLCD_RETURN_US);
}
void clcd_function_set(int DL, int N, int F) {
 int cmd = 1 << 5;
 if( DL != 0 ) { cmd |= (1 << 4); }
 if( N  != 0 ) { cmd |= (1 << 3); }
 if( F  != 0 ) { cmd |= (1 << 2); }
 clcd_write_cmd(cmd);
}
void clcd_set_DDRAM(int address) {
 int cmd = 1 << 7;
 cmd |= address;
 clcd_write_cmd(cmd);
}

void clcd_write_string(char str[]) {
 int i;
 for( i=0; (str[i] != '\0'); i++) {
  clcd_write_data(str[i]);
```

```
 }
}

void clcd_up_shift() {
 clcd_clear_display();
 int i;
 char num[3];
 num[2] = '\0';
 for(i = 0; i<100; i++){
  num[0] = (i / 10) + '0';
  num[1] = (i % 10)  + '0';
  clcd_return_home();
  clcd_set_DDRAM(0x0E);
  clcd_write_string(num);
  usleep(100000);
 }
}
```

a. "Hello"라는 한 개의 문자열을, 커서를 증가/감소 모드로 두고 입력했을 때, CLCD에는 각각 어떤 모습으로 나타나겠는가?

- 커서 1 : 증가모드
  - strlen("Hello") - 1 의 값이 4이기 때문에
  - 왼쪽에서 4칸 공백 이후 Hello 문자가 출력된다.
    - 왼쪽끝____Hello
- 커서 0 : 감소모드
  - strlen("Hello") - 1 의 값이 4이기 때문에
  - 오른쪽에서 4칸 공백 이후 olleh 문자가 출력된다.
    - olleH____오른쪽끝

b. 문자열에서 NUL값을 제외하기 위해 어떤 방법을 사용하였는가? 만약 정상적으로 동작하지 않는다면, 이유는 무엇이며 어떻게 고쳐야 하겠는가?

- argc의 값이 2이하의 case는 error를 출력하도록해서 문자열 null값을 제외했다.
  - 문자열이 들어오지 않는다면 argc 값이 2가 되고,
  - 문자열이 들어온다면 argc 값이 3이 되기 때문
- 그럼에도 불구하고 null값이 들어올 수 있기 때문에, 추후에 코드 상에서 null값인 경우 error를 출력하도록 만드는 것이 좋다.

c. CLCD의 Function Set 기능을 이용할 때, DL, N, F 값은 각각 얼마이며 그 이유는 무엇인가?

- DL : 1
  - 8bit 주소를 사용해야하기 때문에 1로 설정.
  - 0으로 설정시 4bit이용
  - 우리는 제어 명령을 8bit로 전송해야한다.
- N : 1
  - clcd 화면 2줄을 다 사용하기에 1로 설정.
  - 1줄만 이용 시 0으로 설정.
- F : 0

- 우리는 5x8 dots를 이용하기에 0으로 세팅.
- 5x10 dot를 이용하고 싶다면 1로 세팅.

2. 숫자 Counter

```
void clcd_up_shift() {
 clcd_clear_display();
 int i;
 char num[3];
 num[2] = '\0';
 for(i = 0; i<100; i++){
  num[0] = (i / 10) + '0';
  num[1] = (i % 10)  + '0';
  clcd_return_home();
  clcd_set_DDRAM(0x0E);
  clcd_write_string(num);
  usleep(100000);
 }
```

- 숫자를 0부터 99까지 카운트업 하는 함수입니다.
- 0.1초마다 카운트업 됩니다.

1. 먼저 clcd_clear_display()로 화면을 clear 합니다.
2. num[2]에 공백문자를 넣어, 문자열의 끝을 알립니다.
    - 두 자리의 숫자가 출력될 것이기에 3번째 칸에 공백문자를 삽입하는 것 입니다.
3. 숫자를 일의 자리 숫자, 십의 자리 숫자로 나누어서, 문자열에 각각 저장합니다.
4. clcd_reutrn_home()으로 처음 부분으로 커서를 옮깁니다.
5. 그 다음 끝 2자리에 문자를 출력할 수 있도록 커서를 옮깁니다.
    - 0x0E칸에 출력할 것입니다.
6. clcd_wirte_string으로 num을 출력합니다.
7. 다음 숫자 출력까지 0.1초 쉬어서 눈으로 카운트업을 잘 확인할 수 있도록 합니다.
8. 3번부터 다시 반복합니다.