Busan Software Meister High School

MICROPROCESSOR

2309 양유빈

20230518 마이크로프로세서

OVERVIEW

- LED 제어 프로그램(GPIO)
- LED 8개 연속 점등 프로그램
- 버튼으로 LED 제어
- LED 4개씩 버튼 2개로 제어하는 프로그램
- 전처리기
- 버튼으로 스위치 제어
- +과제: 밝기 조절이 가능한 랜턴 프로그램

LED 제어 프로그램(GPIO)

LED control program (GPIO)

```
86  /* Initialize all configured peripherals */
87  MX_GPIO_Init();
88  /* USER CODE BEGIN 2 */
89  typedef struct led {
90      GPIO_TypeDef * port;
91      uint16_t pin;
92  }LED;
93  LED led[8] = {
94          {GPIOC, GPIO_PIN_3}, {GPIOC, GPIO_PIN_2},
95          {GPIOC, GPIO_PIN_1}, {GPIOC, GPIO_PIN_0},
96          {GPIOB, GPIO_PIN_15}, {GPIOB, GPIO_PIN_14},
97          {GPIOB, GPIO_PIN_13}, {GPIOB, GPIO_PIN_12}
98  };
99
00  /* USER CODE END 2 */
01
02  /* Infinite loop */
03  /* USER CODE BEGIN WHILE */
04  while (1)
05  {
06      for(uint8_t i=0; i<8; i++) {
07          HAL_GPIO_WritePin(led[i].port, led[i].pin, GPIO_PIN_SET);
08          HAL_Delay(500);
09          HAL_GPIO_WritePin(led[i].port, led[i].pin, GPIO_PIN_RESET);
10          HAL_Delay(500);
11      }
12    /* USER CODE END WHILE */
```

LED 8개 연속 점등 프로그램

8 LED continuous lighting programs

```
87  MX_GPIO_Init();
88  /* USER CODE BEGIN 2 */
89  typedef struct led {
90      GPIO_TypeDef * port;
91      uint16_t pin;
92  }LED;
93  LED led[8] = {
94          {GPIOC, GPIO_PIN_3}, {GPIOC, GPIO_PIN_2},
95          {GPIOC, GPIO_PIN_1}, {GPIOC, GPIO_PIN_0},
96          {GPIOB, GPIO_PIN_15}, {GPIOB, GPIO_PIN_14},
97          {GPIOB, GPIO_PIN_13}, {GPIOB, GPIO_PIN_12}
98  };
99
00  /* USER CODE END 2 */
01
02  /* Infinite loop */
03  /* USER CODE BEGIN WHILE */
04  while (1)
05  {
06      for(uint8_t i=0; i<8; i++) {
07          HAL_GPIO_WritePin(led[i].port, led[i].pin, GPIO_PIN_SET);
08          HAL_Delay(500);
09          HAL_GPIO_WritePin(led[i].port, led[i].pin, GPIO_PIN_RESET);
10          HAL_Delay(100);
11      }
12      for(uint8_t i=6; i>0; i--) {
13          HAL_GPIO_WritePin(led[i].port, led[i].pin, GPIO_PIN_SET);
14          HAL_Delay(500);
15          HAL_GPIO_WritePin(led[i].port, led[i].pin, GPIO_PIN_RESET);
16          HAL_Delay(100);
17      }
```

버튼으로 LED 제어

LED control with buttons

```
94  while (1)
95  {
96      if(!HAL_GPIO_ReadPin(GPIOD, GPIO_PIN_2)){
97              HAL_GPIO_WritePin(GPIOC, GPIO_PIN_3,1);
98              HAL_GPIO_WritePin(GPIOB, GPIO_PIN_12,1);
99       }else{
00              HAL_GPIO_WritePin(GPIOC, GPIO_PIN_3,0);
01              HAL_GPIO_WritePin(GPIOB, GPIO_PIN_12,0);
02      }
03    /* USER CODE END WHILE */
04
```

LED 4개 각각 버튼 2개로 제어하는 프로그램

A program that controls 4 LEDs each with 2 buttons each

📌 핵심: 구조체 이용

```
86   /* Initialize all configured peripherals */
87   MX_GPIO_Init();
88   /* USER CODE BEGIN 2 */
89   typedef struct led {
90       GPIO_TypeDef * port;
91       uint16_t pin;
92    }LED;
93    LED led[8] = {
94           {GPIOC, GPIO_PIN_3}, {GPIOC, GPIO_PIN_2},
95           {GPIOC, GPIO_PIN_1}, {GPIOC, GPIO_PIN_0},
96           {GPIOB, GPIO_PIN_15}, {GPIOB, GPIO_PIN_14},
97           {GPIOB, GPIO_PIN_13}, {GPIOB, GPIO_PIN_12}
98    };
99
00
01   /* USER CODE END 2 */
```

```
105  while (1)
106  {
107      if(!HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_4)){
108          for(uint8_t i=0; i<4; i++)
109              HAL_GPIO_WritePin(led[i].port, led[i].pin, GPIO_PIN_SET);
110       } else{
111           for(uint8_t i=0; i<4; i++)
112               HAL_GPIO_WritePin(led[i].port, led[i].pin, GPIO_PIN_RESET);
113       }
114      if(!HAL_GPIO_ReadPin(GPIOD, GPIO_PIN_2)){
115          for(uint8_t i=0; i<4; i++)
116              HAL_GPIO_WritePin(led[i+4].port, led[i+4].pin, GPIO_PIN_SET);
117      } else{
118          for(uint8_t i=0; i<4; i++)
119              HAL_GPIO_WritePin(led[i+4].port, led[i+4].pin, GPIO_PIN_RESET);
120      }
121
122    /* USER CODE END WHILE */
```

전처리기

preprocessor

📌 if, endif

0 -> 주석처리

1 -> 실행

```
105 #if 0
106   while (1)
107   {
108       if(!HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_4)){
109           for(uint8_t i=0; i<4; i++)
110               HAL_GPIO_WritePin(led[i].port, led[i].pin, GPIO_PIN_SET);
111        } else{
112            for(uint8_t i=0; i<4; i++)
113                HAL_GPIO_WritePin(led[i].port, led[i].pin, GPIO_PIN_RESET);
114        }
115       if(!HAL_GPIO_ReadPin(GPIOD, GPIO_PIN_2)){
116           for(uint8_t i=0; i<4; i++)
117               HAL_GPIO_WritePin(led[i+4].port, led[i+4].pin, GPIO_PIN_SET);
118       } else{
119           for(uint8_t i=0; i<4; i++)
120               HAL_GPIO_WritePin(led[i+4].port, led[i+4].pin, GPIO_PIN_RESET);
121       }
122 #endif
123
124 #if 1
125
126 #endif
```

버튼으로 스위치 제어

switch control with button

📌 스위치1이 ON 인지 확인. 만약 스위치1이 ON이면 현재 LED의 상태를 확인 후 아래와 같이 조치.
[LED_ON 상태면 가운데 LED 2개를 모두 끄고, 현 상태를 LED_OFF 상태로 변경.
LED_OFF 상태면 가운데 LED 2개를 모두 켜고, 현 상태를 LED_ON 상태로 변경.] 이 과정을 반복.

```
#include "main.h"

/* USER CODE BEGIN PD */

#define  LED_OFF 0

#define  LED_ON 1

/* USER CODE END PD */



/* USER CODE BEGIN PV */

uint8_t led_state = 0;

/* USER CODE END PV */
```

```
115 #if 1
116       if(!HAL_GPIO_ReadPin(GPIOD, GPIO_PIN_2)) {
117           HAL_Delay(100);
118           if(!HAL_GPIO_ReadPin(GPIOD, GPIO_PIN_2)) {
119               if(led_state == LED_ON) {
120                   HAL_GPIO_WritePin(GPIOB, GPIO_PIN_15, 0); //control+space
121                   HAL_GPIO_WritePin(GPIOC, GPIO_PIN_0, 0);
122                   led_state = LED_OFF;
123               } else {
124                   HAL_GPIO_WritePin(GPIOB, GPIO_PIN_15, 1);
125                   HAL_GPIO_WritePin(GPIOC, GPIO_PIN_0, 1);
126                   led_state = LED_ON;
127               }
128           }
129       }
130 #endif
```

+과제: 밝기 조절이 가능한 랜턴 프로그램

Lantern program with adjustable brightness

📌 구조체 이용

```
#include<main.h>
int main(){
typedef struct led{
GPIO_TypeDef* port;
uint16_t pin;
}LED;
LED led[8]={

{GPIOB, GPIO_PIN_12},
{GPIOB, GPIO_PIN_13},
{GPIOB, GPIO_PIN_14},
{GPIOB, GPIO_PIN_15},
{GPIOC, GPIO_PIN_0},
{GPIOC, GPIO_PIN_1},
{GPIOC, GPIO_PIN_2},
{GPIOC, GPIO_PIN_3}
};
uint8_t Led_Number = 0;
while(1){
if(!HAL_GPIO_ReadPin(GPIOD, GPIO_PIN_2)){
HAL_Delay(100);
if(!HAL_GPIO_ReadPin(GPIOD, GPIO_PIN_2)){
if(Led_Number!=0){
HAL_GPIO_WritePin(led[Led_Number-1].port, led[Led_Number-1].pin, GPIO_PIN_RESET);
Led_Number--;
}
}
}
if(!HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_4)){
HAL_Delay(100);
if(!HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_4)){
if(Led_Number!=8){
Led_Number++;
}
HAL_GPIO_WritePin(led[Led_Number-1].port, led[Led_Number-1].pin, GPIO_PIN_SET);
}
}
}
}
```