



آزمایشگاه ریزپردازنده

تمرین هفت

سینا عربی - سالار جهانگیری - امیرعلی وکیلی

سوالات تحلیلی:

نحوه دسترسی به رجیسترهای میکروکنترلر به زبان C را با یک مثال توضیح دهید.

برای انجام این کار باید کتابخانه "stm32f4xx.h" را اینکلود کرده و در ابتدای کد main (یا تعریف یک تابع init برای کد) مقداری را برای مودهای GPIO انجام می‌دهیم.

```
RCC->AHB1ENR    // برای تنظیم کلاک  
GPIOC->MODER    // A, B, C برای تنظیم پین‌های
```

```
GPIOC->ODR // C برای نوشتن در پورت  
GPIOC->IDR // C برای خواندن از پورت
```

به طور مثال کد زیر در صورت فعال بودن پین شماره پنج (active low) A، عددی را در پورت C می‌نویسد.

```
#include <stdint.h>  
#include <stdio.h>  
  
#include "stm32f4xx.h"  
int main(void){  
  
    RCC->AHB1ENR = ((1 << 0) | (1 << 2));  
    GPIOC->MODER |= ((1 << 0) | (1 << 2) | (1 << 4) | (1 << 6)  
| (1 << 8) | (1 << 10) | (1 << 12));  
    GPIOA->MODER &= ((1 << 10) & (1 << 12));  
  
    while (1)  
    {  
        if ((GPIOA->IDR & (1 << 5)) == 0){  
            GPIOC->ODR = 0xC0;  
        }  
    }  
}
```

نحوه کار:

```
1 #include <stdint.h>
2 #include <stdio.h>
3
4 #include "stm32f4xx.h"
5
6 int seg[10] = {0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f};
7 void delay(void)
8 {
9     for (int i = 0; i < 2000000; i++)
10         ;
11 }
12
13 void delay_inside(void)
14 {
15     for (int i = 0; i < 500000; i++)
16         ;
17 }
18
```

در ابتدا آرایه‌ای به اندازه ده عدد برای هر کدام از ارقام 0 تا 9، مپینگ نمایش آنها برای المنت 7-segment را تعریف می‌کنیم. به طور کلی دو delay تعریف می‌کنیم، تابع delay برای ایجاد وقفه نمایش بین هر دو عدد و delay_inside بین ارقام هر عدد.

```
int prime(int number)
{
    int flag = 0;
    if (number == 0 || number == 1)
        flag = 1;

    for (int i = 2; i <= number / 2; ++i)
    {
        if (number % i == 0)
        {
            flag = 1;
            break;
        }
    }
}
```

```

    }
}

if (flag == 0)
    return 1;
else
    return 0;
}

int palindrome(int number)
{
    int r, sum = 0, temp;
    temp = number;
    while (number > 0)
    {
        r = number % 10;
        sum = (sum * 10) + r;
        number = number / 10;
    }
    if (temp == sum)
        return 1;
    else
        return 0;
}

```

دو تابع پر ایم و پالیندروم به زبان C برای تشخیص اول و پالیندرومی بودن اعداد تعریف می‌کنیم.

```

void show_number(int number)
{

    while (number > 0)
    {
        int num = number % 10;
        number /= 10;

        if (num == 0)
        {

```

```
        GPIOC->ODR = ~seg[0];
    }
    else if (num == 1)
    {
        GPIOC->ODR = ~seg[1];
    }
    else if (num == 2)
    {
        GPIOC->ODR = ~seg[2];
    }
    else if (num == 3)
    {
        GPIOC->ODR = ~seg[3];
    }
    else if (num == 4)
    {
        GPIOC->ODR = ~seg[4];
    }
    else if (num == 5)
    {
        GPIOC->ODR = ~seg[5];
    }
    else if (num == 6)
    {
        GPIOC->ODR = ~seg[6];
    }
    else if (num == 7)
    {
        GPIOC->ODR = ~seg[7];
    }
    else if (num == 8)
    {
        GPIOC->ODR = ~seg[8];
    }
    else if (num == 9)
    {
```

```

        GPIOC->ODR = ~seg[9];
    }

    delay_inside();
    GPIOC->ODR = ~0;
}
}

```

در این تابع آرگومان ورودی یک عدد گرفته و نمایش 7-segment آن را در پورت C می‌نویسیم (PC0-PC6). در آخر فراخوانی تابع یک delay_inside 0.5 ثانیه ایجاد می‌کنیم و 7-seg را کلیر می‌کنیم.

```

int main(void)
{

    int prime_pali[20] = {};
    RCC->AHB1ENR = ((1 << 0) | (1 << 2));
    GPIOC->MODER |= ((1 << 0) | (1 << 2) | (1 << 4) | (1 << 6)
| (1 << 8) | (1 << 10) | (1 << 12));
    GPIOA->MODER &= ((1 << 10) & (1 << 12));

    while (1)
    {
        int i = 0;
        int num = 1;

        for (int i = 0; i < 20; i++)
        {
            while (1)
            {
                num++;
                if (prime(num) == 1 && palindrome(num) == 1)
                {
                    prime_pali[i] = num;
                    break;
                }
            }
        }
    }
}

```

```

    }
    while (1)
    {
        int number = prime_pali[i];
        show_number(number);
        delay();
        if ((GPIOA->IDR & (1 << 5)) == 0)
        {
            if(i > 0){
                number = prime_pali[--i];
                show_number(number);
                delay();
            }
        }
        else if ((GPIOA->IDR & (1 << 6)) == 0)
        {
            if(i < 100){
                number = prime_pali[++i];
                show_number(number);
                delay();
            }
        }
    }
    for (;;)
    ;
}
}

```

در تابع main ابتدا مقداردهی مربوط به mode های رجیسترهای GPIO را تعیین می‌کنیم که پورت C خروجی (P0 تا P6) و پورت A ورودی (P5, P6) سویچ‌ها می‌باشد. حال در یک لوپ True اعداد پالیندروم و پرایم را یافته و در یک آرایه ذخیره می‌کنیم، در ادامه در صورتی که سویچ پایین (پورت PA5) فعال بود، عدد قبلی و در صورتی که سویچ بالا (پورت PA6) فعال بود، عدد بعدی نمایش داده می‌شود و سپس یک delay 2 ثانیه‌ای برای وضوح بهتر ایجاد می‌کنیم.