

ESS 2023 Labs 1 Solutions

Which pin is which?

- PD12: LED_GREEN
- PD13: LED_ORANGE
- PD14: LED_RED
- PD15: LED_BLUE

Why is HAL a good idea?

- HAL decouples the software and hardware being used
- Makes it more straightforward to port to a different platform by modifying code in one place
- Properly written HAL can make testing easier

How should we initialise the LEDs? Why?

- We want to turn them off
- Less important for LEDs, but in other devices we could be controlling (motors, actuators etc.) not initialising to 0 could be dangerous
- Could want to turn things on as part of initialisation, for example slave devices that we want to send power to

```
void led_green_init(void)
{
    *(uint32_t*)0x40020C14 = 0x0000;
}

void led_green_on(void)
{
    *(uint32_t*)0x40020C14 = 0x1000;
}

void led_green_off(void)
{
    *(uint32_t*)0x40020C14 = 0x0000;
}
```

```
int main(void)
{
    // Initialise system
    HAL_Init();
    // Initialise peripherals
    ess_helper_init();
    // Main loop
    led_green_init();
    led_green_on();
    while(1) {

    }
}
```

```
int main(void)
{
    // Initialise system
    HAL_Init();
    // Initialise peripherals
    ess_helper_init();
    // Main loop
    while(1) {
        led_green_init();
        led_green_on();
    }
}
```

```
#define PORTD ((volatile uint32_t*)0x40020C14
```

```
void led_green_init(void)
```

```
{
```

```
    *PORTD = 0x0000;
```

```
}
```

```
void led_green_on(void)
```

```
{
```

```
    *PORTD = 0x1000;
```

```
}
```

```
void led_green_off(void)
```

```
{
```

```
    *PORTD = 0x0000;
```

```
}
```


Bit shifts

- Why is `mask = 0x01 << bits_position;` working?

```
static uint32_t led_green_pin;

void led_green_init(uint32_t pin)
{
    uint32_t led_green_pin = 0x01 << bits_position;

    *PORTD = 0x0000;
}

void led_green_on(void)
{
    *PORTD = led_green_pin;
}

void led_green_off(void)
{
    *PORTD = 0x0000;
}
```

```
void led_init(LED_t * led, volatile uint32_t * port, uint32_t pin)
{
    led->port = port;
    led->pin = pin;

    led_off(led);
}
```

```
void led_on(LED_t * led)
{
    *led->port |= 0x01 << (led->pin);
}
```

```
void led_off(LED_t * led)
{
    *led->port &= ~(0x01 << (led->pin));
}
```

```
int main(void)
{
    LED_t led_green;
    LED_t led_orange;
    LED_t led_blue;
    LED_T led_red;

    HAL_Init();
    ess_helper_init();

    led_init(&led_green, PORTD, 12);
    led_init(&led_orange, PORTD, 13);
    led_init(&led_red, PORTD, 14);
    led_init(&led_blue, PORTD, 15);

    led_on(&led_green);
    led_on(&led_orange);
    led_on(&led_red);
    led_on(&led_blue);

    while(1) {

    }
}
```

```
void delay_msec(uint32_t delay)
{

    uint32_t k;

    volatile uint32_t internal_fake;

    while (delay-- >0)
    {
        for (k=0; k<20000; k++)
        {
            internal_fake--;
        }
    }
}
```