Blinking an LED the hard way

Motivation

- Blinking an LED is considered the Hello World of microcontrollers (Arduino)
  - Uses one of the most basic functions of a microcontroller: toggling a pin
  - It's very easy to do using a library
    - But HOW does it actually work?
      - Note: It's hardware specific

Toggling a pin with Arduino's library

```
int pin_number = 1;

void loop() {
        digitalWrite(pin_number, false);
        delay(1000);
        digitalWrite(pin_number, true);
        delay(1000);
}
```

Toggling a pin from scratch

- Read the chip's manual (in this example, ESP32-S3's manual)

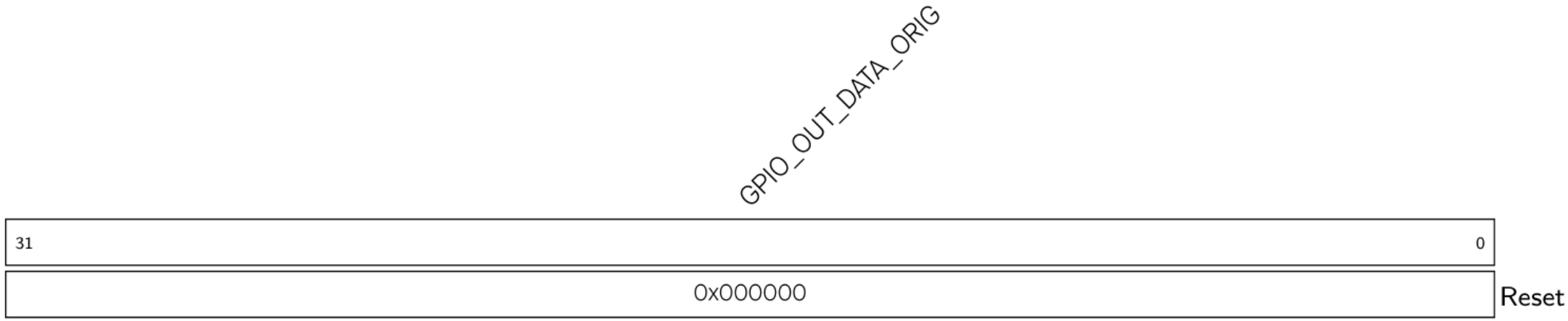
  ### 6.5.3 Simple GPIO Output

  GPIO matrix can also be used for simple GPIO output. This can be done as below:

  - Set GPIO matrix GPIO_FUNC*n*_OUT_SEL with a special peripheral index 256 (0x100);
  - Set the corresponding bit in GPIO_OUT_REG[31:0] or GPIO_OUT1_REG[21:0] to the desired GPIO output value.

Register controlling the GPIO output value

Register 6.2. GPIO_OUT_REG (0x0004)



GPIO_OUT_DATA_ORIG GPIO0 ~ 21 and GPIO26 ~ 31 output values in simple GPIO output mode. The values of bit0 ~ bit21 correspond to the output values of GPIO0 ~ 21, and bit26 ~ bit31 to GPIO26 ~ 31. Bit22 ~ bit25 are invalid. (R/W)

GPIO base address

4.3.5.1 Module/Peripheral Address Mapping

Table 4-3 lists all the modules/peripherals and their respective address ranges. Note that the address space of specific modules/peripherals is defined by "Boundary Address" (including both Low Address and High Address).

Table 4-3. Module/Peripheral Address Mapping

Target	Boundary Address		Size (KB)	Notes
	Low Address	High Address		
UART Controller 0	0x6000_0000	0x6000_0FFF	4	
Reserved	0x6000_1000	0x6000_1FFF		
SPI Controller 1	0x6000_2000	0x6000_2FFF	4	
SPI Controller 0	0x6000_3000	0x6000_3FFF	4	
GPIO	0x6000_4000	0x6000_4FFF	4	
Reserved	0x6000_5000	0x6000_6FFF		
eFuse Controller	0x6000_7000	0x6000_7FFF	4	

Toggling a pin from scratch: coding time

```
unsigned int *gpio_out_register_address = 0x60004004;

void loop() {
        *gpio_out_register_address = 0b00000000000000000000000000000000;
        delay(1000);
        *gpio_out_register_address = 0b00000000000000000000000000000010;
        delay(1000);
}
```

Demo

Thanks