

Lesson 2

GPIO

Lecturer: Harvard Tseng

Development Board

- STM32F0Discovery
 - User LD3: PC9
 - User LD4: PC8
 - B1 USER: PA0
- NucleoSTM32F401RE
 - B1 USER: PC13
 - User LD2: PA5

Reset and clock control (RCC)

- AHB peripheral clock enable register (RCC_AHBENR)

RCC_AHBENR

- Set and cleared by software.
 - 0: clock disable.
 - 1: clock enable.

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|-------|------|---------|---------|----------|---------|---------|---------|--------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | TSCEN | Res. | IOPF EN | IOPE EN | IOPD EN | IOPC EN | IOPB EN | IOPA EN | Res. |
| | | | | | | | rw | | rw | rw | rw | rw | rw | rw | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | CRC EN | Res. | FLITF EN | Res. | SRAM EN | DMA2 EN | DMA EN |
| | | | | | | | | | rw | | rw | | rw | rw | rw |

General-purpose I/Os (GPIO)

- GPIO port mode register (GPIOx_MODER) (x = A..F)
- GPIO port input data register (GPIOx_IDR) (x = A..F)
- GPIO port output data register (GPIOx_ODR) (x = A..F)
- GPIO port bit set/reset register (GPIOx_BSRR) (x = A..F)

GPIOx_MODER

- MODERy[1:0]: Port x configuration bits (y = 0..15)
 - 00: Input mode (reset state)
 - 01: General purpose output mode
 - 10: Alternate function mode
 - 11: Analog mode

[illegible]

GPIOn_IDR

- Bits 15:0 IDRy: Port input data bit ($y = 0..15$)
 - These bits are read-only. They contain the input value of the corresponding I/O port.

[illegible]

GPIORx_ODR

- Bits 15:0 ODRy: Port output data bit ($y = 0..15$)
 - These bits can be read and written by software.

[illegible]

GPIOx_BSRR

- Bits 31:16 B_{Ry}: Port x reset bit y (y = 0..15)
 - 1: Resets the corresponding ODR_x bit
- Bits 15:0 B_{Sy}: Port x set bit y (y= 0..15)
 - 1: Sets the corresponding ODR_x bit
- Writing 0 does nothing.

[illegible]

CAUTION!!!

- When changing certain bits of the register, all the other bits must be left unchanged.

Example 1

Configure PA0 as input

PA0 as input

- GPIOA_MODER:

xxxx xxxx xxxx xxxx xxxx xxxx xxxx xx00

- GPIOA_IDR:

xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxx0 :B2 is released

xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxx1 :B2 is pressed

Example 2

Configure PC8 as output

PC8 as output

- GPIOC_MODER:

xxxx xxxx xxxx xx01 xxxx xxxx xxxx xxxx

- GPIOC_BSRR:

xxxx xxx1 xxxx xxxx xxxx xxxx xxxx :LD4 reset

xxxx xxxx xxxx xxxx xxxx xxx1 xxxx xxxx :LD4 set

- GPIOx_ODR: (option)

xxxx xxxx xxxx xxxx xxxx xxx0 xxxx xxxx :LD4 reset

xxxx xxxx xxxx xxxx xxxx xxx1 xxxx xxxx :LD4 set

Exercise

Main program

- If button pressed, LED on.
- If button released, LED off.
- Constantly scan.
- Use your GPIO driver.

GPIO Interface

Write the functions for GPIO pins

Functions

- Configure GPIO pins

```
void GPIO_Set_Direction(GPIO_TypeDef* GPIOx, uint16_t  
inputMask, uint16_t outputMask);
```

- Read the input value

```
uint8_t GPIO_ReadInputDataBit(GPIO_TypeDef* GPIOx,  
int Pin);
```

- Write value to a pin

```
void GPIO_WriteBit(GPIO_TypeDef* GPIOx, int Pin, uint8_t  
BitVal);
```

Explain

- `uint16_t inputMask = 0x0001 //Px0 as input`
- `uint16_t outputMask = 0x0300 //Px8, Px9 as output`