Lesson 2
GPIO

Lecturer: Harvard Tseng

Development Board

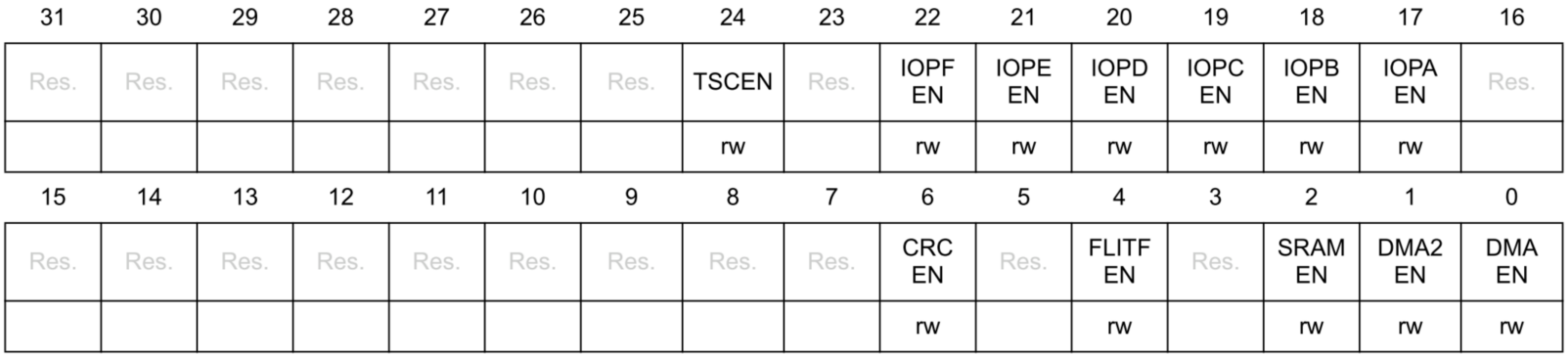
- STM32F0Discovery
  - User LD3: PC9
  - User LD4: PC8
  - B1 USER: PA0
- NucleoSTM32F401RE
  - B1 USER: PC13
  - User LD2: PA5

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	

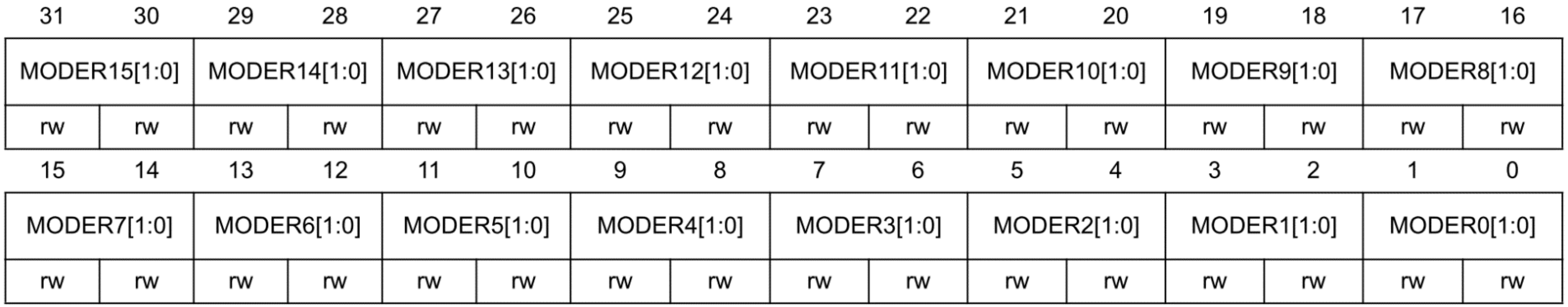
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRC EN	Res.	FLITF EN	Res.	SRAM EN	DMA2 EN	DMA EN
									rw		rw		rw	rw	rw

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

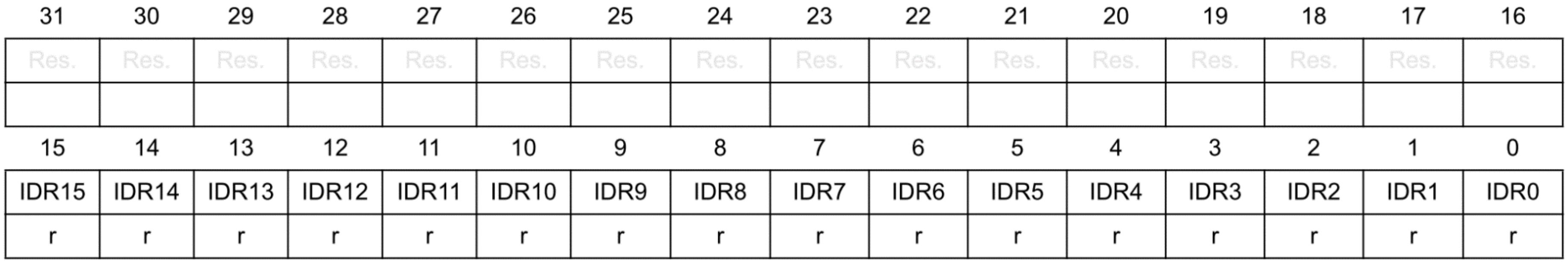
- MODERy[1:0]: Port x configuration bits (y = 0..15)
  - 00: Input mode (reset state)
  - 01: General purpose output mode
  - 10: Alternate function mode
  - 11: Analog mode



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

GPIOx_IDR

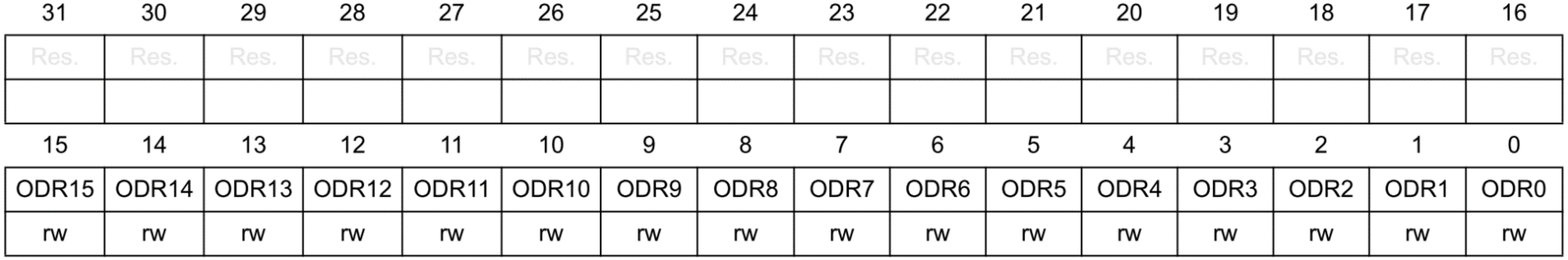
- Bits 15:0 IDRy: Port input data bit (y = 0..15)
  - These bits are read-only. They contain the input value of the corresponding I/O port.



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

GPIOx_ODR

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



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

GPIOx_BSRR

- Bits 31:16 BRy: Port x reset bit y (y = 0..15)
  - 1: Resets the corresponding ODRx bit
- Bits 15:0 BSy: Port x set bit y (y= 0..15)
  - 1: Sets the corresponding ODRx bit
- Writing 0 does nothing.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

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 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

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

Exercise

Main program

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