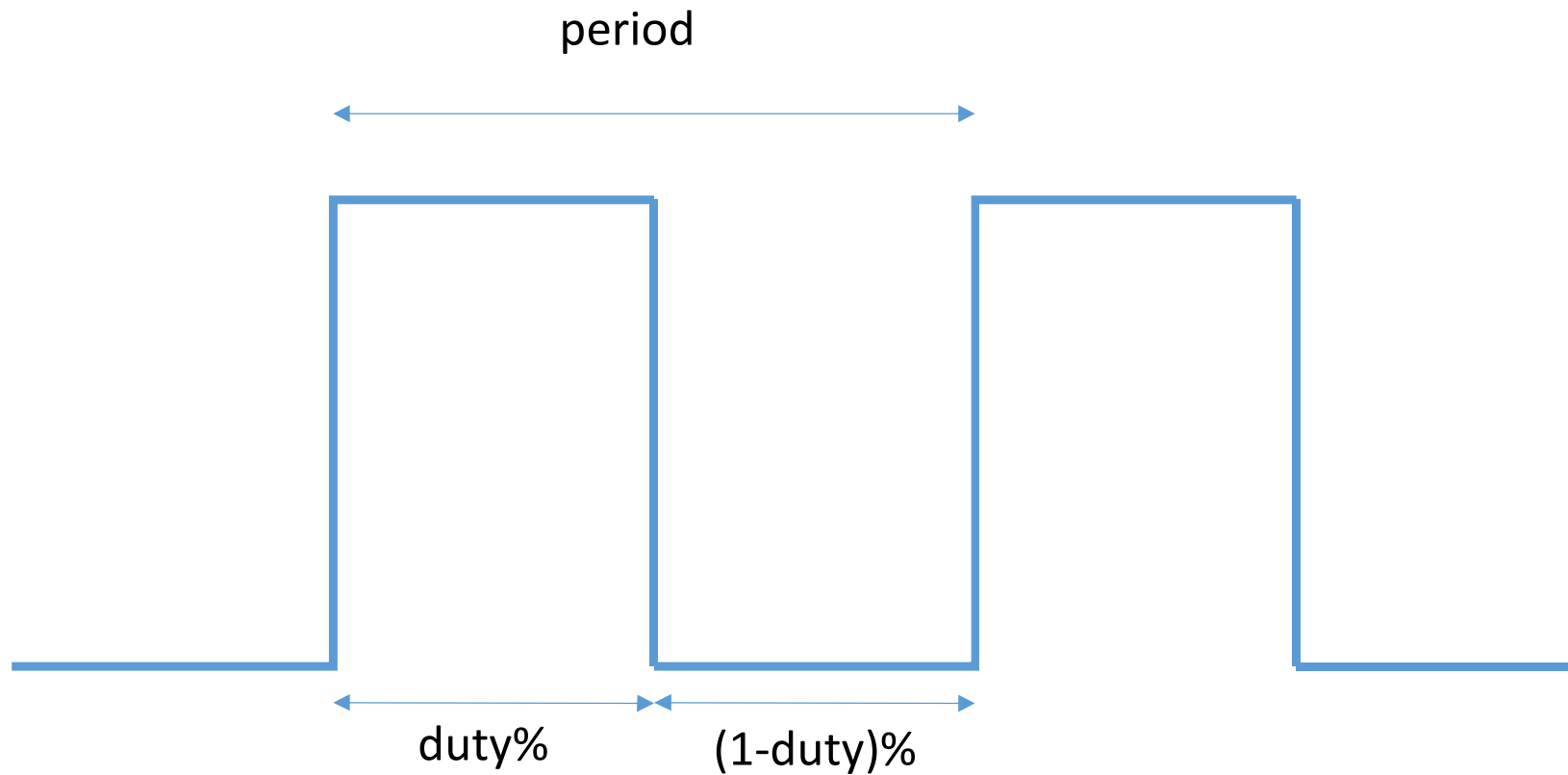


# Lesson 3

## Timer

Lecturer: Harvard Tseng

# How to generate square wave?



We need timer

# Reset and clock control (RCC)

- APB peripheral clock enable register 1  
(RCC\_APB1ENR)

# RCC\_APB1ENR

- Set and cleared by software.
  - 0: clock disable.
  - 1: clock enable.
- Writing **RCC\_APB1ENR\_TIM2EN** to enable Timer2 clock.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			PWR EN	Reserved				I2C3 EN	I2C2 EN	I2C1 EN	Reserved			USART2 EN	Reser- ved
			rw					rw	rw	rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 EN	SPI2 EN	Reserved		WWDG EN	Reserved						TIM5 EN	TIM4 EN	TIM3 EN	TIM2 EN	
rw	rw			rw							rw	rw	rw	rw	

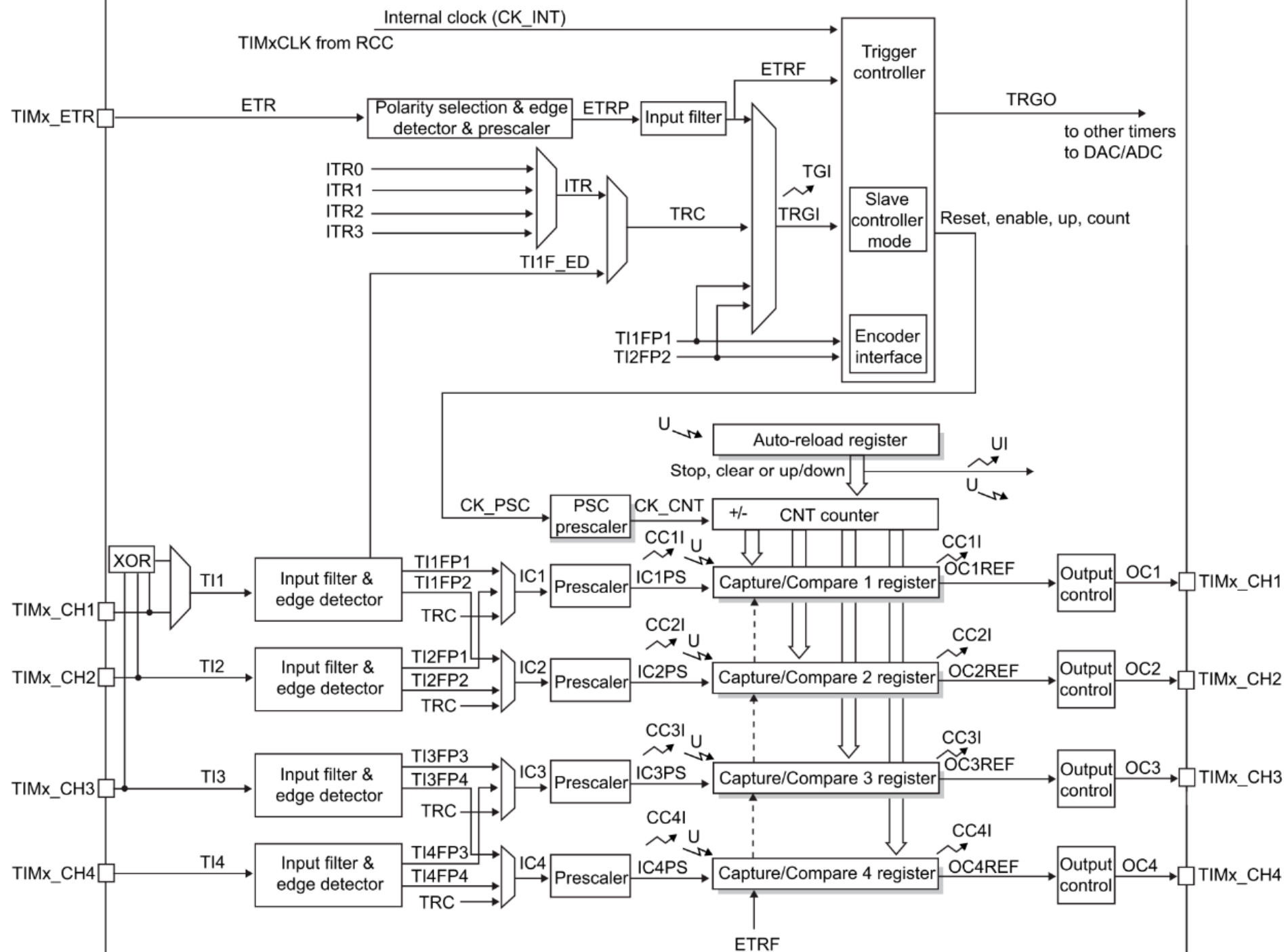
# General-purpose timers

- Using **Timer2**(32-bit timer)
- TIMx\_CNT – Counter Register- is clocked by the prescaler output
- TIMx\_PSC – Prescaler Register- will divide source clock frequency
- TIMx\_ARR – Auto-Reload Register- will automatically reload value to CNT when CNT is overflow/underflow
- TIMx\_CR1 – Control Register 1

# TIMx\_CR1

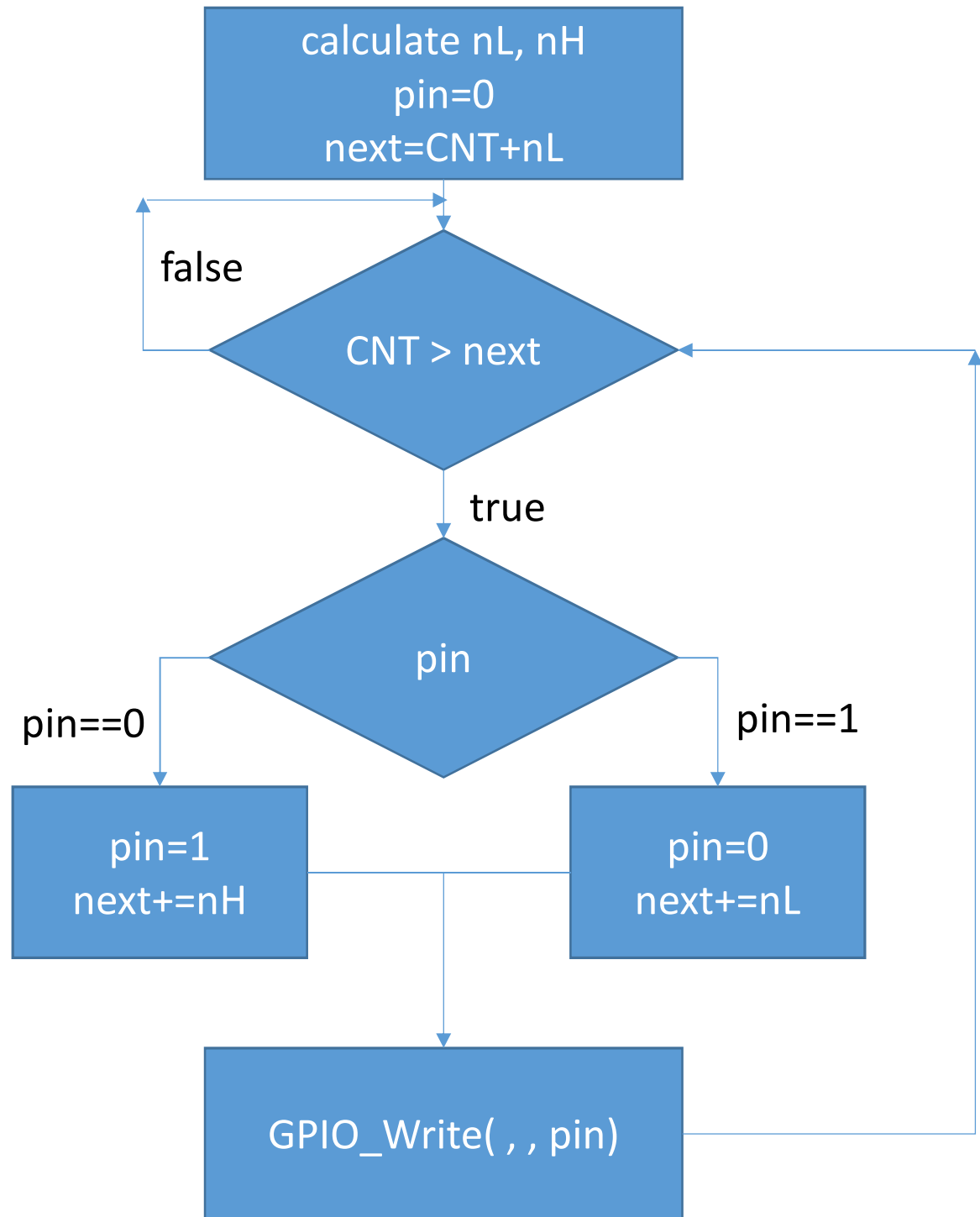
- Bit 4 DIR: Direction
  - 0: Counter used as upcounter
  - 1: Counter used as downcounter
- Writing **TIM\_CR1\_DIR** to have down count counter
- Writing **TIM\_CR1\_CEN** to enable counter

[illegible]





# Generate square wave



Exercise

# Main program

- If button pressed, constantly generate 5Hz, 50% duty signal to LED. If button released, LED turns off.
- When LED turns off, timer should not count.
- Use your own driver.

# RCC interface

- AHB1 Peripheral clocks configuration
  - `void RCC_AHB1PeriphClockCmd(uint32_t RCC_Periph, FunctionalState NewState);`
- APB1 Peripheral clocks configuration
  - `void RCC_APB1PeriphClockCmd(uint32_t RCC_Periph, FunctionalState NewState);`

# Timer interface

- Set prescaler value
  - void TIM\_PrescalerConfig(TIM\_TypeDef\* TIMx, uint16\_t Prescaler);
- Set counter value
  - void TIM\_SetCounter(TIM\_TypeDef\* TIMx, uint32\_t Counter);
- Get counter value
  - uint32\_t TIM\_GetCounter(TIM\_TypeDef\* TIMx);
- Enable/Disable counter
  - void TIM\_Cmd(TIM\_TypeDef\* TIMx, FunctionalState NewState);