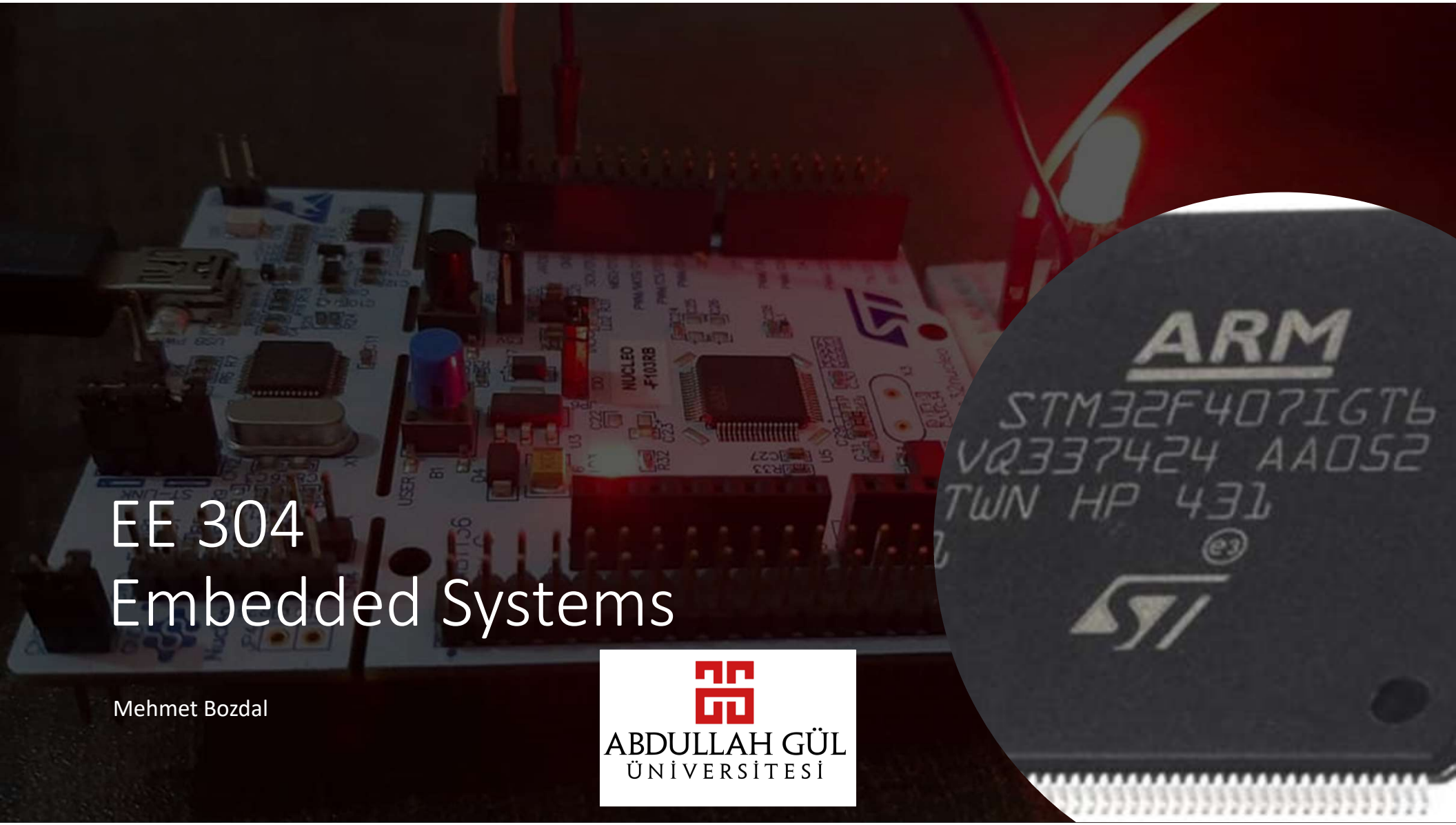


EE 304
Embedded Systems

Mehmet Bozdal

ABDULLAH GÜL
ÜNİVERSİTESİ

Agenda

- System Timer - SysTick

What is timer in MCU?

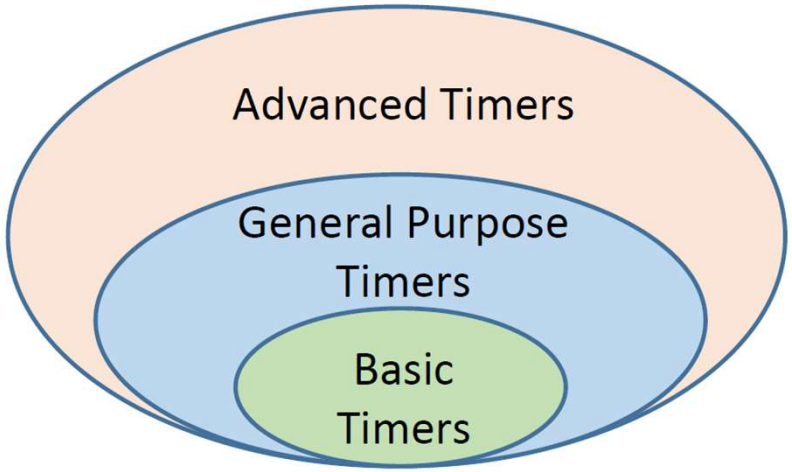
- Timers are hardware components that count based on **clock inputs**, allowing them to measure time and control timing functions ***without requiring continuous processor involvement***, which is crucial for efficient and precise timing operations in various applications.

Why do we use timers?

- Periodic interrupt to perform tasks
  - Execute periodic tasks (sensor readings at every second)
  - Generate music samples
- Provide accurate time delays
  - Instead of software loops
- Generate pulses or periodic waveforms
  - PWM signal for motor control
- Measure duration of an external event
  - Tachometer signal period to measure motor speed
  - Measure execution time

STM32 Timers

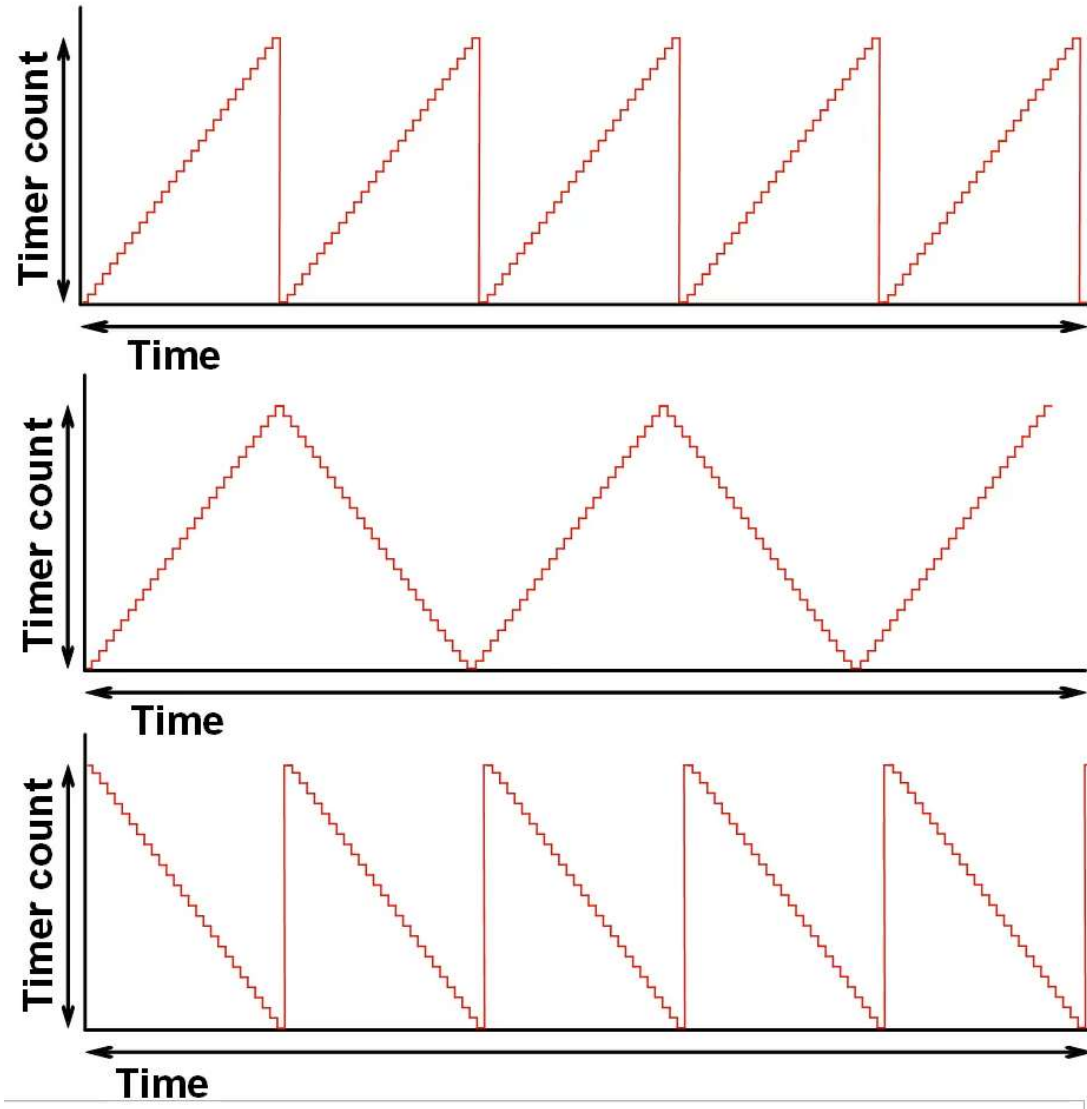
- Basic timers (TIM6 and TIM7)
- General-purpose timers (TIM2 to TIM5)
- General-purpose timers (TIM9 to TIM14)
- Advanced-control timers (TIM1)



- Real-time clock (RTC)
- Independent watchdog (IWDG)
- Window watchdog (WWDG)

Counting Modes

- Up
- Up/Down
- Center-aligned



System Timer

- The processor has a **24-bit** system timer, SysTick, that ***counts down*** from the reload value to zero, reloads (wraps to) the value in the LOAD register on the next clock edge, then counts down on subsequent clocks.
- The SysTick counter runs on the processor clock. If this clock signal is stopped for low power mode, the SysTick counter stops.
- SysTick can be configured to generate interrupts at regular intervals.
- When the processor is halted for debugging the counter does not decrement.

SysTick timer (STK)

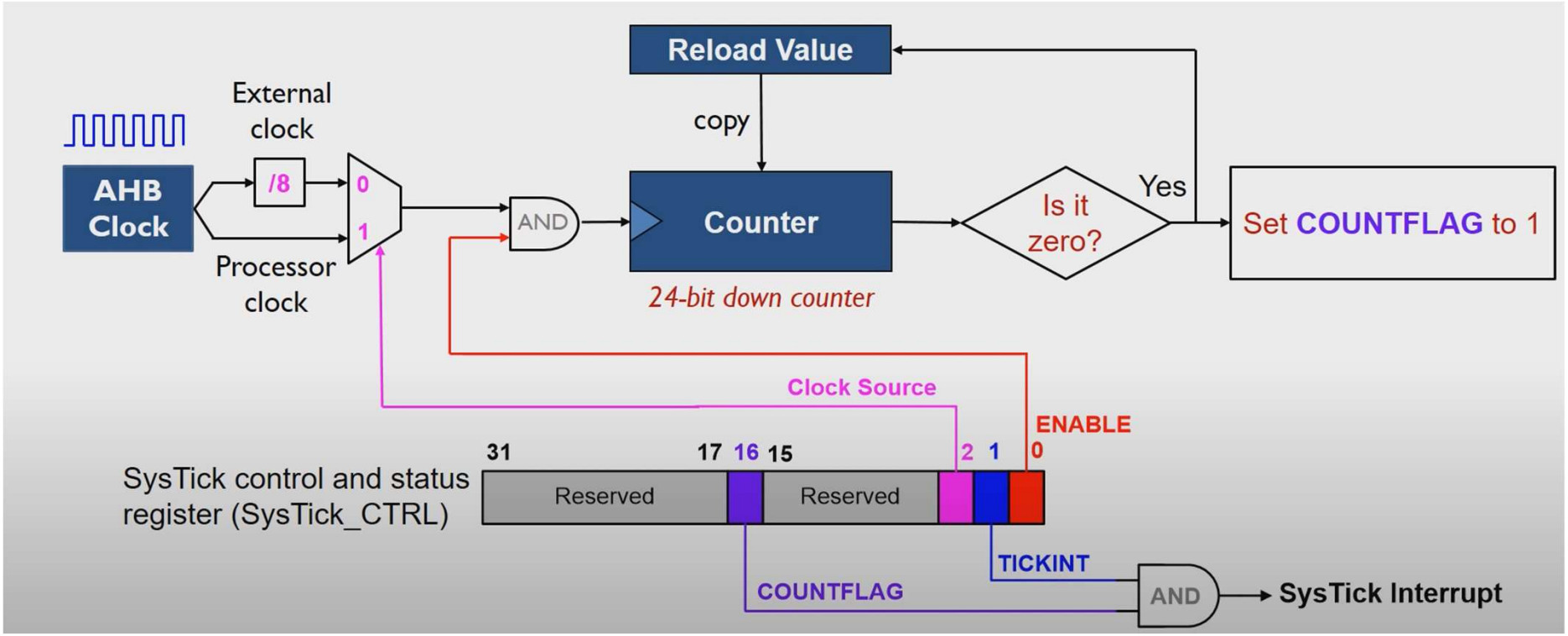


Figure from: Embedded Systems with Arm Cortex-M Microcontrollers in Assembly Language and C - Yifeng Zhu

System Timer - Usage

- The SysTick timer is often used in embedded systems for tasks such as creating accurate time delays, measuring time intervals, and implementing real-time operating system (RTOS) tick counters.

SysTick Register Map

- The SysTick timer is part of the Cortex-M core and is not specific to STM32. The base address of the SysTick register block is 0xE000 E010.

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	STK_CTRL	Reserved															COUNTFLAG	Reserved													CLKSOURCE	TICK INT	EN ABLE
	Reset Value																0														0	0	0
0x04	STK_LOAD	Reserved								RELOAD[23:0]																							
	Reset Value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	STK_VAL	Reserved								CURRENT[23:0]																							
	Reset Value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	STK_CALIB	Reserved								TENMS[23:0]																							
	Reset Value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AGÜ

Check Registers

PM0056
Programming manual

STM32F10xxx/20xxx/21xxx/L1xxxx
Cortex®-M3 programming manual

SysTick - Configuration

- The SysTick timer can be configured through its control and status registers. Key parameters include the clock source selection, enabling or disabling the timer, interrupt option.

SysTick control and status register (STK_CTRL)

Address offset: 0x00

Reset value: 0x0000 0000

Required privilege: Privileged

The SysTick CTRL register enables the SysTick features.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															COUNT FLAG
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													CLKSOURCE	TICKINT	ENABLE
													rw	rw	rw

SysTick control and status register (STK_CTRL)

ENABLE: Enables the counter. When ENABLE is set to 1, the counter loads the RELOAD value from the LOAD register and then counts down. On reaching 0, it sets the COUNTFLAG to 1 and optionally asserts the SysTick depending on the value of TICKINT. It then loads the RELOAD value again, and begins counting.

The SysTick CTRL register enables the SysTick features.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															COUNT FLAG
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													CLKSOURCE	TICKINT	ENABLE
													rw	rw	rw

SysTick control and status register (STK_CTRL)

- ***COUNTFLAG***: Returns 1 if timer counted to 0 since last time this was read.

The SysTick CTRL register enables the SysTick features.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															COUNT FLAG
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													CLKSO URCE	TICK INT	EN ABLE
													rw	rw	rw

SysTick - Clock Source

- The SysTick timer can use either the processor clock (AHB) or AHB/8 as its clock source. **0:** AHB/8 **1:** Processor clock (AHB)

The SysTick CTRL register enables the SysTick features.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															COUNT FLAG
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													CLKSOURCE	TICKINT	ENABLE
													rw	rw	rw

SysTick control and status register (STK_CTRL)

- ***TICKINT:*** SysTick exception request enable

0: Counting down to zero does not assert the SysTick exception request

1: Counting down to zero to asserts the SysTick exception request.

- Note: Software can use COUNTFLAG to determine if SysTick has ever counted to zero.

The SysTick CTRL register enables the SysTick features.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															COUNT FLAG
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													CLKSOURCE	TICKINT	ENABLE
													rw	rw	rw

SysTick - Reload

- The SysTick timer operates as a down counter from a reload value. When the counter reaches zero, it can either generate an interrupt or automatically reload the original value and continue counting.

SysTick reload value register (STK_LOAD)

Address offset: 0x04

Reset value: 0x0000 0000

Required privilege: Privileged

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								RELOAD[23:16]							
								rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RELOAD[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

AGÜ

SysTick - Reload

- The RELOAD value can be any value in the range 0x00000001-0x00FFFFFF (16,777,215). A start value of 0 is possible, but has no effect because the SysTick exception request and COUNTFLAG are activated when counting from 1 to 0.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								RELOAD[23:16]							
								rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RELOAD[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

SysTick - Reload

The RELOAD value is calculated according to its use:

- To generate a multi-shot timer with a period of N processor clock cycles, use a RELOAD value of N-1. For example, if the SysTick interrupt is required every 100 clock pulses, set RELOAD to 99.

- To deliver a single SysTick interrupt after a delay of N processor clock cycles, use a RELOAD of value N. For example, if a SysTick interrupt is required after 400 clock pulses, set RELOAD to 400.

- $SysTick\ Interval\ =\ Clock\ Period * (Reload\ Value + 1)$

SysTick current value register (STK_VAL)

- The VAL register contains the current value of the SysTick counter.
- Reads return the current value of the SysTick counter.
- A write of any value clears the field to 0, and also clears the COUNTFLAG bit in the STK_CTRL register to 0.

SysTick current value register (STK_VAL)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CURRENT[23:16]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CURRENT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

SysTick - Interrupt

- When the SysTick timer reaches zero, it can generate an interrupt. This feature is commonly used periodic tasks.

SysTick control and status register (STK_CTRL)

Address offset: 0x00

Reset value: 0x0000 0000

Required privilege: Privileged

The SysTick CTRL register enables the SysTick features.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															COUNT FLAG
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													CLKSOURCE	TICKINT	ENABLE
													rw	rw	rw

SysTick design hints and tips

- The SysTick counter runs on the processor clock. If this clock signal is stopped for low power mode, the SysTick counter stops.

- Ensure software uses aligned word accesses to access the SysTick registers.

SysTick Setup - Polling

// disable systick

// configure clock source

// set reload value

// clear current value

// enable systick

More on this…

- Lecture 12: System Timer (SysTick)

Q&A

Any questions?