CSCE 438/838: Internet of Things

Last Class

- Introductions and Syllabus


Topologically Equivalent
A
Topologically Distinct

How about you?

- Name
- Major (topic/thesis)
- Why IoT?
- What do you want to get out of this class?
- Career Goal

What is the single most important thing you'd like to learn from this class?

Efficient and secure communication protocols between devices

5 0

Wireless communication between different devices

3 0

How to work with/build IoT devices

2 0

MQTT protocol

4 0

Learn more about data generation through sensors and understanding the stack of getting the data to cloud with the wireless communications perspective.

2 0

Devices Synchronization

1 0

How to integrate devices with cloud?

3 0

How IoT devices communicate

1 0

Wireless_communication

1 0

Wireless Communication Strategies

1 0

Better appreciation of how to integrate and utilize the cloud for IP devices.

0 0

The Future?

Terminal World

Prosthetics

Animism

Enchanted Objects

Next Class

- Let's enchant an object!
- Bring your favorite object to the class on Monday to enchant it!

Internet of Things

- Enchanted Objects
- B2B Things
- Self-reporting IoT

What is in a datasheet?

Data Sheet

- Embedded systems have mostly become chips
- Embedded system design has become circuit design (in hardware)
- When designing a circuit, you need to make many decisions about components, capabilities, parts, etc.
- Where do you find the information to make an informed decision?
- Make data sheets your ally

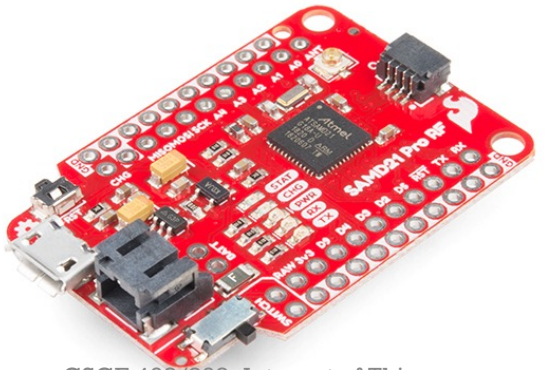
Data Sheet

- Traditionally for hardware
- Software closely follows hardware in embedded systems
- 10-1,000+ pages
- Contain too much data
  - Not enough time to dwell on
  - Need to quickly evaluate the data-sheet information

Reminder

Sparkfun Pro RF - LoRa

- https://www.sparkfun.com/products/14916
- **Microcontroller: SAMD21G18A**
  - ~~Cortex M0+~~
  - 256KB Flash Memory
  - 32MHz External Oscillator
  - 4 Digital and 5 Analog IO Pins with exclusive GND pins
- **Radio Module: Hope RFM95W LoRa modem**
  - Point to Point Radio capabilities
  - LoRa Enabled
  - Frequency range: 915 MHz
  - Range up to 1 mile line of sight
  - U.FL Antenna
- Arduino-based hardware/software



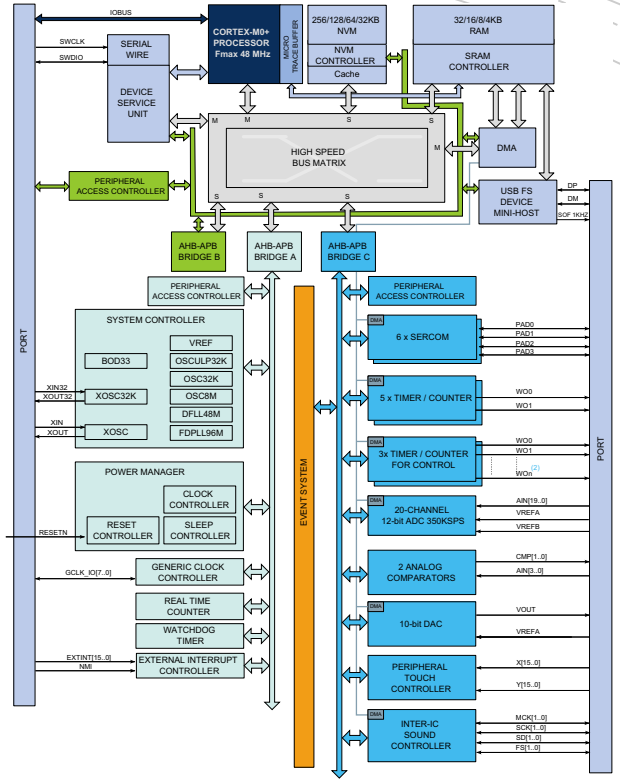
Last Class

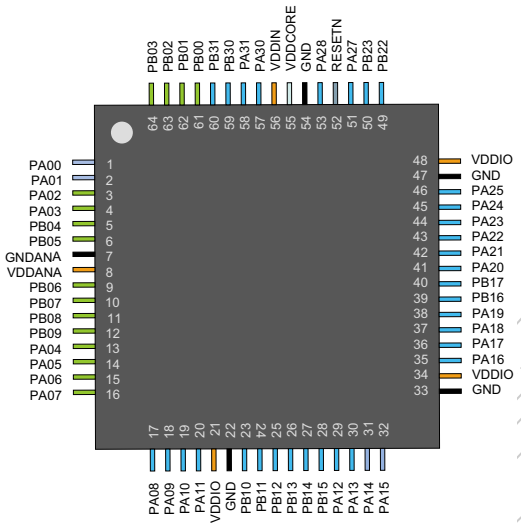
- Four futures
- Enchanted objects
- Datasheet

Common Sections

- Front page
- Functional block diagram
- Device pin-out
- Terminal functions
- Short-form description
- Special function registers

- Memory organization
- Flash memory
- Peripherals
- Peripheral file map
- Electrical characteristics
- Application information





Front Page

- Tear it out !
- Gives a brief overall description – mostly for marketing purposes
- Useful for picking out the crucial features of a device
  - Particular set of peripherals, e.g., A/D converter, I2C interface, etc.
- Inevitably some "cherry-picking" to show the device at its best
- Generally consists of
  - Features
  - Applications (very useful if you are looking at tons of similar components, but not complete)
  - Description

Data Sheet

- Device pin-out: Shows the style of packages and their connections
- Functional block diagram: Shows the main systems within the integrated circuit
- Terminal functions: Show what peripherals can be connected internally to each pin and expands the information shown on the pin-out

Data Sheet

- Short-form description: Gives a brief summary of the
  - CPU
  - Instruction set
  - Operating (low-power) modes
  - Interrupt vector addresses

Data Sheet

- Special function registers: Mainly control interrupts from the central functions rather than the peripherals
- Memory organization: Gives the main features of the memory map
- Flash memory: Gives further information on the segments of flash memory, which is important if you wish to erase it

Peripherals

- Brief summary for most modules
  - They are described fully in the family user guide
  - It is not always obvious how data are split between the two
- Most important for peripherals that have general-purpose inputs and outputs
  - May be connected internally or externally in different ways

Data Sheet

- Peripheral file map: Lists the peripheral registers with their standard names, which should be recognized by the compiler or assembler

Data Sheet

- Electrical characteristics: Cover a vast range of information.
- Keep safely within the "absolute maximum ratings" unless you wish to damage your device
- Probably have to concentrate on one or more sections closely for a particular application
- For example, suppose that the lifetime of a battery is most important
  - Several tables and plots show how the current drawn by the device depends on its operating mode,frequency, supply voltage, temperature, and so on.

Table 36-3. General Operating Conditions

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
V_{DD}	Power supply voltage		1.62(1)	3.3	3.63	V
V_{DDANA}	Analog supply voltage		1.62(1)	3.3	3.63	V
T_A	Temperature range		-40	25	85	°C
T_J	Junction temperature		-	-	100	°C

Notes: 1. With BOD33 disabled. If the BOD33 is enabled, check Table 36-19.

Data Sheet

- Application information: Particularly lengthy
- Tables that show how to configure each pin for its various functions
- The detailed circuitry associated with the pins is useful for hardware designers.
- For example, it shows that the inputs have Schmitt triggers, which are useful to reduce noise

Tables

- Body of the tables includes
  - Parameter identification
  - Test conditions
  - Temperature
  - Parameter data
  - Parameter units (!)
- Let's look at power consumption (Section 36.6)

Parameter Identification

- Parameter symbol and name
- Life would be easier if the parameter names and symbols conformed to universal standards
  - They don't
  - Must translate symbols when scanning multiple data sheets from multiple manufacturers
  - Parameter names or symbols are sometimes inconsistent between pages
  - Do not make assumptions, contact manufacturer

Test Conditions

- Parameter values are valid only when the test conditions prevail
- Ambient temperature, power-supply voltages, source resistance, load resistance, test frequency, common-mode voltage, open-loop gain, input signal, and any other important defining test parameters
- Test conditions sometimes (!) conflict with front page
  - Use test conditions instead of marketing material for your design

Parameter Data

- Subdivides into three columns
  - MIN, minimum; TYP, typical; MAX, maximum
- After initial testing on the first several groups of ICs, the manufacturer applies statistics to the data to obtain the mean value for each parameter
  - Variance and sigma
  - Six times sigma represents the maximum and minimum values
  - Often mean becomes the typical specification

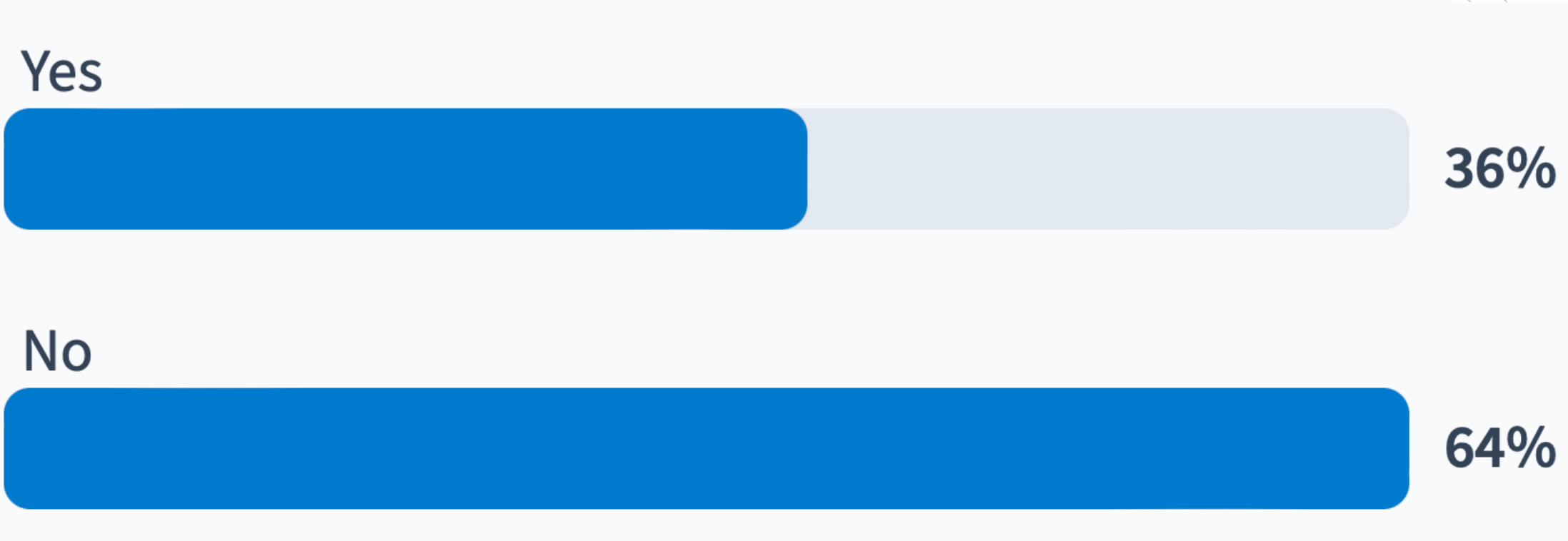
Further Reading

- Mancini – How to read a semiconductor data sheet
  - http://www.edn.com/design/analog/4323765/How-to-read-a-semiconductor-data-sheet

Watchdog Timers

Respond at **pollev.com/canvuran016**

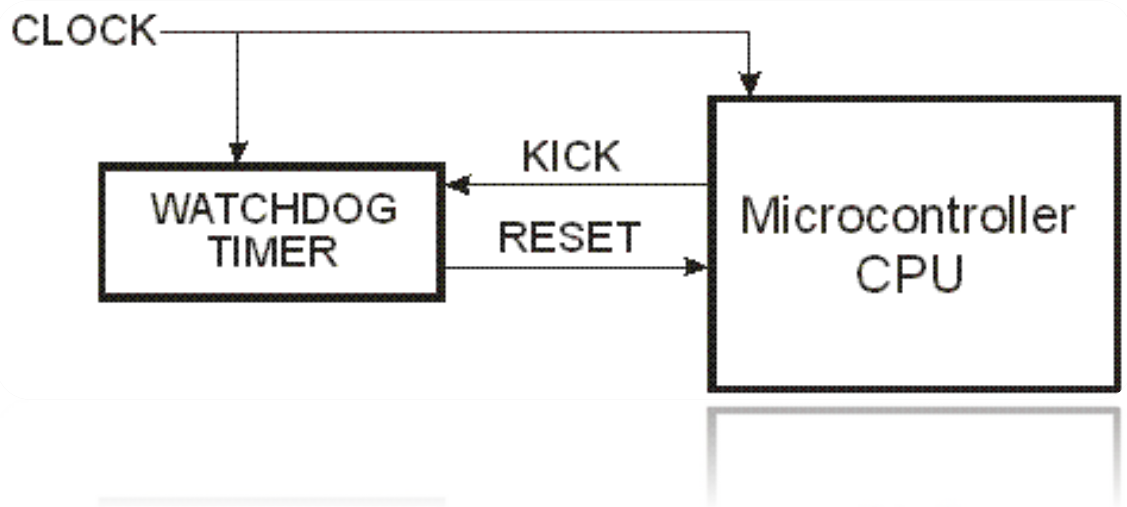
Have you *used* watchdog timers before?



Powered by Poll Everywhere

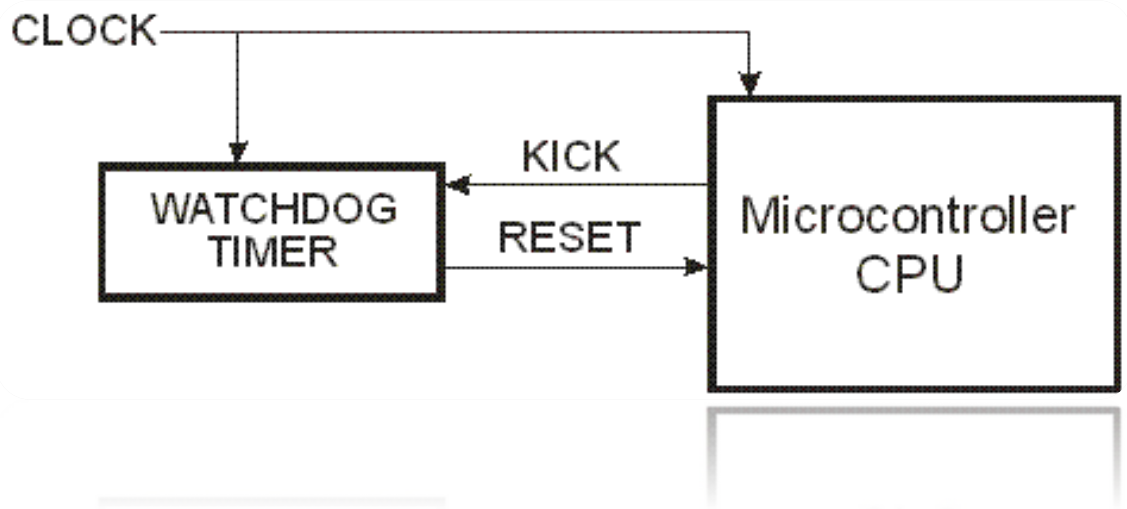
Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

What is a Watchdog Timer?



- Hardware counter that counts down from an initial value (timeout interval) to zero
  - A mechanism to detect when the processor has hung
  - Can automatically recover the system from crash or hung without intervention
- If count reaches zero, it resets the processor (not necessarily the peripherals!)
- To prevent resets, application software kicks (or pets) the watchdog
- Not a magic wand, but a very useful tool

What is a Watchdog Timer?



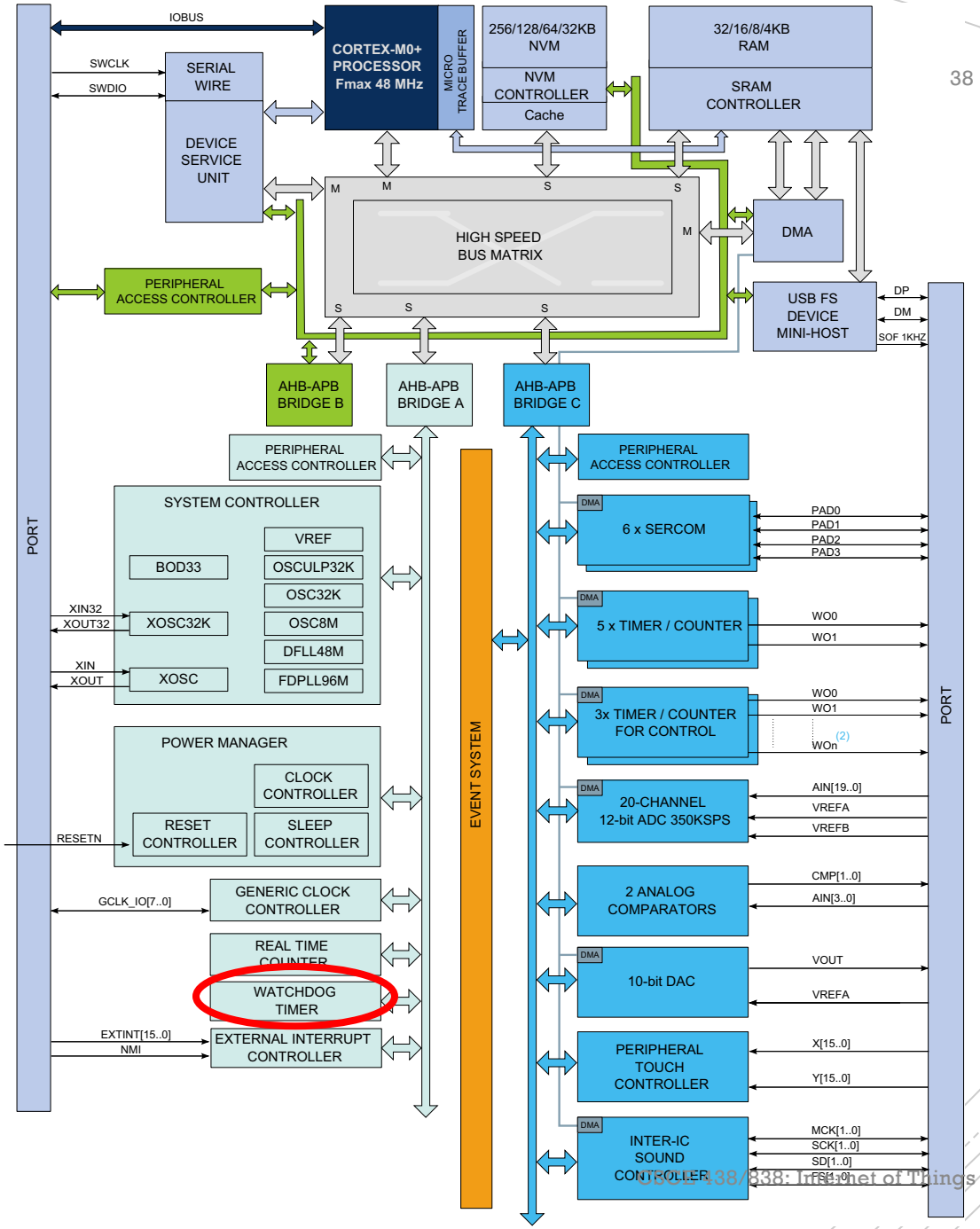
- MCU is driven by clock for processing
- Watchdog uses clock to count down
  - Independent clocks may help detect clock related faults
  - Generally connected to the same clock signal
- RESET: Signal that performs a hardware reset
- KICK: Initiated by the MCU, returns the counter to the initial value and restarts watchdog

Typical Watchdog Operation

- When system is reset, watchdog is turned off
- Necessary startup functions are performed
- MCU kicks the watchdog for the first time
- Need to kick again before it counts down to zero
  - Make sure software cannot turn off or alter watchdog once it is started

NASA recommends using a watchdog and emphasizes that it must be able to detect death of all tasks

SAMD21 Watchdog



SAMD21 Watchdog

- Watchdog Timer (WDT)
- "The Watchdog Timer (WDT) is a system function for monitoring correct program operation. It makes it possible to recover from error situations such as runaway or deadlocked code. The WDT is configured to a predefined time-out period, and is constantly running when enabled. If the WDT is not cleared within the time-out period, it will issue a system reset. An early-warning interrupt is available to indicate an upcoming watchdog time-out condition."
- "When enabled, the WDT will run in active mode and all sleep modes. It is asynchronous and runs from a CPU- independent clock source. The WDT will continue operation and issue a system reset or interrupt even if the main clocks fail."

Table 10-3. Interrupt Line Mapping

Peripheral Source	NVIC Line
EIC NMI – External Interrupt Controller	NMI
PM – Power Manager	0
SYSCTRL – System Control	1
WDT – Watchdog Timer	2
RTC – Real Time Counter	3
EIC – External Interrupt Controller	4
NVMCTRL – Non-Volatile Memory Controller	5
DMAC - Direct Memory Access Controller	6
USB - Universal Serial Bus	7
EVSYS – Event System	8
SERCOM0 – Serial Communication Interface 0	9
SERCOM1 – Serial Communication Interface 1	10
SERCOM2 – Serial Communication Interface 2	11
SERCOM3 – Serial Communication Interface 3	12
SERCOM4 – Serial Communication Interface 4	13
SERCOM5 – Serial Communication Interface 5	14
TCC0 – Timer Counter for Control 0	15
TCC1 – Timer Counter for Control 1	16
TCC2 – Timer Counter for Control 2	17

WDT Features

- Two modes of operation: Normal and window modes
- Selectable time-out periods, from 8 cycles to 16,000 cycles in normal mode or 16 cycles to 32,000 cycles in window mode
- Early Warning interrupt generation
- Asynchronous operation from dedicated oscillator

- Let's look at the datasheet

Watchdog

- Not a magic wand, but a very useful tool
- System is reset if a program takes unexpectedly long to execute
  - Expectations might be wrong
  - Faults that slow the system
  - Faults that results in a hang
- Too frequent ISRs
- Unintended infinite loop
- Corrupted data sources
- Loops that run longer than intended (memory)
- Hardware faults
- Does not help detect
  - Arithmetic errors
  - Conditional logic errors

Good Practices

- Make sure all tasks are executed between kicks
- Kick it in only one place
  - E.g., at the end of the main loop
  - RTOS: Make sure each task contributes to the kick
- Make sure a task cannot crash without tripping the watchdog
- Pick timer interval correctly
  - Too big: Room for system slow down
  - Too close to expected execution time: Occasional unnecessary resets
  - If you are not sure how long the program should take, then do not use a watchdog (or an embedded system!)
- Keep track of watchdog resets
  - Watchdog provides seamless error-prone operation
  - Try to find and log what causes the reset (LEDs, run-time error logs)

Not So Good Practices

- Do NOT kick the watchdog with a timer ISR
  - A hardware timer is used to kick the watchdog
  - The rest of your system might have died except the timer
- Do NOT turn the watchdog off after it has been turned on*
  - Turned off watchdog = No safety net
  - * Some exceptions apply

Heartbeat Timer

- Similar concept for distributed/networked systems (IoT)
- Node sends messages once in a while to tell it is alive
- Since it is run through a timer ISR, may end up with similar problems
- Nevertheless, useful to rule out communication issues


Topologically Equivalent
Topologically Distinct