Kuwait University
College of Engineering and Petroleum

جامعة الكويت
KUWAIT UNIVERSITY

ME319 MECHATRONICS

Part I: The Brains – Microcontrollers, Software and Digital Logic

Lecture 4: Microntroller Peripherals

Spring 2021

Ali AlSaibie

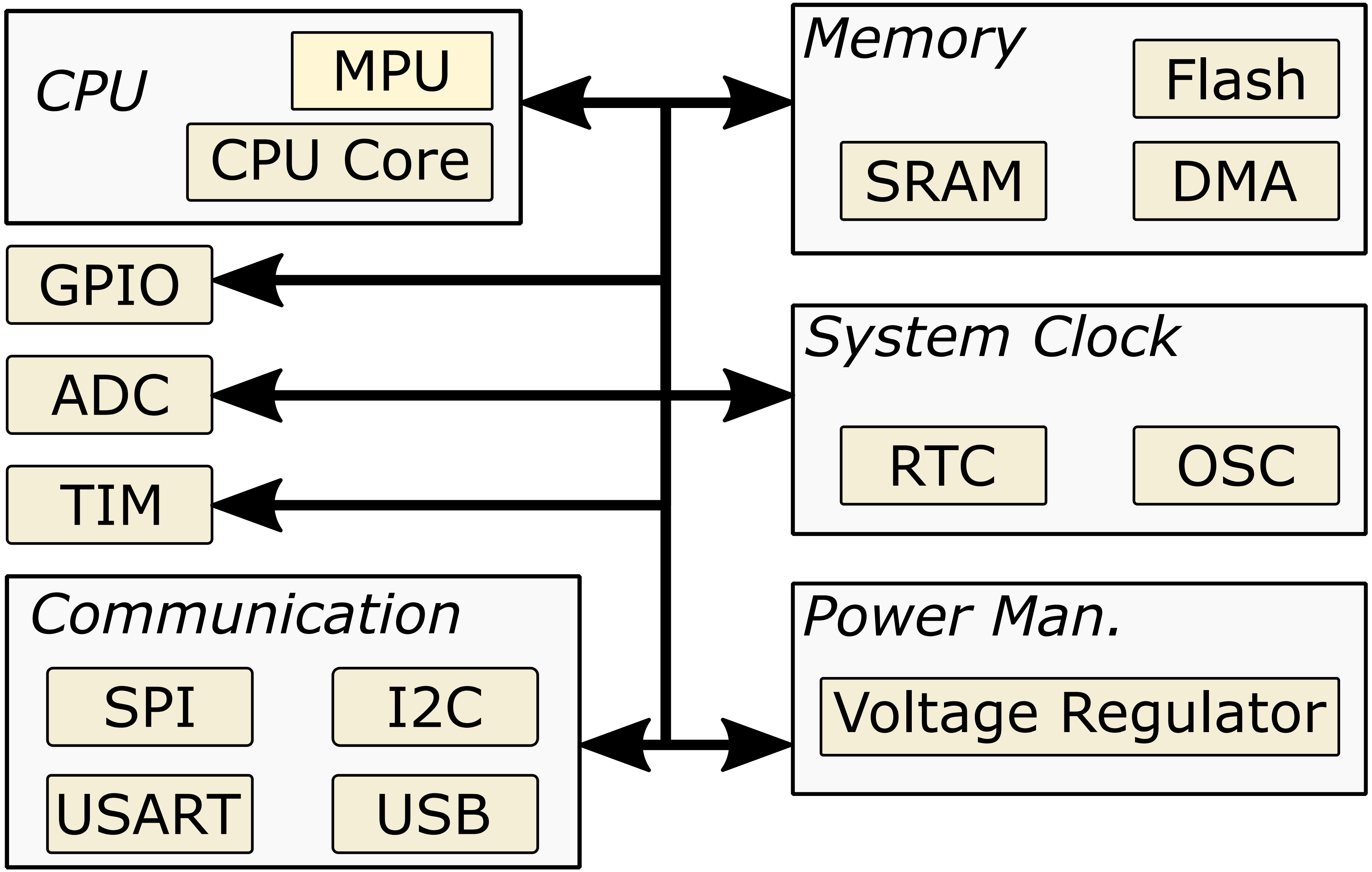
Lecture Plan

- Objectives:
  - Give an overview standard microcontroller peripherals
  - Become familiar with the STM32Nucleo and its peripherals
  - Become familiar with reading a microcontroller datasheet

What is a peripheral

- Remember that a microcontroller is more than just a microprocessor
- Subsystems within the microcontroller that are dedicated to a specific functionality are referred to as a peripheral, for example:
  - An MCU communicates with a digital accelerometer through the I2C protocol. *It will need an I2C Peripheral.*
  - An MCU reads an analog temperature sensor, the analog sensor needs to be converted to digital. *It will need an ADC Peripheral.*
  - An MCU needs to keep track of time, count how many seconds have passed between events. *It will need a Timer Peripheral.*
- There can be multiples of the same peripheral on one MCU
  - (e.g. 3 ADC peripherals)

Microcontroller Architecture



Some Common Peripherals

- **GPIO:** General Purpose Input Output
  - Basic input-output interface
- **UART**: Universal Asynchronous Receiver Transmitter
  - Serial communication protocol
- **SSI**: Synchronous Serial Interface
  - Serial communication protocol
- **JTAG**: Joint Test Action Group
  - Programming and debugging interface
- **USB**: Universal Serial Bus
  - High Speed Serial Communication
- **ADC**: Analog to Digital Conversion
  - Convert Analog Signals to Digital Values
- **I^{2}C**: Inter-Integrated Communication
  - Serial Communication protocol
- **CAN**: Controlled Area Network
  - Vehicle Communication Standard
- **Timers**: Clocked counters
  - Keep track of time within microcontroller
- **Ethernet**: Network Communication Interface
  - Local Area Network Communication Protocol
- **Analog Comparators**
  - Device that compares two voltages or current and outputs a digital signal
- **PWM**: Pulse Width Modulation
  - A pulsed digital output, used for communication and control of analog devices

جامعة الكويت
KUWAIT UNIVERSITY

Microcontroller Peripherals

- Peripherals have their own processing logic/resources, they don't compete with the CPU: *They don't share* your program execution resources*
- Let's introduce a few common peripherals at a high level…..
  - GPIO
  - ADC
  - UART
  - Timers
- We will go through each one in a bit more detail later in the course.

جامعة الكويت
KUWAIT UNIVERSITY

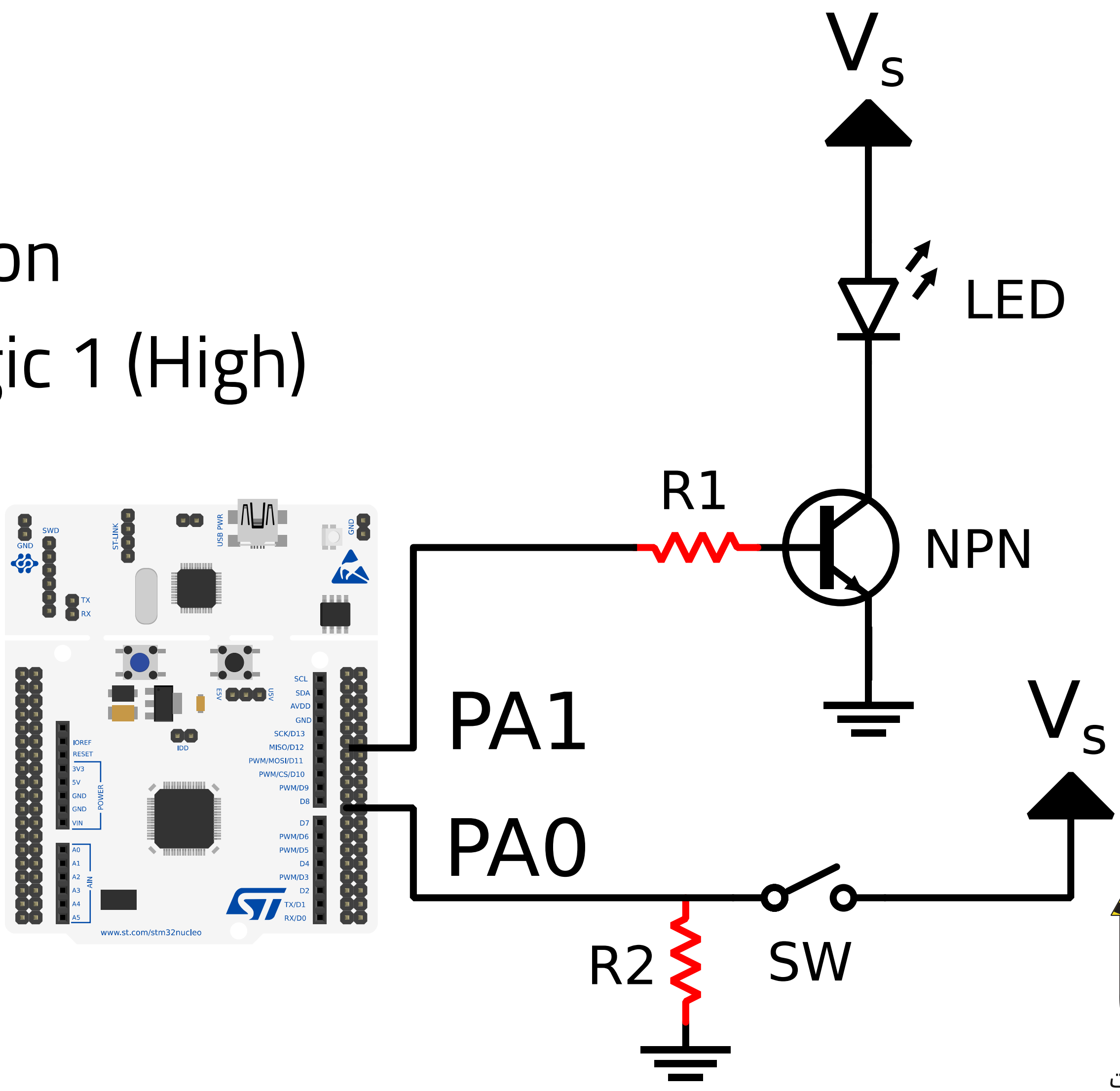
GPIO: General Purpose Input Output

- Provides most basic way to interface MCU with the outside world.
- Things you can do through GPIO (not limited to):
  - Connect a GPIO pin to a switch and read the state of switch press
  - Connect a GPIO pin to an LED and turn the LED on/off
  - Connect a GPIO pin to a relay and switch a machine on/off
  - Connect to an “interrupt” pin, to register when an external event happened.

جامعة الكويت
KUWAIT UNIVERSITY

GPIO: General Purpose Input Output - Example

- PA1 and PA0 are both configured as GPIO Pins
  - PA1 reads: *Port A Pin 1*
- PA1 Set as output and PA0 as Input
- When PA1 is High (Logic 1): LED turns on
- When Switch is pressed: PA0 reads logic 1 (High)
- The GPIO Peripheral usually has:
  - Multiple ports, which have:
    - Multiple pins

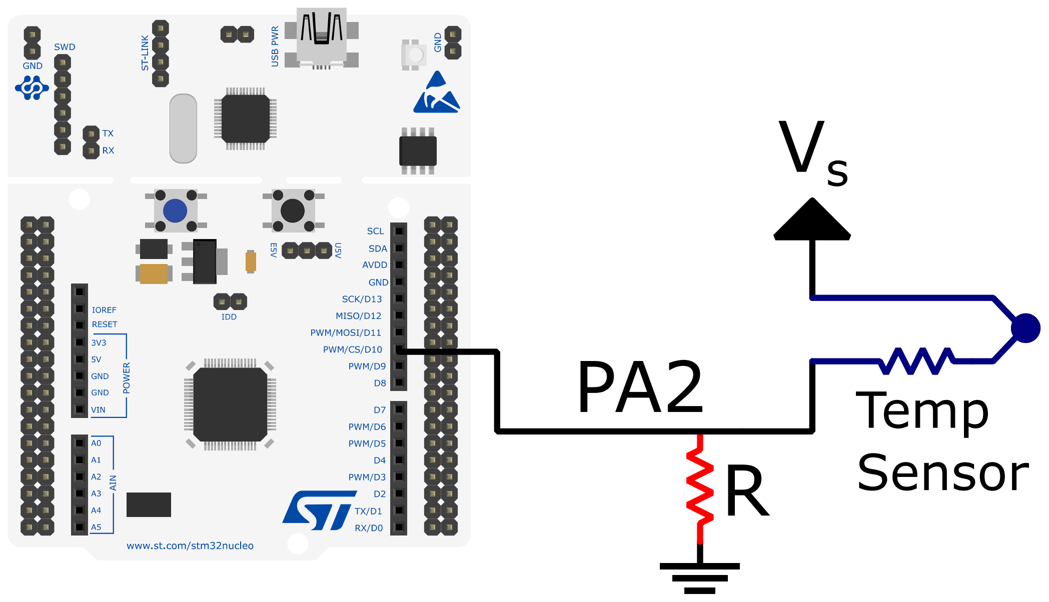


ADC: Analog-to-Digital Convertor

- An analog signal has an infinite resolution. A digital value has finite.
- An analog signal is continuous: (e.g. temperature sensor, strain-gauge scale)
- Digital signal is discrete: (e.g. High/On or Low/Off)
- Give examples of Analog/Digital Signals?
- The ADC peripheral of a microcontroller helps convert an analog **voltage** signal to a digital value. ADC peripheral comes with features, such as:
  - Resolution: $\pm 1mV\,?\, \pm 8mV\,?\, \pm 12mV\,?$
  - Number of Samples: *Output average of how many samples?*
  - Range: Expected incoming signal to be: *0-3V? 0-5V? 0-0.9V?*

ADC: Analog-to-Digital Convertor - Example

- PA2 is configured as an Analog Input Pin
- It can be configured to read the input voltage 10 times ,then
- Take the average of the 10 samples and
- Convert it into a digital number, then
- Store it in a specific memory location
- Which is accessible by the program code
- The program can check for the latest value (poll), or
- The ADC peripheral can also notify the program code (via interrupt), that a new value is ready to be read.

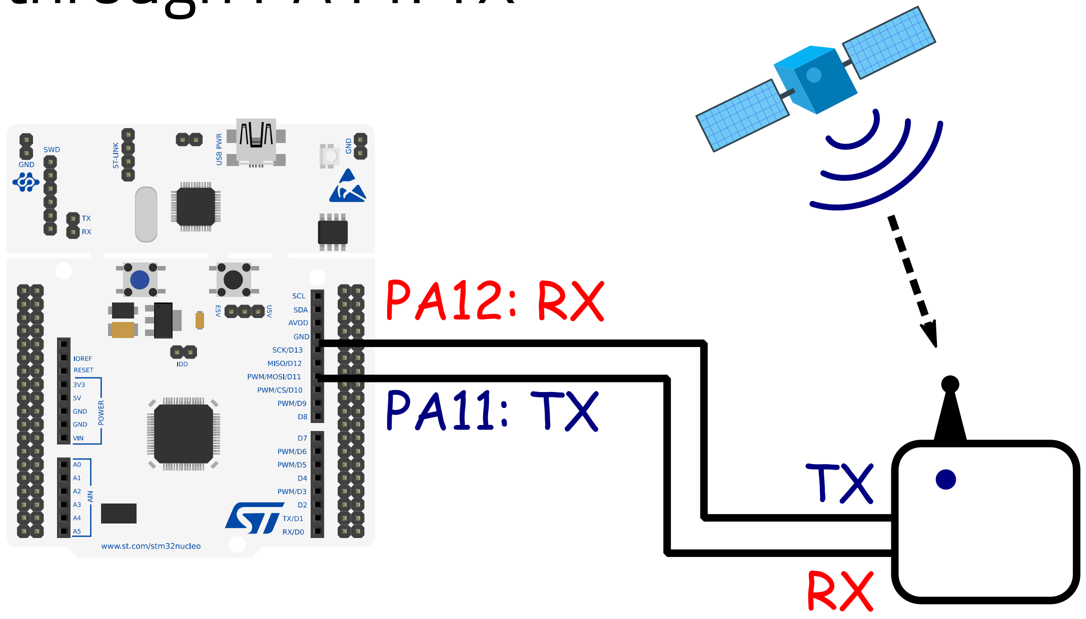


UART: Universal Asynchronous Receiver/Transmitter

- **UART** is one of several **communication protocols** used on MCUs
- Used for communication between PC and MCU (e.g. *Arduino: Serial.print()*)
  - An intermediary UART-to-USB is used
- Also used for: RF wireless communication, sensors (GPS sensors), old dial-up modems
- Asynchronous: Communication not **synchronized** between parties
- One Line for Sending (TX: Transmitter)
- One Line for Receiving (RX: Receiver)
- Modern MCUs have U**S**ART instead (S: Synchronous, they support both)

UART: Universal Asynchronous Receiver/Transmitter - Example

- PA11 is configured as a transmit, PA12 is receive
- UART is configured to use PA11/PA12 as TX/RX
- The GPS signal receives signals from visible satellites
- Converts location + time data to a series of characters, then sends through TX
- MCU receives characters through PA12: RX
- MCU can send configuration commands through PA11: TX

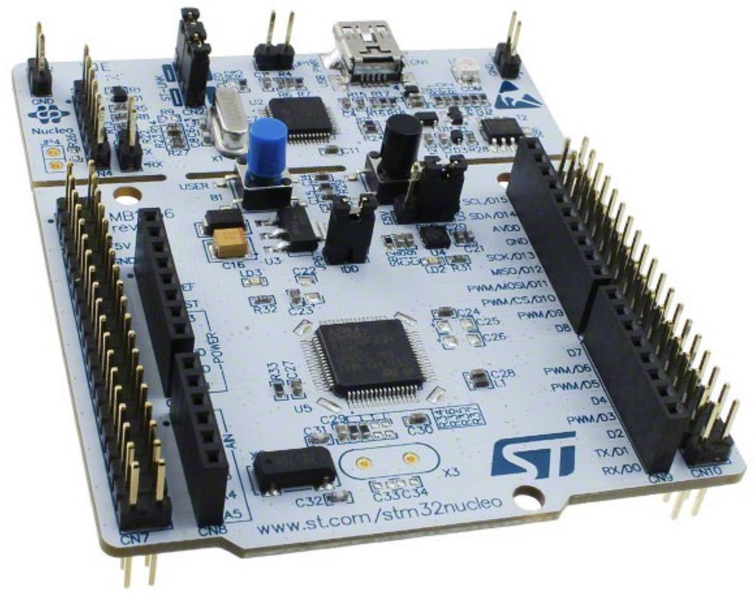


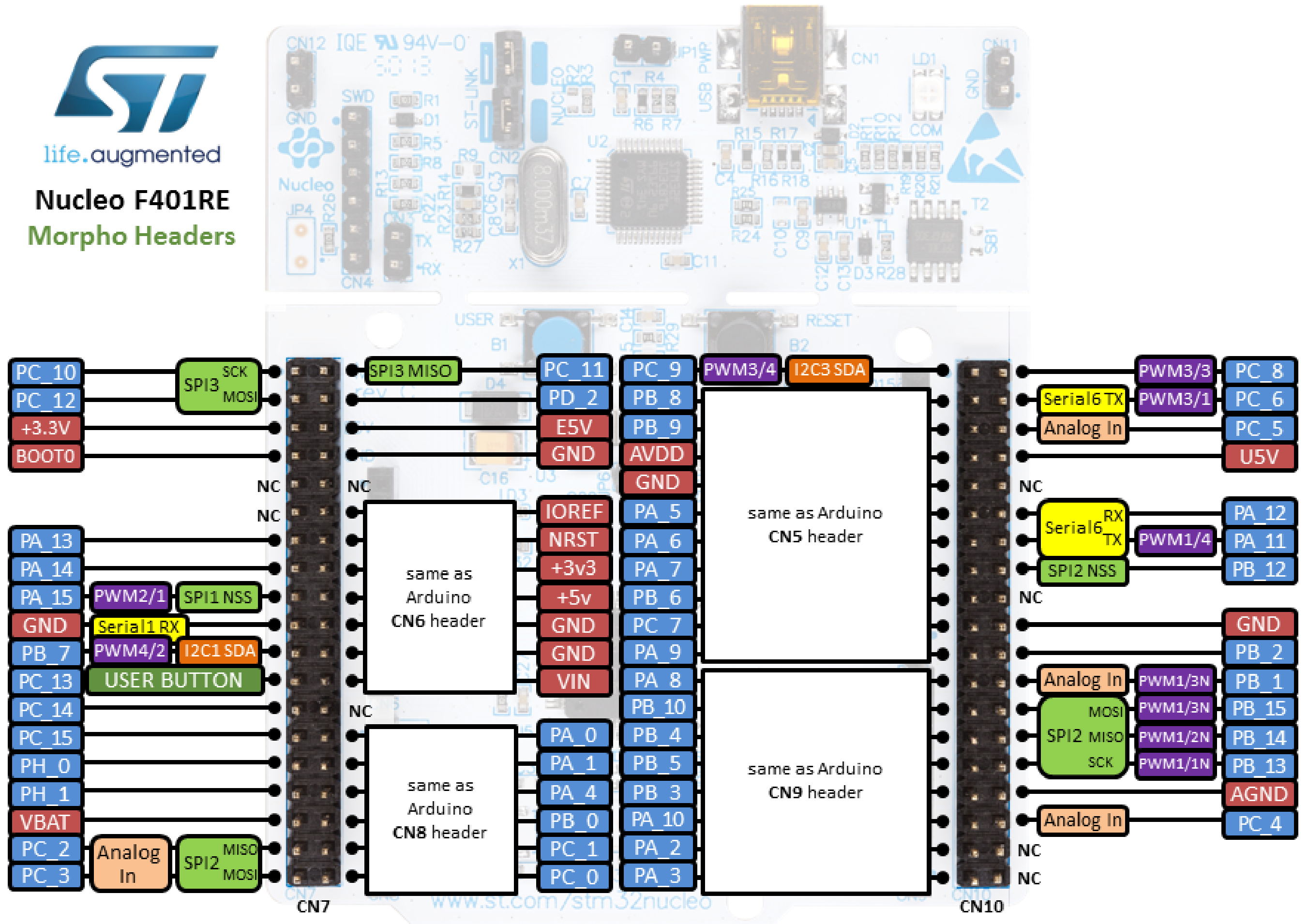
Timers

- Timers provide a way to keep track of time (surprise!).
- They keep a counter that ticks at a configured rate, this can be used to:
  - Tell the time or elapsed time.
    - *E.g. When you call delay(500); a timer peripheral is used*
  - Call a function at a specific and deterministic rate.
    - *E.g. Essential in applied control systems*
  - Generate a signal with a specific frequency & on/off time ratio
    - *E.g. Generate a square wave, PWM or PPM signal*
  - Record the time when an external or internal event occurred
    - *E.g. Register the frequency of a square wave signal*

STM32Nucleo

- We notice a few peripherals looking at the STM32Nucleo **pinout schematic**
- Such as: SPI, GPIO, PWM (TIMER), I2C, ADC, UART





Reading Datasheets

- But how do we know what peripherals are **available** on a specific MCU?
- We refer to the **datasheet**
- The microcontroller datasheet contains all the key information about all the bells and whistles that come in the microcontroller.
    - Information such as: speed, memory size, power consumption, electrical ratings, peripherals and their specifications, chip variants and their pins and physical layout, the dimension of the chip for manufacturing purposes, the functionalities of every pin, ordering information and much more.

Reading Datasheets

- The first or cover page of the datasheet usually contains a summary of the features
- An engineer can tell if a certain MCU is suitable or not, just by looking first at the datasheet cover page.
- Let’s look at the first page of the STM32F401RE microcontroller
    - STM32F401RE is the MCU on the STM32 Nucleo F401RE board we use in this course.
- A datasheet is often provided for a sub-family of microcontrollers
    - SMT32F401**x**D/E, **x** can stand for different letters/numbers representing different variants. D/E: Includes part numbers that end with D and E

STM32F401xD/E Datasheet

STM32F401xD STM32F401xE

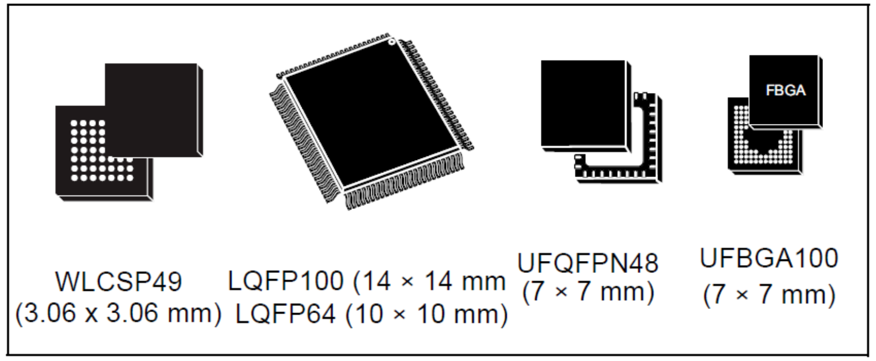
life.augmented

ARM® Cortex®-M4 32b MCU+FPU, 105 DMIPS, 512KB Flash/96KB RAM, 11 TIMs, 1 ADC, 11 comm. interfaces

Datasheet - production data

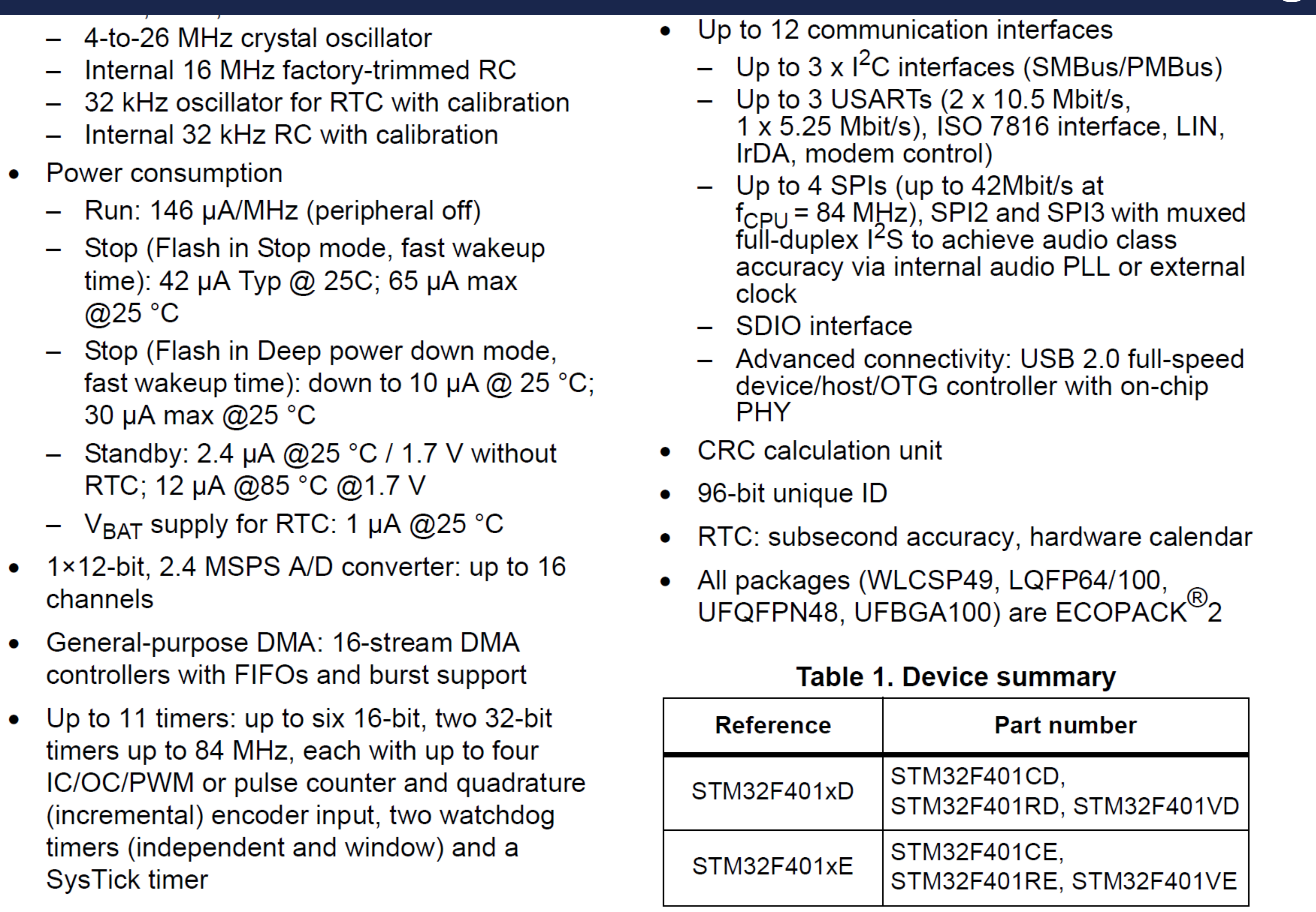
Features

- Core: ARM® 32-bit Cortex®-M4 CPU with FPU, Adaptive real-time accelerator (ART Accelerator™) allowing 0-wait state execution from Flash memory, frequency up to 84 MHz, memory protection unit, 105 DMIPS/1.25 DMIPS/MHz (Dhrystone 2.1), and DSP instructions
- Memories
  - up to 512 Kbytes of Flash memory
  - up to 96 Kbytes of SRAM
- Clock, reset and supply management
  - 1.7 V to 3.6 V application supply and I/Os
  - POR, PDR, PVD and BOR
  - 4-to-26 MHz crystal oscillator



- Debug mode
  - Serial wire debug (SWD) & JTAG interfaces
  - Cortex®-M4 Embedded Trace Macrocell™
- Up to 81 I/O ports with interrupt capability
  - Up to 78 fast I/Os up to 42 MHz
  - All I/O ports are 5 V-tolerant
- Up to 12 communication interfaces

STM32F401xD/E Datasheet (MCU on the STM32Nucleo) – Page 1



- 4-to-26 MHz crystal oscillator
- Internal 16 MHz factory-trimmed RC
- 32 kHz oscillator for RTC with calibration
- Internal 32 kHz RC with calibration

- Power consumption
  - Run: 146 µA/MHz (peripheral off)
  - Stop (Flash in Stop mode, fast wakeup time): 42 µA Typ @ 25C; 65 µA max @25 °C
  - Stop (Flash in Deep power down mode, fast wakeup time): down to 10 µA @ 25 °C; 30 µA max @25 °C
  - Standby: 2.4 µA @25 °C / 1.7 V without RTC; 12 µA @85 °C @1.7 V
  - V_{BAT} supply for RTC: 1 µA @25 °C
- 1×12-bit, 2.4 MSPS A/D converter: up to 16 channels
- General-purpose DMA: 16-stream DMA controllers with FIFOs and burst support
- Up to 11 timers: up to six 16-bit, two 32-bit timers up to 84 MHz, each with up to four IC/OC/PWM or pulse counter and quadrature (incremental) encoder input, two watchdog timers (independent and window) and a SysTick timer
- Up to 12 communication interfaces
  - Up to 3 x I^2C interfaces (SMBus/PMBus)
  - Up to 3 USARTs (2 x 10.5 Mbit/s, 1 x 5.25 Mbit/s), ISO 7816 interface, LIN, IrDA, modem control)
  - Up to 4 SPIs (up to 42Mbit/s at f_{CPU} = 84 MHz), SPI2 and SPI3 with muxed full-duplex I^2S to achieve audio class accuracy via internal audio PLL or external clock
  - SDIO interface
  - Advanced connectivity: USB 2.0 full-speed device/host/OTG controller with on-chip PHY
- CRC calculation unit
- 96-bit unique ID
- RTC: subsecond accuracy, hardware calendar
- All packages (WLCSP49, LQFP64/100, UFQFPN48, UFBGA100) are ECOPACK®2

Table 1. Device summary

Reference	Part number
STM32F401xD	STM32F401CD, STM32F401RD, STM32F401VD
STM32F401xE	STM32F401CE, STM32F401RE, STM32F401VE