Internet of Things

IoT Development Boards
(IoT Hardware Platforms)

Mehdi Rasti
Amirkabir University of Technology

2021

CONTENTS

- Smart Objects (Things)
- Development Boards (Hardware Platforms) for IoT
  - Microcontroller Development Boards (System on a Chip (SoC))
  - Single-Board Computers (SBC)
  - How to Choose the Right Development Board for IoT Projects
- IoT Operating Systems
- Power Consumptions Considerations

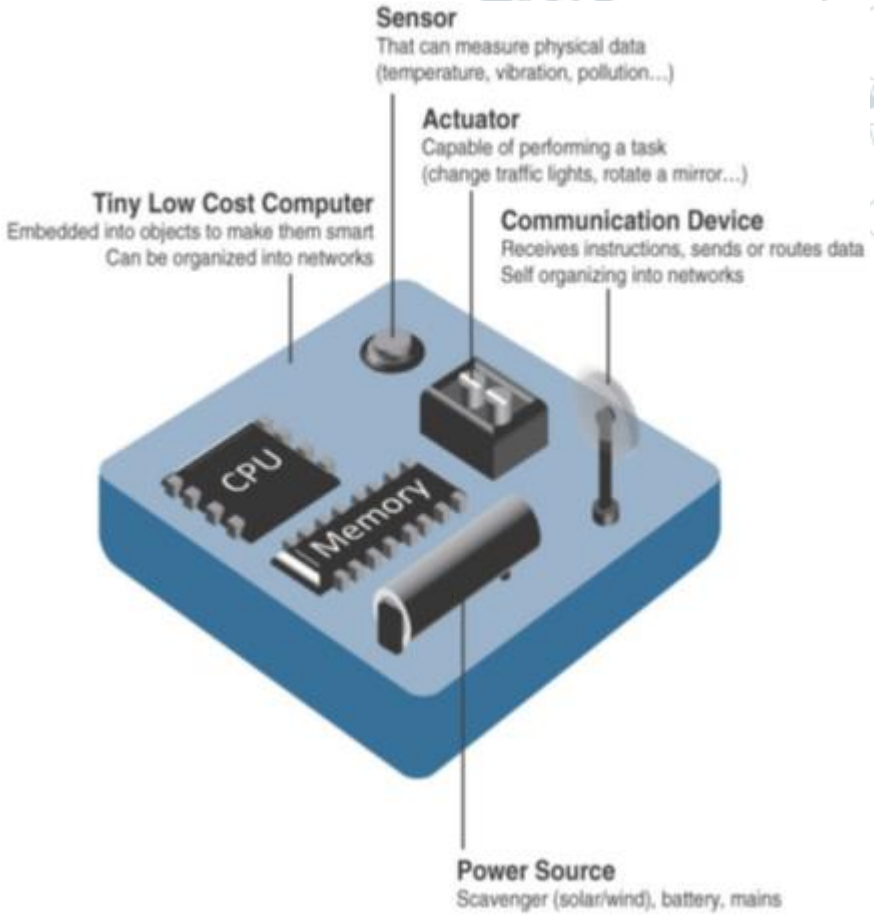
CONTENTS

- Smart Objects (Things)
- Development Boards (Hardware Platforms) for IoT
  - Microcontroller Development Boards (System on a Chip (SoC))
  - Single-Board Computers (SBC)
  - How to Choose the Right Development Board for IoT Projects
- IoT Operating Systems
- Power Consumptions Considerations

Smart Objects

- **Interchangeably with terms such as**
  - smart sensor, smart device, IoT device, intelligent device, thing, smart thing, intelligent node, intelligent thing, ubiquitous thing, and intelligent product

- ***A* smart object has at a minimum, the following four defining characteristics**
  1. Processing unit
  2. Sensor(s) and/or actuator(s)
  3. Communication device
  4. Power source



Smart Objects

Sensors

Touch Sensor

Ultrasonic Sensor

PIR Sensor

Speed Sensor

Temperature Sensor

Actuators

Motor

Buzzer

Heater

Pump

Lights

Trends in Smart Objects

- Size is decreasing
- Power consumption is decreasing
- Process power is increasing
- Communication capabilities is improving
- Communication is being increasingly standardized

Communication Patterns of Smart Objects

- Connected smart objects have one of the following two communication patterns:
  - Event-driven: Transmission of sensory information is triggered only when a smart object detects a particular event or predetermined threshold
  - Periodic: Transmission of sensory information occurs only at periodic intervals

Hardware for IoT Project

- Hardware and software components are designed for IoT applications via a standard design protocol:
  - specification development,
  - conceptual design,
  - prototype,
  - test, and
  - rollout of hardware and software integrated into a network.

* https://developer.ibm.com/technologies/iot/articles/iot-lp101-best-hardware-devices-iot-project/

Hardware for IoT Project

- Some platforms, such as Arduino and Raspberry Pi, may expedite design and allow rapid prototyping without involved customization, thus expediting the time required to implement an IoT configuration.

- Design will require identification of performance requirements, the necessary hardware and software to achieve such requirements, followed by specifications for components – either commercially off-the-shelf (COTS) or customized designs – with due consideration for the operating environment and the application to be used within it.

* https://developer.ibm.com/technologies/iot/articles/iot-lp101-best-hardware-devices-iot-project/

IoT Device Characteristics

- IoT Device: Acquire data by some sensor of the system, process it in a controller unit and store/transmit some information through a provided wireless or wired connectivity,
- IoT devices (smart objects) may be characterized by capabilities:
  - Data acquisition and control
  - Data processing and storage
  - Connectivity
  - Power management

Data Acquisition and Control

- *Data acquisition* (DAQ) gathers analog information at a fixed time interval (the data sample rate), and transmits it as a digital signal to a remote output device for a digital readout.

- DAQ may include signal conditioning (to manipulate and scale raw sensor readings), and analog-to-digital converters to convert the analog sensor readings into digital values so that they can be processed and analyzed.

- Data acquisition and control are performed by Sensors and Actuators

Data Processing and Storage

- IoT devices require specific data processing and storage capability. This helps achieve
  - data aggregation,
  - data transmission, and
  - Data analysis.
- Some IoT devices may process data directly, while others transmit this data to other devices, gateway devices, or cloud applications for further aggregation and analysis.

Data Processing and Storage

- Data processing and storage
  - The processing power and storage used by an IoT application depends on processing required by the services or apps that consume the data.
  - Available memory and processor specifications, clock speed, and number of cores, all of which subsequently determine the devices rate of data processing.
  - The capacity of the non-volatile flash, which is used to persist data until transmitted upstream, determines how much data can be stored on the device.
  - Devices performing edge analytics require substantially more processing capabilities than devices that perform only basic data processing like validating, normalizing, scaling, or converting readings

Connectivity

- Some devices communicate using:
  - Wireless communications:
    - by using 802.11 (Wi-Fi), Bluetooth, RFID,
    - cellular networks, or Low Power wide area network (LPWAN) technologies like LoRa, SigFox or NB-IoT.
  - Wired communication
    - is suited to stationary devices.
      - Such devices may be installed in smart buildings, home automation, and industrial control applications, for example, and connected with Ethernet, or retrofitted with Ethernet over power.
    - Serial communication is a form of wired connectivity between devices, using standard protocols like *Universal Asynchronous Receiver Transmitter* (UART), or the *Controller Area Network* (CAN) protocol, which has its origins in the automotive industry.

Power Management

- Power management is a critical factor for portable and wearable IoT devices that rely on a wireless power source, such as batteries or photovoltaic cells (solar).

- Depending on usage patterns and the power requirements of the attached sensors, actuators, or integrated circuits (ICs), a device may be put into sleep mode or into low-power mode periodically to conserve power.

- For example, a single-board computer like the Raspberry Pi 4 requires around 700 – 1000mA of current to operate under typical usage. If you were transmitting data constantly over a wifi network, or if the device under heavy load was performing a lot of data processing, the power usage would be high, but then it would drop when the device became idle.

- If you connect a camera module, the required amperage increases by about 250mA when the camera is in use. Also, sensors normally require power to operate; the GPIO pins on the Raspberry Pi supply 3.3V or 5V, up to a total of 50mA current across all of the pins. The power consumption of the device increases as you increase the number of components that are attached to the pins.

CONTENTS

- Smart Objects (Things)
- Development Boards (Hardware Platforms) for IoT
  - Microcontroller Development Boards (System on a Chip (SoC))
  - Single-Board Computers (SBC)
  - How to Choose the Right Development Board for IoT Projects
- IoT Operating Systems
- Power Consumptions Considerations

IoT System Design

1. Identification of Requirements
2. Design of the System
3. System Development and Prototyping
4. System Testing
5. Implementation and Results

IoT Hardware for Prototyping

- Developing IoT applications is more accessible with the growing availability of low-cost, commercially available off-the-shelf hardware development boards, platforms, and prototyping kits.
- Prototyping hardware is optimized for:
  - **Flexibility**: with a greater selection of components, designers may substitute new sensors with different specifications.
  - **Affordability** – low cost
  - **Modularity** – compatibility with other hardware ecosystems. You can independently upgrade the networking, data processing, or storage modules of a device for evolving requirements.
  - **Ease of use** – can be setup in minutes and comes with tools
  - **Beginner audiences** – content and examples optimized for beginners

IoT Hardware for Prototyping

- IoT hardware development platforms are basically kits or prebuilt development boards that combine microcontrollers and processors with wireless communication chips and other components in a ready-to-build and ready-to-program bundle.

- They come indifferent configurations and include a variety of peripherals for connecting sensors and interfacing with other hardware components or devices for designing and prototyping IoT devices.

IoT Hardware for Prototyping

- IoT boards, also known as development boards or prototyping boards, are basically hardware platforms that are commonly used to build prototypes of manufacturer's ideas.
- There are dozens of platforms with a diverse choice of hardware, support, security, development infrastructure and communities. In this chapter, we'll focus on some popular platforms and try to figure out the perfect matches for different IoT projects.

* Analysis of Single-Board Computers for IoT and IIoT Solutions in Embedded Control Systems, Pavlo Galkin, Lydmila Golovkina, Igor, 2018 International Scientific-Practical Conference, 2018

IoT Hardware: Types of Off-the-Shelf Hardware for Prototyping IoT Projects

- Two main types of prototyping and developments boards:
  - Microcontroller-based boards or *System-on-a-Chip* (SoC)
    - SoCs bundle capabilities such as data processing, storage, and networking onto a single chip
  - Single Board Computers

* https://developer.ibm.com/technologies/iot/articles/iot-lp101-best-hardware-devices-iot-project/

Most Important Features of IoT Development Hardware Platforms:

- Processing and Memory/storage capacity;
- Connectivity
- Onboard sensors
- Power consumption,
- Size
- Cost;
- Operating Systems (OSes) and programming languages;
- Fexibility/customizability of peripherals of hardware platforms;
- Hardware security features.

IoT Hardware- *System-on-a-Chip* (SoC)

- Microcontroller development boards are PCBs with additional circuitry to support the microcontroller to make it more convenient to prototype with and program the chip.
- Microcontroller based boards or *System-on-a-Chip* (SoC)
  - provides data processing and storage capabilities
  - Contains
    - a processor core (or cores),
    - memory (RAM), and
    - *erasable programmable read-only memory* (EPROM) for storing the custom programs that run on the microcontroller.
  - Sensors and actuators connect to the SoC through digital or analog *General Purpose Input/Output* (GPIO) pins or through a hardware bus.
  - Standard communication protocols like I2C and SPI are used for intra-device communication with the components that are connected with the bus. Adopting standards makes it easier to add or swap out components that are connected with the bus.

* https://developer.ibm.com/technologies/iot/articles/iot-lp101-best-hardware-devices-iot-project/

IoT Hardwares- System-on-a-Chip (SoC)

- Pros of Using a SoC
  - **Size** : You get a lot of functions and features in a small package
  - **Flexibility**: In terms of board size, form factor, and power, it's hard to beat the flexibility that a SoC allows in a design
  - **Cost efficient** : This is especially true for application-specific SoCs such as video codecs, since the alternative is to implement it in software, which can be costly in terms of time and effort
  - **High volume** : SoCs are great if you have a high volume product since it makes it easier to justify the engineering resources and cost

* https://www.semiconductorstore.com/blog/2015/System-on-Chip-vs-Single-Board-Computer-A-Comparison-Guide/689/

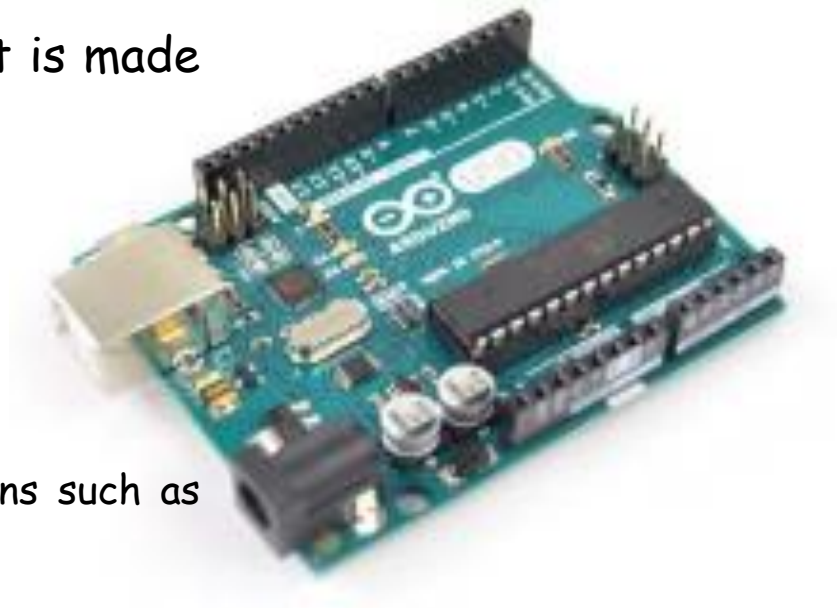
IoT Hardwares- System-on-a-Chip (SoC)

- Cons of Using a SoC
  - **Time to market** : Design cycles are typically between 6-12 months for a SoC
  - **Resource limitations** : If you are limited in resources or if you don't have too much expertise in the field, a SoC is probably not a good fit
  - **Lower volume** : If you are designing a low volume/high margin type of product, there may be alternatives that are more suitable, unless you need a highly specialized hardware. You may be better off leveraging hardware from someone else and utilizing your time and resources for application software

* https://www.semiconductorstore.com/blog/2015/System-on-Chip-vs-Single-Board-Computer-A-Comparison-Guide/689/

Arduino As A SoC

- The Arduino is basically an open-source prototyping platform that is made up of two essential parts:
  - The hardware: The Arduino board with different versions
    - Arduino Uno,
    - Arduino YUN with enabled WiFi connectivity and
    - Arduino MKR family that offers multiple wireless connectivity options such as WiFi, Bluetooth, LoRa, SigFox and Narrowband IoT
  - The software:
    - IDE (integrated development environment) and
    - Recently released Pro IDE for easier and faster coding.
      - The platform has a well-established community, online software tools, various development kits, Arduino IoT Cloud and other resources for building connected devices.



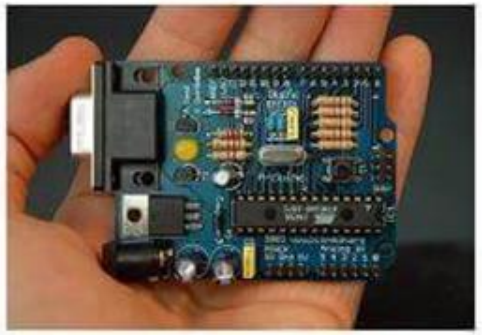
Arduino As A SoC

- The Arduino board is a freely available open source development microcontroller capable to cope up with a variety of communication protocols that is a must to be usable for any kind of IoT device.
- This board is cheap and feature rich with availability of a variety of daughter boards that have an amazing stacking feature to the main mother board.
- The availability of Wi-Fi and Ethernet shield along with the low power BLE-4 Arduino shield makes it suitable for rapid prototyping and programming with ease.
- The easy to use and abundant example programs in the Arduino IDE makes it simple for the user to get started pretty quickly in the process of making IoT device work seamlessly in all kind of environments.

Arduino- Different Versions

- Uno (releasedin2010),
- Mega2560 (releasedin2010),
- Due (released in 2012),
- Yún (released in 2013),
- Arduino/Genuino 101 (release in 2015),
- Arduino/Genuino MKR1000 (released in 2016).

Arduino



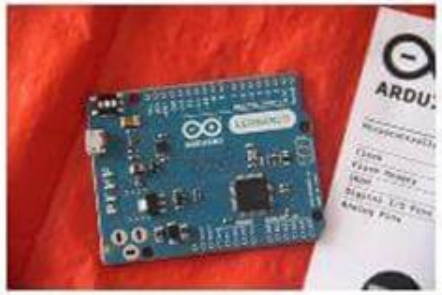
Arduino RS232[32]
(male pins)

Arduino Diecimila[33]

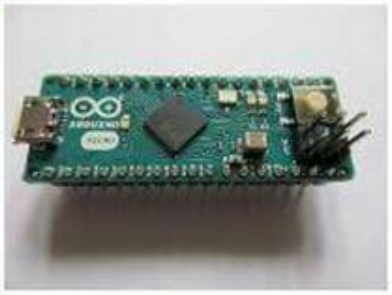
Arduino Duemilanove[34]
(rev 2009b)

Arduino Uno R2[35][36]

Arduino Uno SMD
R3[37]



Arduino Leonardo[38]



Arduino micro (AtMega
32U4)

Arduino pro micro (AtMega32U4)

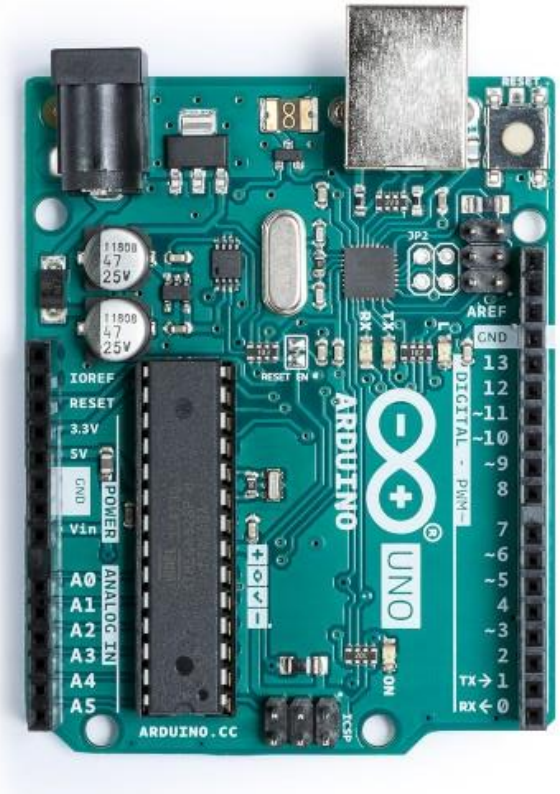
Arduino Pro[39]
(No USB)

Arduino Mega[40]

Arduino
Nano[41]
(DIP-30
footprint)

Arduino Uno Rev3

- Highlighted Features
  - Microcontroller ATmega328P
  - Operating Voltage 5V
  - Digital I/O Pins 14 (of which 6 provide PWM output)
  - 20mA DC current per I/O pin
  - Powered via USB connection or with external power



* https://www.curtisswrightds.com/news/blog/choosing-a-single-board-computer.html

Arduino-Processing and Memory/Storage Capacity

- Heart of every Arduino board: Microcontroller Unit (MCU); a type of Integrated Circuit (IC) with a processor, embedded memory, and programmable I/O peripheral devices
  - Whether 8, 16, or 32-bit MCU, the processor clock rate or clocks peed, measured in Megahertz (MHz) or Gigahertz (GHz), determines how many instructions per second the MCU can execute.
  - The 8, 16, or 32-bit refer to the size of the registers, which are high-speed memory areas on the processor that hold instructions and data currently being processed.
  - Larger number of registers enables the processor to simultaneously handle larger memory areas
  - Thus, the bigger the size of the registers, the faster the MCU can process a set of data.

* Internet of Things A to Z : Technologies and Applications, IEEE Press, 2018, (chapter 6: **(A Tutorial Introduction to IoT Design and Prototyping with Examples)**

Arduino-Processing and Memory/Storage Capacity

- The embedded memory on an MCU consists of a
  - Random Access Memory (RAM) for volatile data storage,
  - A flash memory for storing program code and constants, and
  - An Electrically Erasable Programmable Read-Only Memory (EEPROM) for storing persistent data, such as system configurations or other data that ensures normal operations upon reboot.

Arduino-Processing and Memory/Storage Capacity

- Summary of processing and storage capacity of Arduino Boards.

	Microcontroller			Memory/storage		
Board name	MCU chip	Processor bit	Clock speed	RAM	EEPROM (kb)	Flash memory (kb)
Uno	ATmega328	8	16 MHz	2 KB	1	32
Mega 2560	ATmega2560	8	16 MHZ	8 KB	4	250
Due	AT91SAM3X8E	32	84 MHz	96 KB	–	512
Yún	ATmega32U4	8	16 MHz	2.5 KB	1	32
	AR9331	8	16 MHZ	64 MB, 2.5 KB	1	16
101	Intel Quark SE SoC	32	32 MHz	24 KB	–	196
MKR1000	ATSAMW25 SoC	32	48 MHz, 32.768 KHz	32 KB	–	256

* Internet of Things A to Z : Technologies and Applications, IEEE Press, 2018, (chapter 6: **(A Tutorial Introduction to IoT Design and Prototyping with Examples)**

Arduino- Power Consumption

- A Key requirement for IoT devices is the ability to consume very low power while maintaining an acceptable level of performance, which could enable them to run on batteries for long periods of time.

- Strategies for achieving ultralow-power operation in IoT devices include, among others,
  - employing low-power communication technologies
  - implementing low-duty-cycle operations.

- The battery life (in hours) of an IoT device can be calculated by dividing the capacity of the battery (in milliamp-hour (mAh)) by the average current consumption (in milliampere (mA)) of the device.

- The power consumption or battery life of a hardware development board will largely depend on the operating voltage and and operating current of the board as well as on the components that are connected to it.

* Internet of Things A to Z : Technologies and Applications, IEEE Press, 2018, (chapter 6: **(A Tutorial Introduction to IoT Design and Prototyping with Examples)**

Arduino- Power Consumption

- The voltage and current specifications of the Uno and the Mega 2560 are the same:
  - Their operating voltage is 5V, DC current per I/O pin is 20mA, and DC current for 3.3V pin is 50mA.
- Hence, using 50mA (0.05A) and 5V as the operating current and voltage, respectively, yields 0.05*5= 250 mW
- The average power consumption of both the Uno and the Mega2560 is 250mW

* Internet of Things A to Z : Technologies and Applications, IEEE Press, 2018, (chapter 6: **(A Tutorial Introduction to IoT Design and Prototyping with Examples)**

Arduino- Power Consumption

- The Due has different specifications, its operating voltage is 3.3V, DC current for 3.3V pin is 800mA, and DC current for 5V pin is 800mA, hence its average power consumption is 2.64W.
- The voltage and current specifications of the ATmega32U 4MCU on the Yún are 5V, DC current per I/O pin is 40mA, and DC current for 3.3V pin is 50mA,
- The operating voltage on the Atheros AR9331 processor is 3.3V
- Since the Yún has built-in Ethernet and Wi-Fi support, the operating current can be more than 200mA, therefore the average power consumption can be >1W.

* Internet of Things A to Z : Technologies and Applications, IEEE Press, 2018, (chapter 6: **(A Tutorial Introduction to IoT Design and Prototyping with Examples)**

Arduino- Power Consumption

- Although the Arduino/Genuino 101 uses a low-power Intel Curie SoC with a 20mA DC current per I/O pin, the DC current for 3.3V pin (i.e., operating current) is not specified.
- Considering that the board has a Bluetooth LE functionality, the current consumption of the board cannot be 20mA, hence its average power consumption cannot be estimated correctly from the given specifications.
- The low-power capability of the Arduino/Genuino MKR1000 MCU makes the DC current per I/O pin to be 7mA at 3.3V operating voltage.

* Internet of Things A to Z : Technologies and Applications, IEEE Press, 2018, (chapter 6: **(A Tutorial Introduction to IoT Design and Prototyping with Examples)**

Arduino- Power Consumption

- While the current consumption of the MCU is about 20mA, the Wi-Fi alone can consume about 100mA or more when operational
- Thus, for IoT applications, the total MKR1000 average current consumption can be 120mA or more, which implies an average power consumption larger than (3.3*120) 396mW.

* Internet of Things A to Z : Technologies and Applications, IEEE Press, 2018, (chapter 6: **(A Tutorial Introduction to IoT Design and Prototyping with Examples)**

Arduino- Summary of power consumption, size, and cost

Board name	Average power consumption	Size (mm)	Cost ($)
Uno	250 mW	68.8 × 53.4	22.19
Mega2560	250 mW	101.5 × 53.3	38.83
Due	2.64 W	101.5 × 53.3	39.93
Yún	≥1 W	73 × 53	64.90
101	–	68.6 × 53.4	30.00
MKR1000	≥ 396 mW	65 × 25 × 6	34.38

* Internet of Things A to Z : Technologies and Applications, IEEE Press, 2018, (chapter 6: **(A Tutorial Introduction to IoT Design and Prototyping with Examples)**

Arduino- Operating Systems and Programming Languages

- Compared to standard computers, tablets, and smartphones, many IoT devices are resource constrained, especially in terms of memory footprint.
- Hence they cannot run High-Level Operating System (HLOS) such as Windows and Linux.
- In addition, the diversity of MCU families and architectures (e.g., there are 8-bit, 16-bit, and32-bit processors) is one of the greatest bottlenecks to the development of generic Oses for these devices.
- The above mentioned diversity is also aggravating the restrictions on providing a more generic Os support for IoT heterogeneous hardware.

* Internet of Things A to Z : Technologies and Applications, IEEE Press, 2018, (chapter 6: **(A Tutorial Introduction to IoT Design and Prototyping with Examples)**

Arduino- Operating Systems and Programming Languages

- Owing to stringent resource restrictions, particularly low RAM, the Yún is the only Arduino board under consideration that supports an OS.
  - The onboard Atheros AR9331 processor supports the OpenWrt Linux distribution.
- Processes/threads in the other models of Arduino are managed in the program. However, a number developers/people in the Arduino community have been exploring possibilities for developing portable OSes for many of the Arduino boards.

Arduino- Programming Languages

- The Arduino programming language or Arduino Language (AL) is based on C/C++.
  - AL is composed of a set of C/C++ functions that users can easily call from their code (also known as sketch).
  - While virtually all support libraries are subset of C standard library and not C++ standard library, the Arduino IDE basically uses a simplified version of C++ language.
  - The IDE, which can be downloaded from the Arduino website, is used for writing and uploading the programs to the Arduino board.
  - Another alternative is to use the online IDE (Arduino Web Editor), which allows users to save their sketches in the Cloud.
  - Although Arduino boards cannot be programmed directly in Java Script (JS), it is possible to leverage the web client scripting functionalities of JS using both Firm at a and Johnny-Five libraries.
  - This is particularly important for IoT applications, especially considering how JS is recently gaining more and more popularity among IoT developers and designers as a result of the appearance of Node.JS.

* Internet of Things A to Z : Technologies and Applications, IEEE Press, 2018, (chapter 6: **(A Tutorial Introduction to IoT Design and Prototyping with Examples)**

Arduino- Connectivity and I/O Complements

- Thea bility to communicate with other devices, especially through the Internet is a very essential feature of an IoT hardware development platform.

- Similarly, the availability and accessibility of both low-level (I/O pins) and high-level (hardware communication interfaces) peripherals are important factors that determine the types of projects that can be built with a particular hardware platform.

* Internet of Things A to Z : Technologies and Applications, IEEE Press, 2018, (chapter 6: **(A Tutorial Introduction to IoT Design and Prototyping with Examples)**

Arduino- Connectivity and I/O Complements

- Among the Arduino boards, only the Yún and MKR1000 can directly connect to the Internet without using a shield (an Arduino shield is a compatible board that can be plugged on top of an Arduino board for the purpose of extending its capabilities).
- The Yún has both Ethernet and Wi-Fi support, while the MKR1000 can only connect to the Internet using the WINC1500, a low power 2.4GHz IEEE 802.11b/g/n Wi-Fi, which is the second block of the ATSAMW25 SoCon the MKR1000. The other boards can connect to the Internet using Ethernet, Wi-Fi, or GSM shields.
- The Arduino/Genuino 101 has Bluetooth Low Energy (BLE) capabilities.

* Internet of Things A to Z : Technologies and Applications, IEEE Press, 2018, (chapter 6: **(A Tutorial Introduction to IoT Design and Prototyping with Examples)**

Arduino- Connectivity and I/O Complements

- Despite their versatility and flexibility, Arduino boards cannot be used for general purpose computing. The flexibility of an Arduino board lies in the availability of the peripherals on the MCU as well as their accessibility by users via the program.
- Virtually all Arduino boards have serial communication
- interfaces like Universal Serial Bus (USB) port. The USB port can be used to power an Arduino board from a computer; it is also used for uploading programs from the IDE to the board. All the Arduino boards feature a number of digital I/ O pins that are used as low-level peripherals, which are also known as General Purpose I/O (GPIO) pins.
- The analog pins on the Arduino boards also have all the functionality of GPIO pins.

* Internet of Things A to Z : Technologies and Applications, IEEE Press, 2018, (chapter 6: **(A Tutorial Introduction to IoT Design and Prototyping with Examples)**

Arduino- Connectivity and I/O Complements

- In addition to the digital I/O pins, the Arduino boards support other hardware communication interfaces, such as Serial Peripheral Interface (SPI) communication using the SPI library as well as Universal Asynchronous Receiver/ Transmitter (UART) communication.
- They also support Inter-Integrated Circuit/Two Wire Interface (I2C/TWI) communication using the wire library, and apart from the MKR1000, all the other boards have the In-Circuit Serial Programming (ICSP) header.

Arduino- Connectivity and I/O Complements

- Summary of connectivity and flexibility/customizability of peripherals of the Arduino boards.

Board name	Onboard connectivity	GPIO pins	Analog input	USB ports	ICSP header	Other hardware interfaces
Uno	–	14	6	1	1	SPI, UART, I2C/TWI
Mega2560	–	54	16	1	1	SPI, 4 UART, I2C/TWI
Due	–	54	12	2	1	SPI, 4 UART, I2C, 2 TWI
Yún	Ethernet, Wi-Fi	20	12	2	1	SPI, UART, I2C/TWI
101	BLE	14	6	1	1	SPI, UART, I2C/TWI
MKR1000	IEEE 802.11b/g/n	8	7	1	–	SPI, UART, I2C

* Internet of Things A to Z : Technologies and Applications, IEEE Press, 2018, (chapter 6: **(A Tutorial Introduction to IoT Design and Prototyping with Examples)**

Arduino- Connectivity and I/O Complements

- To be familiar with Serial Communication please see
  - https://learn.sparkfun.com/tutorials/serial-communication/all

Arduino- Onboard Sensors

- While a sensor is meant to perceive and measure some type of input (e.g., temperature, pressure, motion, light, heat, or other environmental phenomena) from the physical environment, an onboard sensor is a device that detects and measures not only what is happening in its surrounding, but also within the board.

- The Arduino/Genuino 101 is the only board among the Arduino boards under consideration that features some on board sensors.
  - The board comes with a six-axis accelero meter and a gyroscope.
  - The two sensors can be used together to form an Inertial Monitoring Unit (IMU) that can be employed to identify with accuracy the orientation of the Arduino board.

Arduino- Hardware Security Features

- The computational and memory limitations of most of the Arduino boards have imposed restrictions on their security capabilities, and hence it will be very difficult for them to run mature technology stacks
- The stacks used for securing Hypertext Transfer Protocol (HTTP), Constrained Application Protocol (CoAP), or Message Queuing Telemetry Transport (MQTT) communications are: Internet Protocol Security (IPsec), Secure Sockets Layer/Transport Layer Security (SSL/TLS) or Datagram Transport Layer Security (DTLS).
- Apart from the Arduino/Genuino MKR1000, all the other Arduino boards covered in this course do not have crypto engine or hardware cryptographic module.
- A hardware crypto engine is a self-contained cryptographic module with a dedicated processor, making it difficult for hackers to access valuable data during cryptographic operations.
- It is designed to be integrated into devices as an alternative to software-based security implementations. The module performs encryption and decryption computations much faster than the software implementation of the same operations.
- TheMKR1000 has an onboard cryptochip; essentially, the third block in the ATSAMW25 is the ECC508 that provides crypto authentication. Additionally, the built-in Wi-Fi module supports SHA-256 certificates for ensuring secure communication

ESP 8266 As A SoC

- The ESP-8266 module little beast is an extremely capable wireless programmable microcontroller board.
- The ESP8266 Wi-Fi board is a SOC with integrated TCP/IP protocol stack that can give any secondary microcontroller access to your Wi-Fi network.
- The ESP8266 board is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor and therefore this is more suitable to be used as a sensing node that is capable to sense the data from various wirelessly connected IoT sensor nodes and send data to the central server.

ESP 32 As A SoC

- ESP32: Being the next generation of EPS8266, the MCU possesses all of the above-mentioned characteristics - and boasts some major improvements (more RAM, more hardware, more cores and, as a result, better overall performance).
- ESP32's key characteristics:
  - The microcontroller deploys two CPU cores operating at a frequency of 160 Mhz (compared to the single-core 80 Mhz EXP8266 board);
  - 512 Kb RAM (which enables the MCU to deal with certain encryption algorithms, thus expanding its application scope);
  - Bluetooth 4.2 connectivity (the board can serve as a gateway for Bluetooth-only devices).

https://www.upwork.com/hiring/for-clients/iot-boards-comparison-choosing-hardware-iot-project/

NodeMCU

- The NodeMCU is a popular development board based on the ESP8266.
- ESP8266 is a microcontroller with WiFi capability. it requires external flash memory and some antenna to work.
- NodeMCU is a development board with esp8266 and a firmware with the same name.

Particle Electron (cellular-enabled IoT board)

- Particle is a full-stack IoT device platform with device, connectivity hardware, cloud, and even SIMs for cellular products
- Particle SIM card, a microcontroller, the input and output pins, antenna and USB cable, battery, buttons and LEDs. It has a 120MHz ARM Cortex M3 microcontroller, 1MB flash, 128KB RAM, RGB status LED, 30 mixed-signal GPIO and advanced peripherals plus an open source design.
- *Advantages*: It complies with any cellular standard and comes with 2G/3G connectivity, SIM card, a low-cost data plan, and some great software for making cellular-connected products.
- *Disadvantages*: Cellular connectivity entails license fees and typically, the Particle Electron Iot board is bigger, heavier, more expensive and power consuming than an Iot board that has Bluetooth low energy or Wi-Fi.

Technical specifications for Arduino Uno, Particle Electron, and Espressif Systems ESP8266-01 microcontrollers

Characteristic	Feature	Arduino Uno	Particle Electron	Espressif Systems ESP8266-01
Data acquisition and control				
	GPIO pins	6 Analog in 14 Digital – 6 PWM	12 Analog in 2 Analog out 30 Digital – 13 PWM	2 Digital 1 Analog
	Logic level voltage	5V	3.3V	3.3V
Data processing and storage				
	Processor	ATMega328P	32-bit STM32F205 ARM Cortex M3	32-bit Tensilica L106
	Processor speed	16 KHz	120 MHz	80 MHz
	Memory	32 kB flash, 1 kB EEPROM	1 Mb flash, 128 kB RAM	1 Mb
Connectivity				
	Network Interfaces	None by default. Can be added with shields.	Integrated cellular modem (2G / 3G)	Integrated wifi
Power				
	Recommended Power Supply	9-12V DC 0.5 – 2A barrel, or 5V 500mA USB, or 9 – 12V on VIN pin	5V micro USB or 3.9V-12VDC on VIN pin	Regulated 3.3V 300mA supply on VCC pin
Other				
	Dimensions	2.7 in X 2.1 in	2.05 in x 0.8 in	1.4 in x 1 in
	Typical cost	$20	$39 – $59	$10

IoT Hardware- Single Board Computers (SBC)

- Single Board Computers (SBC)
  - Is more like a mini-PC, and run an embedded operating system, typically a streamlined Linux distribution
  - Is centered on a single microprocessor with RAM, input/output and all other features needed to be a functional computer on the one board.
  - Allows you to attach peripheral devices like keyboards, mice, and screens
  - Offers more memory and processing power

* https://developer.ibm.com/technologies/iot/articles/iot-lp101-best-hardware-devices-iot-project/

IoT Hardwares- Single Board Computers (SBC)

- Pros of Using a SBC
  - **Easy to use** : skip the hundreds of pages of design rule checking (DRC) documents and layout guidelines of the SoCs
  - **Proven hardware** : Making one simple mistake in a SoC board design can be very expensive. Boards reduce this risk
  - **Customizable** : Ability to customize a board means you are paying for exactly what you need
  - **Single source** : BOM of a simple SoC board can still be in the 100s. SBC makes it easy for logistics
  - **Time to Market** : Much faster design cycle than for a SoC

* https://www.semiconductorstore.com/blog/2015/System-on-Chip-vs-Single-Board-Computer-A-Comparison-Guide/689/

IoT Hardwares- Single Board Computers (SBC)

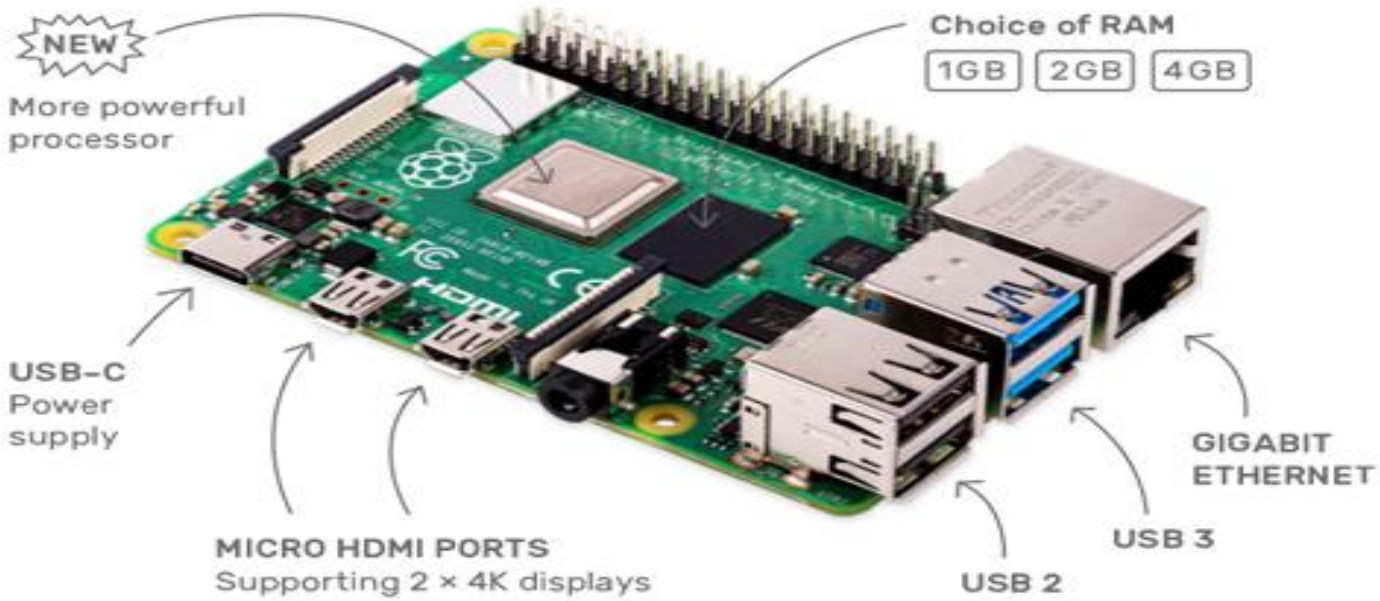
- Cons of Using a SBC
  - **Cost** : If your product is in the high volume category, then it may make more sense to do your own design and justify the engineering costs
  - **Flexibility** : If you need a lot of customization on a given SBC, it will be worthwhile to consider doing a SoC design
  - **Knowledge** : If you plan to use the same/similar SoC for multiple products, then investing the time and effort to develop deep understanding of the product is worth it

* https://www.semiconductorstore.com/blog/2015/System-on-Chip-vs-Single-Board-Computer-A-Comparison-Guide/689/

Raspberry Pi As A SBC

- The Raspberry Pi from the Raspberry Pi Foundation is the world's most popular and most compatible single board computer.

Raspberry Pi 3



Raspberry Pi 4

* https://www.curtisswrightds.com/news/blog/choosing-a-single-board-computer.html

Raspberry Pi As A SBC

- The Raspberry pi Development Board is small sized Broadcom BCM 2835 SoC based ARM11 power minicomputer.
- The raspberry pi can be easily plugged into monitor because of its inbuilt GPU and audio-visual capabilities.
- Also it uses standard mouse and keyboard.
- This is easily programmable by powerful languages like C, python etc, giving it a capability to store and analyze the data.
- The inbuilt wifi, BLE, storage capability of this board and the available RAM being very huge in comparison to other boards enables it to act as an IoT server in most of the IoT network configurations.

Raspberry Pi As A SBC

- Raspberry Pi boards are originally small fully-fledged computers with a range of connectivity options, a processor and up to 8GB of memory storage.
- It's much more powerful and speedy than other IoT boards and can handle complex functionality including data-heavy audio and video streaming.
- Just like Arduino, Raspberry Pi has its own community, accessory set, set-up and troubleshooting guides and multiple resources for developers.
- However, Raspberry Pi is closed-source hardware, so to build a Pi-based application, you'll need to use the boards, accessories and kits offered by the producer.
- Unlike Arduino, Pi is closed-source hardware and is produced only by the Raspberry Pi Foundation (UK). Therefore, there are several models of Pis.

Raspberry Pi-Different Versions

- Raspberry Pi Zero (released in November 2015),
- Raspberry Pi Zero Wireless (W) (released in February 2017),
- Raspberry Pi 1 model B+ (released in July 2014),
- Raspberry Pi 2 model B (released in February 2015),
- Raspberry Pi 3 model B (released in February 2016)
- Raspberry Pi 4 model B (released in 2019)

Raspberry Pi-Different Versions

(a) Raspberry Pi Zero

(b) Raspberry Pi Zero W

(c) Raspberry Pi 1 B+

(d) Raspberry Pi 2 B

(e) Raspberry Pi 3 B

Raspberry Pi-Processing and Memory/Storage Capacity

- In comparison to the Arduino hardware development platforms, the Raspberry Pi SBCs have faster processors, bigger memory, and larger storage capacity
- All Raspberry Pi models are based on the Broadcom SoC that includes :
  - ARM CPU
  - on-chip Graphics Processing Unit (GPU),
  - VideoCore IV low-power mobile multimedia processor that supports up to 1920×1200 resolution.
- For instance,
  - both the Raspberry Pi Zero and the Pi Zero W feature a Broadcom BCM2835 SoC with a 1GHz ARM1176JZF-S CPU core, a member of the ARM11 family (which are 40% faster than Raspberry Pi 1) and have a 512MB Low-Power DDR2 (LPDDR2) SDRAM.
  - The Raspberry Pi 1 B+ is based on the same Broadcom BCM2835 SoC, but with a 700MHz ARM1176JZF-S Core processor and a 512MB RAM (Bell, 2014).
  - The Raspberry Pi 2 B uses a Broadcom BCM2836 SoC with a 900MHz 32-bit quad-core ARM CortexA7 CPU and a 1GB RAM.
  - Similarly, the Raspberry Pi 3 B uses a Broadcom BCM2837 SoC with a 1.2GHz 64-bit quad-core ARM Cortex-A53 CPU and 1GB RAM .8AllRaspberryPiSBCsfeatureamicroSDcardslot.

* Internet of Things A to Z : Technologies and Applications, IEEE Press, 2018, (chapter 6: **(A Tutorial Introduction to IoT Design and Prototyping with Examples)**

Raspberry Pi-Processing and Memory/Storage Capacity

- Summary of processing and storage capacity of Arduino Boards.

	Broadcom SoC				Memory/storage	
Raspberry Pi	SoC	CPU core	Processor architecture	Clock speed	RAM	Storage
Zero	BCM2835	ARM1176JZF-S	–	1 GHz	512 MB	Micro SD
Zero W	BCM2835	ARM1176JZF-S	–	1 GHz	512 MB	Micro SD
1 B+	BCM2835	ARM1176JZF-S	–	700 MHz	512 MB	Micro SD
2 B	BCM2836	Quad-core ARM Cortex-A7	32	900 MHz	1 GB	Micro SD
3 B	BCM2837	Quad-core ARM Cortex-A53	64	1.2 GHz	1 GB	Micro SD

* Internet of Things A to Z : Technologies and Applications, IEEE Press, 2018, (chapter 6: **(A Tutorial Introduction to IoT Design and Prototyping with Examples)**

Raspberry Pi- Power Consumption

- Being a low-power mini computer, The Raspberry Pi operates on a 5VDC power supply at 1–2.5A, powered through a micro-USB port.
- Although the Raspberry Pi consumes different amounts of power in its four distinct power modes namely,
  - run,
  - standby,
  - shutdown,
  - and dormant modes,

- The current draw of each model presented in the official Raspberry Pi magazine are in two modes: idle (standby) and load (run) modes. The current consumptions in the two power modes for each model are:
  - 0.1 and 0.25A for Raspberry Pi Zero;
  - 0.25 and 0.31A for the Raspberry Pi 1 B+;
  - 0.26 and 0.42A fortheRaspberryPi2;
  - 0.31 and 0.58A for the Raspberry Pi3

* Internet of Things A to Z : Technologies and Applications, IEEE Press, 2018, (chapter 6: **(A Tutorial Introduction to IoT Design and Prototyping with Examples)**

Raspberry Pi- Power Consumption

- However, other sources, including the Raspberry Pi Foundation and the element community show more realistic current consumption.

- For example, the average current consumption of the Raspberry Pi Zero when running is approximately 160mA, which shows that the average power consumption is about 0.8W.

- According to Shabaz (2016), the current draw of RaspberryPi3 that runs the Rasbian but doing nothing is 266mA and the current draw when running cpuburn-a53 (i.e., a stress test that maxes out the CPU of a Pi completely) is 1.45A.

- The tests results show that the power consumption of the Raspberry Pi3 ranges from 1.33 to 7.25W.

* Internet of Things A to Z : Technologies and Applications, IEEE Press, 2018, (chapter 6: **(A Tutorial Introduction to IoT Design and Prototyping with Examples)**

Raspberry Pi- Power Consumption

- While the power consumption of the Raspberry Pi actually depends on the usage as well as the model of the device, afew best practice techniques that could be applied during operation to reduce power consumption include the following:
  1. Disconnect every peripheral that is not in use
  2. Switch-off Internet connectivity when not needed
  3. Shutdown the device or put it in dormant mode when not in use
  4. When using the device in headless mode (i.e., accessing it via network connections without a keyboard or display), the High-Definition Multimedia Interface (HDMI) could be switched off
  5. Avoid running several daemons at a time, and run only power efficient applications

* Internet of Things A to Z : Technologies and Applications, IEEE Press, 2018, (chapter 6: **(A Tutorial Introduction to IoT Design and Prototyping with Examples)**

Raspberry Pi- Summary of power consumption, size, and cost

Raspberry Pi	Average power consumption (W)	Size (mm)	Cost ($)
Zero	0.8	65 × 30 × 5.4	5
Zero W	0.9	65 × 30 × 5.4	10
1 B+	3.0	85.6 × 56 × 21	25
2 B	4.0	85.6 × 56 × 21	35
3 B	1.33–7.25	85.6 × 56 × 21	35

* Internet of Things A to Z : Technologies and Applications, IEEE Press, 2018, (chapter 6: **(A Tutorial Introduction to IoT Design and Prototyping with Examples)**

Raspberry Pi- Operating Systems and Programming Languages

- The Raspberry Pi does not come with an OS. Hence, based on their projects, users can choose the type of OS that best suits their needs
- Raspbian is a Debian-based OS freely provided by the Raspberry Pi Foundation for the Raspberry Pi hardware.
- Over the years, the Raspbian OS has remarkably gained popularity among Raspberry Pi users for its high performance. Like a full-fledged OS for traditional computers, it comes with all the basic program utilities and more than 35,000 packages.
- Despite the fact that the Raspbian OS is officially supported by the Raspberry Pi Foundation, the Raspberry Pi SBC is capable of supporting a wide variety of OSes.
  - The Linux-based OSes supported by the Raspberry Pi include Ubuntu MATE, Snappy Ubuntu, Pidora(a Fedora OS for Raspberry Pi), Linutop, SARPi (i.e., Slackware ARM on a Raspberry Pi), Arch Linux ARM, Gentoo Linux, FreeBSD, and Kali Linux.
  - The Raspberry Pi 2 and 3 can also run Windows based Oses like Windows10 IoT Core.

* Internet of Things A to Z : Technologies and Applications, IEEE Press, 2018, (chapter 6: **(A Tutorial Introduction to IoT Design and Prototyping with Examples)**

Raspberry Pi- Programming Languages

- The two programming languages that normally come with the Raspberry Pi by default are
  - Scratch and
  - Python,
- However, over the course of years, several programming languages have been adapted for the Raspberry Pi. Additionally, skilled individuals in the user community who desired to see their favorite languages on the Raspberry Pi have played a significant role in ensuring that their languages of choice have been adapted for the Raspberry Pi.
- Some of the programming languages now available for users to program on the Raspberry Pi using different IDEs include Java, C, C++, Objective C, JS, and Ruby.

Raspberry Pi- Connectivity and I/O Complements

- Almost all the Raspberry Pi SBCs under consideration can be connected to the Internet directly
- The Raspberry Pi Zero does not have an onboard Ethernet port and Wi-Fi capability. It can nonetheless be connected to the Internet in different ways. For example, it can be connected to the Internet by using
  - USB On The Go (OTG) cable along with RJ45 USB converter,
  - or by using USB OTG and Wi-Fi dongle.

Raspberry Pi- Connectivity and I/O Complements

- On the other end of the spectrum, the Pi Zero W has connectivity functionality that includes
  - IEEE 802.11 b/g/n wireless LAN,
  - Bluetooth 4.1, and BLE.14
- Both the Raspberry Pi 1 B+ and Pi 2 B have Ethernet port but no embedded Wi-Fi chip.
- The Raspberry Pi 3 B, however, is equipped with
  - Ethernet
  - Wi-Fi
  - Bluetooth 4.1 and BLE

Raspberry Pi- Connectivity and I/O Complements

- Summary of connectivity and flexibility/customizability of peripherals of the Raspberry Pi boards.

Raspberry Pi	Onboard connectivity	GPIO pins	USB ports	Display ports/ interfaces	Camera port	Other hardware interfaces
Zero	–	40	1 mini	Mini-HDMI	CSI	UART, SPI, I2C
Zero W	Wi-Fi, Bluetooth 4.1, BLE	40	1 mini	Mini-HDMI	CSI	UART, SPI, I2C
1 B+	Ethernet	40	4	HDMI, DSI, 3.5 mm Video Jack	CSI	UART, SPI, I2C
2 B	Ethernet	40	4	HDMI, DSI, 3.5 mm Video Jack	CSI	UART, SPI, I2C
3 B	Ethernet, Wi-Fi, Bluetooth 4.1, BLE	40	4	HDMI, DSI, 3.5 mm Video Jack	CSI	UART, SPI, I2C

Display Serial Interface (DSI)
Camera Serial Interface (CSI)

* Internet of Things A to Z : Technologies and Applications, IEEE Press, 2018, (chapter 6: **(A Tutorial Introduction to IoT Design and Prototyping with Examples)**

Raspberry Pi- Onboard Sensors and Hardware Security Features

- Essentially, no member of the Raspberry Pi family features any onboard sensors;
- However, a typical Linux computer would likely come with onboard monitoring sensors.
  - Thus, the Broadcom SoC includes on-chip temperature and voltage sensors that can be queried using the vcgencmd utility in the firmware package of the Raspberry Pi.
  - The on-chip temperature sensor can be used to measure the temperature of the CPU and GPU.
  - Similarly, the on-chip voltage sensor can be used to measure different voltages, including the core voltage, the SDRAM I/O voltage, and the SDRAM physical memory voltage
  - The temperature or voltage values can be viewed by typing the appropriate commands in the terminal window of the Raspberry Pi.

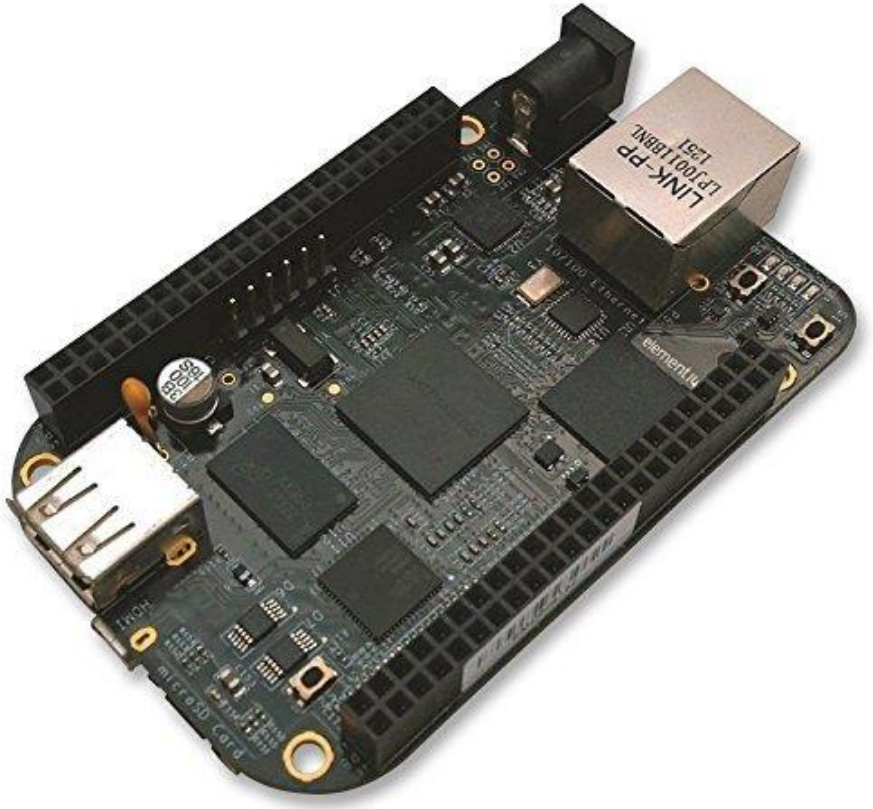
* Internet of Things A to Z : Technologies and Applications, IEEE Press, 2018, (chapter 6: **(A Tutorial Introduction to IoT Design and Prototyping with Examples)**

Raspberry Pi- Onboard Sensors and Hardware Security Features

- Despite the fact that Raspberry Pi is a versatile Linux-based hardware development platform that provides unlimited possibilities for different applications, it lacks hardware-based security engines.
- Additionally, securing private keys for public key cryptography or shared keys for symmetric key cryptography is a very big issue.
  - This is because both data and user code reside on the same SD card, which is usually exposed.

BeagleBone Black As A SBC

- Built on the proven BeagleBoard.or open source Linux approach, BeagleBone AI fills the gap between small SBCs and more powerful industrial computers.
- A low-power open-source single-board computer produced by Texas Instruments in association with Digi-Key and Newark element



* https://www.curtisswrightds.com/news/blog/choosing-a-single-board-computer.html

BeagleBone Black As A SBC

- The BeagleBone Black's features include 512MB DDR3 RAM, 4GB 8-bit eMMC on-board flash storage, 3D graphics accelerator, NEON floating-point accelerator and 2x PRU 32-bit microcontrollers. On the connectivity front, it has USB client for power & communications, a USB host, Ethernet, HDMI and 2x 46 pin headers.
- *Advantages*: The BeagleBoard, especially the BeagleBone Black version, is easy and inexpensive to set up and use. It consumes low power and thus needs no additional cooling or heat sinks. Plus the BeagleBone Black features 65 input and output pins and a huge number of supported interfaces, making it ideal for projects dealing with electronics.
- Moreover, there's the PocketBeagle, which provides identical computing performance to BeagleBone Black along with 50 percent reduction in size, 75 percent reduction in weight and 40 percent cheaper purchase price.

* https://www.curtisswrightds.com/news/blog/choosing-a-single-board-computer.html

BeagleBone Black As A SBC

- *Disadvantages*: The BeagleBoard is not the best bet for complex multimedia and Linux-based projects as it doesn't have the graphical and audio capabilities that the Raspberry Pi has.

* https://www.curtisswrightds.com/news/blog/choosing-a-single-board-computer.html

Technical specifications for Raspberry Pi 4, BeagleBone Black and DragonBoard SBCs

Characteristic	Feature	Raspberry Pi 4	BeagleBone Black	Qualcomm DragonBoard 410c
Data acquisition and control				
	GPIO pins	40 I/O pins, including 29 Digital	65 Digital – 8 PWM 7 Analog in	12 Digital
	Logic level voltage	3.3V	5V	1.8V
Data processing and storage				
	Processor	ARM Cortex A72	AM335X ARM Cortex A8	ARM Cortex A53
	Processor speed	1.5GHz	1 GHz	1.2 GHz
	Memory	1-4 Gb	512 Mb RAM, 4 Gb Flash	1Gb, 8Gb Flash
Connectivity				
	Network Interfaces	Wifi, Ethernet, Bluetooth	Ethernet, USB ports allow external wifi / Bluetooth adaptors	Wifi, Bluetooth, GPS
Power				
	Recommended Power Supply	5V 3A USB Type C	5V 1.2A – 2A barrel	6.5 – 18V 2A barrel
Other				
	Dimensions	3.5 x 2.3 in	3.4 x 2.1 in	3.3 in x 2.1 in
	Typical cost	$35	$55	$75

Comparison of Arduino, Raspberry Pi, and ESP 8266 for IoT Projects

Parameters	Arduino Uno	Raspberry Pi B+	ESP-8266
Processor	ATMega328P	Quad-core ARM Cortex A53	-
GPU	-	Broadcom VideoCore IV with 400 MHz	-
Operating voltage	5 V	5 V	3.3 V
Clock speed	16 MHz	1.2 GHz	26 MHz – 52 MHz
System memory	2 kB	1 GB	< 45 kB
Flash memory	30 kB	-	Up to 128 MB
EEPROM	1 kB	-	-
Communication supported	IEEE 802.11 b/g/n IEEE 802.15.4 433RF BLE 4.0 via Shield	IEEE 802.11 b/g/n IEEE 802.15.4 433RF BLE 4.0, Ethernet Serial	IEEE 802.11 b/g/n
Development environments	Arduino IDE	Any linux compatible IDE	Arduino IDE, Lua Loader
Programming language	Wiring	Python, C, C++, Java, Scratch Ruby	Wiring, C, C++
I/O Connectivity	SPI I2C UART GPIO	SPI DSI UART SDIOCSI GPIO	UART, GPIO

Comparing Arduino vs. Raspberry Pi for IoT Projects

- **IoT Arduino**
  - As a rule, Arduino is used in the systems with simple repetitive tasks that imply only a single action at a time.
    - For example, using pins, you can connect to the board an analog sensor that monitors humidity in the room and program it to turn on a humidifier when the humidity level drops down to a certain level.
  - In many cases, Arduino will be a more cost and effort efficient solution compared to Raspberry Pi.

*Comparing Arduino vs. Raspberry Pi for IoT Projects, [online] Available at https://www.digiteum.com/comparing-arduino-raspberry-pi-iot

Comparing Arduino vs. Raspberry Pi for IoT Projects

- **IoT Raspberry Pi**
  - Faster and more powerful Raspberry Pi can handle multitasking and run more complex functionality than Arduino.
  - It includes running media playback, making calculations and collecting various parameters at the same time.
  - So if an IoT system needs to collect data from several sensors, pull data from the Internet, connect to a smartphone and provide a complex output on a display, Raspberry Pi is the right option.

*Comparing Arduino vs. Raspberry Pi for IoT Projects, [online] Available at https://www.digiteum.com/comparing-arduino-raspberry-pi-iot

Comparing Arduino vs. Raspberry Pi for IoT Projects

- One more thing: The choice should not always be Arduino vs. Raspberry Pi.

- In fact, it can be confusing choosing between the two when both boards have their own pros and cons. Instead, these boards can perfectly work together if an IoT project requires so.
- F
- or example, in an IoT agriculture system, uncomplicated and cheap Arduino boards can be used to read measurements from multiple soil sensors, while a Raspberry Pi can work as a think tank and decide how to act on the data collected by sensors.

Examples of IoT Projects Using IoT Developments Boards

- *Particle Electron or Arduino projects*: The ideal projects include making a GPS + Cellular tracker (to track kids or even valuable objects), monitoring remote sensors and making a high current alarm.

- *Raspberry Pi projects*: One can host simple websites by using Raspberry Pi as a low-cost web server. You can also use it to create a captive portal for your guest Wi-Fi, as also for creating facial recognition projects and for building a Raspberry Pi powered robot or even a laptop!

- *BeagleBoard projects*: BeagleBoard is ideal for beginners in electronics and programming. So projects that require reading from external sensors, commanding actuators (such as motors or light systems), and networking are simpler and more efficient with the BeagleBone Black than with a Raspberry Pi.

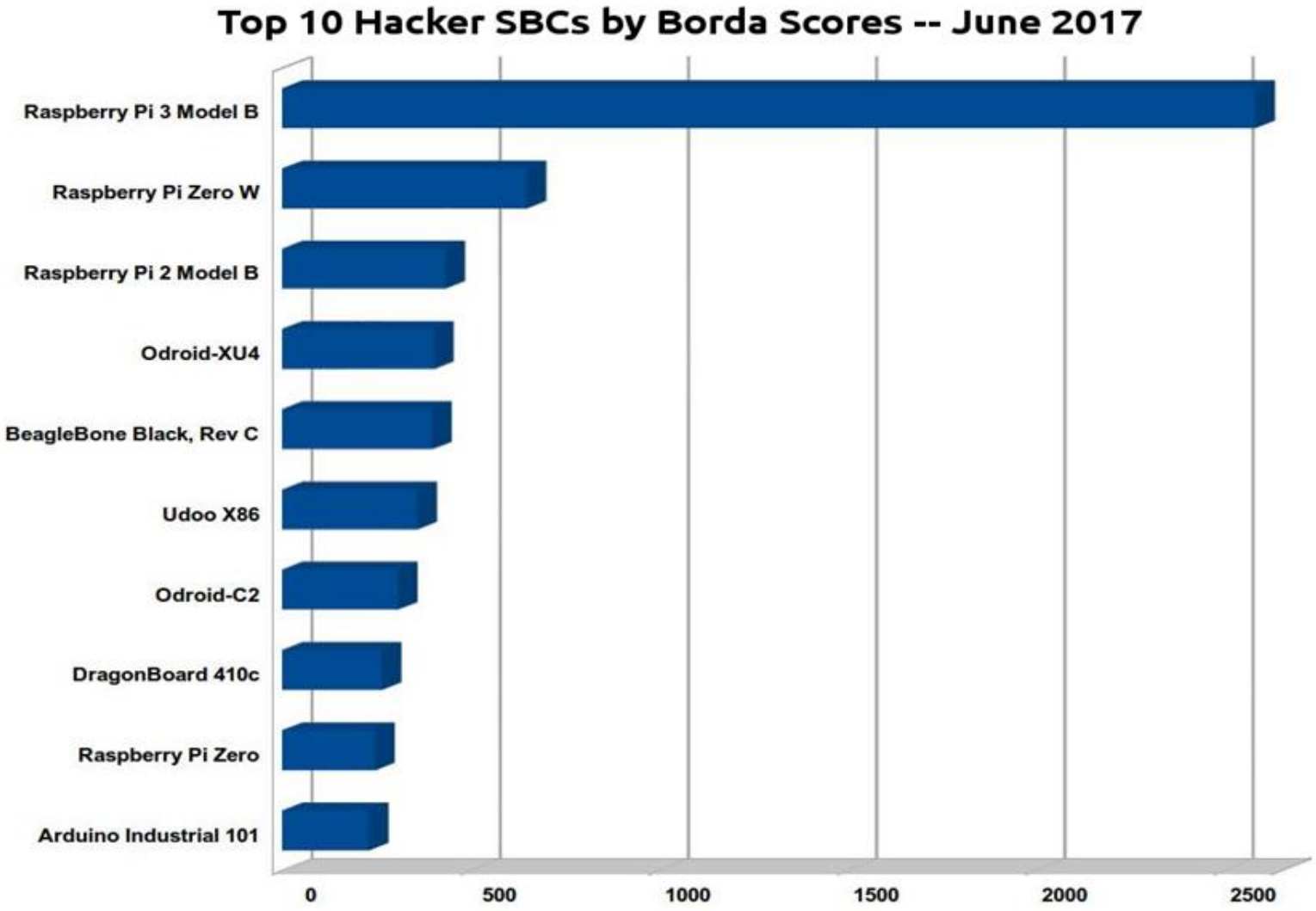
Choosing Right IoT Development Board

- Choosing between microcontroller development boards and single-board computers

- IoT hardware requirements for deploying your IoT project

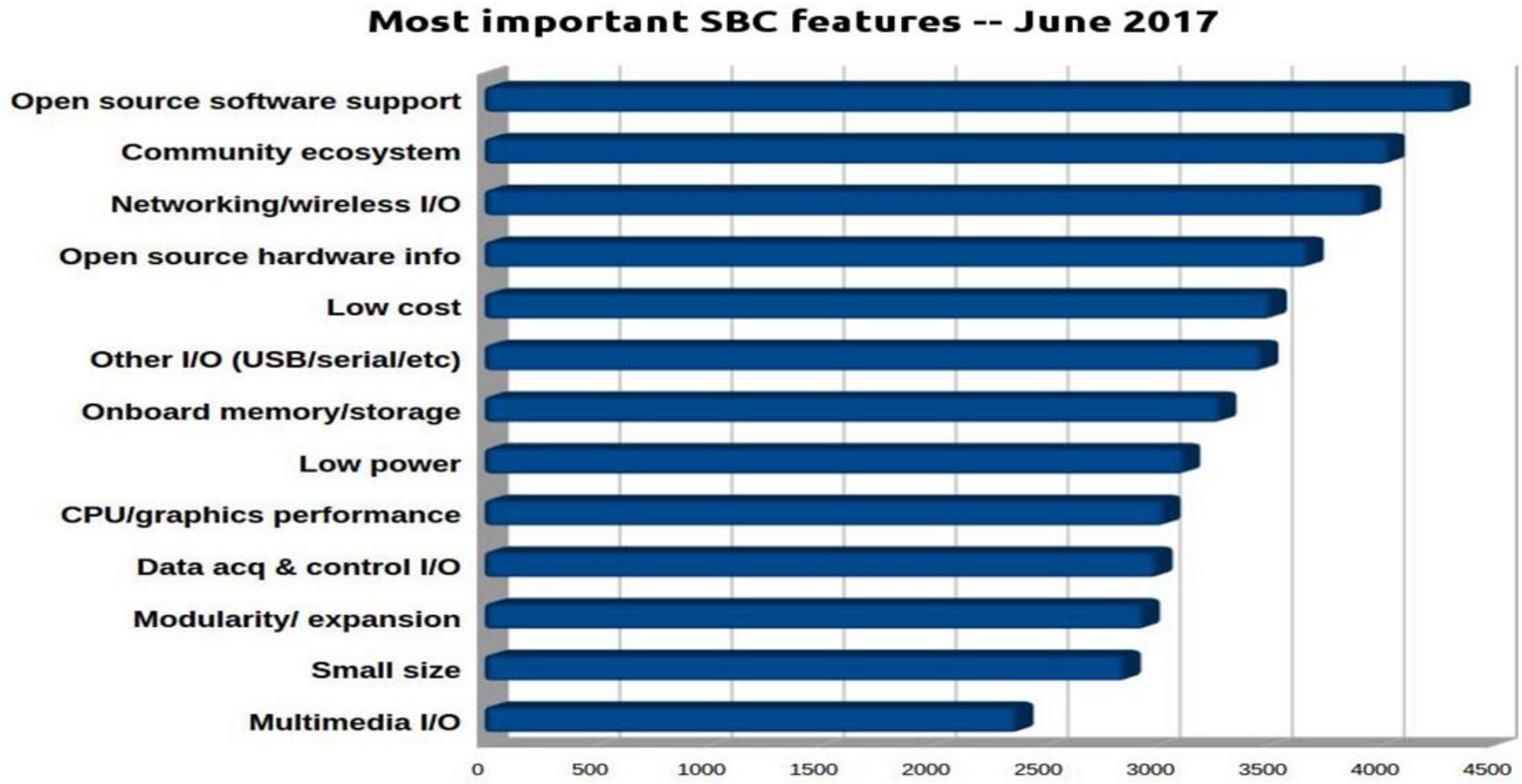
Choosing Right IoT Development Board

- It would be wrong to say which IoT board is better, because each of the boards has its own benefits and fundamental differences.
- IoT boards are great for prototyping.
  - Arduino is easier to set up and go, while Raspberry Pi provides more onboard features.



* http://linuxgizmos.com/2017-hacker-board-survey-raspberry-pi-still-rules-but-x86-sbcs-make-gains/

Choosing Right IoT Development Board



* http://linuxgizmos.com/2017-hacker-board-survey-raspberry-pi-still-rules-but-x86-sbcs-make-gains/

Choosing Right IoT Development Board

- **Choosing the right single board computer (SBC) for an application requires many considerations.**
  - IoT Boards Capabilities:
    - Processor and Storage Capacity
    - Connectivity Capabilities and I/O Complement
    - Power Requirements
  - Other Criteria:
    - Expansions and Add-on Modules
    - Operating System and Programming
    - Board Size and Form Factor
    - Cost

Choosing Right IoT Development Board

- **Storage and Processor Capacity**
  - **Processor choice**
    - The three primary choices seen in the embedded COTS market place are Intel, Power Architecture, and more recently ARM. Some of the considerations that follow will also influence processor choice.
    - Application requirements vary vastly from low performance (and with that low power), to extremely high performance applications like SigInt with terrflops of processing requirements. Choose your processor wisely, as there's no need to pay for more performance and power than the application calls for.
  - **Storage**

    With advancing technology, memory requirements have increased dramatically. And a gigabyte of memory was unheard of. Today many Intel processors offer 16 to 32 GB of memory. The memory demand of an application can influence the processor choice. For instance, compared to Intel, Power Architecture and ARM typically offer less memory and with that less power.

* https://www.curtisswrightds.com/news/blog/choosing-a-single-board-computer.html

Choosing Right IoT Development Board

- **Storage and Processor Capacity**
  - When comparing Arduino board vs. Raspberry Pi for a simple single-action IoT application:
    - memory storage maybe not the most important factor.
    - However, if your IoT device is multifunctional, it may require substantial RAM.
  - As a microcontroller, Arduino has just enough memory to run a simple execution code.
    - For example, Arduino UNO has only 32 K bytes of flash memory and 2 K bytes of SRAM.
  - Raspberry Pi are small computers so they have much larger memory capacity.
    - For example, the latest models are said to provide up to 8GB of SDRAM. It's enough to run multiple functionalities for a more complex IoT system.

*Comparing Arduino vs. Raspberry Pi for IoT Projects, [online] Available at https://www.digiteum.com/comparing-arduino-raspberry-pi-iot

Choosing Right IoT Development Board

- **Connectivity and I/O Complement**
  - Connectivity capabilities are, probably, the most important features when it comes to building an IoT system using IoT development boards.
  - For a particular application, a development board must provide the right I/O complement in the right types and quantities, such as Ethernet, DIO, SATA, USB, serial ports (232, 422, 485) as well as board interconnect (VME, SRIO, PCIe, Ethernet). Related to this consideration is available support for add-on of a mezzanine card to expand on I/O not provided by the base SBC.
    - The latest Raspberry Pi models have WiFi, Bluetooth, Ethernet connectivity installed by default, offer a pretty good speed and processing power which can handle video and audio data.
    - Arduino requires shields or modules to add basic connectivity options like WiFi, GPS or Bluetooth. Though it's not complicated and takes just a few steps.
    - One of the most popular LPWRN options in IoT, LoRa can be easily connected to both Raspberry Pi and Arduino with a special module.

*Comparing Arduino vs. Raspberry Pi for IoT Projects, [online] Available at https://www.digiteum.com/comparing-arduino-raspberry-pi-iot

Choosing Right IoT Development Board

- **Power supply**
  - Choose an SBC that matches the power requirements for the specific application. As a complement to power, make sure the thermal management capabilities of the system are up to the task of cooling the chosen board for the system.
    - In general, Raspberry Pi and Arduino are low power devices. Both are powered externally.
    - In the case of Raspberry Pi, Micro USB or general-purpose I/O headers are used. The range is strictly limited to 5V.
    - Arduino boards are more flexible. The recommended range is 7-12V and, as a rule, USB ports are used for power supply.

*Comparing Arduino vs. Raspberry Pi for IoT Projects, [online] Available at https://www.digiteum.com/comparing-arduino-raspberry-pi-iot

Choosing Right IoT Development Board

- **Expansion**
  - Both Arduino and Raspberry Pi boards can be expanded to add features and functions to the system.
  - Arduino is expanded with shields (boards installed on top of the main board using general pin headers)
    - Shields allow adding on functionality like GPS, SD card, connection to the Internet via Ethernet, LCD display, etc.
    - More than one shield can go on top as long as it's properly connected. In this case, the hardware becomes a bit bulky and loses its portability features.
  - Raspberry Pi has many of the previously mentioned features on the original board.
    - The latest Raspberry Pi models have media and audio input/output, several connectivity options, SD port, and other features.
    - Additional modules and accessories like camera, display and expansion boards (HAT, hardware attached on top) can also be installed.

*Comparing Arduino vs. Raspberry Pi for IoT Projects, [online] Available at https://www.digiteum.com/comparing-arduino-raspberry-pi-iot

Choosing Right IoT Development Board

- **Board Size and Form Factor**

  Important in any decision is the form factor of available SBCs. Form factors common in COTS embedded SBCs are: VME, VPX, Compact PCI and ComExpress. In the past 6U VME has been the most commonly used, but that is slowly changing to VPX, with increased interest in 3U for SWaP reasons.

* https://www.curtisswrightds.com/news/blog/choosing-a-single-board-computer.html

Choosing Right IoT Development Board

- **Operating System and Programming**
  - Operating systems many impact the processor and development board choice. Typical OS-es available for embedded use are: Linux (of which there are many versions), INTEGRITY, GreenHills LynxOS, QNX, and Wind River VxWorks. A variety of processors support Linux.
  - Raspberry Pi runs on Rasberian OS provided by the Raspberry Pi Foundation, but can also run on some third-party operating systems like Linux, Android and Android Things designed specifically for IoT projects.
  - Many programming languages including Python, C/C+ and Scratch will work to build IoT projects using Raspberry Pi.
  - Arduino doesn't have a dedicated operating system but has a cross-platform (Windows, MacOS, Linux) Arduino IDE (integrated development environment) with many free software libraries and code editor for faster and simpler programming. The environment supports C/C+. However, basically, any programming language can be used to write the functionality for Arduino.

Choosing Right IoT Development Board

- **Cost**
  - The average price for a new Arduino board is around 20-25 USD.
  - Raspberry Pi provides a wider price range starting from 5 USD for the very basic Pi Zero model. Though, Pi models loaded with features and larger memory capacity may reach 55 USD.

CONTENTS

- Smart Objects (Things)
- Development Boards (Hardware Platforms) for IoT
  - Microcontroller Development Boards (System on a Chip (SoC))
  - Single-Board Computers (SBC)
  - How to Choose the Right Development Board for IoT Projects
- IoT Operating Systems
- Power Consumptions Considerations

IoT Operating Systems

- An IoT operating system, often called an embedded OS is an OS designed for the particular demands and specifications of IoT devices and applications.
- An IoT OS is critical to connectivity, security, networking, storage, remote device management, and other IoT system needs.
- Some IoT operating systems also have real-time processing capabilities and are referred to as real-time operating systems, or RTOS.
- An IoT OS enables devices and applications to connect with each other and other systems, such as cloud platforms and services.
- The IoT OS also manages the processing power and other resources needed to collect, transmit, and store data.

* https://www.arm.com/glossary/iot-operating-system

IoT Operating Systems

- Classification of IoT Oss
  - High-End IoT OS
    - which mostly is linux based. This operating system runs over SBCs or SoCs which are with high resources, memory and power e.g. Raspberry pi
  - Low-End IoT OS
    - which may be of linux based or non-linux based. This operating system runs over SoCs and sensors/actuators which are small board with limited resources, computational power and capacity e.g. Arduino

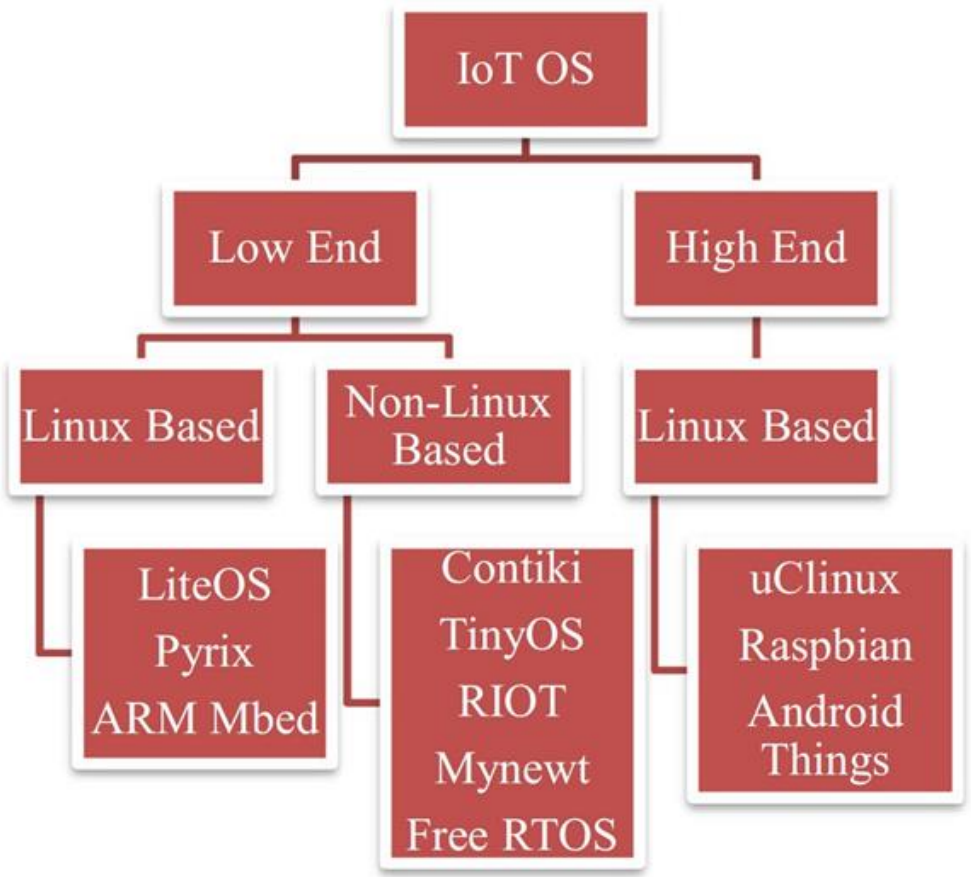
* IoT Ecosystem: A Survey on Devices, Gateways, Operating Systems, Middleware and Communication, Springer, 2019

IoT Operating Systems

- Compared to standard computers, tablets, and smartphones, many IoT devices are resource constrained, especially in terms of memory footprint.
- Hence they cannot run High-Level Operating System (HLOS) such as Windows and Linux.
- In addition, the diversity of MCU families and architectures (e.g., there are 8-bit, 16-bit, and32-bit processors) is one of the greatest bottlenecks to the development of generic Oses for these devices.
- The above mentioned diversity is also aggravating the restrictions on providing a more generic Os support for IoT heterogeneous hardware.

* Internet of Things A to Z : Technologies and Applications, IEEE Press, 2018, (chapter 6: **(A Tutorial Introduction to IoT Design and Prototyping with Examples)**

IoT Operating Systems



* IoT Ecosystem: A Survey on Devices, Gateways, Operating Systems, Middleware and Communication, Springer, 2019

IoT Operating Systems

- RIOT: A lightweight operating system for memory-constrained, wireless networked systems with a focus on low-power IoT devices.
  - RIOT supports a wide range of devices and offers higher complexity that allows for larger and time sensitive IoT applications.

- Contiki: An open-source operating system for connecting memory-constrained, low-power wireless networked systems to the Internet.
  - It is similar to RIOT but less advanced in terms of latency and performance (i.e., multithreading) capabilities.
  - One of the main advantages of Contiki is its simulator (i.e., Cooja).

IoT Operating Systems

OS	Programming languages	Simulator	Required RAM, ROM	Multithreading	Real-time capabilities
RIOT	C, C++	None	1.5 KB, 5 KB	Yes	Yes
Contiki	Partial C	Cooja	2 KB, 30 KB	No	No

- Other operating systems can also be used, such as Linux, but due to its requirement for larger ROM and RAM (i.e., ~1 MB), it may not be suitable for tiny IoT devices.

CONTENTS

- Smart Objects (Things)
- Development Boards (Hardware Platforms) for IoT
  - Microcontroller Development Boards (System on a Chip (SoC))
  - Single-Board Computers (SBC)
  - How to Choose the Right Development Board for IoT Projects
- IoT Operating Systems
- Power Consumptions Considerations

Power Consumption

- One of the first decisions in IoT system design should be the power source.
- If the device will be powered by batteries then all design decisions must consider how to preserve power.
- Many networking technologies will not be a good fit with battery power.
- Frequency of communication does have an influence on power selection, too (will be discussed later in Chapter 3)

Power Consumption

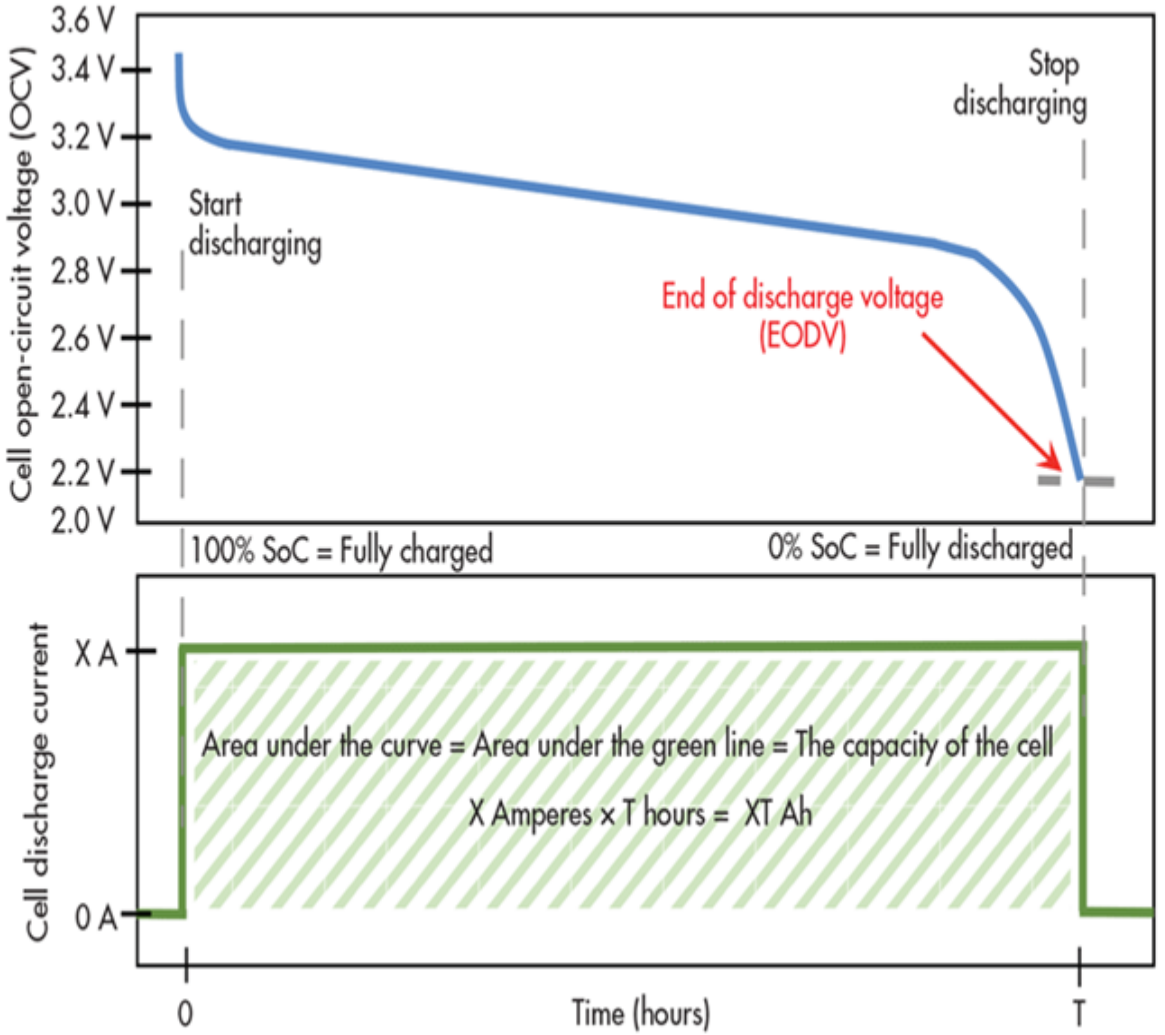
- A Key requirement for IoT devices is the ability to consume very low power while maintaining an acceptable level of performance, which could enable them to run on batteries for long periods of time.
- Strategies for achieving ultralow-power operation in IoT devices include, among others,
  - Employing low-power communication technologies
  - Implementing low-duty-cycle operations.

Battery Capacity and Battery Life

- Each battery has a capacity measured in ampere-hours (Ah).
  - one ampere-hour means that you can draw one ampere from the battery for one hour.
  - Likewise, 1 Ah also means you can draw 2 A for 0.5 hours, or 0.25 A for 4 hours
- Battery capacity can also be measured in watt-hours (Wh))
  - Wh capacity is a measure of stored energy. Looking at units,
  - One watt is one joule per second.
  - If you multiply watts × time, you get joules.
- Wh to mAh Conversion Formula
  - Wh= (mAH ÷ 1,000) × V
  - mAh = (Wh × 1,000) ÷ V
- For example, let's convert 750 mAh at 3.7 Volt
  - Capacity in Wh=(750 mAh ÷ 1,000) × 3.7 V=2.775

Battery Capacity

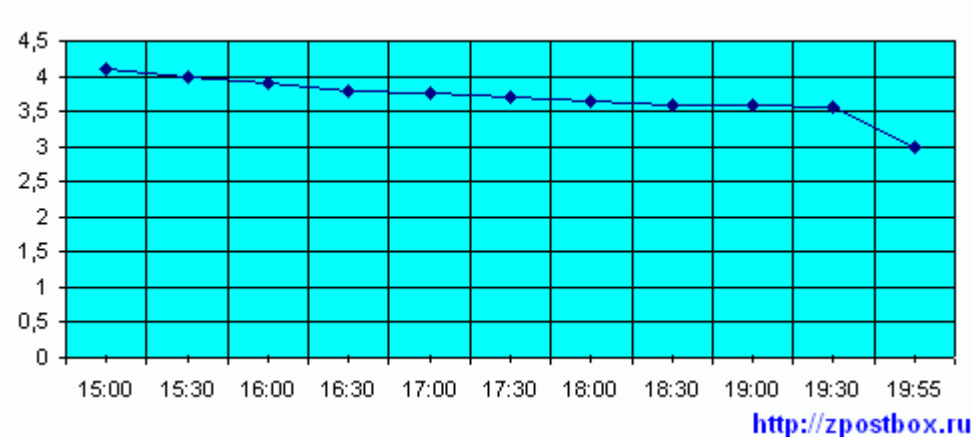
- To measure the battery capacity:
  - starting with a fully charged cell, draw a constant current of X amperes until it is discharged.
  - The cell is considered discharged when the cell's voltage reaches the end of discharge voltage
- The illustration shows the discharge profile of a lithium-ion cell.
  - The top line is the voltage vs. time, starting at fully charged and continuing until the end-of-discharge voltage is reached.
- The current is constant during this discharge. The time measured is the length of time required to discharge. The capacity of the cell is the area under the discharge curve.



*Measuring Cell Capacity, [online] Available at https://www.electronicdesign.com/technologies/test-measurement/article/21805209/measuring-cell-capacity

Battery Capacity

- Voltage across the battery NB-11L in the discharge process by the current of 100 mA.



  - It took almost 5 hours (at the current of 0.1 A) for the voltage across the battery to drop to 3 Volts. At the end the discharge voltage declines faster.
  - Now we can calculate the battery capacity: $C = I * t = 0.1 * 5 = 0.5\ A = 500\ mA*h$.
  - So the real capacity of the battery NB-11L of unknown brand is 1.5 times less than it is marked capacity.

*How to measure the capacity of battery: a simple and accurate way [online] Available http://zpostbox.ru/how_to_measure_the_capacity_of_battery_a_simple_and_accurate_way.html

Battery Capacity

Chemistry	Nominal voltage (V)	Capacity under 50 mA constant drain (mAh)	Max. energy at nominal voltage and 50 mA drain (Wh)	Rechargeable
Zinc–carbon	1.50	400–1700	2.55	No
Alkaline	1.50	1800–2850[17]	3.90	Some
Li-FeS_2	1.50	2700-3400	5.10	No
Li-ion	3.60–3.70	600-840 (1600 mAh at 1.5V)	2.88-2.96	Yes
NiCd	1.20	600-1,000	1.20	Yes
NiMH	1.20	600–2,750	3.42	Yes
NiZn	1.60-1.65	1500-1800	2.97	Yes

Battery Capacity

Power Supply

- The current consumption is given by

$$I_{on} = \frac{\sum_{i=1}^{N_s}(T_{Si} I_{Si})}{T_{on}}$$

where I_{on} is the average current that can be calculated while the device is awake in a connection event T_{on} and the time T_{Si} spent in each state with its corresponding current consumption I_{Si}. N_S represents the number of states and i is the index of the state.

* E. Garcia-Espinosa etl, , Power Consumption Analysis of Bluetooth Low Energy Commercial Products and Their Implications for IoT Applications, Electronics, 2018 [Online] available at https://www.mdpi.com/2079-9292/7/12/386

Power Supply

- The total average current I_{avg} during a connection interval T_{ci}, considering the sleep current I_{sleep} while the system is in an idle state, can be computed by

$$I_{avg} = \frac{I_{on}T_{on} + I_{sleep}(T_{ci} - T_{on})}{T_{ci}}$$

- Subsequently, the battery lifetime in hours is given by

$$T_{bat} = \frac{C_{bat}}{I_{avg}}$$

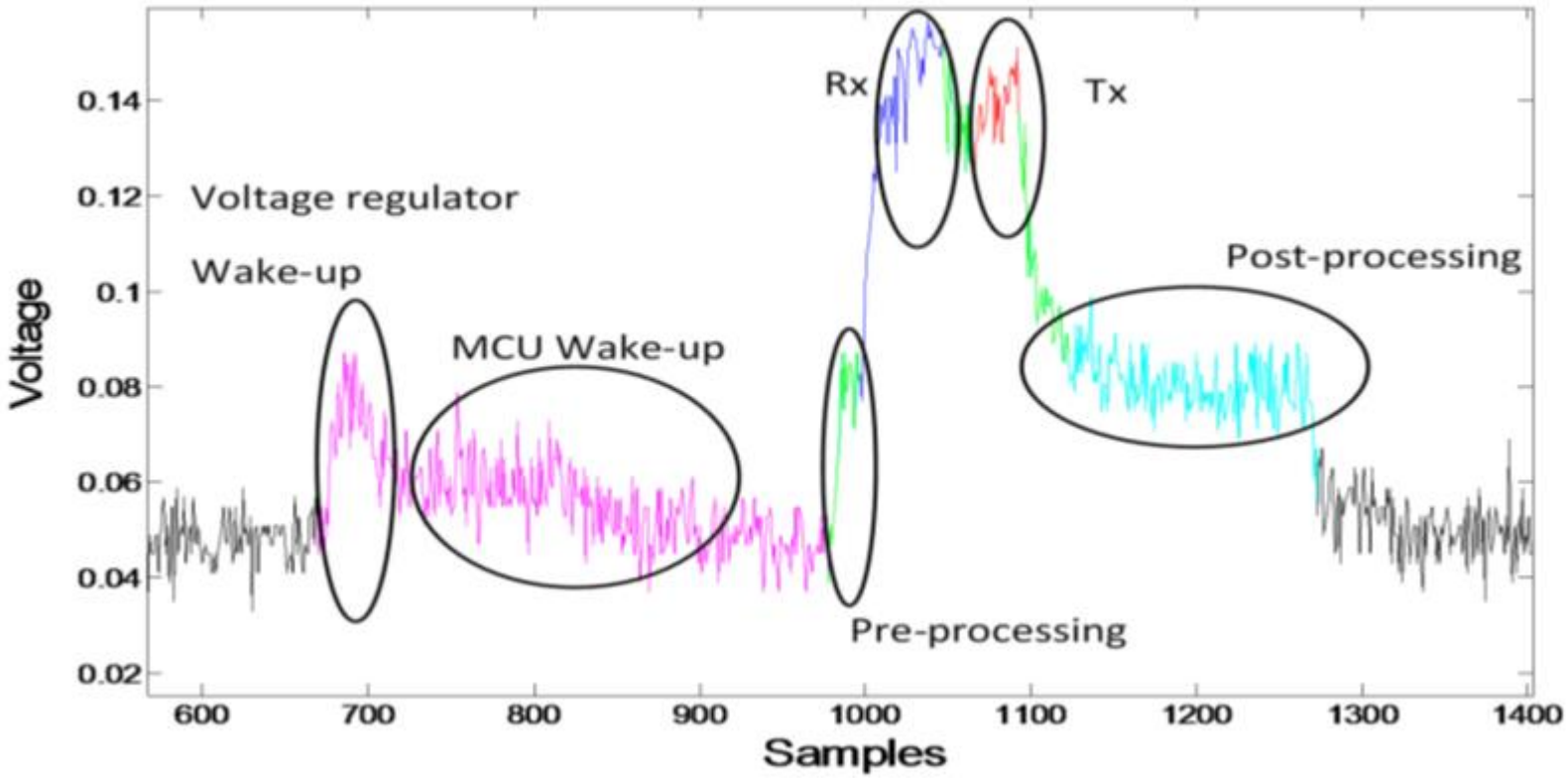
where C_{bat} is the battery capacity in mA.

- The battery life (in hours) of an IoT device can be calculated by dividing the capacity of the battery (in milliamp-hour (mAh)) by the average current consumption (in milliampere (mA)) of the device.

* E. Garcia-Espinosa etl, , Power Consumption Analysis of Bluetooth Low Energy Commercial Products and Their Implications for IoT Applications, Electronics, 2018 [Online] available at https://www.mdpi.com/2079-9292/7/12/386

Power Supply

- The current consumption of a connection event processed with Matlab to obtain
- the average current values*



* E. Garcia-Espinosa etl, , Power Consumption Analysis of Bluetooth Low Energy Commercial Products and Their Implications for IoT Applications, Electronics, 2018 [Online] available at https://www.mdpi.com/2079-9292/7/12/386

Power Supply

- The power consumption or battery life of a hardware development board will largely depend on the operating voltage and and operating current of the board as well as on the components that are connected to it.
- Data is acquired by some sensor of the system, processed in a controller unit and some information is then sent through a wireless or wired channel.
- Based on this assumption, the power required to operate an IoT device can be broken down into three main blocks:
  - For data sensing or acquisition PACQ,
  - For data handling or processing (PRC) ,
  - For data communication, or networking (PNET)
  - For system management tasks, such as running a real-time operating system (RTOS) or rising the system at periodic wake-ups (PSYS)
- These elements together form the general expression for the device's power consumption (PDEV):
  - PDEV = PACQ + PRC+ PNET + PSYS

*B. Martinez and etl, The Power of Models: Modeling Power Consumption for IoT devices, IEEE Sensor Journals, 2015 [online] Available at https://ieeexplore.ieee.org/document/7122861

Power Supply

- Time between consecutive messages T_{MSG}
- While in many situations the sampling period T_S is determined by the underlying physical magnitude and filtering requirements (e.g. anti-aliasing Low-Pass Filters)
- The time between consecutive records T_{RCD} is scheduled from the application layer, thus providing a mechanism for balancing energy consumption and sensing accuracy.

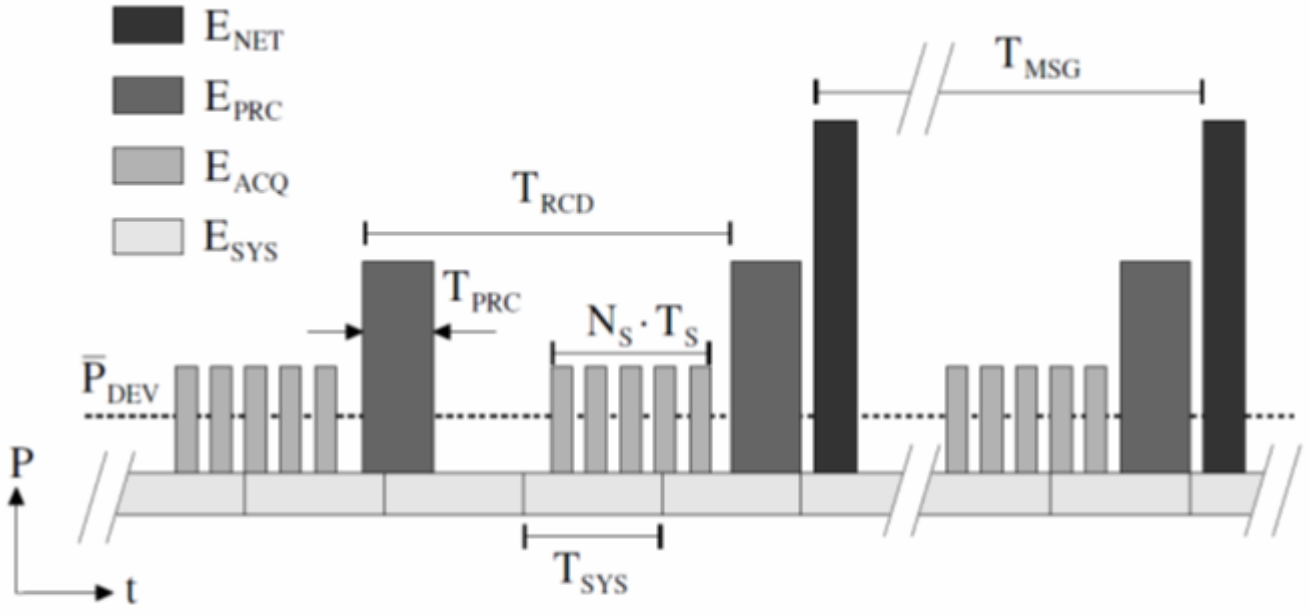


Fig. 2. Characteristic time evolution of energy usage. The vertical dimension represents the instantaneous power consumption of the device. Consequently, shaded areas depict the accumulated energy for each task. The dashed line in the figure represents the average power of the device $\bar{P}_{DEV}$.

*B. Martinez and etl, The Power of Models: Modeling Power Consumption for IoT devices, IEEE Sensor Journals, 2015 [online] Available at https://ieeexplore.ieee.org/document/7122861

Power Supply

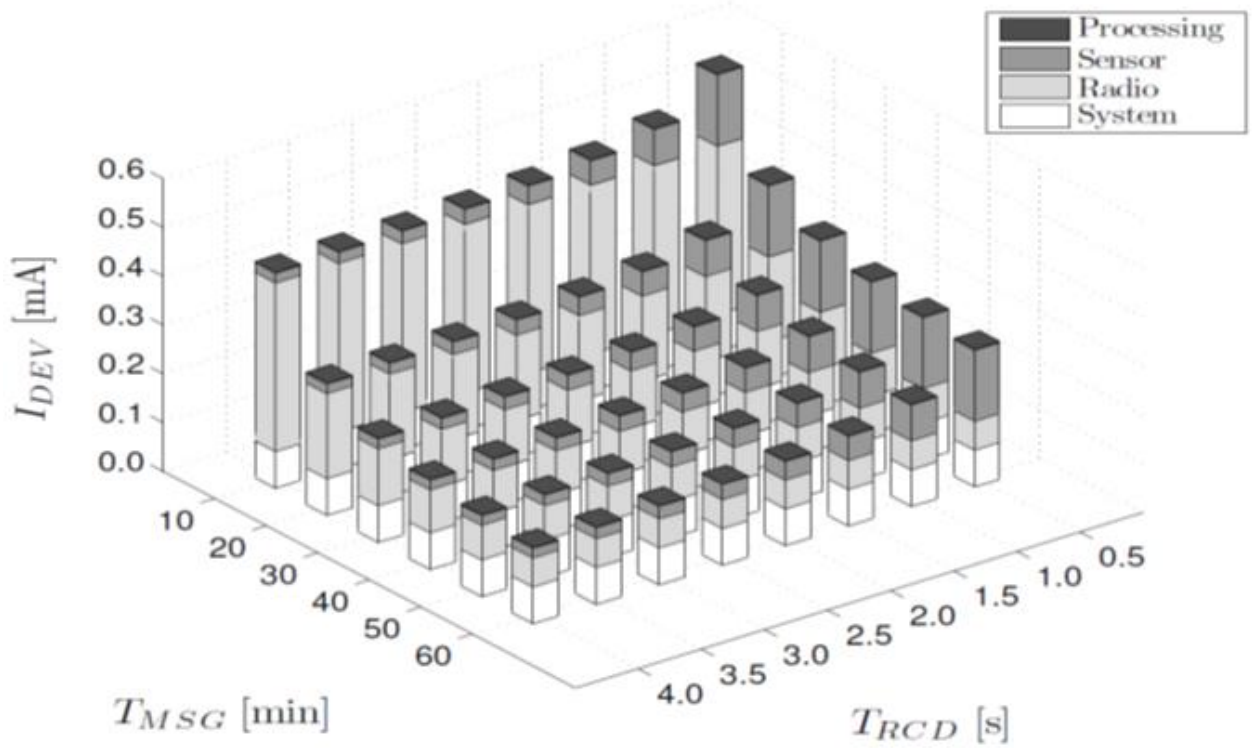


Fig. 12. Simulated consumption for a simple periodic-reporting application as function of T_{MSG} and T_{RCD}. [Legend Rev.]

*B. Martinez and etl, The Power of Models: Modeling Power Consumption for IoT devices, IEEE Sensor Journals, 2015 [online] Available at https://ieeexplore.ieee.org/document/7122861

Power Supply

- Extrapolating the values for different connection intervals, an approximation of the battery life associated to a connection interval for each platform is shown in following Table.
- The calculation of the battery life is based on a CR-2032 battery, which has a typical capacity of 235 mAh.

Connection Interval (ms)	I_{avg} (mA)				Battery Life (in Days)			
	TI CC2540	Intel A-101	Cypress CY8CKIT-042-BLE-A	NXP FRDM-KW41Z	TI CC2540	Intel A-101	Cypress CY8CKIT-042-BLE-A	NXP FRDM-KW41Z
10	2.78	0.66	1.404	1.176	3.5	14	7	8
50	0.558	0.130	0.281	0.236	17	73	35	41
100	0.279	0.067	0.141	0.118	35	145	69	82
300	0.093	0.023	0.047	0.04	104	423	205	243
500	0.056	0.014	0.029	0.024	172	686	337	399
800	0.035	0.009	0.018	0.015	273	1054	528	624
1000	0.028	0.007	0.015	0.012	339	1283	651	767
2000	0.014	0.004	0.008	0.006	655	2269	1221	1423
3000	0.01	0.003	0.005	0.004	951	3050	1725	1990
4000	0.007	0.002	0.004	0.003	1229	3684	2171	2485

Power Consumption

- The current consumption of the MCU is about **20mA**,
  - With a 5V supply, the required power is P = 0.02 * 5 = 0.1 Watts
- The Wi-Fi alone can consume about **100mA** or more when operational
  - With a 5V supply, the required power is P = 0.1 * 5 = 0.5 Watts
- The Camera Module requires **250mA** when operational
  - With a 5V supply, the required power is P = 0.25 * 5 = 1.25 Watts
- With a 1000 mAh battery (at 5V), if MCU, Wi-Fi and Camera are in operation all the time, how much is the battery (device) life?
  - 1000 ÷ 370=2.7 hours

Average Power Consumption

- For Arduino versions

Board name	Average power consumption
Uno	250 mW
Mega2560	250 mW
Due	2.64 W
Yún	≥1 W
101	–
MKR1000	≥ 396 mW

- For Raspberry Pi

Raspberry Pi	Average power consumption (W)
Zero	0.8
Zero W	0.9
1 B+	3.0
2 B	4.0
3 B	1.33–7.25

Power supply

- For example in ESP8266,
  - The MCU can consume anything between 100 mA (Wi-Fi operation and data transmission) to 10 mA (deep sleep mode);
    - the amount of energy consumed by the board actually depends on the setup and harness.
  - With additional circuits, the deep sleep mode power consumption can be further reduced to less than 0.1 mA (typically, 0.07-0.08mA).
    - It means that a 3000 mAh device (the size of an Android smartphone battery) can keep the MCU going for about 3 years!

* https://www.upwork.com/hiring/for-clients/iot-boards-comparison-choosing-hardware-iot-project/

Power supply

- In ESP32:
  - The microcontroller uses about 150mA with WiFi and Bluetooth data transmissions active at the same time, 5 mA if the deep sleep mode is activated and around 0.04 mA with reduced schematics.
  - When measured against a 3000 mAh smartphone battery, the device can potentially last for around 5 years.

* https://www.upwork.com/hiring/for-clients/iot-boards-comparison-choosing-hardware-iot-project/

Raspberry Pi- Power Consumption

- Being a low-power mini computer, The Raspberry Pi operates on a 5VDC power supply at 1–2.5A, powered through a micro-USB port.
- Although the Raspberry Pi consumes different amounts of power in its four distinct power modes namely,
  - run,
  - standby,
  - shutdown,
  - and dormant modes,

Raspberry Pi- Power Consumption

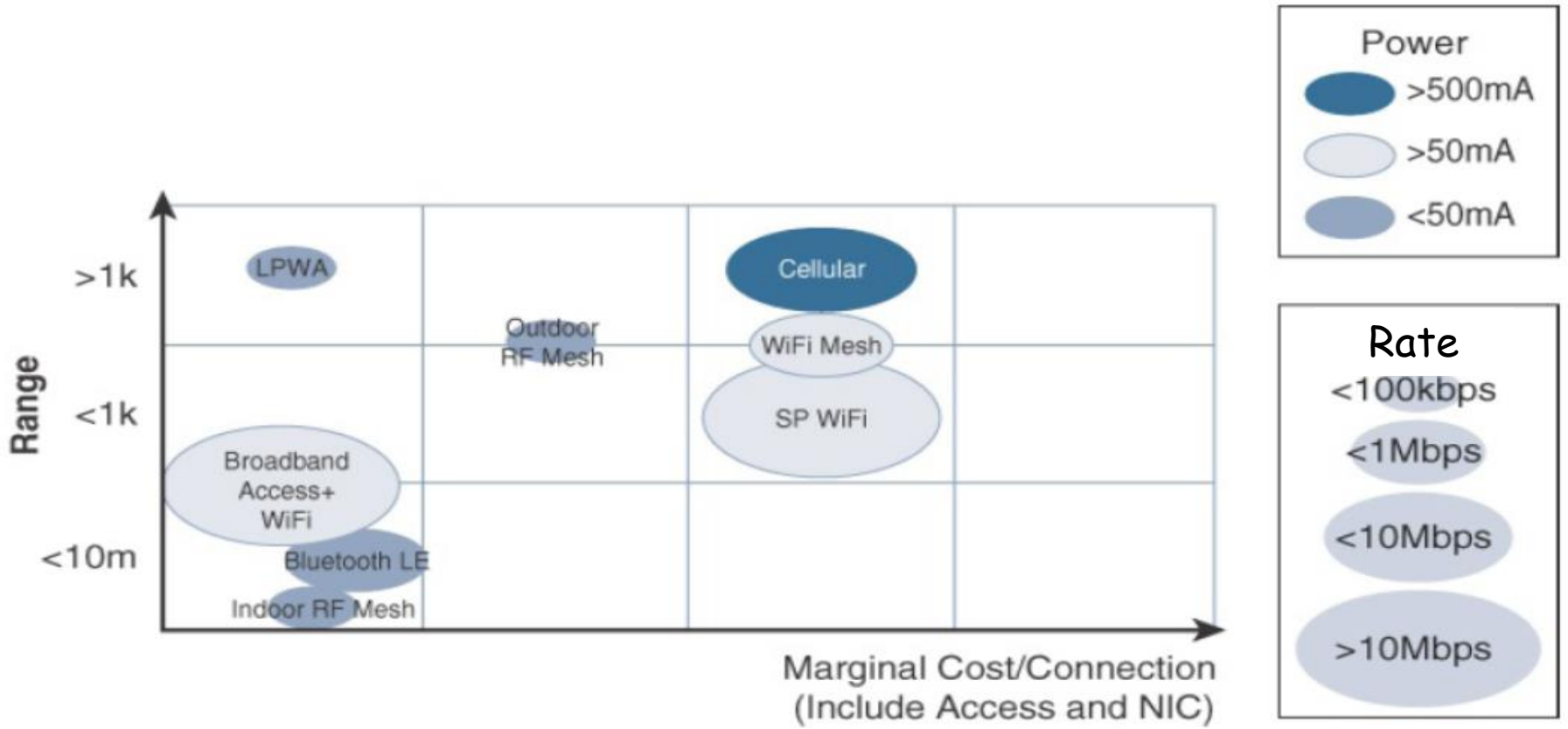
- The current draw of each model presented in the official Raspberry Pi magazine are in two modes: idle (standby) and load (run) modes. The current consumptions in the two power modes for each model are:
  - 0.1 and 0.25A for Raspberry Pi Zero;
  - 0.25 and 0.31A for the Raspberry Pi 1 B+;
  - 0.26 and 0.42A fortheRaspberryPi2;
  - 0.31 and 0.58A for the Raspberry Pi3

Raspberry Pi- Power Consumption

- Many networking technologies will not be a good fit with battery power.

- Frequency of communication does have an influence on power selection, too.

Comparison Between Common Last-Mile Technologies in Terms of Range Versus Power and Rate



Power Consumption

- While the power consumption of the IoT boards actually depends on the usage as well as the model of the device, a few best practice techniques that could be applied during operation to reduce power consumption include the following:
  1. Disconnect every peripheral that is not in use
  2. Switch-off Internet connectivity when not needed
  3. Shutdown the device or put it in dormant mode when not in use
  4. When using the device in headless mode (i.e., accessing it via network connections without a keyboard or display), the High-Definition Multimedia Interface (HDMI) could be switched off
  5. Avoid running several daemons at a time, and run only power efficient applications

* Internet of Things A to Z : Technologies and Applications, IEEE Press, 2018, (chapter 6: **(A Tutorial Introduction to IoT Design and Prototyping with Examples)**

Next Lecture

- An overview of
  - Physical and Link Layers Protocols (IoT Access Technologies)
  - Network Layer Protocols (IP as the IoT Network Layer)
  - Transport Layer Protocols
  - Application Layer Protocols