

SNJB's KBJ College of Engineering
Chandwad-423101 (Nashik)

Department
of
Information Technology

Subject : Internet of Things (IOT) of BE 2015
Pattern

Unit 5

IoT PLATFORMS

What is an IoT Device

- A "Thing" in IoT can be any object that has unique identifier and which can send/receive data over a network
- IoT devices are connected to the Internet and send information about themselves or about their surroundings over a network
- Allow actuation upon the physical entities/ environment around them remotely

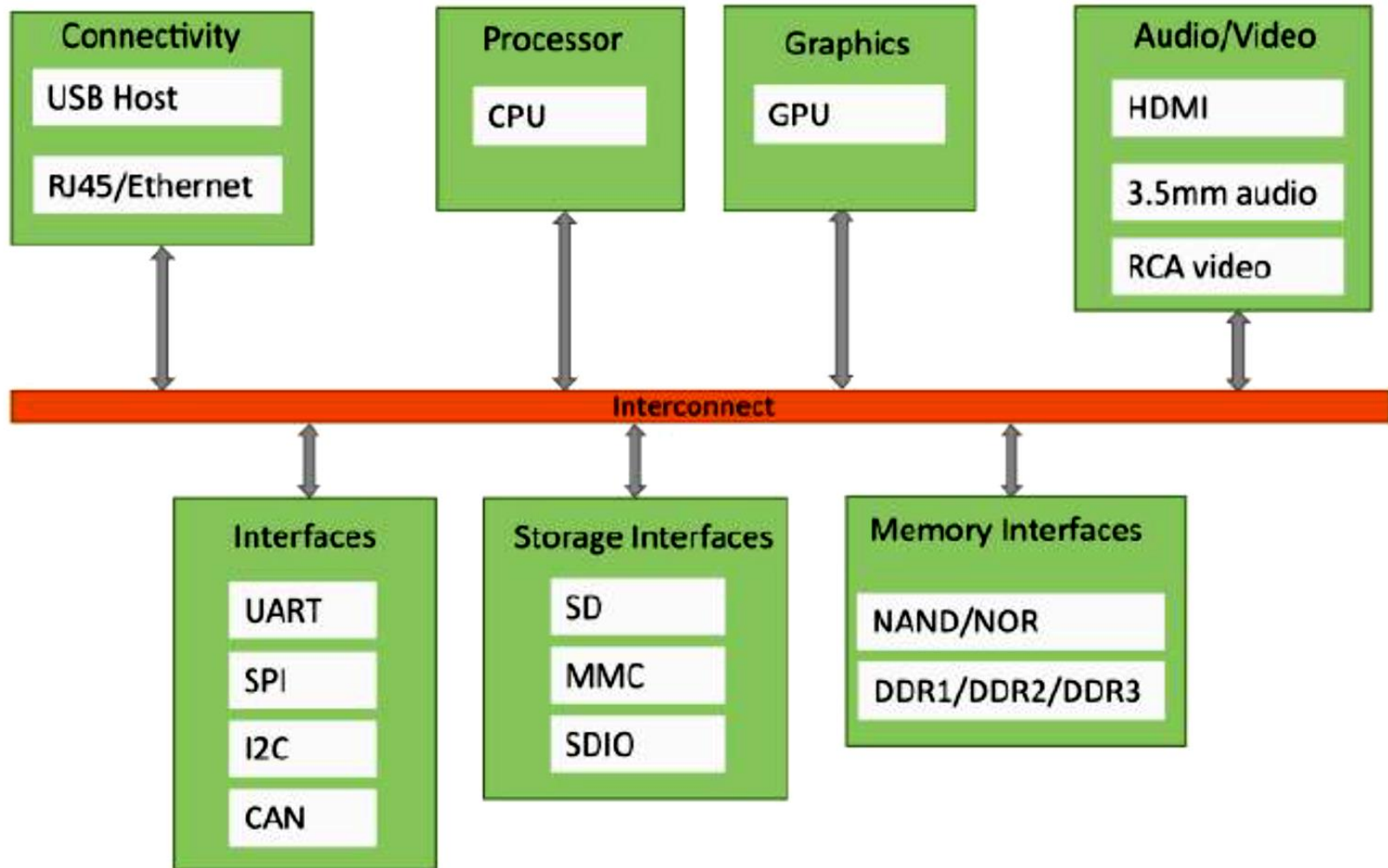
IoT Device Examples

- A home automation device
- An industrial machine
- A car
- A wireless-enabled wearable device

Basic building blocks of an IoT Device

- Sensing
 - Sensors can be either on-board the IoT device or attached to the device
- Actuation
 - IoT devices can have various types of actuators
- Communication
 - Exchanging of data between client-server or cloud
- Analysis & Processing

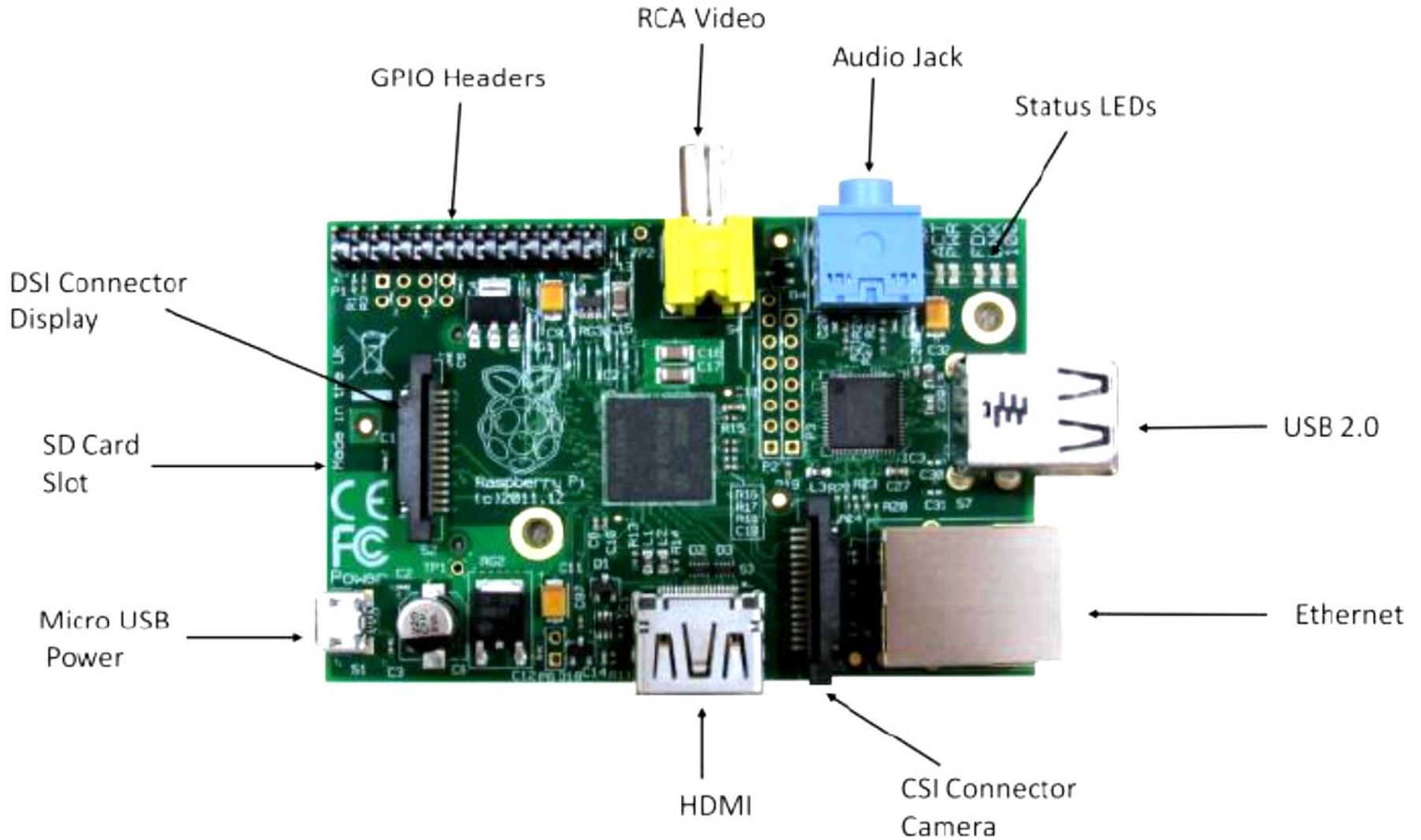
Block Diagram of an IoT Device



Exemplary Device: Raspberry Pi

- Low-cost mini-computer with the physical size of a credit card
- Runs various flavors of Linux and can perform almost all tasks that normal desktop can do
- Also allows interfacing sensors and actuators through the general purpose I/O (GPIO) pins
- Since Raspberry Pi runs Linux operating system, it supports Python

Raspberry Pi



Linux on Raspberry Pi

Raspbian

- Raspbian Linux is a Debian Wheezy port optimized for Raspberry Pi

Arch

- Arch is an Arch Linux port for AMD devices

Pidora

- Pidora Linux is a Fedora Linux optimized for Raspberry Pi

RaspBMC

- RaspBMC is an XBMC media-center distribution for Raspberry Pi

OpenELEC

- OpenELEC is a fast and user-friendly XBMC media-center distribution

RISC OS

- RISC OS is a very fast and compact operating system

Raspberry Pi GPIO



Alternate Function					Alternate Function
	3.3V PWR	1		2	5V PWR
I2C1 SDA	GPIO 2	3		4	5V PWR
I2C1 SCL	GPIO 3	5		6	GND
	GPIO 4	7		8	UART0 TX
	GND	9		10	UART0 RX
	GPIO 17	11		12	GPIO 18
	GPIO 27	13		14	GND
	GPIO 22	15		16	GPIO 23
	3.3V PWR	17		18	GPIO 24
SPI0 MOSI	GPIO 10	19		20	GND
SPI0 MISO	GPIO 9	21		22	GPIO 25
SPI0 SCLK	GPIO 11	23		24	GPIO 8
	GND	25		26	GPIO 7
	Reserved	27		28	Reserved
	GPIO 5	29		30	GND
	GPIO 6	31		32	GPIO 12
	GPIO 13	33		34	GND
SPI1 MISO	GPIO 19	35		36	GPIO 16
	GPIO 26	37		38	GPIO 20
	GND	39		40	GPIO 21
					SPI0 CS0
					SPI0 CS1
					SPI1 CS0
					SPI1 MOSI
					SPI1 SCLK

Raspberry Pi Interfaces

Serial

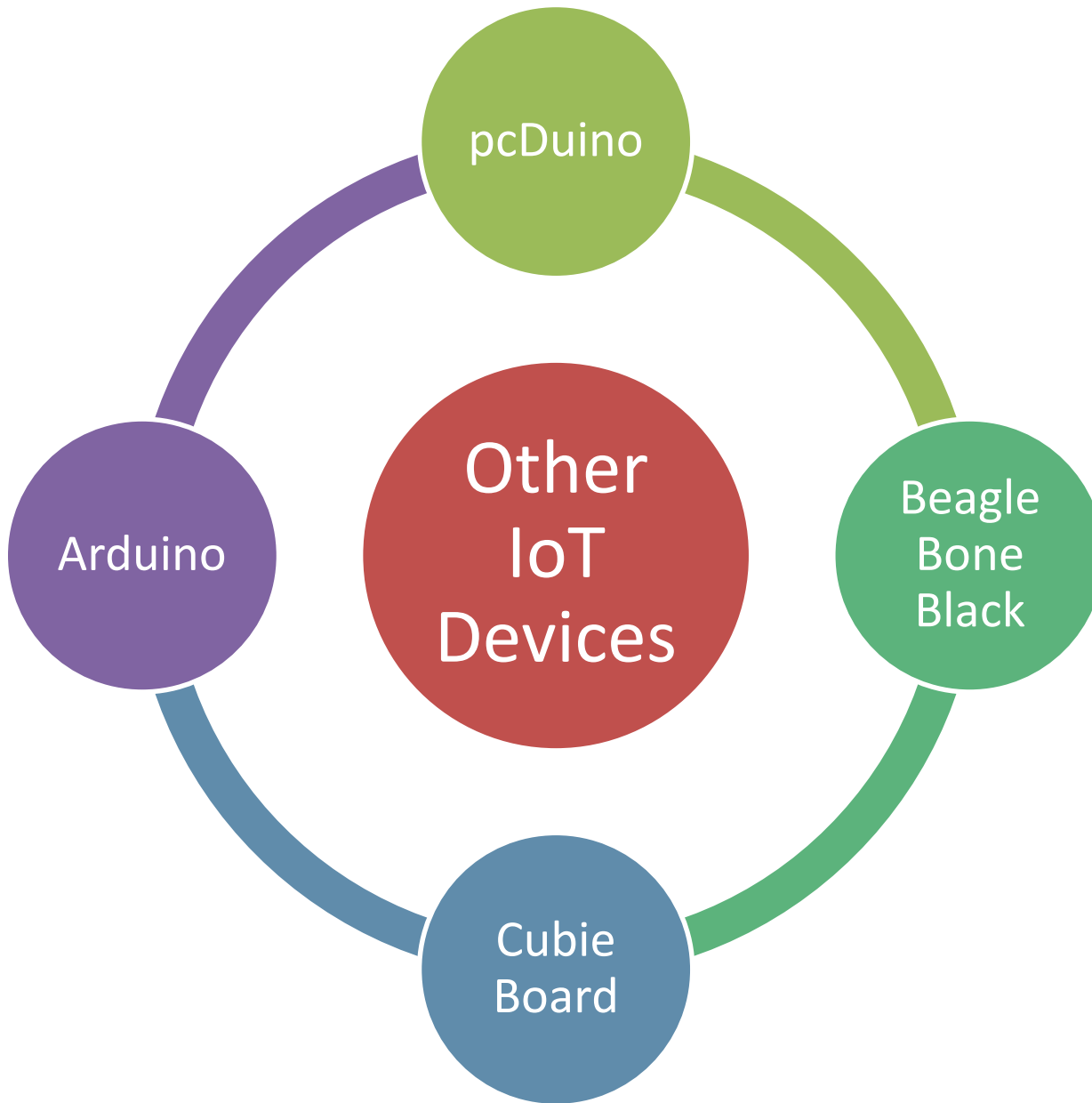
- Has receive (Rx) and transmit (Tx) pins for communication with serial peripherals

SPI

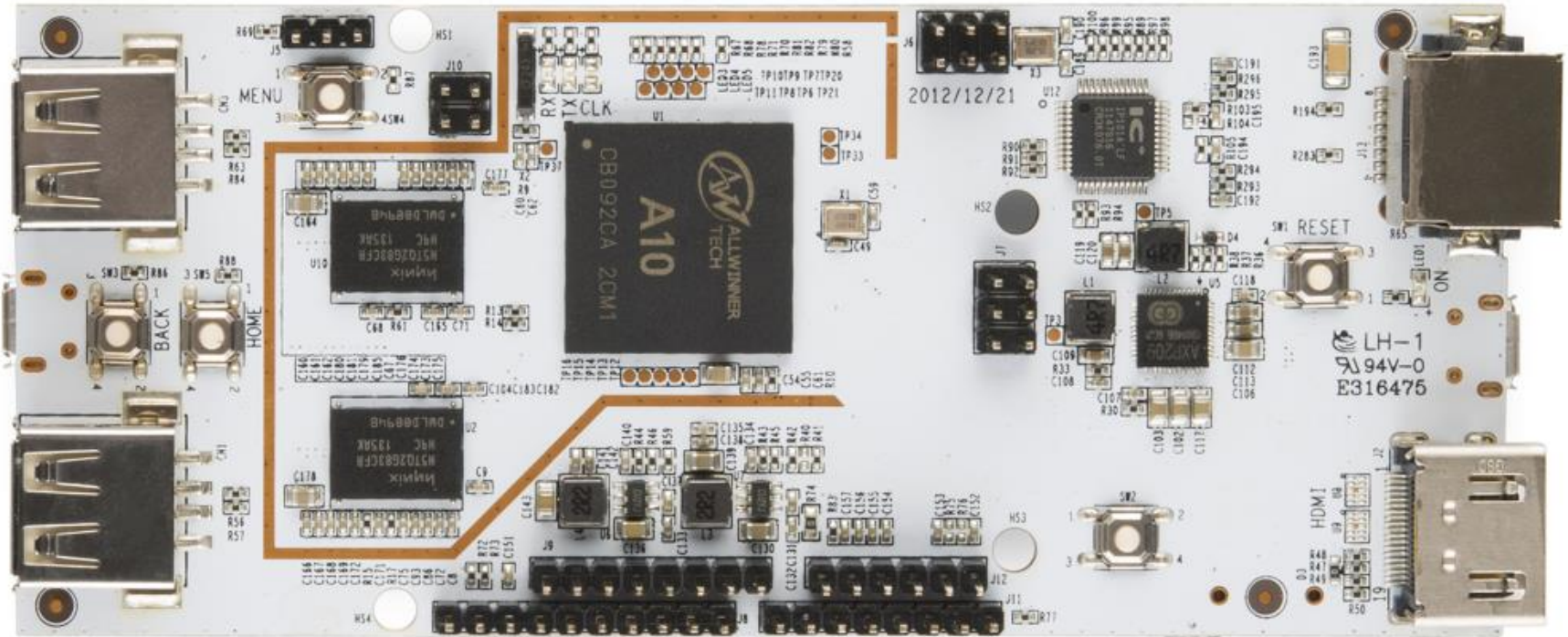
- Serial Peripheral Interface
- Synchronous serial data protocol used for communicating with one or more peripheral devices

I2C

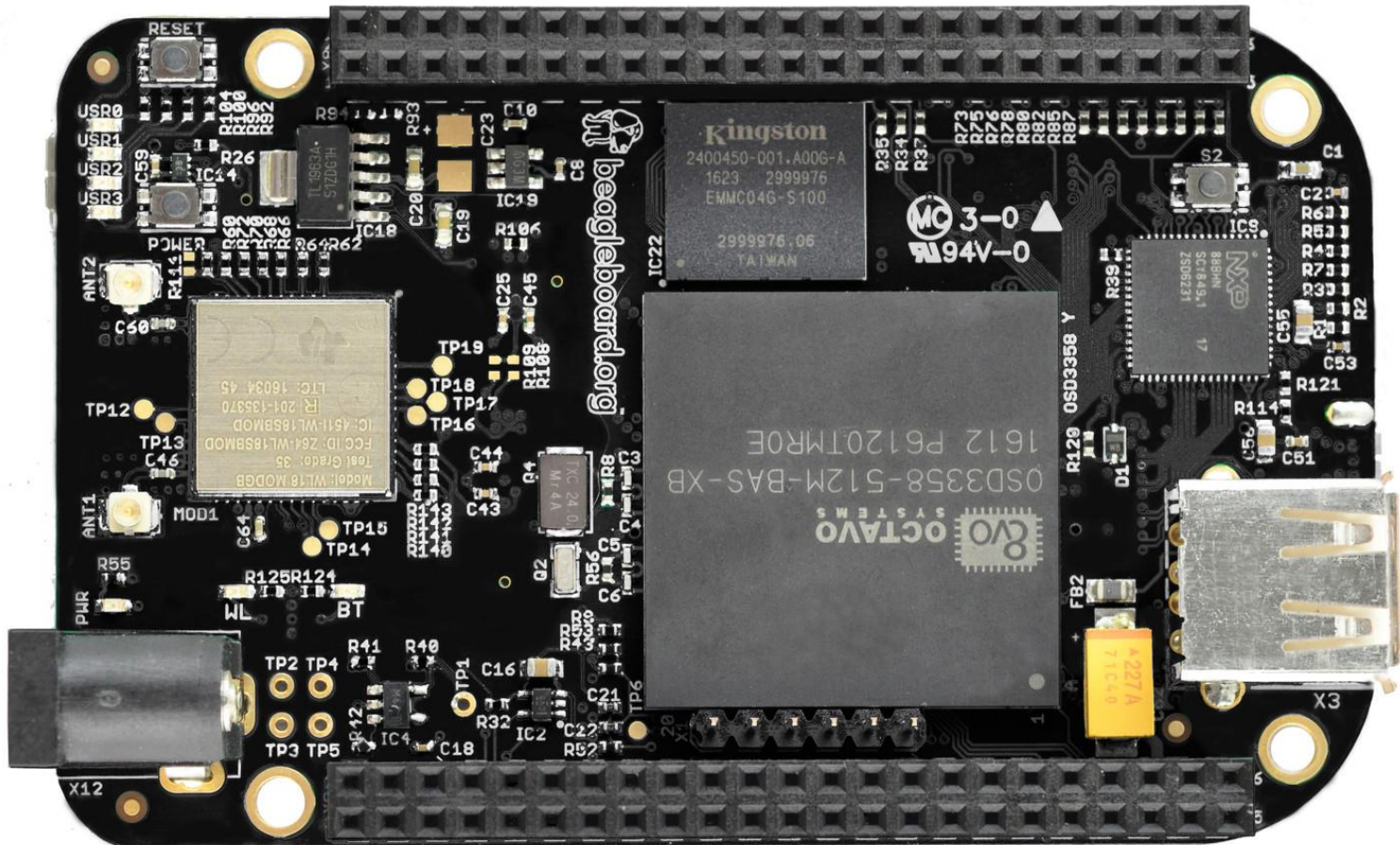
- Allows you to connect hardware modules
- Allows synchronous data transfer with just two pins - SDA (data line) and SCL (clock line)



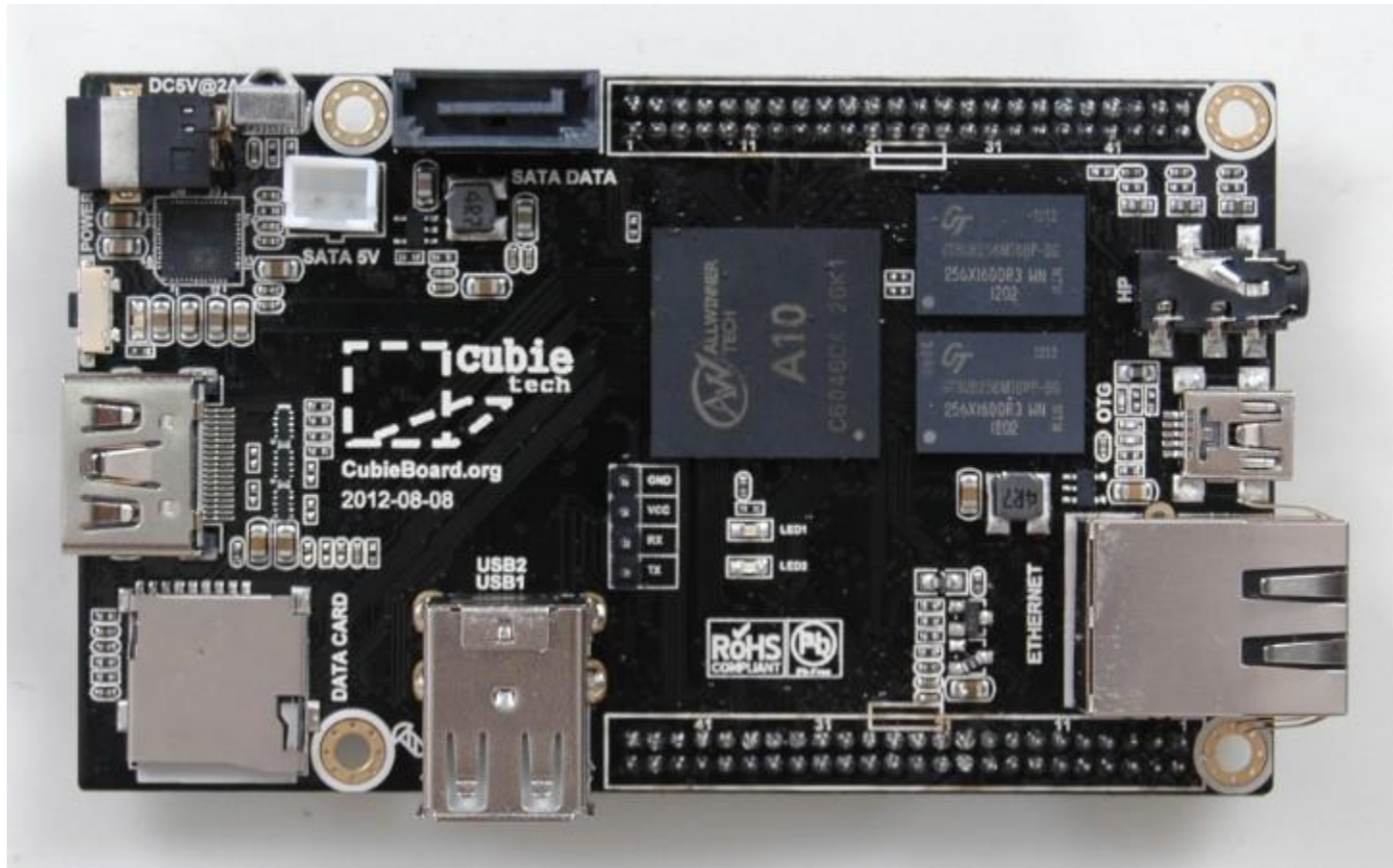
pcDuino



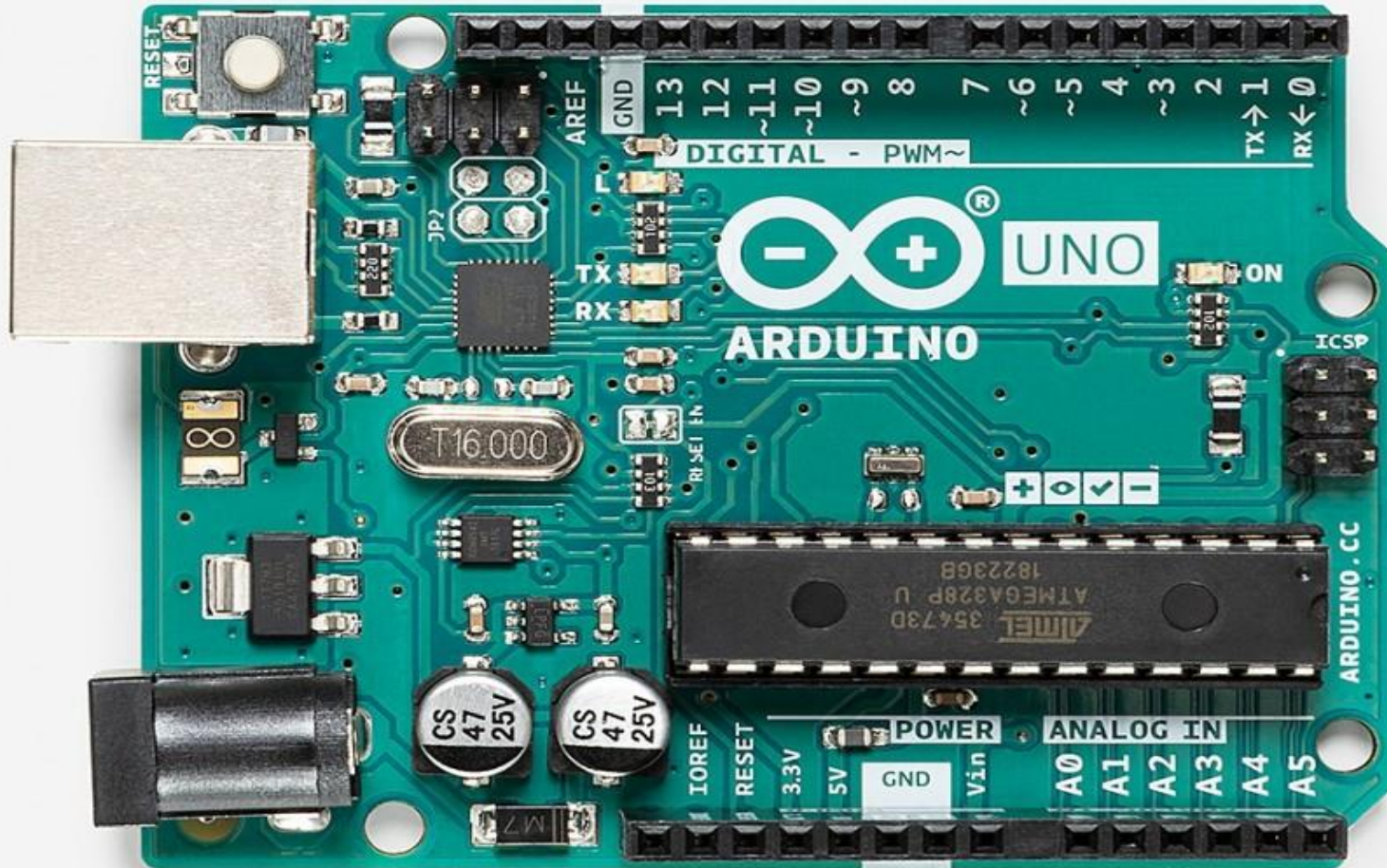
Beagle Bone Black



Cubie Board



Arduino



What is Arduino?

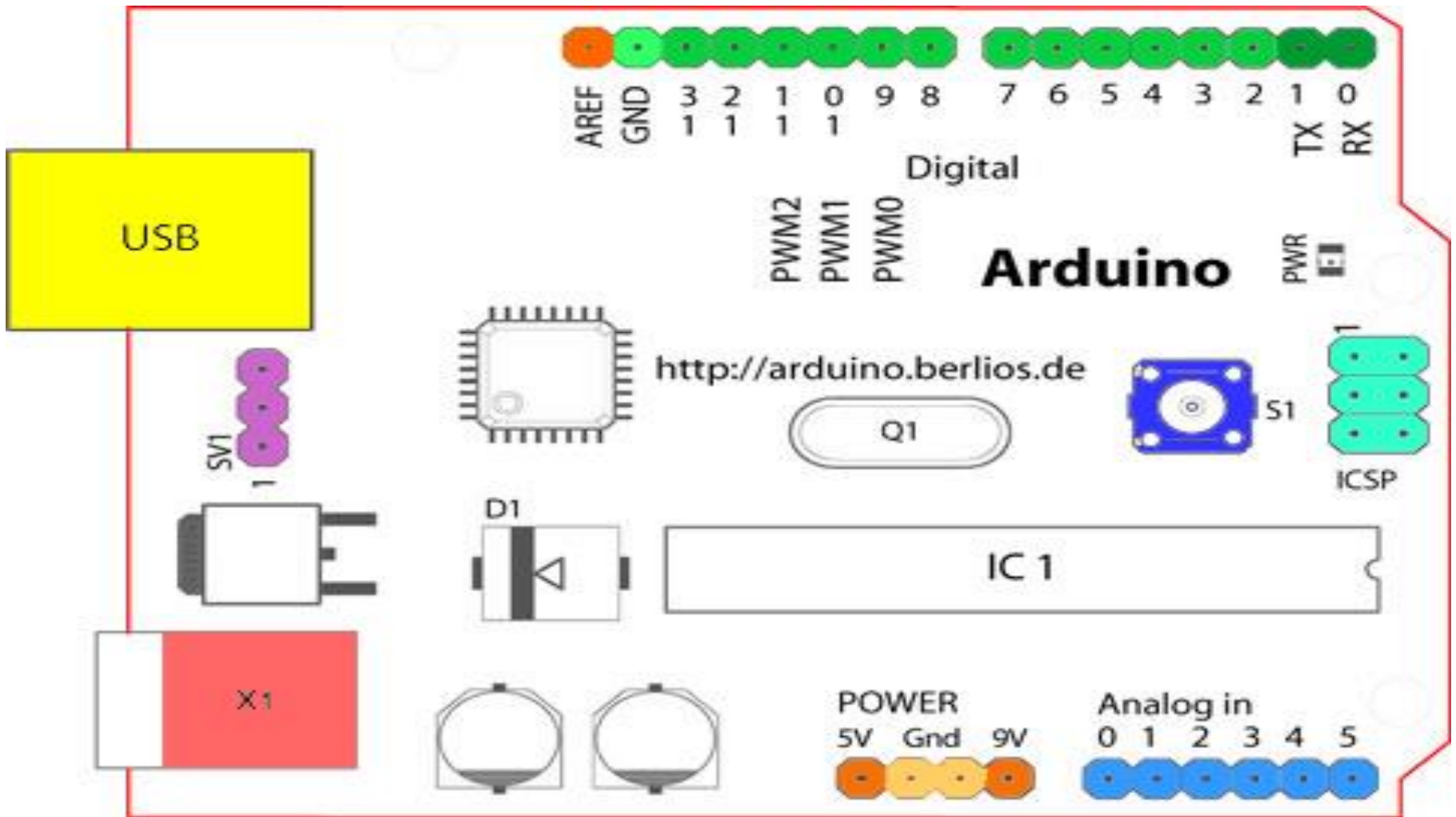
- Open-source electronics platform based on easy-to-use hardware and software.
- Able to read inputs - light on a sensor, a finger on a button, or a Twitter message
- Turn input into an output - activating a motor, turning on an LED, publishing something online.
- You can tell your board what to do by sending a set of instructions to the microcontroller on the board.
- To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

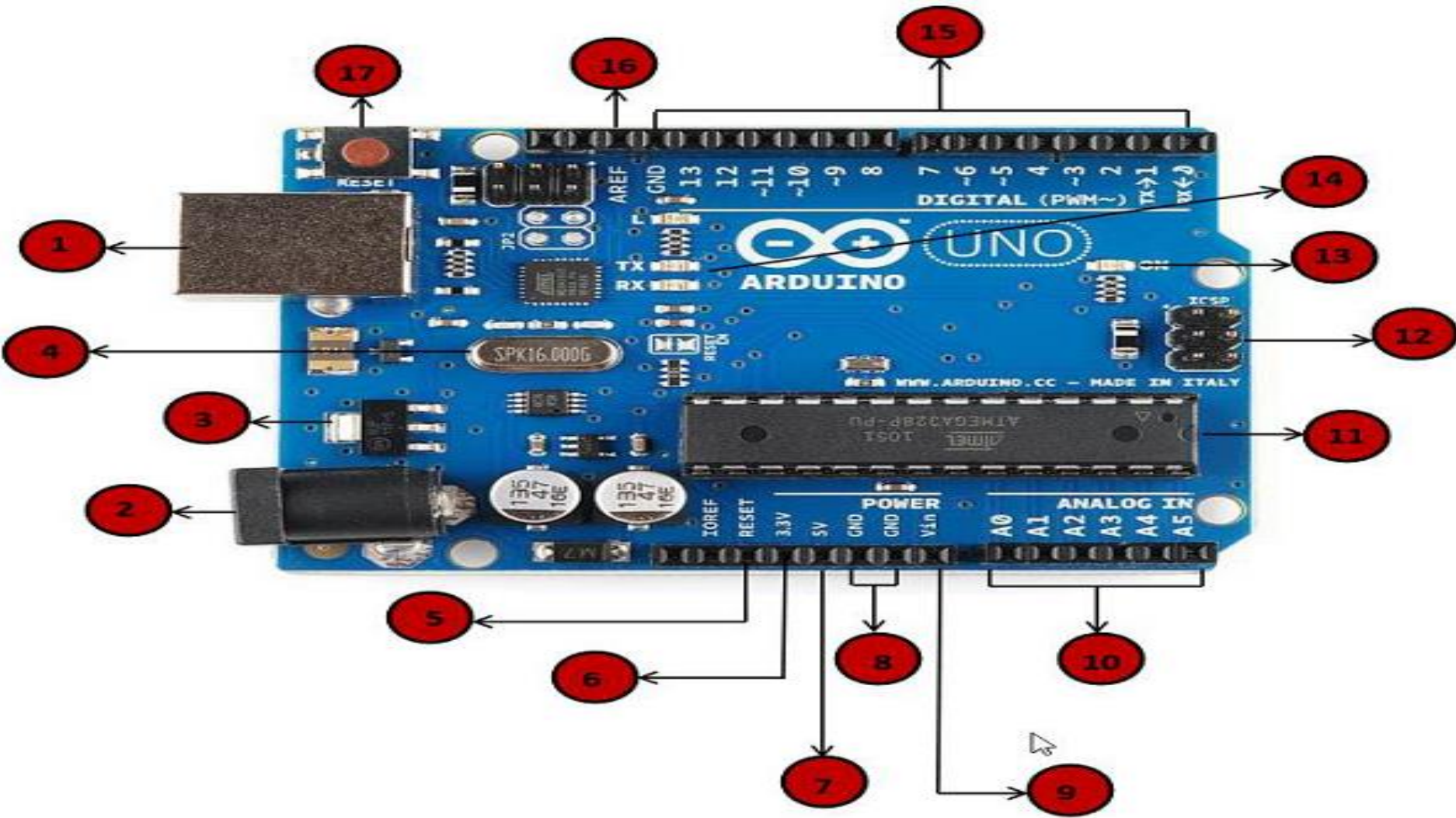
Why Arduino?

- Simple and accessible user experience
- Arduino software is easy-to-use for beginners,
- Flexible enough for advanced users.
- Runs on Mac, Windows, and Linux.
- Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics.

Why so popular?

- Inexpensive
- Cross-platform
- Simple, clear programming environment
- Open source and extensible software
- Open source and extensible hardware





Microcontrollers

- ATmega328P (used on most recent boards)
- Digital I/O Pins - 14 (of which 6 provide PWM output)
- Analog Input Pins - 6 (DIP) or 8 (SMD)
- DC Current per I/O Pin - 40 mA
- Flash Memory - 32 KB
- SRAM - 2 KB
- EEPROM - 1KB

Digital Pins

- Digital pins on an Arduino board can be used for general purpose input and output via the
 - `pinMode()`
 - `digitalRead()`
 - `digitalWrite()`
- Each pin has an internal pull-up resistor which can be turned on and off using `digitalWrite()` (w/ a value of HIGH or LOW, respectively) when the pin is configured as an input.
- The maximum current per pin is 40 mA.

Digital Pins

- Serial: 0 (RX) and 1 (TX).
 - Used to receive (RX) and transmit (TX) TTL serial data.
 - On the Arduino Diecimila, these pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip.
 - On the Arduino BT, they are connected to the corresponding pins of the WT11 Bluetooth module.
 - On the Arduino Mini and LilyPad Arduino, they are intended for use with an external TTL serial module (e.g. the Mini-USB Adapter).
- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.

Digital Pins

- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the `analogWrite()` function. On boards with an ATmega8, PWM output is available only on pins 9, 10, and 11.
- BT Reset: 7. (Arduino BT-only) Connected to the reset line of the bluetooth module.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- LED: 13. On the Diecimila and LilyPad, there is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

Analog Pins

- In addition to the specific functions listed below, the analog input pins support 10-bit analog-to-digital conversion (ADC) using the `analogRead()` function.
- Most of the analog inputs can also be used as digital pins:
 - Analog input 0 as digital pin 14 through analog input 5 as digital pin 19.
 - Analog inputs 6 and 7 (present on the Mini and BT) cannot be used as digital pins.

Power Pins

- VIN (sometimes labelled "9V")
- 5V
- 3V3
- GND

Other Pins

- AREF
 - Reference voltage for the analog inputs. Used with `analogReference()`.
- Reset
 - Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Difference between Raspberry Pi and Arduino

Parameters	Raspberry Pi 4	Raspberry Pi Zero W	Arduino Uno	Arduino Portenta H7
Price	From \$35	\$10	\$23	\$101.88
RAM	1GB to 8GB	512MB	2KB SRAM	Base model: 8MB, Custom: 64MB
CPU / Microcontr oller	1.5GHz quad-core ARMv8-A	1GHz ARM1176JZF-S	16MHz ATmega328P	STM32H747XI dual Cortex-M7+M4 32bit low power Arm MCU
GPIO	40	40	14 including, 6 Analog inputs, 6 PWM	28 including, 7 Analog inputs
Storage	Micro SD, USB	Micro SD, USB	32KB Flash memory	Base model: 16MB Flash, Custom: Up to 128MB Flash storage
Network	Gigabit Ethernet, Wi-Fi b/g/n & 5GHz AC	802.11 b/g/n wireless LAN, Bluetooth 4.1, Bluetooth Low Energy (BLE)	Requires Shield Add On	Murata 1DX dual Wi-Fi 802.11b/g/n 65 Mbps and Bluetooth 5.1
Power	5V 3A	5V	6 - 20V via DC input	5V via USB C or VIN pin