

Introduction to Arduino Hardware

Objectives

- ▣ Understand the Hardware associated with the Arduino
- ▣ Start to become familiar with vocabulary associated with Arduino

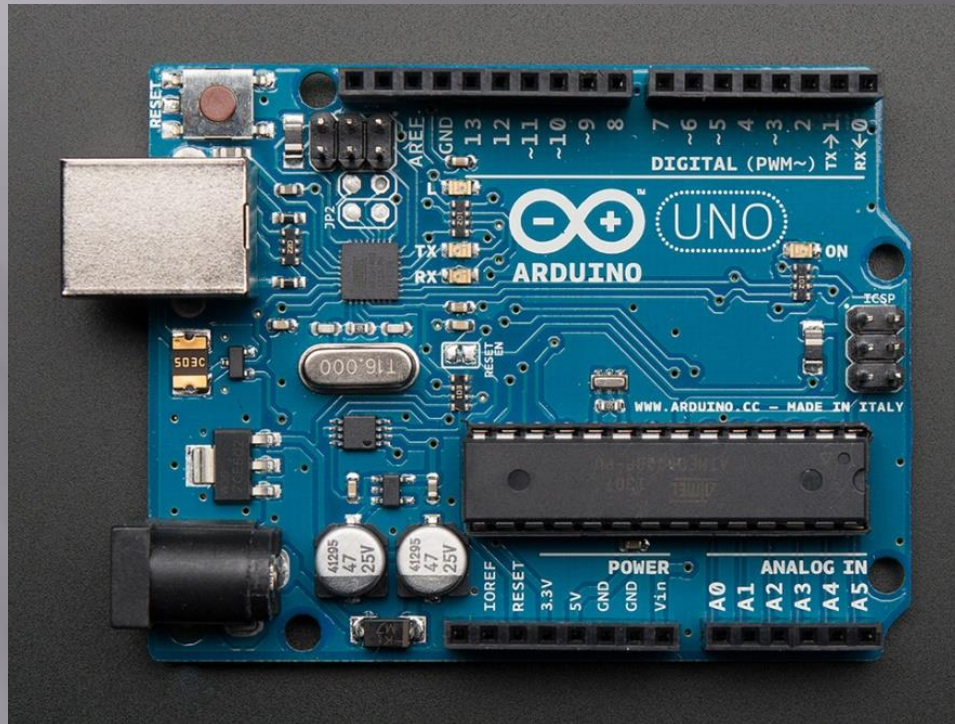
Physical Computing

- ▣ What makes this class different than other programming classes?
- ▣ *Physical computing*, in the broadest sense, means building interactive **physical** systems by the use of software and hardware that can sense and respond to the *analog* world.
 - Wikipedia

Questions for design

- ▣ What do you want to interface with?
- ▣ What do you want to control?
- ▣ What type of interface do you need?

The Arduino – what is it?



An open-source platform for developing interactive programs that control external hardware, like LED's, switches, or even communicating over the internet.

The Arduino: Why use it?

- ▣ Open source – the software is free, along with hardware designs
- ▣ It is a relatively easy language to learn
- ▣ The cost is low
- ▣ There are LOTS of online resources
- ▣ There are lots of versions for adding more complexity

- ▣ *Let's take a look at some of the hardware...*

It all starts with a microcontroller

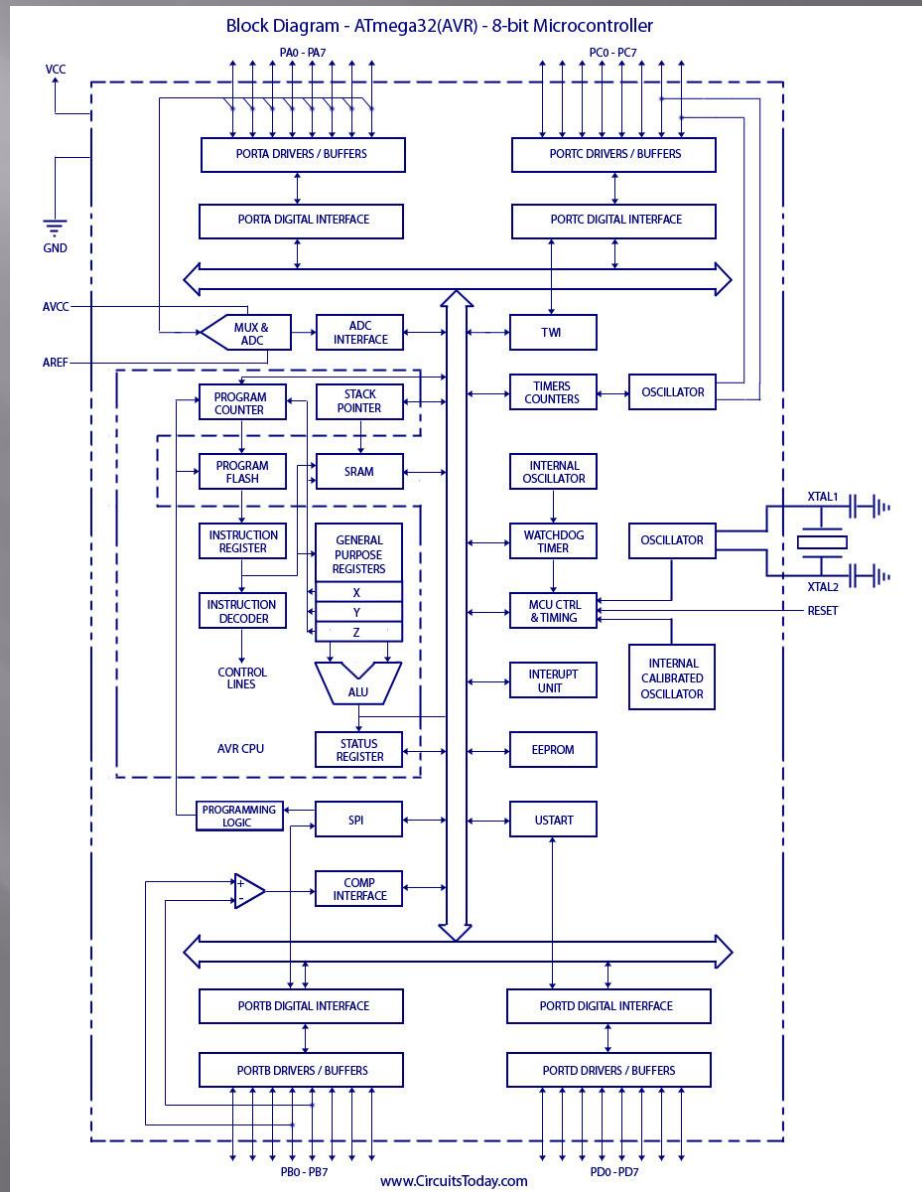
- ▣ Atmel Atmega 328
- ▣ 28 Pin Microcontroller



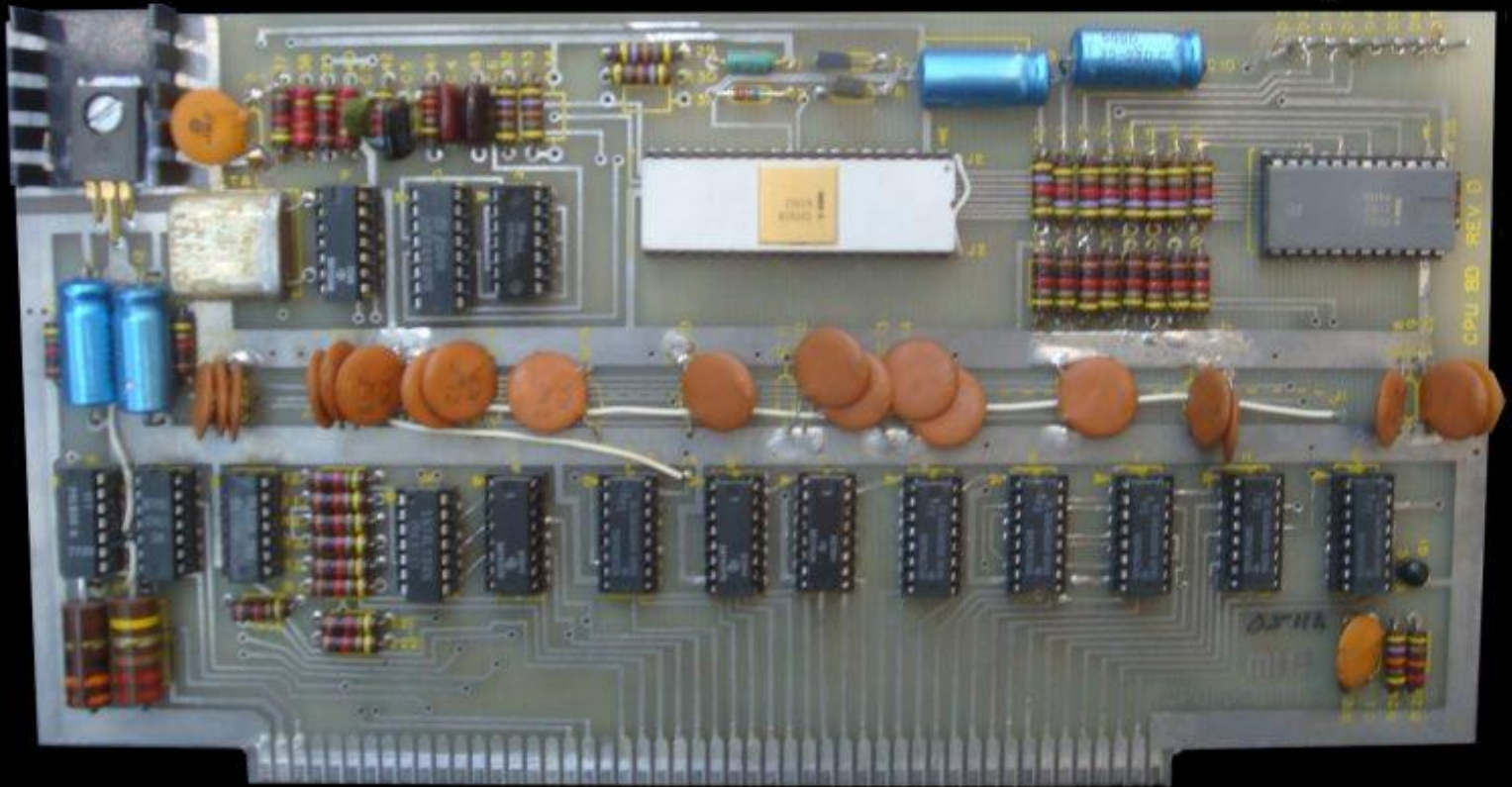
What makes it different than a microprocessor?

- ▣ Microprocessor:
 - ALU (Arithmetic Logic Unit)
 - Clock (timing) circuit
 - Local memory (registers)
- ▣ A Microcontroller adds Memory and I/O to make a complete “System on a chip”

What is in the '328?

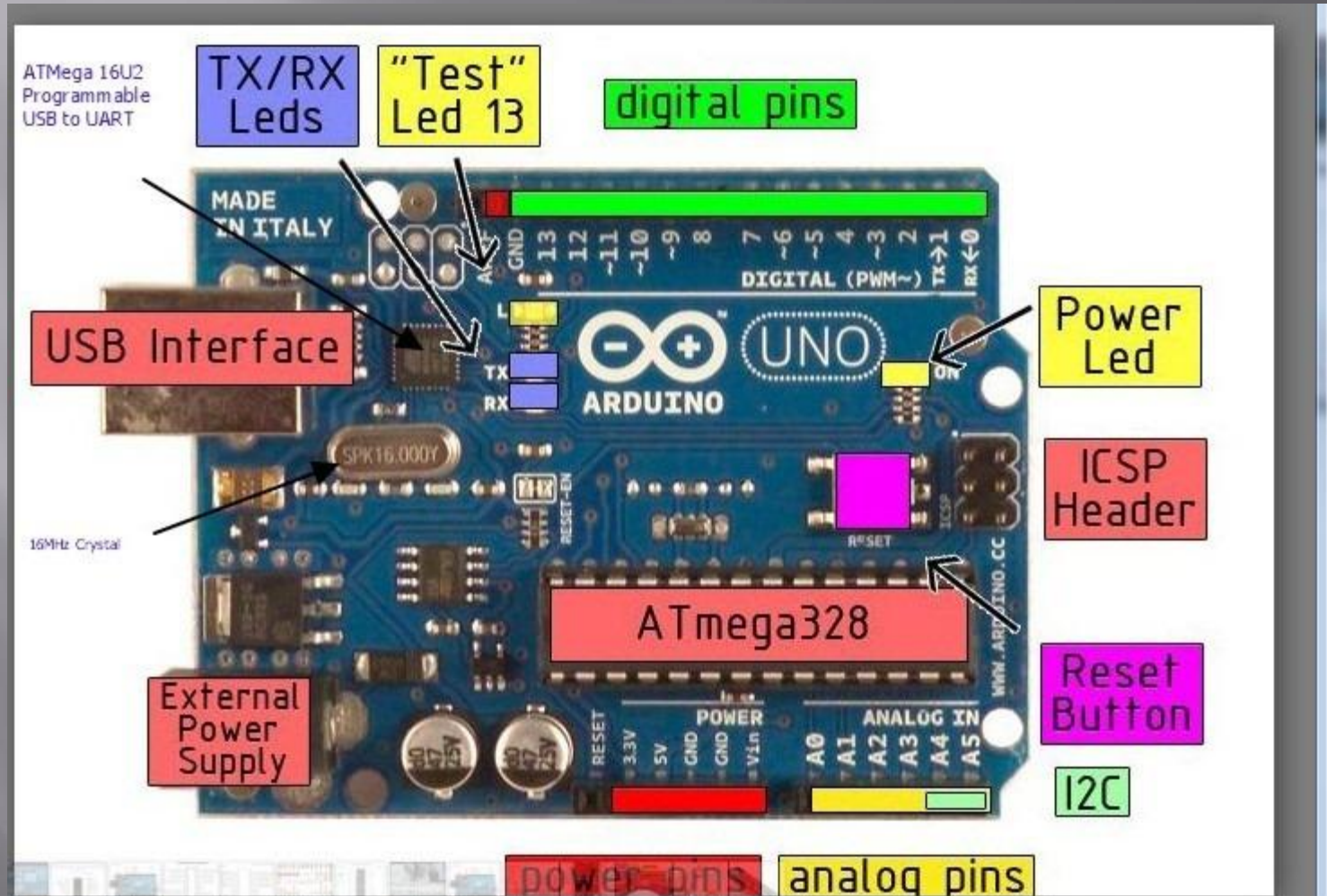


The same “part” – circa 1975



ALTAIR 8080 CPU Board

Arduino - a development board



Terms associated with Arduino / Microcontrollers

- ▣ **Hardware:** Physical components on the Arduino board

- ▣ **Firmware:** Code or programs that are stored in the memory of the microcontroller

Terms associated with Arduino / Microcontrollers

▣ *Memory*

- ▣ **RAM:** *Volatile* (temporary) – used for short term variables
- ▣ **FLASH:** *Non-volatile* (indefinite, can be modified) Used for constants, text, etc.

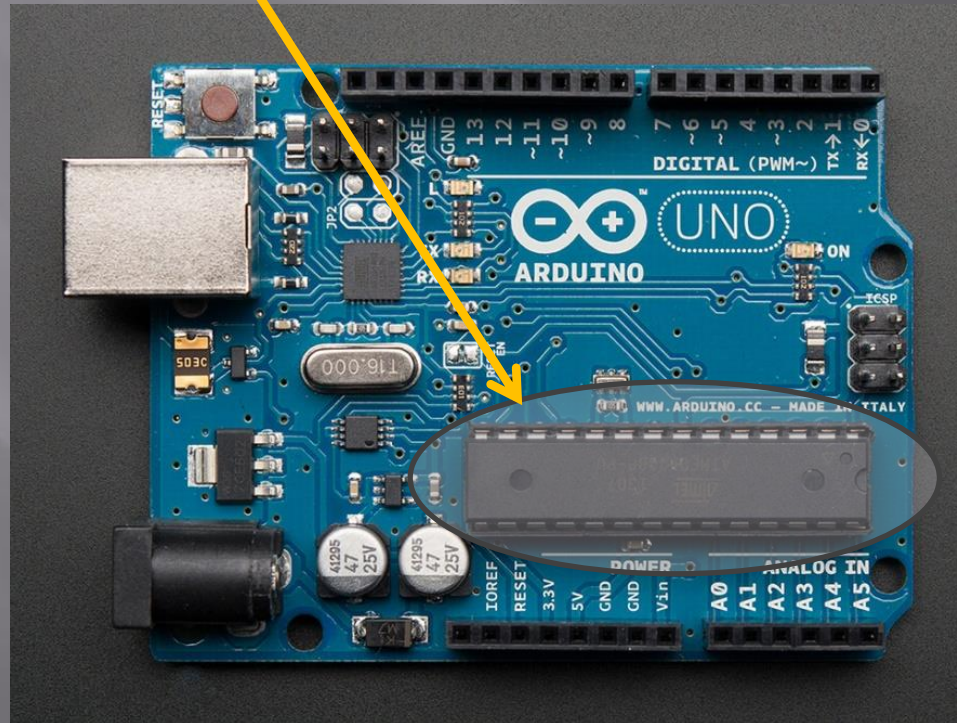
▣ **I/O** (Inputs & outputs)

- ▣ **Digital** (discrete, zeroes and ones): Hi / Low voltage values (0 or 5 volts)
- ▣ **Analog** (continuous): A range of voltage values, typically between 0 and 5 volts.



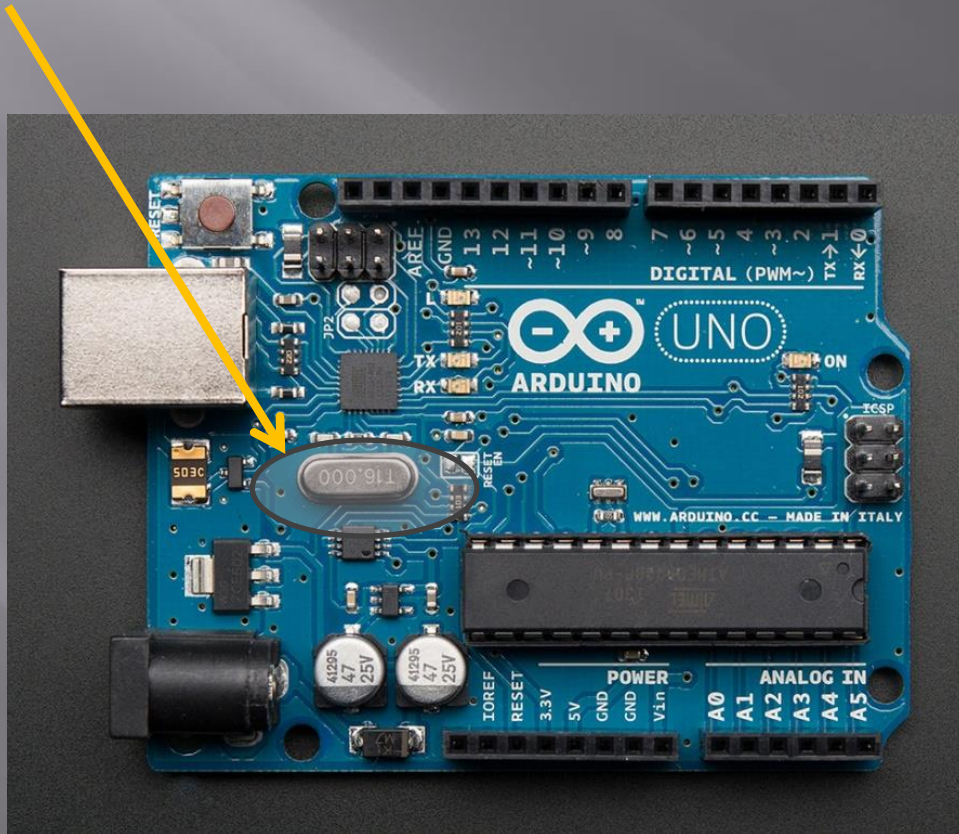
Arduino Hardware

- ▣ Microcontroller – Atmel 328. The “brain” similar to a microprocessor, but includes “peripherals,” memory and I/O



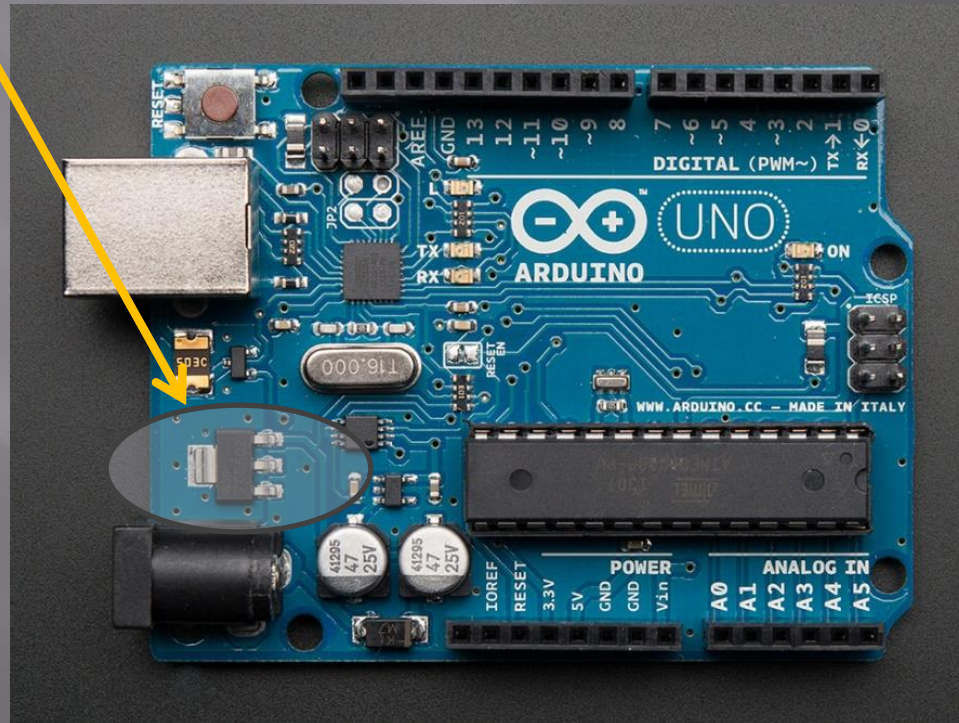
Crystal

- Crystal - “The heart” provides a precise clock to the Arduino - typically 16 MHz



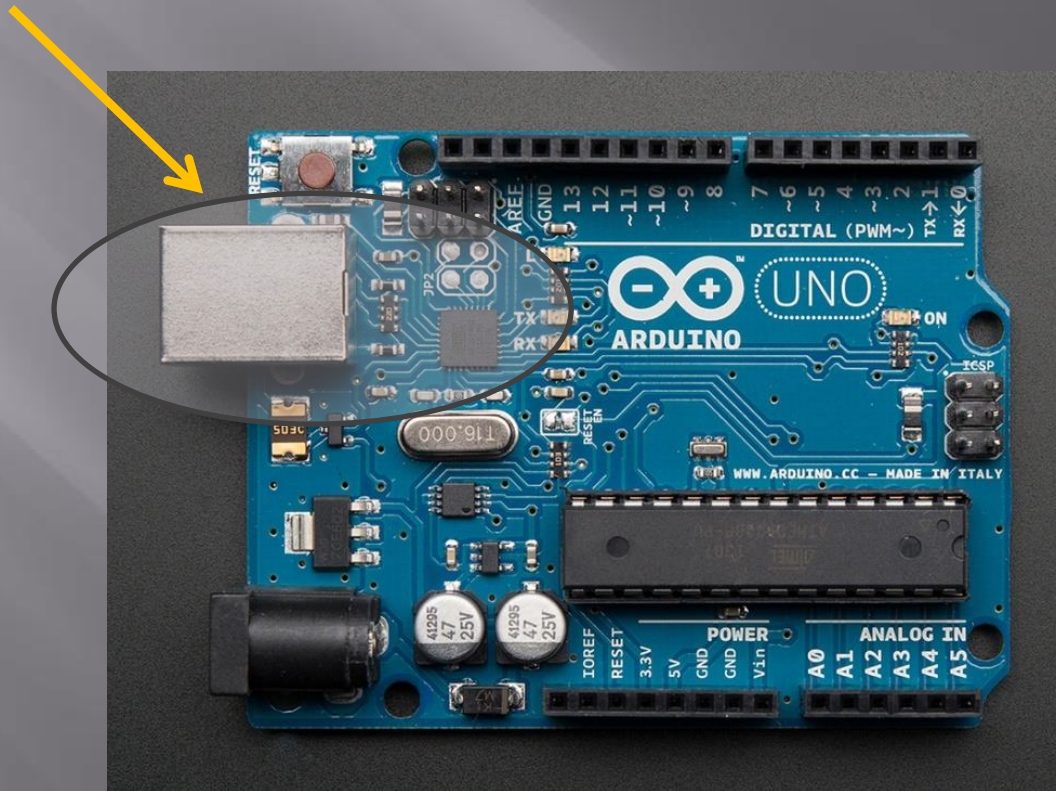
Voltage Regulator

Reduces 9-12 Volts in from external power supply (adapter) to 5V



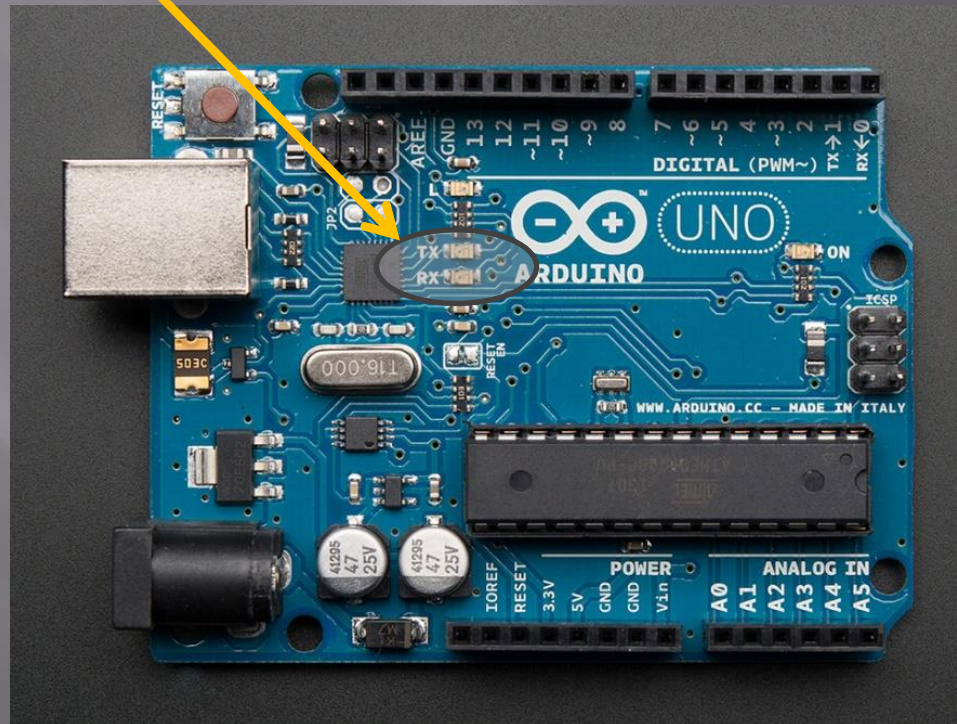
USB Interface

User interface to program the Arduino / use as a terminal to send / receive user input



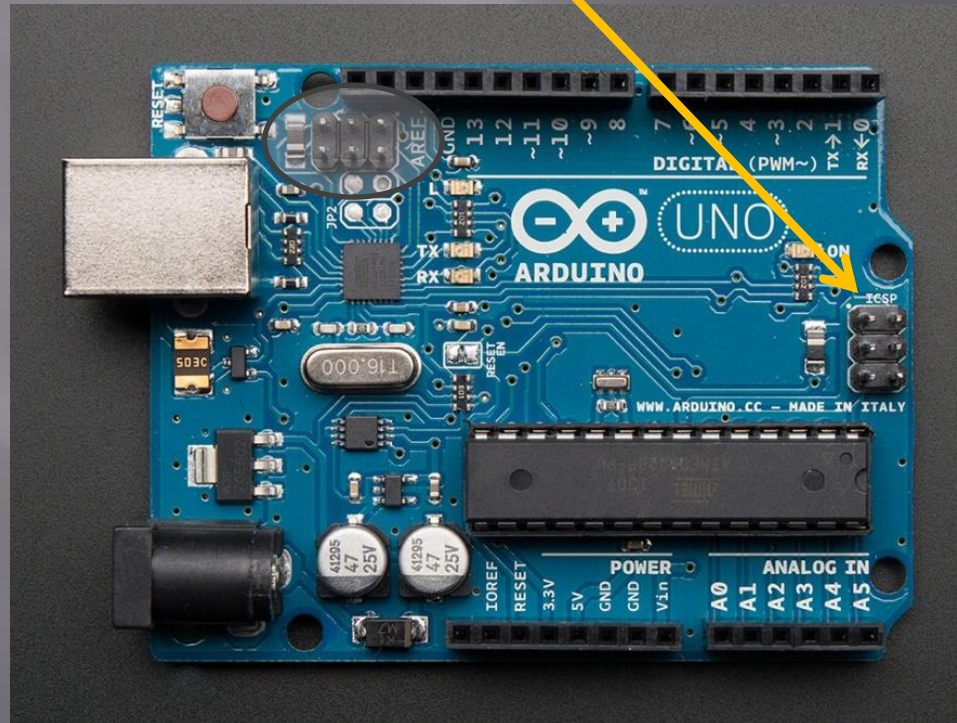
Transmit / Receive LED's

Used to indicate communication – should flash when Uploading or sending data to and from programs.



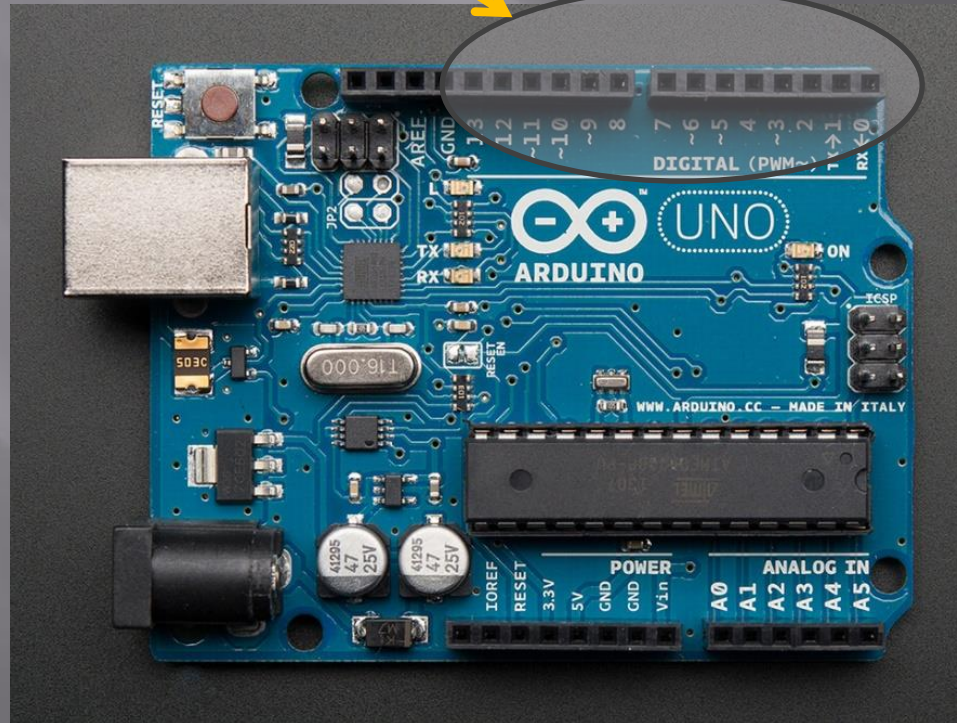
Programming header

ICSP Header – “In-Circuit Serial Programming”
for hardware programming / interfacing with
dedicated programmers

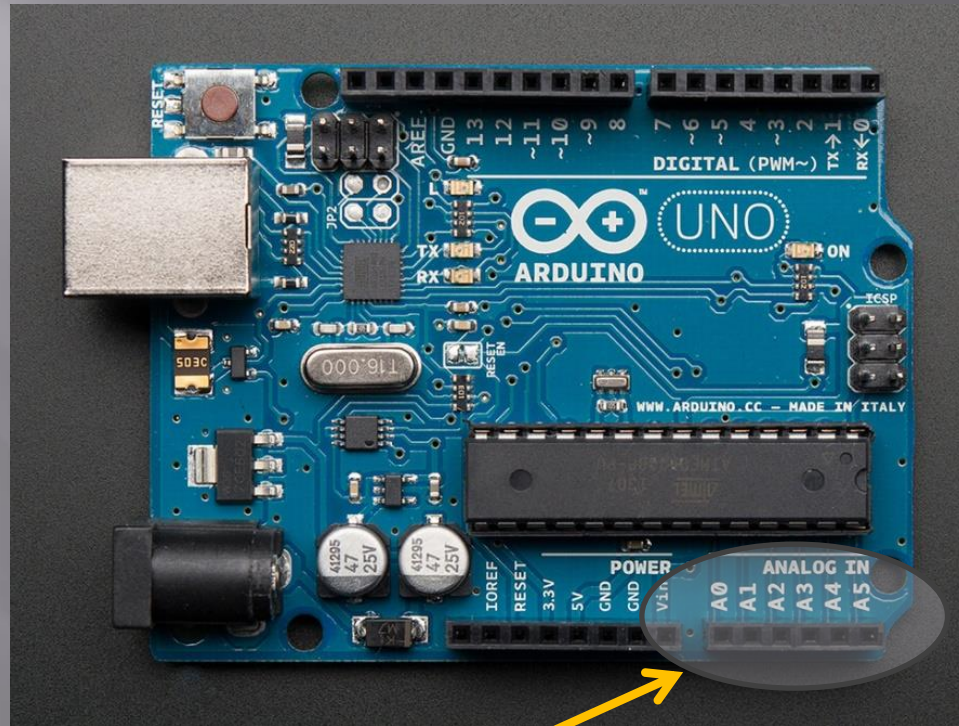


Headers - digital

Headers – sockets used to connect the Arduino to other hardware for *digital* interface

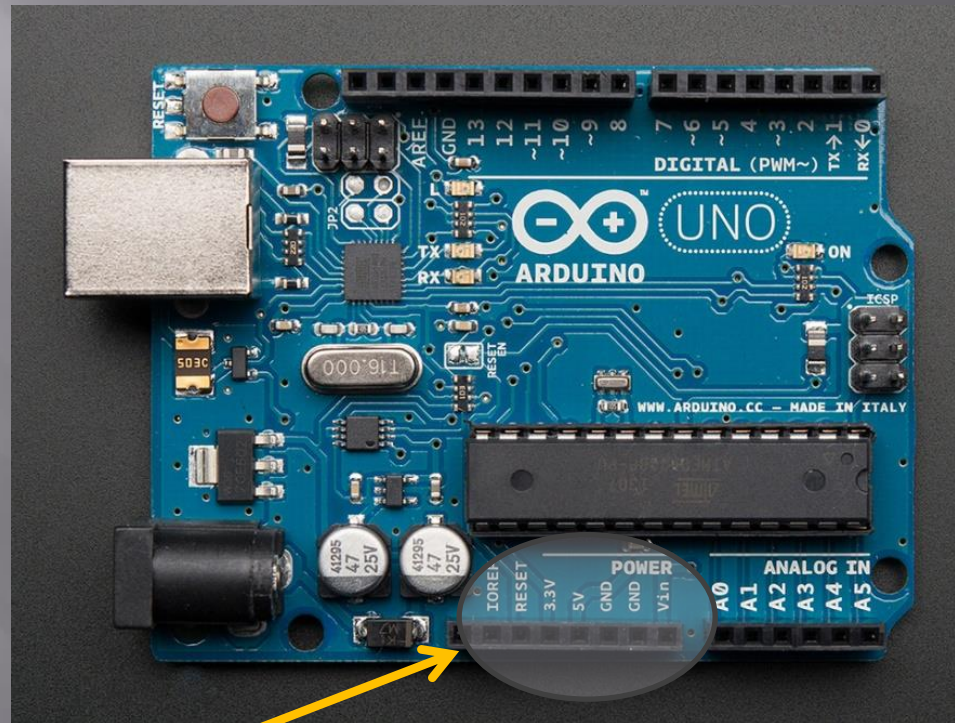


Headers - Analog



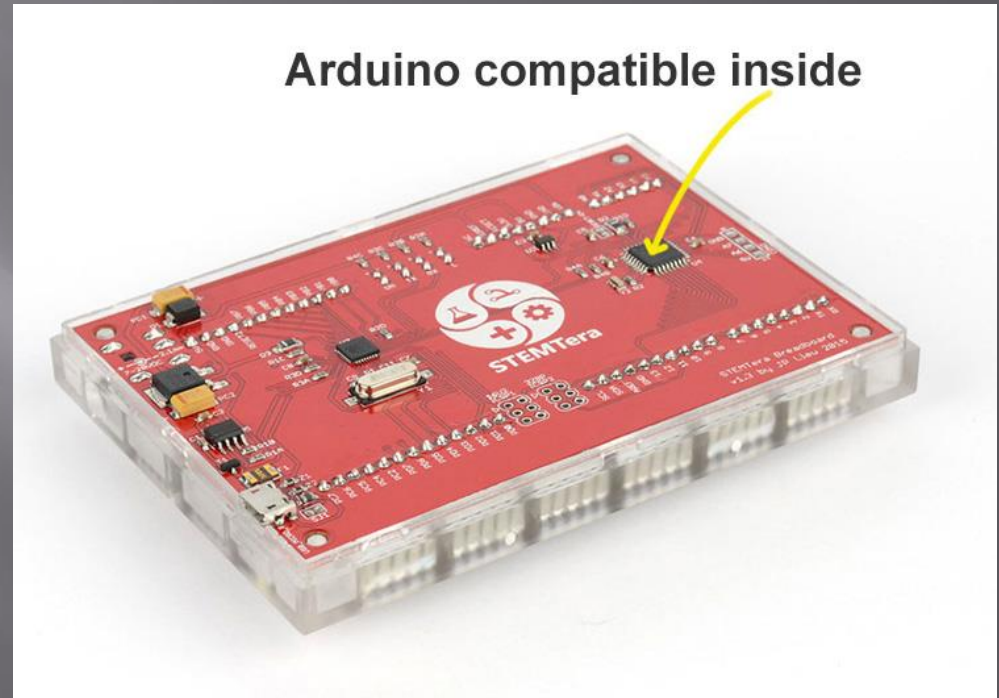
Analog

Power Inputs / Reset



Power / Reset

The STEMtera board



□ <http://learn.stemtera.com/en/begin/>

So, how is it programmed?

- ▣ Compilers convert a high level language (i.e. “C”) to the machine code that the microcontroller understands:

C language	Assembly Language	Machine Code
delay(1000);	MAIN_LOOP: LDA 10	0000: F7 00 10
	FST_LOOP: LDB 1000	0003: F6 10 00
	IN_LOOP: DBNZ	0007: D4 00 03
	DBNZ	000A: D3 00 02

Old school



Intel SDK 8085 Development system

