

CSE423: Embedded System Summer-2020

Introduction to Arduino



ARDUINO
OPEN-SOURCE
COMMUNITY

Today's Lecture



- *Arduino Overview*
- *Arduino Board Types*
- *Arduino specifications*
- *Terminology*

What is Arduino ?



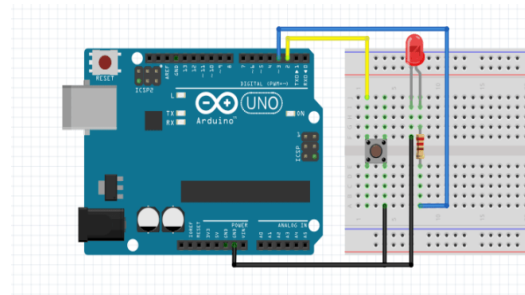
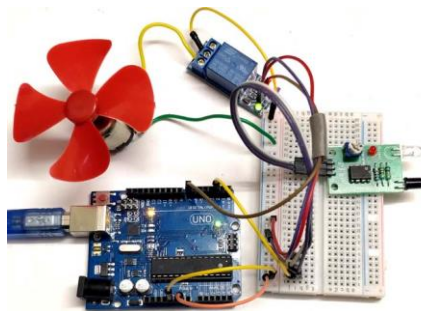
Arduino is a prototype platform (opensource) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.



What Arduino can do?



Arduino boards are able to read analog or digital input signals from different sensors and act accordingly to generate some output like activating a motor, turning LED on/off, connect to the cloud and many other actions.










Arduino Board Types



https://www.sparkfun.com/standard_arduino_comparison_guide

Arduino Board Types (Standard)



<p>Arduino Uno - R3</p> 	<p>Arduino Pro Mini - 5V/16Ml</p> 
<p>Arduino Uno - R3 SMD</p> 	<p>Arduino Fio</p> 
<p>Arduino Leonardo</p> 	<p>Arduino Pro Mini - 3V/8MH:</p> 
<p>Arduino Mega 2560 R3</p> 	

Arduino Board Types (contd.)



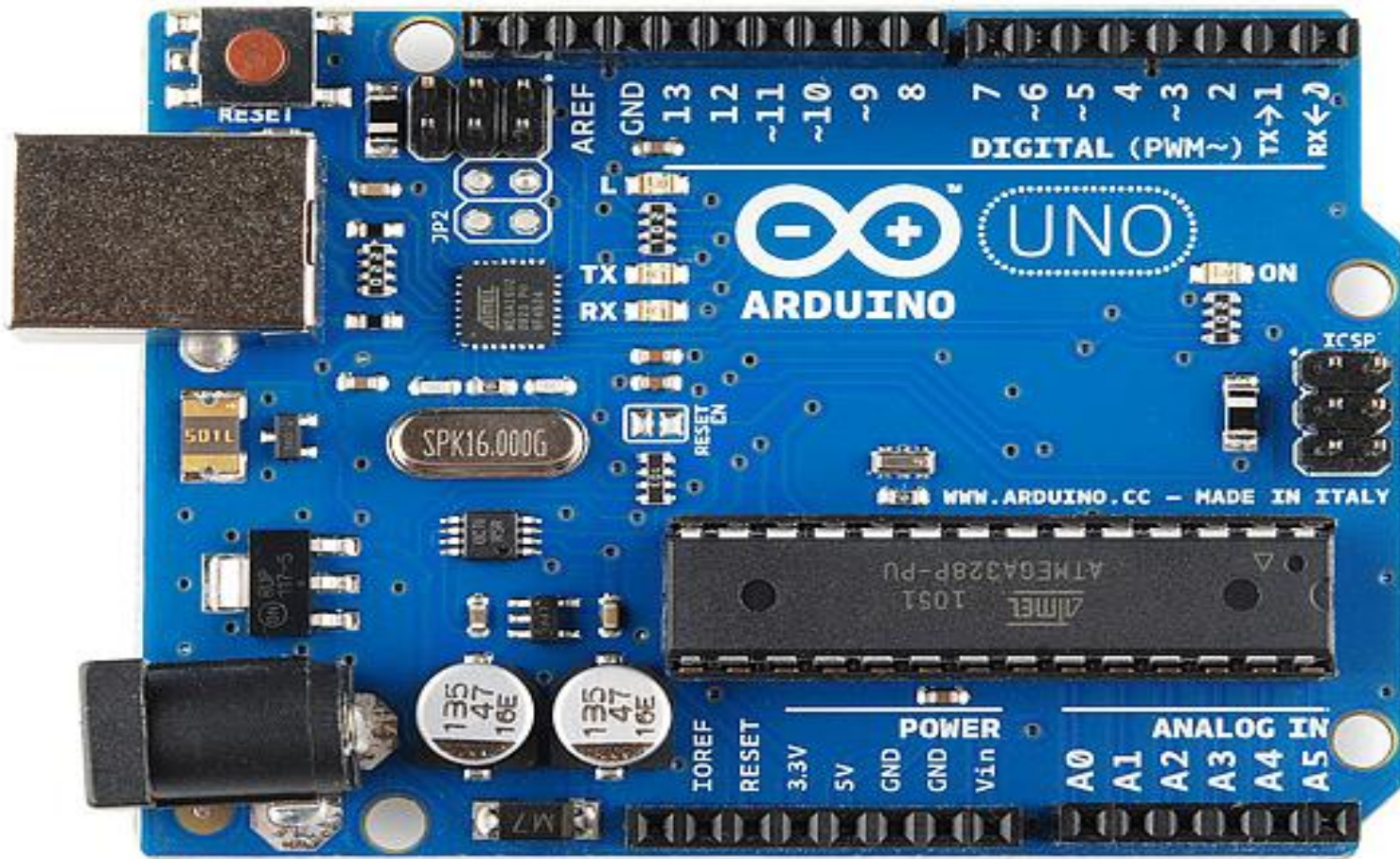
Name	Processor	CPU Speed	Analog In/Out	Flash [kB]	UART
101	Intel® Curie	32MHz	6/0	196	-
Gemma	ATtiny85	8 MHz	1/0	8	0
LilyPad	ATmega168V ATmega328P	8MHz	6/0	16	-
LilyPad SimpleSnap	ATmega328P	8 MHz	4/0	32	-
LilyPad USB	ATmega32U4	8 MHz	4/0	32	-
Mega 2560	ATmega2560	16 MHz	16/0	256	4
Micro	ATmega32U4	16 MHz	12/0	32	1
MKR1000	SAMD21 Cortex-M0+	48MHz	07-Jan	256	1
Pro	ATmega168 ATmega328P	8 MHz 16 MHz	6/0	16 32	1
Pro Mini	ATmega328P	8 MHz 16 MHz	6/0	32	1
Uno	ATmega328P	16 MHz	6/0	32	1
Zero	ATSAMD21G18	48 MHz	06-Jan	256	2

Arduino Board Types (contd.)

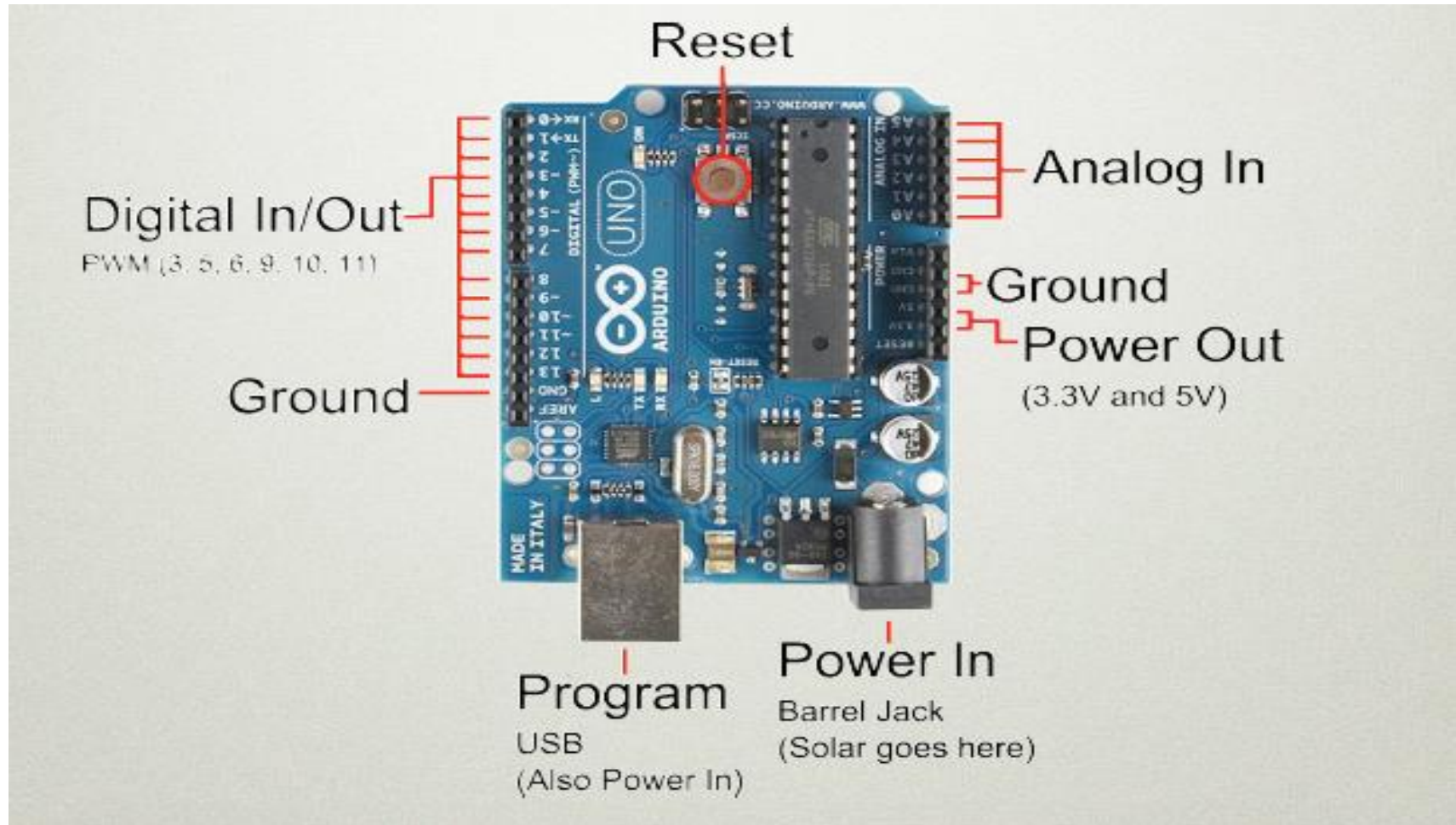


Name	Processor	CPU Speed	Analog In/Out	Flash [kB]	UART
Due	ATSAM3X8E	84 MHz	12-Feb	512	4
Esplora	ATmega32U4	16 MHz	-	32	-
Ethernet	ATmega328P	16 MHz	6/0	32	-
Leonardo	ATmega32U4	16 MHz	12/0	32	1
Mega ADK	ATmega2560	16 MHz	16/0	256	4
Mini	ATmega328P	16 MHz	8/0	32	-
Nano	ATmega168	16 MHz	8/0	16	1
	ATmega328P			32	
Yún	ATmega32U4	16 MHz	12/0	32	1
	AR9331 Linux	400MHz		64MB	
Arduino Robot	ATmega32u4	16 MHz	6/0	32 KB (ATmega32u4) of which 4 KB used by bootloader	1
MKRZero	SAMD21 Cortex-M0+ 32bit low power ARM MCU	48 MHz	7 (ADC 8/10/12 bit)/1 (DAC 10 bit)	256 KB	1

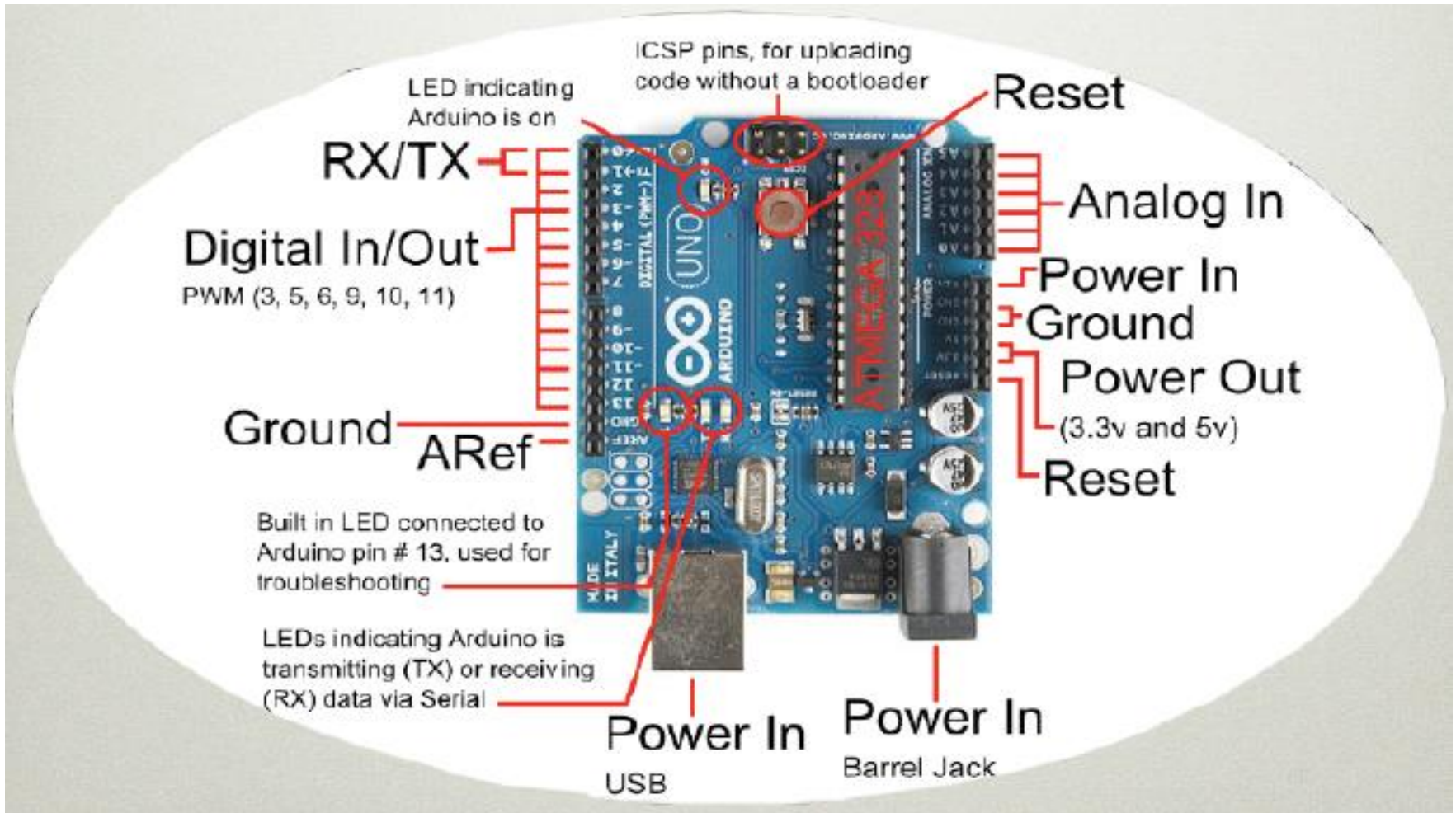
Arduino UNO



Arduino UNO

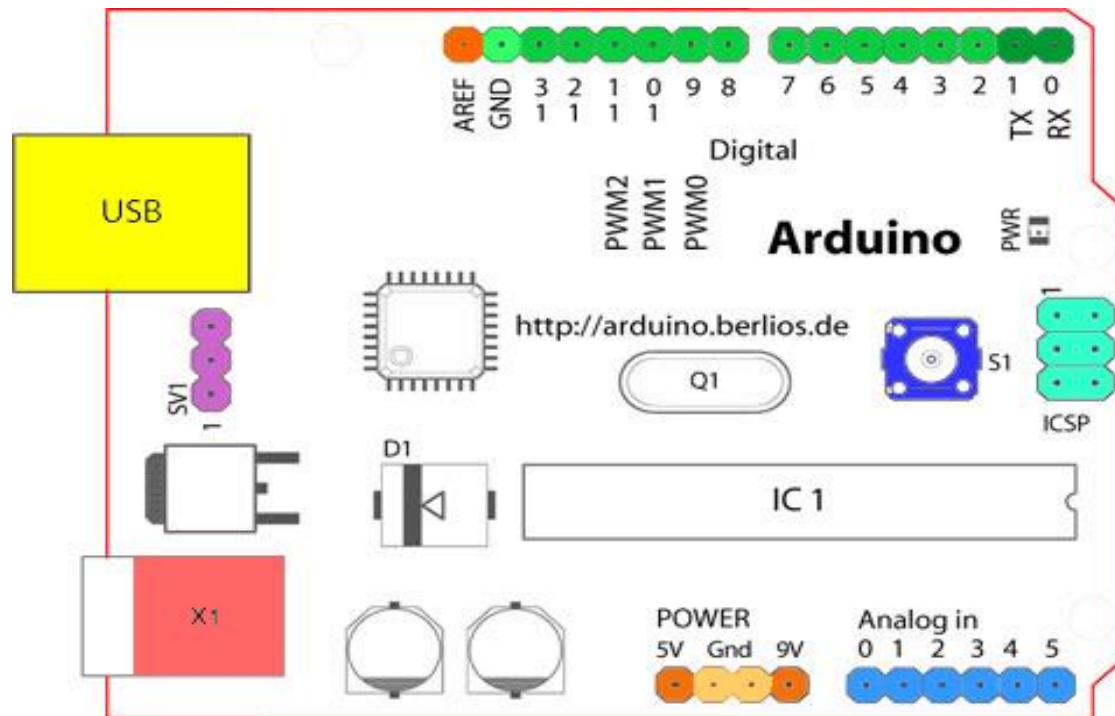


Arduino UNO



Arduino UNO

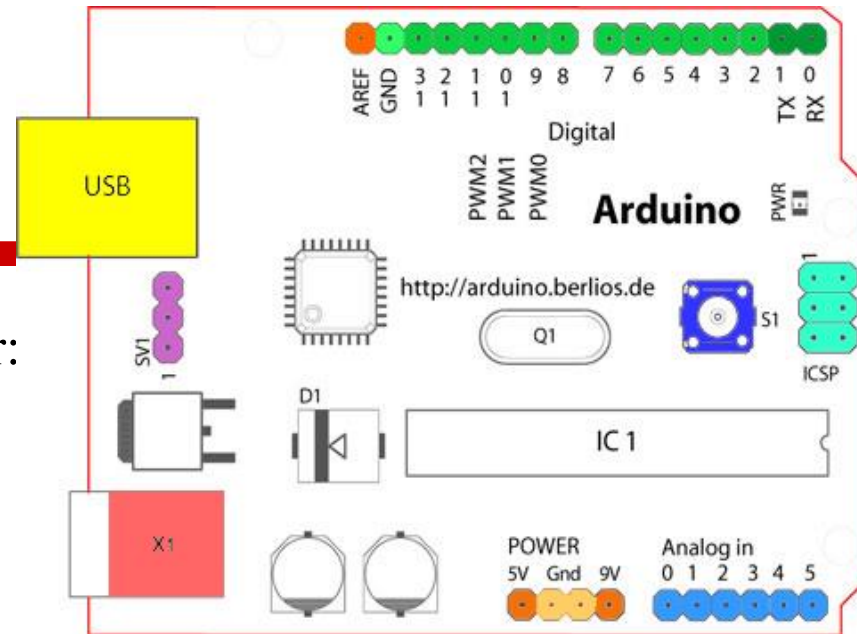
- Looking at the board from the top down, this is an outline of what you will see (parts of the board you might interact with in the course of normal use are highlighted):



Arduino UNO

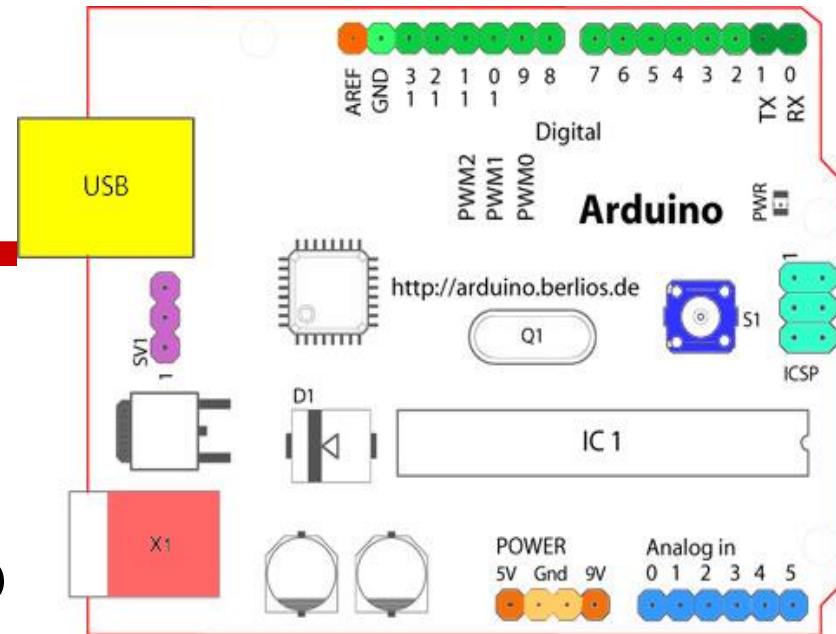
□ Starting clockwise from the top center:

- Analog Reference pin (**orange**)
- Digital Ground (**light green**)
- Digital Pins 2-13 (**green**)
- Digital Pins 0-1/Serial In/Out - TX/RX (**dark green**)
 - ❖ *These pins cannot be used for digital I/O (**digitalRead** and **digitalWrite**) if you are also using serial communication (e.g. **Serial.begin**).*
- Reset Button - S1 (**dark blue**)
- In-circuit Serial Programmer or ICSP (**blue-green**)



Arduino UNO

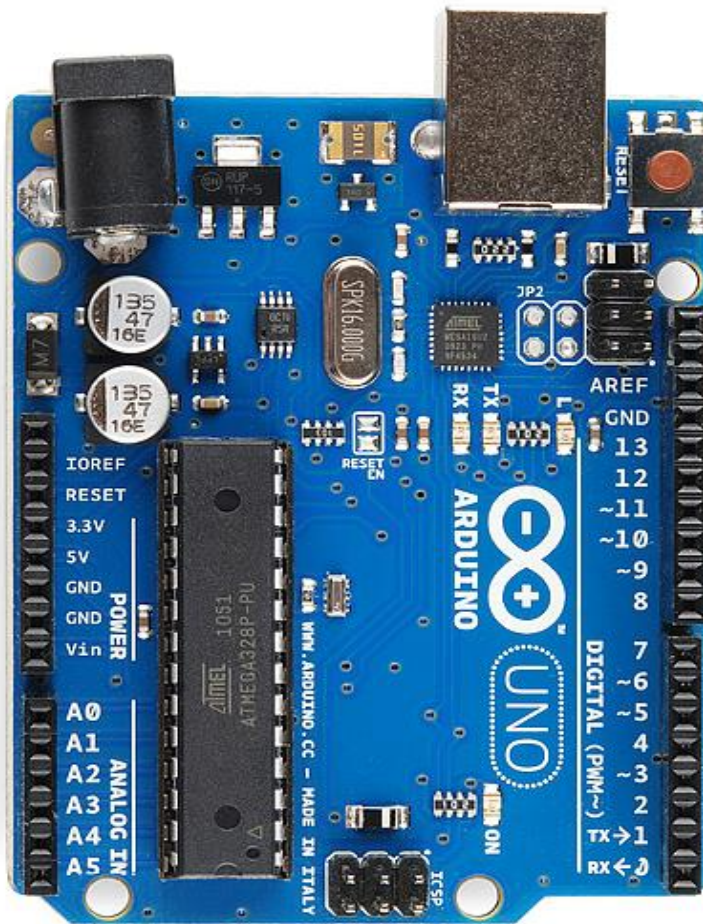
- Analog In Pins 0-5 (light blue)
- Power and Ground Pins
(power: orange, grounds: light orange)



- External Power Supply In (9-12VDC) - X1 (pink)
- Toggles External Power and USB Power (place jumper on two pins closest to desired supply) - SV1 (purple)
- USB (used for uploading sketches to the board and for serial communication between the board and the computer; can be used to power the board) (yellow)



Arduino UNO Specifications



Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recom)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (6 PWM)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (0.5 KB boot loader)
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g



Arduino Terminology

- **“sketch”** – a program you write to run on an Arduino board
- **“pin”** – an input or output connected to something.
e.g. output to an LED, input from a knob.
- **“digital”** – value is either HIGH or LOW. (on/off, one/zero)
e.g. switch state
- **“analog”** – value ranges, usually from 0-255.
e.g. LED brightness, motor speed, etc.

Memory



- There are three pools of memory in the microcontroller used on AVR(ATmega 328)-based Arduino boards :
 - **Flash memory**: Flash memory known as **program space**, is where the Arduino sketch is stored.
 - **SRAM (static random access memory)**: SRAM is where the sketch **creates and manipulates variables** when it runs.
 - **EEPROM**: EEPROM is memory space that programmers can use to **store long-term information**.

```
Flash 32k bytes (of which .5k is used for the bootloader)
SRAM 2k bytes
EEPROM 1k byte
```

Memory



- Flash memory and EEPROM memory are **non-volatile** (the information persists after the power is turned off). SRAM is **volatile** and will be lost when the power is cycled.
- Notice that there's not much SRAM available in the Uno. It's easy to use it all up by having lots of strings in your program. For example, a declaration like:
- `char message[] = "I support the Cape Wind project.";`
- puts 33 bytes into SRAM (each character takes a byte, plus the '\0' terminator). This might not seem like a lot, but it doesn't take long to get to 2048, especially if you have a large amount of text to send to a display, or a large lookup table, for example.