



Introduction to Arduino

Microprocessor

- A **microprocessor** is a computer processor which incorporates the functions of a **computer's central processing unit** (CPU) on a single **integrated circuit** (IC), or at most a few integrated circuits.
- The microprocessor is a multipurpose, clock driven, **register** based, **programmable** electronic device which accepts **digital** or **binary** data as input, processes it according to instructions stored in its memory, and provides results as output.
- Examples- Intel 8080, ARM2, NUC

Microcontrollers

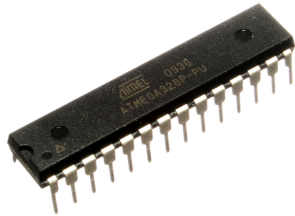
- A **microcontroller** (or **MCU**, short for *microcontroller unit*) is a small computer (SoC) on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals.
- A typical microcontroller program consists of two parts a bootloader and the main application. A simple bootloader will do some basic initialization and then call the main application.

Processors vs MicroProcessors



MicroProcessors vs MicroControllers

- Looks don't matter, peripherals do!

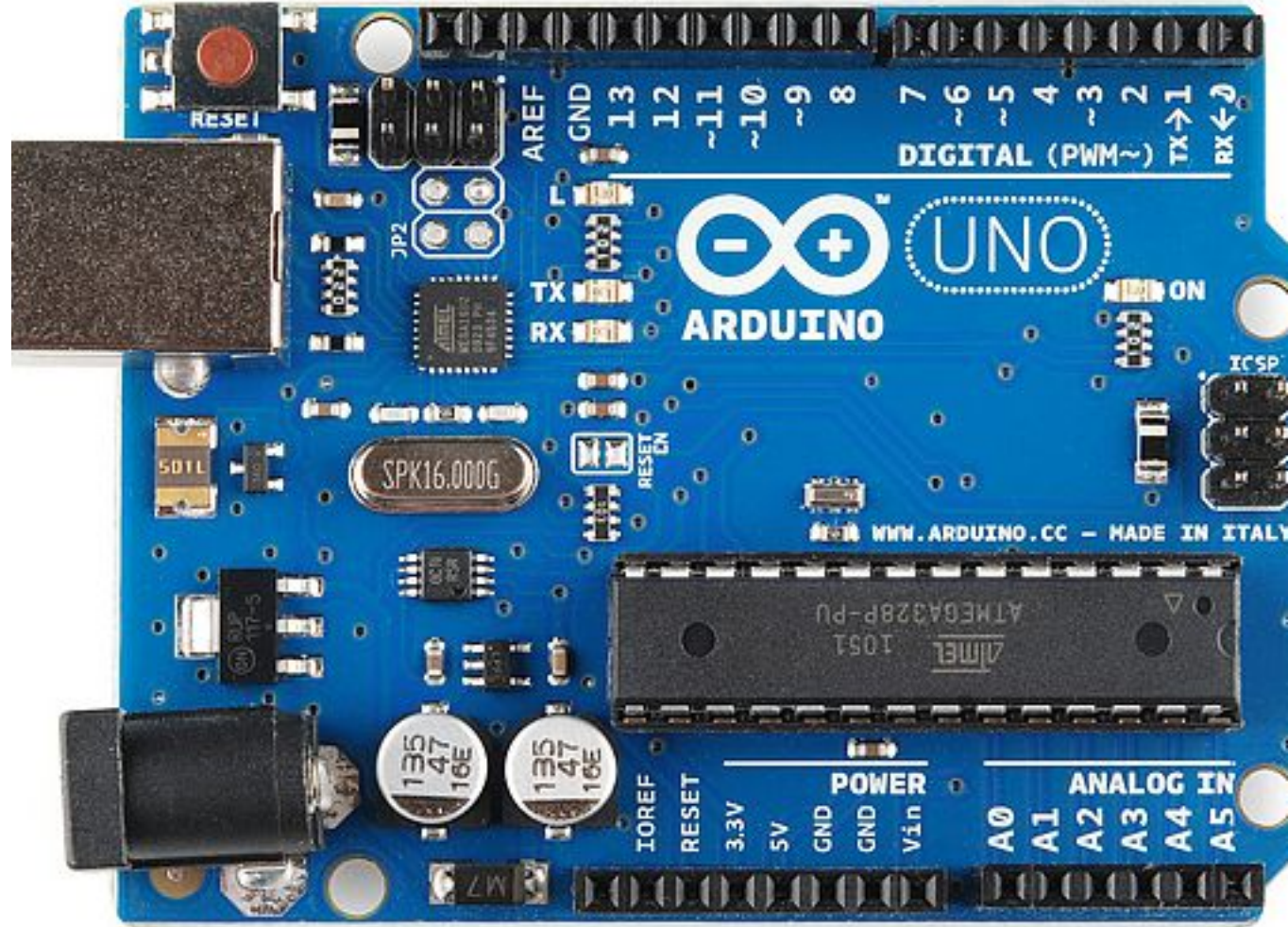


Peripherals of a Microcontroller

- Digital I/O : Provides the ability to control and read digital outputs.
- A/D Conversion :The ability to measure an analog voltage.
- D/A Conversion :The ability to output analog voltages.
- Serial Interface The serial interface is used to transfer data between devices within a system and between systems.
- Timers :Provides the ability to measure durations between events.
- Flash Memory Used to store program code and data.

Arduino

Arduino is an open-source electronics platform based on ATmega microcontroller. It comes in different models to cater to different needs. The commonly used ones are Arduino Uno and Arduino Mega. Arduino IDE is used for programming an Arduino.



Why Arduino?

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

Standard Functions

- `PinMode(pin_number, INPUT/OUTPUT)` : This function declares a given pin to be input or output pin.
- `digitalWrite(pin_number,0/1/HIGH/LOW)`:This function produces a digital signal of either HIGH(5V) or LOW(0V) on the given pin.
- `analogWrite(pin_number, value)`:This function produces an analog signal varying between 0 to 5V on the given pin.
- `analogRead(pin_number)`:This function reads the voltage on the given pin and outputs its value as an integer ranging from 0 to 1023.
- `digitalRead(pin_number)`:This function reads the voltage on the given pin and outputs its value as either 0 or 1.

Standard Functions

- `Serial.begin(baud_rate)`: This function begins serial communication between the microcontroller and the arduino.
- `Serial.print()`: This function prints the value in its argument on the serial monitor of the arduino IDE.
- `delay(time)`: This function provides delay of given time (in milliseconds) during execution of the code.
- `void setup()` : The code inside this function is run only once, when new code has been uploaded.
- `void loop()` : The code inside this runs on a loop. Without any delay function, the default rate of delay is 1ms.

Application 1

- Now, we will try to use an arduino for a simple task like blinking an LED.
- For this purpose, we will use will use Autodesk's online circuit simulator.
- Visit <https://circuits.io/lab>
- Sign Up and then open electronics lab hub.
- For this workshop, most of our circuits will be constructed in this simulator.
- The blink test is used for checking whether the Arduino is working or not.

Application 2

- We will now try to obtain an input from a potentiometer use it to blink an LED.
- The problem statement: When the voltage measured at a given analog pin is less than 2V, blink an LED at the frequency of 10 blinks per second.
- Once this is done, we try to change the frequency of blinks as the voltage on analog pin is changed.

