

# Development Boards

Module Code: GEEN1064

Module Name: Engineering Design and Implementation

Lecturer: Seb Blair BEng(H) PGCAP MIET MIHEEM FHEA

# What is (and isn't) a Microcontroller ( $\mu$ C)

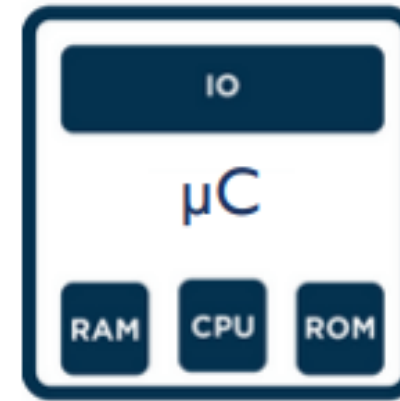
- A  $\mu$ C is a small, self-contained computer that is housed on a single integrated circuit, or microchip.
- They differ from your desktop computer in that they are typically dedicated to a single function, and are most often embedded in other devices (e.g. mobile phones; household electronics).
- A  $\mu$ C is an Integrated Circuit (IC) device used for controlling other portions of an electronic system
- These devices are optimized for embedded applications that require both processing functionality and agile, responsive interaction with digital, analogue, or electromechanical components.

## What is (and isn't) a Microcontroller ( $\mu$ C)

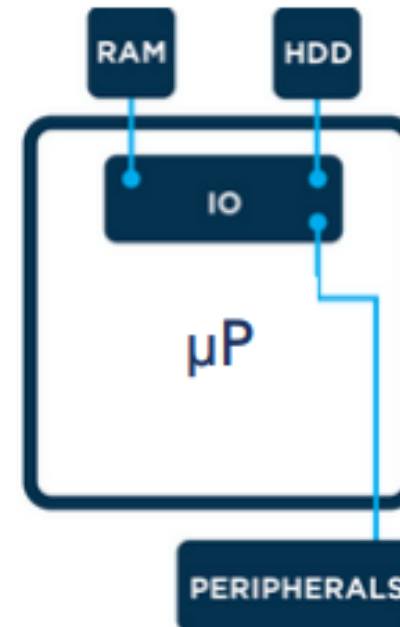
- $\mu$ C” is a well-chosen name because it emphasizes defining characteristics of this product category. The prefix “ $\mu$ ” implies smallness and the term "controller" here implies an enhanced ability to perform control functions.
- As stated above, this functionality is the result of combining a digital processor and digital memory with additional hardware that is specifically designed to help the  $\mu$ C interact with other components.

# Microcontroller vs a Microprocessor

- $\mu P$  contain only a Central Processing Unit (CPU), and therefore require added peripherals to perform tasks.
- CPU is the brain of any computer and it is the part of a computer that performs calculations, actions, and runs programs.
  - Intel Core i series, AMD



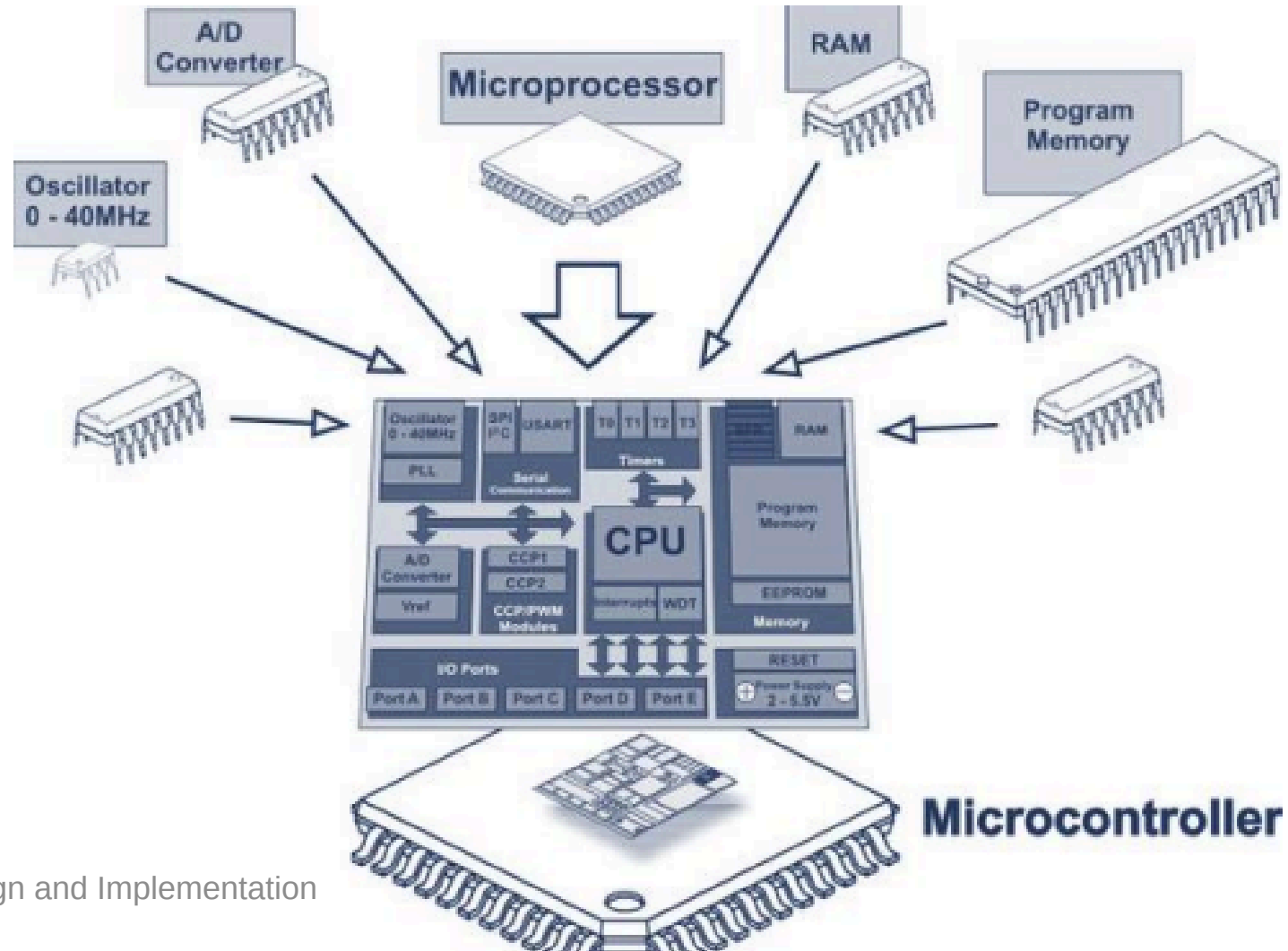
$\mu C$  vs  $\mu P$



# RAM and ROM

- $\mu$ Cs, on the other hand, contain RAM, ROM, and similar peripherals, which allow them to perform (simple) tasks independently.
- Random-Access Memory (RAM)
  - everyday tasks, such as loading applications, browsing the internet, editing a spreadsheet, or experiencing the latest game
- Read-Only Memory (ROM)
  - computer memory chips containing permanent or semi-permanent data.  
Unlike RAM, ROM is non-volatile; even after you turn off your computer, the contents of ROM will remain

# Exploded View



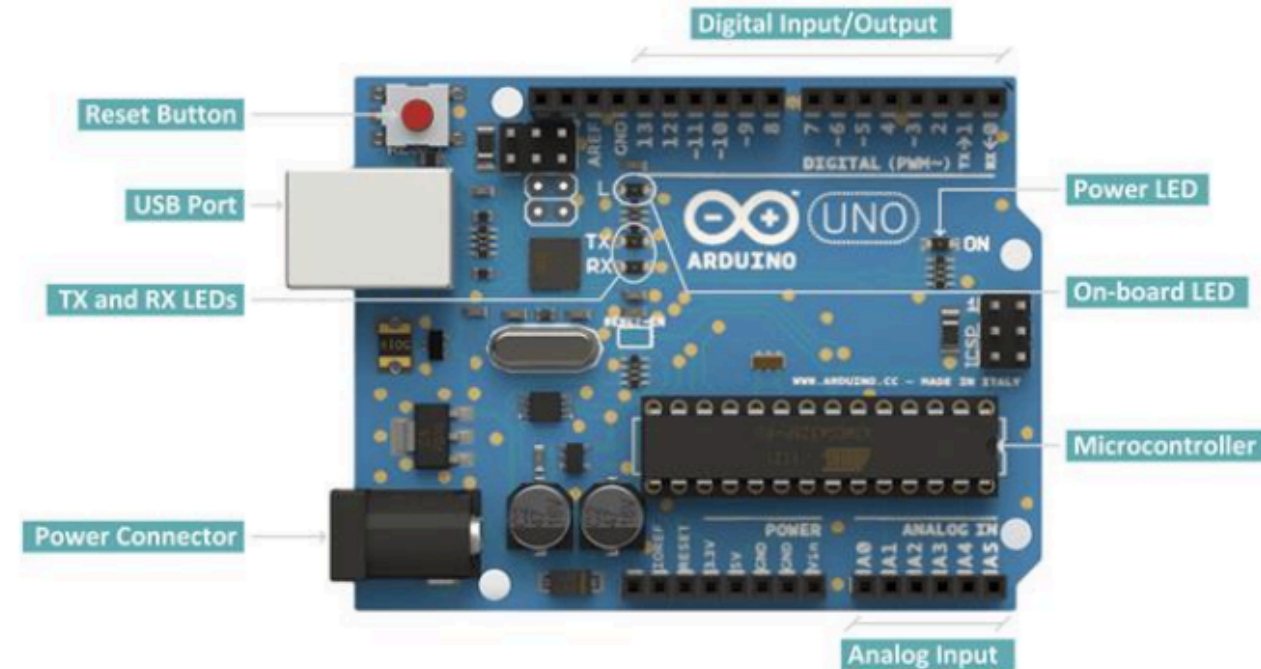
# Deployment Platforms

- Prototyping platforms
  - Arduino family -> Uno, Mega, MKR series...
  - Single board computers -> Raspberry Pi, Orange Pi, Odroid, Jetson Nano....
- Used to build physical computing projects
  - Taking information from the world around us by using inputs such as sensors and switches and responding to that information with outputs like actuators, graphical indicators, lights etc.
- Open source hardware and software
- Open source projects emphasise the importance of documentation and sharing, making the community of users a great resource for learners.

# Arduino Uno

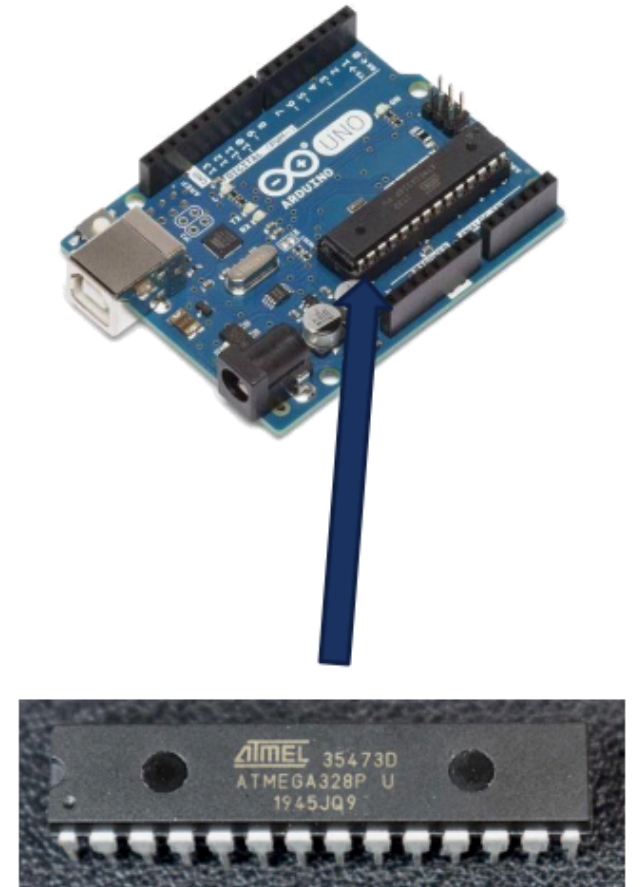
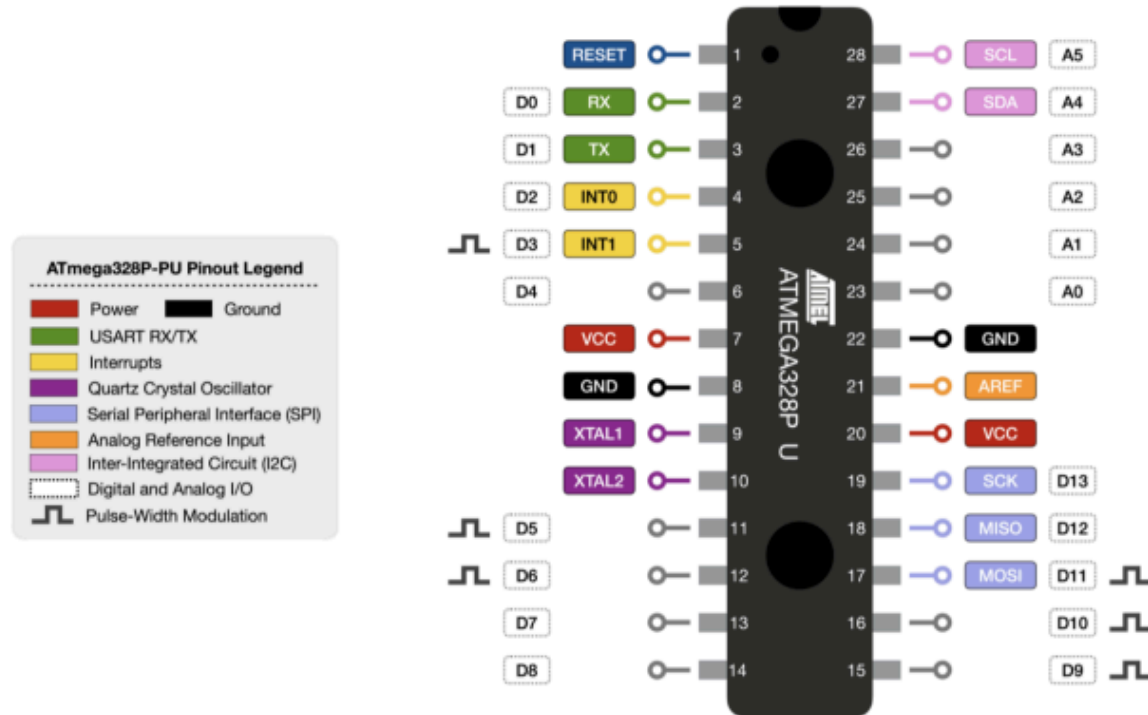
## Prototype Board from Arduino

- The Uno is a  $\mu$ C development board with an Atmega328P Chip
- 14 digital Input/Output (I/O) pins
- 6 Analogue Input pins (e.g. temperature sensors)
- 6 PWM Output pins (e.g. actuators)
- A 16 MHz quartz crystal

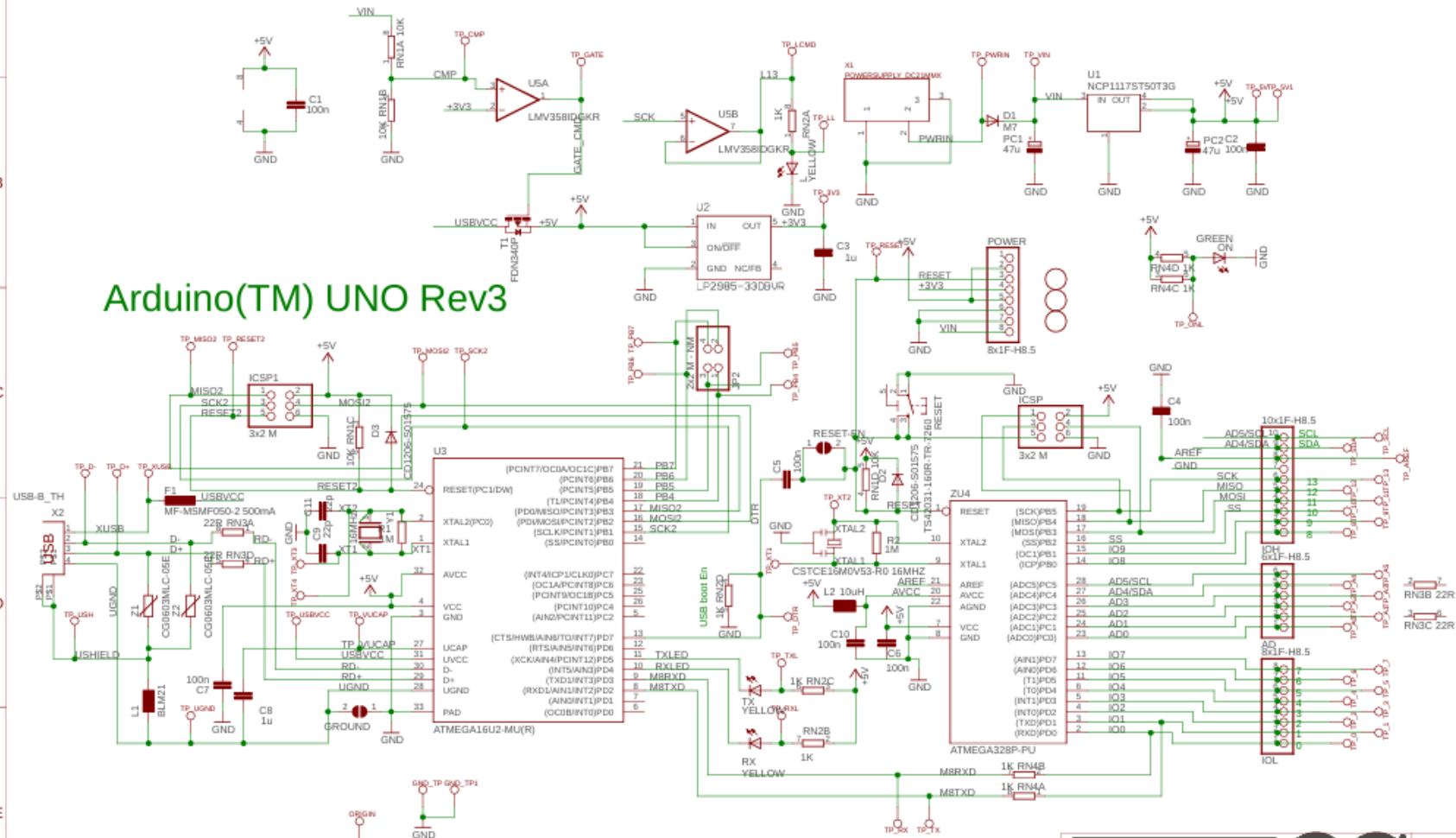




# ATmega328P Chip



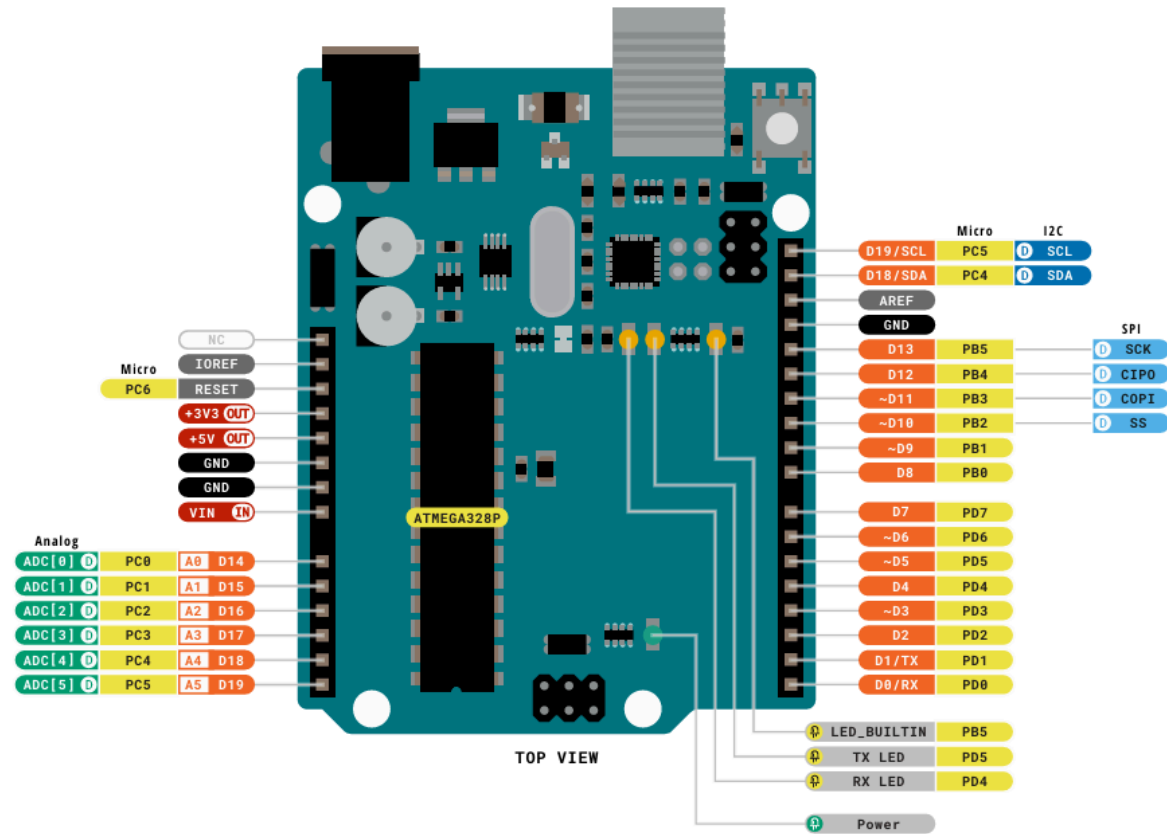
# Arduino(TM) UNO Rev3



Reference Design is PROVIDED "AS IS" AND "WITH ALL FAULTS." ARDUINO SA DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. ARDUINO SA may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "under development" or "untested." ARDUINO SA reserves the right to change specifications or instructions without notice. The product information on the Web Site of ARDUINO is subject to change without notice. Do not finalize a design with this info. ARDUINO and other Arduino brands and logos are Trademarks of Arduino SA. All Arduino SA Trademarks cannot be used without owner's formal permission.



UNO-TH\_Rev3e  
06/03/19 12:33  
Sheet: 1/1



#### Legend:

<span style="color: red;">■</span> Power	<span style="color: red;">IN</span> Power Input	<span style="color: orange;">■</span> GPIO Digital External	<span style="color: blue;">■</span> I2C	<span style="color: blue;">Ⓢ</span> Default	<span style="color: red;">Ⓢ</span> LED
<span style="color: grey;">■</span> Ground	<span style="color: red;">OUT</span> Power Output	<span style="color: lightorange;">■</span> Analog External	<span style="color: lightblue;">■</span> SPI	<span style="color: lightblue;">Ⓢ</span> Default	<span style="color: blue;">Ⓢ</span> RGB LED
		<span style="color: yellow;">■</span> Main Part	<span style="color: purple;">■</span> UART/USART	<span style="color: purple;">Ⓢ</span> Default	<span style="border: 1px solid black;">□</span> Other
		<span style="color: brown;">■</span> Secondary Part	<span style="color: lightblue;">■</span> Other SERIAL Communication	<span style="color: green;">Ⓢ</span> Default	
		<span style="color: grey;">■</span> Internal Component	<span style="color: green;">■</span> Analog		
		<span style="color: black;">■</span> Other Pins (Reset, System Control, Debugging)	<span style="color: lightgreen;">■</span> PWM/Timer		

- ⚠ **MAXIMUM** current per I/O pin is 20mA
- ⚠ **MAXIMUM** current per +3.3V pin is 50mA
- ! VIN 6-20V input to the board
- i CIP0/COP1 have previously been referred to as MISO/MOSI



SKU code: A000066  
Full Pinout - Page 1 of 5  
Last update: 6 Oct, 2022

DOCS.ARDUI.NO.CC



This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

# Base Code

```
int led = 12; // set a variable, in this case pin number 12

void setup(){
  // Should only run once
  pinMode(led, OUTPUT); // set pin to send signal out
  Serial.begin(9600); // set the speed at which data is transmitted to the serial monitor
}

void loop(){
  // This runs forever
  digitalWrite(led, HIGH); // Turn pin 12 on
  Serial.println("On");
  delay(500); // on for 0.5 secs
  digitalWrite(led, LOW); // turn pin 12 off
  Serial.println("Off");
  delay(500); // off for 0.5 secs
}
```

# Basic Functions

```
pinMode(int, <OUTPUT/INPUT>); // pin, direction
digitalWrite(int, <HIGH/LOW>); // pin, state

analogRead(int); // pin
analogWrite(int, int); // pin, value (0-255)

Serial.begin(int); // speed (4800, 9600, 19200 ...)
Serial.print(Type); // will write to serial any data type, without a new line terminator "\n"
Serial.println(Type); // will write to serial any data type, with a new line terminator "\n"

delay(int); // a specified amount of time in ms the cpu will wait before continuing to the next line
```