



UNIVERSITÀ
DEGLI STUDI
DI BERGAMO

Department
of Engineering
and Applied Sciences

Microcontrollori in Elettronica Industriale

Elettronica Industriale

Dipartimento di Ingegneria e Scienze Applicate

Dhiego Fernandes Carvalho

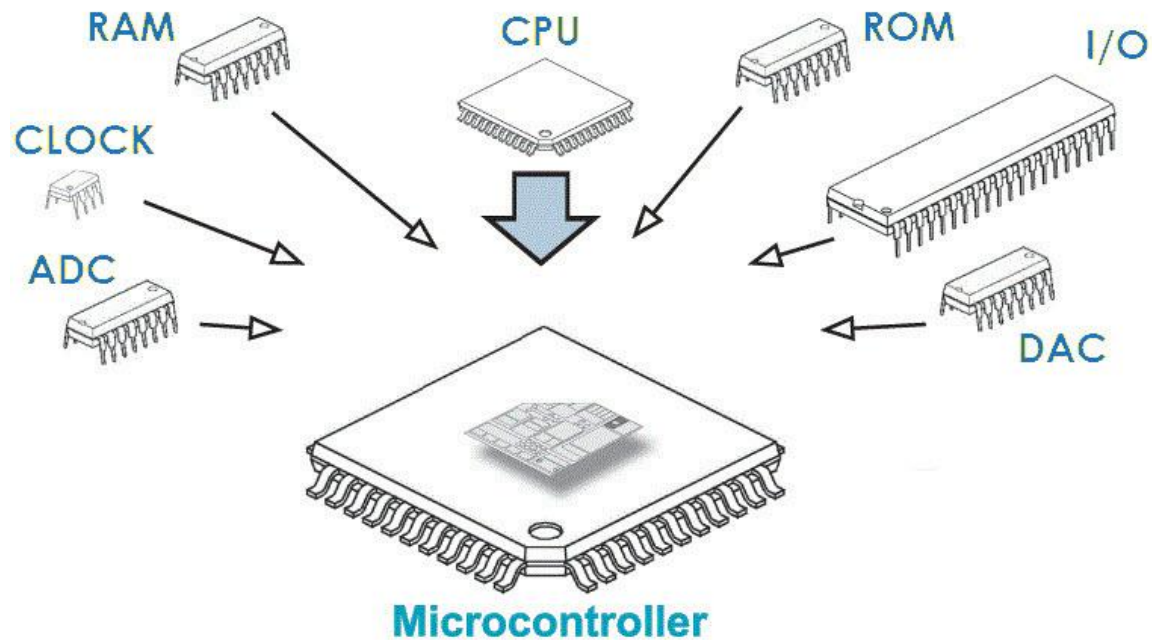
dhiego.fernandes@unibg.it

Obiettivi della lezione

- Al termine della lezione lo studente sarà in grado di:
 - Definire cos'è un microcontrollore.
 - Descrivere la sua architettura di base.
 - Identificare esempi concreti di utilizzo nell'industria.
 - Riconoscere i principali tipi di comunicazione supportati.
 - Conoscere la composizione e il ruolo del kit Arduino Zero to Hero.
 - Comprendere l'ambiente di sviluppo Arduino IDE per programmare i microcontrollori

Che cos'è un microcontrollore

- Un **microcontrollore (MCU)** è un dispositivo elettronico programmabile che integra su un unico chip **CPU**, **memoria**, **periferiche di input/output** ecc. È progettato per il controllo diretto di sistemi elettronici ed elettromeccanici.



Atmel AVR



AVR



ATX Mega



ATmega 328P



PIC 16F877A



8051



Arduino



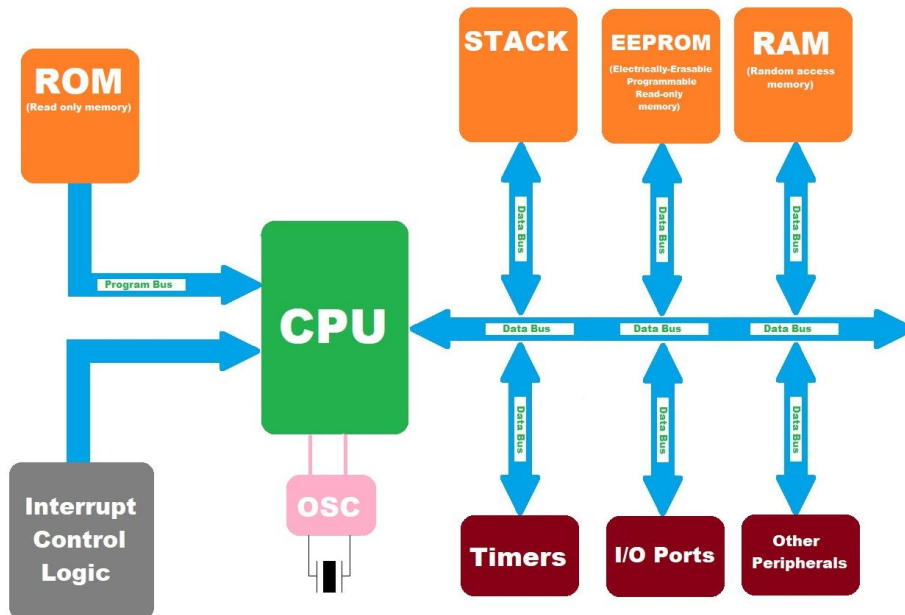
ARM

Architettura di un microcontrollore

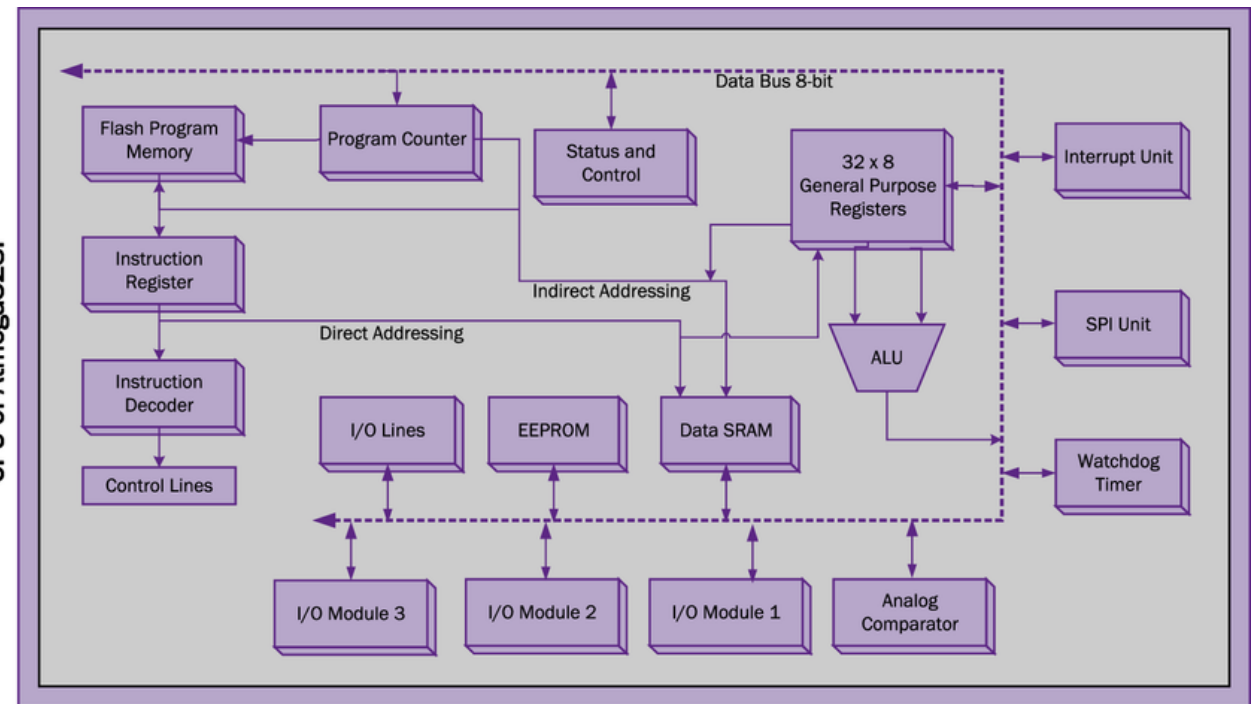
- **Unità di elaborazione:** CPU (ALU, registri, unità di controllo).
- **Memoria:** Flash (programma), SRAM (dati temporanei), EEPROM (dati permanenti).
- **Periferiche:** ADC, DAC, PWM, timer, watchdog.
- **Linee di I/O:** ingressi e uscite digitali e analogiche.
- **Clock e alimentazione:** oscillatore e sistemi di gestione energetica.

Microcontrollers Architecture

(What is a Microcontroller ?)



CPU of Atmega328P



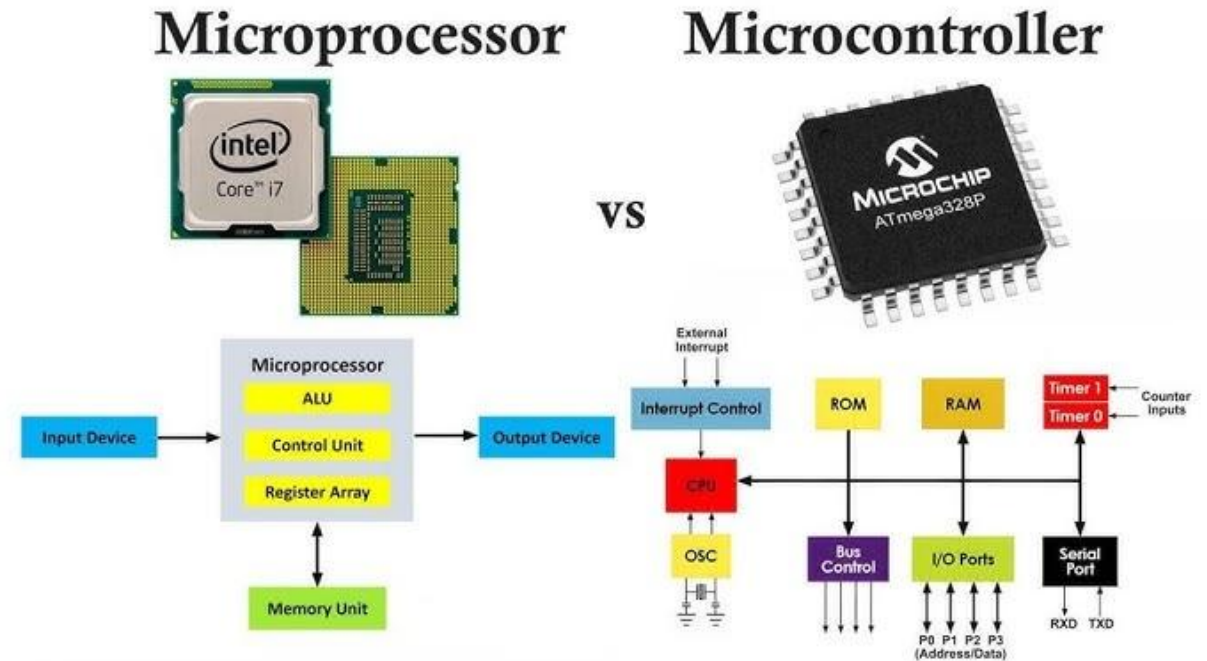
Microcontrollori vs Microprocessori

- **Microcontrollore (MCU)**

- Integra CPU + memoria + periferiche su un solo chip.
- Usato in sistemi embedded e dispositivi di controllo.
- Esempi: ATmega328P, STM32, ESP32.

- **Microprocessore (CPU)**

- Contiene solo la unità di elaborazione, richiede RAM, storage e periferiche esterne.
- Usato in PC, server, smartphone.
- Esempi: Intel Core, AMD Ryzen, ARM Cortex-A.

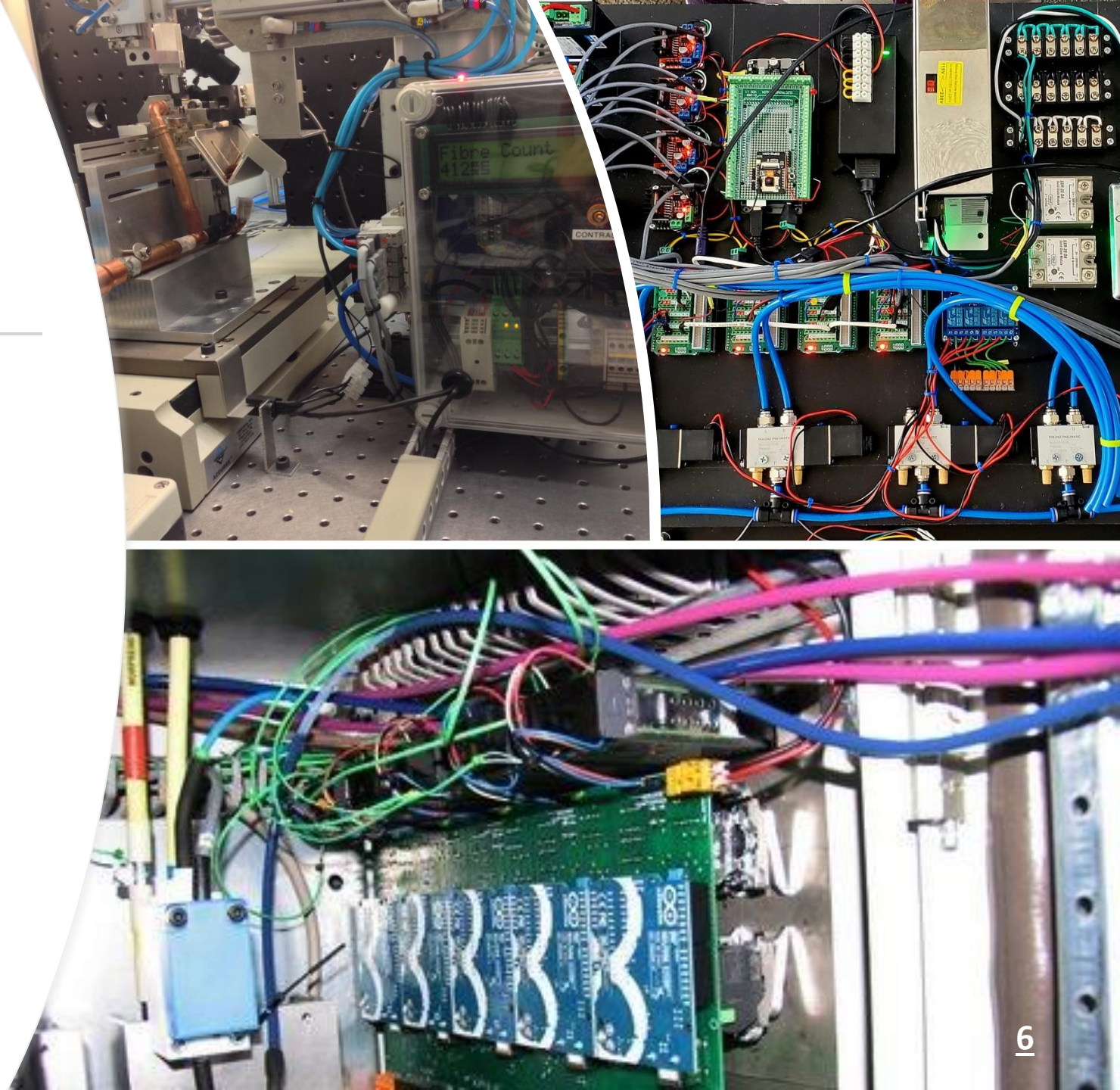


✓ **Differenza chiave:** il microcontrollore è “autonomo” per il controllo, il microprocessore è “potente” ma dipendente da componenti esterni.

Esempi di applicazioni nell'industria

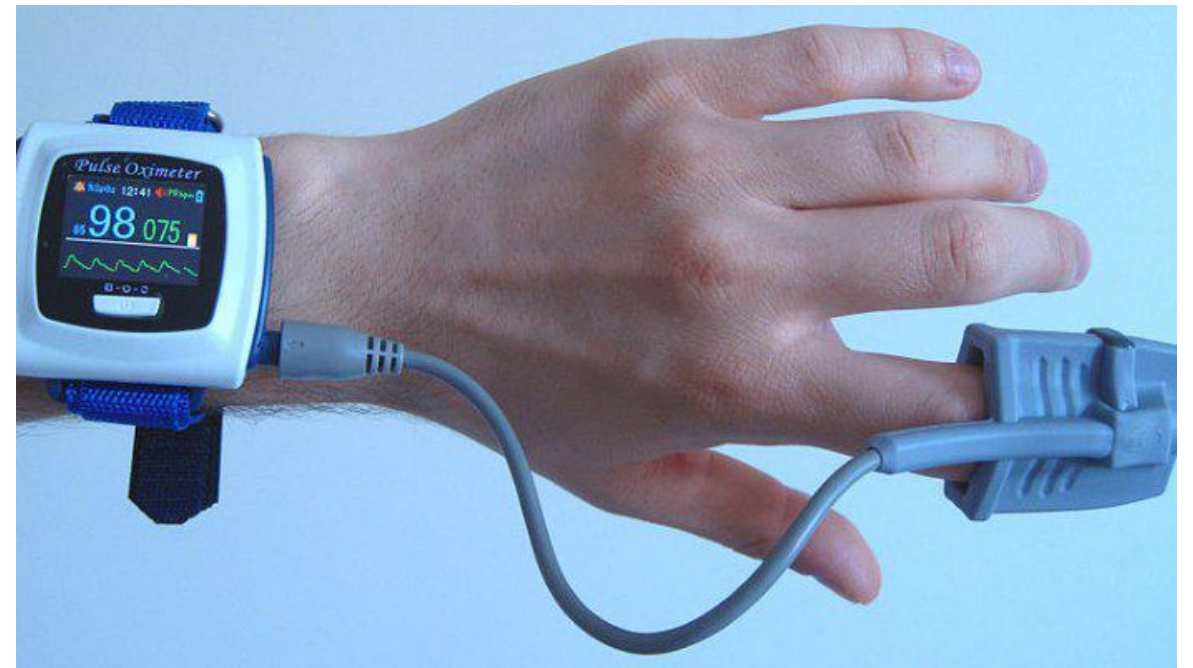
- **Automazione industriale**

- Controllo di motori elettrici, valvole e pompe.
- Regolazione di processi semplici con logiche di controllo.
- Sostituzione di PLC in applicazioni a basso costo (????)



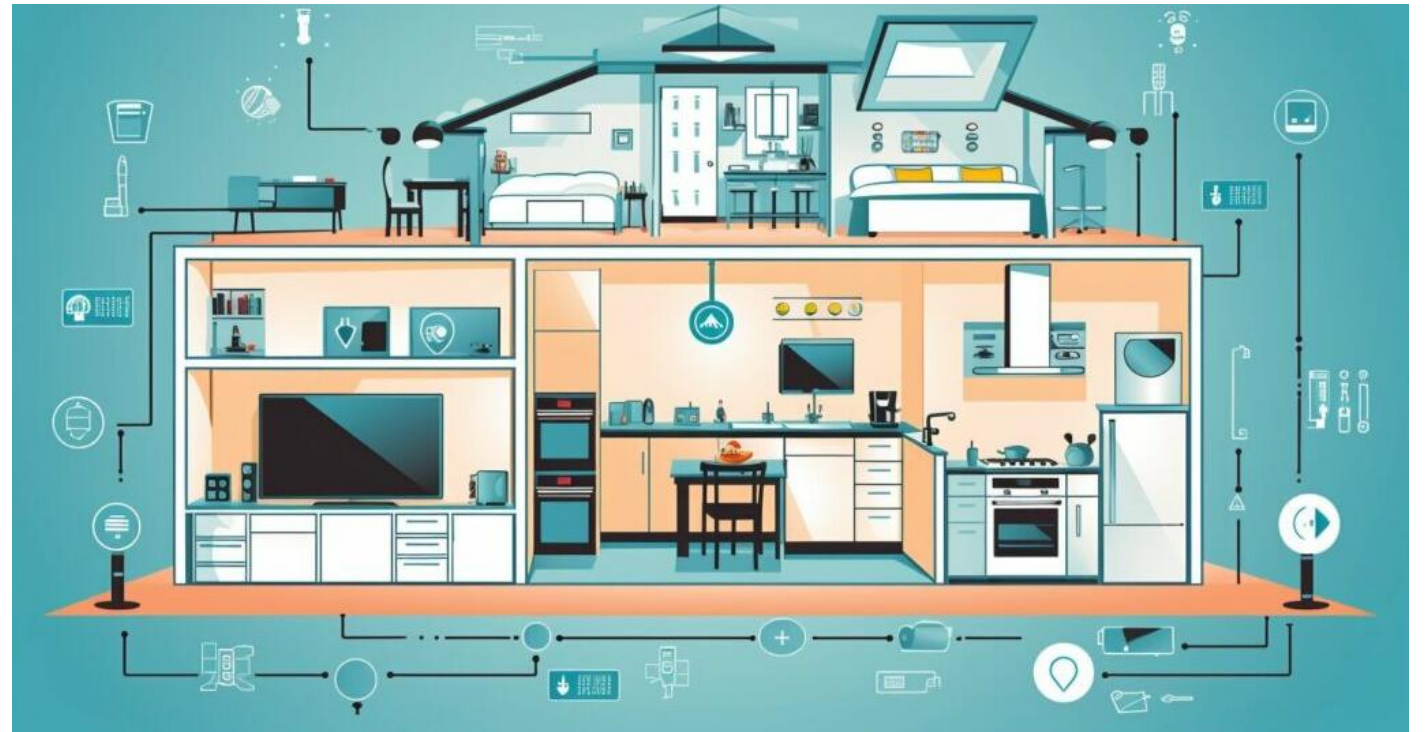
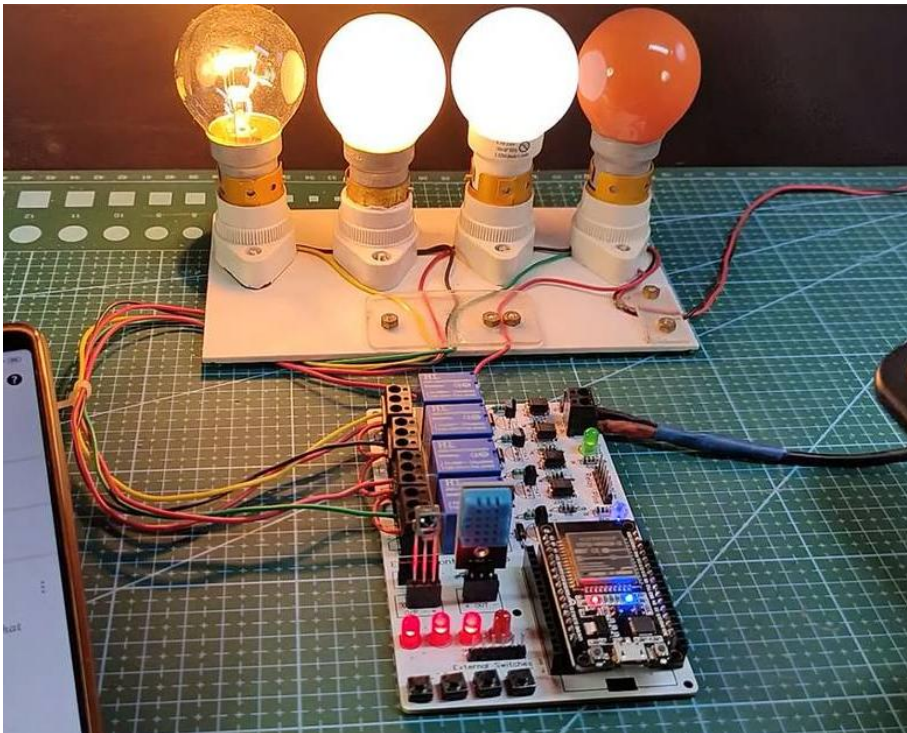
Sistemi Embedded

- Dispositivi embedded con MCU per controllo locale e acquisizione sensori.
- Pre-elaborazione dei segnali.
- Moduli dedicati in varie applicazioni.



IoT – Smart Homes

- Domotica: controllo luci, climatizzazione, sicurezza domestica.
- Sensori ambientali connessi a piattaforme cloud.
- Integrazione con assistenti vocali e sistemi wireless (Wi-Fi, Zigbee, BLE).



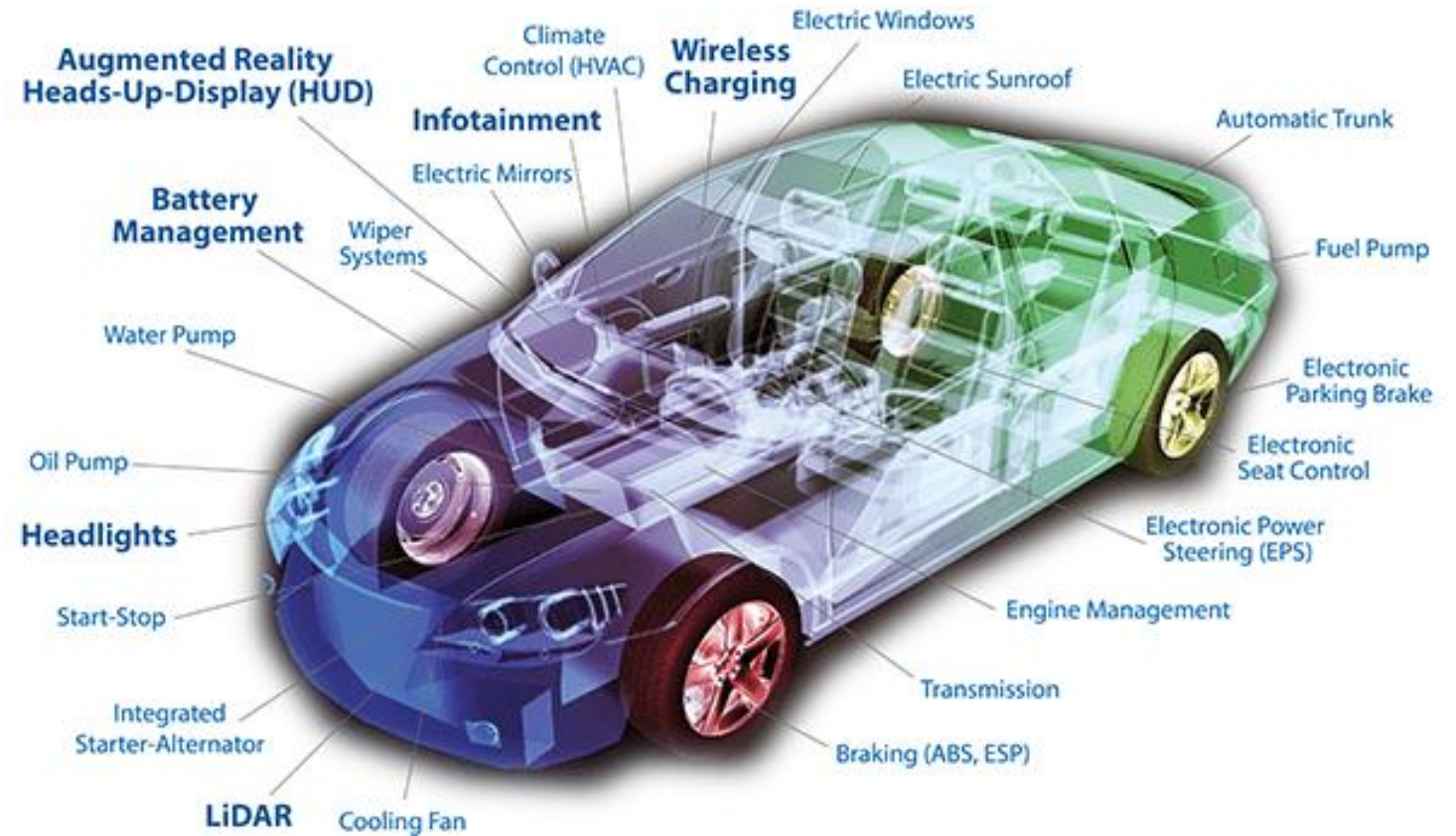
IoT – Smart City

- Monitoraggio ambientale (aria, acqua, rumore).
- Illuminazione pubblica intelligente.
- Sistemi di gestione del traffico e parcheggi smart.
- Reti di sensori distribuiti



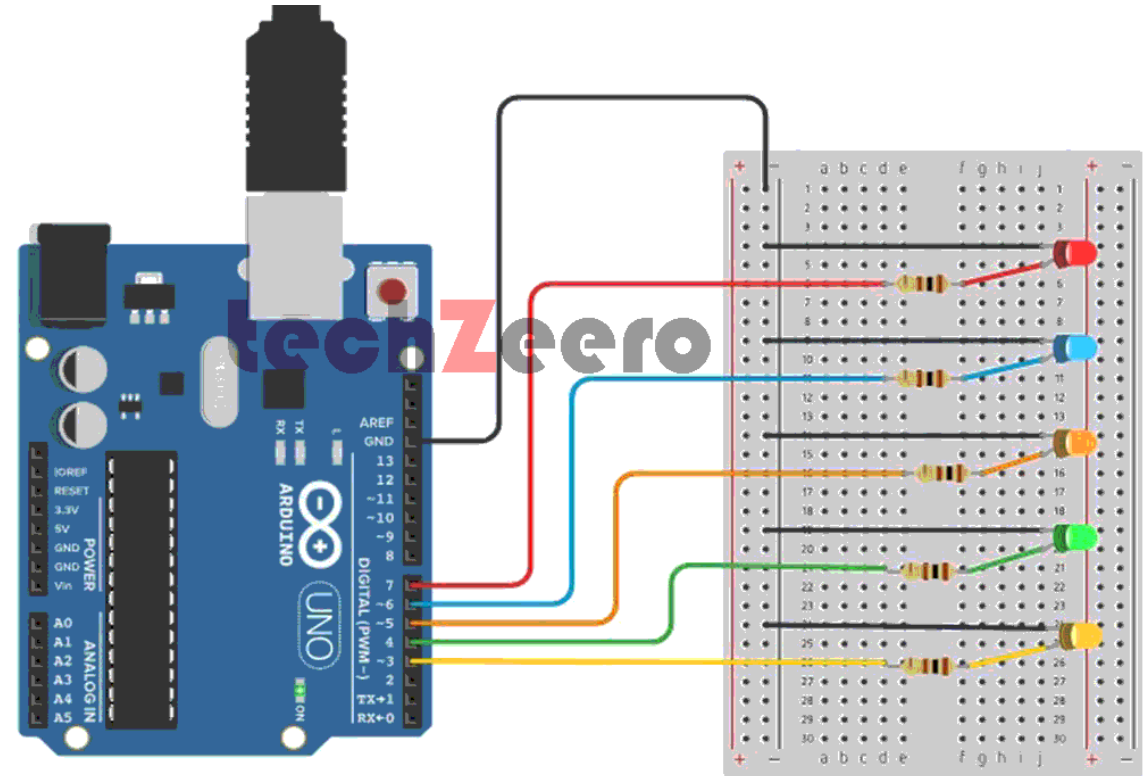
Automotive

- Centraline elettroniche di controllo (ECU).
- Sistemi di sicurezza (ABS, airbag).
- Connettività veicolare.



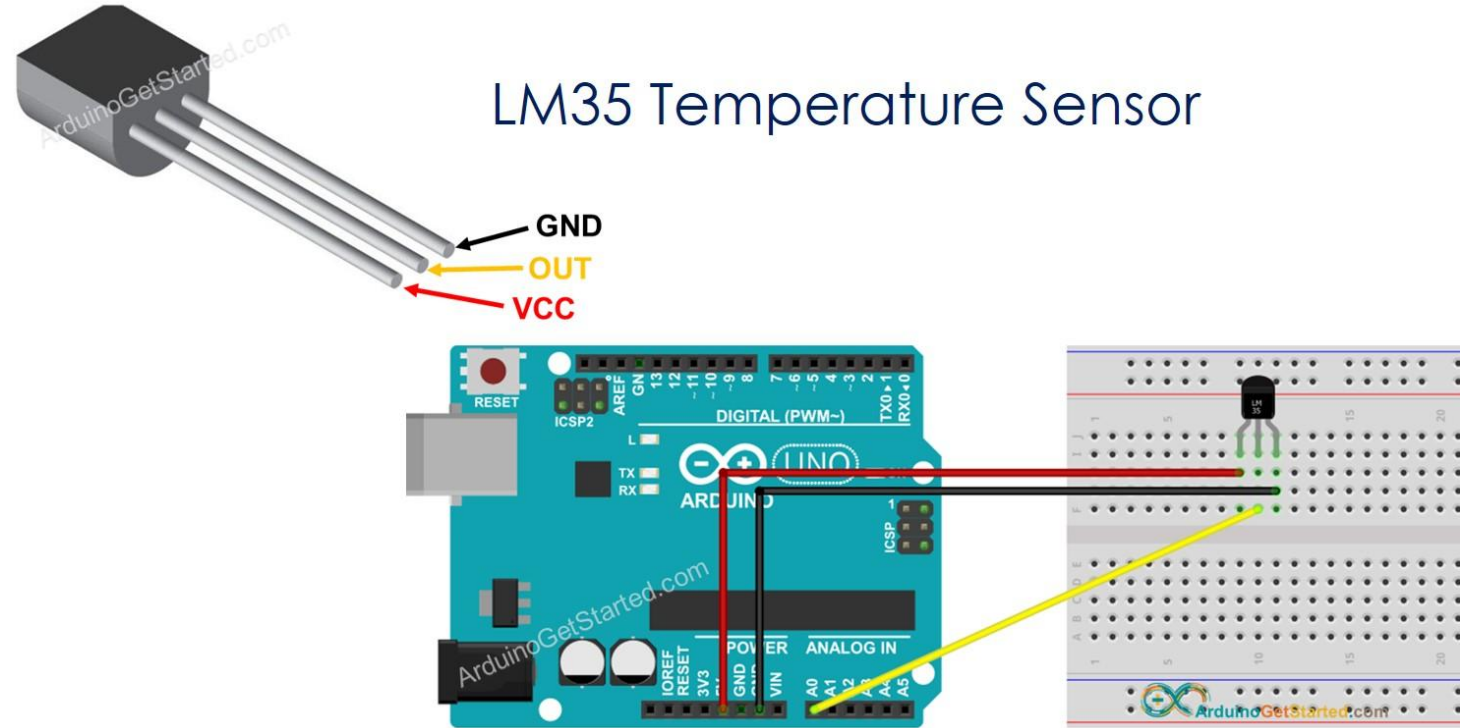
Tipi di comunicazione

- **Digitale (GPIO)**
 - Segnali logici ON/OFF.
 - Controllo diretto di LED, pulsanti, relè.
 - Base per il collegamento di periferiche semplici.



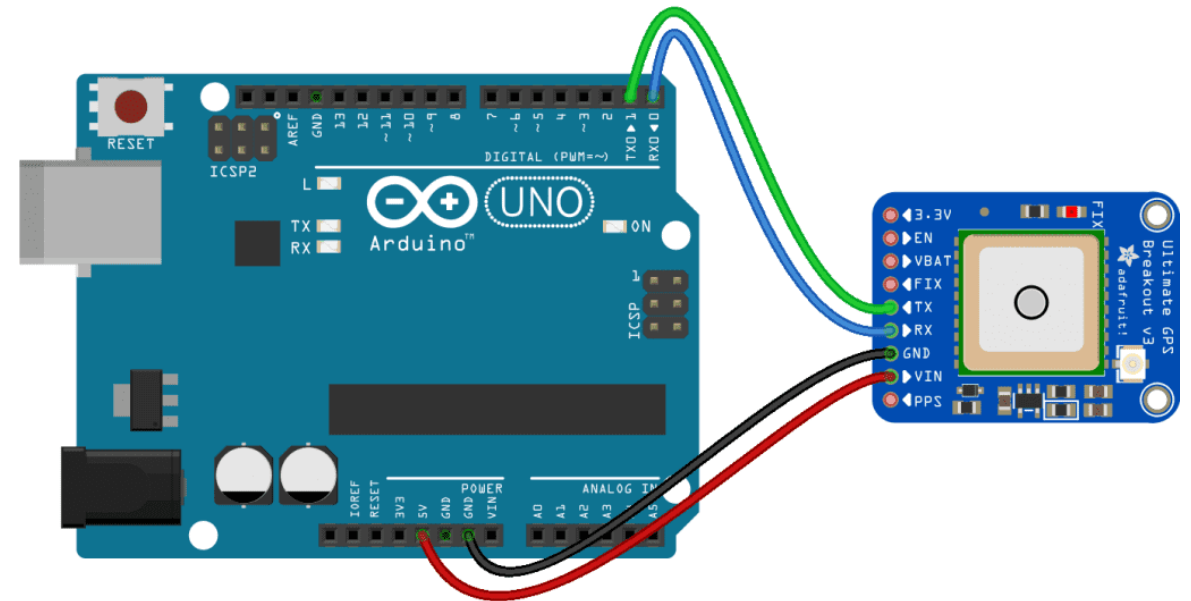
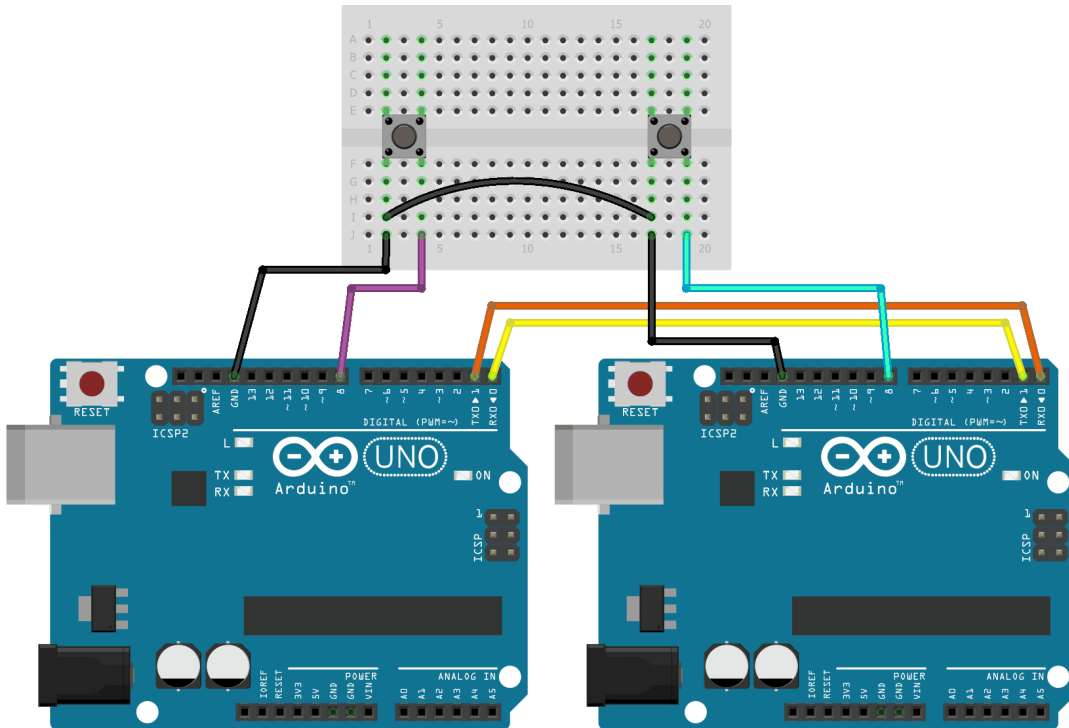
Analoga (ADC/DAC)

- Lettura di segnali analogici (sensori di temperatura, luce, pressione).
- Conversione ADC integrata nel microcontrollore.
- Uscite analogiche simulate tramite PWM o DAC (dove disponibile).



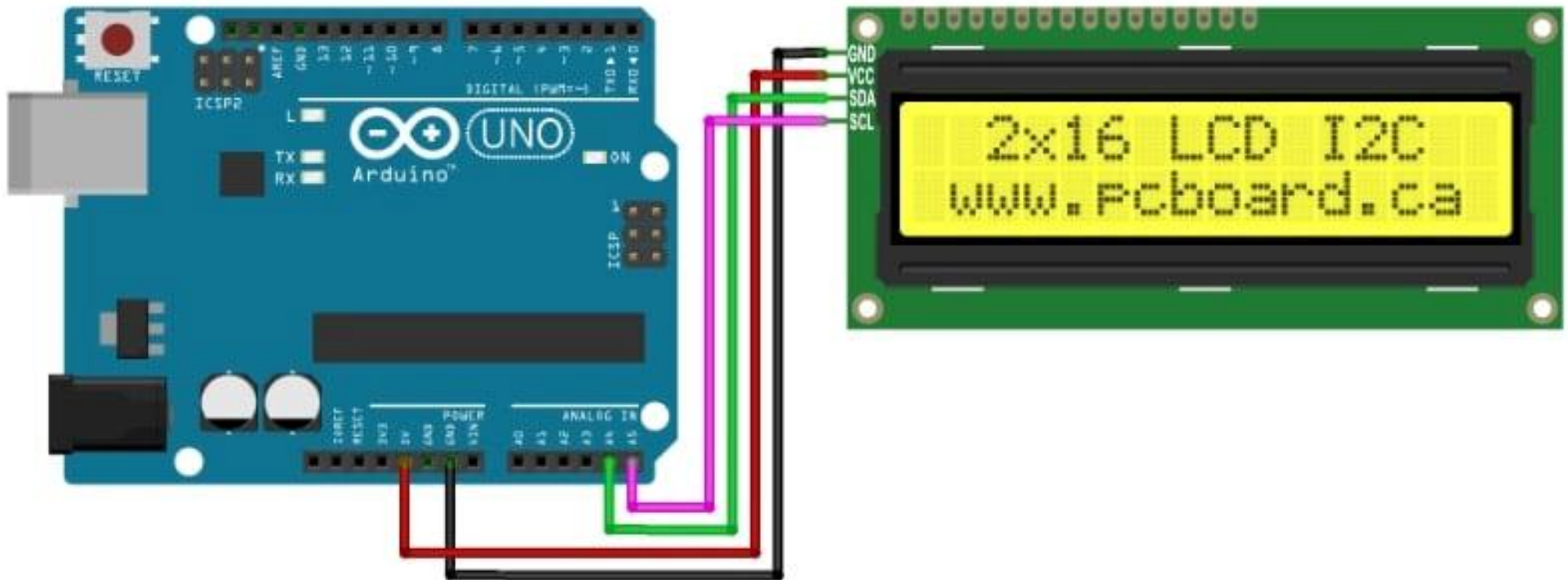
UART

- Comunicazione seriale asincrona punto-punto.
- Usata per debug e connessione con moduli esterni (GPS, Bluetooth, ecc.).
- Supportata nativamente dall'Arduino tramite i pin 0 (RX) e 1 (TX).



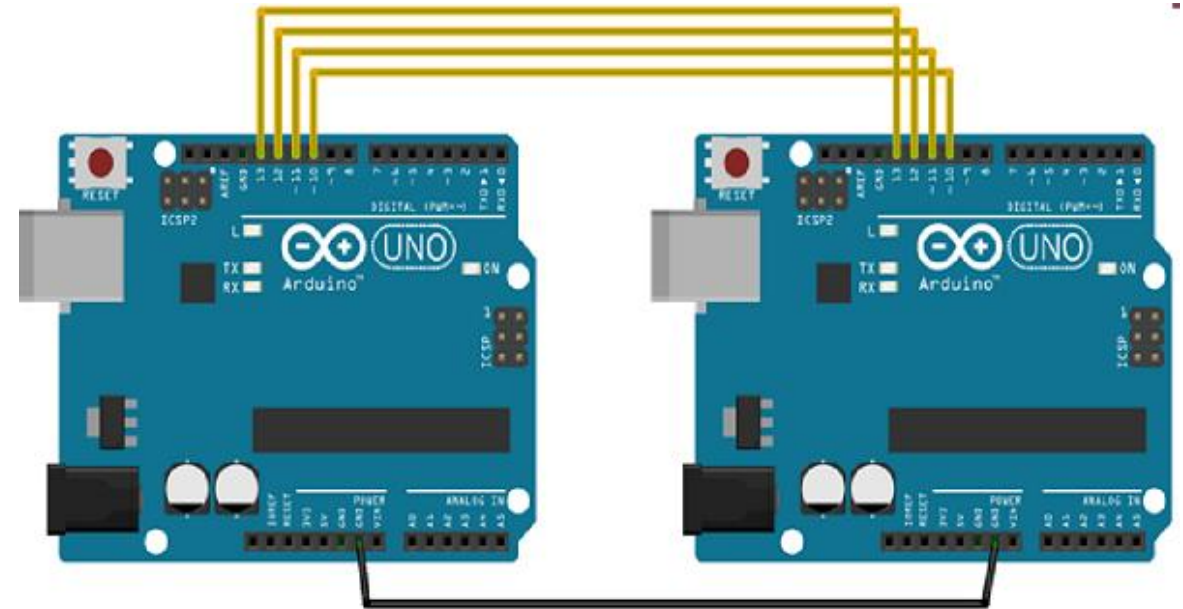
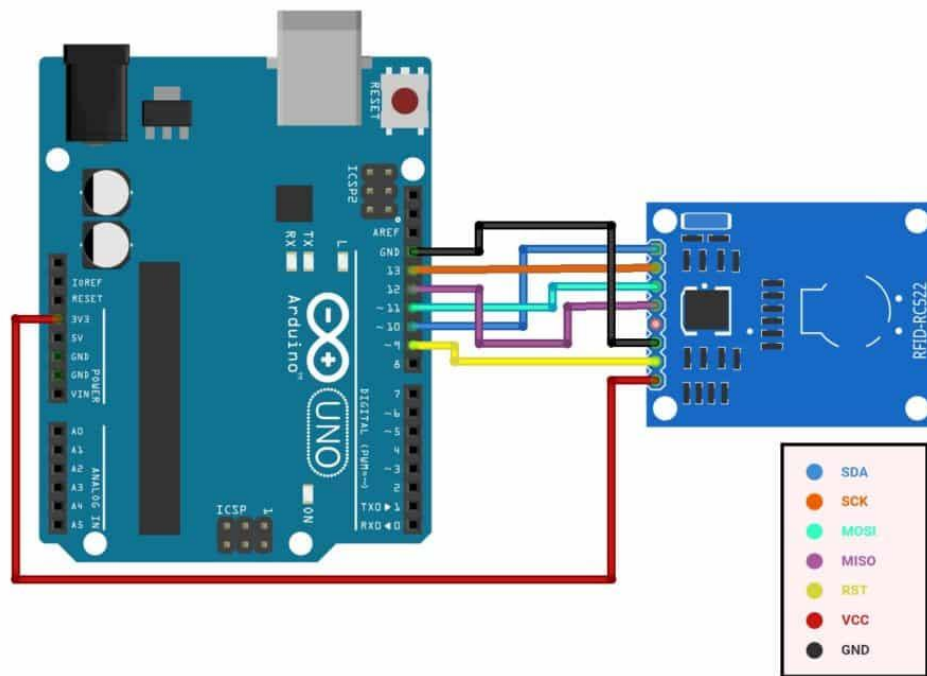
I²C

- Bus a 2 fili: SDA (dati) e SCL (clock).
- Permette la connessione di più dispositivi con indirizzamento.
- Molto usato per sensori, display, memorie EEPROM.

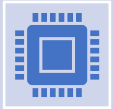


SPI

- Comunicazione veloce full-duplex.
- Pini: MOSI, MISO, SCK e SS.
- Utilizzata per display TFT, moduli SD, sensori ad alta velocità.



Altri protocolli (per IoT)



RS-485 e CAN: reti robuste per industria e automotive.



USB e Ethernet: interfacce per sistemi embedded avanzati.



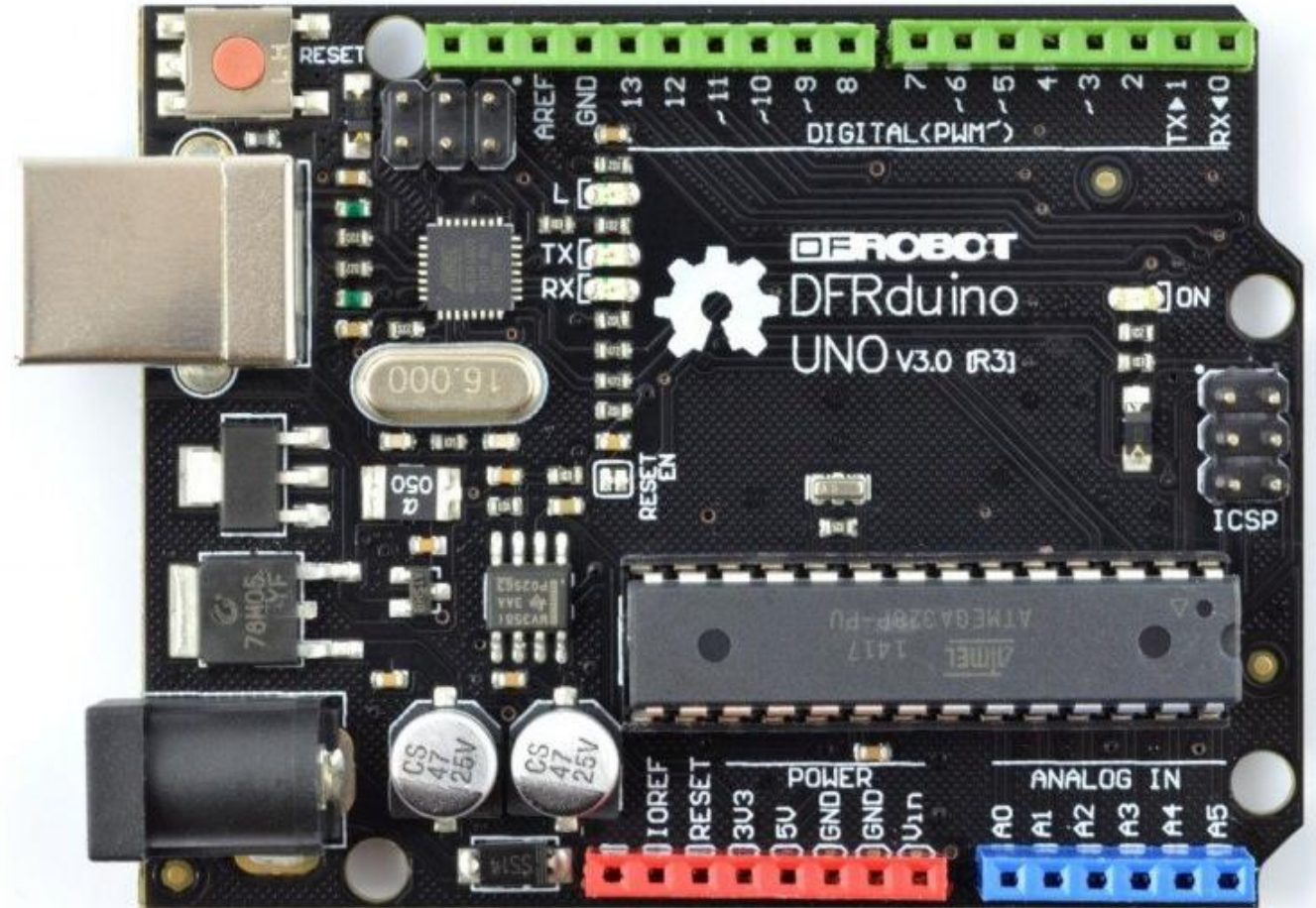
Wireless: Wi-Fi, BLE, Zigbee, LoRa/LoRaWAN, NB-IoT.



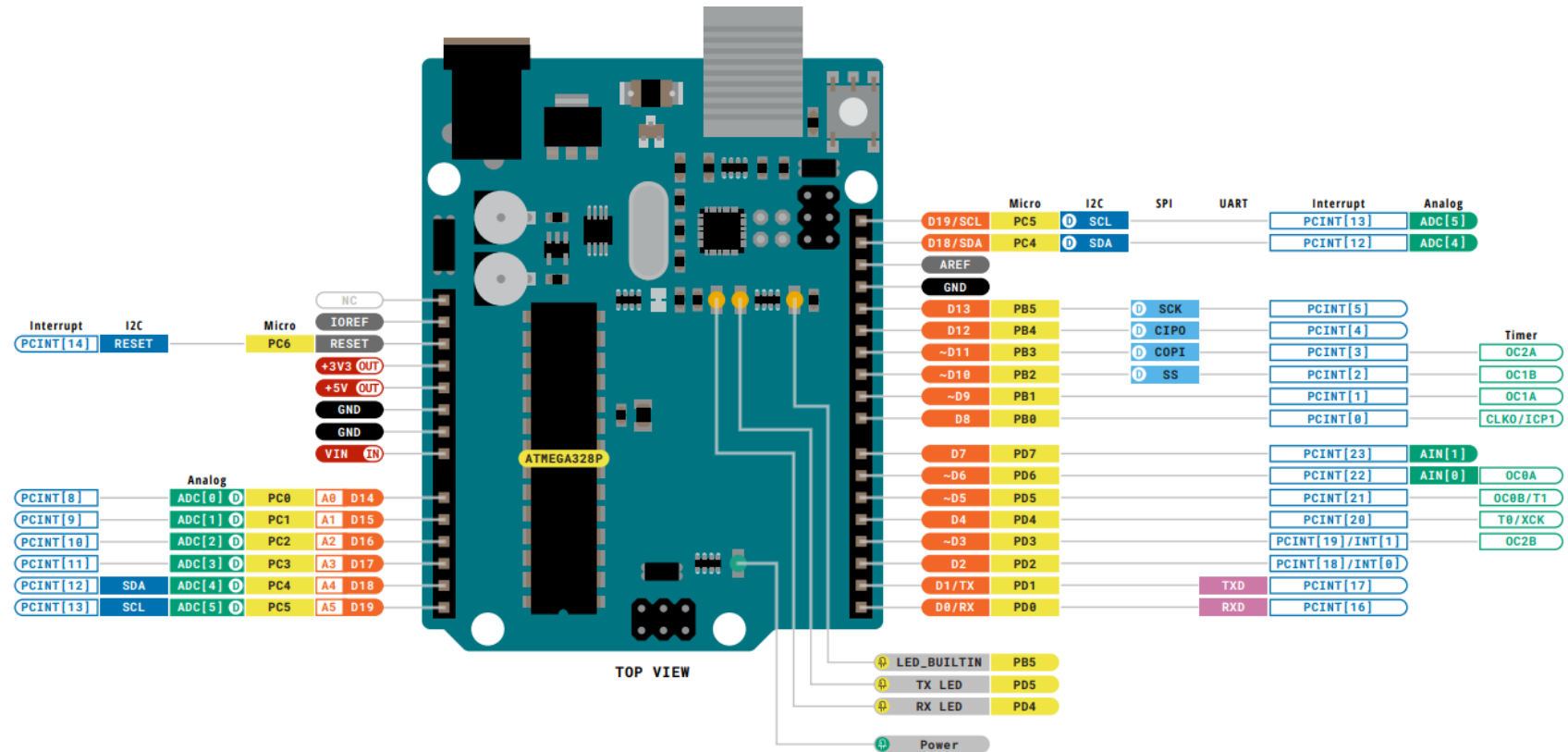
Abilitano applicazioni di smart house e smart city con connettività distribuita

Il nostro kit – Arduino Zero to Hero (DFRduino UNO R3)

- **Scheda principale:** DFRduino UNO R3 (compatibile Arduino UNO).
- **MCU:** ATmega328P a 16 MHz.
- **I/O:** 14 pin digitali (6 PWM), 6 ingressi analogici.
- **Memoria:** 32 KB Flash, 2 KB SRAM, 1 KB EEPROM.
- **Interfacce disponibili:** UART, I²C, SPI.



GPIO DFRduino UNO R3



Legend:				
Power	Power Input	GPIO Digital External	I2C	Default
Power Output	Power Output	Analog External	SPI	Default
Ground		Main Part	UART/USART	Default
		Secondary Part	Other SERIAL Communication	Default
		Internal Component	Analog	Default
		Other Pins (Reset, System Control, Debugging)	PWM/Timer	
			LED	
			RGB LED	
			Other	

MAXIMUM current per I/O pin is 20mA

MAXIMUM current per +3.3V pin is 50mA

VIN 6-20V input to the board

CIP0/COPI have previously been referred to as MISO/MOSI

ARDUINO UNO R3 BOARD

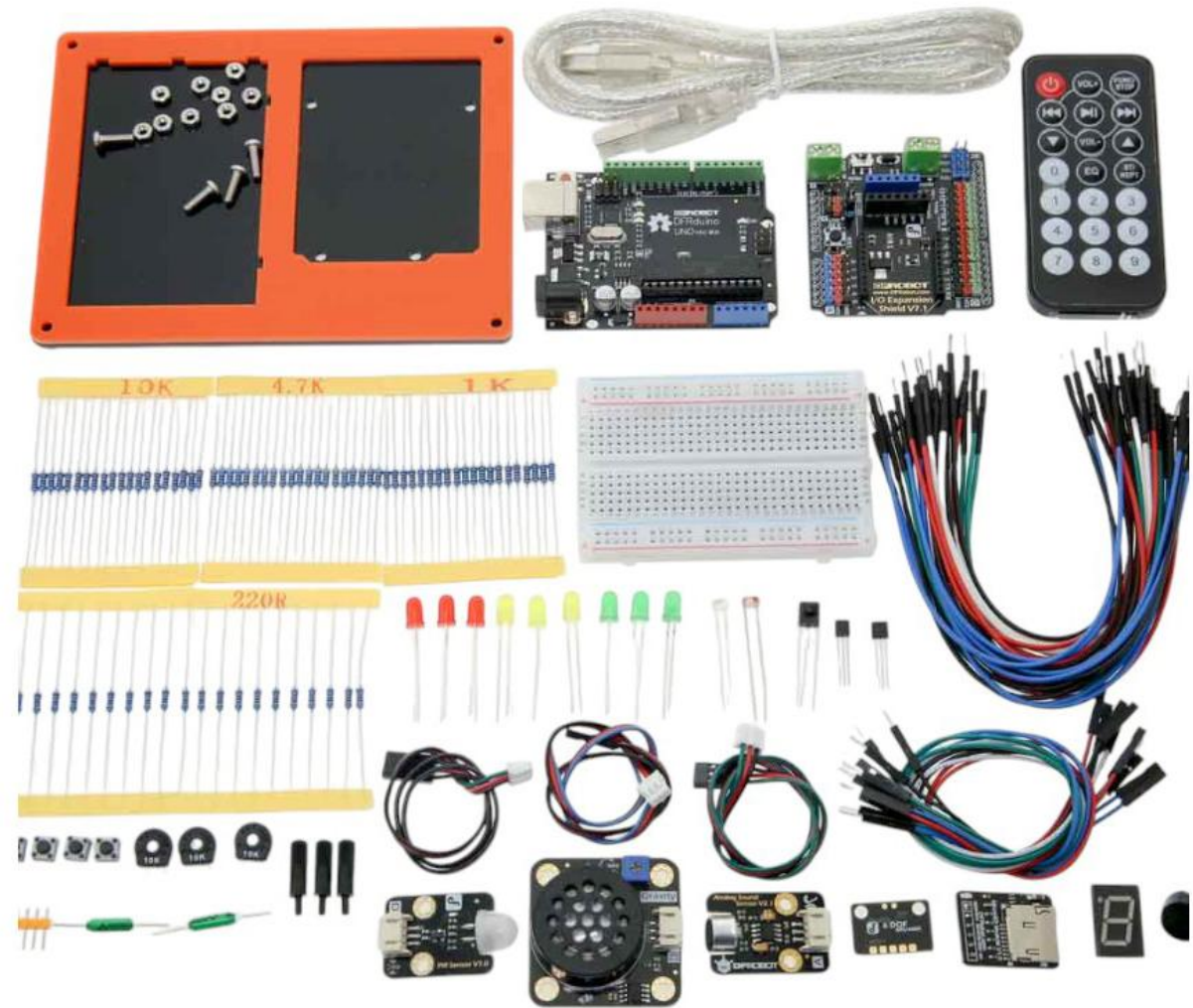
SKU code: A000066
Full Pinout - Page 3 of 5
Last update: 6 Oct, 2022

DOCS.ARDUINO.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 1868, Mountain View, CA 94042, USA.

Altri componenti del kit

- Resistenze, LED, pulsanti, transistor.
- Sensori (temperatura LM35, luce, tilt switch, PIR, microfono, MPU6050).
- Attuatori (buzzer, speaker, display 7 segmenti).
- Breadboard, jumper, moduli di espansione (microSD, IR, ecc.).

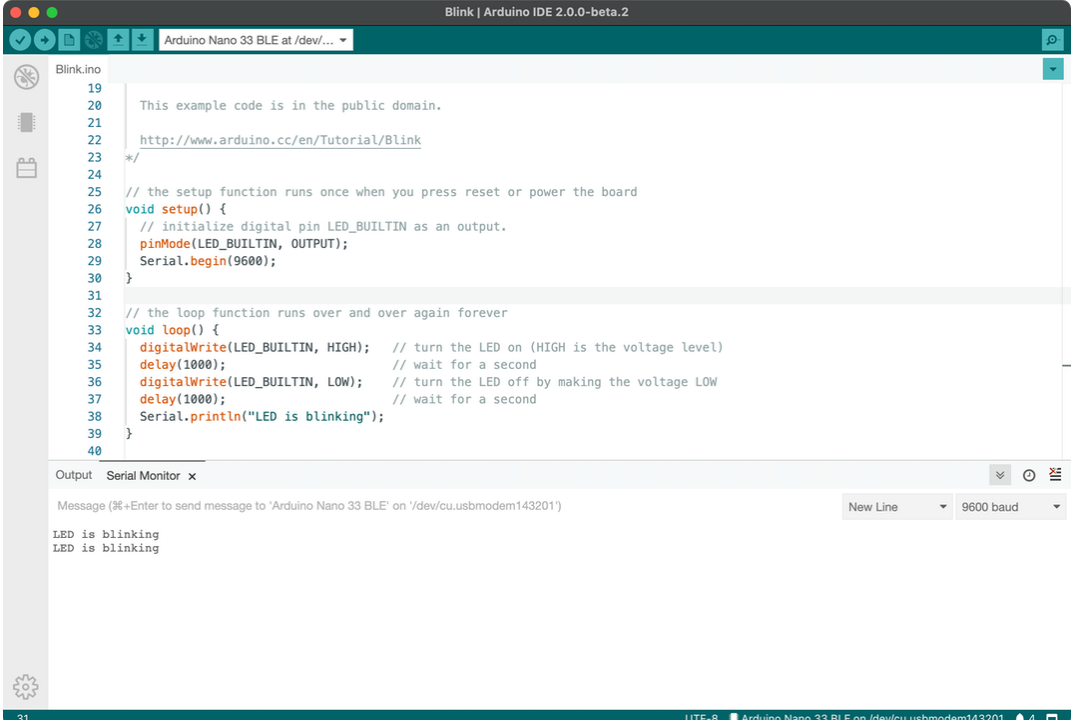


Lista Completa del kit

- | | |
|--|---|
| 1. DFRduino UNO R3 ×1 | 15. Sensore di inclinazione (Tilt Switch) ×1 |
| 2. Cavo USB A-B ×1 | 16. Display a 8 segmenti LED ×1 |
| 3. Cavi jumper M/M ×30 | 17. Sensore di temperatura LM35 ×1 |
| 4. Cavi jumper F/M ×10 | 18. Buzzer ×1 |
| 5. Resistore 220 Ω ×20 | 19. Potenzionometro 10 k Ω ×3 |
| 6. Resistore 4,7 k Ω ×20 | 20. Mini telecomando (con batteria CR2025 inclusa) ×1 |
| 7. Resistore 10 k Ω ×20 | 21. Breadboard da 400 punti senza saldatura, interbloccabile ×1 |
| 8. Resistore 1 k Ω ×20 | 22. Supporto in acrilico per breadboard e Arduino ×1 |
| 9. LED 5 mm ×10 | 23. Gravity: IO Expansion Shield per Arduino V7.1 ×1 |
| 10. Diodo ricevitore IR ×1 | 24. Gravity: Sensore analogico di suono per Arduino ×1 |
| 11. Mini pulsante (push button) ×4 | 25. Gravity: Modulo altoparlante digitale ×1 |
| 12. Sensore di luce ambientale PT5I850AC ×1 | 26. Gravity: Sensore PIR digitale (movimento) per Arduino ×1 |
| 13. Sensore di luce ambientale GL5528 (LDR) ×1 | 27. Sensore a 6 DOF – MPU6050 ×1 |
| 14. Transistor 2N3904 ×1 | 28. Modulo MicroSD per Arduino ×1 |

Introduzione all'Arduino IDE

- **L'Arduino IDE (Integrated Development Environment)** è l'ambiente di sviluppo usato per programmare la DFRduino UNO R3:
 - **Editor:** scrittura del codice in linguaggio C/C++ semplificato.
 - **Compilatore & Upload:** traduce il codice in linguaggio macchina e lo carica via USB nella memoria Flash del microcontrollore.
 - **Monitor seriale:** permette di inviare e ricevere dati attraverso la porta UART.
 - **Librerie:** semplificano l'uso di sensori, attuatori e protocolli di comunicazione.
 - **Esempi predefiniti:** come "Blink" e "Serial Monitor" per iniziare rapidamente.



The screenshot shows the Arduino IDE 2.0.0-beta.2 interface. The main editor window displays the 'Blink.ino' file with the following code:

```
19 This example code is in the public domain.
20
21 http://www.arduino.cc/en/Tutorial/Blink
22
23 */
24
25 // the setup function runs once when you press reset or power the board
26 void setup() {
27   // initialize digital pin LED_BUILTIN as an output.
28   pinMode(LED_BUILTIN, OUTPUT);
29   Serial.begin(9600);
30 }
31
32 // the loop function runs over and over again forever
33 void loop() {
34   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
35   delay(1000); // wait for a second
36   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
37   delay(1000); // wait for a second
38   Serial.println("LED is blinking");
39 }
40
```

Below the code editor, the 'Serial Monitor' window is open, showing the output: 'LED is blinking'.



Arduino IDE

Conclusioni

- I microcontrollori sono fondamentali nell'elettronica industriale per la loro flessibilità e costo contenuto.
- Consentono di realizzare sistemi di acquisizione, elaborazione e controllo in un unico dispositivo.
- L'uso del **kit Arduino** e dell'**IDE** fornisce una base pratica per sviluppare prototipi e comprendere il funzionamento dei sistemi di controllo.

Attività

- Identificazione dei componenti del kit.
- Analisi del pinout della scheda DFRduino UNO R3.
- Montaggio di un semplice circuito LED + resistenza sulla breadboard. (github)
- Scaricare [Arduino IDE](#) e installarlo su computer (prossime lezioni).

Dubbi o domande?
