

Le sfide dell'edge computing

Panoramica e casi d'uso

Alberto Cabri - Ph.D.

Vega Research Labs s.r.l.

DSS #11 21 luglio 2021 - Genova

VEGA
research
laboratories

Cosa si intende per **edge computing**?

- ◇ tecnologie hardware ed il relativo software applicativo, progettate per elaborare i dati dove vengono generati
- ◇ rappresentano il punto di accesso alla rete di un'azienda
- ◇ possono realizzare molteplici funzioni diverse

Ad esempio, il monitoraggio della temperatura o delle vibrazioni di una macchina in produzione.



1. **Introduzione**
2. Arduino
3. Raspberry PI
4. Jetson Nano
5. Conclusioni

i dispositivi per edge computing possono

- ◇ raccogliere dati da dispositivi legacy
- ◇ semplificare l'analisi elaborando i dati all'origine
- ◇ possono estrarre informazioni in tempo reale
- ◇ ridurre la mole di dati trasferiti ai server di memorizzazione



1. **Introduzione**
2. Arduino
3. Raspberry PI
4. Jetson Nano
5. Conclusioni

- ◇ la sicurezza diventa locale ad ogni dispositivo: eventuali violazioni di sicurezza non interessano l'intero sistema
- ◇ possono operare in modalità disconnessa in aree dove la rete non è sempre disponibile
- ◇ possono semplificare le problematiche di privacy eliminando la memorizzazione di dati sensibili
- ◇ l'elaborazione in real-time può richiedere elevata capacità di calcolo



1. **Introduzione**
2. Arduino
3. Raspberry Pi
4. Jetson Nano
5. Conclusioni

La struttura tipica contiene

- ◇ sensori od attuatori, per raccogliere i dati da apparecchiature remote oppure per eseguire azioni da remoto
- ◇ conversione A/D e D/A delle misure rilevate o delle impostazioni
- ◇ un micro-controllore per effettuare pre-elaborazione ed analisi dati decentralizzate
- ◇ un accesso alla rete per connettere i sistemi aziendali



1. **Introduzione**
2. Arduino
3. Raspberry PI
4. Jetson Nano
5. Conclusioni

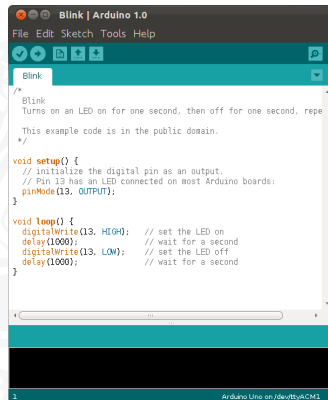
- ◇ è una piattaforma hardware per il **rapid prototyping** con molte schede a micro-controllore
- ◇ si integra facilmente con sensori ed attuatori
- ◇ totalmente **Open** in hardware e software
- ◇ si avvale di un **IDE multi-piattaforma** che utilizza un linguaggio di programmazione C-like
- ◇ i programmi sono denominati **sketches**

Il sito web di riferimento è www.arduino.cc



1. Introduzione
2. **Arduino**
3. Raspberry PI
4. Jetson Nano
5. Conclusioni

- ◇ Open Source
- ◇ disponibile su Windows, Linux e MacOS
- ◇ programmazione semplice con controllo sintattico **inline**
- ◇ supporta la compilazione degli sketches ed il caricamento su tutte le schede
- ◇ supporta l'intera famiglia di dispositivi Arduino



The screenshot shows the Arduino IDE window titled "Blink | Arduino 1.0". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu bar is a toolbar with icons for opening, saving, and running sketches. The main text area displays the "Blink" sketch, which is a classic example for controlling an LED. The code is as follows:

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards;
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);             // wait for a second
  digitalWrite(13, LOW);  // set the LED off
  delay(1000);             // wait for a second
}
```

At the bottom of the window, a status bar indicates "1" and "Arduino Uno on /dev/ttyACM1".



1. Introduzione
2. **Arduino**
3. Raspberry Pi
4. Jetson Nano
5. Conclusioni

- ◇ attualmente disponibile in 24 diverse versioni
- ◇ utilizzano micro-controllori della serie megaAVR
- ◇ sono dotati di un regolatore di tensione ed un oscillatore al quarzo
- ◇ le schede di espansione sono chiamate **shields**
- ◇ gli shield disponibili sono elencati in <http://shieldlist.org>

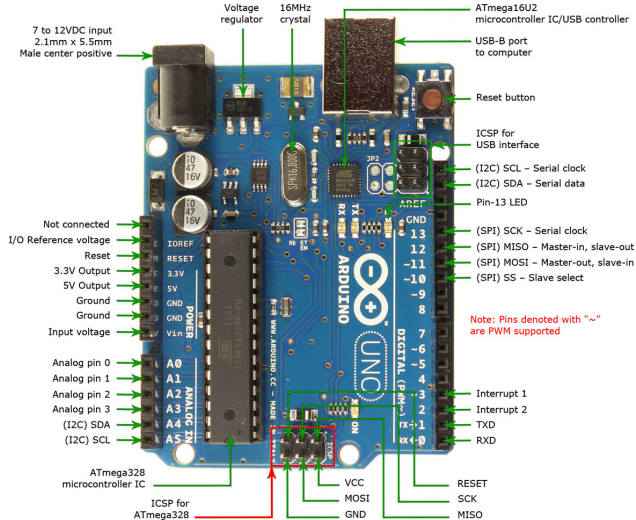


1. Introduzione
2. **Arduino**
3. Raspberry PI
4. Jetson Nano
5. Conclusioni

- ◇ Micro-controllore: ATmega328P (RISC 8-bits) @ 16MHz di clock
- ◇ Tensione di Lavoro: 5V
- ◇ Alimentazione suggerita: 7-12V
- ◇ 14 pin di I/O digitale, dei quali 6 per le uscite PWM (8-bits)
- ◇ 6 pin di ingresso analogico
- ◇ Massima corrente di uscita 20mA
- ◇ 32KB di memoria Flash, 2KB SRAM



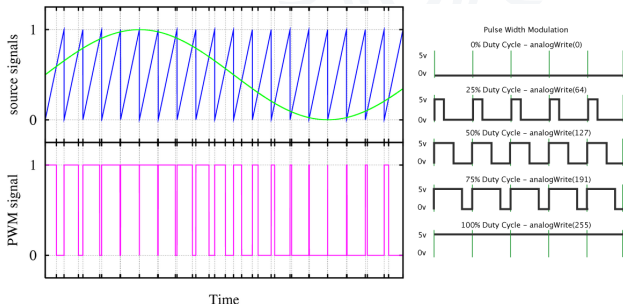
1. Introduzione
2. **Arduino**
3. Raspberry PI
4. Jetson Nano
5. Conclusioni



1. Introduzione
2. **Arduino**
3. Raspberry PI
4. Jetson Nano
5. Conclusioni

PWM o Pulse Width Modulation

- ◇ è una modulazione digitale usata per ottenere una tensione media che dipende dal **duty cycle** $d = \frac{t_{L1}}{T}$
- ◇ può essere utilizzata per variare la velocità di un motore elettrico o la luminosità di un diodo LED



1. Introduzione
2. **Arduino**
3. Raspberry PI
4. Jetson Nano
5. Conclusioni

Lo sketch contiene

- ◇ una sezione per definire le variabili globali e le costanti
- ◇ una funzione di **setup** eseguita al boot del dispositivo per l'inizializzazione delle variabili o le impostazioni di default
- ◇ una funzione **loop** eseguita in modo continuo che contiene il vero programma



1. Introduzione
2. **Arduino**
3. Raspberry Pi
4. Jetson Nano
5. Conclusioni

Codice per far lampeggiare il LED integrato (associato al pin 13)

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```



1. Introduzione
2. **Arduino**
3. Raspberry PI
4. Jetson Nano
5. Conclusioni

laboratories

- ◇ è una serie di computer a basso costo, su singola scheda, **credit card sized**
- ◇ si collega ad un monitor o un TV
- ◇ supporta tastiera e mouse standard
- ◇ può fare le stesse cose di un desktop computer
- ◇ supporta i principali linguaggi di programmazione su sistemi operativi dedicati
- ◇ spesso utilizzato come **headless computer**

Il sito web di riferimento è www.raspberrypi.org



1. Introduzione
2. Arduino
3. **Raspberry PI**
4. Jetson Nano
5. Conclusioni

- ◇ attualmente disponibile in 13 differenti versioni
- ◇ monta CPU compatibili ARM o processori ARM-Cortex originali
- ◇ alcuni modelli offrono connettività di rete ethernet o wireless
- ◇ dimensione RAM da 256MB a 8GB
- ◇ connettore GPIO da 26 oppure 40 pin a seconda del modello
- ◇ real-time clock disponibile su scheda aggiuntiva



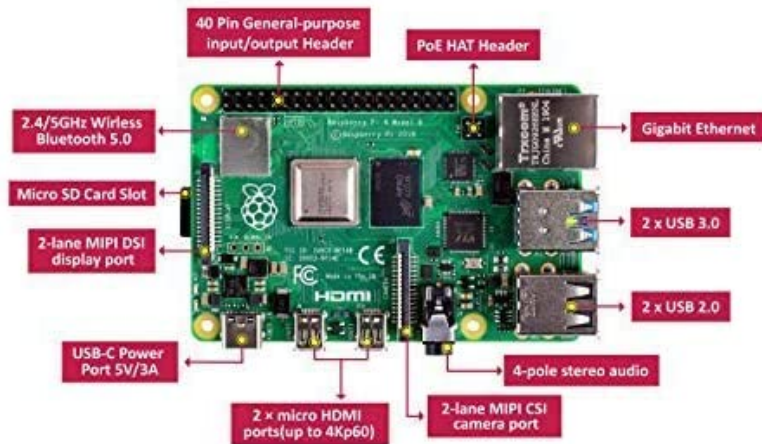
1. Introduzione
2. Arduino
3. **Raspberry PI**
4. Jetson Nano
5. Conclusioni

Le caratteristiche principali sono

- ◇ Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- ◇ Alimentazione: 5V (minimo 3A)
- ◇ 2GB, 4GB o 8GB RAM a seconda del modello
- ◇ connettore GPIO da 40 pin
- ◇ 2 porte micro HDMI per connettere fino a 2 schermi
- ◇ connettività wireless 2.4 GHz e 5.0 GHz IEEE 802.11ac e Gigabit Ethernet
- ◇ Bluetooth 5.0, BLE



1. Introduzione
2. Arduino
3. **Raspberry PI**
4. Jetson Nano
5. Conclusioni



1. Introduzione
2. Arduino
3. **Raspberry PI**
4. Jetson Nano
5. Conclusioni

- ◇ sono disponibili molti sistemi operativi per il Raspberry Pi
- ◇ Raspberry Pi OS è il sistema operativo ufficiale
- ◇ Sono supportati anche altri sistemi operativi di terze parti
 - ◇ Ubuntu Desktop, Server e Core per progetti IoT
 - ◇ LibreELEC o OSMC per i media servers
 - ◇ Mozilla WebThings Gateway per progetti IoT
 - ◇ Kali Linux
 - ◇ OpenMediaVault per il Network Attached Storage



1. Introduzione
2. Arduino
3. **Raspberry Pi**
4. Jetson Nano
5. Conclusioni

- ◇ è un piccolo e potente computer che permette l'esecuzione parallela di reti neurali per le applicazioni
- ◇ è una piattaforma di facile utilizzo che consuma anche solo 5W
- ◇ appartiene ad una famiglia di moduli con accelerazione GPU per edge computing ad alte prestazioni
- ◇ utilizza NVIDIA JetPack SDK (una distribuzione basata su Ubuntu) per realizzare applicazioni AI di ridotta complessità

Il sito web di riferimento è developer.nvidia.com



1. Introduzione
2. Arduino
3. Raspberry PI
4. **Jetson Nano**
5. Conclusioni

- ◇ è un circuito elettronico specializzato per la creazione di immagini in un **frame buffer**
- ◇ consente di accelerare applicazioni di HPC, apprendimento automatico ed altro
- ◇ originariamente utilizzate per accelerare la grafica 3D e le operazioni video
- ◇ usate anche nei sistemi embedded, nei PC o nelle console di gioco
- ◇ CUDA o Compute Unified Device Architecture è un'architettura hardware di NVIDIA per l'elaborazione parallela
- ◇ l'ambiente di sviluppo CUDA semplifica la creazione di applicazioni di calcolo parallelo



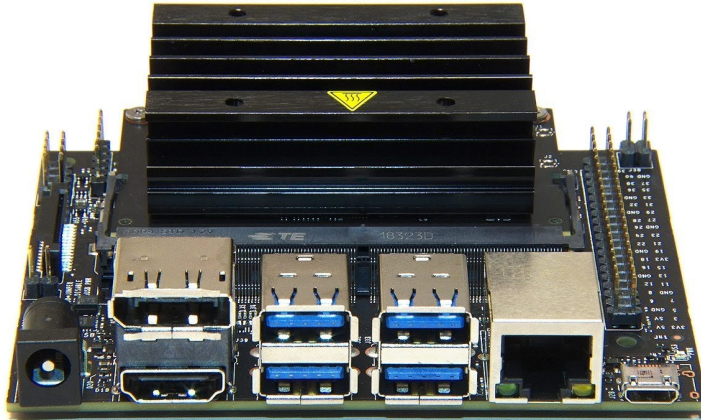
1. Introduzione
2. Arduino
3. Raspberry PI
4. **Jetson Nano**
5. Conclusioni

Le principali caratteristiche sono

- ◇ Quad-core ARM A57 @ 1.43 GHz
- ◇ Alimentazione: 5V
- ◇ GPU 128-core
- ◇ 4GB RAM
- ◇ connettore GPIO a 40 pin
- ◇ HDMI e display port
- ◇ Gigabit Ethernet (no wi-fi)



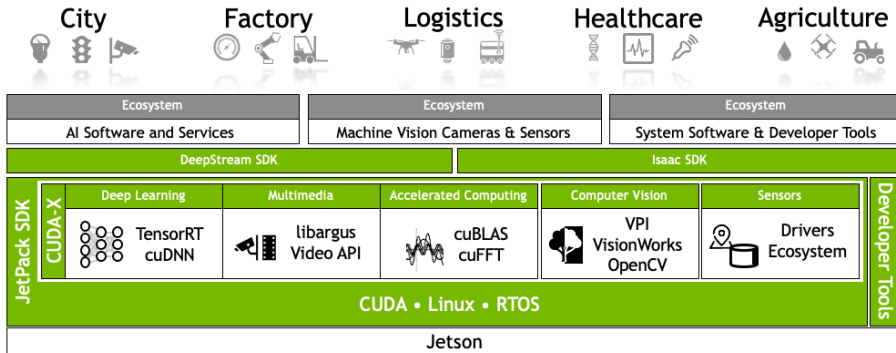
1. Introduzione
2. Arduino
3. Raspberry PI
4. **Jetson Nano**
5. Conclusioni



1. Introduzione
2. Arduino
3. Raspberry PI
4. **Jetson Nano**
5. Conclusioni

laboratories

Software per il Jetson Nano



1. Introduzione
2. Arduino
3. Raspberry PI
4. **Jetson Nano**
5. Conclusioni

La configurazione del Jetson

- ◇ dipende dal framework di AI da utilizzare
- ◇ dipende dalla versione di NVIDIA SDK
- ◇ richiede l'installazione di particolari versioni di pacchetti
- ◇ deve ripartire da zero in caso di errore per garantire la riproducibilità dell'ambiente operativo
- ◇ deve essere automatizzabile e facilmente portabile anche su altre piattaforme



1. Introduzione
2. Arduino
3. Raspberry PI
4. **Jetson Nano**
5. Conclusioni

Per semplificare la creazione di una soluzione con tali caratteristiche si può utilizzare **docker**

Docker è un ambiente di virtualizzazione a livello di sistema operativo linux che consente la creazione ed esecuzione di *container* indipendenti sulla stessa macchina.

- ◇ è una virtualizzazione leggera
- ◇ ogni container è isolato dagli altri in termini di processi e risorse
- ◇ consente la creazione rapida, il test e la distribuzione di applicazioni
- ◇ portabilità con zero o poche modifiche



1. Introduzione
2. Arduino
3. Raspberry PI
4. **Jetson Nano**
5. Conclusioni

- ◇ installazione e documentazione su docs.docker.com
- ◇ richiede un Dockerfile per creare l'immagine dell'applicazione
- ◇ tutte le modifiche a runtime vengono perse allo shutdown del container
- ◇ le aree di storage persistente devono essere configurate esplicitamente
- ◇ l'utilizzo di X11 o l'accesso ai device (ad es. webcam) richiedono opzioni specifiche per **docker run**

```
# Build this Dockerfile by running the following commands:
#
#   $ docker build -t alkabris/yolov5-pytorch:0.0.2
#
#   This is a working version with torchvision 0.8.0a
#
ARG BASE_IMAGE=nvcr.io/nvidia/14t-pytorch:r32.5.0-pt1.7-py3
FROM ${BASE_IMAGE}

# environment setup
#
ENV DEBIAN_FRONTEND=noninteractive
ENV SHELL /bin/bash
ENV CUDA_HOME="/usr/local/cuda"
ENV PATH="/usr/local/cuda/bin:${PATH}"
ENV LD_LIBRARY_PATH="/usr/local/cuda/lib64:${LD_LIBRARY_PATH}"

WORKDIR /yolov5-pytorch

# install procedure for Jetpack4.5
#
RUN apt-get update -y && \
    apt-get install -y --no-install-recommends \
    python3-pip \
    python3-dev \
    python3-matplotlib \
    python3-numpy \
    python3-tk \
    build-essential \
    git \
    jetson@jetson:~/airaw/yolo/yolov5-tensorrt/docker$
```



1. Introduzione
2. Arduino
3. Raspberry PI
4. **Jetson Nano**
5. Conclusioni

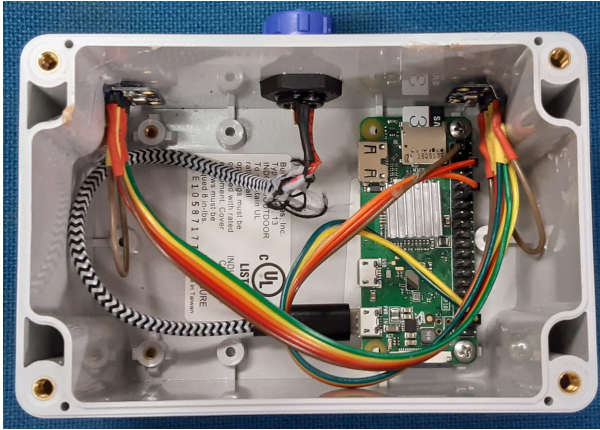
Due possibili applicazioni di esempio:

- ◇ un dispositivo per la registrazione ed il trasferimento di flussi audio in rete
 - ◇ Raspberry PI Zero con Raspbian OS e Python
 - ◇ Microfono MEMS in stereo
 - ◇ flusso WAV non compresso a blocchi di 5 sec
- ◇ una soluzione per il riconoscimento di componenti elettronici in tempo reale su Jetson Nano (demo video su YouTube)

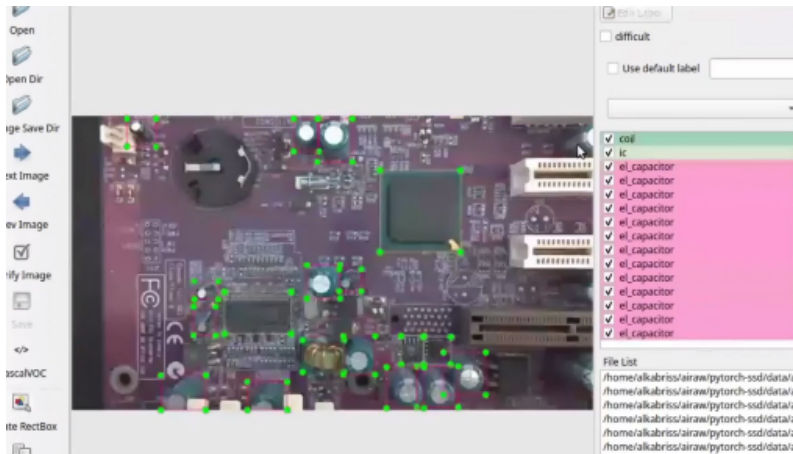


1. Introduzione
2. Arduino
3. Raspberry PI
4. **Jetson Nano**
5. Conclusioni

Streaming audio in real-time con RaspberryPI Zero



1. Introduzione
2. Arduino
3. Raspberry PI
4. **Jetson Nano**
5. Conclusioni



1. Introduzione
2. Arduino
3. Raspberry PI
4. **Jetson Nano**
5. Conclusioni

Selezionare l'immagine per riprodurre da YouTube

Edge computing

- ◇ consente un risparmio di **bandwidth** inviando soltanto informazioni utili e fruibili
- ◇ consente una riduzione della **latenza** evitando inutili trasferimenti di dati
- ◇ incrementa la **sicurezza** grazie alla raccolta dati distribuita
- ◇ favorisce il rispetto della **privacy** non registrando dati sensibili che vengono invece elaborati on-the-edge
- ◇ migliore affidabilità, grazie allo store & forward, e scalabilità
- ◇ potrebbe trascurare informazioni di valore se non finalizzate al principale obiettivo



1. Introduzione
2. Arduino
3. Raspberry PI
4. Jetson Nano
5. **Conclusioni**

Thank you!



Alberto Cabri



alberto.cabri@vegaresearchlabs.com



+39-340-9640772