

Documentazione Completa — JetRacer AI Kit

1. Componenti Hardware

Il kit JetRacer AI Kit comprende i seguenti elementi principali:

- **Scheda di calcolo**

- NVIDIA **Jetson Nano Developer Kit (B01)**
- 4 GB RAM, quad-core ARM A57 CPU, GPU Maxwell 128 CUDA cores

- **Telaio e meccanica**

- Telaio in acrilico/alluminio
- Ruote in gomma racing
- Motore DC (x2) con encoder
- Servo per sterzo
- Supporti per camera e scheda

- **Scheda di espansione JetRacer**

- Alimentazione 5 V per Jetson Nano
- Driver motore duale **TB6612FNG**
- Sensore corrente/tensione **INA219**
- ADC **ADS1115** per sensori esterni
- Expander I²C **PCA9685** per servo/motori
- Display **OLED 0.91” 128×32**
- Circuito di protezione batterie **S-8254AA + AO4407A**
- Connettori dedicati per ventole, motori, servo, camera CSI

- **Alimentazione**

- Alloggiamento per **3 batterie 18650** (non incluse)
- Regolatore step-down **APW7313** per stabilizzare 5 V
- Carica tramite modulo di bilanciamento incluso

- **Camera**

- Fotocamera CSI 160° grandangolare, risoluzione 5MP

- **Altro**
 - Ventola di raffreddamento
 - Modulo WiFi (in base alla versione del Jetson Nano)
 - Cavi e minuteria per assemblaggio
-

2. Componenti Software

Il lato software del JetRacer è organizzato in diversi livelli:

2.1 Sistema operativo

- **JetPack 4.5 / Ubuntu 18.04 LTS** per Jetson Nano

(contiene driver NVIDIA, CUDA, cuDNN, TensorRT, OpenCV ottimizzato)

2.2 Librerie principali

- **PyTorch** (torch, torchvision) — deep learning
- **Torch2TRT** — ottimizzazione modelli per TensorRT
- **NumPy, Matplotlib, Pandas** — supporto scientifico
- **OpenCV** — visione artificiale
- **traitlets** — gestione configurazioni dinamiche
- **Adafruit libraries** — controllo componenti I²C/SPI (OLED, PCA9685, ecc.)

2.3 Framework e tool di controllo

- **Jupyter Lab** (server preinstallato sul Jetson Nano)

→ consente di eseguire notebook Python dal browser

- **JetRacer Python API** — libreria ufficiale per:
 - Controllo sterzo (car.steering)
 - Controllo accelerazione (car.throttle)
 - Parametri di tuning (steering_gain, steering_offset, throttle_gain)
- **Gamepad/Teleoperation module** — controllo remoto via joystick
- **DonkeyCar Framework** — supportato per auto-driving alternativo

2.4 Esempi software inclusi

- **Basic motion** — test di base dei motori e servo
- **Teleoperation** — controllo manuale via gamepad o browser

- **Interactive regression** — raccolta dati, training modello AI (regressione lineare/neural net)
 - **Road following** — guida autonoma basata su rete neurale addestrata (line-following)
 - **Camera calibration & display** — test e calibrazione della fotocamera
 - **Model optimization** — conversione dei modelli in TensorRT per velocizzare l'inferenza
-

3. Architettura del sistema

1. Sensori e attuatori

- Camera CSI cattura immagini in tempo reale
- Encoder motori → velocità/posizione
- Servo → sterzo
- INA219 → monitoraggio corrente/tensione

2. Elaborazione Edge

- Jetson Nano esegue modelli AI (PyTorch/TensorRT) in locale
- I notebook Jupyter consentono di addestrare e validare i modelli direttamente sul veicolo

3. Controllo e interfaccia

- Accesso via browser al server Jupyter (<http://<IP>:8888>)
- Possibile uso di gamepad USB per teleoperation
- OLED sul veicolo mostra info di rete, stato batteria, memoria

4. Cloud / PC esterno (opzionale)

- Training più pesante può essere eseguito su PC esterno con GPU, poi modello trasferito sul JetRacer
-

4. Funzioni principali dimostrative

- **Guida autonoma su pista** (road following con reti neurali)
- **Controllo remoto manuale** (gamepad / browser)
- **Apprendimento interattivo** (raccolta dati + training in tempo reale)
- **Integrazione con DonkeyCar** (piattaforma di auto-driving open source)

- **Monitoraggio energetico in tempo reale** (display OLED + INA219)