



UNIVERSITÀ DEGLI STUDI
DI GENOVA

CORSO DI LAUREA IN INGEGNERIA ELETTRONICA E TECNOLOGIE DELL'INFORMAZIONE

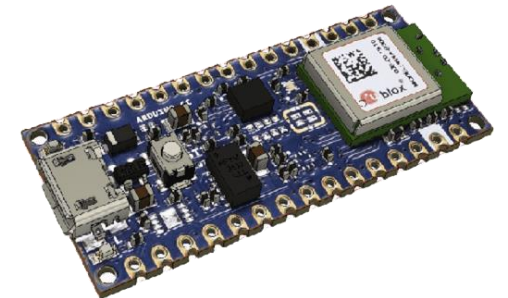
Progetto e implementazione di un sistema embedded per il controllo remoto di auto radiocomandate

Candidato: Lorenzo Sgrò
Relatore: Prof. Riccardo Berta
Correlatore: Dott. Matteo Fresta

OBIETTIVO DEL PROGETTO

L'obiettivo di questo progetto è quello di modificare una macchina radiocomandata esistente, sostituendo il suo meccanismo di comunicazione originale con una applicazione smartphone basata su Bluetooth Low Energy (BLE) e un microcontrollore Arduino Nano 33 BLE Sense

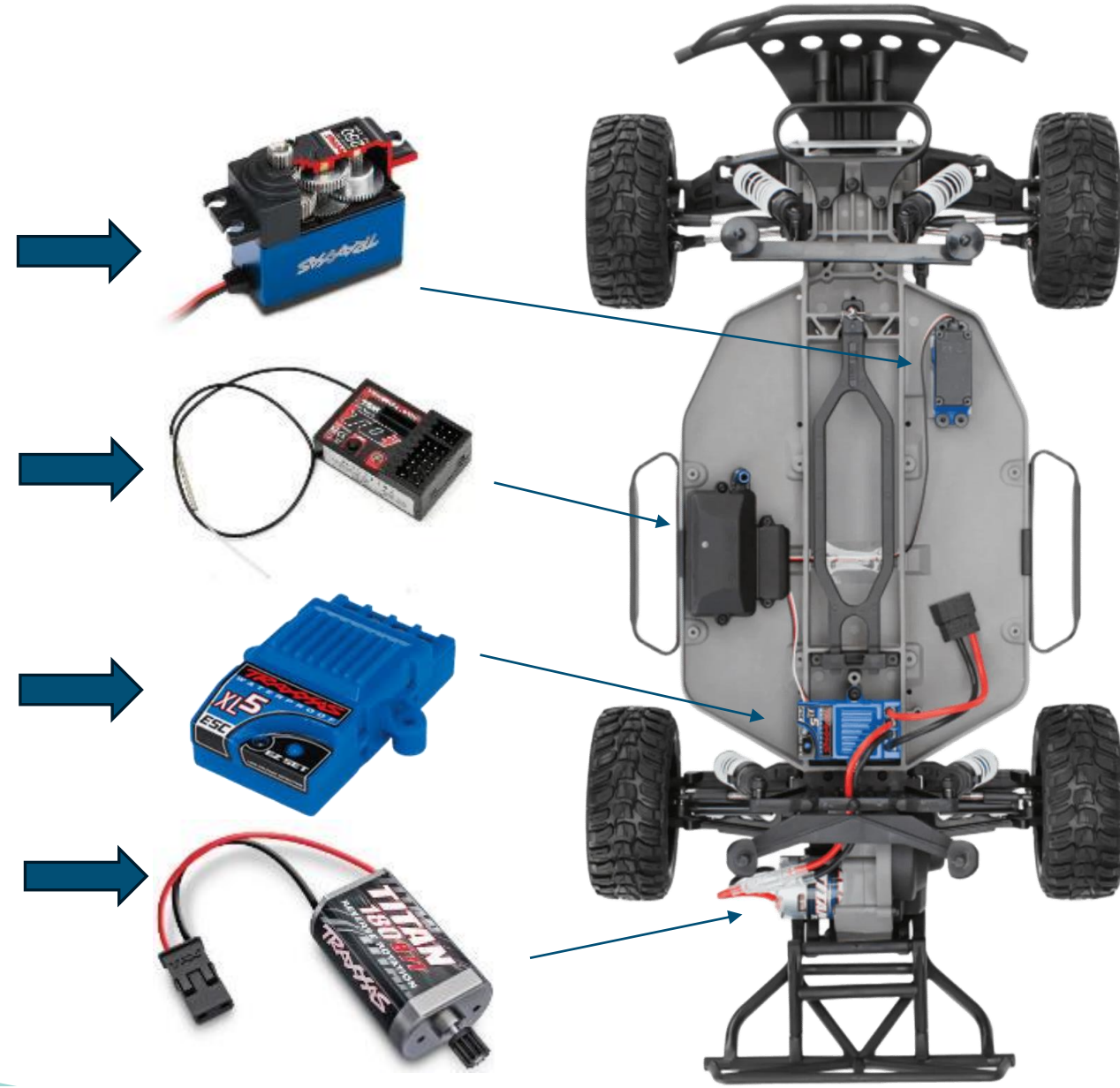
Così facendo sarà possibile nei prossimi sviluppi del progetto implementare algoritmi di guida autonoma



ANALISI PRELIMINARE (REVERSE ENGINEERING)

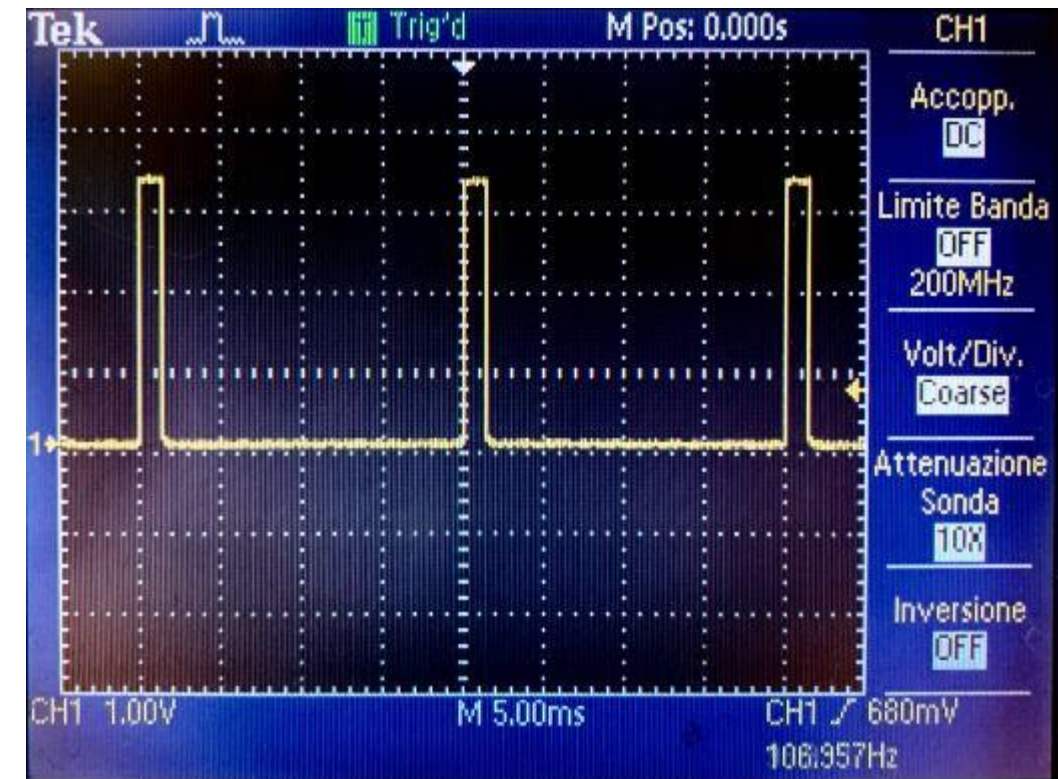
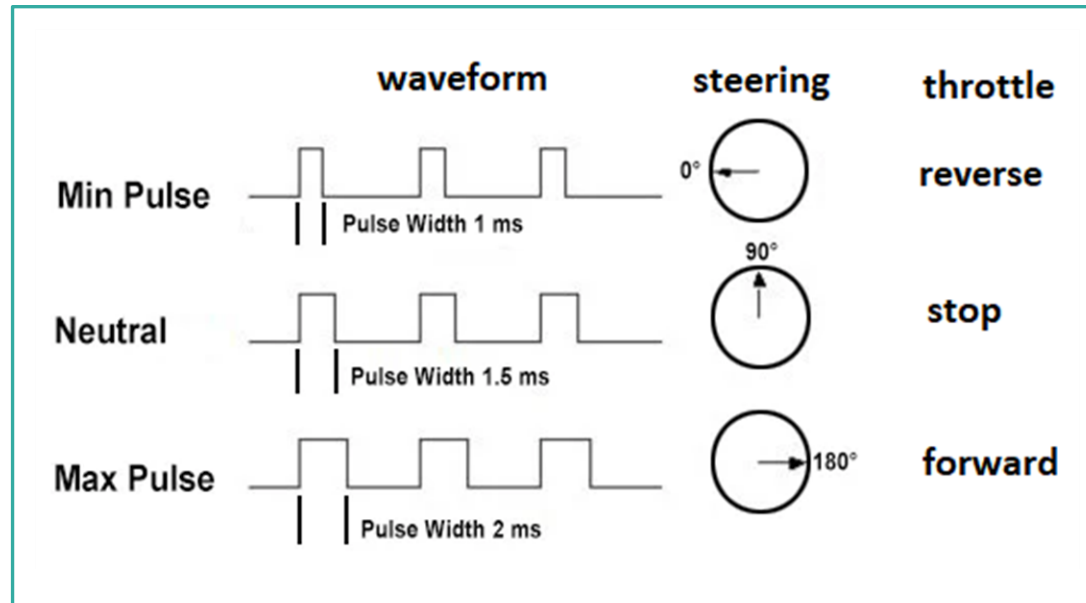
COMPONENTI DELLA MACCHINA:

- **Servomotore:** comanda il movimento dello sterzo
- **Ricevitore:** riceve i comandi dal telecomando e li trasmette al servomotore e all'ESC
- **ESC (Electronic Speed Control):** comanda la velocità di rotazione del motore
- **DC motor:** fornisce trazione alle ruote permettendo il movimento della macchina



SEGNALE RILEVATO CON OSCILLOSCOPIO

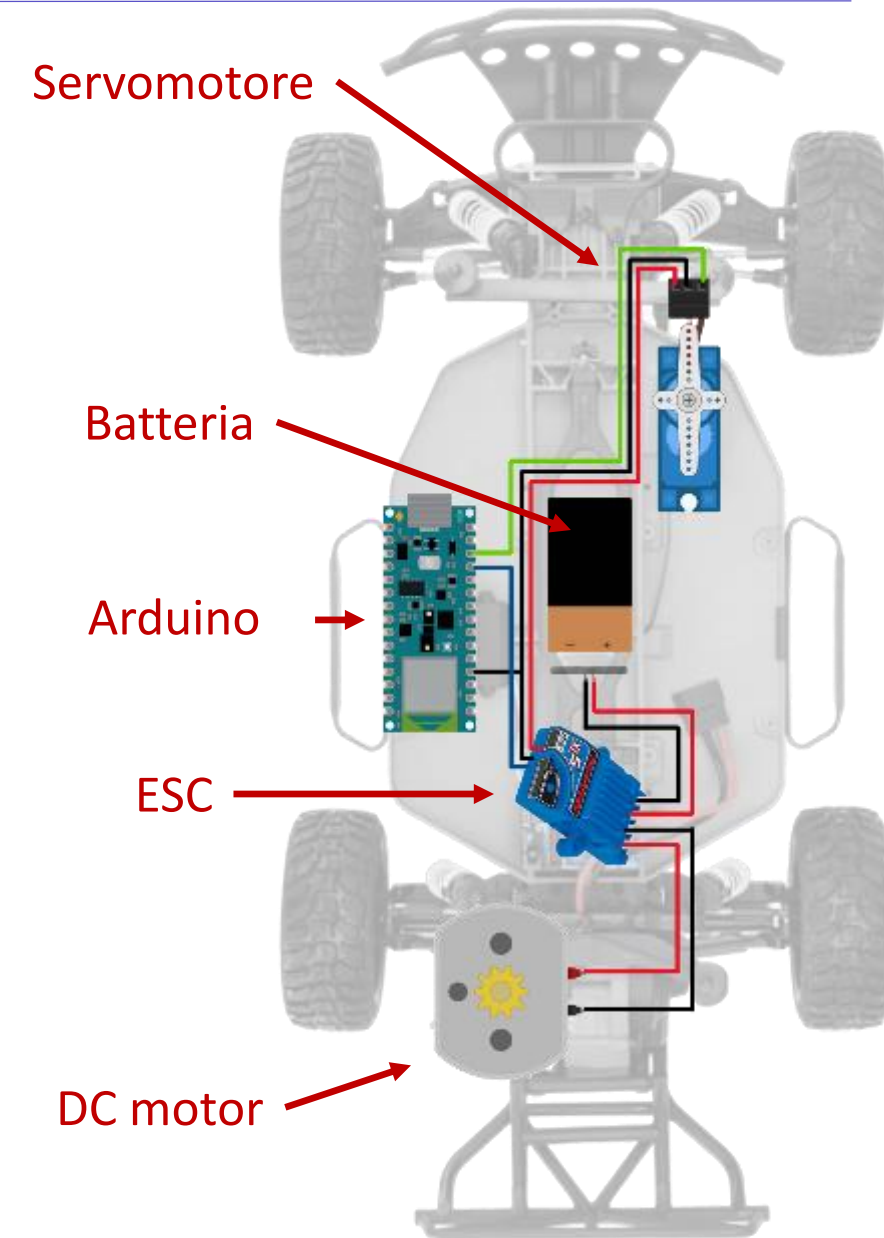
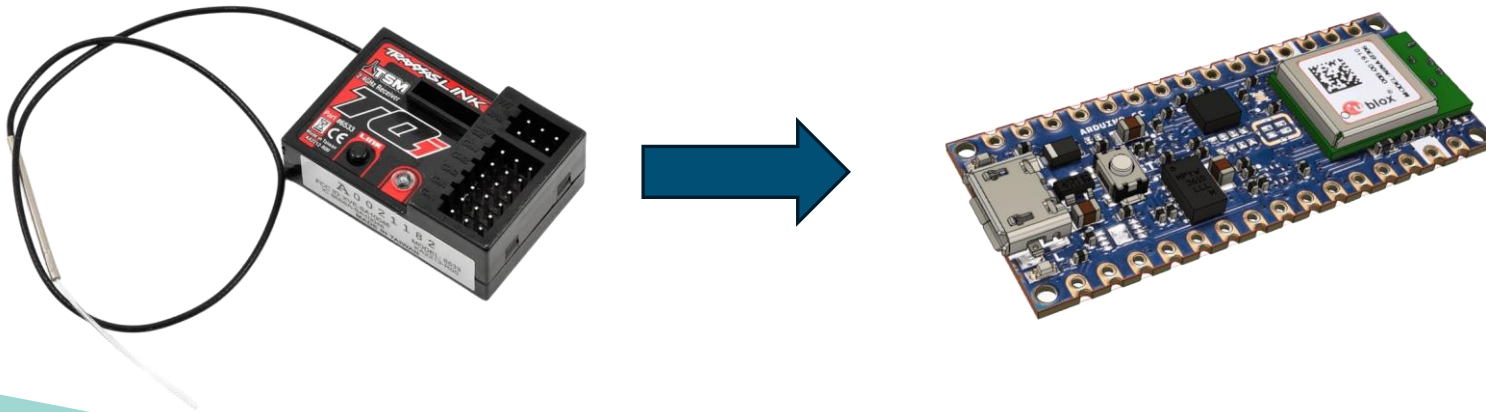
PWM (Pulse Width Modulation): è un segnale d'onda quadra con duty cycle variabile
Il ricevitore genera due segnali di questo tipo: uno per l'ESC ed uno per il servomotore



SOSTITUZIONE DEL RICEVITORE CON ARDUINO NANO

Il ricevitore originale della macchina è stato sostituito con un Arduino Nano 33 BLE Sense, sono stati quindi realizzati dei nuovi collegamenti hardware tra il microcontrollore, il servomotore e l'ESC

- Segnali
- Massa
- Alimentazione



CONTROLLO DELLA MACCHINA ATTRAVERSO ARDUINO

Libreria servo: questa libreria è in grado di comandare servomotore ed ESC generando un segnale PWM identico a quello generato originariamente dal ricevitore

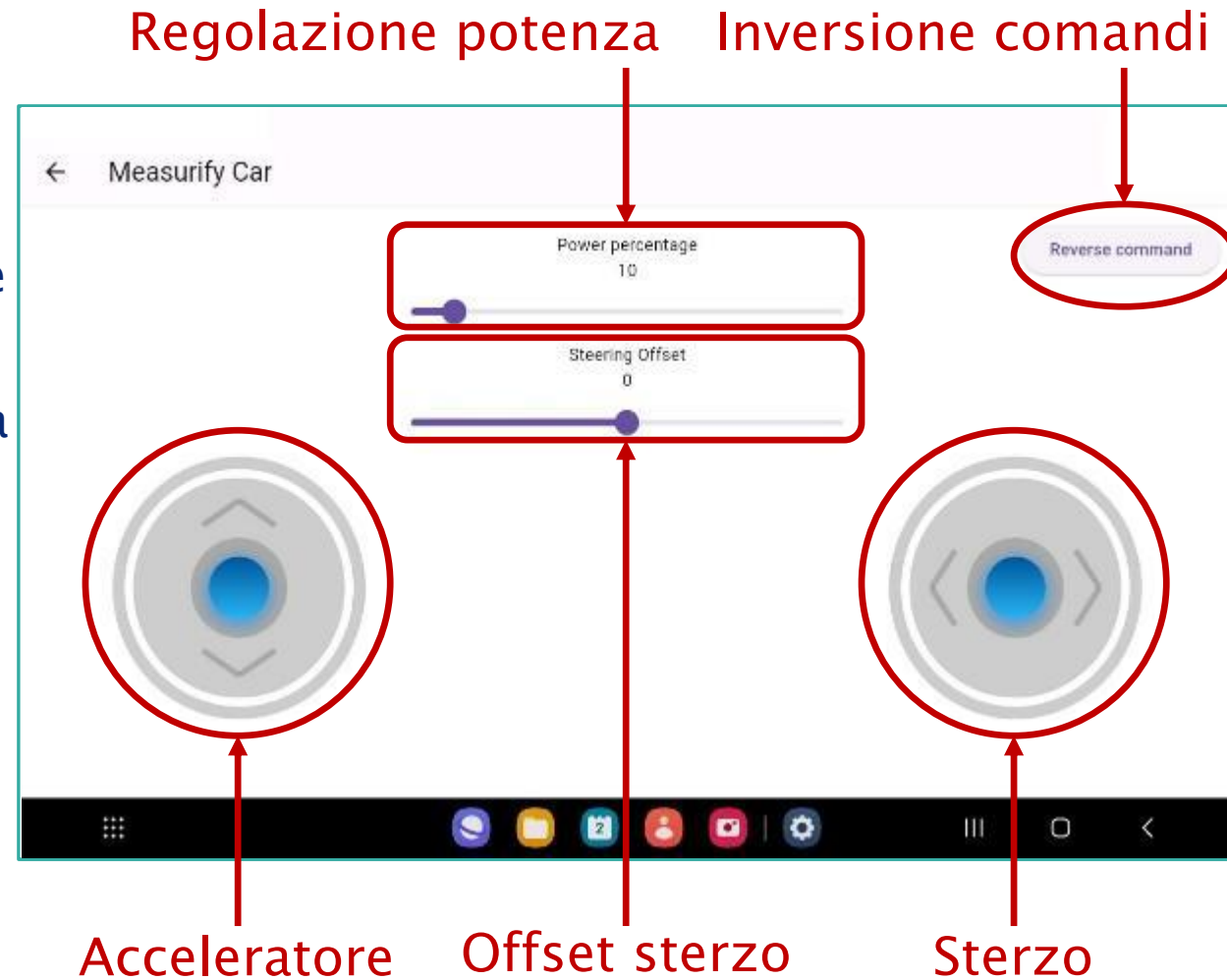
La libreria utilizza i timer interni di Arduino per generare un segnale preciso ma ciò è possibile solo su alcuni pin specifici di Arduino che supportano la generazione di questo tipo di segnale

```
//Write received command on servo
void onMovingCharacteristicWrite(BLEDevice central, BLECharacteristic characteristic){
    //steering
    ster.write(movingCharacteristic[0]);
    //throttle
    thr.write(movingCharacteristic[1]);
    //setting last time received command and reset error
    error=0;
    idle_previous=millis();
}
```

APPLICAZIONE

Nell'applicazione possiamo utilizzare 5 comandi per controllare la macchina:

- **Acceleratore:** muove il veicolo avanti e indietro
- **Sterzo:** muove il veicolo a destra e a sinistra
- **Inversione dei comandi:** sposta il pad dell'acceleratore al posto del pad dello sterzo e viceversa
- **Regolazione potenza:** regola la potenza massima erogata dalla macchina
- **Offset sterzo:** regola la posizione neutra dello sterzo



STRUTTURA DELLA PAGINA

In questa parte di codice viene riportata una parte principale del codice di comando della macchina.

In particolare questo è il codice relativo al posizionamento ed al movimento dei due joystick che comandano sterzo e acceleratore.

Il loro movimento modifica due variabili che verranno memorizzate ed inviate tramite BLE.

```
//joystick steering
alignment: Alignment(stateCommand * (0.8), 0.5),
child: Joystick(
  mode: JoystickMode.horizontal,
  listener: (details) {
    setState(() {
      steering = idleValueSteering +
        ((details.x * rangeValueSteering).round());
    });
  },
),
Align(
//joystick throttle
alignment: Alignment(stateCommand * (-0.8), 0.5),
child: Joystick(
  mode: JoystickMode.vertical,
  listener: (details) {
    setState(() {
      throttle = idleValueThrottle -
        ((details.y * rangeValueThrottle * powerSliderValue).round());
    });
  },
),
),
```


CODICE DI SICUREZZA

Infine è stato implementato un codice di sicurezza che agisce in caso di disconnessione o problemi di comunicazione tra i due dispositivi: se l'Arduino a bordo della macchina rileva errori durante la comunicazione, con il dispositivo di comando, blocca i motori impedendo l'avanzamento, mettendosi quindi in condizioni di sicurezza



VIDEO DIMOSTRATIVO



CONSIDERAZIONI FINALI

- Il mio contributo personale è stato quello di partecipare al reverse engineering, sviluppare l'applicazione, sviluppare il sistema embedded installato sulla macchina e la gestione della comunicazione tra i due dispositivi
- Una possibile miglioria del progetto potrebbe essere relativa alla gestione della sicurezza obbligando gli utenti ad autenticarsi, aggiungendo all'app un'interfaccia di autenticazione per evitare la connessione di utenti non in possesso dei permessi necessari
- Questo progetto è da considerarsi come punto di partenza per applicazioni future: in particolare potrebbe essere utile per implementare sistemi di guida autonoma

Grazie per l'attenzione

