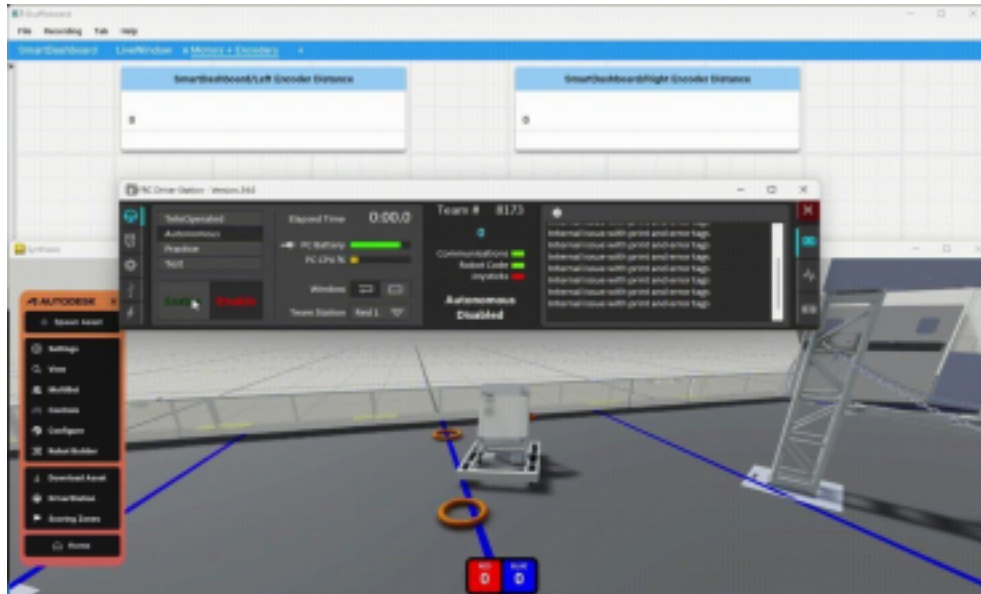


Esecuzione del codice FRC su un robot 3D virtuale con Autodesk Synthesis



I tuoi ingegneri stanno ancora lavorando su prototipi o aggiungendo componenti? Il tuo robot è chiuso in una stanza ma vuoi programmarlo a casa? Questa situazione è rilevante per noi del Team Gear Zero, quindi qui esamineremo un possibile modo per superare questi problemi.

Metodo di simulazione predefinito nella libreria WPILib Utile ma raddoppia il carico di lavoro rispetto all'esecuzione del codice sul robot: quando si fa clic ["Zamistlitaw ha detto: Disegna RoboT" DrawT WPILibVisual Studio](#), ciò consente di eseguire il codice utilizzando un comando [Pannello di controllo intelligente VerichauxM Shuffleboard](#), in modalità controllata dal conducente o autonoma. Tuttavia, quando inizi a provare il gruppo propulsore in questa simulazione, [questo ZoriT Lesznotaw ha detto: Ataw ha detto: CrtaT tkou Hruboo ha detto:T BopN Osate](#) .

Se il tuo team fosse simile al nostro, ciò aggiungerebbe complessità che non potrai completare nella frenetica raffica di programmazione che inevitabilmente precede la competizione e che il tuo progetto è molto più difficile da eseguire il debug. Che senso ha una simulazione se riesci a trovare un vero robot più velocemente?

È qui che Autodesk Synthesis torna utile. Si integra direttamente con la

funzione "Simulazione del codice robot", ma ti consente di vedere la posizione del robot e ispezionare parti del codice, il tutto comodamente da un computer! In più funziona in 3D e, se puoi sbrigarti, ti permette di creare e testare il codice su un modello CAD del robot.

Ciò presenta alcune sfide e limitazioni, poiché attualmente supporta solo motori ed encoder, non altri sensori, e inoltre non è completamente facile da installare (da qui questo tutorial). Lo ha sviluppato Stagisti, non Autodesk stessa, quindi non puoi aspettarti professionalità, ma è sicuramente meglio che non poter testare senza un robot.

Quando dovrebbe essere usato?

Autodesk Synthesis è molto utile per alcune cose e inutile per altre. In generale, è utile per imparare la programmazione FRC, per principianti e guidatori esperti, ma non per programmare sistemi di controllo avanzati. Ecco un riepilogo di base delle sue capacità a marzo 2024:

La sintesi può...	La sintesi non può...
-------------------	-----------------------

Verifica del codice autonomo dell'utente e del codice gestito dal conducente MOTORAM, in particolare i propulsori, nell'encoderT MOTORAM. Questo lo rende adatto ai principianti e ai team di apprendimento. Se si utilizza CAN invece di driver motore PWM o encoder non in quadratura, vedere la sezione "Utilizzo dei driver del motore" PUÒ, tj.	Controllare i componenti che non lo sono motori e codici, ad esempio sensori o giroscopi.
Gli autisti lo utilizzeranno per esercitarsi nella guida controllata dal conducente.	Funzionano bene in modalità autonoma avanzata perché Questa condizione di solito richiede giroscopi e/o sensori.
Servi come nuovo sviluppatore per imparare a programmare in FRC, senza robot.	Simula l'intera routine autonoma per i team esperti che cercano di acquisire controllare.

Come sempre, dovresti sempre utilizzare un vero bot di prova prima di decidere che il tuo codice è completamente operativo.

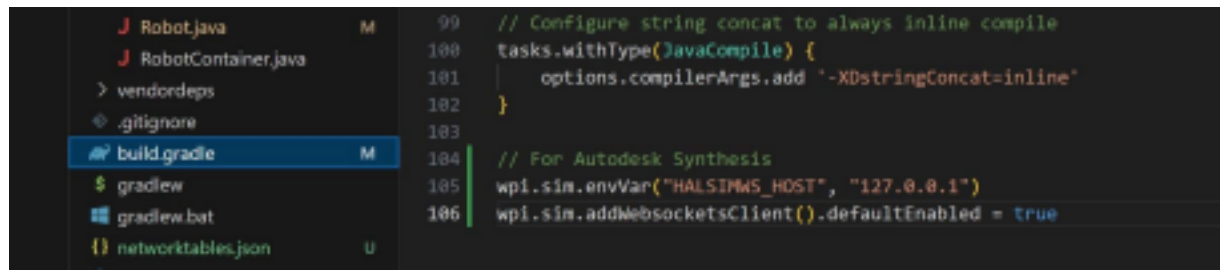
Installazione i configurazione

- Innanzitutto è necessario [SayR Okolotaw ha detto: Il contenuto ha detto: WPILib Java](#). Se vuoi scrivere e/o simulare codice. Esistono altri linguaggi di programmazione, ma Java è ampiamente parlato ed è l'unico linguaggio supportato da Sinteza.
- Quindi installa [Synthesis Autodesk Abbo je rekao:R Platformaovaj saT KN](#) .
- UWPIlib VS Code inserisce le seguenti 2 righe nella parte inferiore del file

```

        build.gradle: wpi.sim.envVar("HALSIMWS_HOST",
                                "127.0.0.1") o =
wpi.sim.addWebsocketsClient().defaultEnabled o
                                prav
                                i

```



Utilizzando Controller o encoder CAN per motori nekvadratni

Se si utilizzano già controllori motore collegati tramite [PWM \(Ufficio polacco del lavoro\)](#), non è necessario modificare il codice e puoi ignorare la sezione successiva.

Un aspetto irritante di Synthesis è che non supporta i driver dei motori CAN e lo consente [solo encoder M-quadrate](#), che hanno 2 numeri di canale nel loro

builder, non l'altro tipo. Tuttavia, se utilizzi i driver dei motori CAN, devi aggiornare il codice per abilitare i driver dei motori PWM (Ufficio polacco del lavoro). Nella simulazione: è più semplice delle classi alternative, cinematica e distanza, separate, soprattutto per i principianti.

Per supportare il robot con altro hardware, è necessario aggiungere una piccola quantità di codice aggiuntivo:

CRS/PRINCIPALE/JAVA/FRC/ROBOT/[SimulazioneHardware.java](#))

```

src/main/java/frc/robot/SimulationHardware.java

package frc.robot;

import edu.wpi.first.wpilibj.motorcontrol.VictorSP;
import edu.wpi.first.wpilibj.Encoder;

/**
 * A file for compatibility with Autodesk Synthesis robot simulation, which ensures
 * PWM motors and Quadrature encoders, which link with Synthesis,
 * can be used as well as the physical encoders during simulation.
 */
class SimulationHardware {
    /**
     * -----
     * The following code makes this act as a singleton class:
     * https://www.geeksforgeeks.org/singleton-class-java/.
     * This means whenever you want to interact with SimulationHardware you can
     * just use SimulationHardware.getInstance().<motor/encoder name>,
     * which will create the instance the first time, and use the same instance
     * every next time. No initialisation is needed, so you don't
     * need much extra code outside this file!
     */
    /**
     * Singleton instance.
     */
    private static SimulationHardware _instance;
    /**
     * Get the singleton instance, and set it up if it hasn't been created yet.
     */
    public static SimulationHardware getInstance() {
        if(SimulationHardware._instance == null) {
            SimulationHardware._instance = new SimulationHardware();
        }
        return SimulationHardware._instance;
    }
    /**
     * Constructor for the SimulationHardware instance, which creates motors using
     * PWM and Quadrature encoders for use in Autodesk Synthesis simulation.
     *
     * This is a private constructor so cannot be accessed directly. Use
     * SimulationHardware.getInstance() to get the singleton instance instead.
     */
    private SimulationHardware() { // Private constructor
        this.leftDriveMotor = new VictorSP(0); // Port 0
        this.rightDriveMotor = new VictorSP(1); // Port 1
        this.leftDriveEncoder = new Encoder(0, 1); // Channels 0,1
        this.rightDriveEncoder = new Encoder(2, 3); // Channels 2,3
    }

    /**
     * -----
     * The motors and encoders below used in the simulation can be accessed at
     * SimulationHardware.getInstance().<motor/encoder name>, and will be
     * set up the first time they are called. You should surround any code using
     * these with if(!RobotBase.isReal()) { } so it won't be run on the
     * actual robot.
     *
     * If you want to add your own motors/encoders you should add them as fields
     * below and also set them up in the constructor above. Use VictorSP for
     * motor controllers and Encoder for encoders.
     */

    public VictorSP leftDriveMotor;
    public VictorSP rightDriveMotor;
    public Encoder leftDriveEncoder;
    public Encoder rightDriveEncoder;
}

```

Crea [Dijelitaw ha detto: Singalton](#) Uno nuovo chiamato Simulation Equipment,

che viene eseguito nel costruttore

Driver per motori PWM (Istituto Polacco del Lavoro Compatibile con encoder synth e quadratura e trasmette numeri pin diversi per ciascuno. Ovunque comunichi con i tuoi motori/programmatori CAN, aggiungi il seguente snippet di codice che utilizza il controller e l'encoder Hardware di simulazione invece dell'hardware del robot in simulazione. È così facile (a patto di utilizzare i metodi giusti per [BeckR Hamano ha detto:P Hamedu ha detto:Z questo A qual è il focus di T Imaginarno](#)(!

Ovunque comunichi motori e programmatori,

```
whenever you interact with the motors and encoders.

// ... This is part of a larger file, as example code of how to interact with
// SimulationHardware ...
if(!RobotBase.isReal()) {
    // Simulation only - Autodesk Synthesis Compatibility

    // Simulated PWM Motor Controllers
    SimulationHardware.getInstance().leftDriveMotor.set(1.0);
    SimulationHardware.getInstance().rightDriveMotor.set(1.0);

    // Simulated Encoders
    SmartDashboard.putNumber("Left Encoder Distance",
        SimulationHardware.getInstance().leftDriveEncoder.getDistance());
    SmartDashboard.putNumber("Right Encoder Distance",
        SimulationHardware.getInstance().rightDriveEncoder.getDistance());
} else {
    // Real robot - Real Hardware (Here, REV 'CANSparkMax's and their associated
    // 'AbsoluteEncoder's)

    // CAN Motor Controllers
    leftDriveMotor.set(1.0);
    rightDriveMotor.set(1.0);

    // Motor Controllers' Integrated Encoders
    SmartDashboard.putNumber("Left Encoder Distance",
        leftDriveEncoder.getPosition());
    SmartDashboard.putNumber("Right Encoder Distance",
        rightDriveEncoder.getPosition());
}
// .....
```

TODO: codificarlo; assicurati che sia spesso D

Diagnostica per immagini

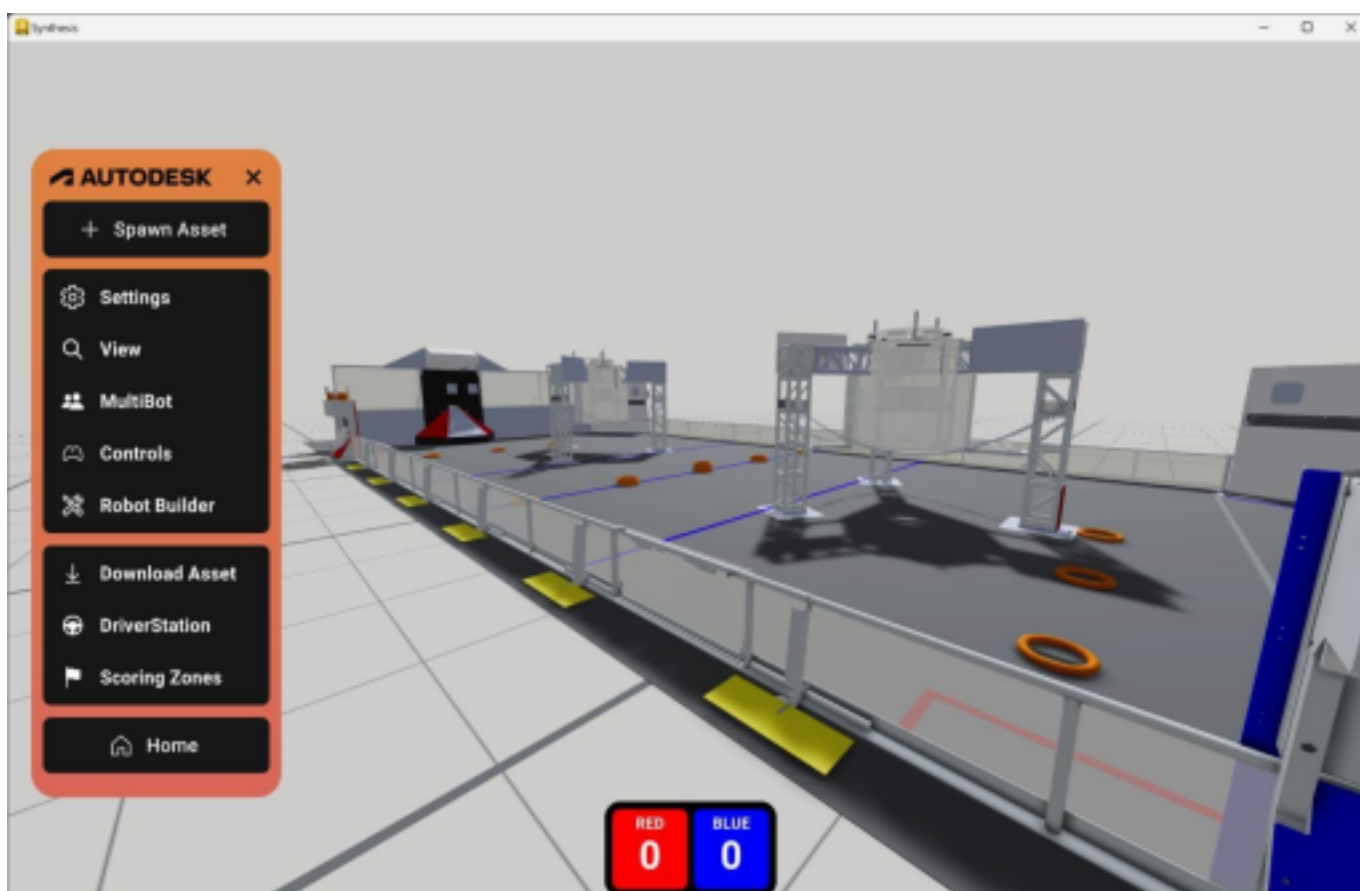
1. dio: sintetico

Inizio della sintesi

Per simulare il codice nell'applicazione Synthesis, aprire prima l'applicazione Synthesis. Quindi fare clic su:

- Simulatore di operazioni
- Un giocatore
- Metodo dell'esercizio

Seleziona la casella relativa all'attuale competizione FRC e quindi fai clic su "Carica". Inserisci una vista tridimensionale del campo:



Ora puoi fare clic e trascinare il mouse sulla cornice per ruotare il punto di vista ingrandendo o rimpicciolendo.

Aggiungi RoboT

Per aggiungere un crawler a un campo, fare clic su:

- Multibo T
- Aggiungi
- regolare

Ora seleziona il robot in base al design del tuo sistema di azionamento. Il tuo robot apparirà sul campo. Visualizza il cambio robot MultiBot, puoi selezionare un

robot e rimuoverlo, nonché sostituirlo aggiungendo un nuovo robot in qualsiasi momento. Per la maggior parte degli usi per principianti, non lo consigliamo, ma puoi farlo [בלייב Ataw ha detto: Maudu Hruboo ha detto:T CAD \(inglese\) saT Sa-Fusion 360 I use Ui KN BamakoM](#).

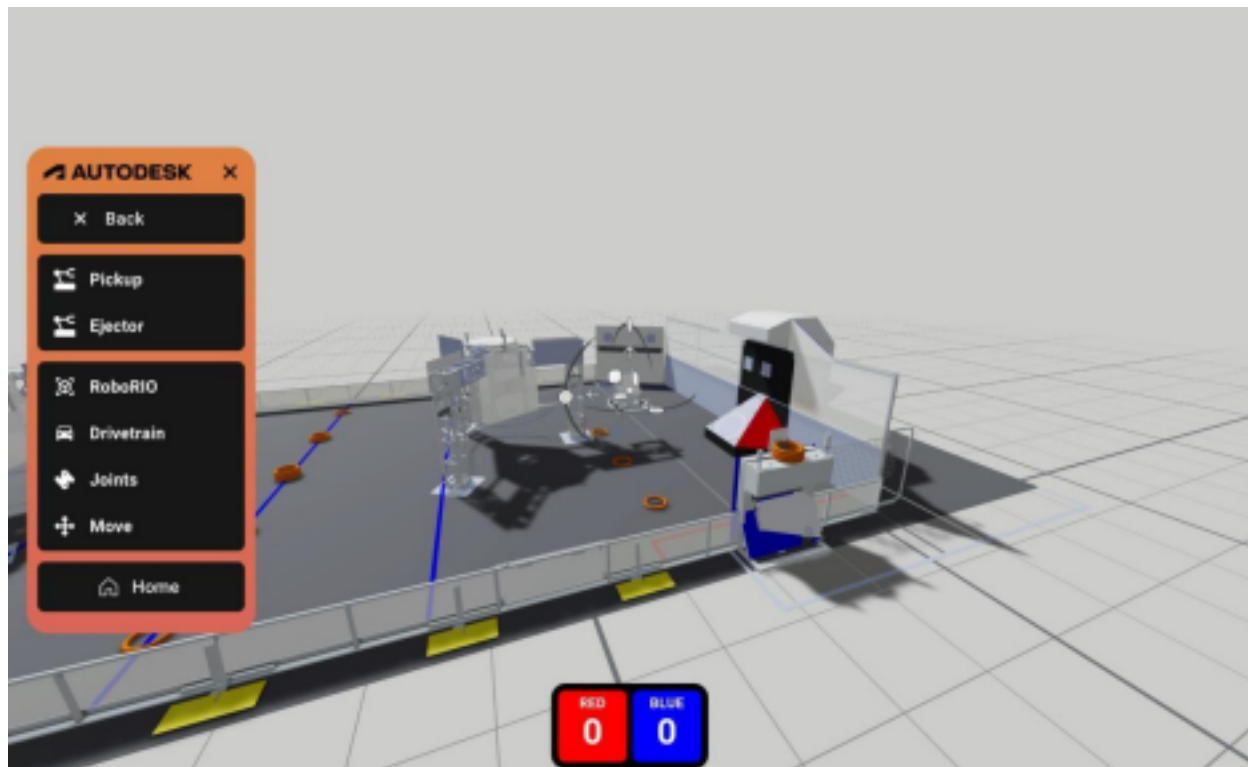
Configurazione del robot

Posizione del robot

Fai clic su "Configura" nel menu Autodesk, dove troverai gli strumenti per modificare il robot e la sua posizione. Finché non fai clic su "Indietro", la gravità non funzionerà e il tuo robot non risponderà al codice.

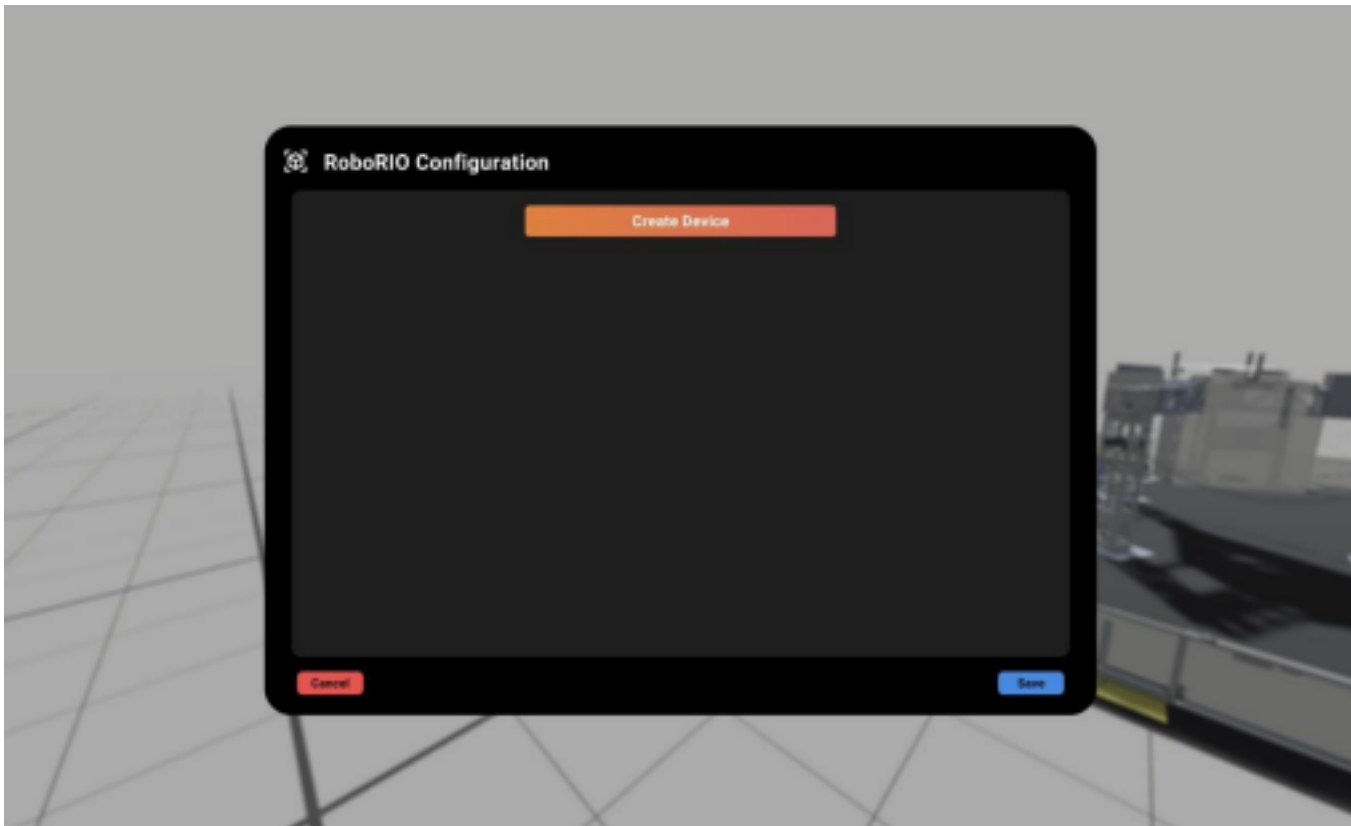
Quando fai clic indietro, la fisica e il codice verranno simulati.

Innanzitutto, utilizza lo strumento di spostamento per fare clic e trascinare le frecce e spostare il robot in un punto sopra il pavimento nella posizione iniziale, come mostrato di seguito.



Motori e encoderi

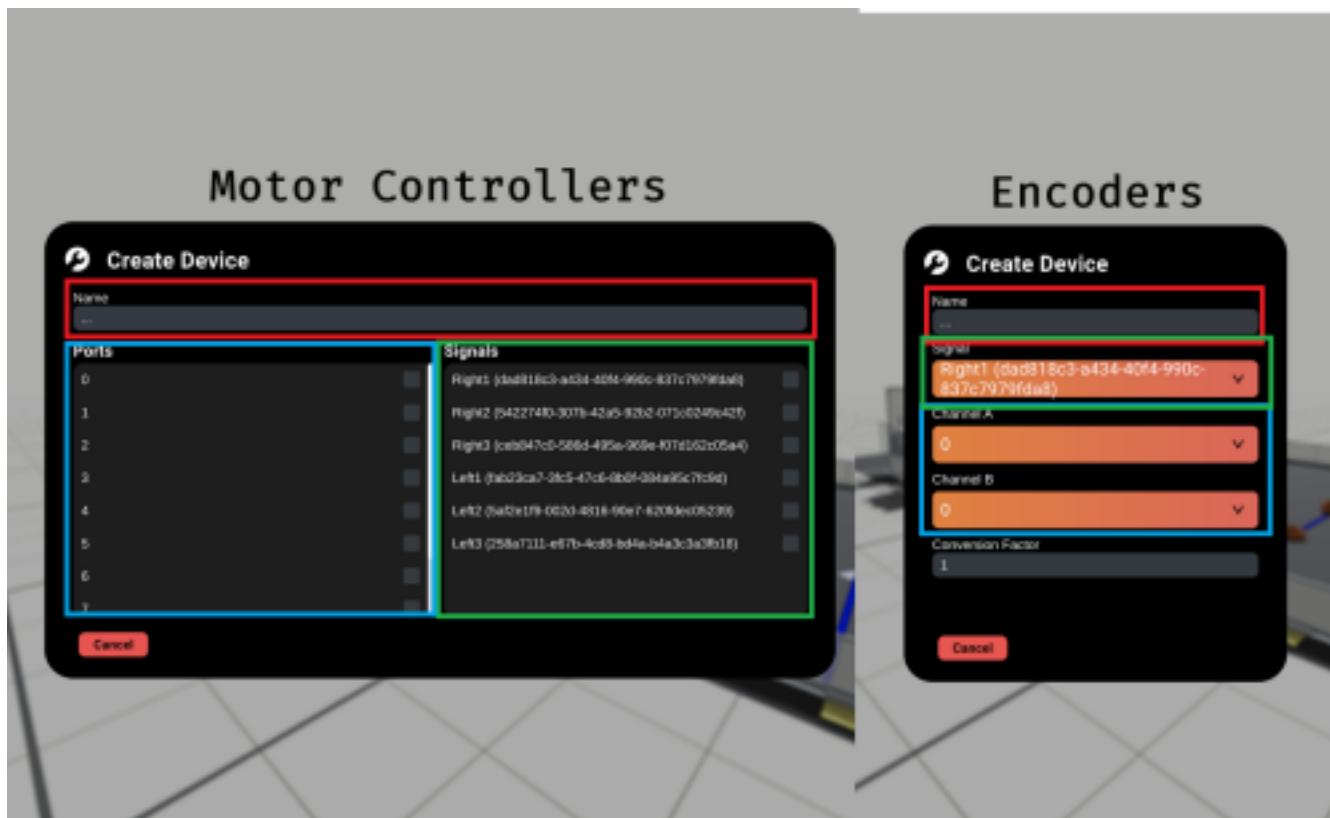
Quindi fare clic su "RoboRIO" che consente di configurare le uscite per i motori e gli encoder e il modo in cui si collegheranno al modello di robot, come in [roboRIO PizT](#). Dovresti vedere le seguenti informazioni:



Per ogni controller motore o encoder che stai utilizzando, fai clic su "Crea dispositivo" e seleziona "PWM" per il controller motore e "Encoder" per l'encoder. Ora trova il numero pin PWM del driver del motore/numero del canale dell'encoder, all'interno del costruttore nel tuo codice.

```
// PWM Port Number / Encoder Channel Numbers in the brackets after the new ... constructor  
  
leftDrive = new PWMSparkMax(channel:0);  
rightDrive = new PWMSparkMax(channel:1);  
  
leftEncoder = new Encoder(channelA:0, channelB:1);  
leftEncoder = new Encoder(channelA:2, channelB:3);
```

Nella schermata successiva, scegli un nome descrittivo per la mappatura del controller motore/encoder (riquadro rosso sotto), aggiungi le porte o i canali specificati nel codice (prima Ais nel costruttore, poi B; riquadro blu sotto), e scegli quale "segnali" nel robot virtuale che desiderano connettere (riquadro verde sotto). Ad esempio, se il tuo sistema di azionamento ha un motore sul lato destro del robot, puoi collegarlo ai segnali Right1, Right2 e Right3 nel KitBot. Quindi salva la mappatura e salva tutta la configurazione RoboRIO dopo il completamento.

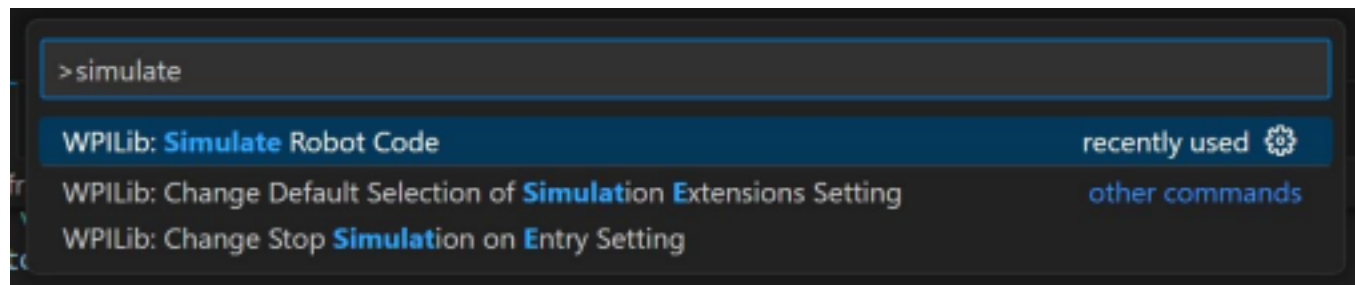


La prossima volta che aprirai Synthesis, dovrai aggiungere il robot al campo e spostarlo nuovamente, ma la configurazione di RoboRIO dovrà essere salvata.

Parte 2: il tuo codice

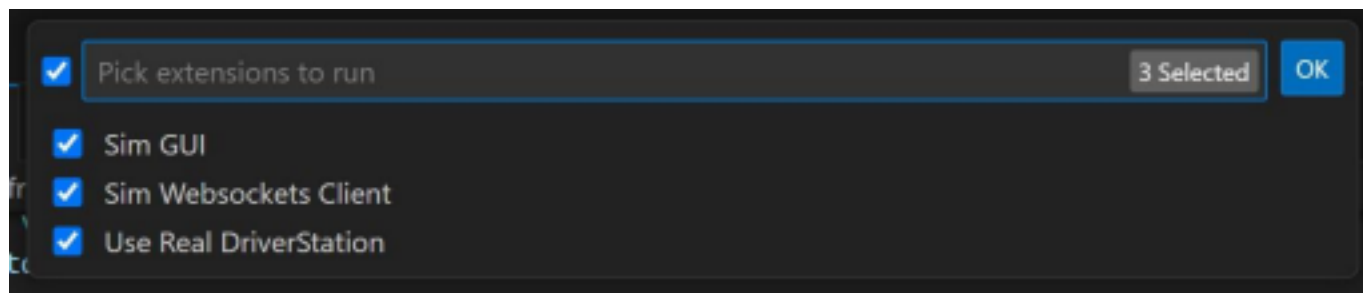
Esecuzione di "simulazioni di codici robot"

Ora che hai Syntezus aperto con un robot virtuale, puoi simulare il tuo codice dal codice WPILib VS. Una volta scritto il codice, utilizza Ctrl+Shift+P sulla tastiera per aprire il pannello di comando, cerca e quindi fai clic su "WPILib: Simulate Robot Code".



Ora attendi la compilazione del codice e apparirà questa finestra di dialogo. Assicurati che tutte e tre le estensioni siano selezionate facendo clic sulle caselle, quindi fai clic su "OK". Fai attenzione a non fare clic all'esterno di questa finestra di dialogo poiché ciò annullerà la simulazione e dovrai ripetere il

passaggio precedente.



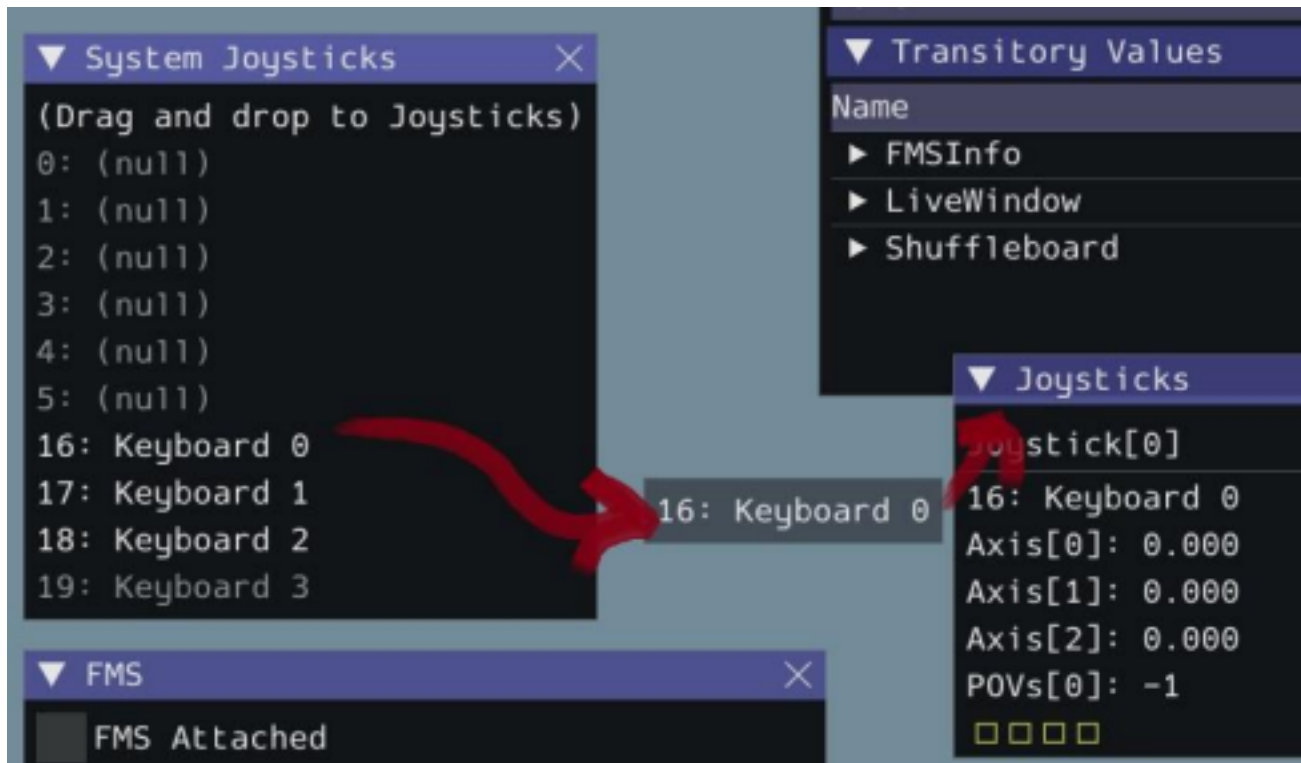
Se la sezione "Sim Websockets Client" non viene visualizzata, devi assicurarti di aver aggiunto le nuove righe sopra menzionate e di aver salvato il file `build.gradle`.

Joystick M

Questa sezione si applica solo al codice gestito dal conducente. Altrimenti vai alla sezione "Continua come al solito".

Dopo che il codice è stato compilato, dovrebbe aprirsi la finestra "Simulazione Robot". Dovrai collegare il joystick tramite USB o utilizzare semplicemente un mg [WistickT il dattilografo ha detto: Virtuale \(aggiunto immediatamente KN\)](#), trascinando il joystick appropriato dalla sezione "joystick di sistema" alla sezione "joystick", solitamente sul joystick]0[Se ne hai solo uno.

Per tua informazione, joystick della tastiera 0 usato [WASD \(semina a secco\)](#) per 2 Le sue prime contrazioni (solo joystick) e in I/A Terza.

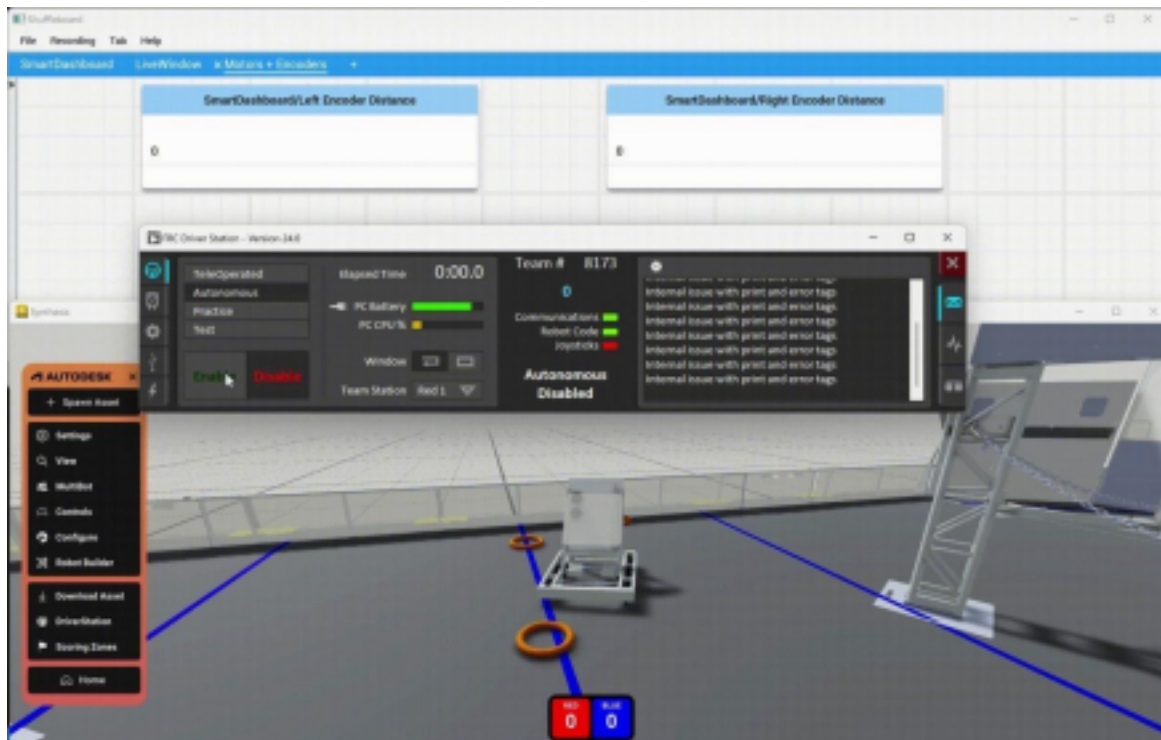


Vai avanti come Reggie L

Anche questa finestra "simulazione robot". [Mapesz \(Mapesh\)R uT VlastituT AspekataM ProgressiveM rabbinM tkou Simulacijaovaj](#), ma di solito non è necessario altro che impostare il joystick.

Aprire [Apptaw ha detto: Posizione del conducente](#) Puoi continuare a eseguire il codice come al solito, ma questa volta vedrai i risultati nel sintetizzatore e non nel lavoro reale. Puoi anche aprire [Njegar KlizaČM saT, ChegoN Shuffleboard](#) Se lo usi ed è utile per il debug dei log.

Se il tuo codice genera un errore durante l'esecuzione e la postazione di guida visualizza il messaggio "nessuna comunicazione con il robot", è fastidioso, ma non preoccuparti: è lo scopo della simulazione! È possibile visualizzare il registro di debug nella console del codice VPILib e tornare indietro dopo aver eseguito la "Simulazione del codice robot" dopo aver corretto e salvato le modifiche.



Questo robot non è molto intelligente. Non fa nulla con le chiamate dell'encoder
tranne visualizzarle.

Il tuo potrebbe essere migliore!

Divertiti con la programmazione FRC! WPILib può essere un mondo complesso,
ma se hai difficoltà, basta chiedere online, ad esempio sul principale forum
Delphi specifico per FRC: la maggior parte dei team (inclusi noi) sono disposti a
condividere le proprie conoscenze e il proprio supporto!