

# Light Rail Transit Installation Manual

## **Introduction:**

Welcome to the installation manual for the train system Python script, which is composed of five software sub-modules. This manual will guide you through the process of installing the necessary software and dependencies, configuring the user interface, and running the script on a machine that supports Windows 10 with an IDE that supports Python.

The train system Python script uses Application Programming Interfaces (APIs) to communicate with the other sub-modules in the train system, denoted by various “Signals” classes. It uses information sent through the APIs to simulate transit from dispatching a train from CTC, arriving to its set destination, then eventually returning to the yard. The system consists of five main module classes: the CTC class, for dispatching the train, the track controller class for controlling the switch of the track, the track model class for modeling the physical track layout, the train model class for modeling train movement along changes in terrain of the track, and the train controller class for controlling power of the train and regular train operations. All sub-module classes can be viewed through displays built from Python graphical user interfaces (GUI). All sub-module UI's are controlled by a “launcher” class which provides buttons to select which module's UI to display.

By following the steps in this manual, you will be able to install the necessary software and dependencies, configure the user interface, and run the script to run your own train system.

## **1. Install Python:**

Download the latest version of Python for Windows from the official website:

<https://www.python.org/downloads/windows/>

Run the installer and follow the installation wizard to complete the installation process.

Make sure to add Python to your PATH environment variable during the installation process.

## **2. Install an IDE that supports Python:**

Download and install an IDE that supports Python, such as PyCharm, Visual Studio Code, or Spyder.

Follow the installation wizard to complete the installation process.

## **3. Clone or download the larger train system repository:**

Navigate to the repository's website or GitHub page and clone or download the repository to your local machine.

## **4. Install additional Python packages:**

In the IDE's terminal or command prompt, navigate to the directory where you downloaded the larger train system repository.

Run the following command to install PyQt5 and Qt Designer:

```
pip install PyQt5 pyqt5-tools
```

#### **5. Open the launch.py Python script:**

Navigate to the directory containing the launch.py Python script within the larger train system repository. This file should be found after expanding the “train system” folder.

Open the launch.py Python script using the IDE you installed in step 2.

#### **6. Install the required dependencies:**

If an error occurs while trying to run the launch.py file, follow the steps below:

In the IDE's terminal or command prompt, navigate to the directory where you opened the launch.py Python script.

Run the following command to install the required dependencies:

```
pip install pyqt5
```

```
pip install numpy
```

#### **7. Run the train simulation script:**

Launch.py opens a Python GUI containing all the sub-modules' user interfaces. Once a train is dispatched by the CTC sub-module, select the following sub-module UI's as follows:

1. Click the “Track Model” button. Upload the line's .csv file. You may select the “track” tab if you wish to watch the train move along the track.
2. Click the “Track Controller” button. Upload the PLC script for the corresponding line you selected in the track model window.
3. Click the “CTC” button. Select the manual tab, select the corresponding line from the dropdown menu. Then select the stations using the other dropdown menu next to the “Dispatch” button. You can increase the simulation speed using the spin box located at the bottom of the window. Click the “Dispatch” button to send a train to move along to track to reach the destination selected.
4. Click the “Train Model” button. If there are multiple trains, there will be a dropdown menu present under the “Train Model” button which consists of all trains dispatched from CTC and present on the track. To change the internal train temperature, use the spin box in the “Temperature” section.
5. Click the “Train Controller” button. If there are multiple trains, there will be a dropdown menu present under the “Train Controller” button which consists of all trains dispatched from CTC and present on the track. All train controls are present in this window.

## 8. Install the Track Controller Hardware Module

### Software setup:

- 1) For the hardware, follow the instructions above but instead of running the launch.py file, run the TestHWlaunch.py file in the Track\_Control\_Hardware folder. This will run the same launcher as before except it will use the hardware track controller instead of the software. This program will not run without the hardware set up so to ensure the hardware is set up right, see the hardware setup below.

### Hardware setup:

Components needed to run this system are:

- Arduino Mega
- Joystick
- LCD
- Wires
- USB connector for Arduino

1) Pin Connections:

- a) Here is a list of pins for each component and where they connect.
- b) Arduino:

**NOTE:** (Power and ground need an intermediate connection, such as a breadboard, since it goes to two different components)

Arduino	Connected to
SCL	LCD (SCL)
SDA	LCD (SDA)
Digital 2	Joystick (SW)
A0	Joystick (VRx)
A1	Joystick (VRy)
Power (5V+)	Joystick (5V), LCD (5V)
Ground (GND)	Joystick (GND), LCD(GND)

c) Joystick:

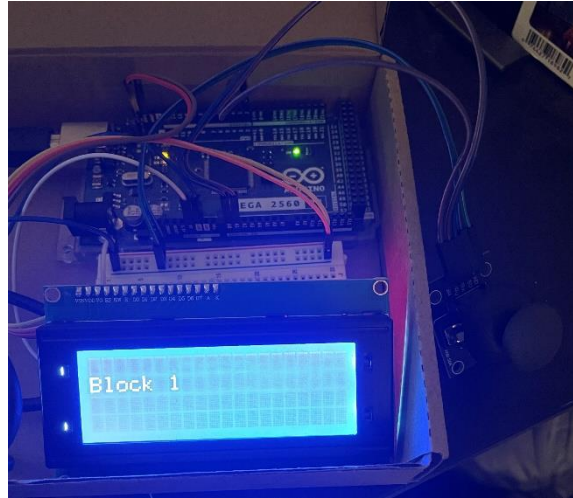
Joystick pin	Connected to
SW	Arduino (Digital 2)
VRx	Arduino (A0)
VRy	Arduino (A1)
Power (5V+)	Arduino (5V)
Ground (GND)	Arduino (GND)

d) LCD:

LCD pin	Connected to
---------	--------------

SCL	Arduino (SCL)
SDA	Arduino (SDA)
Power (5V+)	Arduino (5V)
Ground (GND)	Arduino (GND)

e) The final system should look like this:



- 2) Once this is all set up, connect the Arduino to the computer with the USB connector.
- 3) Open Arduino IDE and open the "Track\_control\_2.0" program downloaded from GitHub. Ensure the correct board is selected and the correct COM port number is selected. All necessary libraries for Arduino are included in the downloaded repository. Once everything is connected and set up properly, load the program onto the board.

**NOTE:** The COM port for the computer this was made on was 3, if your connection is different the HWProgrammerUI.py may need to be changed to reflect that. On line 9 of HWProgrammerUI.py it says `"port = serial.Serial('COM3', 115200)"`. On this line where it says 'COM3', you may need to change the 3 to the COM port you are connected to if your port connection is different.

- 4) Once the program is loaded onto the board, you are ready to run your software. The Arduino will reset when you start the program giving you a clean slate with a fresh program.
9. The following steps should allow you to successfully install and run the train system Python script as well as all sub-modules within the train system on your Windows 10 machine using an IDE that supports Python.