Logbook:

Day 1: 2024-10-7

Downloaded Arduino IDE on desktop and libraries such as ESP32 board library, Legoino and NimBLE.

Day 2: 2024-10-24

Tested Arduino IDE download on a laptop but couldn’t do it because of some technical issues.

Day 3: 2024-10-29

In the program, in this line of code: myTrainHub.init(), you could add a specific address in between the parentheses. This will only allow the ESP32 to connect to the Lego train with the specific address that is in between the parentheses. Example of an address of a Lego train: 90:84:2b:03:19:7f.

The address of the green Lego train: 90:84:2b:cd:a5:89

The address of the yellow Lego train: 90:84:2b:1a:a0:cd

The block of code that has been changed. The green trains address is in parentheses. So the ESP32 only connects to the green train. Any code that starts with // is just a comment so the computer will skip it :

myTrainHub.init("90:84:2b:cd:a5:89"); // initalize the PoweredUpHub instance

//myTrainHub.init("90:84:2b:03:19:7f"); //example of initializing an hub with a specific address

//90:84:2b:cd:a5:89 Green train

//90:84:2b:1a:a0:cd Yellow train

Results: The ESP32 will only connect to the green train

Success

Day 4: 2024-11-8

The program can now handle two trains. I initialized both trains’ addresses so the computer will recognize them both.

I made variables so the computer won’t mix both trains up

------------------------------------------

Lpf2Hub greenTrainHub;

Lpf2Hub yellowTrainHub;

-------------------------------------------

-------------------------------------------

This part is to a initialize both addresses and try to connect them to the ESP32.

// Check and connect to the green train hub

if (!greenTrainHub.isConnected() && !greenTrainHub.isConnecting()) {

greenTrainHub.init("90:84:2b:cd:a5:89"); // Green train address

}

if (greenTrainHub.isConnecting()) {

greenTrainHub.connectHub();

if (greenTrainHub.isConnected()) {

Serial.println("Connected to Green Train HUB");

} else {

Serial.println("Failed to connect to Green Train HUB");

}

}

// Check and connect to the yellow train hub

if (!yellowTrainHub.isConnected() && !yellowTrainHub.isConnecting()) {

yellowTrainHub.init("90:84:2b:1a:a0:cd"); // Yellow train address

}

if (yellowTrainHub.isConnecting()) {

yellowTrainHub.connectHub();

if (yellowTrainHub.isConnected()) {

Serial.println("Connected to Yellow Train HUB");

} else {

Serial.println("Failed to connect to Yellow Train HUB");

}

}

-------------------------------------------------------------------------------------------------

Results: The trains have connected to each other, they could either go one by one or go at the same time.

Success.

Next Goal: See if I could add more modules to this project(Light sensors ultrasonic sensors etc…)

Day 5 2024-12-2

I decided to use one train to test if I could add a light sensor or photoresistor to stop the train. The code modified a lot. Here are parts of the code that have been modified:

#define LDRPIN 2 // Ensure this matches the analog pin connected to the LDR

const int threshold = 300; // Adjust this value based on testing

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

bool trainStopped = false; // Flag to track whether the train has stopped

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

pinMode(LDRPIN, INPUT); // Set the LDR pin as input

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

int val = analogRead(LDRPIN); // Read the LDR value

Serial.print("LDR Value: ");

Serial.println(val);

if (val < threshold && !trainStopped) {

// Train detected at the sensor

Serial.println("Train stopped! Waiting for 5 seconds...");

myTrainHub.stopBasicMotor(port); // Stop the train

trainStopped = true; // Set the flag to indicate the train has stopped

delay(5000); // Wait for 5 seconds

Serial.println("Train moving again.");

myTrainHub.setBasicMotorSpeed(port, 100); // Resume the train movement

myTrainHub.setBasicMotorSpeed(portB, 100); // Resume second motor (if applicable)

}

else if (val >= threshold) {

// If the train is away from the sensor, keep it moving

if (trainStopped) {

// Only resume movement if it was previously stopped

trainStopped = false;

Serial.println("Train running.");

myTrainHub.setBasicMotorSpeed(port, 100); // Keep the train running

myTrainHub.setBasicMotorSpeed(portB, 100); // If needed for both motors

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Results: The code works, the train stops when the train goes over the photoresistor but also after 5 seconds it starts moving again and that cycle continues!

Next Goal: Try to use two photoresistors or more to control to trains.

Day 6 - 2025-01-21

The photoresistor I used was acting weird. It sometimes stops the train, and sometimes it doesn't. So I decided to order a NEW photoresistor(DIYables Digital Light Sensor for Arduino, ESP32, ESP8266, Raspberry Pi, 4 Pieces) from Amazon. Which has a screwdriver to control the sensitivity. I

Day 7 - 2025-01-22

After some thinking I used the flashlight to get A BRIGHTER light on a steady basis. And it worked every time the train passed above the sensor, the value decreased and then the train stopped.

Day 8 - 2025-02-23

We have started experimenting with the ESP32, photoresistor, and also the train. But for some reason we couldn’t make it work again. It gave us random values, the crazy part is that even if you unplug the photoresistor, it still gives random values and not just 0. After some research, even if you unplug the photoresistor from the analog pin, it will detect ambient noises or background noises. Causing “Phantom Values”. We tried to turn off the light, turn on the light of the other lights, and even reversed the threshold to value (value > threshold to value < threshold).

Day 9 - 2025-3-08

We have it working again(Kinda)! We tried a new photoresistor, and these are the things that we observed. For some reason when there is light on the photoresistor, its value is 400, but when there is darkness, it becomes 500. When the train goes over the sensor now, it actually stops. But the values make small changes, which causes the train to stop and go stop and go on the sensor. But it does work.

Day 10 - 2025-3-15

The train stops at the sensor now and then it moves YIPEE. But it has a problem, while the train moves around the track the sensor has a chance of detecting falsely. So when the 3 second cooldown is on the sensor wont detect the train because it cant detect for 3 seconds. But it does still work sometimes. This will possibly be fixed.

Day 11 - 2025-3-17

Fixed the issue of train stopping in odd places during cool down time. I reduced the cooldownI reduced the cool down time to 1sec.

Day 12 2025-3-18: We have made a few programs of 2 train junction stopping but they either stop at random spots or don’t stop at all. The LDR values are now like in the thousands even using pin IO2 gives 1000 when it should give values of 400 - 500.

We have found out that one of the sensors were broken so we used another sensor and then it gave real values. The values were from 1800 - 4095 so our threshold became 3500. Then it actually worked the train junction crossing logic worked.