```
/*
 * A Legoino example to control two trains with motors connected
 * to Port A and Port B of each Hub
 *
Adapted the code from Cornelius Munz. Made changes for our trains and sensors.
 */

#define LDRGREEN 14
#define LDRYELLOW 4
#include "Lpf2Hub.h"

// Create two hub instances for two trains
Lpf2Hub greenTrainHub;
Lpf2Hub yellowTrainHub;

// Variables
boolean TrainPresent = false;
unsigned long previousMillis = 0;
int LDRGREENValue = 0;
int LDRYELLOWValue = 0;
int light_sensitivity = 3500;
//int light_sensitivity_yellow = 1300;
int interval = 3200; // 3 seconds for crossing

// Possible states
const byte TRACK_SECTION_CLEAR = 0;
const byte YELLOW_TRAIN_DETECTED = 1;
const byte GREEN_TRAIN_DETECTED = 2;
const byte YELLOW_TRAIN_WAITING = 3;
const byte GREEN_TRAIN_WAITING = 4;

byte state = TRACK_SECTION_CLEAR;

byte portA = (byte)PoweredUpHubPort::A;
byte portB = (byte)PoweredUpHubPort::B;

void setup() {
    Serial.begin(115200);
}

void loop() {
    // Ensure both trains are connected before proceeding
    if (!greenTrainHub.isConnected() && !greenTrainHub.isConnecting()) {
        greenTrainHub.init("90:84:2b:cd:a5:89");
    }
```

```
    if (greenTrainHub.isConnecting()) greenTrainHub.connectHub();

    if (!yellowTrainHub.isConnected() && !yellowTrainHub.isConnecting()) {
        yellowTrainHub.init("90:84:2b:1a:a0:cd");
    }
    if (yellowTrainHub.isConnecting()) yellowTrainHub.connectHub();

    // Proceed only if both hubs are connected
    if (greenTrainHub.isConnected() && yellowTrainHub.isConnected()) {
        LDRYELLOWValue = analogRead(LDRYELLOW);
        LDRGREENValue = analogRead(LDRGREEN);
        //yellowTrainHub.setBasicMotorSpeed(portA, 50);
        //greenTrainHub.setBasicMotorSpeed(portA, 40);
        Serial.println("Trains moving");
        Serial.print("LDR Yellow: "); Serial.print(LDRYELLOWValue);
        Serial.print("  LDR Green: "); Serial.println(LDRGREENValue);
        delay(50);

        switch (state) {
            case TRACK_SECTION_CLEAR:
                yellowTrainHub.setBasicMotorSpeed(portA, 60);
                greenTrainHub.setBasicMotorSpeed(portA, 60);
                greenTrainHub.setBasicMotorSpeed(portB, 100);
                Serial.println("Trains moving");

                if (LDRGREENValue > light_sensitivity) {
                    previousMillis = millis();
                    state = GREEN_TRAIN_DETECTED;
                    Serial.println("Green Train Detected");
                } else if (LDRYELLOWValue > light_sensitivity) {
                    previousMillis = millis();
                    state = YELLOW_TRAIN_DETECTED;
                    Serial.println("Yellow Train Detected");
                }
                break;

            case YELLOW_TRAIN_DETECTED:
                if (LDRGREENValue > light_sensitivity) {
                    state = GREEN_TRAIN_WAITING;
                    Serial.println("Green Train Detected");
                    greenTrainHub.setBasicMotorSpeed(portA, 0); // Green train
stops
                    greenTrainHub.setBasicMotorSpeed(portB, 100);
                    yellowTrainHub.setBasicMotorSpeed(portA, 60); // Yellow train
keeps moving
```

```
                Serial.println("Green Train has stopped");
            }
            if (millis() - previousMillis > interval) {
                state = TRACK_SECTION_CLEAR;
                Serial.println("Trains Resuming");
            }
            break;

        case GREEN_TRAIN_DETECTED:
            if (LDRYELLOWValue > light_sensitivity) {
                state = YELLOW_TRAIN_WAITING;
                Serial.println("Yellow Train Detected");
                yellowTrainHub.setBasicMotorSpeed(portA, 0);
                greenTrainHub.setBasicMotorSpeed(portA, 60);
                greenTrainHub.setBasicMotorSpeed(portB, 100);
                Serial.println("Yellow Train has stopped");
            }
            if (millis() - previousMillis > interval) {
                state = TRACK_SECTION_CLEAR;
                Serial.println("Trains Resuming");
            }
            break;

        case YELLOW_TRAIN_WAITING:
            if (millis() - previousMillis > interval) {
                yellowTrainHub.setBasicMotorSpeed(portA, 60);
                state = TRACK_SECTION_CLEAR;
                Serial.println("Yellow Train moving");
            }
            break;

        case GREEN_TRAIN_WAITING:
            if (millis() - previousMillis > interval) {
                greenTrainHub.setBasicMotorSpeed(portA, 60);
                greenTrainHub.setBasicMotorSpeed(portB, 100);
                state = TRACK_SECTION_CLEAR;
                Serial.println("Green Train moving");
            }
            break;
    }
} else {
    Serial.println("A train hub is disconnected");
}
}
```