



Simple Train Blocking System

A simple train blocking (and universal) system in Arduino for Märklin trains and DCC++ trains. More informations and code available here <https://gitea.com/EZTrain/STBS>

 Difficulté **Moyen**

 Durée **2 jour(s)**

 Catégories **Électronique**

 Coût **42 EUR (€)**

Sommaire

Introduction

Video d'introduction

Étape 1 - Cut the rails

Étape 2 - Create your units and upload the code

Étape 3 - Connect the tracks

Étape 4 - Connect the traffic lights and sensors

Étape 5 - Connect the relays

Étape 6 - Connect the Arduinos together with the COM port

Étape 7 - Power up everything

Étape 8 - Possible troubleshooting

Commentaires

Introduction

The simple train blocking system. An easy to install train blocking system using Arduino. Compatible for MFX and DCC++ decoders.

This project comes from another project called ATBS (Arduino Train Blocking System) the problem with this was that it was barely reproducible. I mean that we made 3 zones (3 blocks) and each time we needed to add one zone or one block we needed to change the code and all the hardware and stuff.

Also, there was so much wires that it was impossible to troubleshoot.

This project has the same goal as ATBS: prevent accidents using a train blocking system (as in IRL) for Märklin trains as well as DCC++ decoders. But STBS is also very easy and fast to build.

The principle is that there is one STBS unit by block, and each unit is connected with the next one and the precedent one.

You can find all the information about the project here: <https://gitea.com/EZTrain/STBS>

WARNING: This project is still in beta phase

Matériaux

- Relays (3 per block)
- Rails and isolators (optional, you can basically cut the rails, but it's quite brutal)
- Arduino boards, one per block (recommended: Nano)
- LEDs and IR sensors (can be replaced by other sensors)
- The software 'Arduino IDE' to upload the code to the Arduino board
- One bridge rectifier for the whole network
- A AC power supply, doesn't have to be Märklin or anything, just used to be converted into DC by the bridge rectifier
- Your usual power supply, used to control train (ex: Märklin station)
- 5V power supply for Arduino boards

Outils

- Computer
- Rails
- Voltmeter
- Basic electronic stuff

Étape 1 - Cut the rails

You'll need to cut the rails or isolate them to make sure each block is independent.

The last track is the security track and is isolated from the rest.

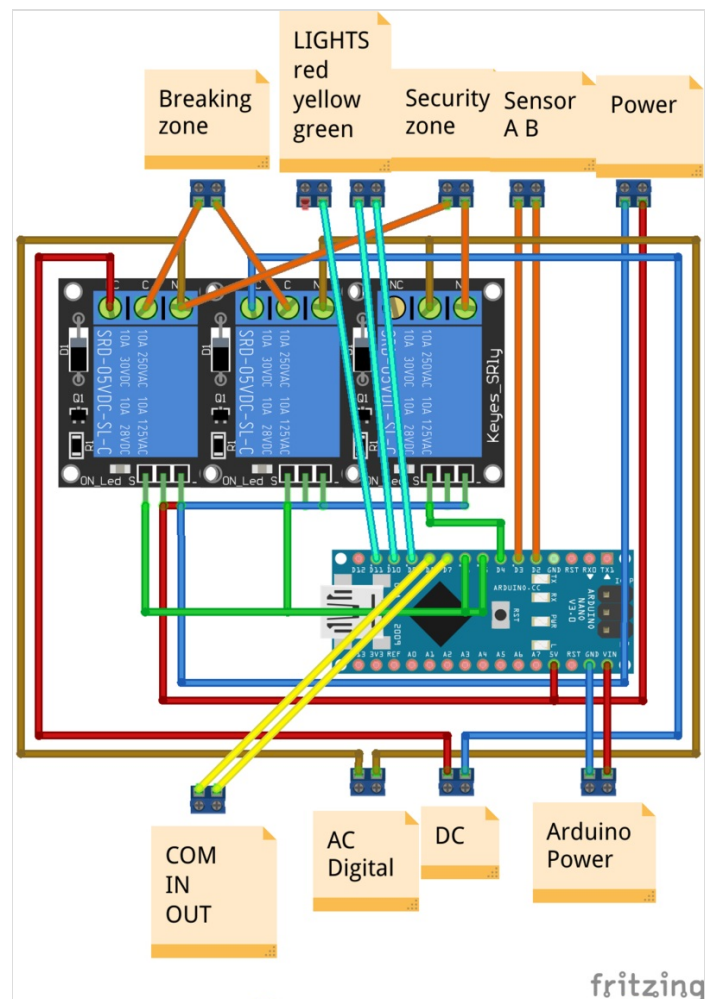
The sticker on the plaque is right on the security track. What you see before is the main track (stop zone) and what you see after is the stop zone of the next block.

Étape 2 - Create your units and upload the code

You'll first need to reproduce this schema as shown above. Excepted that here, we made 2 blocks next to another, that's the reason why there are 2 Arduino boards. The little wire connectors shown on the schema are here the black wire connectors.

The schema will be the same for all the blocks.

You can now copy and paste the code into the Arduino IDE. Inside the software you can select the port where is connected your Arduino (USB) and the type of board.



Étape 3 - Connect the tracks

You can connect the tracks with the 'Breaking zone or stop zone' and another one with the 'security zone' (security track)

Étape 4 - Connect the traffic lights and sensors

Traffic lights (top view) with the B sensor on the right

The lights can be prebuilt or made like this on a breadboard (prototyping board only) you'll just need one resistor, some wires and 3 leds. The resistor is connected to the GND of the Arduino ('Power' on the schema).

You can see on the right of the LEDs a IR sensor, this is the 'OUT SENSOR' (or B sensor on the schema) this is detecting when a train goes out the zone.

The pins are, from the left to the right: OUT, GND, VCC. Basically you just need to connect the OUT to the B or A Sensor like on the schema; and the GND and VCC to the 'Power' of the Arduino.

Étape 5 - Connect the relays

On this global view of one unit you can see the relays (blue boxes) like on your schema. But you'll also need to connect the GND and VCC port of the relays to the 'Power' of the Arduino (like on the schema).

Étape 6 - Connect the Arduinos together with the COM port

To connect to the Arduino, you have the COM IN and COM OUT on the schema. The COM OUT port of your current block goes to the COM IN of the precedent block. And the COM IN port of your current block goes to the COM OUT of the next block.

Étape 7 - Power up everything

You can now connect the Arduinos to the 5V power supply using some wires. The Nano boards don't need what you seen on the pictures because they just need 2 wires (VCC and GND).

If the power supply is not enough to power the rest of the system, you can try to modify the straight of the signal to 5.5V for example. Or more if necessary.

The Märklin station is used to power the 'AC digital' ports on the schema. (black box)

The white box is a AC generator used to be converted to DC using the bridge rectifier (used to go into the DC port on the schema)

Étape 8 - Possible troubleshooting

When you power up everything, the lights should be off. If one of them is red, check the sensors, they don't like the light of sun. If one of them is green (but no red), then check the communication with the next unit.

If you have some issues with the uploading, try to change the bootloader settings.

If the relays are inverted, try to change the code by changing all the lines like this one:

```
digitalWrite(relay, HIGH);
```

```
// or
```

```
digitalWrite(relay, LOW);
```

by inverting HIGH and LOW. You need to do that because it depends on the relay's side. Totally depending on the brand who make them.

Hope you enjoyed that guide and I hope this helped you! If you have any question you can create an issue 🐱
