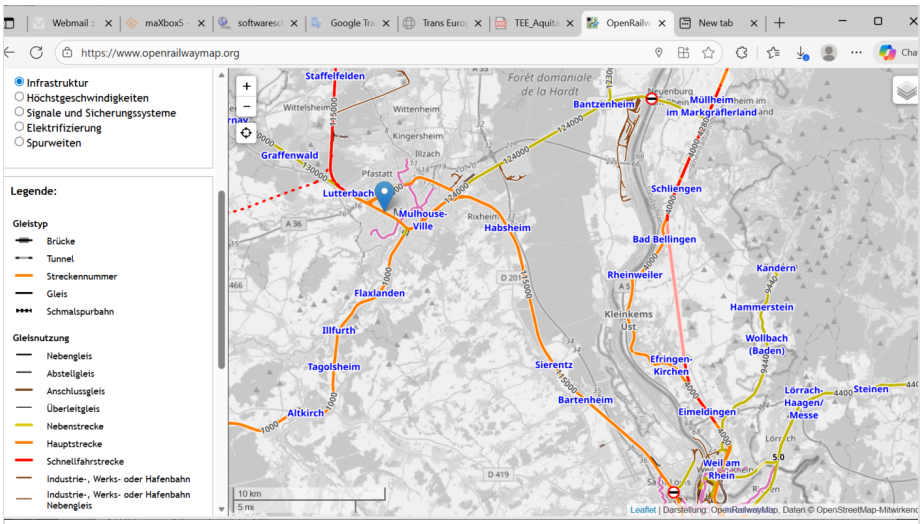


Railuino II

March 2, 2026

We then switch from Railuino after short updates in Arduino to Locoduino in the second part!

LOCODUINO – Accueil

1. Railuino

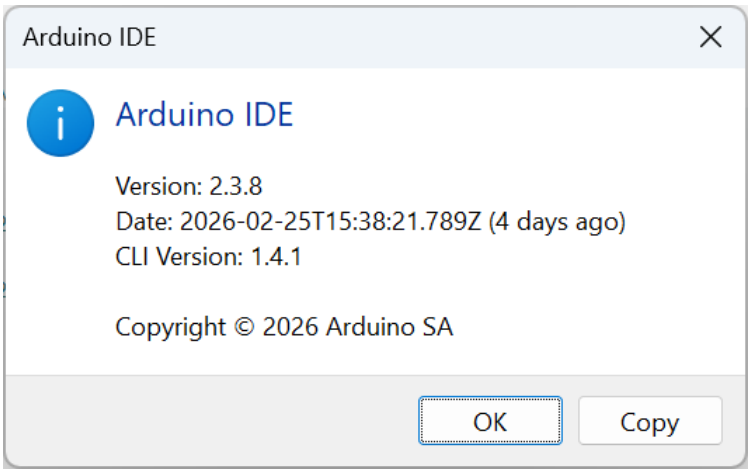
```
This Railuino library allows you to control your digital
Maerklin railway
using Arduino. It can be used in two flavors:

(1) As a Mobile Station 2. You can use a Sparkfun CAN shield and
a
    cable to connect your Arduino to the Digital Connector Box
of
    a Mobile Station 2 and probably (untested) also to a Central
    Station 2. This allows you to control locomotives, functions
    and turnouts using MM2, DCC and other protocols. You can
even
    read and write decoder CVs, as long as the decoder's
protocol
    supports it (DCC does both, MM2 only write, others vary).

(2) As an infrared controller. You can use a simple infrared LED
    and a resistor to have your Arduino talk to the very basic
IR
    receiver box that comes with various starter packages. This
    allows you to control four MM2 locomotives on fixed
addresses
    (the classic Delta addresses). You can also control sixteen
    turnouts (this is an undocumented feature of the IR box, and
    the original Marklin IR controller doesn't support it).
https://github.com/MBuratto/railuino/tree/master/Railuino/src

  1 | #include <Railuino.h>
```

```
2
3  void setup() {
4    Serial.begin(115200);
5    Railuino.begin(Serial); // Verbindung zur CS2/CS3 oder Glei
6  }
7
8  void loop() {
9    // Beispiel: Lokstatus abfragen
10   locoData loco;
11   if (Railuino.getLoco(3, &loco)) { // Lok mit Adresse 3
12     Serial.print("Geschwindigkeit: ");
13     Serial.println(loco.speed);
14     Serial.print("Richtung: ");
15     Serial.println(loco.direction ? "Vorwärts" : "Rückwärts")
16   }
17
18   // Beispiel: Weichenstatus abfragen
19   accessoryData acc;
20   if (Railuino.getAccessory(5, &acc)) { // Magnetartikel mit
21     Serial.print("Weichenstellung: ");
22     Serial.println(acc.state ? "Gerade" : "Abzweig");
23   }
24
25   delay(1000);
26 }
```

The `getLoco()` and `getAccessory()` functions provide you with the current state known to the control unit.

For this to work, the control unit must send status messages to the Arduino (using the mfx/MM2/DCC protocol). The connection is usually established via a Serial Gateway sketch (e.g., Serial_RS4.ino) on a second Arduino or directly via Ethernet/Wi-Fi, depending on the hardware.

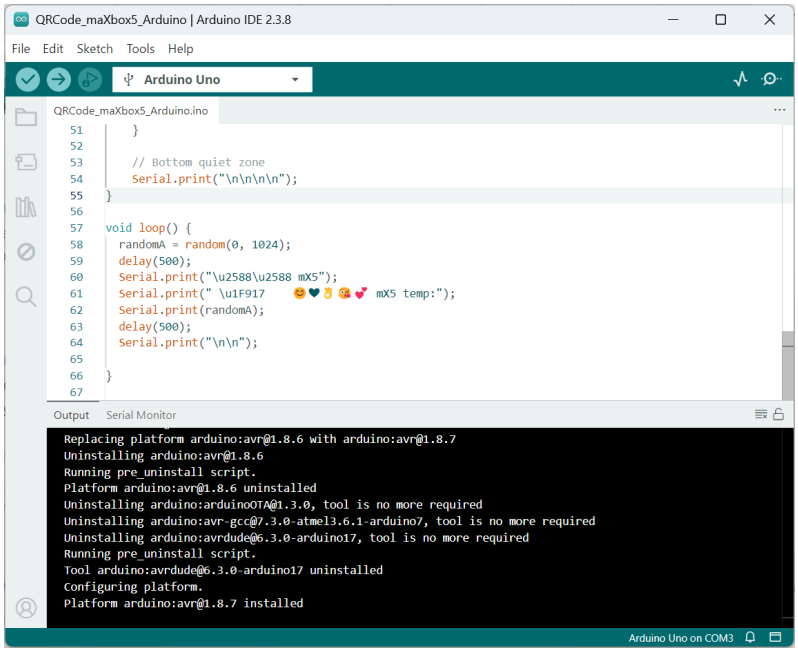
First we update the arduino IDE and the platform too (replacing):



Update installer

The Arduino platform consists of several core elements:

- Arduino boards: These include models like the Arduino Uno, Nano, and Mega.
- Arduino IDE: A cross-platform development environment used to write, compile, and upload code to the board.
- Arduino programming: Based on simplified C/C++, making it accessible to beginners.
- Libraries and examples: Ready-to-use code that simplifies working with sensors, displays, motors, and more.

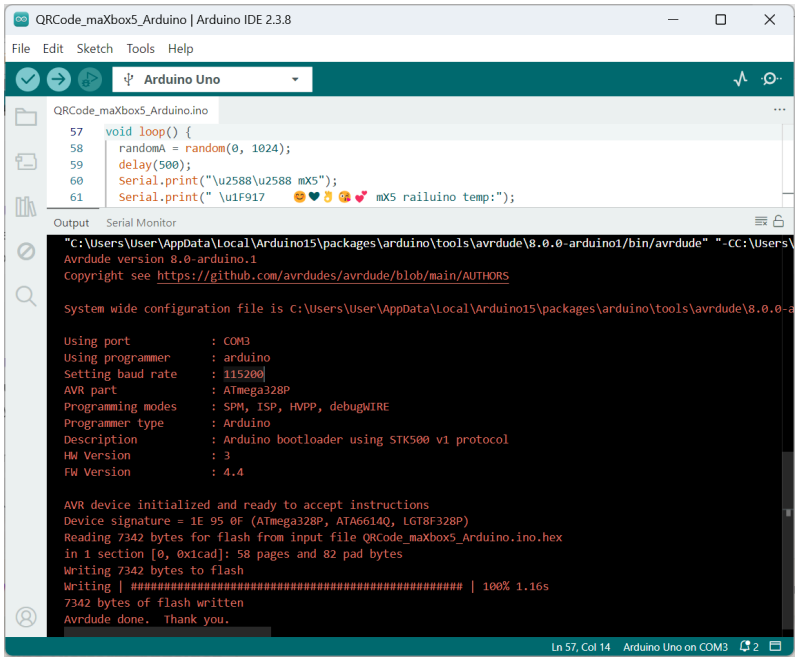


Arduino Platform update to 1.8.7

AVR programming involves using tools like AVRDUDE and AVRFLASH for progr amming AVR microcontrollers.

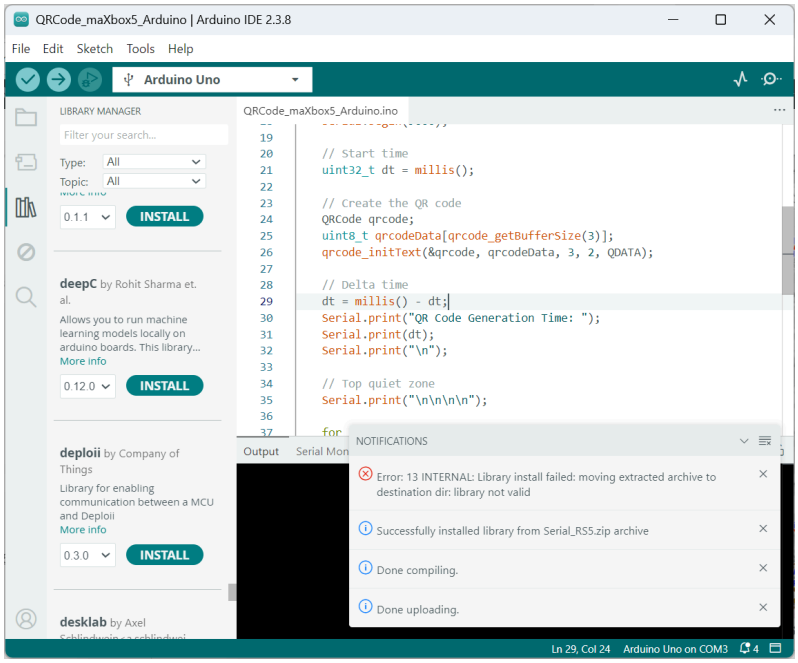
AVRFLASH is a free software application designed for Windows that enables pr ogramming of AVR microcontrollers, including those soldered into printed circui t boards. It is compatible with multiple Windows versions and provides a user-friendly interface for AVR programming.

AVRDUDE is a software tool that allows you to program various AVR controllers , including those with different firmware versions. It supports multiple program ming methods and can be used with various adapters. 1



Flash update

So we install a serial library and try to install the old railuino 0.9 but it failed:



failed notification

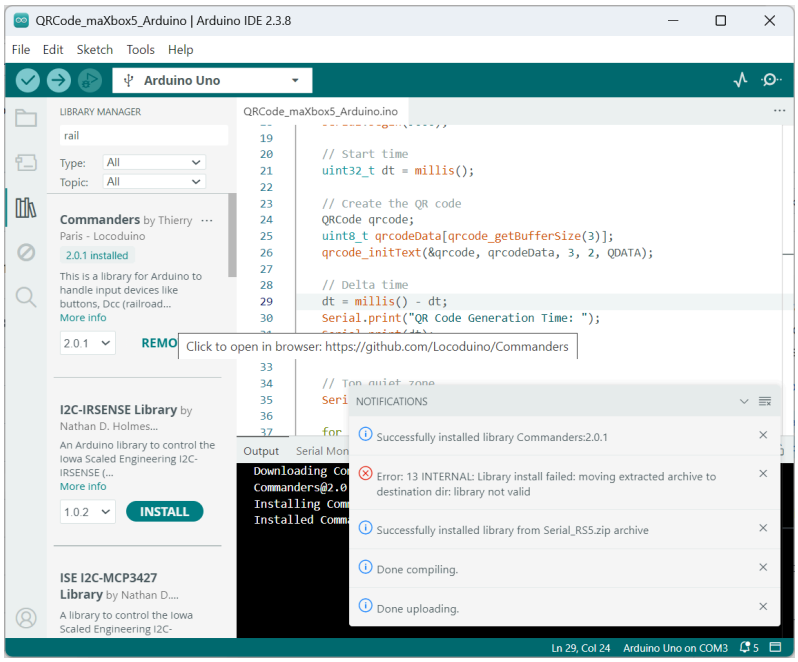
2. Locoduino

So we try to install commanders from Locoduino!

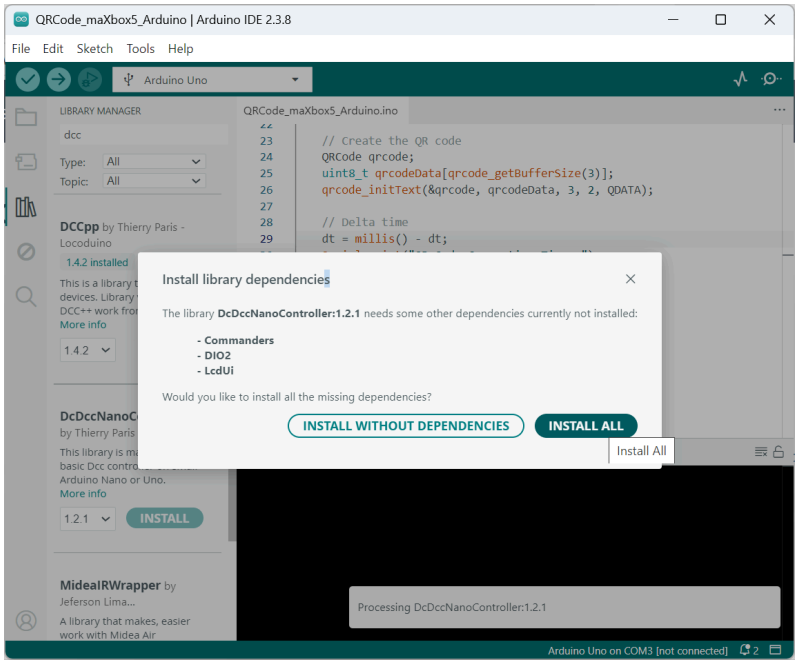
This is a library for Arduino to handle buttons, Dcc (Railroad Modeling), CAN bus, I2C bus, or serial interface to give orders.

In extra directory, you will find

-DIO2 : this library MUST be installed. Commanders cannot operate without it. This is a fast access input/ouput ports library almost whithout changing classic Arduino syntax. -mcp-can : if you need a CAN interface with Commanders, this library is able to work with any shield or circuit using MCP2515 chip. If you comment the line #define NO_CANCOMMANDER, you MUST use this library. -DCC_Monitor_Accessories : This independant ino sketch is only a way to check the Dcc packets arriving to the Arduino. The DCC_decoder have been adapted to focus on accessories packets.



Install Commanders

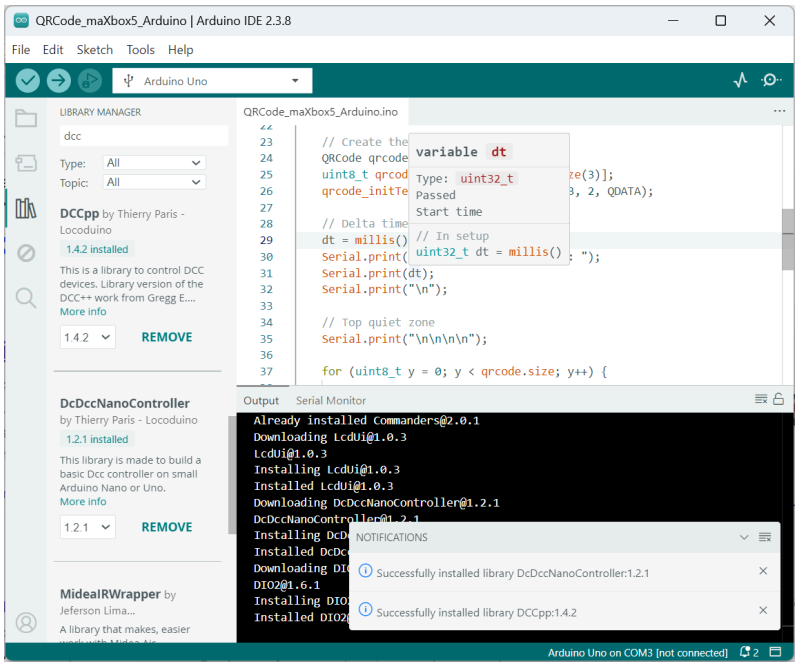


Install Dependencies

Libraries

Accessories	This is a library for Arduino to handle accessories like lights, motors.
Commanders	This is a library for Arduino to handle input devices like buttons, Dcc (railroad modeling), CAN or I2C bus, or serial interface to give orders.
DCCpp	This is a library to control DCC devices.

DcDccNanoController	This library is made to build a basic Dcc controller on small Arduino Nano or Uno.
DIO2	Fast digital input/output functions.
EEPROMextent	This is a library for Arduino to read/write complex data to EEPROM.
LcdUi	This library is made to build sophisticated User Interface on Arduino for text Lcds.
LightDimmer	This library allows to drive LEDs to simulate railroad signals.
LightEffect	This library allows you to easily create light animations from an Arduino board or an ATtiny microcontroller (traffic lights, chaser, shopkeeper sign, etc.)
MemoryUsage	Use this library to check your SRAM / Stack memory usage.
RingBuffer	This library allows to use ring buffer with and without interrupts.
ScheduleTable	This library allows to schedule multiple actions along time.
SlowMotionServo	This library allows to move multiple servos slowly.



Complete Setup

Locoduino – Arduino Libraries

How it works and how to connect?

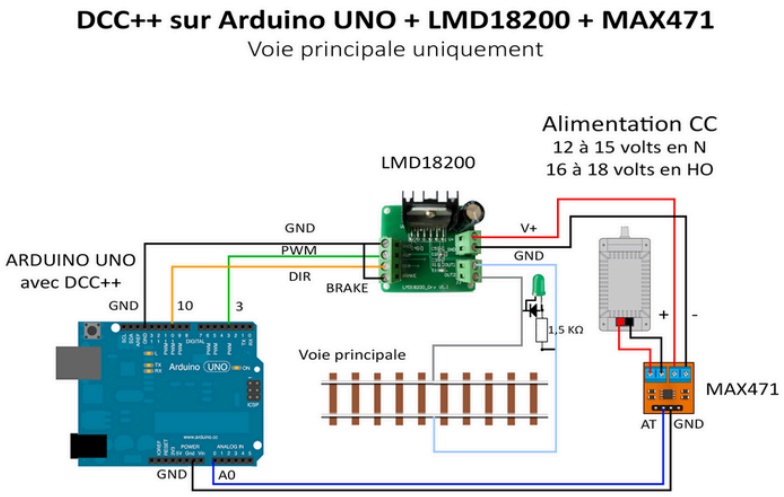
The major innovation of DCC technology lies in transmitting this "control" information through the power circuit; everything passes through the track. And since our Arduinos, which generate the control information, are quite incapable of producing the necessary currents and voltages on even a modest layout, a motor driver board must be added.

The motor driver board amplifies the Arduino's output signal, preserving the shape of the input signal, and applies an output voltage from an external power supply of +15 to -15V (for N scale) or +18 to -18V (for HO scale), with a current that is typically between 1 and 3 amps.

The signal shown above will no longer be a few hundred milliamps at 5V at the Arduino output, but several amps at 15 or 18 volts. This brings us to the power supply, which we will now briefly discuss.

LOCODUINO – Réalisation de centrales DCC avec le logiciel libre DCC++ (3)

To put it simply, we can consider that an HO network with 4 to 5 locomotives operating simultaneously consumes approximately 2 to 2.5A and the same in N, approximately 1.5 to 2A.



LOCODUINO – Réalisation de centrales DCC avec le logiciel libre DCC++ (3)

For reasons of good experience, we have limited our choice to the LMD18200 alone, whose qualities have often been highlighted here on Locoduino and in particular that of being able to support currents of around 3 Amps, each board being supplied with a heatsink.

The Arduino Motor Shield is a versatile add-on board designed to control DC motors, stepper motors, and servo motors. It is based on the L298 dual full-bridge driver, which allows it to handle inductive loads like motors and relays. This shield is ideal for robotics and automation projects, enabling precise control over motor speed, direction, and current sensing.

L'avantage d'un shield réside dans la simplification du câblage et l'encombrement du montage. Les deux shields moteurs que nous vous présentons disposent de deux sorties permettant d'alimenter la voie principale et la voie de programmation.

Dans le fichier `config.h` de **DCC++**,

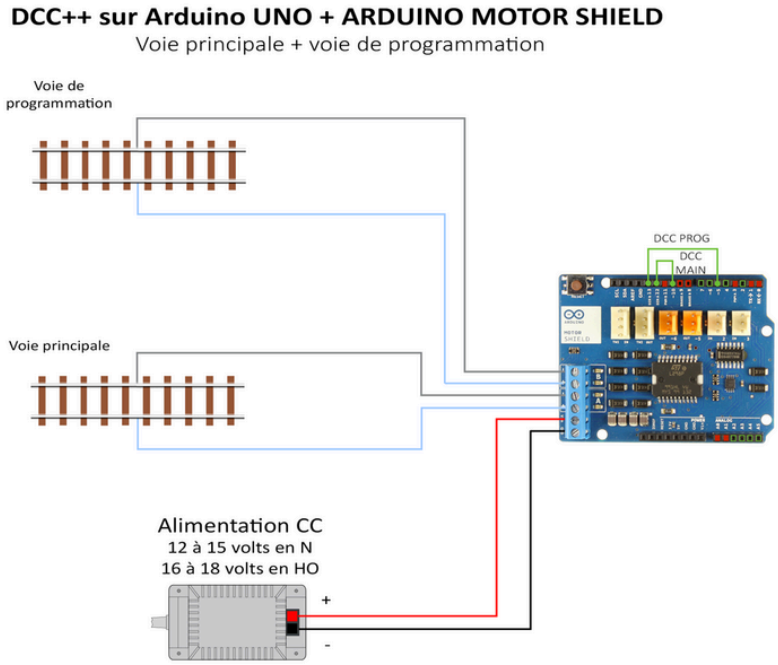
`#define MOTOR_SHIELD_TYPE` devra être à 0 mais c'est l'option par défaut.

```
// DEFINE MOTOR_SHIELD_TYPE ACCORDING TO THE FOLLOWING TABLE:
//
//  0 = ARDUINO MOTOR SHIELD          (MAX 18V/2A PER CHANNEL)
//  1 = POLOLU MC33926 MOTOR SHIELD   (MAX 28V/3A PER CHANNEL)

#define MOTOR_SHIELD_TYPE   0
```

Vous pourrez en profiter pour remplacer `ARDUINO MOTOR SHIELD` par :

```
// 0 = LMD18200 MOTOR SHIELD (MAX 28V/3A PER CHANNEL)
```

Config.h



LOCODUINO – Réalisation de centrales DCC avec le logiciel libre DCC++ (3)

Conclusion

In conclusion:
You now have several proven hardware configurations and the means to implement them to use DCC++ on your layouts.

The Arduino Motor Shield is an economical and fairly simple solution to set up. It includes built-in overvoltage, short-circuit, and over-temperature detection. It has one output for the main track and one for the programming track. However, its performance limits its use to small and medium-sized layouts. Pay attention to the build quality. Not all L298-based boards are created equal.

The POLOLU Dual MC33926 Motor Driver Shield for Arduino offers a complete and integrated solution; it includes overvoltage, short-circuit, and over-temperature detection. It has one output for the main track and one for the programming track. However, its performance is slightly below that of an LMD18200, and its price is high, around €30. The product and build quality seem impeccable to us.

The LMD18200 is a bit of a favorite at Locoduino. It's true that its performance is very good and its price is affordable (€10). It does require a little more wiring and the addition of an external MAX471 (because the integrated current

measurement isn't satisfactory). If you only plan to use the main track, it's undoubtedly the right choice.

As we've already highlighted in previous articles the ease of communication with DCC++, once your circuit is built, you can easily test it using the Arduino IDE monitor. Enter the three characters <1> to power up your circuit. The LEDs on your shield, or those we've added to the LMD18200 circuit, should light up.

You need below hardwares to run Locoduino with Motoshield.

- Arduino board (duemilanove328 or UNO R3 or compatible one, for example Arduino UNO R3, Akiduki's AT-MEGA board)
- MC33926 Motor Driver Shield (based on L298 dual full-bridge driver)
- USB cable (suitable with your arduino board)
- Windows 11 PC. Please install .NET Framework 4.0 runtime package if not installed.
- Single Rail Tracks for N or H0
- Power supply for 12 – 18VDC up to 3 amps (36VA if 12V).
- Track cable and power supply cable for programming or main tracks

```
/*******************
 * Centrale DCC minimum avec un potentiomètre de vitesse connecté
 * sur le port analogique 0,
 * un bouton poussoir connecté entre le 0V et l'entrée digitale 4
 * Le signal DCC est délivré sur la Pin 9, et est capable de piloter
 * un booster à base de LMD18200 directement.
 *******************/

#include <DCCPacket.h>
#include <DCCPacketQueue.h>
#include <DCCPacketScheduler.h>

DCCPacketScheduler dps;
unsigned int analog_value;
char speed_byte, old_speed = 0;
byte count = 0;
byte prev_state = 1;
byte F0 = 0;

void setup() {
  Serial.begin(9600);
  dps.setup();  // initialisation de la librairie
```

LOCODUINO – Comment piloter trains et accessoires en DCC avec un Arduino (1)

CC 40100 with a DCC compatible Decoder

Electrotren and Hornby DCC

The Hornby system differs from the conventional type of 12V controllers in as much that it is the individual locomotives that are controlled internally rather than the controlling of a locomotive's speed and direction by varying the current to the track.

Posted in arduino, Controller, Trains

Leave a comment

← Cité du Train News 2025