

Code Blog

Railuino

👤 [maxbox4](#) 🕒 November 26, 2020 📁 [Arduino, Code, Operation Systems](#) ✎ [Edit](#)



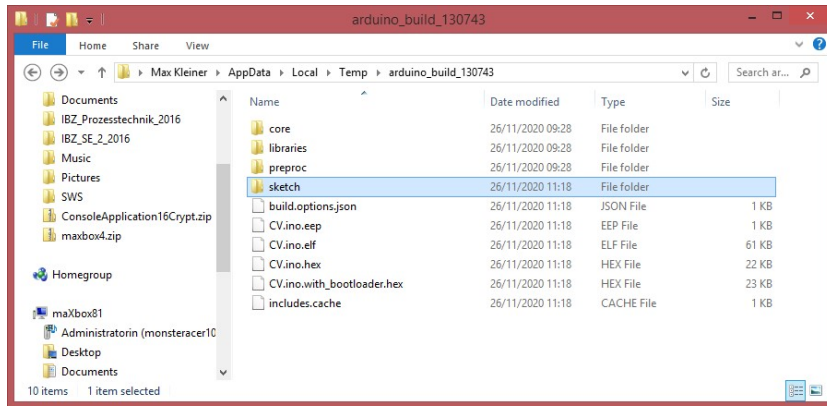
Hacking your Märklin

This library allows you to control your digital Märklin railway using Arduino.

<https://code.google.com/archive/p/railuino/>

I want to show especially the output of the C++ compiled library

include <Railuino.h>



Any ISP will need what Arduino calls output binaries and the rest of the world calls HEX files. These are produced when you Verify/Compile your sketch and contain the data the AVR microcontroller needs to run and in my case “C:\Users\max\AppData\Local\Temp\arduino_build_130743”.

The Arduino IDE creates them in temporary folders in your user libraries. If everything has gone right, your folder should be full of output files, mostly with .o and .d extensions. These are used by linker and can be ignored.

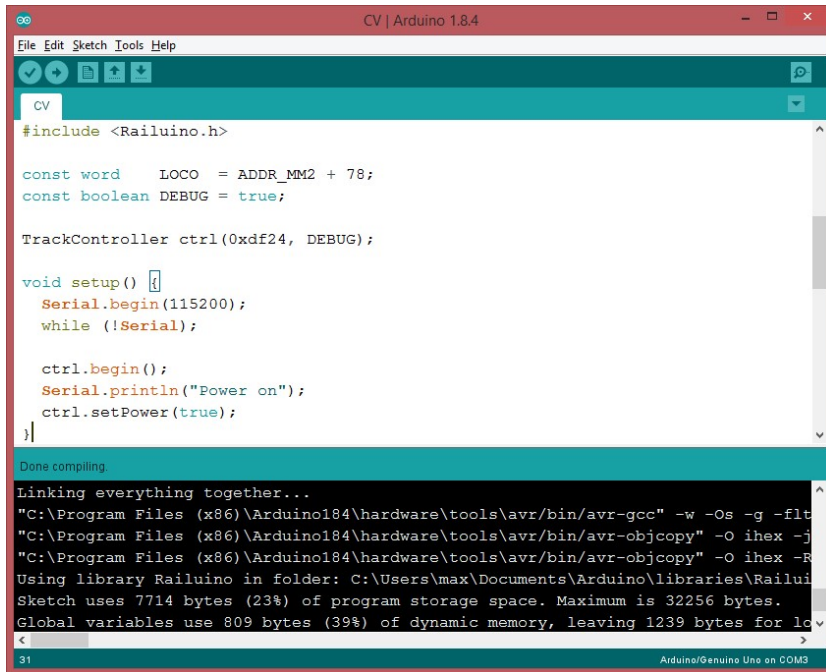
Where to find Arduino Hex files or Output Binaries

The important files will be these, where the sketch is called CV.ino, so look for your own sketch name – so I show the top 6 files in the right sequence with explanation (below you can see the full verbose log mode):

- CV.ino – the sketch as script before transformed to cpp++
- CV.ino.cpp – output C file, actually C++
- CV.ino.cpp.o – output for the linker
- CV.ino.elf – output file for a debugger
- CV.ino.eep – EEPROM file for programmer
- CV.ino.hex – flash (code) file for programmer

The C file (.cpp) and Elf file (.elf) can be used in AVR Studio development environment if you want to move away from just using Arduino IDE.

The important files for the programmer are the .Hex and .EEP files.



```

CV
#include <Railuino.h>

const word    LOCO = ADDR_MM2 + 78;
const boolean DEBUG = true;

TrackController ctrl(0xdf24, DEBUG);

void setup() {
  Serial.begin(115200);
  while (!Serial);

  ctrl.begin();
  Serial.println("Power on!");
  ctrl.setPower(true);
}

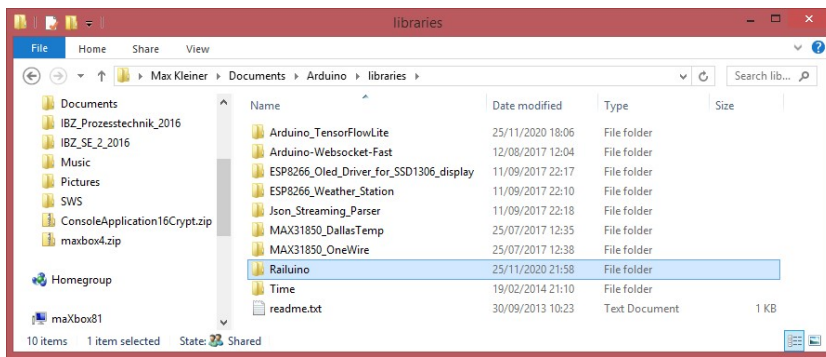
Done compiling
Linking everything together...
"C:\Program Files (x86)\Arduino184\hardware\tools\avr\bin\avr-gcc" -w -Os -g -flt
"C:\Program Files (x86)\Arduino184\hardware\tools\avr\bin\avr-objcopy" -O ihex -j
"C:\Program Files (x86)\Arduino184\hardware\tools\avr\bin\avr-objcopy" -O ihex -R
Using library Railuino in folder: C:\Users\max\Documents\Arduino\libraries\Railuino
Sketch uses 7714 bytes (23%) of program storage space. Maximum is 32256 bytes.
Global variables use 809 bytes (39%) of dynamic memory, leaving 1239 bytes for local variables.
31
Arduino/Genuino Uno on COM3

```

Arduino 1.8.4 with compiled Railuino

Install the Package (zip)

Installation is easy: Just get the latest release from the downloads page (see above Google Code Archive) and place the contents of the “src” directory in a “Railuino” directory under your Arduino “libraries” directory. I did also a properties file to better integrate in the Arduino IDE with paragraph and includes, but that’s not so important and not a must.



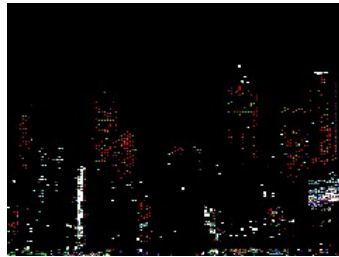
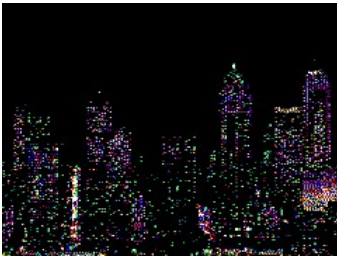
Lib Location of Arduino

In case of problems:

There are three common causes of the invalid library warning:

- 1 | 1. You saved a sketch to the libraries folder. Sketches are only allowed in
- 2 | 2. Incorrect installation of a valid library. The library folder must be in
- 3 | 3. Something that's neither library nor sketch in the libraries folder. Fi

Then restart Arduino. You should now see a bunch of new examples that teach you how to use Railuino. The “Misc/Tests” example is a good way of validating your setup.



PiXMap

For documentation on the functions I currently recommend to read the comments in the “Railuino.h” header file. There are also several sets of slides on the downloads page that describe the overall approach and the hardware. Finally, there is a video from LinuxTag and another one from DroidConNL on YouTube.



Full Verbose Log

```
1 | C:\Program Files (x86)\Arduino184\arduino-builder -dump-prefs -logger=mach:
```

- CV.ino – the Sketch as Script

```
C:\Program Files (x86)\Arduino184\arduino-builder -compile -logger=machine -hardware
C:\Program Files (x86)\Arduino184\hardware -hardware C:\Users\max\AppData\Local
\Arduino15\packages -tools C:\Program Files (x86)\Arduino184\tools-builder -tools C:\Program
Files (x86)\Arduino184\hardware\tools\avr -tools C:\Users\max\AppData\Local\Arduino15
\packages -built-in-libraries
```

```
C:\Program Files (x86)\Arduino184\libraries -libraries C:\Users\max\Documents\Arduino
\libraries -fqbn=arduino:avr:uno -ide-version=10804 -build-path C:\Users\max\AppData\Local
\Temp\arduino_build_130743 -warnings=none -build-cache C:\Users\max\AppData\Local
\Temp\arduino_cache_811198 -prefs=build.warn_data_percentage=75
-prefs=runtime.tools.avrdude.path=C:\Program Files (x86)\Arduino184\hardware\tools\avr
-prefs=runtime.tools.avr-gcc.path=C:\Program Files (x86)\Arduino184\hardware\tools\avr
```

```
-prefs=runtime.tools.arduinoOTA.path=C:\Program Files (x86)\Arduino184\hardware\tools\avr
-verbose C:\Users\max\Documents\Arduino\libraries\Railuino\src\examples\01.Controller
\CV\CV.ino
Using board 'uno' from platform in folder: C:\Program Files (x86)\Arduino184\hardware
\arduino\avr
```

```
Using core 'arduino' from platform in folder: C:\Program Files (x86)\Arduino184\hardware
\arduino\avr
```

Detecting libraries used...

```
"C:\Program Files (x86)\Arduino184\hardware\tools\avr\bin\avr-g++" -c -g -Os -w -std=gnu++11
-fpermissive -fno-exceptions -ffunction-sections -fdata-sections -fno-threadsafe-statics -flto -w -x
c++ -E -CC -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO=10804
-DARDUINO_AVR_UNO -DARDUINO_ARCH_AVR "-IC:\Program Files (x86)\Arduino184
\hardware\arduino\avr\cores\arduino" "-IC:\Program Files (x86)\Arduino184\hardware\arduino
\avr\variants\standard" "C:\Users\max\AppData\Local\Temp\arduino_build_130743\sketch
\CV.ino.cpp" -o "nul"
```

```
"C:\Program Files (x86)\Arduino184\hardware\tools\avr\bin\avr-g++" -c -g -Os -w -std=gnu++11
-fpermissive -fno-exceptions -ffunction-sections -fdata-sections -fno-threadsafe-statics -flto -w -x
c++ -E -CC -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO=10804
-DARDUINO_AVR_UNO -DARDUINO_ARCH_AVR "-IC:\Program Files (x86)\Arduino184
\hardware\arduino\avr\cores\arduino" "-IC:\Program Files (x86)\Arduino184\hardware\arduino
\avr\variants\standard" "-IC:\Users\max\Documents\Arduino\libraries\Railuino" "C:\Users
\max\AppData\Local\Temp\arduino_build_130743\sketch\CV.ino.cpp" -o "nul"
```

- CV.ino.cpp

```
Using cached library dependencies for file: C:\Users\max\Documents\Arduino\libraries\Railuino
\Railuino.cpp
```

Generating function prototypes...

```
"C:\Program Files (x86)\Arduino184\hardware\tools\avr\bin\avr-g++" -c -g -Os -w -std=gnu++11
-fpermissive -fno-exceptions -ffunction-sections -fdata-sections -fno-threadsafe-statics -flto -w -x
c++ -E -CC -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO=10804
-DARDUINO_AVR_UNO -DARDUINO_ARCH_AVR "-IC:\Program Files (x86)\Arduino184
\hardware\arduino\avr\cores\arduino" "-IC:\Program Files (x86)\Arduino184\hardware\arduino
\avr\variants\standard" "-IC:\Users\max\Documents\Arduino\libraries\Railuino" "C:\Users
\max\AppData\Local\Temp\arduino_build_130743\sketch\CV.ino.cpp" -o "C:\Users
\max\AppData\Local\Temp\arduino_build_130743\preproc\ctags_target_for_gcc_minus_e.cpp"
"C:\Program Files (x86)\Arduino184\tools-builder\ctags\5.8-arduino11\ctags" -u -language-
force=c++ -f - -c++-kinds=svpf -fields=KSTtzn -line-directives "C:\Users\max\AppData\Local
\Temp\arduino_build_130743\preproc\ctags_target_for_gcc_minus_e.cpp"
```

Compiling sketch...

```
"C:\Program Files (x86)\Arduino184\hardware\tools\avr\bin\avr-g++" -c -g -Os -w -std=gnu++11
-fpermissive -fno-exceptions -ffunction-sections -fdata-sections -fno-threadsafe-statics -MMD -flto
-mmcu=atmega328p -DF_CPU=16000000L -DARDUINO=10804 -DARDUINO_AVR_UNO
-DARDUINO_ARCH_AVR "-IC:\Program Files (x86)\Arduino184\hardware\arduino\avr\cores
\arduino" "-IC:\Program Files (x86)\Arduino184\hardware\arduino\avr\variants\standard"
"-IC:\Users\max\Documents\Arduino\libraries\Railuino" "C:\Users\max\AppData\Local
\Temp\arduino_build_130743\sketch\CV.ino.cpp" -o "C:\Users\max\AppData\Local
```

\Temp\arduino_build_130743\sketch\CV.ino.cpp.o”

- CV.ino.cpp.o

Compiling libraries...

Compiling library “Railuino”

Using previously compiled file: C:\Users\max\AppData\Local\Temp\arduino_build_130743\libraries\Railuino\Railuino.cpp.o

Compiling core...

Using precompiled core

Linking everything together...

“C:\Program Files (x86)\Arduino184\hardware\tools\avr\bin\avr-gcc” -w -Os -g -flto -fuse-linker-plugin -Wl,-gc-sections -mmcu=atmega328p -o “C:\Users\max\AppData\Local\Temp\arduino_build_130743\CV.ino.elf”

- CV.ino.elf

“C:\Users\max\AppData\Local\Temp\arduino_build_130743\sketch\CV.ino.cpp.o” “C:\Users\max\AppData\Local\Temp\arduino_build_130743\libraries\Railuino\Railuino.cpp.o” “C:\Users\max\AppData\Local\Temp\arduino_build_130743\..\arduino_cache_811198\core\core_arduino_avr_uno_e2943c849c7d54ca2ad3fdcoef151476.a” “-LC:\Users\max\AppData\Local\Temp\arduino_build_130743” -lm

“C:\Program Files (x86)\Arduino184\hardware\tools\avr\bin\avr-objcopy” -O ihex -j .eeprom --set-section-flags=.eeprom=alloc,load --no-change-warnings --change-section-lma .eeprom=0

“C:\Users\max\AppData\Local\Temp\arduino_build_130743\CV.ino.elf” “C:\Users\max\AppData\Local\Temp\arduino_build_130743\CV.ino.eep”

- CV.ino.eep

“C:\Program Files (x86)\Arduino184\hardware\tools\avr\bin\avr-objcopy” -O ihex -R .eeprom

“C:\Users\max\AppData\Local\Temp\arduino_build_130743\CV.ino.elf” “C:\Users\max\AppData\Local\Temp\arduino_build_130743\CV.ino.hex”

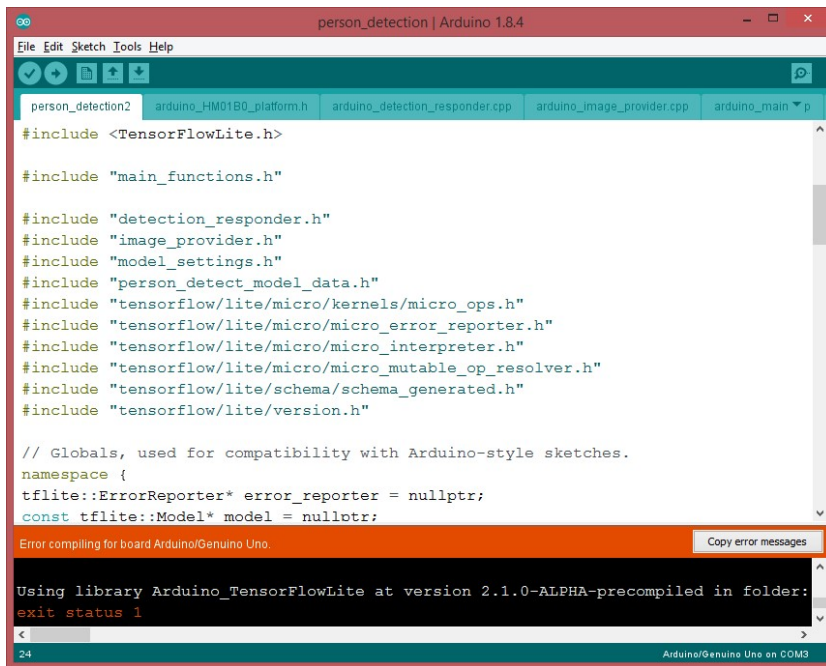
- **CV.ino.hex**

Using library Railuino in folder: C:\Users\max\Documents\Arduino\libraries\Railuino (legacy)

Sketch uses 7714 bytes (23%) of program storage space. **Maximum is 32256 bytes.**

Global variables use 809 bytes (39%) of dynamic memory, leaving 1239 bytes for local variables.

Maximum is 2048 bytes.



```
person_detection | Arduino 1.8.4
File Edit Sketch Tools Help
person_detection2 | arduino_HM0180_platform.h | arduino_detection_responder.cpp | arduino_image_provider.cpp | arduino_main.p
#include <TensorFlowLite.h>

#include "main_functions.h"

#include "detection_responder.h"
#include "image_provider.h"
#include "model_settings.h"
#include "person_detect_model_data.h"
#include "tensorflow/lite/micro/kernels/micro_ops.h"
#include "tensorflow/lite/micro/micro_error_reporter.h"
#include "tensorflow/lite/micro/micro_interpreter.h"
#include "tensorflow/lite/micro/micro_mutable_op_resolver.h"
#include "tensorflow/lite/schema/schema_generated.h"
#include "tensorflow/lite/version.h"

// Globals, used for compatibility with Arduino-style sketches.
namespace {
  tflite::ErrorReporter* error_reporter = nullptr;
  const tflite::Model* model = nullptr;
}

Error compiling for board Arduino/Genuino Uno.

Using library Arduino_TensorFlowLite at version 2.1.0-ALPHA-precompiled in folder:
exit status 1
24 Arduino/Genuino Uno on COM3
```

This is another context from Railuino to TensorFlowLite

There's a second framework called Ardurail.

<https://sourceforge.net/p/ardurail/wiki/Home/>

This library allows you to create a digital Märklin(tm)-Motorola(tm) compatible signal for driving model-rail locomotives and track switches. You also need:

1. a booster (a kind of digital amplifier, see http://en.wikipedia.org/wiki/Digital_model_railway_control_systems#Booster)
2. some external controller (light pots, switches, buttons, ...) or
3. some external hard/software which speaks the Märklin(tm) P50-protocol over a serial interface (like [RocRail (<http://wiki.rocrail.net>) or [srcpd (<http://srcpd.sourceforge.net/>)] in conjunction with the derived Ardurail_P50 class.


```

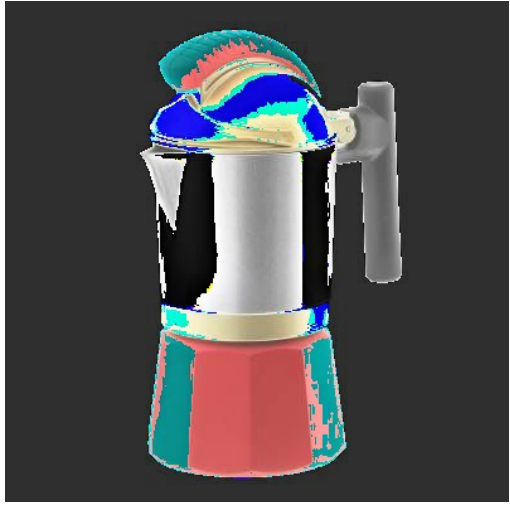
maxBox4 Edit Fenster
maxBox4 ScriptStudio 210_public_private_cryptosystem4_libz.txt
File Program Options View Debug Output Help
Load Run Replace/Reflect Go Compiler Use Class Tutorial Resources Setup
//*****
program Public_Private_Crypto_Simulation4_LIBz;
//calculates simple public private with strong string and hash back SHA1 ;
//http://en.wikipedia.org/wiki/Hash_function,#eign:crossover: EBG@acbook: 11.01.2021 22:22:21
//public-private crypto system for education (ppcse), #loc>230
//
//Crypto Systems Ref:
1. Prime1: 13 and Prime2: 23
2. RSA Modul Public [p*q] as N: 299
3. phi (N) Private: 264
4. RSA Exponent: 5
5. Private d: 53
6. Public (5,299) - Private (53,299)
//
Const
  AFILe = 'maxbox4.exe';
  AMESSAGE = 'HI! HELLO WORLD OF CRYPTOBOX 3!'; //only to ASCII!
  OFFSET = 32; //64; // space depends on N mod range
//
Var
  hexd: THexArray;
  prime1, prime2, rsa exponent: integer;
maxBox4 7:\Applications\maxBox4\sample\210_public_private_cryptosystem4_libz.txt Compiled: 11.01.2021 22:22:23 Mem: 61% Runtime: 0:0:6.873 Thread: 3
HI! URYYB JBEYQ BS PELCGBOBX 3!
HI! HELLO WORLD OF CRYPTOBOX 3!
maxBox4 executed: 11.01.2021 22:22:28 Runtime: 0:0:6.873 Memload: 61% use
PascalScript: maxBox4 - RemObjects & SynEdit

```

maXbox Mac

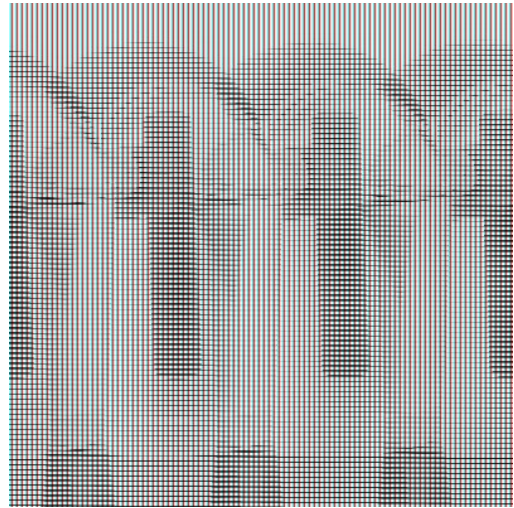


NS 1310 and V220 Loc





ppm filter



```

1  function ColorToGray(Color: TColor): TColor;
2  var L: Byte;
3  begin
4      L:= round(0.2126*GetRValue(Color)+0.7152*GetGValue(Color)+0.0722*
5          GetBValue(Color));
6      Result:= RGB(L, L, L);
7  end;
8
9  procedure TBitmapHelperSaveAsPPM_4(FileName: TFileName; abit: TBitmap;
10                                     useGrayScale: Boolean);
11  var
12      i, j: Integer;
13      Header: AnsiString;
14      ppm: TMemoryStream;
15      agb: TBytes;
16  begin
17      ppm:= TMemoryStream.Create;
18      try
19          Header:= Format('P6'#10'd %d'#10'255'#10, [abit.Width, abit.Height]);
20          writeln(Header);
21          ppm.WriteBuffer((Header), Length(Header));
22          setlength(agb,3)
23          for i:= 0 to abit.Width- 1 do
24              for j:= 0 to abit.Height- 1 do begin
25                  if useGrayScale then
26                      agb:= InttoBytes(ColorToGray(ColorToRGB(abit.Canvas.Pixels[j,i]
27                  else
28                      agb:= InttoBytes(ColorToRGB(abit.Canvas.Pixels[j,i]));
29                  ppm.Write(stringOf(agb), 3);
30                  //ppm.Write(ByteTostring(rgb), 3);
31              end;
32          ppm.SaveToFile(FileName);
33      finally
34          ppm.Free;
35      end;
36  end;
37
38  bitmap:= TBitmap.Create;
39  bitmap.LoadFromFile(exepath+'\web\coffeemax.bmp');
40  //SaveAsPPM('Output.ppm');
41  writeln(itoa(bitmap.width))
42  TBitmapHelperSaveAsPPM_4(exepath+'\web\coffeemaxg.ppm', bitmap, true);
43  bitmap.Free;
44
45  //http://paulcuth.me.uk/netpbm-viewer/
46  //https://rosettacode.org/wiki/Bitmap/Read_a_PPM_file#Delphi

```

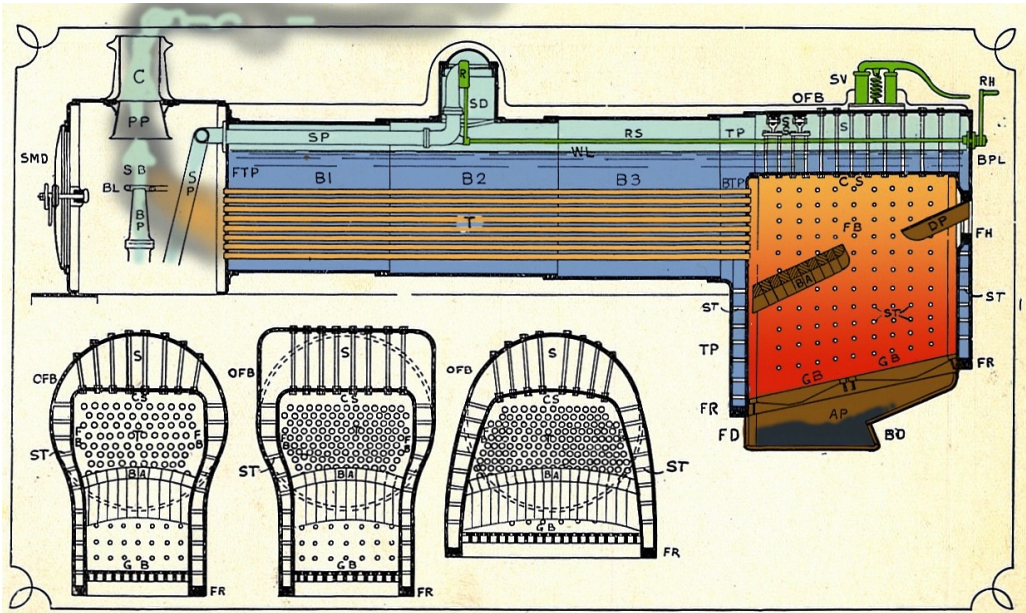
```

1  function ReadCharM(ppm: TMemoryStream): AnsiChar;
2  var mystr: string;
3  begin
4      writeln(itoa(ppm.size))
5      SetLength(mystr, 1);
6      //insize:= MemStream.read(strBuff2, length(strBuff2));
7      ppm.Read(mystr, length(mystr));
8      result:= mystr[1];
9      writeln('res: '+mystr)
10 end;



```

Artificial Intelligence and Machine Learning are much trending and also confused terms nowadays. Machine Learning (ML) is a subset of Artificial Intelligence. ML is a science of designing and applying algorithms that are able to learn things from past cases. If some behaviour exists in past,

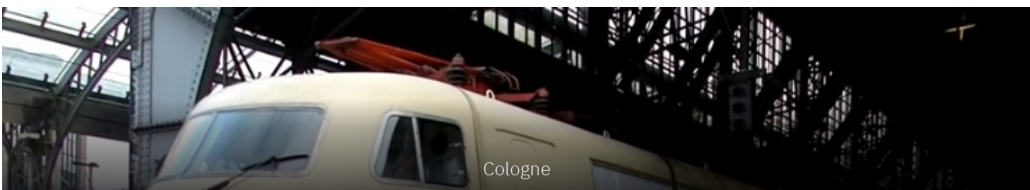
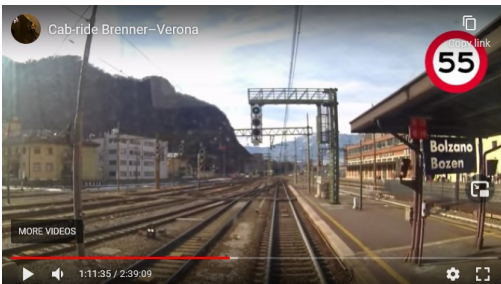
then you may predict if or it can happen again. Means if there are no past cases then there is no prediction. [...]



AP	ASH PAN	CS	CROWN SHEET	RS	REGULATOR SPINDLE	WL	WATER LEVEL
B1	BOILER BARREL (RING 1)	DP	DEFLECTOR PLATE	S	STAYS		
B2	BOILER BARREL (RING 2)	FB	FIREBOX (INNER)	SB	SMOKEBOX		
B3	BOILER BARREL (RING 3)	FD	FRONT DAMPER	SD	STEAM DOME		
BA	BRICK ARCH	FH	FIRE HOLE	SMD	SMOKEBOX DOOR		
BD	BACK DAMPER	FR	FOUNDATION RING	SP	STEAM PIPE		
BL	BLOWER	FTP	FRONT TUBE PLATE	SS	SLING STAYS		
BP	BLAST PIPE	GB	GRATE BARS	ST	SCREW STAYS		
BPL	BACK PLATE	PP	PETTICOAT PIPE	SV	SAFETY VALVES		
BTP	BACK TUBE PLATE	R	REGULATOR	T	TUBES		
C	CHIMNEY	RH	REGULATOR HANDLE	TP	THROAT PLATE		


 J.F. GAINES (1923).
 RAILWAYS FOR ALL,
 FACING P.49, WARD LOCK.
 PUBLIC DOMAIN.


Water treatment for Steam Locomotive





Train Cabs

Advertisements

Occasionally, some of your visitors may see an advertisement here, as well as a [Privacy & Cookies banner](#) at the bottom of the page. You can hide ads completely by upgrading to one of our paid plans.

[UPGRADE NOW](#)[DISMISS MESSAGE](#)

 [maxbox4](#)  [November 26, 2020](#)  [Arduino, Code, Operation Systems](#)  [Edit](#)

Published by maxbox4

Code till the End [View more posts](#)

2 thoughts on “Railuino”

[maxbox4](#)

[November 26, 2020 at 12:32 pm](#)



This was also a personal moment cause we have been visited the first Maker Faire in 2013 at Hannover! and the initiator of Railuino Jörg Pleumann had this presentation at the Maker Faire Hannover on a very hot summer day 03.08.2013 with a great pleasure and full of rich details.

 Like

[Reply](#)

maxbox4

November 26, 2020 at 1:24 pm



Modelleisenbahnen sind heute eine extrem digitalisierte Geschichte, bei der das Gleis sowohl die Betriebsspannung liefert als auch als Datenbus zu den Loks dient. Wenn man eine Märklin-Bahn mit dem Rechner oder dem Handy steuern möchte, braucht man zusätzlich zu Gleisen und Rollmaterial Digital-Equipment für 500-1000 Euro aufwärts (Central Station, Router, iPhone). Der Vortrag stellt eine Alternative vor: Es gibt zwei günstige Anschlussboxen, die in vielen Startpackungen zu finden sind: Eine sehr einfache IR-Fernbedienung und eine kabelgebundene sogenannte Mobile Station. Mit Hilfe des Arduino lassen sich beide leicht "hacken". Der Arduino erzeugt in diesem Fall die Signale, die eigentlich das Handgerät erzeugen würde. Zusätzlich kann der Arduino zum Beispiel per Ethernet mit anderen Geräten verbunden werden. Damit lassen sich mit relativ wenig finanziellem Aufwand, etwas Zeit und viel Spaß Anwendungsszenarien umsetzen, die sonst der High-End-Hardware vorbehalten sind. Zusätzlich zeigt der Vortrag, wie ein standardisierter Rückkanal ("S88-Bus") mit dem Arduino implementiert werden kann. Dieser liefert Feedback darüber, welche Gleise von Zügen passiert werden. Beide Mechanismen zusammen ergeben eine einfache, erweiterbare und auf Wunsch sogar autonome Steuerung für die Modellbahn.

★ Like

[Reply](#)

Leave a Reply