Voorwoord

Het idee om zelf een modelbaanbesturingsprogramma te maken kwam een aantal jaren terug. Ik had net een Intellibox (zie appendix B) gekocht. Op de cd-rom hierbij stonden een aantal demo's van modeltreinbesturingsprogramma's. Wat mij meteen opviel was dat deze programma's niet op verschillende computers tegelijk konden draaien, ter visualisatie van een deel van de modelbaan, terwijl er maar 1 computer de sturing verzorgde.
Bijna 2 jaar later begon ik als junior-programmeur bij Dolmen IP, waar ik enorm veel ervaring heb opgedaan in de automatisatie, seriële communicatie en het programmeren in Delphi.
Ook bestaan er enkel modeltreinbesturingsprogramma's voor Windows, en een enkele voor OS/2.

De bestaande gratis alternatieven, bezitten dezelfde beperkingen. Ze werken soms ook niet volledig zonder eerst te registreren. En, opnieuw, maximum 1 keer registreren toegestaan.

De deur naar de Open-Source wereld moet met dit project voor mij openen. Er bestaan voorlopig nog geen actieve projecten van deze aard, dus zou dit een mooie primeur worden.

Tenslotte wil ik volgende mensen bedanken.
Veerle Tielemans, voor het geduld en het nakijken van alle tekst.
Pascal Bracke, voor de uitleg en vele tips in verband met sommige specifieke programmeertechnieken, zoals threads.
Mijn familie voor de steun.
Top 1 Toys - Techno Hobby - Vadec Halle, voor de ideeën, testen en materiaal.
En uiteraard de school en de leraren voor de fantastische schooljaren en de aangeboden kennis.

Inhoudstafel

1 Situeringsplan

1.1 Huidige situatie: software

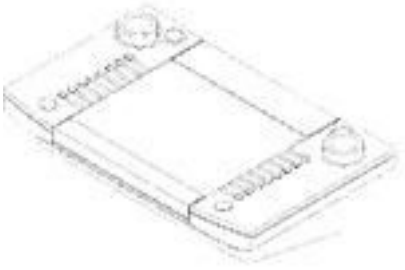
De bestaande software voor dit doel, zit in de duurdere prijsklasse.
Zoals in het voorwoord al vermeld, zijn er enkel voor Windows programma's beschikbaar.
Ook heb ik geen enkel programma gevonden dat op client-server basis werkt.
Tenslotte zijn er nog nauwelijks open-source alternatieven beschikbaar.

1.2 Huidige situatie: hardware

In den beginne van de digitale modelbaan, voor Märklin was dit omstreeks 1985, waren er vele apparaten nodig voor een kleine modelbaan. Deze zijn nu nog te verkrijgen.

Zo heeft men een centrale eenheid, het hart van het systeem, dat enkel bedoeld is om de instructies te verwerken en door te sturen.
Daarnaast heeft men de control 80 & control 80 f. Enkel het besturen van de locomotieven
De keyboard en memory om respectievelijk wissels en wisselstraten te bedienen.
Tot slot alle decoders, terugmelddecoders en de interface. De interface dient om de modelbaan aan een PC te verbinden.
Iets later heeft men de control 80 f en de centrale eenheid in 1 toestel geïntegreerd.
In 1998 zijn Uhlenbrock en Modeltreno met een toestel, de Intellibox, gekomen dat al deze toestellen integreert.
Dit is een gigantisch succes, de fabrikanten kunnen de vraag nauwelijks aan, er komt geen antwoord van Märklin en de Intellibox ondersteunt alle uitbreidingen van het Märklin Digitaal systeem, van Roco, Lenz,...

Nu, anno 2004, komt Märklin met een antwoord: Märklin Systems.
Met terugmelding van de decoders zelf, ethernet aansluiting, ...



2 Functionele analyse

2.1 Algemeen

De bedoeling van GhotiTrain is het besturen van een (model)spoorbaan. Dit wordt in eerste instantie verwezenlijkt door de Intellibox.
Verder moet het mogelijk zijn om, voor grote banen, meerdere PC's in een netwerk elk een deel van de baan te laten zien, zonder hiervoor telkens een andere intellibox te moeten gebruiken. In een later stadium mag het platform (=windows, linux, macintosh) geen belemmering vormen.
Tot slot wordt dit opensource, waardoor iedereen vrij is om het verder te ontwikkelen, of op een ander systeem te gebruiken.
De gebruikte databank moet geneste SQL-statements ondersteunen. Ik gebruik MySQL 4.1 Alpha.

2.2 Client

De client verzorgt de visualisatie, programmatie en ingave van alle gegevens en situaties. De client connecteert met de server via de databank. Zo is het mogelijk om meerdere clients te gebruiken.

Omdat het programma voor alle soorten banen te gebruiken is, moet de visualisatie van de baan volledig dynamisch gebeuren. Dit houdt in dat de gebruiker zelf de baan moet kunnen tekenen.
Verder moeten alle eigenschappen, zoals adres, status, invers aangesloten,... achteraf aan te passen zijn. Ook moeten blokken eenvoudig aan te duiden zijn, door een groep rails te selecteren.

De locomotieven moeten erg eenvoudig te bedienen zijn. Er moet een keuze bestaan in verschillende bedieningssystemen. Enkel gedefinieërde functies kunnen bediend worden.

2.3 Server

Voor de server gelden volgende regels:
- 1 server per project;
- 1 intellibox per server.

De server voert alle geprogrammeerde logica uit. De resultaten hiervan gaan weer in de databank, waarna de client deze opnieuw uitleest en de visualisatie verzorgt.
De server bevat dus geen visualisatie.

De server moet eenvoudig op te starten zijn en verder mag er nouwelijks interactie met de gebruiker zijn. De server heeft een eenvoudige interface, enkel bij fouten met de databank worden deze door de server zelf getoond.

2.4 Databank

Doordat ik gebruik maak van geneste sql's, is het noodzakelijk dat de gebruikte databank dit ondersteunt.

Een voorbeeld in Delphi-code:

```
lsSql := 'SELECT ' + aField_Name
+ ' FROM ' + aTable
+ ' WHERE project_id=(SELECT project_id FROM project'
+                '  WHERE project_name=''' + aProjectName + ''')';
```

Dit geeft als sql:

```
SELECT locomotive_name FROM locomotive
 WHERE project_id=(SELECT project_id FROM project WHERE
project_name='Project01')
```

Hier ziet u tussen de haakjes voor project_id een sql staan die een uniek resultaat weergeeft.

Zoals reeds vermeld in de functionele analyse, gebruik ik hiervoor MySQL 4.1 Alpha. Ik gebruik componenten in delphi die MySQL rechtstreeks aansturen, waardoor er enkel met MySQL kan gewerkt worden.

Wat moet er bijgehouden worden?

Aangezien we met verschillende schermen werken en clubs meerdere banen hebben, moeten we in eerste instantie bijhouden over welk project en welk scherm het gaat.
Verder moeten rails, decoders, seinen, rollend materiaal en extra materiaal (=straatverlichting, ...) worden bijgehouden.
Voor de logica houden we blokken, wisselstraten en treinverbanden bij.
Als laatste houden we uiteraard bij waar we de intellibox kunnen vinden.

Voor elke reisweg in een wisselstraat wordt er een blok aangemaakt.
Voor elke wissel in de wisselstraat is er een entry.

Bij de turnoutstreet tabel geldt volgende regel: elke wisselstraat moet een enkel begin en een enkel einde hebben. Dus moeten verschillende wissels achtereen bij elkaar genomen worden. Omdat voor elke wisselstand een entry bestaat, is dit mogelijk. Dit wordt enkel meegegeven.

reserveringsprioriteit: eerste heeft hoogste, rangeren laagste, max 99

3 De normalisatie van de database of data-analyse

Enkele afspraken: Het aanduiden van keys:

- Primary key: **vet en onderstreept**
- Foreign key: onderstreept

3.1 Opsomming van gegevens (0NV)

project_name	locomotive_realspeed
screen_name	decodertype
railtype	locomotive_memo
railstatus	locomotive_changed
designboard_left	locomotive_forward
designboard_top	locomotive_function_activefn
designboard_address	locomotive_function_activefx
designboard_addres2	locomotive_function_function
designboard_blocknr1	locomotive_function_Funcfx
designboard_blocknr2	locomotive_function_name01
designboard_changed	locomotive_function_name02
designboard_inverce	locomotive_function_name03
designboard_inverc2	locomotive_function_name04
blocklinking_turnoutstreet	locomotive_function_name05
blocklinking_lastturnoutstreet	locomotive_function_name06
blocklinking_nextturnoutstreet	locomotive_function_name07
blocklinking_designboard	locomotive_function_name08
blocklinking_railstatus	locomotive_function_name09
block_type	locomotive_function_name10
block_name	locomotive_function_name11
block_number	locomotive_function_name12
turnoutstreet_name	locomotive_function_name13
turnoutstreet_status	locomotive_function_name14
turnoutstreet_number	locomotive_function_name15
turnoutstreet_changed	locomotive_function_name16
turnoutstreet_block	ibstatus_version
feedback_address	ibstatus_comport
feedback_turnoutstreet	ibstatus_baudrate
feedback_status	ibstatus_protocol
feedback_type	ibstaus_firmwareversion
feedback_changed	train_locomotive
feedback_function	train_trainstructure
ibcommand_command	train_changed
ibcommand_value	train_blockway
ibcommand_status	train_starttime
ibcommand_priority	train_stoptime
locomotive_name	wagon_name
locomotive_bitmap	wagon_trainstructure
decoder_address	wagon_type
virtual_address	wagon_bitmap
locomotive_speed	wagon_memo
locomotive_emergency	trainstructure_name
locomotive_minimumspeed	
locomotive_maximumspeed	

3.2 Het groeperen van gegevens die bij elkaar horen (1NV)

project: **project_id**, project_name

screen: **screen_id**, project_id, screen_name

designboard: **designboard_id**, screen_id, designboard_left, designboard_top, designboard_address, designboard_addres2, designboard_blocknr1, designboard_blocknr2, designboard_changed, designboard_inverce, designboard_railtype, designboard_railstatus

block: **block_id**, project_id, block_type, block_name, block_number

turnoutstreet: **turnoutstreet_id**, block_id, turnoutstreet_number, turnoutstreet_name, turnoutstreet_status, turnoutstreet_changed

blocklinking: **blocklinking_id**, turnoutstreet_id, pastturnoutstreet, nextturnoutstreet, designboard_id, railstatus

feedback: **feedback_id**, turnoutstreet_id, feedback_address, feedback_status, feedback_type, feedback_function, feedback_changed

ibcommand: **ibcommand_id**, ibcommand_command, ibcommand_value, ibcommand_status, ibcommand_priority

locomotive: **locomotive_id**, project_id, locomotive_name, locomotive_bitmap, decoder_address, virtual_address, locomotive_function_activefn, locomotive_function_activefx, locomotive_function_function, locomotive_function_Funcfx, locomotive_function_name01, locomotive_function_name02, locomotive_function_name03, locomotive_function_name04, locomotive_function_name05, locomotive_function_name06, locomotive_function_name07, locomotive_function_name08, locomotive_function_name09, locomotive_function_name10, locomotive_function_name11, locomotive_function_name12, locomotive_function_name13, locomotive_function_name14, locomotive_function_name15, locomotive_function_name16, locomotive_speed, locomotive_emergency, locomotive_minimumspeed, locomotive_maximumspeed, locomotive_realspeed, locomotive_memo, locomotive_changed, locomotive_forward, decodersteps, decodertype

ibstatus: **ibstatus_id**, project_id, ibstatus_version, ibstatus_comport, ibstatus_baudrate, ibstatus_protocol, ibstatus_firmwareversion

trainstructure: **trainstructure_id**, trainstructure_name

wagon: **wagon_id**, wagon_bitmap, wagon_memo, wagon_name

train: **train_id**, locomotive_id, trainstructure_id, train_changed, train_blockway, train_starttime, train_stoptime

3.3 Het verwijderen van dubbele gegevens (2NV)

3.3.1 De tabellen die nu worden gebruikt.

project: **project_id**, project_name
screen: **screen_id**, project_id, screen_name

boardrailtype: **boardrailtype_id**, boardrailtype_name
railstatus: **railstatus_id**, boardrailtype_id, railstatus_name
railtype: **railtype_id**, boardrailtype_id, railtype_name
designboard: **designboard_id**, screen_id, designboard_left, designboard_top, designboard_address, designboard_addres2, designboard_blocknr1, designboard_blocknr2, designboard_changed, designboard_inverce, designboard_inverc2, railtype_id, railstatus_id

decodertype: **decodertype_id**, decodertype_name
decoder: **decoder_id**, decodertype_id, project_id, decoder_address, decoder_virtual
function: **function_id**, function_name, function_address, function_virtualaddress, function_activefn, function_activefx, function_function, function_Funcfx, function_changed, function_name01, function_name02, function_name03, function_name04, function_name05, function_name06, function_name07, function_name08, function_name09, function_name10, function_name11, function_name12, function_name13, function_name14, function_name15, function_name16, function_status
locomotive: **locomotive_id**, locomotive_name, locomotive_address, locomotive_virtualaddress, locomotive_bitmap, function_id, locomotive_speed, locomotive_emergency, locomotive_minimumspeed, locomotive_maximumspeed, locomotive_realspeed, locomotive_memo, locomotive_changed, locomotive_forward

3.3.2 De tabellen die nog niet worden gebruikt.

blocktype: **blocktype_id**, blocktype_name

block: **block_id**, project_id, blocktype_id, block_name, block_number

turnoutstreetstatus: **turnoutstreetstatus_id**, turnoutstreetstatus_name

turnoutstreet: **turnoutstreet_id**, block_id, turnoutstreet_number, turnoutstreet_name, turnoutstreetstatus_id, turnoutstreet_changed

blocklinking: **blocklinking_id**, turnoutstreet_id, pastturnoutstreet, nextturnoutstreet, designboard_id, railstatus

feedbacktype: **feedbacktype_id**, feedbacktype_name

feedbackstatus: **feedbackstatus_id**, feedbackstatus_name

feedback: **feedback_id**, turnoutstreet_id, feedbackstatus_id, feedbacktype_id, feedback_address, feedback_function, feedback_changed

trainstructure: **trainstructure_id**, trainstructure_name

wagontype: wagontype_id, wagontype_name

wagon: **wagon_id**, project_id, trainstructure_id, wagontype_id, wagon_bitmap, wagon_memo, wagon_name

train: **train_id**, locomotive_id, trainstructure_id, train_changed, train_blockway, train_starttime, train_stoptime

ibcommand: **ibcommand_id**, ibcommand_command, ibcommand_value, ibcommand_status, ibcommand_priority

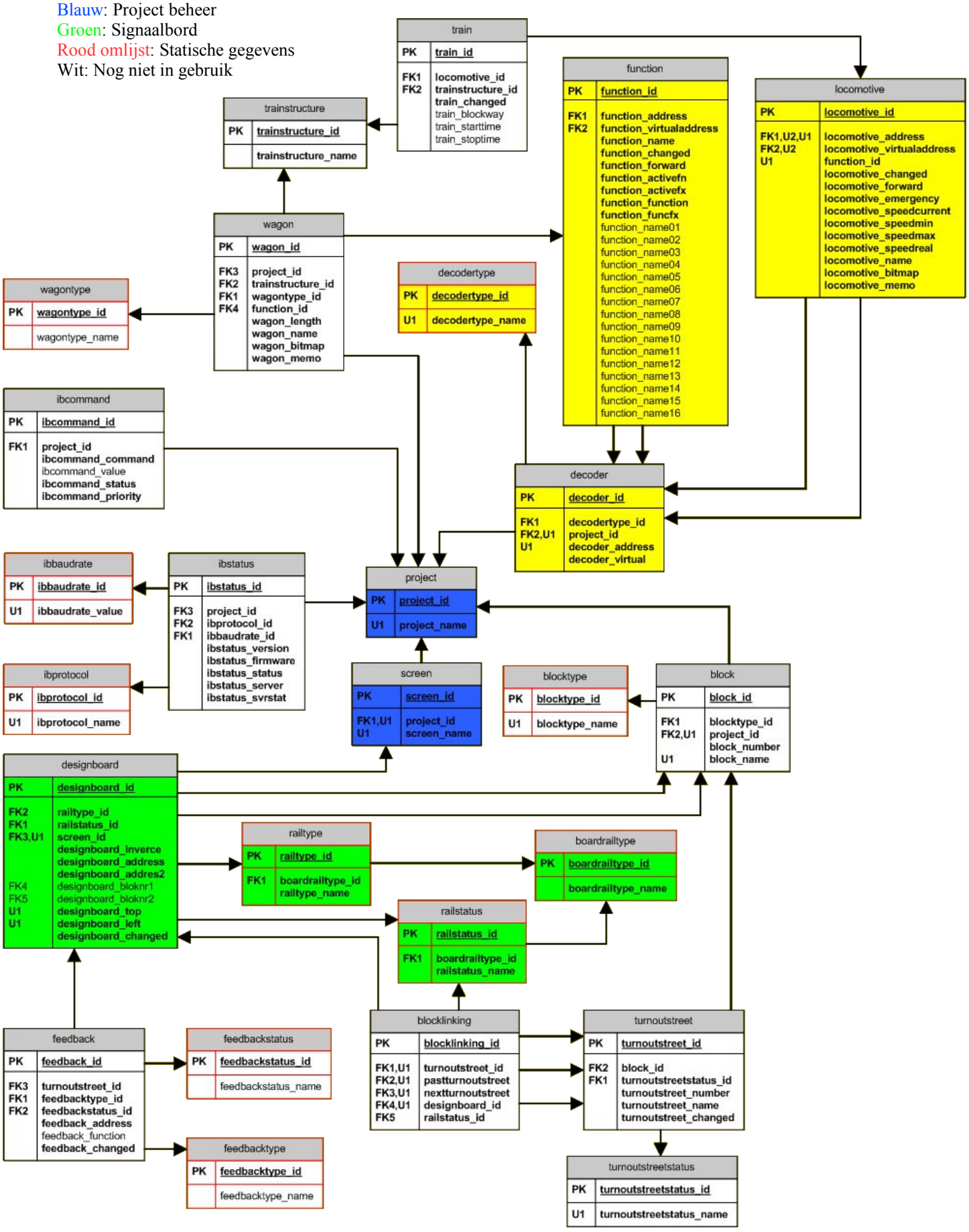
ibbaudrate: **ibbaudrate_id**, ibbaudrate_value

ibprotocol: **ibprotocol_id**, ibprotocol_name

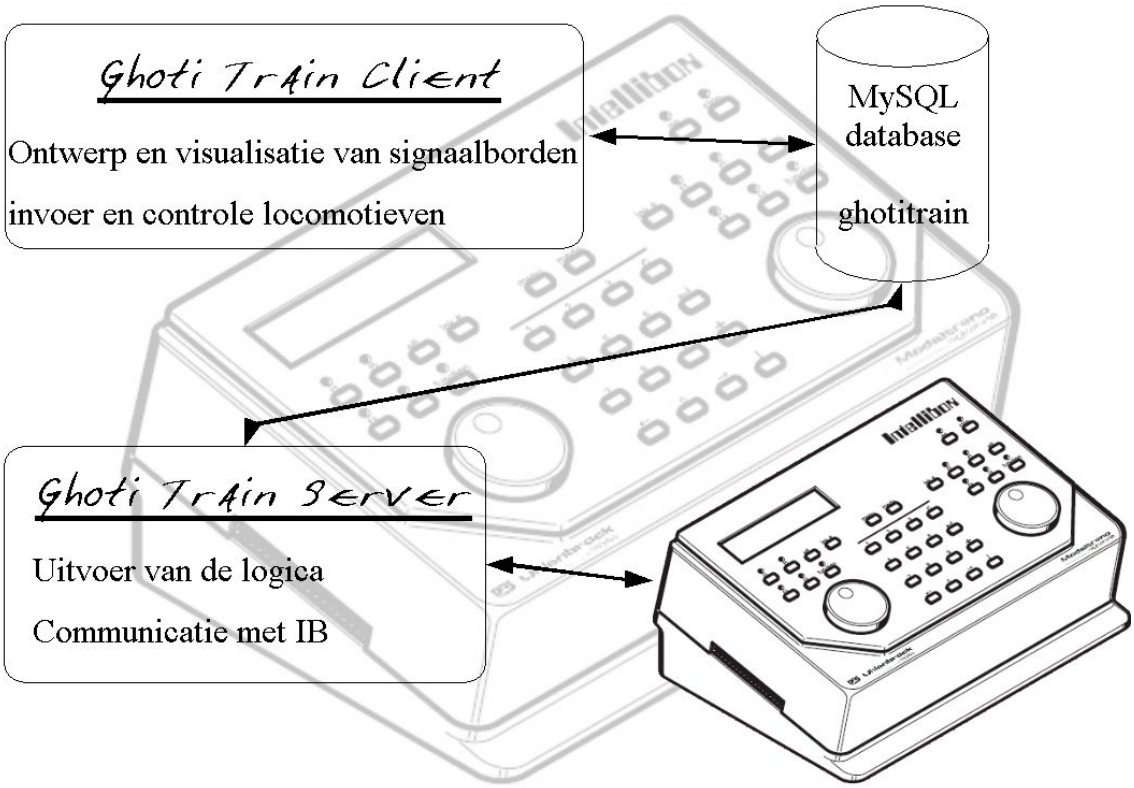
ibstatus: **ibstatus_id**, project_id, ibstatus_version, ibstatus_comport, ibbaudrate_id, ibprotocol_id, ibstatus_firmwareversion

3.4 Grafische voorstelling van de databank

Geel: Rollend materieel
Blauw: Project beheer
Groen: Signaalbord
Rood omlijst: Statische gegevens
Wit: Nog niet in gebruik



4 De werking van het programma.



Op dit schema zie je de algemene werking van het programma.
De client haalt zijn gegevens uit en stuurt ze naar de databank. Dit gaat enkel over de gegevens voor de visualisatie.
De server doet dit ook, maar dan enkel met de data die zijn veranderd en de events van de Intellibox. Dit kan de server 'zien' aan een 'changed' veld. De server communiceert ook met de Intellibox.

5 De werking van de server.

Voordat we hierop ingaan, een klein woordje uitleg over enkele belangrijke begrippen.

5.1 Klassen en objecten

Een klasse is een door de gebruiker gedefinieerd gegevenstype dat een status (de representatie) en een aantal bewerkingen (het gedrag) bevat. Een klasse heeft een aantal interne gegevens en methoden in de vorm van procedures of functies. Meestal wordt in een klasse een beschrijving gegeven van de algemene eigenschappen en het gedrag van een aantal gelijksoortige objecten.

Een object is een instantie van een klasse. Anders gezegd is een object een variabele van het gegevenstype dat als een klasse is gedefinieerd. Objecten zijn feitelijke entiteiten. Als het programma is gestart, maken objecten gebruik van het geheugen voor hun interne representatie.

De relatie tussen object en klasse is dezelfde als de relatie tussen resp. variabele en type.

Overerving.
Wanneer je een nieuwe klasse maakt, baseer je dit op een andere klasse of een basisklasse. Hierbij erven we de eigenschappen en het gedrag van deze 'moeder'-klasse over. Vandaar dat we spreken over 'overerving' bij het object-georiënteerd programmeren. Het grote voordeel hiervan is dat je niet opnieuw alles moet gaan programmeren. Je kan ook beslissen dat je een bepaald gedrag verandert. Hiervoor schrijf je een nieuwe methode, met de zelfde naam, zodat het origineel niet meer wordt uitgevoerd. Soms is het wenselijk dat dit toch gebeurt. Dit geef je dan aan door de vorige methode aan te roepen in uw eigen nieuwe methode.

5.2 Threads

Onder alle moderne besturingssystemen, zoals MacOS, LINUX, Windows, OS/2,... is het mogelijk om verschillende programma's 'tergelijkertijd' te draaien. Dit wordt multitasking genoemd.
In feite zal het besturingssysteem de beschikbare processortijd verdelen over de programma's. Ieder programma krijgt een aantal milliseconden toegewezen. Als deze op zijn of de bewerking is afgelopen, is het de beurt aan een ander programma. Is de bewerking nog niet afgelopen, dan wordt deze onherroepelijk onderbroken. Het voordeel is dat de tijd die niet wordt gebruikt door een programma, kan worden gebruikt door een ander programma.

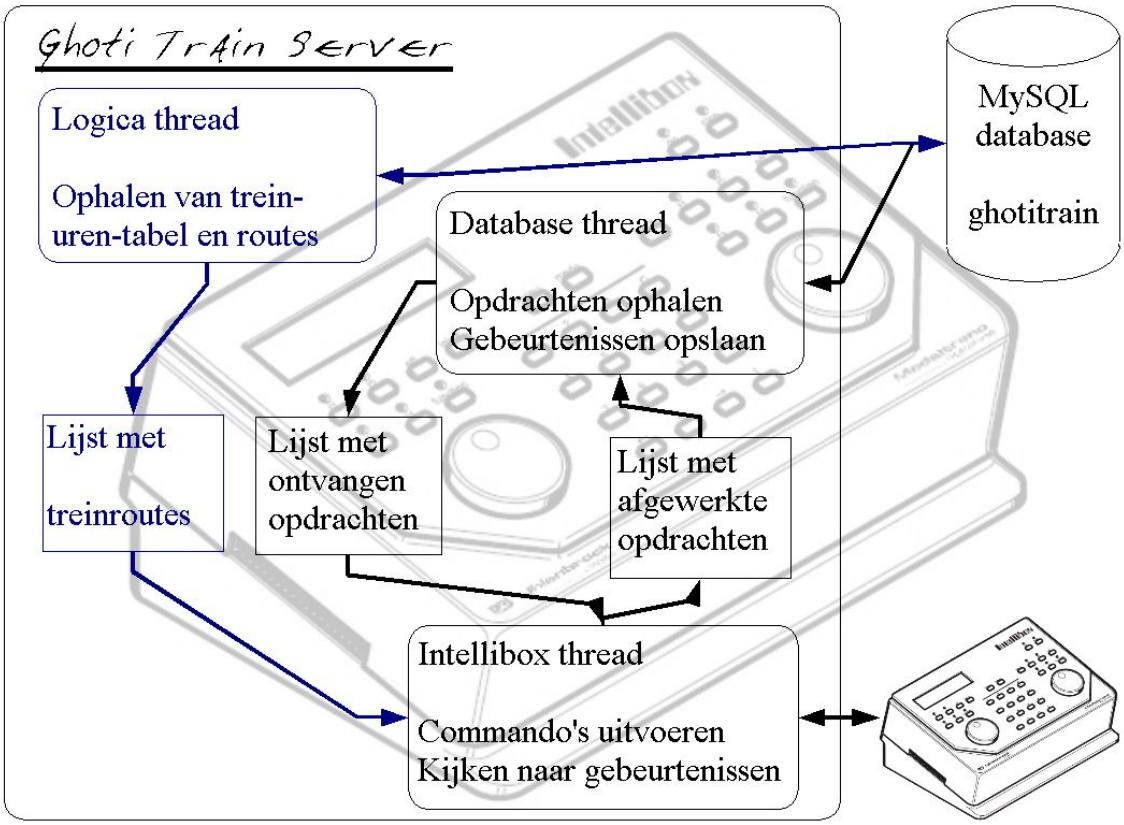
Door het gebruik van threads is het mogelijk om binnen een programma verschillende processen te laten lopen. Dit komt voor het besturingssysteem overeen met een ander programma, dus zal de code die hier wordt uitgevoerd, naast het eigenlijke programma worden uitgevoerd. Het voordeel van verschillende processen binnen één programma tegenover verschillende programma's, is dat je het geheugen veel eenvoudiger kan delen. Dit houdt wel in dat je ervoor moet zorgen dat deze geheugenlocaties niet ter zelfder tijd worden beschreven en/of gelezen. Hiervoor kan je cirtical sections gebruiken.

Praktisch gezien moet je een nieuwe klasse maken, afgeleid van tThread. Je 'overschrijft' de execute-methode en hierin zet je de code die moet worden uitgevoerd.

5.3 Critical Sections

Een 'critical section', of kritieke sectie, werkt als volgt: ergens in het geheugen wordt een plaatsje gereserveerd voor een getal. Deze plaats wordt opgevuld met een 0, maar als je een critical section binnen gaat (`criticalsection.enter`), dan wordt hier de waarde 1 bij opgeteld. Verder wordt er ook bijgehouden wie deze aanvraag heeft gedaan. Als het resultaat van de optelling 1 is, dus het is de eerste aanvraag, dan kunnen we door, anders wordt er gewacht tot de vorige aanvrager de cirtical section verlaat (`criticalsection.leave`). Dit weet de critical section doordat er 1 is afgetrokken van de vorige waarde.

5.4 Schema



Noot: het blauwe zal in de toekomst worden geïmplementeerd.

Hier zien we dat het programma in 7 delen wordt verdeeld, waarvan er 3 threads zijn.

- Het programma zelf (thread 1).

De gebruikersinterface heeft enkel als functie om alle instellingen te beheren en de server te starten.

- LogicaThread (wordt thread 4).

Deze moet nog worden geïmplementeerd. Deze thread zal de uurroosters en routes van de treinen uit de databank uitlezen, deze verwerken en als alles is goedgekeurd, de overeenkomstige commando's (= wissels en decoders) doorsturen.

- DatabaseThread (thread 2).

Hier wordt de databank aangesproken. Ik heb gekozen om dit in een appart proces te steken. De reden is vrij eenvoudig. Als de gebruiker iets wil nakijken op de gebruikersinterface, kan dit zonder enig probleem. Indien de databank op zich laat wachten, blijft de gebruikersinterface niet blokkeren.
De gebruikte componenten om MySQL aan te spreken, gebruiken zelf ook 2 threads.
Dit brengt het huidig aantal threads op 5.

- IntelliboxThread (thread 3).

De communicatie met de Intellibox heb ik ook in een apparte thread gestoken, zodat deze geen last ondervindt van acties met de databank. Verder zal de gebruikersinterface niet blokkeren als er op de Intellibox moet worden gewacht.
Er is echter 1 maar: als de Intellibox niet is aangesloten of geen stroom krijgt, dan zal het volledige programma toch blokkeren. Dit probleem zal er op een later tijdstip worden uitgehaald.

- Lijst met treinroutes.

Deze lijst zal dienen om de uurtabellen en routes in op te slaan.
Één record in deze tabel bestaat uit een trein, het start- en aankomstuur, het huidige blok en de bestemming + de berekende route.

- Lijst met ontvangen opdrachten.

Om de opdrachten van de DatabaseThread naar de IntelliboxThread door te geven, moeten we zorgen dat er geen conflicten ontstaan. Hiervoor heb ik deze en de volgende lijst in een apparte klasse gestoken, zodat er een critical section rondgebouwd kon worden.
Deze lijst bevat enkel de nog uit te voeren opdrachten.

- Lijst met afgewerkte opdrachten.

Deze lijst bevat de al uitgevoerde opdrachten en de events die de Intellibox doorzendt.

5.5 De communicatie met de intellibox.

Voor de volledige beschrijving van alle gangbare commando's verwijs ik naar Appendix B.

Alle commando's worden hexadecimaal weergegeven, tenzij anders aangegeven.
Vb: hC4.

5.5.1 De communicatie.

De connectie met de PC verloopt over een seriële RS232 verbinding.
De mogelijke baudrates zijn 2400, 4800, 9600, 19200 en 38400. Praktisch gezien heb je genoeg aan een 4800 verbinding. Bij hogere snelheden heb je een afgeschermde kabel nodig, want dan gaan er gegevens verloren en slaat de Intellibox tilt.
De gebruikte flowcontrol is RTS-CTS, de databits 8, de stopbits 2 en er wordt geen gebruik gemaakt van pariteit.

De intellibox kent meerdere communicatieprotocols, nl.
het Märklin Motorola P50 formaat en het door Uhlenbrock uitgebreidere P50x formaat. Deze kunnen door elkaar worden gebruikt.

5.5.2 Initialisatie.

Om te kunnen beginnen, moet de connectie worden geïnitialiseerd.
Deze initialisatie gaat als volgt:

1. Het leegmaken van de seriële buffer.
2. De poort instellen, met als beginwaarde 2400 voor de baudrate.
3. We sturen het blinde commando hC4, dat eveneens de eerste s88 module uitleest.

   (een s88 module ontvangt signalen vanop de modelbaan met schakelaars)
4. We lezen het aantal teruggezonden bytes in.
   Als er 2 bytes worden ontvangen, hebben we de Intellibox gevonden, sluiten we de poort en gaan we door naar stap 5
   Als er 1 byte wordt onvangen lezen we deze byte in.
   Als deze byte een 'D' bevat moeten we onmiddellijk stoppen, want de Intellibox staat in downloadmode. Dit wil zeggen dat hij wacht om een nieuwe firmware.
   Anders zetten we de 2 protocols op.
   Als er 0 of meer bytes worden ontvangen, hebben we geen goede baudrate gekozen, zetten we de baudrate een stap hoger en gaan terug naar stap 3.
5. Als we de Intellibox hebben gevonden, kunnen we doorgaan. Anders wordt de initialisatie stopgezet.
6. We openen de poort weer.
7. We sturen nu het caracter 'X' door, om te kunnen kijken of we wel degelijk de 2 protocols hebben aangezet. We herhalen dit totdat we data ontvangen.
8. We lezen het aantal ontvangen bytes uit.
   Als we 2 of 3 bytes terugkrijgen, dan zijn de 2 protocols ingeschakeld. We legen de buffer en lezen de in de Intellibox ingestelde baudrate uit.
   Als we 11 of meer bytes terugkrijgen dan stellen we de 2 protocols in en gaan terug naar stap 2.
   Tussen 3 en 11 bytes is het geen Intellibox dat we gevonden hebben.
   We sluiten de poort en stoppen de initialisatie.
9. We stellen de gevonden baudrate in en openen de poort.

5.5.3 Sturen van een commando.

Als we een commando gaan sturen, moeten we met het volgende rekening houden:

1. Is het een P50 of P50x commando?
2. Is het een ASCII of een binair commando?

Als het een P50x commando is, dan zenden we eerst een 'x', gevolgd door het volledige commando en afgesloten met karakter 13. Dit is het Enter karakter. Als je een LineFeed mee doorzendt, dan wis je het commando. Hoofd- of kleine letters spelen geen rol.

De gebruikte commando's zijn: (tussen haakjes is verplicht, vierkante haakjes moet niet worden meegegeven.

- x kijken of het uitgebreide P50x protocol staat ingesteld.
- hA4 Uitlezen van een Special Option (SO)
- xB[baud] De intellibox naar de meegegeven baudrate forceren.
- xL(adres),[speed],[dir],[fn],[f1],[f2],[f3],[f4]
  Een locomotief aansturen.
  adres is het decoder adres;
  speed is de decoderstap: 0 is stop, 1 is noodstop, 2 tot 127 de snelheid.
  Dit is decoderafhankelijk.
  dir is de richting.
  fn zegt of de verlichting aan moet.
  f1 tot f4 zijn de extra functies die de decoder kan bezitten.
- xF(adres),[f1],[f2],[f3],[f4],[f5],[f6],[f7],[f8]
  Een functiedecoder aansturen.
  adres is het decoderadres. Dit kan ook een locomotiefdecoder zijn.
  f1 tot f8 zijn al de mogelijke aangesloten functies.
- xT(adres),(kleur),(status)
  Een magneet-artikel aansturen. Een wissel is bv. een magneet-artikel.
  adres is het decoderadres.
  kleur is de te schakelen uitgang. R is rood, G is groen.
  status laat de Intellibox weten of deze uitgang stroom moet krijgen of juist niet.
  Dit moet zo omdat een wissel ongeveer 150 milliseconden moet bekrachtigen,
  maar een ontkoppelrail moet minstens 1 seconde actief zijn.

Dit zijn de belangrijkste events, die nu volledig zijn geïmplementeerd.

5.5.4 Events ophalen.

Als er geen commando's voor handen zijn, kan de IntelliboxThread events van de Intellibox ophalen. Het commando hiervoor is hC8 en is een binair commando.

Het antwoord wordt als volgt ontleed:

- Bit 8 van de eerste byte is 1: er volgt nog een byte.
- Bit 8 van de tweede byte is 1: er volgt nog een byte.
- Maximum 3 bytes als antwoord.

De betekenis van deze bytes is als volgt:

byte 1

8	*7*	*6*	*5*	*4*	*3*	*2*	*1*
NEXT	LocoNet	TURN	PC	PWR	s88	IR	LOC

LOC: er is een locomotief bediend maar niet door een PC.
IR: er is een infrarood-bediening gebruikt.
s88: er is een sensor bediend geweest.
PWR: de stroom naar de rails staat af.
PC: er wordt geprobeerd iets te schakelen dat door de PC werd gereserveerd.
TURN: er is een magneet-artikel geschakeld, maar niet door de PC.
LocoNet: niet gedocumenteerd.
NEXT: er zijn nog events.

byte 2

8	*7*	*6*	*5*	*4*	*3*	*2*	*1*
NEXT	IBstatus	Overheat	Contact	Overload	IB	LokMaus	BOOST

BOOST: kortsluiting op een externe booster.
LokMaus: kortsluiting op de lokMaus bus.
IB: kortsluiting op de rails die op de Intellibox zijn aangesloten.
Overload: overbelasting op de programmeerrail.
Contact: fout elektrisch contact tussen de programmeerrail en de gewone rails.
Overheat: de Intellibox is oververhit.
Ibstatus: de status van de intellibox is veranderd.
NEXT: er zijn nog events.

byte 3:

8	*7*	*6*	*5*	*4*	*3*	*2*	*1*
NEXT	N/A	N/A	N/A	N/A	N/A	RS232	PT

PT: er is een programmeer-event.
RS232: de buffer van de seriële poort zit vol.
N/A: nog niet voorzien in de Intellibox.
NEXT: nog niet voorzien in de Intellibox.

Voor alle events in deze bytes moet je verder kijken. Ik ga er 2 bespreken, nl. deze die al zijn geïmplementerd.

- Het locomotive-event hC9, dit is een binair commando,
  en geeft maximum 119 locomotieven terug.

Byte 1: de snelheid.
In deze byte wordt de huidige decoderstap doorgegeven. Dit is van 0 tot 127.
Als deze byte de waarde 128 bevat, dan is er geen data aanwezig. Er zijn dus geen loc-events meer.
Boven de 128 wordt niet gebruikt.

Byte 2: de functies F1 tot F8
Bit 1 voor F1, Bit 2 voor F2 enzovoort.

Byte 3: lage bits van het decoderadres.
Eerste deel van het adres van de locomotief: A7 tot A0

Byte 4: hoge bits van het decoderadres, de richting en de status van de verlichting

8	***7***	***6***	***5***	***4***	***3***	***2***	***1***
DIR	FL	A13	A12	A11	A10	A9	A8

A13 tot A8: Plak deze bits achter de vorige byte. Tesamen heb je het adres.
FL: De status van de verlichting van de locomotief.
DIR: Richting van de locomotief.

- Het turnout-event hCA, dit is een binair commando,
  en geeft maximum 32 magneet-artikelen terug.

Byte 1: Het aantal magneet-artikelen die zijn veranderd, maar niet door de PC.
Per magneet-artikel worden er 2 bytes doorgestuurd.
De eerste byte bevat de lage bits van het decoderadres: A7 tot A0.
De tweede byte bevat de hoge bits van het decoderadres en de actieve kleur:

8	***7***	***6***	***5***	***4***	***3***	***2***	***1***
kleur	N/A	N/A	N/A	N/A	A10	A9	A8

A10 tot A8: Plak deze bits achter de vorige byte. Tesamen heb je het adres.
kleur: Welke uitgang is er actief.

6 De code van de client

6.1 De opbouw van het scherm.

Ik heb gekozen om verschillende componenten te schrijven voor elk soort rail.

 De rail-componenten.

Van links naar rechts: gewone rails, wissels, kruisingen, driewegwissels, ontkoppelrails, speciale functies (lamp) en contactrails.
Deze componenten werken allemaal op hetzelfde principe. Het grote verschil zit hem in wat er getekend wordt. Enkele specifieke verschillen in wat er op het scherm komt en de afmetingen, wat het gedrag is bij het klikken en hoe de status moet worden opgeslagen.

- Alle rails, zonder uitzondering, moeten het 'databank ID' onthouden.
- Alle rails, behalve de lamp, hebben een bloknummer.
  Met deze eigenschap weet het programma tot welke groep rails het behoort.
  De gewone rail heeft 2 bloknummers.
  Het tweede bloknummer is enkel nodig als je een 'dubbel schuin rail' tekent.
  (Zie 7.3 Een signaalbord tekenen.)
- Ook hebben ze allemaal, behalve de gewone rail, een adres. Dit adres komt uiteraard overeen met het decoder adres op de modelbaan. Enkel de driewegwissel heeft 2 adressen. Dit zijn nl. 2 samengestelde wissels, die elk een andere kant uitgaan.
- Verder moet de status van de rails kunnen worden aangepast.
- Enkel de wissel, kruising en driewegwissel componenten hebben een inverce-eigenschap, die aangeeft dat de rail op de modelbaan 'verkeerd' is aangesloten. Dit kan ook worden gebruikt om bv. rails die in bochten liggen juist te laten schakelen tegenover het scherm.
- Tot slot moet er bij het klikken op de rail het volgende gebeuren.
  Met uitzondering van de contactrail, zal elke rail tijdens het rijden van status veranderen en deze naar de databank doorsturen. Op het scherm gebeurt er niets. Als het hoofdscherm het component aanspreekt, omdat er in de databank staat aangegeven dat de status is veranderd, dan pas wordt het scherm vernieuwd.
  Elk component zal, bij het tekenen van een signaalbord, een menuutje laten zien, ten minste als je er met de rechter muistoets op klikt. Hiermee kan je een rail wissen of de eigenschappen aanpassen.

 De componenten om een signaalbord te tekenen.

Van links naar rechts: pallet, clipbord en tekenbord.

- Het pallet laat alle mogelijk te tekenen rails zien. Deze kunnen aangeklikt worden.
  In '7.3 een signaalbord tekenen' vind je een beschrijving hoe dit werkt.
  Als je een rail aanklikt, wordt dit opgeslagen in het clipbord.
- Het clipbord bevat een lijst met alle mogelijke rails en hun eigenschappen.
  Als er een rail wordt aan toegewezen, wordt de eventuele vorige rail eerst gewist.
  Deze is nu klaar om op het tekenbord te worden geplaatst.
- Het tekenbord haalt zijn gegevens uit het clipbord. Aan de hand van de muiscoördinaten wordt berekend waar precies het nieuwe component moet worden getekend. Ook kan je een grid laten zien. De punten hierop komen overeen met het punt dat als oorsprong dient om een railcomponent te tekenen. Het is mogelijk om dit scherm in te zoomen.
  Dit component bevat zelf een create methode die de rails tekent.

 De locomotieflijst.

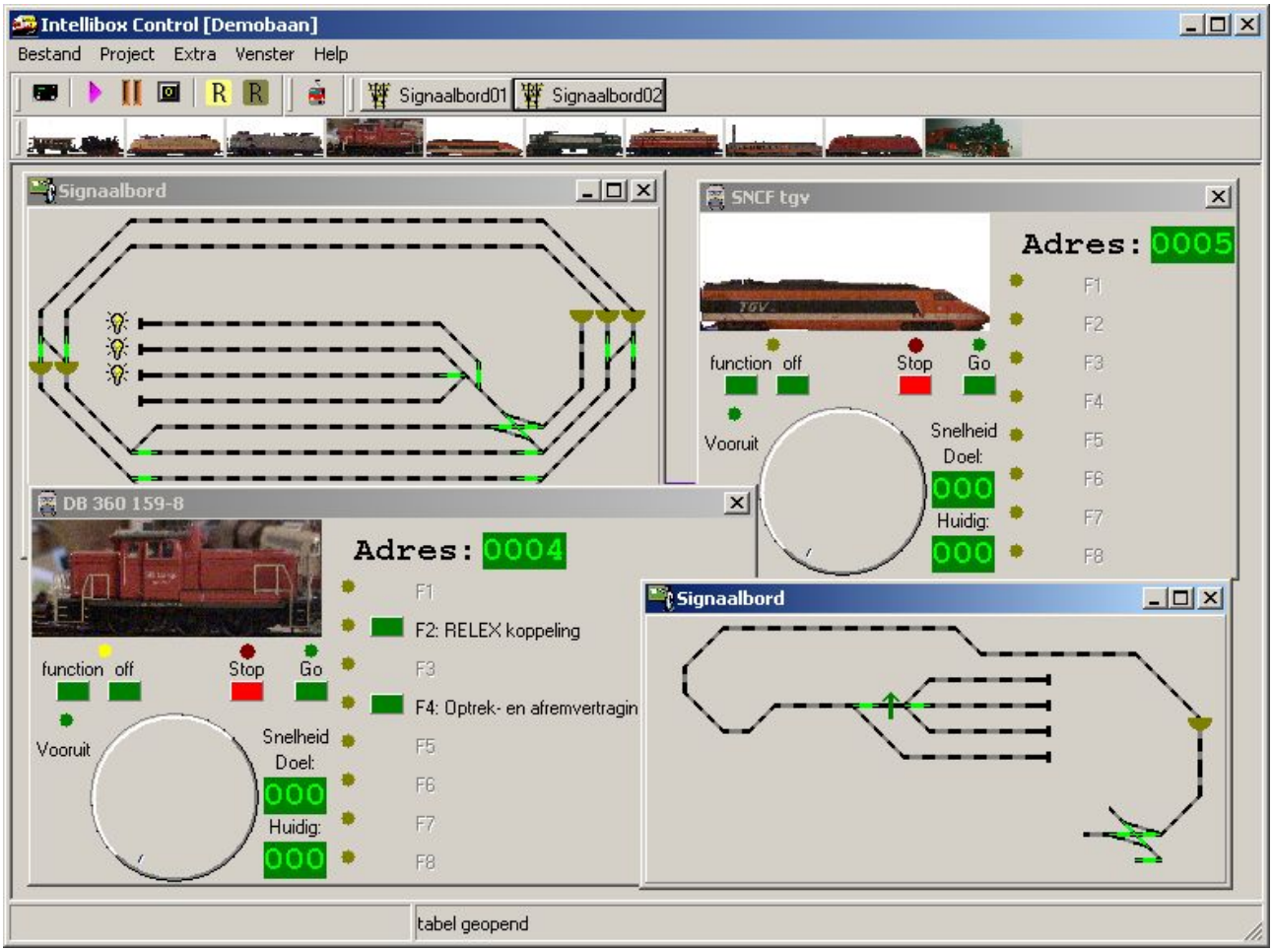
Dit component laat een overzicht zien van alle locomotieven in het actieve project. Als je hier op een locomotief klikt, dan zal dit een nieuw scherm openen dat een locomotiefcontrol bevat. Zie '7.5 bedienen van signaalborden en locomotieven' voor een uitgebreide beschrijving.

6.2 Het updaten van de zichtbare schermen.

Er is een algemene routine die, telkens er een bepaald aantal ingestelde milliseconden verstrijken, wordt gestart. Deze routine gaat in de databank gaan kijken naar veranderde rails, blokken en locomotieven, die wachten om op het scherm te worden vernieuwd. Dit weet hij door naar de '_changed' te kijken. Deze moet 9 als waarde hebben.
Vervolgens loopt deze routine alle aanwezige schermen af. Binnen deze schermen zal de routine elke aanwezige component bekijken. Als de overeenkomstige component gevonden is, wordt de status aangepast. De component zelf gaat dan in de databank aangeven dat het is vernieuwd, door de '_changed' terug op 0 te zetten.

7 Het gebruik van het programma.

In het start-menu van Windows, onder programma's of programs, vind je de groep ghoti. Hier ga je naar Ghoti Train, waar je Ghoti Train Client vindt en opstart.



Een vlugge beschrijving van de opbouw van het scherm:

- Bovenaan heb je de titelbalk, zoals bij elk programma.
- Daaronder heb je het menu.
- Nog daaronder heb je een balk met verschillende knoppen.
  Deze knoppen zijn momenteel nog niet uitgewerkt,
  maar zullen de Intellibox doen starten, stoppen of resetten.
- De laatste rij is de locomotieflijst. Deze lijst bevat elke locomotief die in het actieve project is voorzien.
- In het midden krijg je alle bedieningsschermen te zien.
- Onderaan heb je de statusbalk.

7.1 De client instellen.

Als de client de eerste keer wordt gestart, wordt het instellingen-scherm getoond.
Controleer hier dat alle instellingen juist zijn.
Deze zijn ook aanpasbaar in het ini-bestand (zie ook 2.8).

Het tab-blad 'Database':

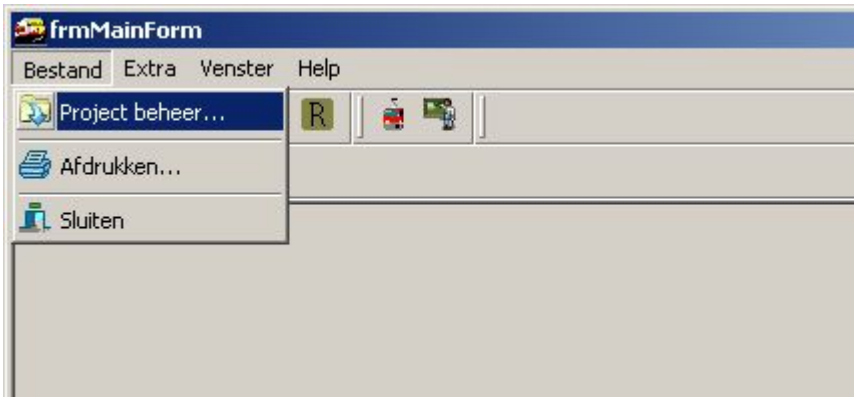
Server naam:	De naam van de PC waarop MySQL staat. Localhost staat voor dezelfde PC als waar de client staat.
Gebruikersnaam:	De gebruikersnaam van MySQL. Standaard is dit root.
Paswoord:	Het paswoord van deze gebruiker.
Unix socket:	De socket-naam of nummer van de MySQL server.
Database naam:	De naam van de database van Ghoti Train. Enkel aan te passen als u de database manueel heeft geïnstalleerd en de database-naam heeft aangepast.
Poort:	De TCP-IP poort van de MySQL-server.

In het tab-blad 'Algemeen' vind je het volgende:

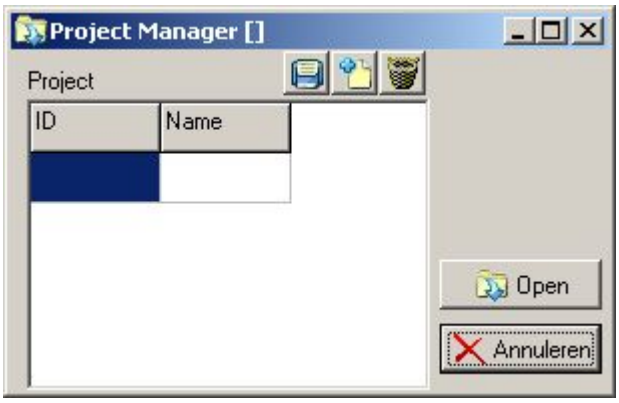
Foto map:	De map die de foto's bevat. Als je foto's kiest die hier niet staan, worden ze naar hier gekopiëerd.
Project:	Het actieve project. Het is noodzakelijk dat u eerst de 'database'-tab instelt.
Interval:	Geeft het tijdsinterval in milliseconden op om het scherm te vernieuwen.
Märklin knop:	De snelheidsknop bij de locomotief-controle instellen. (zie 2.6)

7.2 Een project en een signaalbord creëren.

Als de client goed is ingesteld, kunnen we nu een project en enkele signaalborden creëren.
Je gaat naar 'bestand' ⇒ 'Project beheer'...



Je krijgt het volgende scherm:



Hier staan 3 belangrijke ikoontjes.

 Opslaan

 Nieuw

 Wissen

Om een nieuw project te maken, kies je dus het middenste.
Het eerste dient om een naamswijziging op te slaan en het laatste om een project te wissen.

Druk dus op het middenste knopje:

Hier kan je een naam opgeven. Vb: Demobaan.

Druk op 'OK' , kies het project, zodat de naam ervan in de titelbalk verschijnt en druk op 'Open'.

Je ziet dat de project-naam bovenaan in de titelbalk te voorschijn komt.
Nu gaan we een signaalbord aanmaken. Kies 'Project' ⇒ 'Nieuw' ⇒ 'Signaalbord'.

In het volgende scherm geef je een naam op. Vb: Signaalbord01

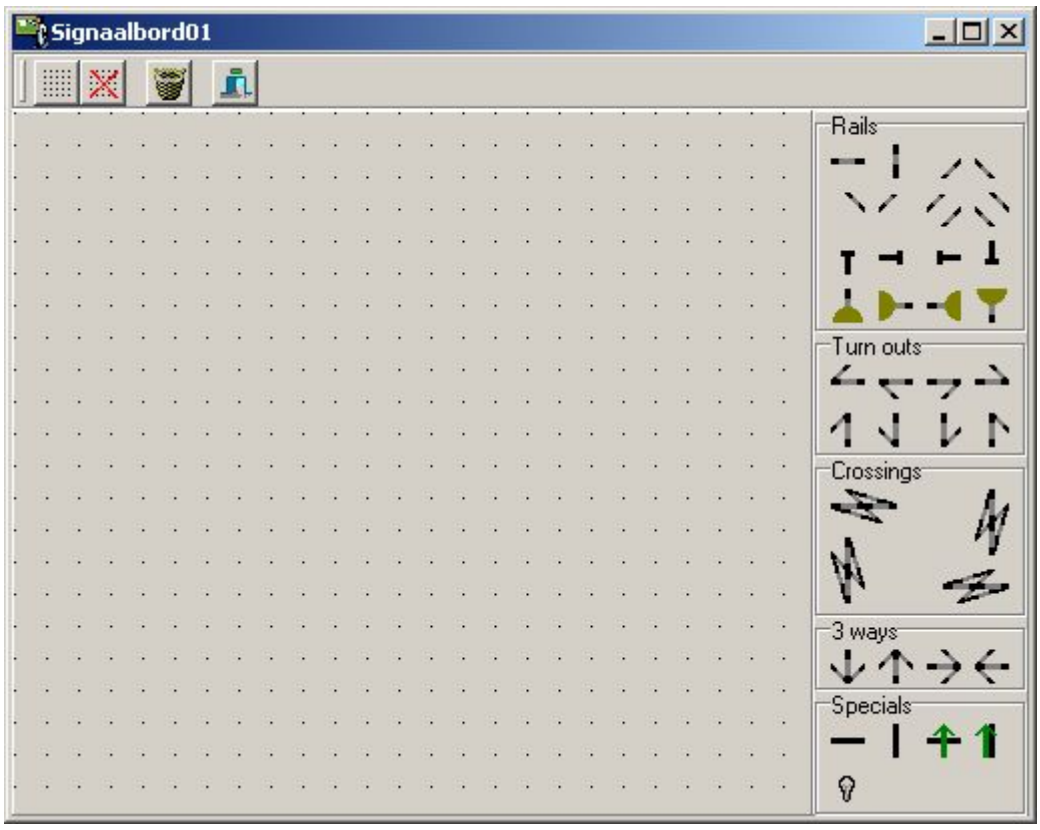
Druk op OK en je bent klaar om het signaalbord te gaan tekenen!
Er verschijnen nu op verschillende plaatsen nieuwe menu-items:

'Venster' ⇒ 'Signaalbord' ⇒ 'Signaalbord01';
'Project' ⇒ 'Aanpassen' ⇒ 'Signaalbord' ⇒ 'Signaalbord01';
Een nieuwe knoppenbalk met 'Signaalbord01'.

7.3 Een signaalbord tekenen.

Nu gaan we over tot het tekenen van een signaalbord.
Ga naar 'Project' ⇒ 'Aanpassen' ⇒ 'Signaalbord' ⇒ 'Signaalbord01'.

Je krijgt nu volgend scherm



Het scherm is in 3 gedeeld.
- Bovenaan vind je 4 icoontjes.
- Rechts kun je de rails kiezen.
- Links is het tekenbord.

7.3.1 De icoontjes.


Toon grid.


Verberg grid.


Wis het volledige scherm.

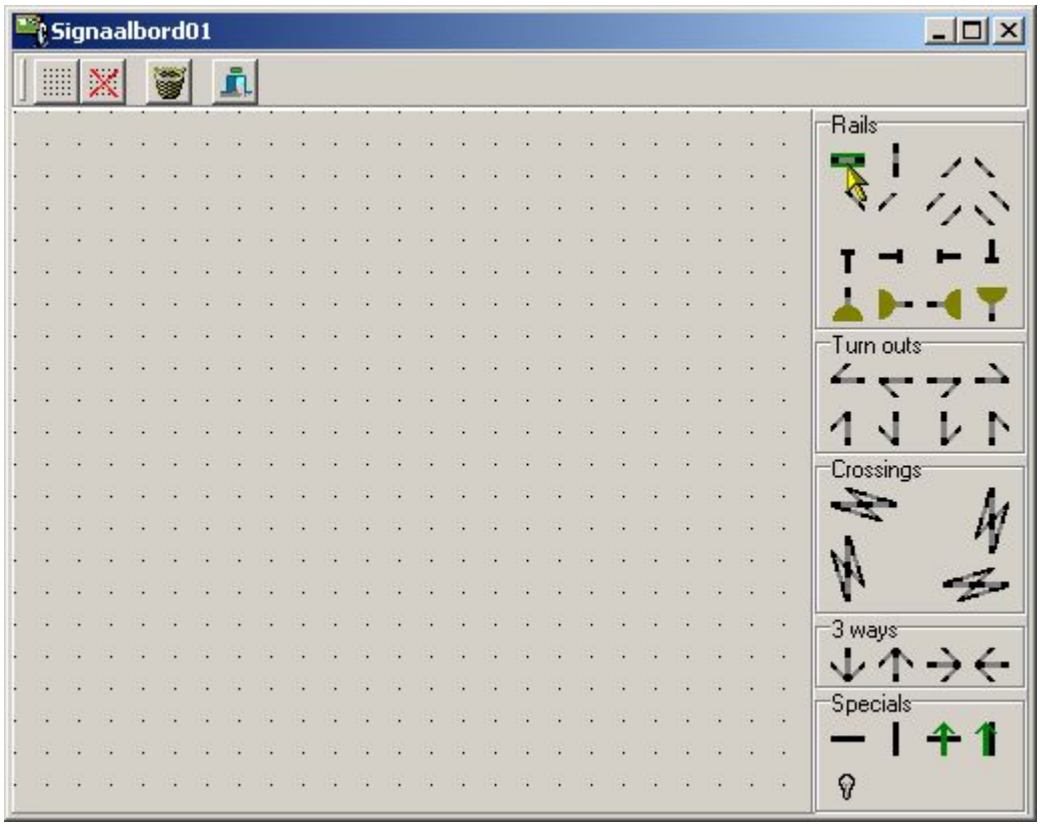

Sluit het tekenbord.

7.3.2 Rails

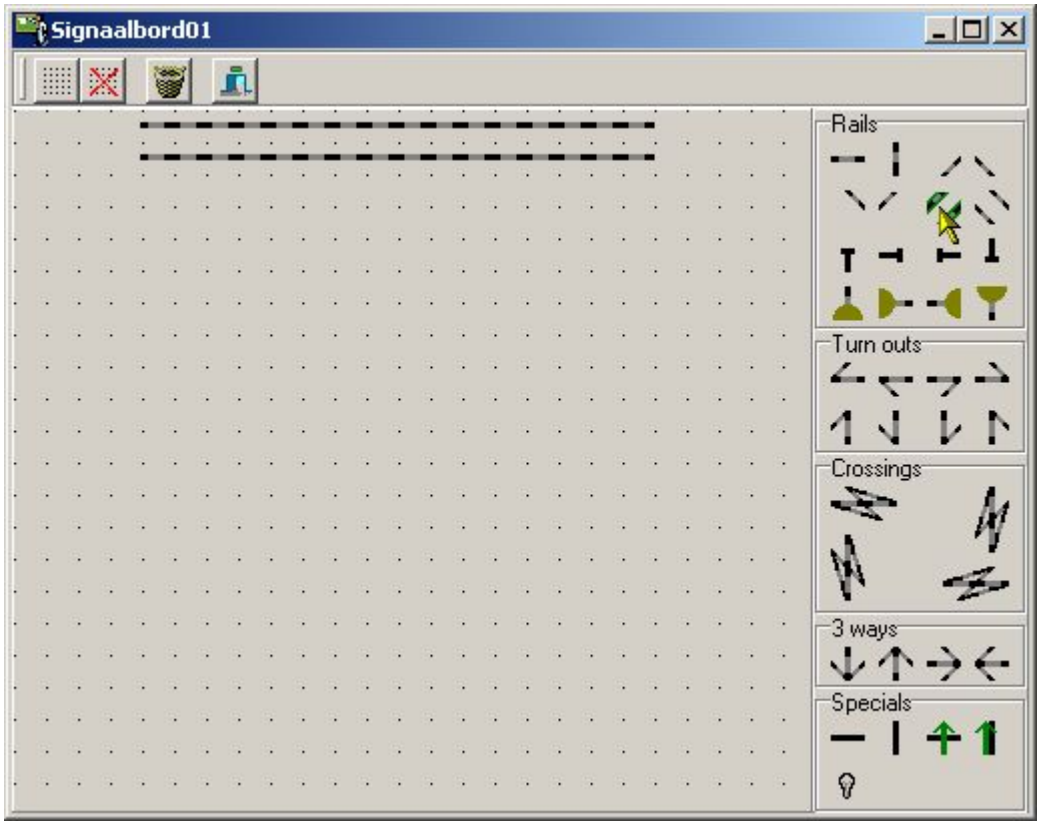
Dit gedeelte bestaat uit 5 groepen.
- De rails
  Dit is de verzameling van gewone rechte en schuine rails, plus eindblokken en tunnels.
- Wissels
  Alle wissels kan je hier vinden.
- Kruisingen
  De kruisingen nemen twee vakjes in. Dit zie je hier.
- 3-weg wissels
  Deze plaatsbesparende wissels kun je ook gebruiken.
- Speciale rails
  Alle overige rails (momenteel contact-, ontkoppelrails en een lamp) staan hier.

7.3.3 Tekenen

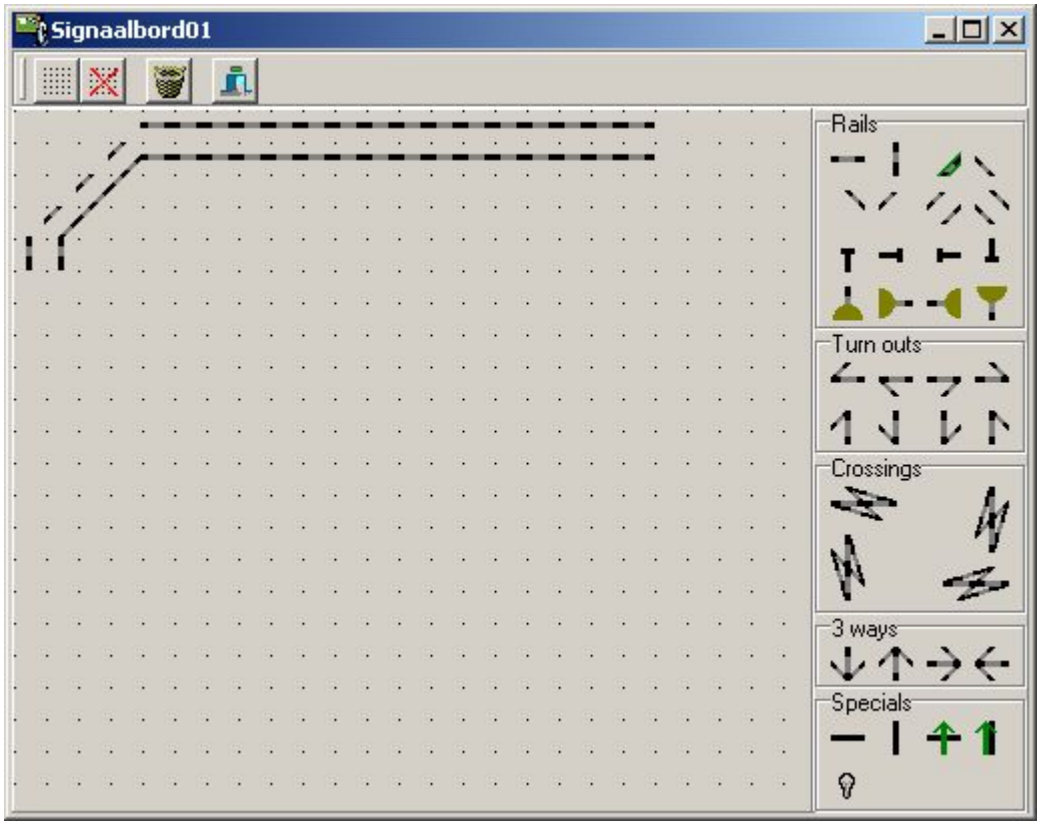
Door een rail aan te duiden, verandert de status. Klik maar een rail aan.



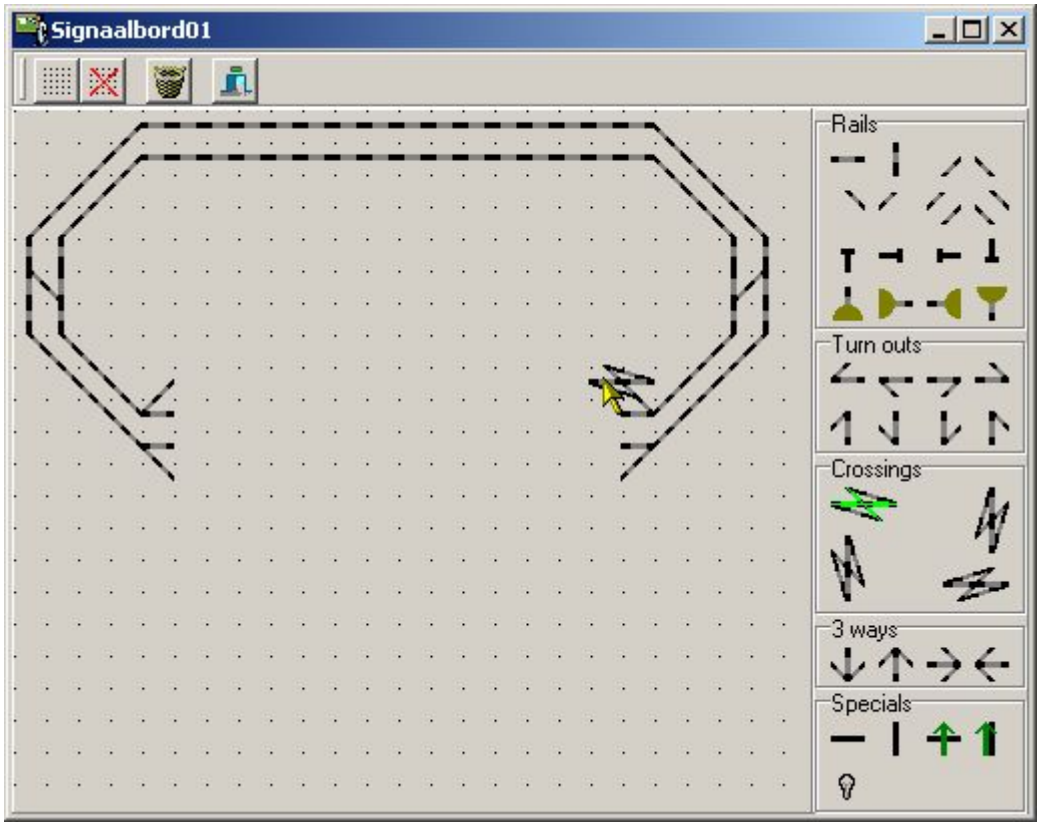
Klink nu in het tekenbord waar je de gekozen rail wenst. Een lijn, zoals hieronder, teken je rail per rail. Duidt nu de eerste dubbele schuine rail aan.



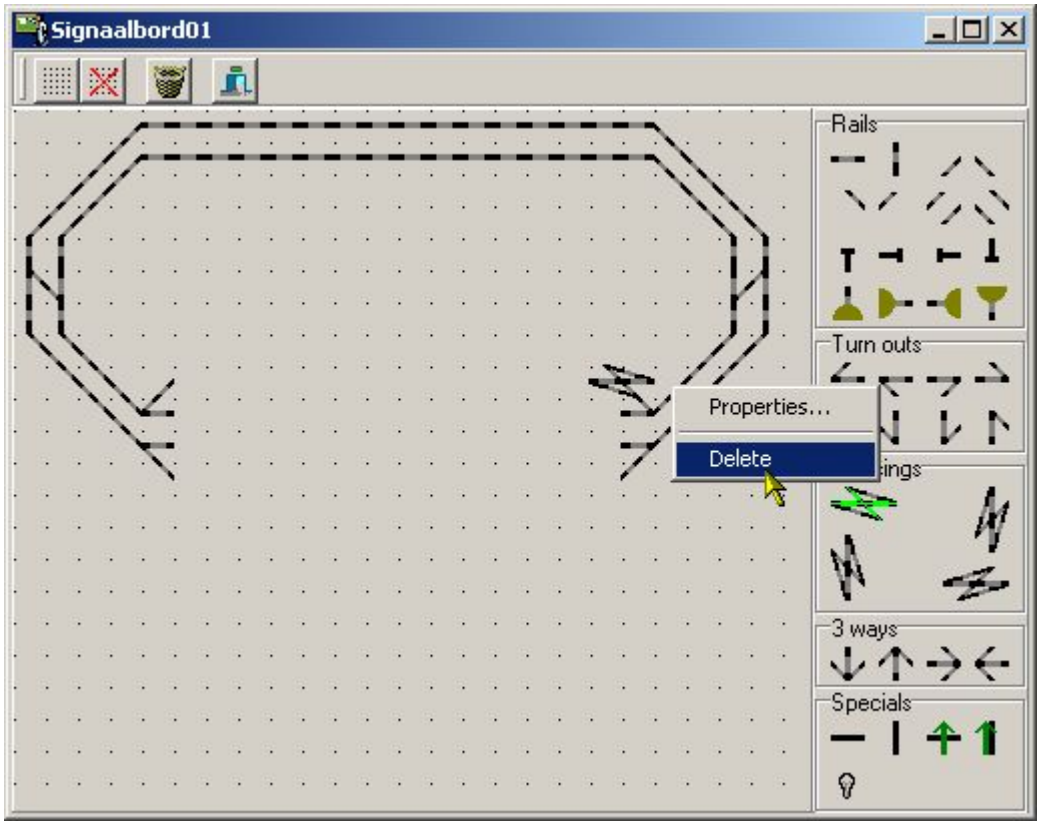
Hieronder zie je dat de onderliggende openingen al zijn opgevuld met 2 enkele schuine rails. De 4 overige (bovenste) gaten zullen nu worden ingevuld.



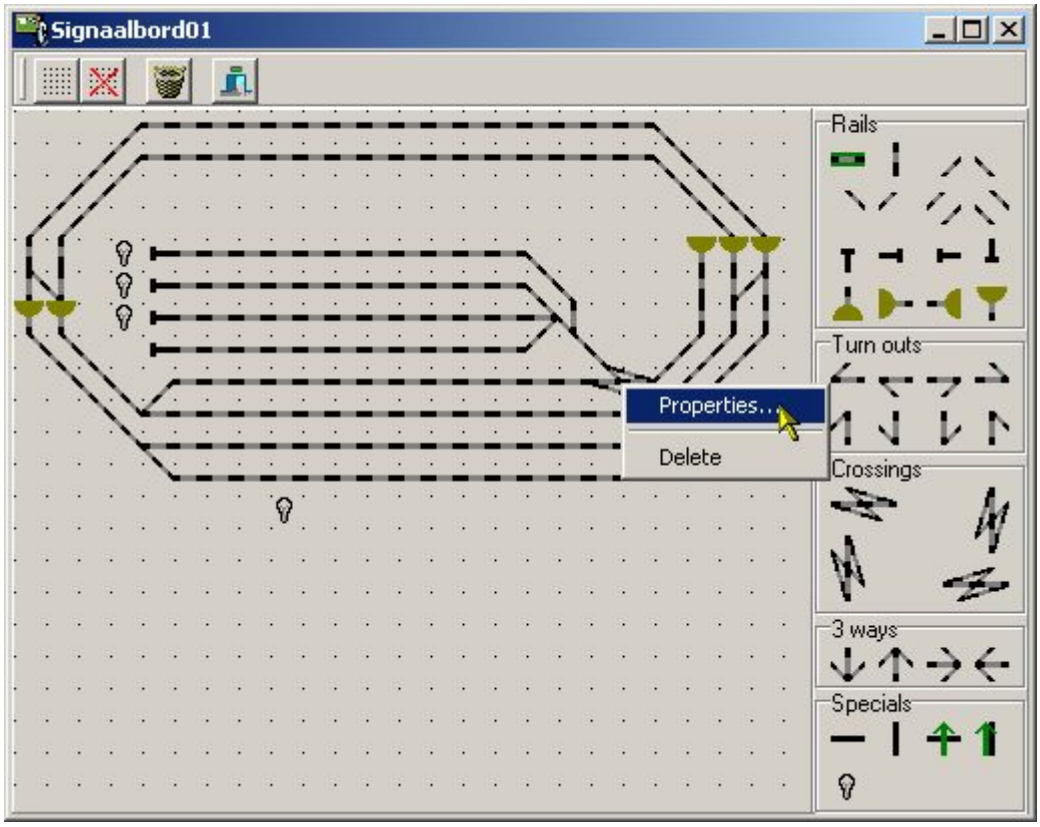
Hier hebben we enkele wissels getekend, plus een kruising. De muiscursor staat nog waar de kruising getekend werd. We zien achter de kruising een volgend probleem ontstaan. Het blok wordt al gebruikt door een enkele schuine rail.



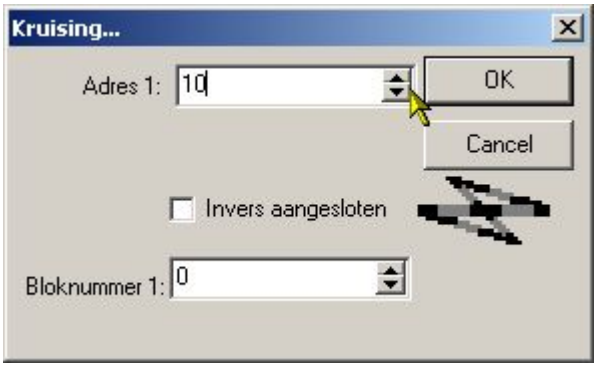
We wissen deze schuine rail en vervangen deze met een dubbele schuine rail.
Druk met de rechter muistoets op de rail die weg moet en kies 'Delete'.



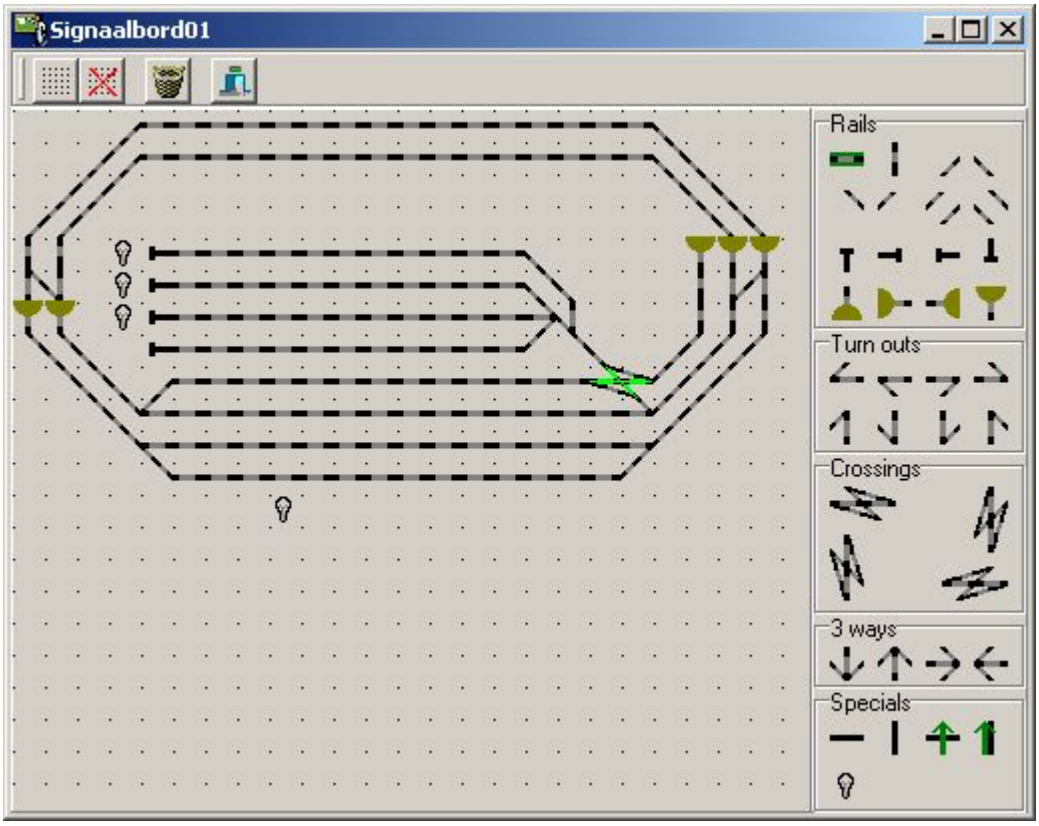
Eens alles getekend, kunnen we de wissels voorzien van adressen.
Klik hiervoor met de rechter muistoets op een wissel en kies 'Properties...'. Hier hebben we de kruising genomen.



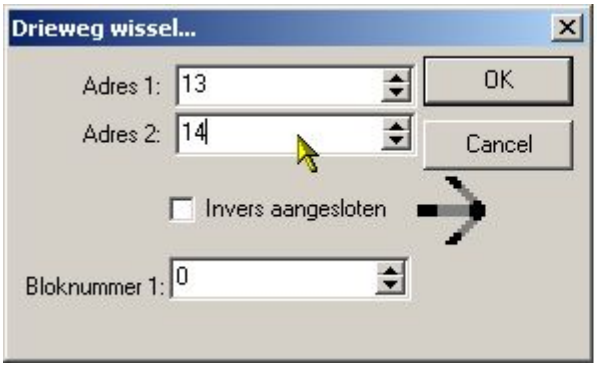
We geven het adres in. We duiden aan dat deze eventueel invers aangesloten is en het bloknummer (Nog in ontwikkeling).

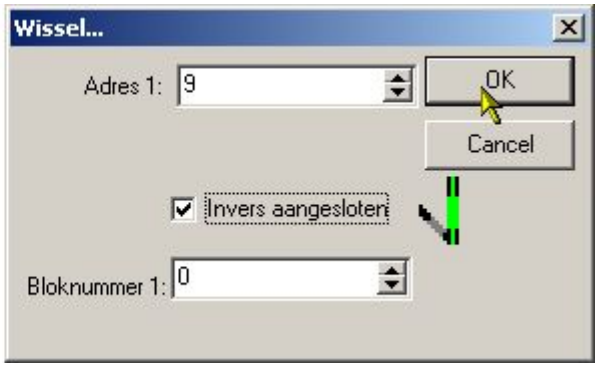


We zien dat de kruising nu een weg aanduidt, namelijk rechtdoor.

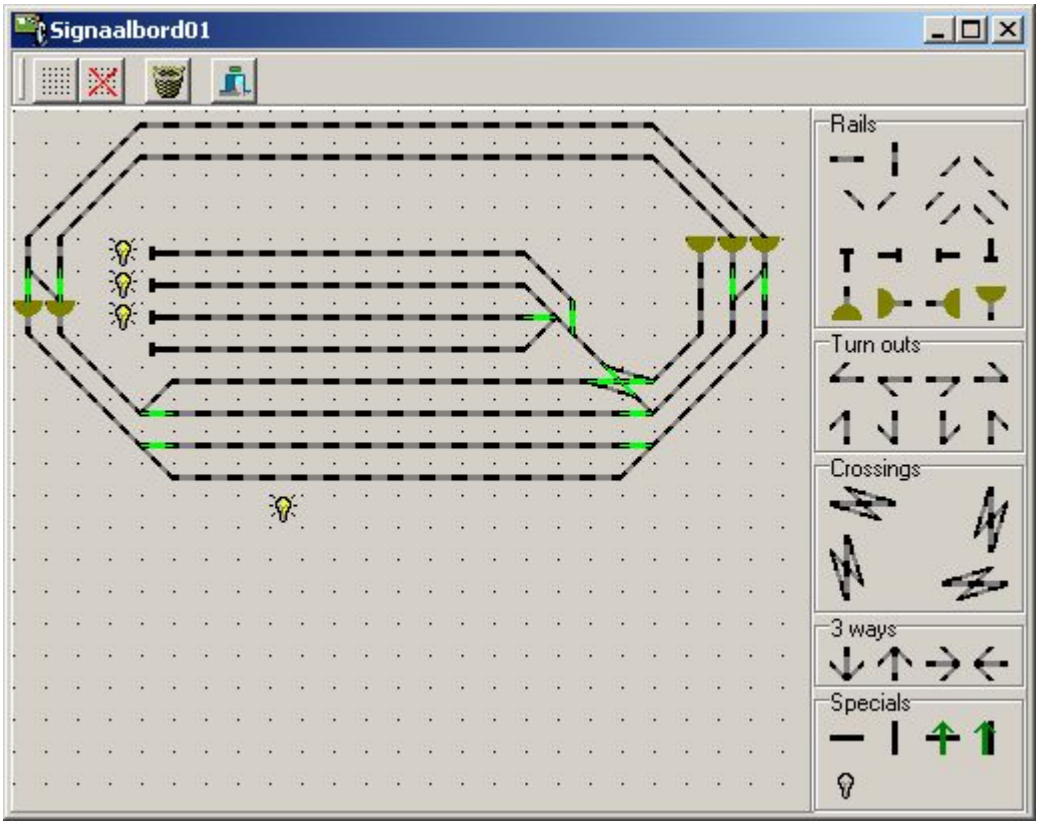


We geven alle parameters mee aan elke wissel.
Hieronder vindt u een voorbeeld voor een 3-weg wissel en de wissel ervoor (deze moet geïnverteerd worden).





Zo, afgewerkt. Druk nu op de sluiten-knop.



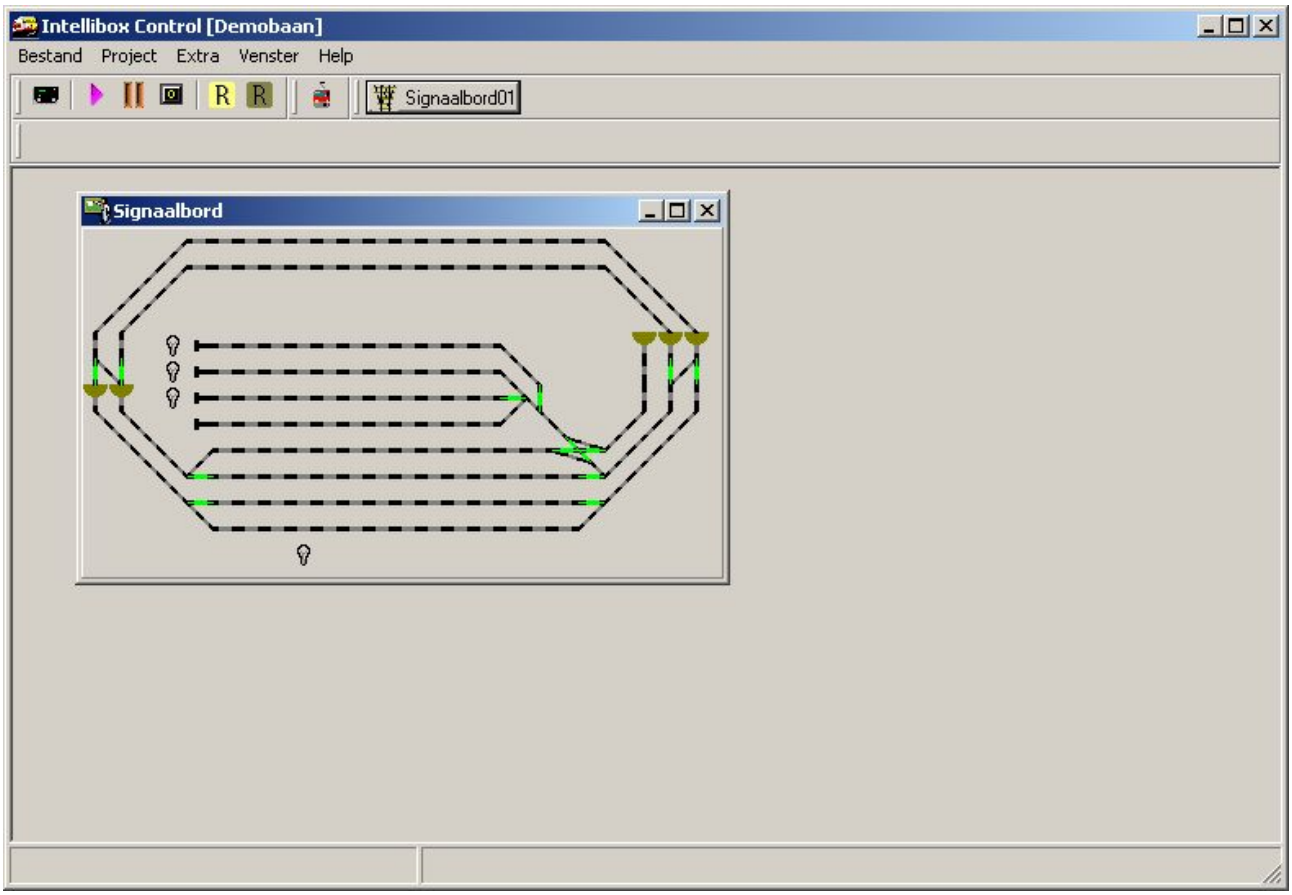
Uiteraard klikken we op 'Ja' of 'Yes'.

Het tekenen is nu afgelopen. Nu kunnen we het scherm tonen en uittesten.

7.3.4 Het signaalbord testen.

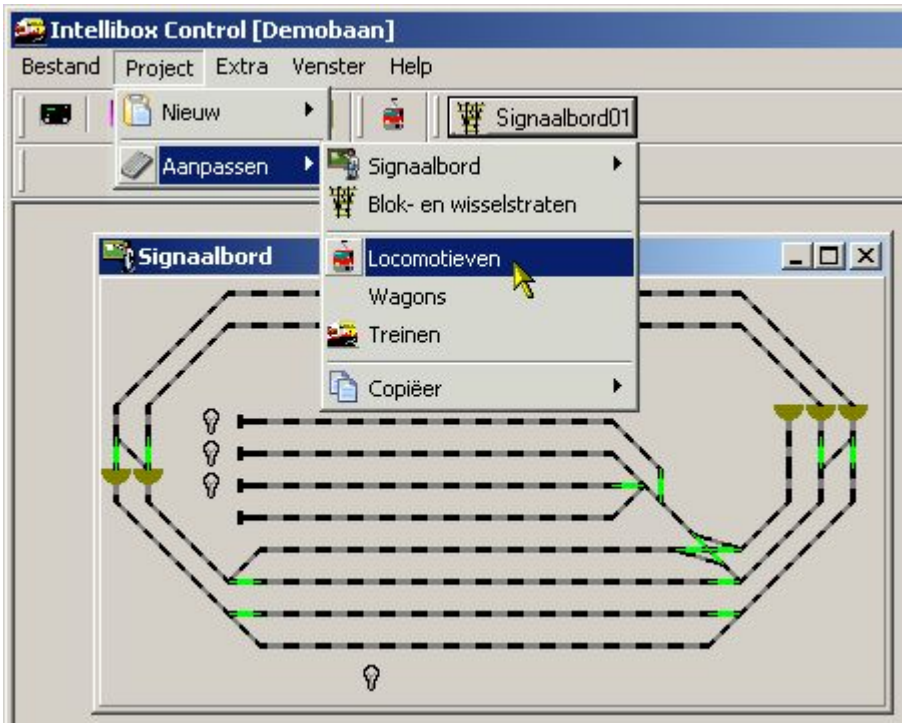
Druk op de knop in de knoppenbalk met de naam van het nieuwe signaalbord erop, of kies 'Venster' ⇒ 'Signaalbord' ⇒ 'Signaalbord01'.

Hier zie je het signaalbord. Als je nu klikt op een wissel, dan wordt dit doorgestuurd naar MySQL, waarna de server het commando uitvoert.

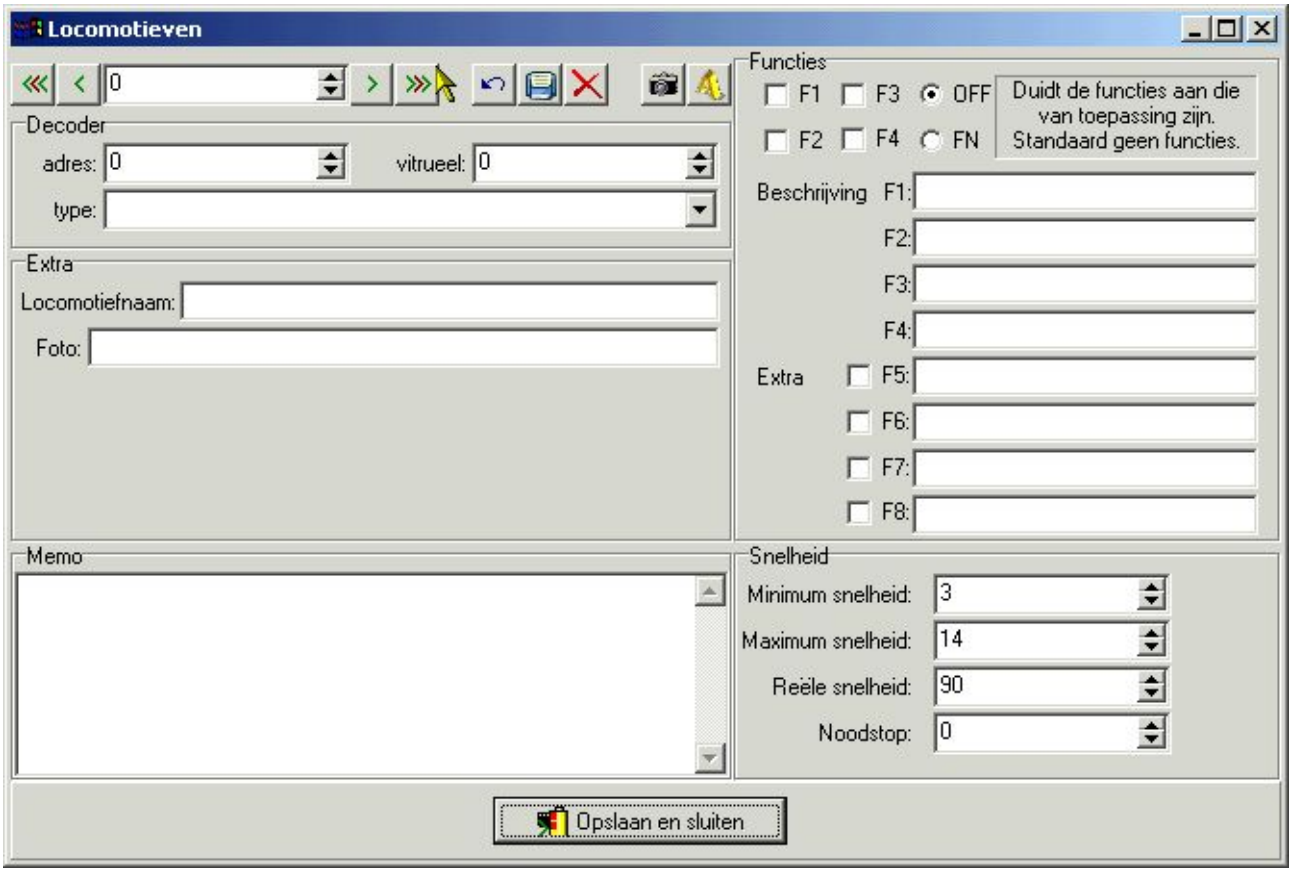


7.4 Locomotieven toevoegen.

Nu gaan we locomotieven toevoegen. Klik hiervoor op 'Project' ⇒ 'Aanpassen' ⇒ 'Locomotieven'



Een snelle kijk op het scherm leert ons dat er veel gegevens gevraagd worden.



- De knoppenbalk
  -  Eerste locomotief in de lijst.
  -  Vorige locomotief in de lijst.
  -  Volgende locomotief in de lijst, nieuwe locomotief als de laatste zichtbaar is.
  -  Laatste locomotief in de lijst.
  -  Maak wijzigingen ongedaan van de actieve locomotief.
  -  Sla locomotief op.
  -  Verwijder locomotief uit de lijst.
  -  Haal een bitmap op. Als deze niet op de standaard plaats staat, wordt deze gekopieerd.
  -  Haal de naam uit de bitmap en gebruik deze als naam voor de locomotief.

- Decoder
  Hier geef je het decoder-adres, het eventuele virtuele adres en het decodertype in.
  De adressen kunnen van 1 tot 9999 gaan.
  Om aan te duiden dat er geen virtueel adres is, geeft u 0 in.
  Het type decoder moet ook ingesteld worden.
  Dit is een overzicht van de verschillende types:
  - MotorolaOld
  - MotorolaNew
  - DCC14
  - DCC27
  - DCC28
  - DCC28DAC
  - DCC128
  - DCC128DAC
  - Selectrix

  Zie de gebruikershandleiding van de intellibox om het verschil te weten tussen deze, en hoe je kan weten welk type uw loc heeft.

- Extra
  De naam en de bitmap van de locomotief geef je hier in.
  De naam is verplicht. Deze kan eenvoudig uit de bitmap-naam worden gehaald.
  De bitmap kan je ophalen met het bovenstaande knopje met het fototoestel, waarna je op de A-knop drukt, zodat de naam wordt ingevuld.

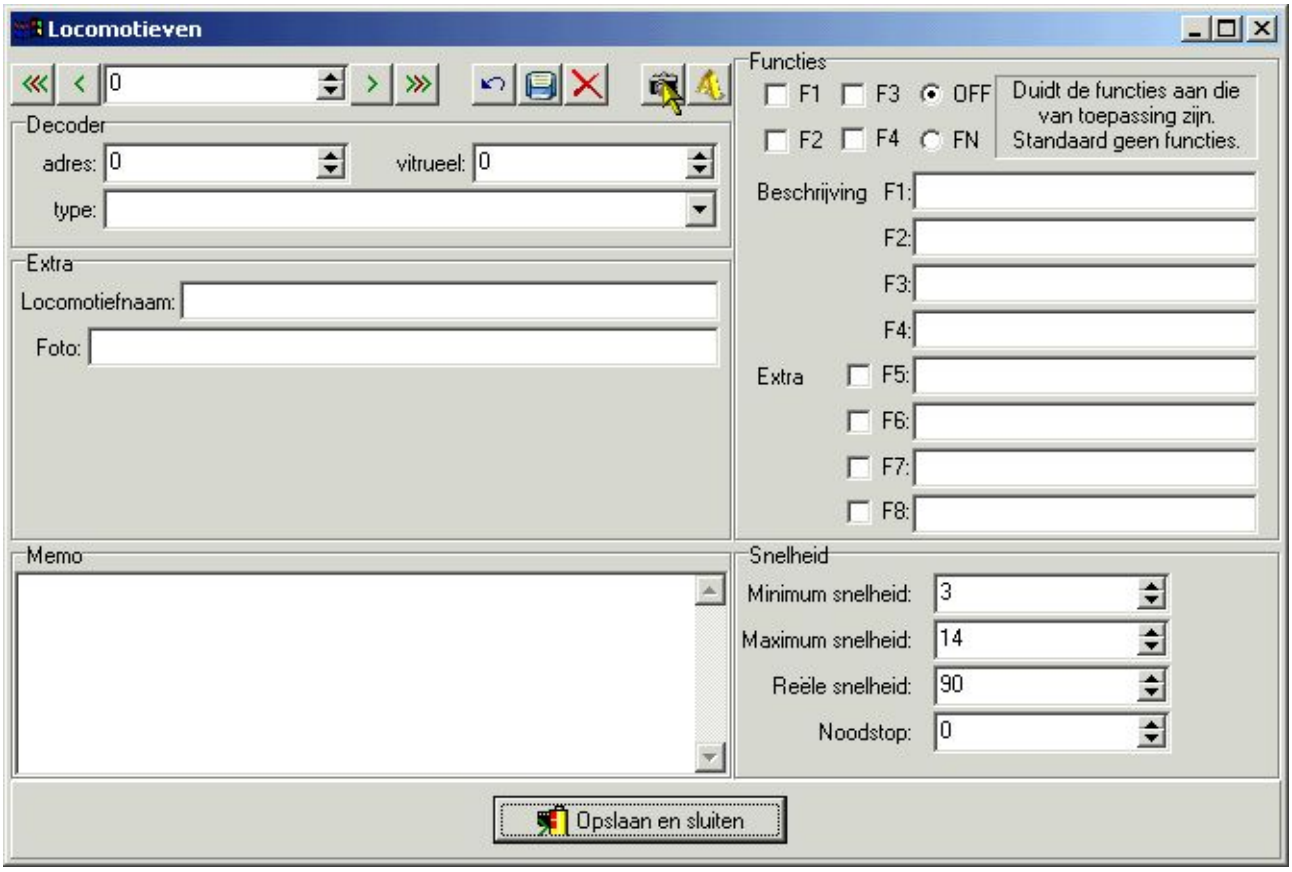
- Memo
  Hier kan je commentaar ingeven. Voorbeelden zijn: Merk, anecdote, aankoopdatum,...

- Functies
  Je kan maximaal 8 functies ingeven, plus aanduiden of deze locomotief koplampen heeft.
  Hier stel je in dat deze functies aanwezig zijn, dus niet dat deze aan moeten staan!
  Je kan ook voor elke functie een beschrijving geven, ten minste als je deze hebt aangeduid.

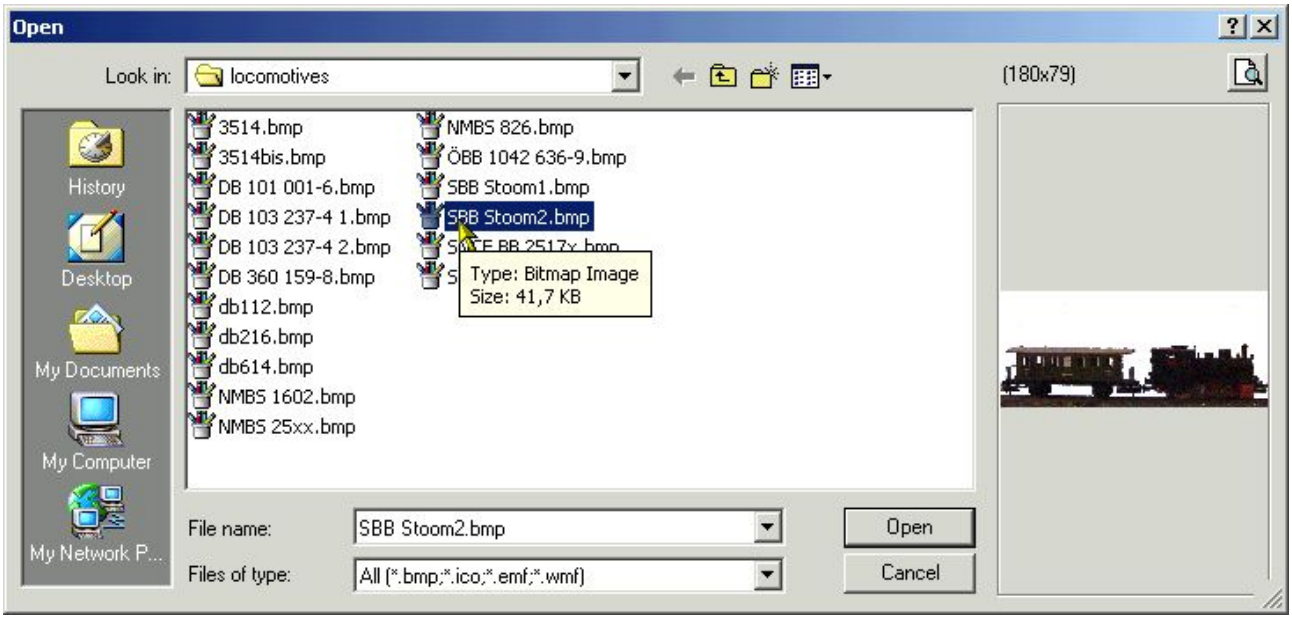
- Snelheid
  Een locomotief heeft verschillende belangrijke snelheden.
  Zo heb je de reële snelheid. Dit is de maximum snelheid die het grote voorbeeld kan halen.
  De minimum snelheid is de snelheid waarmee de loc moet rijden. Dit wordt ingegeven in decoder-stappen.
  De maximum snelheid is de maximale decoder-stap die de loc mag rijden.

Deze twee zijn interessant voor locs die onder een bepaalde decoderstap niet vooruit gaan en/of boven een bepaalde decoderstap te snel rijden en dus uit de bocht vliegen.
Als laatste is er de Noodstop. Deze is de decoderstap die de decoder als noodstop ziet.

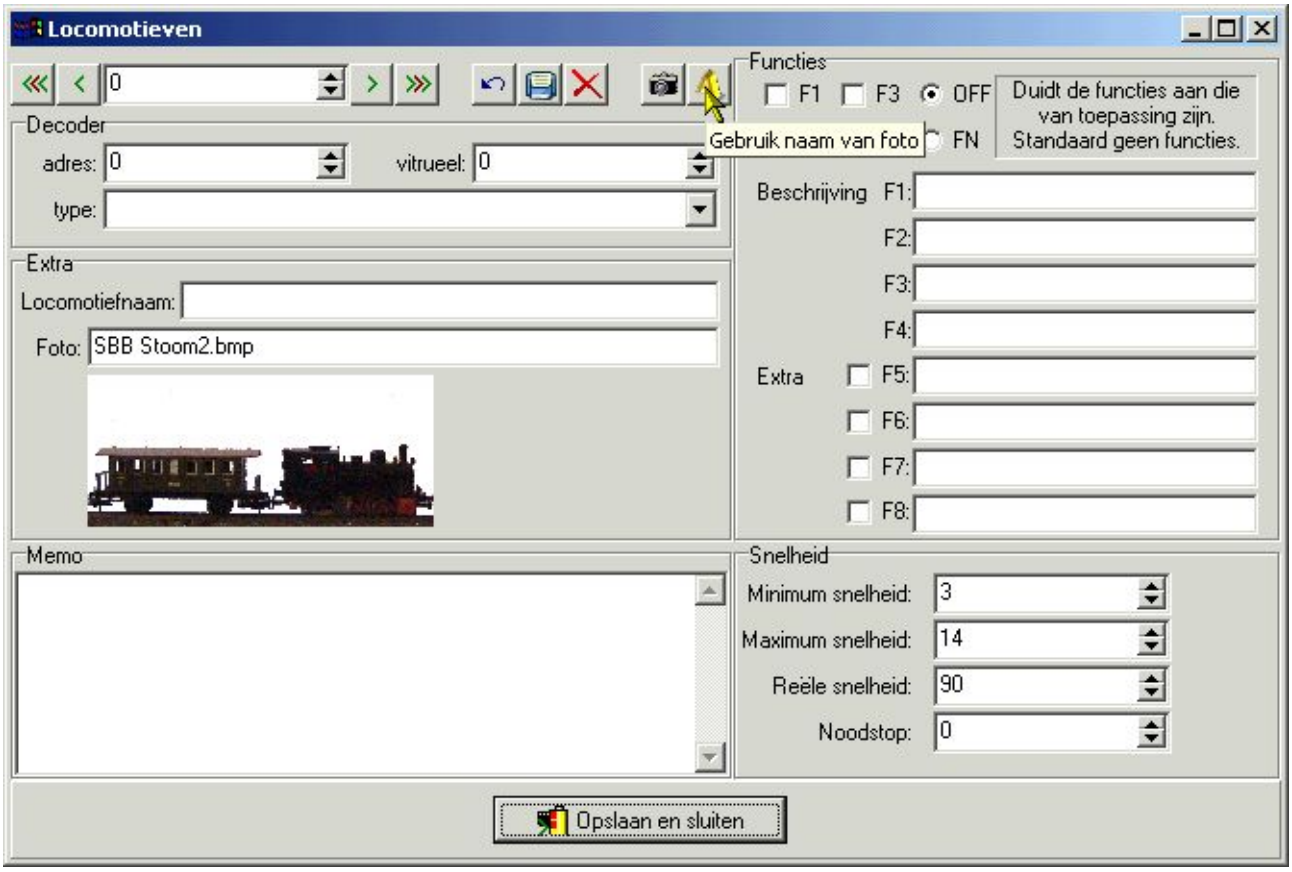
De eerste stap die je neemt is het ophalen van een bitmap.



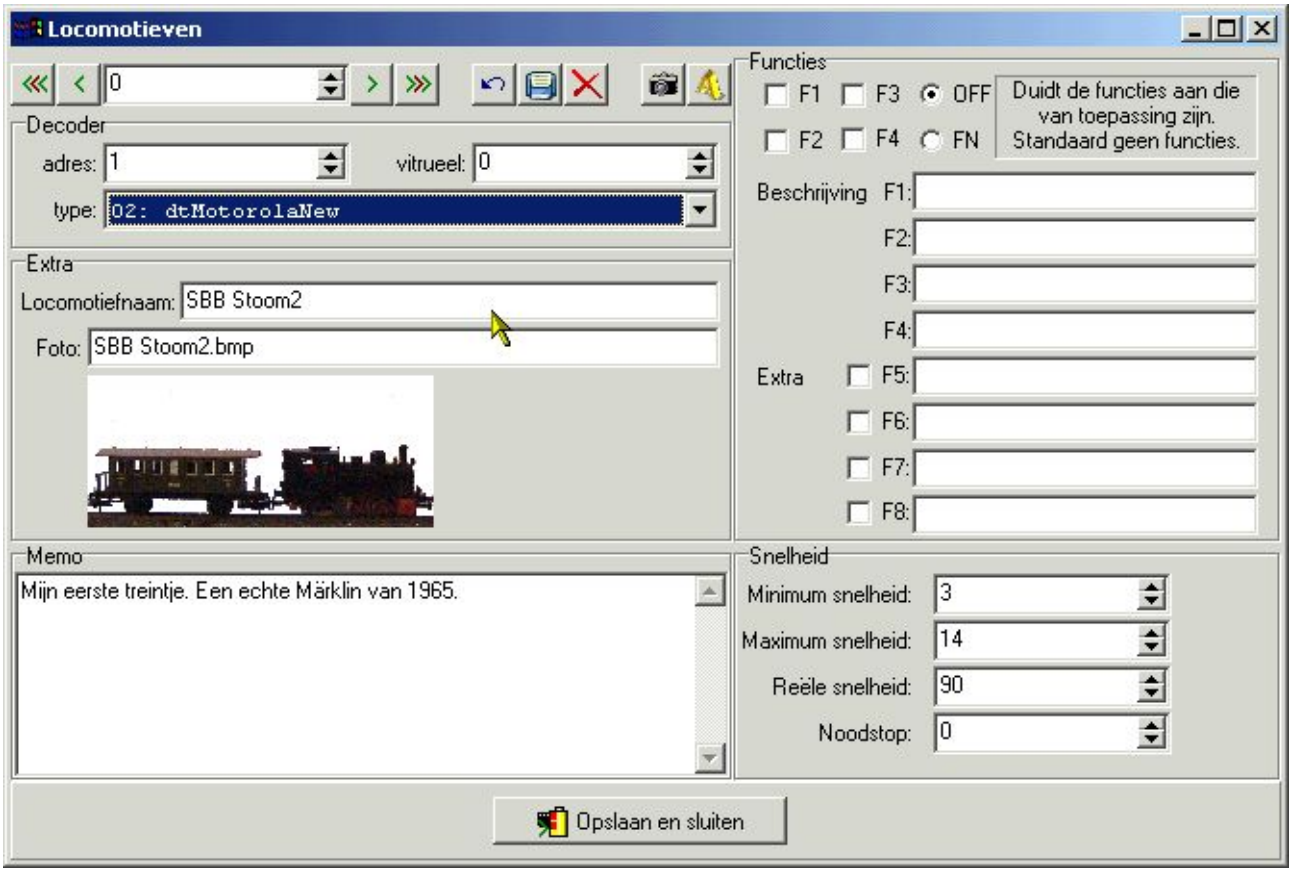
Er zijn al enkele bitmaps meegeleverd.



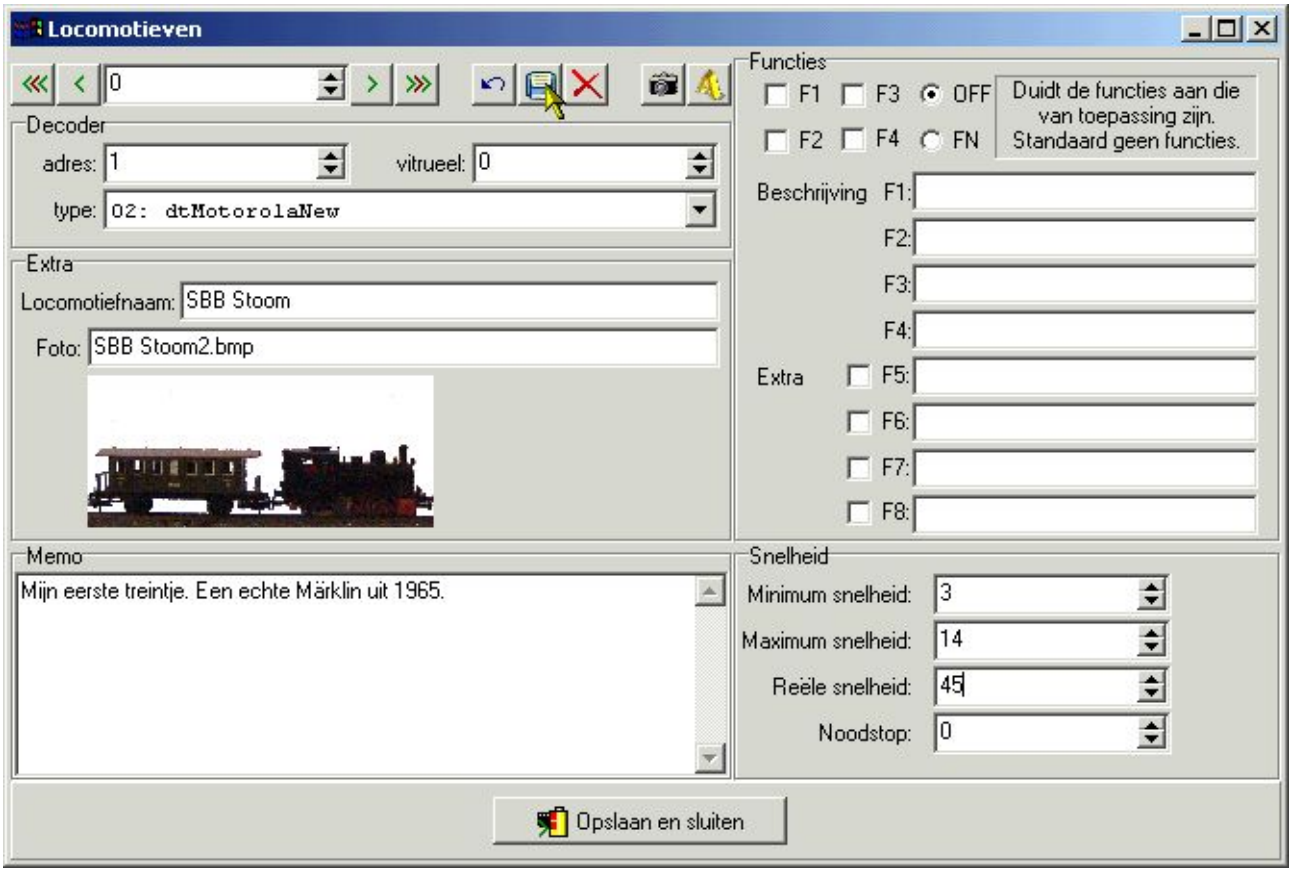
Nu halen we er de naam uit.



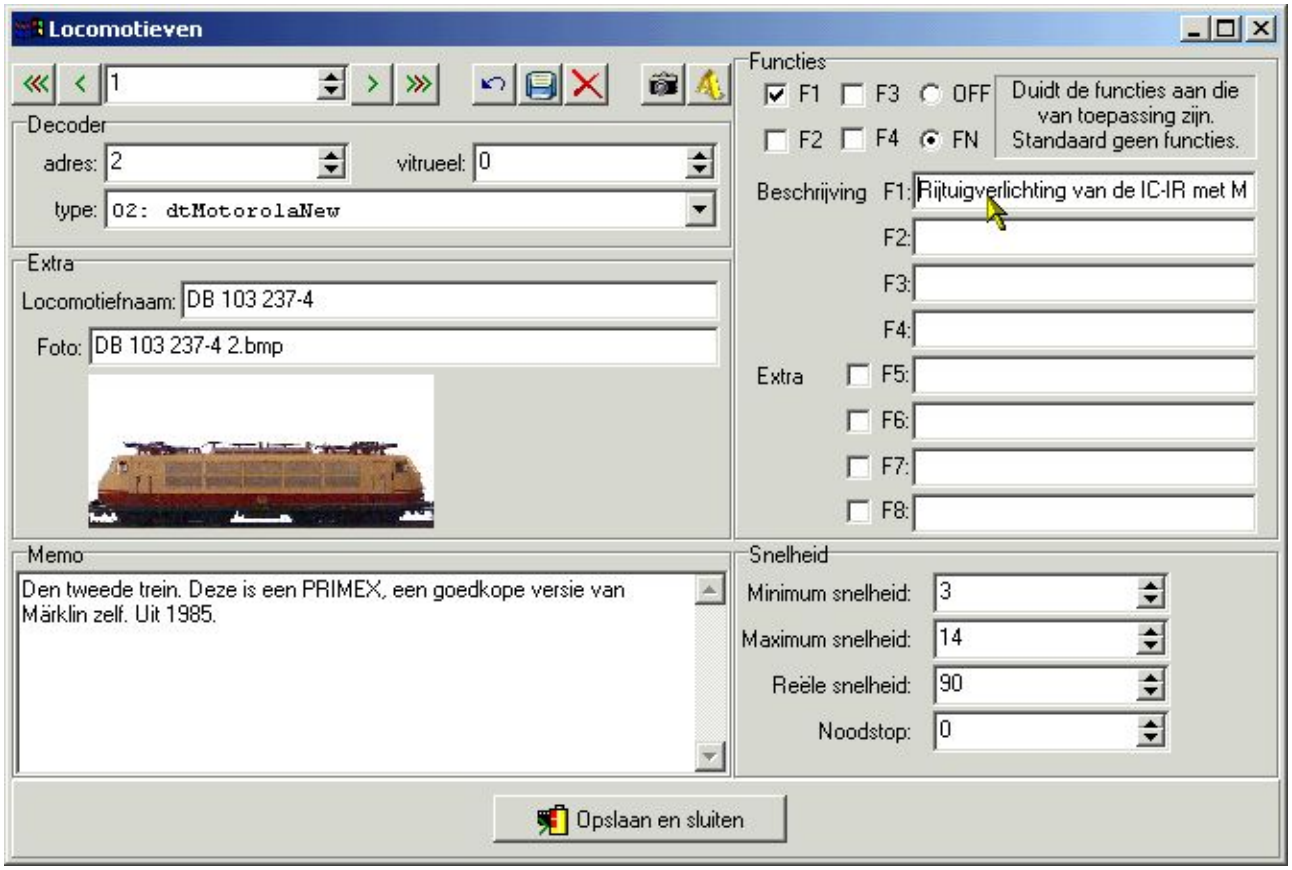
Pas eventueel de naam aan, geef het adres en het decodertype in.



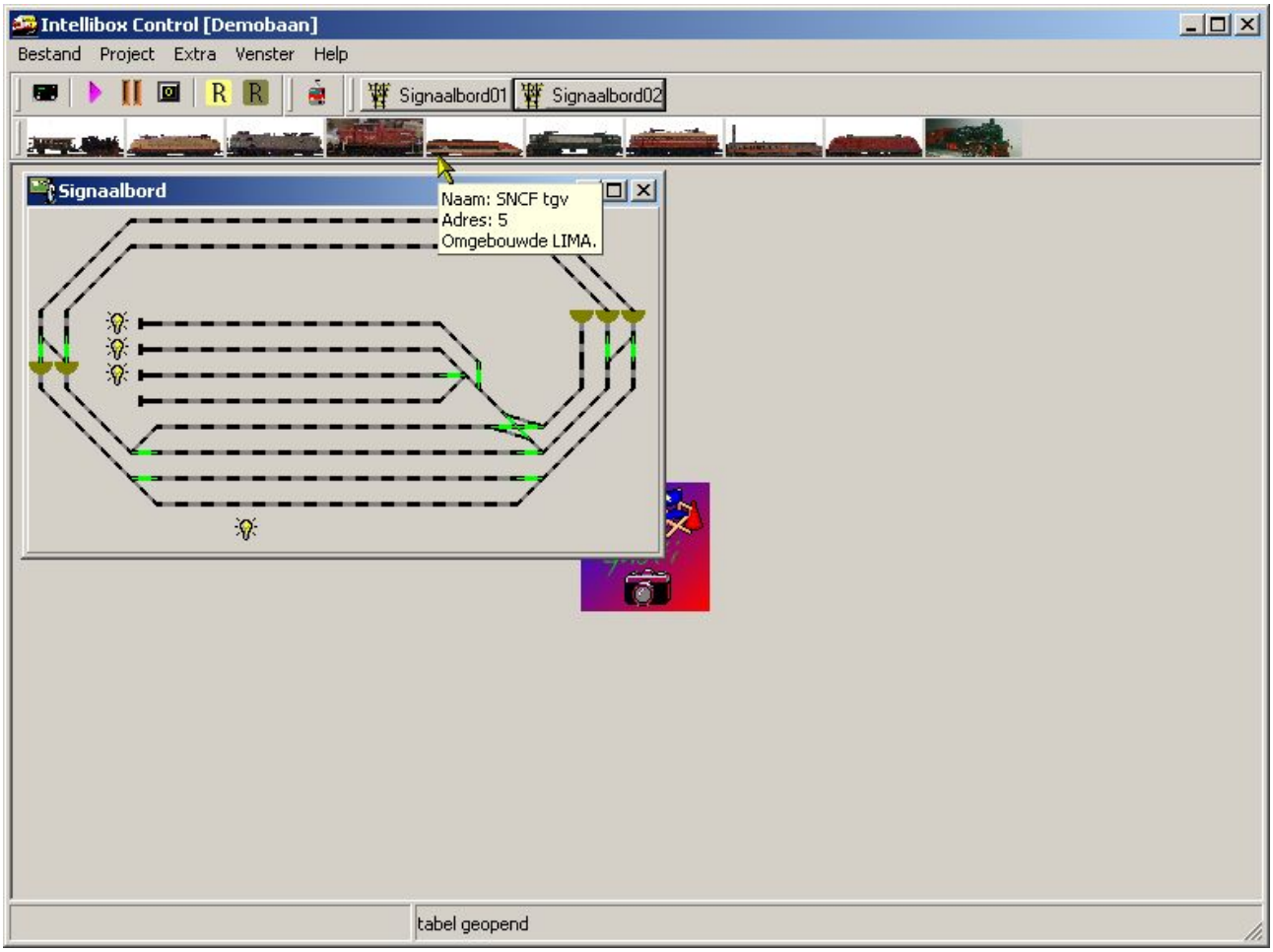
Nog een memo'tje en we slaan deze loc op.
In dit voorbeeld is een loc gebruikt die geen functies heeft.



Een voorbeeld van een loc met functies. FN en F1 aangeduid, dus deze kunnen bediend worden.



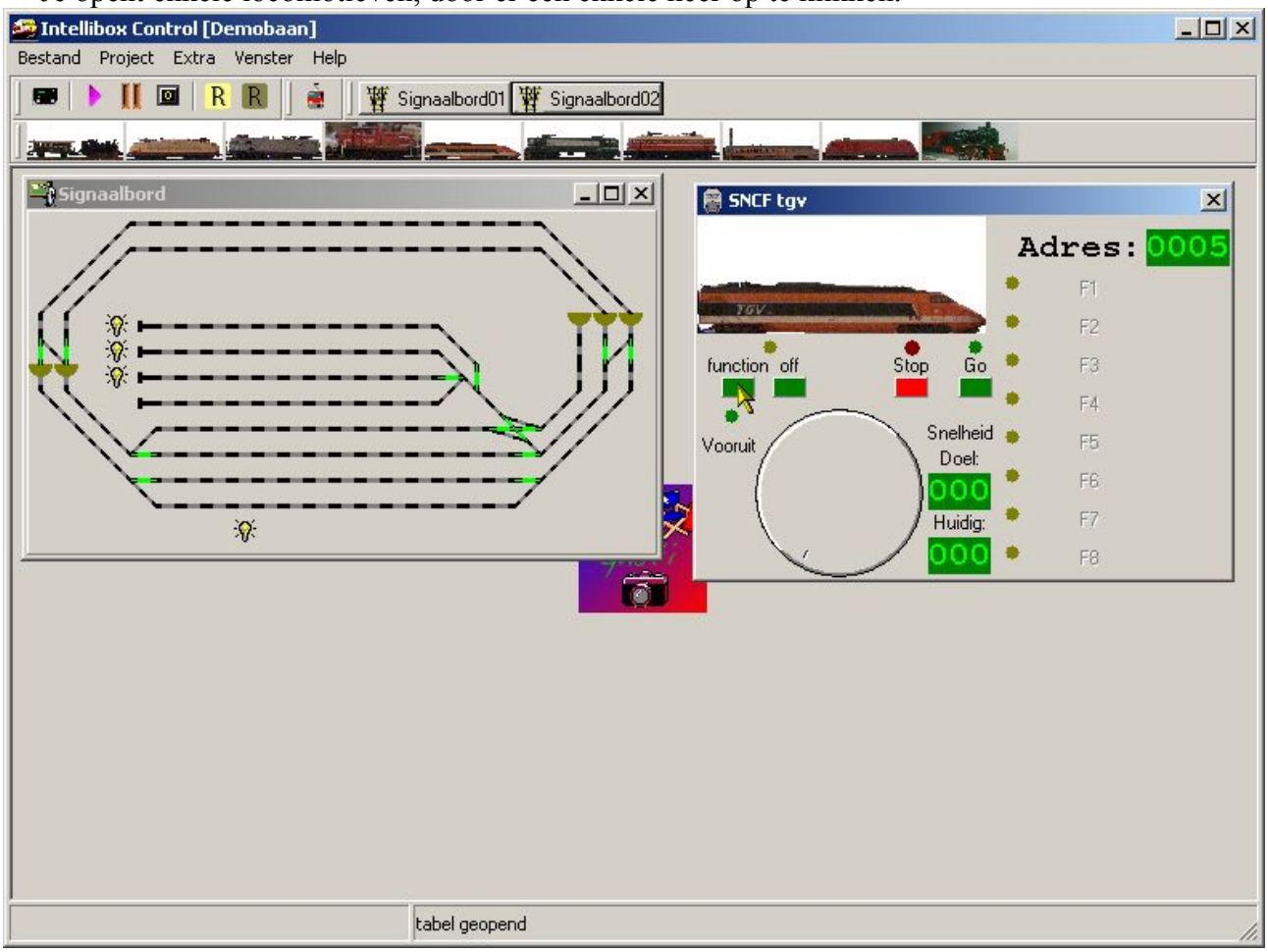
Als alle locomotieven zijn ingegeven, druk dan op 'Opslaan en sluiten'.
Je ziet de hele reeks locomotieven bovenaan verschijnen.



Zoals je ziet komt er een korte inhoud van de loc als je met de muiscursor erover gaat.

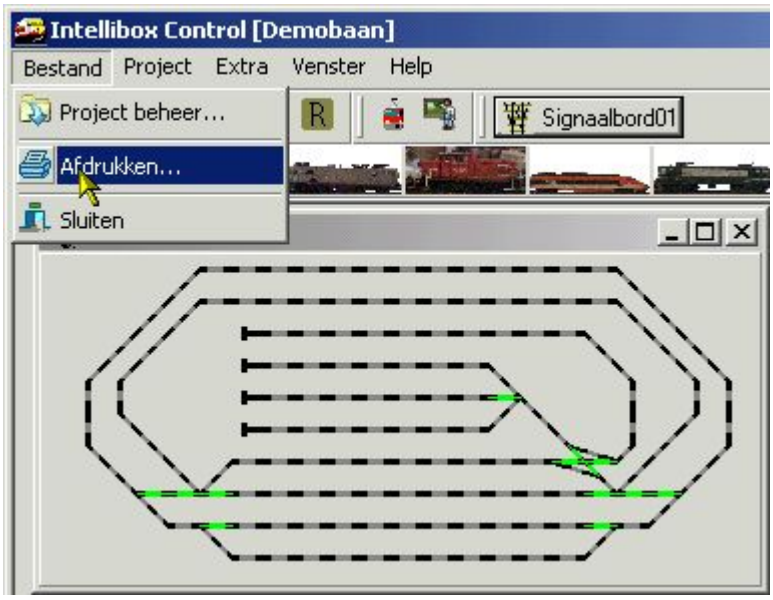
7.5 Bedienen van signaalborden en locomotieven.

Je opent enkele locomotieven, door er een enkele keer op te klikken.



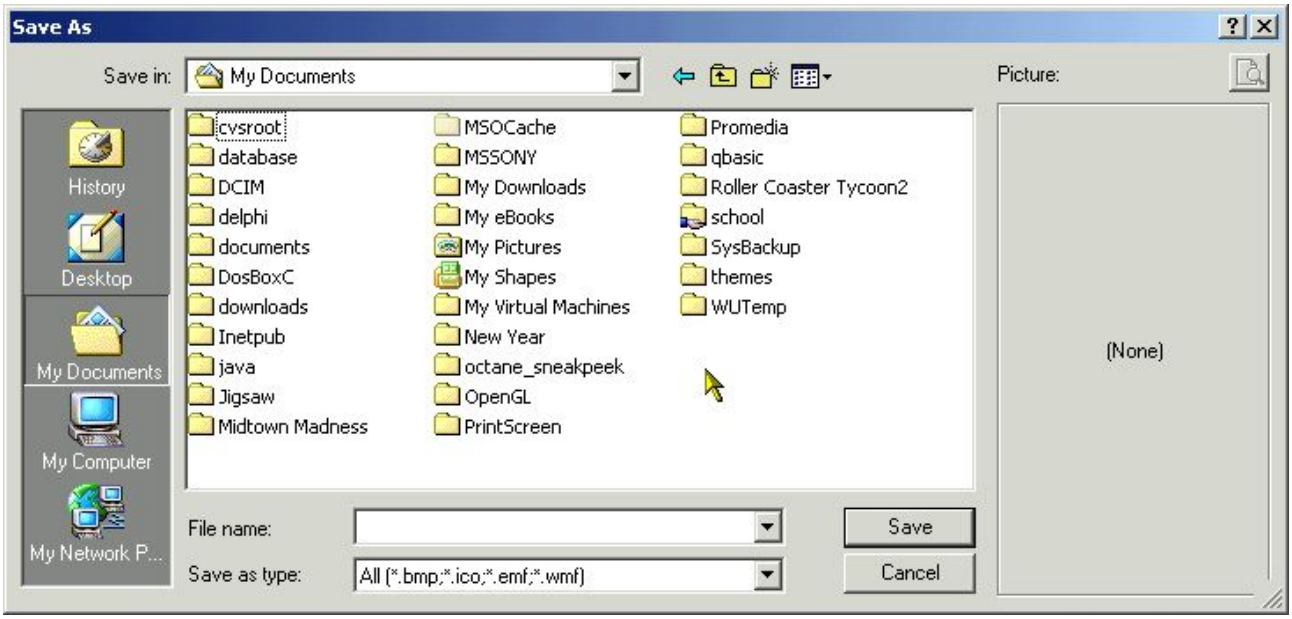
Nu kan je op een wissel klikken, of de verlichting van een loc aanzetten, een loc versnellen,...

7.6 Afdrukken van een signaalbord.

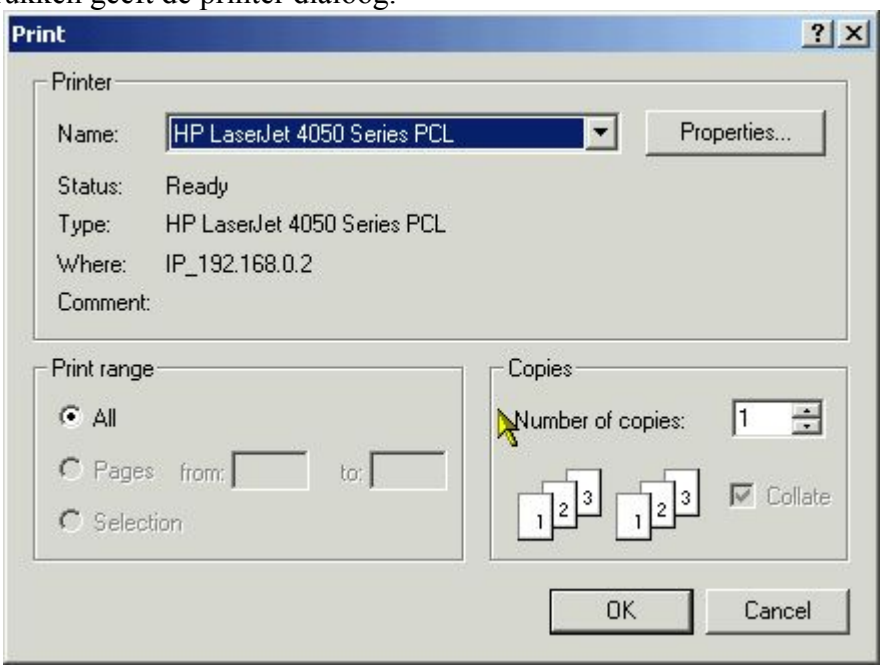


Momenteel kunnen enkel signaalborden afgeprint of geëxporteerd worden.

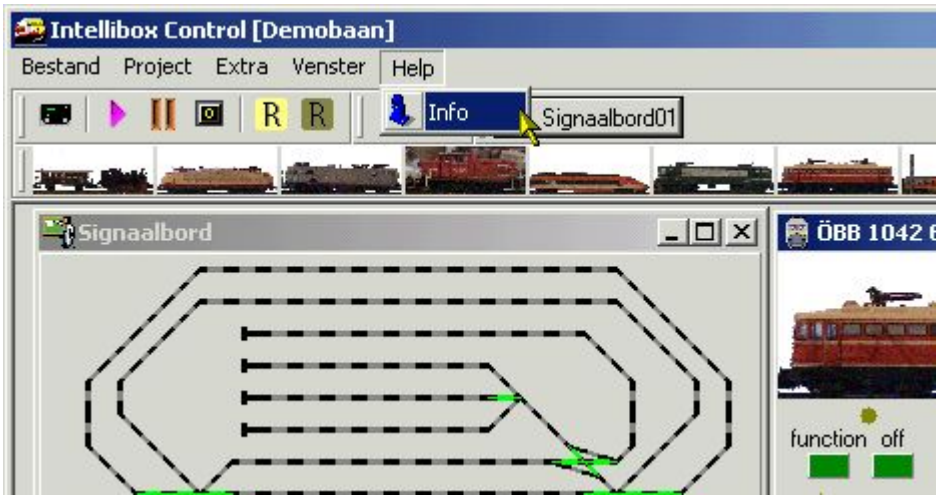
Als je 'Naar bitmap' aanduidt, dan wordt er naar een bestansnaam gevraagd.



Gewoon afdrukken geeft de printer dialoog.



7.7 Versiecontrole en informatie.



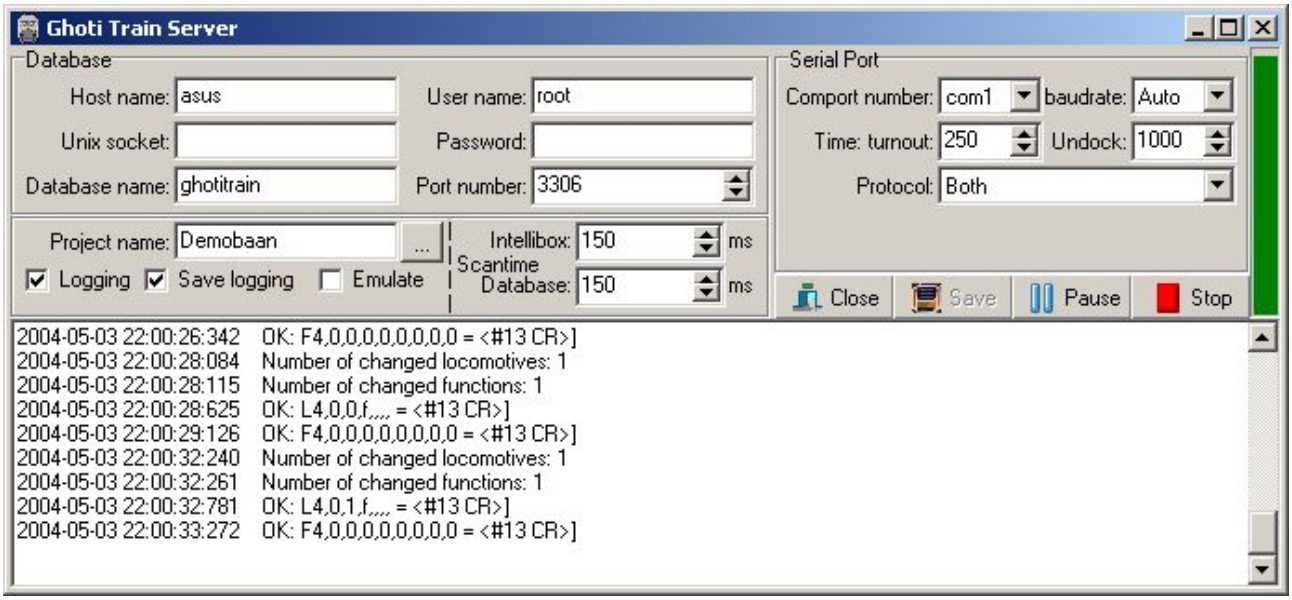
7.8 Het ini-bestand.

```
[Image]
Dir=D:\delphi\projects\Ghoti\GhotiTrain\images

[General]
MärklinSpd=1
MärklinSpeed=1
UpdateInterval=250

[MySQL]
Host=localhost
UnixSocket=
DataBase=ghotitrain
Port=3306
User=root
Password=

[Project]
Name=Demobaan
CountScreen=1
CountLocomotives=1

[Form]
WindowState=0
Height=503
Width=787
Top=-4
Left=-4

[Screen]
Screen0=Signaalbord01
Screen0-ID=0
Screen0-top=0
Screen0-left=0
Screen1=Signaalbord02
Screen1-ID=1
Screen1-top=107
Screen1-left=360

[Locomotives]
Locomotives0=DB 360 159-8
Locomotives0-ID=3
Locomotives0-top=28
Locomotives0-left=317
```

`[Image]`	De map waarin de bitmaps staan.
`[General]`	De algemene instellingen.
`[MySQL]`	De instellingen voor MySQL.
`[Project]`	Het actieve project, het aantal signaalborden en locomotieven.
`[Form]`	De formstaat en de parameters.
`[Screen]`	Elk geopend scherm. Dit kunnen er meer zijn, deze worden nl. niet gewist.
`[Locomotives]`	Elke geopende loc. Dit kunnen er eveneens meer zijn, om de zelfde reden als bij `[screen]`.

Belangrijk extra! Standaard heeft het ini-bestand dezelfde naam als de client.
Als je trainclient opstart met als parameter een ander ini-bestand, wordt dit genomen.

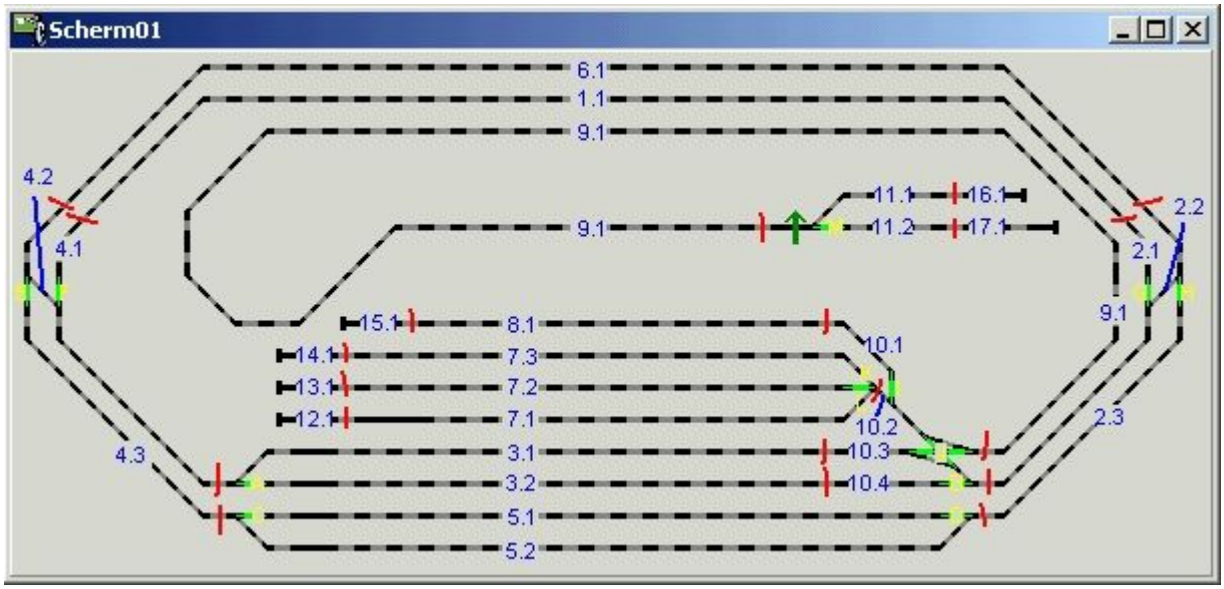
7.9 Instellen van de server.



Bij de server vinden we 6 delen terug.

- Database: Dit is hetzelfde als bij de client.
- Serial Port: Hier stel je de compoort en de intellibox in.
  De compoort waar de intellibox op aan gesloten is.
  Als je de baudrate die in de intellibox is ingesteld wenst te gebruiken, dan stel je deze in op 'Auto'.
  De tijden: Turnout geeft aan hoe lang een wissel bekrachtigd wordt.
  Undock geeft aan hoe lang een ontkoppelrail actief is.
  Het protocol dat moet gebruikt worden. Dit is momenteel enkel 'Both'.
- Project: Uiteraard moet je opgeven welk project moet worden gevolgd.
  Verder geef je hier aan of de logging actief is;
  Dat deze logging moet worden opgeslagen;
  Dat er geëmuleert moet worden (=zonder Intellibox werken);
  Om de hoeveel tijd dat de Intellibox en resp. de databank moet worden aangesproken.
- De knoppen: Met 'close' schakel je de server uit. Alleen langs hier werkt dit.
  Zo kan je niet per ongeluk de server uitschakelen.
  De 'save' knop slaat alle veranderde instellingen op.
  'Pause' pauseert de server.
  'Start' doet de server starten.
- Voortgangsindicator: Hieraan zie je dat de logging werkt.
  De server kan je enkel aan de 'Start' knop zien dat deze werkt.
- Logging: Hier zie je wat de server doet.

Appendix A: Een voorbeeld van wisselstraten.



id	block_id	wisselstr	volgend blok	volgend wisselstr	vorig blok	vorig wisselstr	wissel	stand
1	1	1	2	1	4	1	/	/
2	2	1	10	4	1	1	G	afbuigen
3	2	1	10	4	1	1	B	recht
4	2	1	10	3	1	1	G	afbuigen
5	2	1	10	3	1	1	B	afbuigen
6	2	1	10	3	1	1	J	afbuigen
7	2	1	10	2	1	1	G	afbuigen
8	2	1	10	2	1	1	B	afbuigen
9	2	1	10	2	1	1	J	recht
10	2	1	10	2	1	1	I	afbuigen
11	2	1	10	1	1	1	G	afbuigen
12	2	1	10	1	1	1	B	afbuigen
13	2	1	10	1	1	1	J	recht
14	2	1	10	1	1	1	I	recht
15	2	2	10	4	6	1	H	afbuigen
16	2	2	10	4	6	1	G	recht
17	2	2	10	4	6	1	B	recht
18	2	2	10	3	6	1	H	afbuigen
19	2	2	10	3	6	1	G	recht
20	2	2	10	3	6	1	B	afbuigen
21	2	2	10	3	6	1	J	afbuigen
22	2	2	10	2	6	1	H	afbuigen
23	2	2	10	2	6	1	G	recht
24	2	2	10	2	6	1	B	afbuigen
25	2	2	10	2	6	1	J	recht
26	2	2	10	2	6	1	I	afbuigen
27	2	2	10	1	6	1	H	afbuigen
28	2	2	10	1	6	1	G	recht
29	2	2	10	1	6	1	B	afbuigen
30	2	2	10	1	6	1	J	recht
31	2	2	10	1	6	1	I	recht
32	2	3	6	1	5	1	H	recht
33	2	3	6	1	5	1	D	recht
34	2	3	6	1	5	2	H	recht
35	2	3	6	1	5	2	D	afbuigen

36	3	1	4	1	10	3	A	afbuigen
37	3	1	4	1	10	3	F	afbuigen
38	3	1	4	2	10	3	A	afbuigen
39	3	1	4	2	10	3	F	recht
40	3	2	4	1	10	4	A	recht
41	3	2	4	1	10	4	F	afbuigen
42	3	2	4	2	10	4	A	recht
43	3	2	4	2	10	4	F	recht
44	4	1	1	1	3	1	A	afbuigen
45	4	1	1	1	3	1	F	afbuigen
46	4	1	1	1	3	2	A	recht
47	4	1	1	1	3	2	F	afbuigen
48	4	2	6	1	3	1	A	afbuigen
49	4	2	6	1	3	1	E	afbuigen
50	4	2	6	1	3	1	F	recht
51	4	2	6	1	3	2	A	recht
52	4	2	6	1	3	2	E	afbuigen
53	4	2	6	1	3	2	F	recht
54	4	3	5	1	6	1	C	recht
55	4	3	5	1	6	1	E	recht
56	4	3	5	2	6	1	C	afbuigen
57	4	3	5	2	6	1	E	recht
58	5	1	4	3	2	3	C	recht
59	5	1	4	3	2	3	D	recht
60	5	2	4	3	2	3	C	afbuigen
61	5	2	4	3	2	3	D	afbuigen
62	6	1	4	2	2	2	E	afbuigen
63	6	1	4	2	2	2	H	afbuigen
64	6	1	4	2	2	3	E	afbuigen
65	6	1	4	2	2	3	H	recht
66	6	1	4	3	2	2	E	recht
67	6	1	4	3	2	2	H	afbuigen
68	6	1	4	3	2	3	E	recht
69	6	1	4	3	2	3	H	recht
70	7	1	12	1	10	2	L	afbuigen
71	7	1	12	1	10	2	K	recht
72	7	2	13	1	10	2	L	recht
73	7	2	13	1	10	2	K	recht
74	7	3	14	1	10	2	L	recht
75	7	3	14	1	10	2	K	afbuigen
76	8	1	15	1	10	1	/	/
77	9	1	10	1	11	1	J	afbuigen
78	9	1	10	1	11	1	I	recht
79	9	1	10	1	11	1	M	afbuigen
80	9	1	10	2	11	1	J	afbuigen
81	9	1	10	2	11	1	I	afbuigen
82	9	1	10	2	11	1	M	afbuigen
83	9	1	10	3	11	1	J	recht
84	9	1	10	3	11	1	M	afbuigen
85	9	1	10	1	11	2	J	afbuigen
86	9	1	10	1	11	2	I	recht
87	9	1	10	1	11	2	M	recht
88	9	1	10	2	11	2	J	afbuigen
89	9	1	10	2	11	2	I	afbuigen
90	9	1	10	2	11	2	M	recht
91	9	1	10	3	11	2	J	recht
92	9	1	10	3	11	2	M	recht
93	10	1	8	1	9	1	I	recht
94	10	1	8	1	9	1	J	afbuigen
95	10	1	8	1	2	1	I	recht
96	10	1	8	1	2	1	J	afbuigen
97	10	1	8	1	2	1	G	afbuigen

98	10	1	8	1	2	2	I	recht
99	10	1	8	1	2	2	J	afbuigen
100	10	1	8	1	2	2	G	recht
101	10	2	7	1	9	1	I	afbuigen
102	10	2	7	1	9	1	J	afbuigen
103	10	2	7	1	9	1	K	recht
104	10	2	7	1	9	1	L	afbuigen
105	10	2	7	1	2	1	G	afbuigen
106	10	2	7	1	2	1	I	afbuigen
107	10	2	7	1	2	1	J	recht
108	10	2	7	1	2	1	K	recht
109	10	2	7	1	2	1	L	afbuigen
110	10	2	7	1	2	2	G	recht
111	10	2	7	1	2	2	I	afbuigen
112	10	2	7	1	2	2	J	recht
113	10	2	7	1	2	2	K	recht
114	10	2	7	1	2	2	L	afbuigen
115	10	2	7	2	9	1	I	afbuigen
116	10	2	7	2	9	1	J	afbuigen
117	10	2	7	2	9	1	K	recht
118	10	2	7	2	9	1	L	recht
119	10	2	7	2	2	1	G	afbuigen
120	10	2	7	2	2	1	I	afbuigen
121	10	2	7	2	2	1	J	recht
122	10	2	7	2	2	1	K	recht
123	10	2	7	2	2	1	L	recht
124	10	2	7	2	2	2	G	recht
125	10	2	7	2	2	2	I	afbuigen
126	10	2	7	2	2	2	J	recht
127	10	2	7	2	2	2	K	recht
128	10	2	7	2	2	2	L	recht
129	10	2	7	3	9	1	I	afbuigen
130	10	2	7	3	9	1	J	afbuigen
131	10	2	7	3	9	1	K	afbuigen
132	10	2	7	3	9	1	L	recht
133	10	2	7	3	2	1	G	afbuigen
134	10	2	7	3	2	1	I	afbuigen
135	10	2	7	3	2	1	J	recht
136	10	2	7	3	2	1	K	afbuigen
137	10	2	7	3	2	1	L	recht
138	10	2	7	3	2	2	G	recht
139	10	2	7	3	2	2	I	afbuigen
140	10	2	7	3	2	2	J	recht
141	10	2	7	3	2	2	K	afbuigen
142	10	2	7	3	2	2	L	recht
143	10	3	3	1	9	1	J	recht
144	10	3	3	1	2	1	J	afbuigen
145	10	3	3	1	2	1	G	afbuigen
146	10	3	3	1	2	2	J	afbuigen
147	10	3	3	1	2	2	G	recht
148	10	4	3	2	2	1	G	afbuigen
149	10	4	3	2	2	2	G	recht
150	11	1	16	1	9	1	M	afbuigen
151	11	1	17	1	9	1	M	recht
152	12	1	/	/	7	1	/	/
153	13	1	/	/	7	2	/	/
154	14	1	/	/	7	3	/	/
155	15	1	/	/	8	1	/	/

Appendix B: De intellibox.