VPEB Van Perlo Elektronica & Besturingstechniek

OC32 3.0

Uitgebreide Configuratie

Auteur: Leon J.A. van Perlo
Versie: 3.0
Datum: 27 september 2015

Slimme elektronica voor de miniatuurwereld

Release beheer

Deze handleiding is van toepassing op

- Print
  - OC32 Rev 00
  - OC32 Rev 01
  - OC32 Rev 02
  - OC32 Rev 03
  - OC32 Rev 04

- Firmware
  - OC32 Rel 3.0.0.0

- Software
  - OC32Config Rel 3.0.0.0
  - OM32Config 1.41

©2011-2015 Dit document, dan wel enige informatie hieruit, mag niet worden gekopieerd en/of verspreid, geheel of gedeeltelijk, in welke vorm dan ook zonder uitdrukkelijke schriftelijke toestemming van de oorspronkelijke auteur. Het maken van kopieën en afdrukken door gebruikers van de OC32 module uitsluitend ten behoeve van eigen gebruik is toegestaan.

Voorwoord / Leeswijzer

Dit document is bedoeld voor degenen die al enige ervaring hebben met de OC32. De onderwerpen zijn bedoeld voor een beter begrip bij het gebruik van geavanceerdere instellingen. Ze stellen je in staat je eigen apparaatdefinities te maken of bestaande definities naar eigen inzicht aan te passen.

Dit document is een aanvulling op de handleiding OC32 3.0 en behandelt één of meer complexere onderwerpen. Deze handleiding verwacht dat je de handleiding OC32 3.0 reeds hebt bestudeerd en op hoofdlijnen hebt begrepen.

Evenals in de OC32 3.0 handleiding is er een leeshulp aangebracht in de kantlijn, die een indicatie geeft welk expertiseniveau er noodzakelijk wordt geacht om de materie te kunnen toepassen

Groen	Beginnersniveau: hiermee moet je de basisfuncties werkend kunnen krijgen. Biedt geen uitgebreide keuzemogelijkheden, slimme besparingen of complexe combinaties
Blauw	Geavanceerd niveau; Vereist basiskennis elektronica, enige handigheid met PC software, enig logisch inzicht of een combinatie daarvan. Het vereist dat je zelf een aantal keuzes maakt en dus kunt beoordelen wat die keuzes inhouden. In principe voor iedereen uitvoerbaar, maar niet voor iedereen handig om meteen mee te beginnen
Oranje	Expert niveau: Vereist redelijke tot goede kennis van elektronica, goed logisch inzicht, enige kennis van programmeertechnieken of een combinatie daarvan. Wat hier beschreven wordt kan mogelijk bij foutieve uitvoering schadelijk zijn voor elektronica of andere onderdelen. Dus alleen doen als je het ook snapt.

Veel plezier!

Inhoud

1 OC32Config - Enkele handige bedieningselementen 5
2 Pin-configuratie 7
2.1 Inleiding 7
2.2 Automatisch (Clear) 8
2.3 Servo mode 8
2.4 PWM Mode 11
2.5 Input mode 14
3 Aspect Configuratie 15
3.1 Principes 15
3.2 Aspects, Instructies en Parameters 15
3.3 Aspect Configuratie - Normale Instructies 18
3.4 Aspect Configuratie - Timed Instructies 20
3.5 Aspect Configuratie - Sequence Instructies 20
3.6 Event Timers en Periode Timers 22
4 Apparaatdefinities maken 23
5 Toekennen adressen aan Pinnen 25

1 OC32Config - Enkele handige bedieningselementen

Op het tabblad OC32Device Definition zitten een aantal aanvullende bedieningselementen, die vooral handig zijn als je zelf je apparaatdefinities gaat maken:

Om dit uit te proberen, start OC32Config vers op, ga naar het tabblad OC32 Device Configuration, vink "Show Details" aan en vink het selectierondje "4" of "12" achter het label Aspects aan.

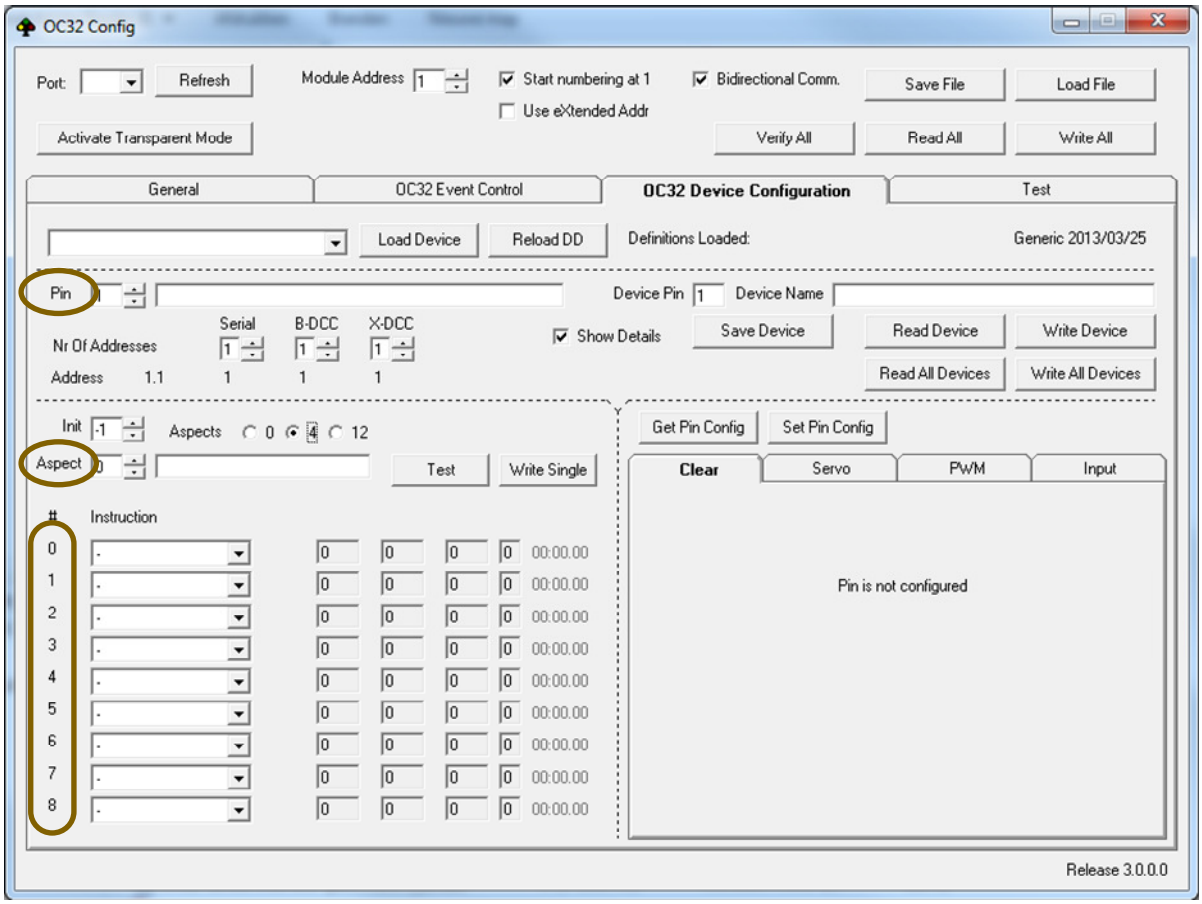


Fig 1: Aanvullende bedieningselementen

Klik op de tekst "Pin" links van het vak waar je het Pin nummer kunt selecteren. Je krijgt een pop-up schermpje met de volgende mogelijkheden:

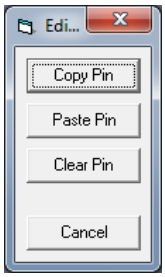


Fig 2: Edit Pin

- Copy Pin: kopieert ALLE instellingen van de geselecteerde Pin naar een tijdelijk buffer
- Paste Pin: plakt de eerder met Copy Pin gekopieerde instellingen op de geselecteerde Pin
- Clear Pin: wist ALLE instellingen van de geselecteerde Pin
- Cancel: Sluit het scherm zonder verder iets te doen.

Klik op de tekst "Aspect" ergens linksboven in het vak linksonder.Je krijgt een pop-up schermpje met de volgende mogelijkheden:

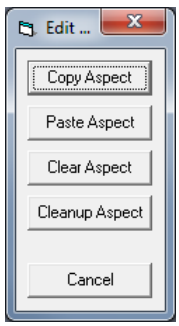


Fig 3: Edit Aspect

- Copy Aspect: kopieert de instructies van de geselecteerde Pin/Aspect naar een tijdelijk buffer
- Paste Aspect: plakt de eerder met Copy Aspect gekopieerde instructies naar de geselecteerde Pin/Aspect
- Clear Aspect: wist alle instructies van de geselecteerde Pin/Aspect
- Cleanup Aspect: wist alle instructies na één of meerdere lege instructies van de geselecteerde Pin/Aspect
- Cancel: Sluit het scherm zonder verder iets te doen.

Klik op één van de nummertjes 0..8 (het nummer van de instructie) aan de linkerzijde van het vak linksonder. Je krijgt een pop-up schermpje met de volgende mogelijkheden:

- Insert Step: Schuift de geselecteerde en alle volgende instructies één positie naar beneden en creëert een lege instructie. Let op: De laatste instructie valt van de lijst af.
- Delete Step: Verwijdert de geselecteerde instructie en schuift alle volgende instructies één positie naar boven.
- Cancel: Sluit het scherm zonder verder iets te doen.

Fig 4: Edit Step

2 Pin-configuratie

2.1 Inleiding

Pin-configuratie beschrijft de manier waarop een pin (uitgang) wordt aangestuurd. Er zijn grofweg drie mogelijkheden: PWM (normale aansturing), Servo en Input. Alle mogelijkheden kennen uitgebreide instellingen die we onderstaand beschrijven.

Pin Configuratie vind je onder de tab "OC32 Device Configuration". Binnen deze tab moet je het vinkje "Show Details" activeren. Je ziet dan ruwweg het scherm van figuur 5. Het gedeelte binnen de stippellijn rechtsonder is het gedeelte waarin je de Pin Configuratie bepaalt.

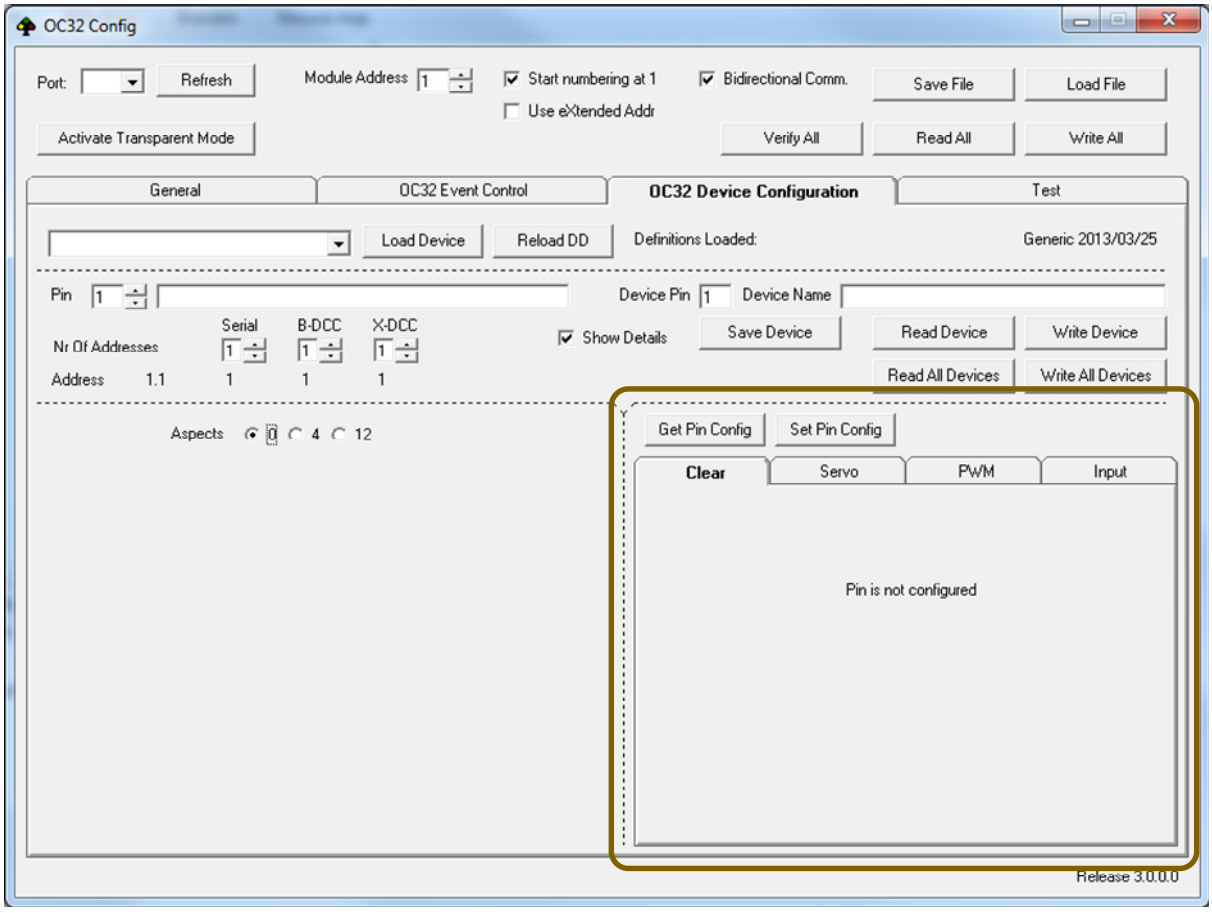


Fig 5: OC32 Pin Configuratie instellen

Merk op dat, wanneer je een voorgebakken apparaatdefinitie gebruikt, deze meestal reeds een voorstel doet voor de Pin Configuratie van de pinnen die door het betreffende apparaat worden gebruikt. In dergelijke gevallen hoef je de keuze dus niet zelf te maken.

Handmatige Pin-configuratie zoals beschreven in deze paragraaf is bedoeld als de apparaatdefinities niet toereikend zijn, meestal dus als je speciale dingen wilt realiseren of maximale controle wilt over het gedrag van een apparaat.

Er zijn 4 hoofdinstellingen die onderstaand gedetailleerd worden beschreven:

- Automatisch (not configured).
- Servo Mode
- PWM Mode
- Input Mode

Bovenin het door de stippellijn omgeven gedeelte vind je 2 knoppen die bedoeld zijn om de Pin Configuratie naar de OC32 te sturen of uit de OC32 te lezen:

- Get Pin Config: Leest de actieve Pin Configuratie uit de OC32 voor de geselecteerde pin;
- Set Pin Config: Schrijft de Pin Configuratie naar de OC32 als actieve configuratie voor de geselecteerde pin.

De termen "Get" en "Set" zijn afwijkend van "Read" en "Write" om aan te geven dat het hier om de actuele en **vluchtige** (tijdelijke) instellingen gaat. "Set" stelt dus alleen de actieve parameters voor de Pin in en schrijft niets naar het configuratiegeheugen van de OC32. "Get" lees de daadwerkelijke, huidige stand van de diverse parameters en dit kan dus afwijken van hetgeen in het configuratiegeheugen is opgeslagen. "Get" en "Set" zijn bedoeld om te testen of de instellingen die je aanpast bevallen.

Het permanent wegschrijven van de instellingen naar de OC32 doe je vanaf versie 3.0.0.0 door middel van de "Write Device" knop.

2.2 Automatisch (Clear)

Een pin die "Clear" (niet geconfigureerd) is wordt niet aangestuurd totdat een commando wordt ontvangen voor de betreffende pin. Indien dit een OM32 commando is wordt de pin automatisch in PWM mode (pulsbreedtemodulatie) gezet en wel met parameters die overeenkomen met het gedrag van een OM32 uitgang. Deze stand is de standaard instelling is komt dus vrijwel volledig overeen met OM32 modus en is bedoeld voor situaties waarin de OC32 als OM32-compatible apparaat wordt bestuurd.
Als een pin in een andere stand is terechtgekomen kun je hem desgewenst terugzetten in de "automatische" stand door deze tab te kiezen en te klikken op de knop "Set Pin Config" (tijdelijk) of "Write Device" (permanent).

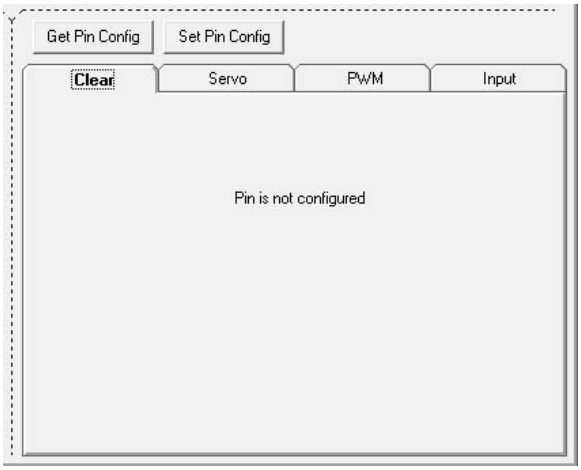


Fig 6: OC32 Pin Config: Automatisch

2.3 Servo mode

De tab "Servo" geeft de mogelijkheid de pin te configureren voor het aansturen van een servo motor en geeft opties voor het instellen van de dynamische parameters

De OC32 kan servomotoren op 2 manieren aansturen:

- Direct, zonder vertraging. De OC32 zet de servo in de gewenste stand. De draaisnelheid wordt bepaald door de servo zelf en is het maximum dat de servo zelf aan kan.
- Dynamisch. Het servogedrag wordt bepaald door een natuurkundig model, vergelijkbaar met een massa aan een veer. Door de parameters van dit natuurkundig model te veranderen kun je zeer uiteenlopend gedrag nabootsen, zoals een "voorwerp" dat met lage snelheid beweegt, uitslingereffecten, etc. Interessant is de zogenaamde "bounce" functie. Je kunt dit vergelijken met een "voorwerp" dat aan het eind van een traject tegen een eindpunt stuitert en na een aantal stuiterbewegingen tot stilstand komt. Omdat het gedrag gebaseerd is op een natuurkundig model, oogt het gedrag zeer natuurlijk.

  Hoewel het op basis van deze beschrijving lastig lijkt de juiste parameters in te stellen valt dat in de praktijk erg mee. Er is een Excel-worksheet waarin het hele model is

uitgewerkt en waar je met de diverse parameters kunt spelen om het gewenste gedrag te bereiken (zie figuur 7). Vul die parameters vervolgens in in OC32Config en de echte servo gedraagt zich identiek.

In de apparaatdefinities zitten ook een aantal "voorgebakken" configuraties voor een aantal standaard-situaties om het nog wat gemakkelijker te maken. Je kunt deze standaarddefinities vervolgens bewerken om het helemaal naar eigen inzicht te optimaliseren.

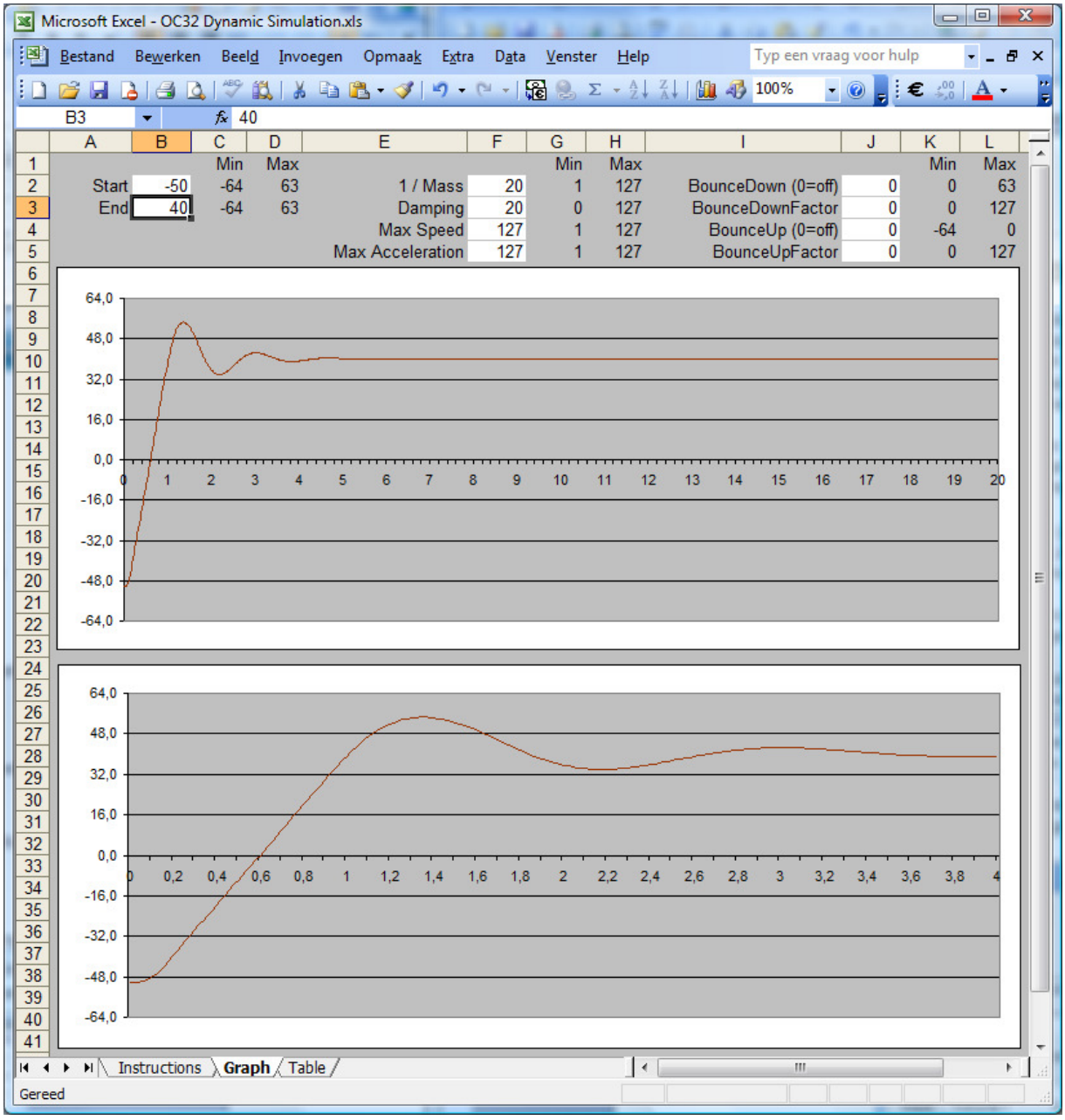


Fig 7: Screenshot van het Dynamisch servo model

Om de servo-parameters te configureren selecteer je het tabblad "Servo" binnen het tabblad "OC32 Pin Config". Hierin kun je de volgende parameters instellen:

Servo range: De servo range stelt het bereik van de servo in. Dat is de maximale uitslag die de servo kan maken. "Off" betekent dat de servo niet wordt aangestuurd. "S"is het kleinste bereik, "XS" het grootste bereik.

Midpoint:	Dit bepaalt de middenpositie van de servo. De mate waarin je het midpoint kunt aanpassen hangt af van de range. Bij een kleine range kun je het midpoint in grotere mate aanpassen dan bij een grote range. Bij range XL heeft een aanpassing van het midpoint geheel geen effect, want dan worden reeds de maximale aanstuurmogelijkheden benut.
1/Mass:	Dit bepaalt de virtuele massa van de "slinger". Hoe kleiner de waarde, hoe groter de virtuele massa. Een grote massa, dus een klein getal, geeft trage bwegingen, een kleine massa, dus een grot getal, leidt tot snelle bewegingen. Een waarde van "0" schakelt het dynamisch gedrag volledig uit.
Damping:	Dit is de virtuele wrijving die de "slinger"ondervindt en bepaalt de snelheid waarmee een slingereffect uitdempt. Hoe groter de wrijving, hoe sneller de slingerbeweging stopt. Handig: Een waarde rond de 58 zorgt voor de snelst mogelijke benadering van de eindpositie zonder uitslingering. Een kleinere waarde dan 58 zorgt voor uitslingeren rondom de eindpositie. Een grotere waarde zorgt ervoor dat de eindpositie niet overschreden wordt, maar dat die positie slechts langzaam bereikt wordt. Hoe groter de waarde, hoe langer het duurt.
MaxSpd:	Hiermee wordt de maximale snelheid beperkt. Noot: dit heeft een ander effect dan een grotere massa
MaxAcc:	Dit bepaalt de maximele versnelling die de beweging kan ondergaan. Wordt voornamelijk gebruikt om het begin van een beweging te vertragen. Pas op met deze parameter: gebruik het simulatiemodel voordat je hem instelt.
BounceDown:	Dit is het bovenste bounce-point, dus het punt dat de bovenzijde van de beweging bepaalt en waar tegen de "massa" naar beneden "stuitert". Indien waarde=0 wordt dit bouncepoint niet gebruikt.
BounceUp:	Dit is het onderste bounce-point, dus het punt dat de onderzijde van de beweging bepaalt en waar tegen de "massa" naar boven "stuitert". Indien waarde=0 wordt dit bouncepoint niet gebruikt.
BounceFactor:	Dit is de hoeveelheid "energie" die behouden blijft bij het "stuiteren" tegen een bouncepoint. De maximale waarde (127) betekent een ideale botsing: het voorwerp stuitert terug met dezelfde snelheid waarmee het het eindpunt naderde. De minimale waarde (0) betekent volledige absorptie van energie: Het voorwerp stopt volledig en stuitert dus geheel niet terug.
Suspended:	Dit vinkje zorgt ervoor dat de servo niet wordt aangestuurd totdat er een initiële positie voor de betreffende servo wordt ontvangen door de OC32.
Initial Position:	Dit is de positie die de servo aanneemt als de OC32 opstart. Initial Position werkt alleen als het vinkje "Suspended" uit staat. Anders wordt de servo bij opstarten geheel niet aangestuurd totdat een eerste positie wordt ontvangen.

Met de slider onderin het vak kun je de werking van de servo daadwerkelijk testen. Hiervoor is het van belang dat je eerst de parameters ook daadwerkelijk activeert (met de "Set Pin Config" knop).

In plaats van het schuifje te gebruiken kun je ook een getal invullen bij "Position" en vervolgens op "<" klikken.

Om dynamisch gedrag te testen moet je het vakje "Dynamic" aanvinken.

Merk op dat wanneer 1/Mass gelijk aan 0 is het niet uit maakt of "Dynamic" aan of uit staat. In dat geval is dynamisch gedrag namelijk uitgeschakeld.

Met de knop "Jump" wordt een eventuele

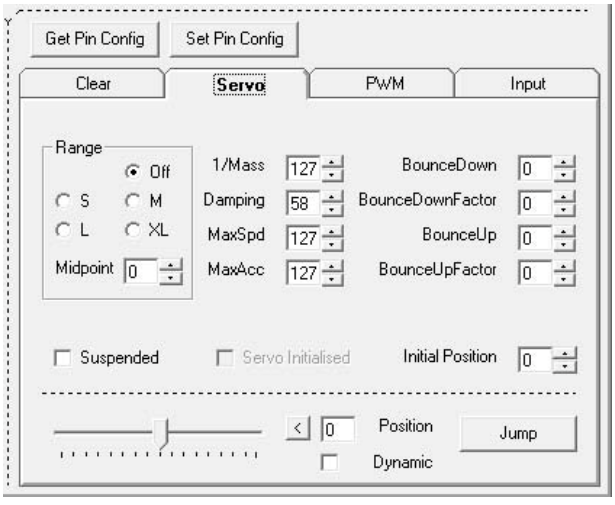


Fig 8: OC32 Pin Config: Servo

dynamische beweging gestopt en gaat de servo direct naar de door “Position” ingestelde stand.

Denk er aan dat je de parameters met “Write Device” naar de OC32 stuurt om ze definitief te maken als je tevreden bent over het resultaat.

Als je de huidige stand (schuifje) wilt gebruiken als initiele positie kun je het getal naast “Position” overtypen in het vak “Initial Position”of dit instellen met de up/down knopjes. Gemakkelijker is het in dit geval om te dubbelklikken op het veld “Initial Position”

2.4 PWM Mode

De tab “PWM” geeft de mogelijkheid de pin te configureren voor het aansturen door middel van PWM. Merk op dat dit ook de mode is die automatisch gekozen wordt als de pin “not congured” is, alleen door een pin expliciet in PWM mode te zetten heb je meer controle over het gedrag.

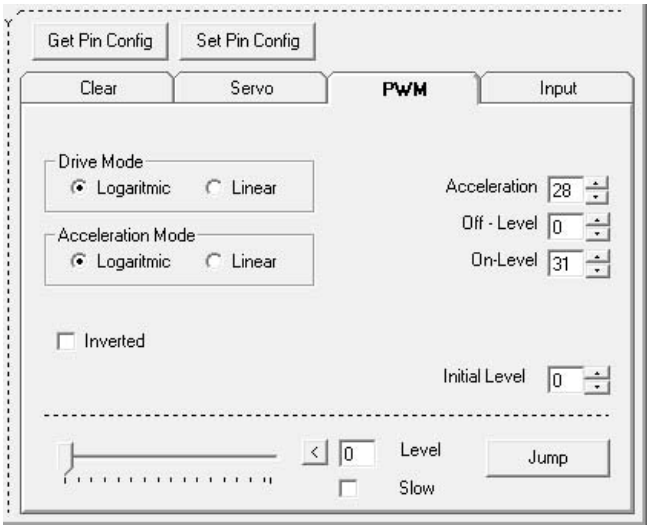


Fig 9: OC32 Pin Config: PWM

PWM staat voor “Pulse Width Modulation” (Pulsbreedtemodulatie). Dit betekent dat een uitgang in zeer hoog tempo wordt aan en uitgeschakeld, waarbij de verhouding tussen de “aan” en de “uit” tijd kan worden gevarieerd. Op die manier is het mogelijk om een aangesloten lamp of LED met een variabele intensiteit te laten branden of een motor met een variabele snelheid te laten draaien. Het prettige van PWM is dat het zeer geschikt is voor gebruik in combinatie met digitale elektronica en dat het zeer efficiënt omgaat met energie. Nadelen zijn er voor onze toepassingen niet of nauwelijks, mits de schakelfrequentie voldoende hoog is (zo hoog dat het oog de schakelfrequentie niet meer kan volgen)

Onder PWM mode heb je de volgende instelmogelijkheden:

Drive Mode: Je kunt deze instellen op lineair of logaritmisch.
De OC32 kent per uitgang 32 intensiteitsstappen. Met deze parameter bepaal je hoe de “stappentabel” wordt vertaald naar een uitgangsniveau. Bij de instelling “lineair” is stap 8 ruwweg 25%, stap 16 ruwweg 50% en stap 24 ruwweg 75%. Deze instelling gebruik je bijvoorbeeld voor het aansturen van een elektromotor.
Bij de instelling “logaritmisch” heeft de “grafiek” een logaritmisch verloop. De reden hiervoor is dat het menselijk oog ook een soort logaritmische gevoeligheid kent voor licht. Een LED die 2x zo veel energie uitstraalt ziet er niet 2x zo helder uit. Wil je een LED optisch 50% dimmen, dan moet je het vermogen wel tot iets van 15% a 20% reduceren. Stuur je een LED aan met een logaritmische karakteristiek, dan zal stap 16 er ongeveer uit zien als 50% gedimd, maar de aansturing is in dat geval dus aanzienlijk minder.
Gloeilampjes hebben van zichzelf al een soort logaritmische karakteristiek. De eerste hoeveelheid energie die je er in stopt gaat uitsluitend “verloren” in de vorm van warmte. Om een gloeilamp 50% van het licht te laten geven dat hij uitstraalt bij vol vermogen moet je ruimschoots meer dan 50% van dat volle vermogen er in stoppen. Gloeilampjes stuur je dus bij voorkeur ook aan in “Linear Drive Mode”.

Accell. Mode: Je kunt deze instellen op lineair of logaritmisch.
De "acceleration" staat voor het geleidelijk aannemen van een nieuw ingestelde waarde. Een gloeilamp die wordt aangezet is niet meteen aan, maar heeft even tijd nodig om op temperatuur te komen. Evenzo is een gloeilamp die wordt uitgezet niet meteen uit, maar het duurt even voor hij is afgekoeld. Dit aan –en nagloeien kennen LEDs niet. Wil je een LED laten aan en uit gaan alsof het een gloeilamp is, dan kun je dat met de OC32 softwarematig doen. De karakteristiek waarmee een gloeilamp aan –en nagloeit is logaritmisch.
Wil je een motor langzaam laten aanlopen, dan is de versnelling die je wilt meestal lineair. Kies in dat geval voor "Linear Acceleration Mode".

Acceleration: Dit is de snelheid waarmee de "acceleration" plaatsvindt. Het getal dat je hier invult is een index. Hoe groter het getal, hoe sneller het gaat. Index 1 is dus extreem langzaam, index 31 is zeer snel. De relatie tussen index en "full-swing" acceleratie (van 0 tot 100%) vind je in figuur 10.
Index 0 betekent dat "acceleration" volledig is uitgeschakeld. Je kunt dit desgewenst gebruiken bij uitgangen die altijd stapsgewijs moeten worden aangestuurd, bv een uitgang voor het schakelen van een relais.

Inverted: Indien deze optie is geselecteerd wordt de uitgang omgekeerd aangestuurd. Dit is bv handig als je een 5V uitgang hebt en LEDs aanstuurt die een gemeenschappelijke anode (+) hebben. De LED zit dan tussen +5V en een uitgang. Je zou dit uiteraard ook kunnen oplossen door de uitgang precies omgekeerd aan te sturen, alleen het vervelende is in dat geval dat de logaritmische stuurkarakteristiek dan precies verkeerd-om is. Het inverted vinkje lost het "probleem" dus op een andere manier op, die wel correct werkt. Inverted mode is ook handig in een aantal uitzonderingsgevallen, zoals het met aan –en nagloeieffecten aansturen van 3 knipperlichten op spoorwegbomen met slechts 2 draadjes. Die theorie zullen we hier niet behandelen, maar het aansluitschema vind je wel in de handleiding (paragraaf **Fout! Verwijzingsbron niet gevonden.**) en hiervoor is een standaard apparaatdefinitie gemaakt.

Off/On-Level: Met ingang van release 0.0.2.0 is het aantal "random functies" van de OC32 sterk uitgebreid. Zo kun je een uitgang met een bepaalde kans aan of uit zetten. Door het instellen van het "On" -en "Off" Level kun je hier nog meer variatie in aanbrengen. Als de kansberekening aangeeft dat de uitgang "uit" moet zijn wordt de waarde aangenomen die is ingevuld bij "Off-Level". Als de kansberekening aangeeft dat de uitgang "aan" moet zijn wordt de waarde aangenomen die is ingevuld bij "On-Level". Een voorbeeld van wat je hiermee kunt maken is het simuleren van het flakkereffect van gaslantaars.

Met het schuifje onderin het vak kun je de werking daadwerkelijk testen. Hiervoor is het van belang dat je eerst de parameters ook daadwerkelijk activeert (met de "Set Pin Config" knop).

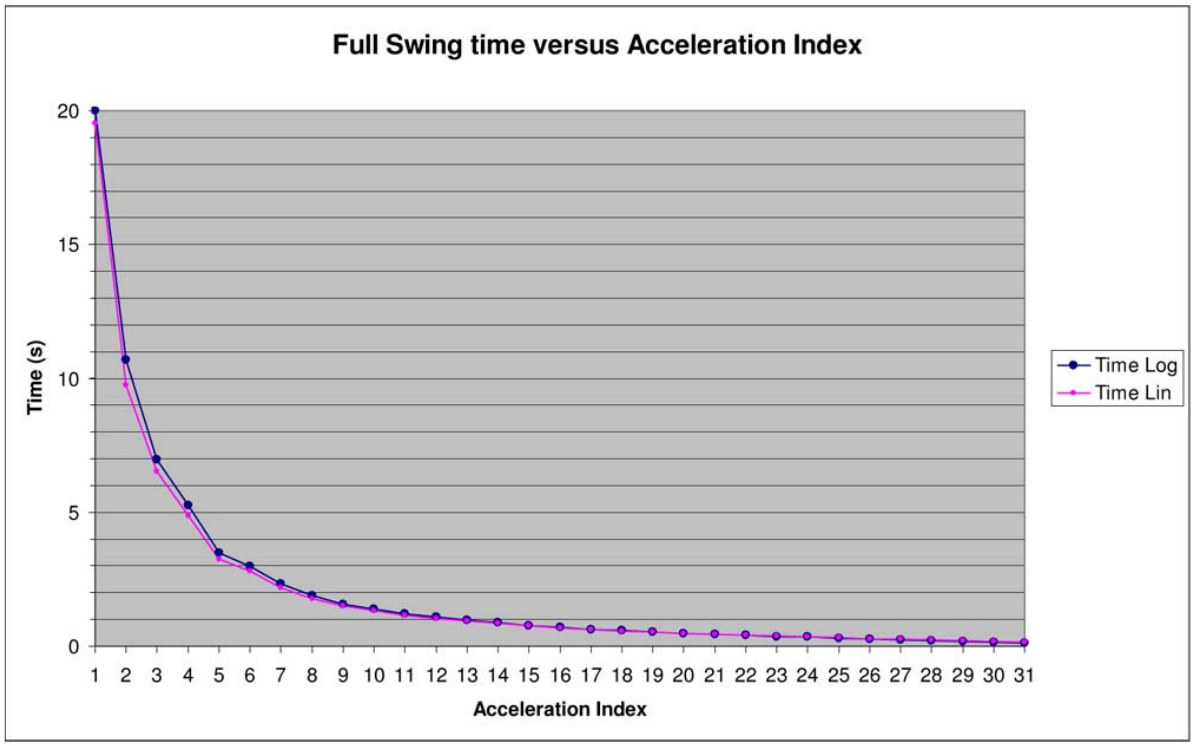
In plaats van het schuifje te gebruiken kun je ook een getal invullen bij "Level" en vervolgens op "<" klikken.

Als je de dynamische parameters wilt testen (acceleratie) dien je het vinkje "Slow" bij de slider gezet aan te zetten. Anders wordt het nieuwe niveau direct aangenomen.

Met de knop "Jump" wordt eventueel dynamisch gedrag gestopt en gaat de uitgang direct naar de ingestelde waarde..

Denk er aan dat je de parameters met "Write Device" naar de OC32 stuurt om ze definitief te maken als je tevreden bent over het resultaat.

Als je de huidige stand (schuifje) wilt gebruiken als initieel niveau kun je het getal naast "Level" overtypen in het vak "Initial Level"of dit instellen met de up/down knopjes. Gemakkelijker is het in dit geval om te dubbelklikken op het veld "Initial Level"



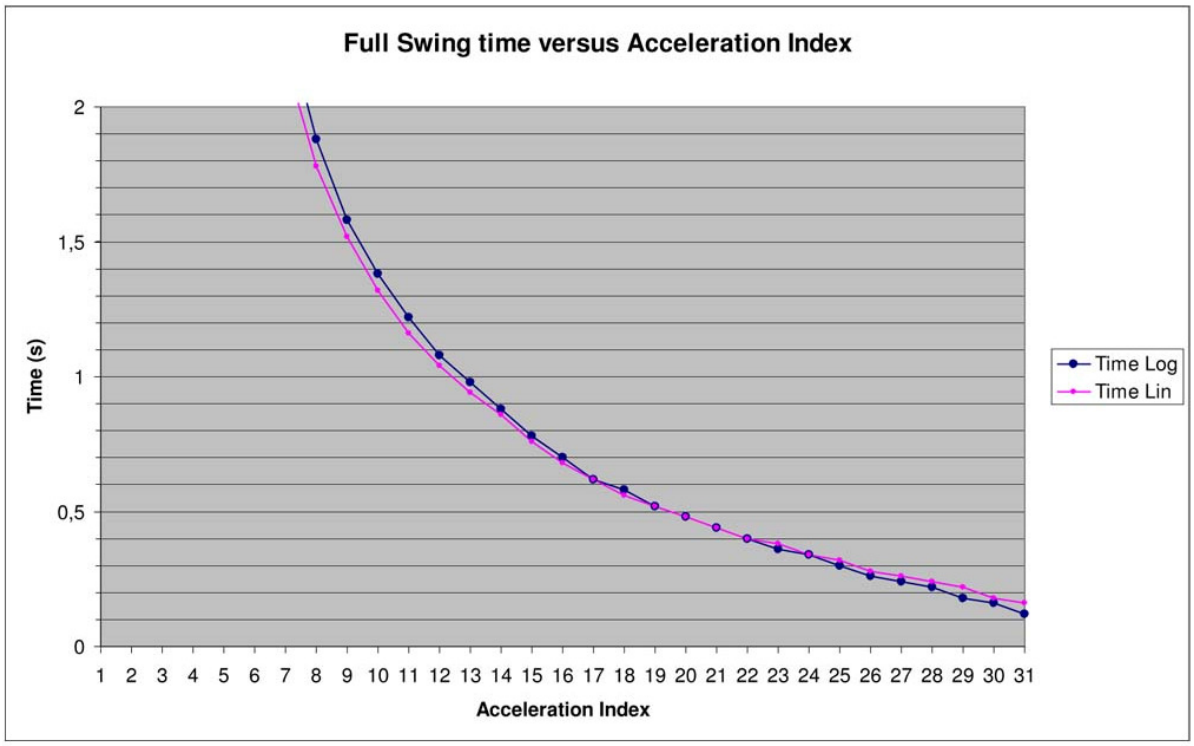


Fig 10: Full Swing Time als functie van de Acceleration Index
Noot: Beide grafieken zijn identiek, allen de verticale schaal is verschillend

2.5 Input mode

De tab “Input” geeft de mogelijkheid de Pin te configureren als ingang.

Een Pin die als ingang wordt gebruikt kan twee toestanden aannemen: Actief (ON) en inactief (OFF). De spanning op de ingang kan elke spanning zijn tussen 0V en +5V. Een spanning tussen 0,8V en 3,5V is in principe ongedefinieerd. Een lage spanning is dus lager dan 0,8V, een hoge spanning is hoger dan 3,5V

Je kunt configureren hoe de spanning op de ingang geinterpreteerd moet worden.

- Hoge spanning = actief, lage spanning = inactief (“High Active”) of
- Lage spanning = actief, hoge spanning = inactief (“Low Active”)

Als je “Low Active” geselecteerd hebt zal de OC32 ervoor zorgen dat de ingang een hoge spanning voert (dus inactief is) als er niets op is aangesloten. Dat gebeurt met een zwak stroompje van enkele tienden mA. Je kunt dan bv een drukschakelaar aansluiten tussen de Pin en GND zonder dat je andere componenten nodig hebt.

Als je “Low Active” niet selecteert (dan is het dus High Active), dan zweeft de ingang als er niets op is aangesloten. De toestand is dan ongedefinieerd.

Schakelaars hebben de neiging te “denderen” of “klapperen” als ze geopend of gesloten worden. Om te voorkomen dat het openen en sluiten van een schakelaar leidt tot heel veel toestandswijzigingen achter elkaar zit er een “debounce” functie in de OC32. Je kunt de debounce tijd instellen tussen 0 en 31.

- OFF-delay is de tijd die de OC32 wacht voordat de toestand van een ingang als Inactief wordt gezien.
- ON-delay is de tijd die de OC32 wacht voordat de toestand van een ingang als Actief wordt gezien.

Een (zeer) korte delay kan leiden tot klapperen. Een lange delay leidt er toe dat een vertraging in de toestandswijziging optreedt en bijvoorbeeld heel korte pulsen (het zeer kort activeren van een schakelaar) wordt gemist. In sommige gevallen kan dat echter ook weer een handige functie zijn.

- Indien de toestand wijzigt van Inactief naar Actief wordt Aspect 1 geactiveerd van de betreffende Pin
- Indien de toestand wijzigt van Actief naar Inactief wordt Aspect 0 geactiveerd van de betreffende Pin

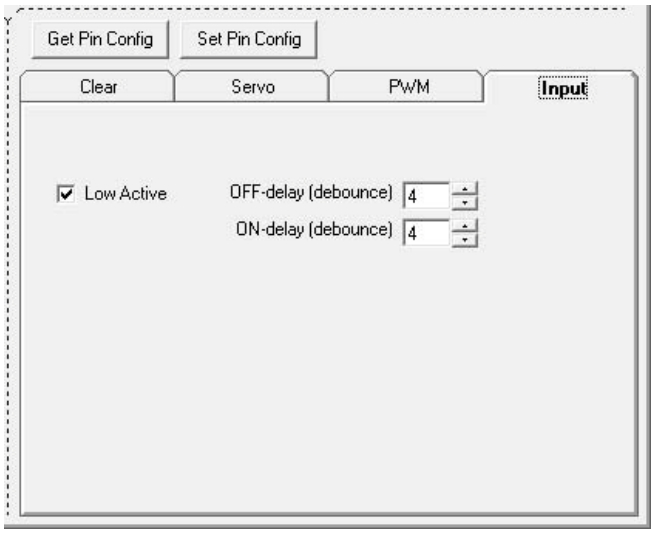


Fig 11: OC32 Pin Config: Input

3 Aspect Configuratie

3.1 Principes

Onder een "Aspect" wordt verstaan: een stand van een apparaat. Zo kan bijvoorbeeld een NS sein met 3 lampen verschillende standen tonen:

- Rood
- Groen
- Geel
- Geel knipper
- Groen knipper

Het betreft hier dus een sein met 5 standen (5 "Aspects"). De Aspects die gelden voor een apparaat worden in principe gekoppeld aan één Pin. We noemen dit de "Device Control Pin", of simpelweg de "Device Pin".

Elke Aspect in de OC32 bevat een of meerdere "Instructies". Die Instructies bepalen wat de OC32 doet met de Pinnen die behoren tot het apparaat. Een Instructie kan iets doen met één Pin, maar sommige Instructies kunnen dat ook met meerdere Pinnen tegelijk. Een voorbeeld daarvan is de "Multibit" instructie, die een reeks opeenvolgende Pinnen in een patroon kan schakelen. Om het bovengenoemde NS sein in de stand Groen te zetten heb je maar één Instructie nodig, namelijk een "Multibit3, 010". Multibit3 wil zeggen: Schakel 3 opeenvolgende Pinnen en de 010 zegt: De eerste Pin uit, de tweede aan, de derde uit. Als je het sein aansluit zoals de apparaatdefinitie voorschrijft, namelijk rood op de eerste pin, groen op de tweede en geel op de derde toont het sein dan ook daadwerkelijk Groen.

Niet alleen kan een Instructie werken op een reeks Pinnen, maar veel Instructies kunnen ook werken op een andere Pin dan de Device Control Pin waaronder de Instructie is opgeslagen. Het laten werken op een andere Pin dan "de eigen Pin" doen we met een "Pin Offset" (P.O.). Een Instructie die het NS sein op Geel knipper zet is "Inverteer" met een "Pin-Offset" van 2 en een "Periode" van 23/60 seconden. Als die Instructie is opgeslagen onder de Device Control Pin, dat is de eerste, dus die van de rode lamp, dan laat deze instructie de Pin knipperen die 2 posities verder zit (geel).

De reden dat gewerkt wordt met een (relatieve) Pin Offset en niet met een absoluut Pin-nummer is dat je op deze manier apparaatdefinities universeel kunt maken en eenvoudig kunt kopiëren. De instructie die de gele lamp van een NS sein laat knipperen kun je simpelweg 3 pinnen verderop kopiëren voor het volgende NS sein en die instructie werkt dan nog steeds.

Een Aspect definitie kan meerdere Instructies bevatten. Die instructies worden dan allemaal direct achter elkaar uitgevoerd of met een vertraging. Dat hangt af van de Instructie, daarover later meer.

Om het NS sein op geel knipperen te zetten gebruik je twee Instructies die direct achter elkaar worden uitgevoerd:

- Multibit3, PO=0, Patroon = 000
- Invert, PO=2, Tijd =23/60

De eerste instructie zet alle lampen uit, anders zou de rode lamp van een sein dat op rood stond blijven branden, de tweede instructie laat de gele lamp knipperen.

3.2 Aspects, Instructies en Parameters

Aspect Configuratie vind je onder de tab "OC32 Pin Configuration". Binnen deze tab moet je het vinkje "Show Details" activeren. Het gedeelte binnen de stippellijn linksonder is het gedeelte waarin je de Aspect Configuratie bepaalt.

Per Pin kun je kiezen hoeveel Aspects die Pin moet kunnen aansturen:

- 0 Aspects (uitgeschakeld)
- 4 aspects met 9 instructies per aspect

- 12 aspects met 3 instructies per aspect

Als je de instelling “12 Aspects” of “4 Aspects” selecteert krijg je er nog een paar zaken bij:

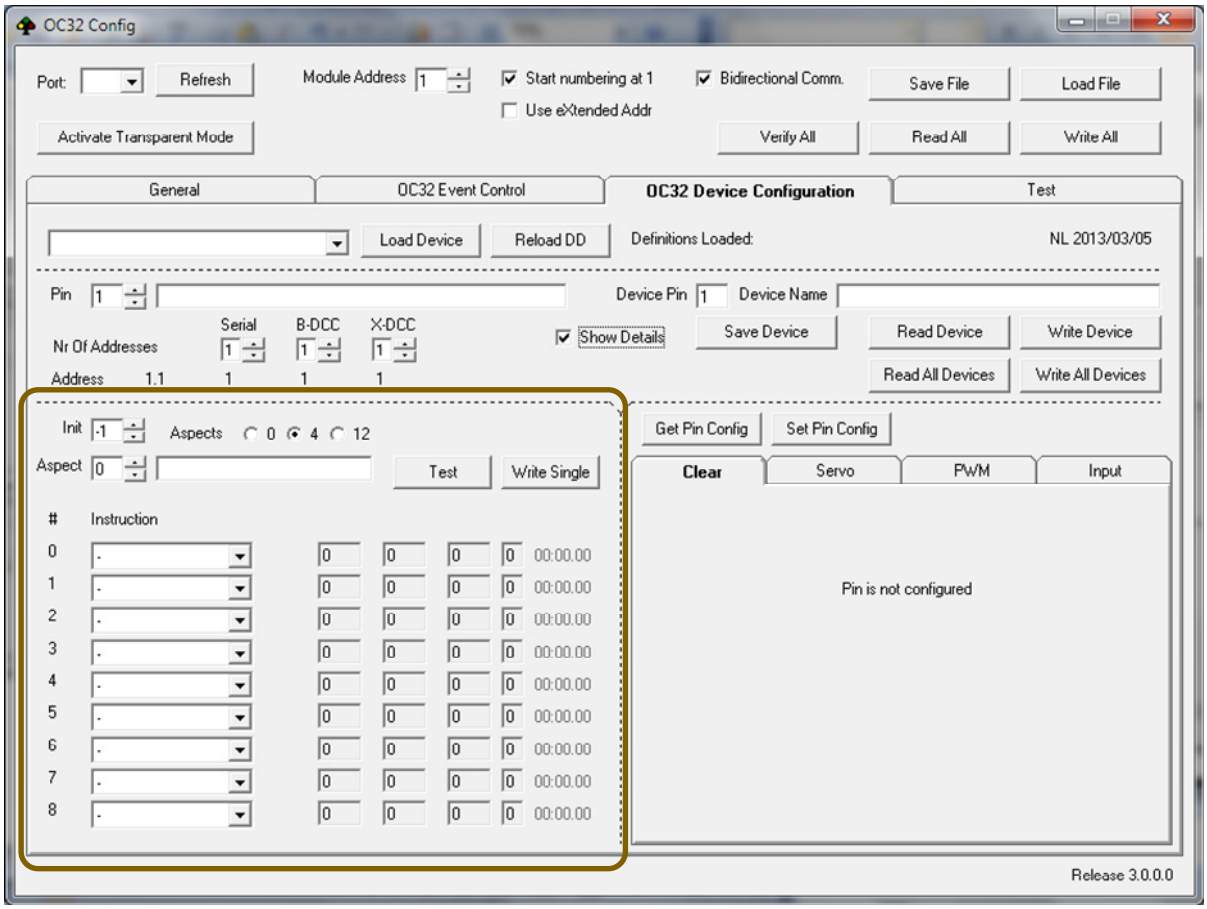


Fig 12: OC32 Aspect Configuratie

- Een vakje om het Aspect (0..3 of 0..11) te selecteren dat je aan het bewerken bent
- Een vak met de beschrijving van het betreffende Aspect. Doet op zich niets, maar wordt wel opgeslagen als je een bestand opslaat en geladen als je het bestand inleest. Het is dus handig voor je eigen administratie;
- Een vakje om de initiële Aspect (0..3 of 0..11) te selecteren die gekozen wordt als de OC32 opstart (-1 = geen initialisatie).
- De knop “Write Single”: Hiermee kun je de instructies behorende bij één enkele aspect opslaan in de OC32. Handig om te testen als je slechts 1 aspect hebt aangepast.
- De knop “Test”: Hiermee kun je de werking van de betreffende Aspect testen, uiteraard nadat je hem eerst hebt opgeslagen in de OC32.
- Een reeks instructies, 3 of 9 stuks, met parameters

Voor de goede orde: De knoppen “Write Single” en “Test” werken niet (correct) voordat je eerst de apparaatdefinitie door middel van een “Write Device” hebt opgeslagen in de OC32. Meer hierover in hoofdstuk 4.

De instructies dien je te vullen van boven naar beneden, dus beginnend bij 0. De instructies worden bij het activeren van de betreffende Aspect op volgorde uitgevoerd totdat een “lege” instructie wordt bereikt of de laatste instructie is uitgevoerd. Er mogen dus geen “gaten” in de reeks instructies zitten. Per instructie tref je 8 velden aan, die afhankelijk van de gekozen instructie actief worden en afhankelijk van de gekozen instructie een andere betekenis kunnen hebben.

#:	Hieronder staat het volgnummer van de instructie, te beginnen bij 0
Instruction:	Hier selecteer je de gewenste instructie uit een drop-down list. Een gedetailleerde omschrijving tref je hieronder aan. Afhankelijk van de gekozen instructie worden een of meerdere velden rechts hiervan actief.
Fade:	Als dit vinkje geactiveerd is wordt de betreffende functie uitgevoerd met dynamisch effect, dus aan/nagloeien bij PWM en dynamisch gedrag bij een servo opdracht. In sommige gevallen heeft dit vinkje een afwijkende betekenis (zie onderstaande beschrijvingen)
Param0..3:	4 parameters waarvan de betekenis varieert afhankelijk van de gekozen instructie (zie onderstaande toelichting)
00:00.00	Weergave van tijd in minuten en seconden, op 2 decimalen. Wordt weergegeven indien er instructie is gekozen die de Event Timer zet.

Param 0..3 zijn parameters waarvan de betekenis varieert, afhankelijk van de gekozen instructie. Boven de betreffende velden staan labels. Die labels worden getoond voor die regel waar in je een veld hebt geselecteerd. Dus stel je klikt op het eerste parameterveld in de regel van instructie 2, dan worden de labels getoond die horen bij de instructie die je gekozen hebt op die regel.

Onderstaand een uitleg wat je zoal tegen kunt komen. Meer uitleg vind je bij de beschrijving van de instructies in onderstaande paragrafen.

P.O.:	Staat voor "Pin Offset". Dit is de pin (uitgang) ten opzichte van de huidige (geselecteerde) pin waarop de instructie werkzaam is. Het bereik van de offset is afhankelijk van het geselecteerde pinnummer en loopt maximaal van -31 tot +31
T/60:	Tijd in eenheden van 1/60 seconde (0..127).
Asp#:	Aspect nummer, kom je tegen bij "Jump"
Ins#:	Instructie nummer, kom je tegen bij "Jump"
Type:	Typeselectie.
Patt:	Pattern (Patroon). Komt voor bij Multibit instructies en geven het bitpatroon waarin de opeenvolgende uitgangen worden geschakeld. Dubbelklik op het veld voor een handige invulhulp.
Level:	Een PWM niveau (0..31). 0 = 0%, 31 = 100%
Pos.:	Servo Positie (-64..63).
ETImap:	De parameter om diverse acties uit te voeren op Event Inputs. Dubbelklik op de parameter voor een invulhulp.
Accel.:	Acceleratie Index (-1..31). "-1" Betekent: Kies de Default Acceleration Index ingesteld bij de Pin Configuratie. Elke andere waarde is een expliciete Acceleration Index.
Chance:	Kans bij kansberekeningen (0..31). 0 = 0% en LET OP: 31 = 97%. Wil je 100%, kies dan de "Level" variant van de instructie
Source:	Pinnummer (0..31) waar vandaan gekopieerd wordt. Komt voor bij "Clone"
Time/Base:	Deze velden zijn actief bij zogenaamde "Timed Instructions" en "Sequence Instructions", dat zijn instructies waarbij een timer gezet wordt die het uitvoeren van een actie op een later tijdstip bewerkstelligt. Zie paragraaf 3.4 t/m 3.6 voor meer informatie. "Time" is de tijdsduur van de timer (0..63) "Base" bepaalt de eenheden waarin "Time" werkt: 0 = 0,02 sec 1 = 0,16 sec 2 = 1,28 sec 3 = 10,24 sec De kortste tijd kan dus 1 x 0,02 sec = 0,02 sec zijn De langste tijd is 63 x 10,24 sec = 10 minuten en 45.12 sec Door te dubbelklikken op "Time" of "Base" krijg je een

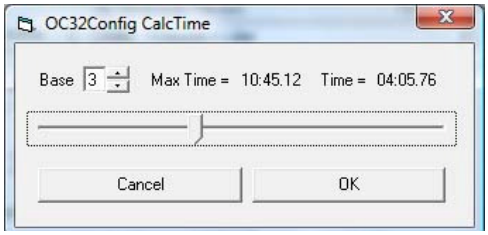


Fig 52: Hulpscherm voor het instellen van Time en Base

hulpscherm waarmee je deze parameters redelijk handig kunt instellen.

3.3 Aspect Configuratie - Normale Instructies

Normale instructies worden zonder vertraging uitgevoerd. Nadat de uitvoering klaar is wordt zonder vertraging de volgende instructie uitgevoerd.

Jump: Ga met de uitvoering verder bij Instructie "Ins#" van Aspect "Asp#" van Pin <huidige pin + Pin Offset>.
Deze instructie springt dus naar een andere Instructie van (eventueel) een andere Aspect van (eventueel) een andere Pin. De rest van de instructies (die achter de "Jump" staan) worden dus niet meer uitgevoerd.

NS G->Y->R: Dit is een speciale instructie waarmee een NS sein op rood wordt gezet. De instructie gaat er van uit dat de lampen zijn aangesloten in de volgorde rood-groen-geel-(cijfer) . Als de huidige stand van het sein (een variant van) groen is, wordt het sein eerst kort op geel gezet en daarna op rood.
Type = 0 voor een 3-lichts sein
Type = 1 voor een 3-lichts sein met cijferbak

Blink alt: Laat 2 opeenvolgende uitgangen knipperen met tijdsperiode <T/60> = halve periode in eenheden van 1/60 sec met een maximum van 127. Alternerend wil zeggen: als de ene uitgang aan is, is de andere uit.

Clone: **Clone is een lastige en ietwat verouderde instructie, Wat je met Clone kunt is vrijwel altijd ook met andere instructies te bereiken. Advies is deze instructie niet te gebruiken voor nieuwe apparaatdefinities aangezien deze instructie in een toekomstige release wel eens kan verdwijnen.**
Hiermee kan het gedrag van één uitgang exact worden gekopieerd naar een andere uitgang. <Source> Is de Pin (dus NIET de Pin-offset) waar vandaan gekopieerd wordt. 0 Is de eerste Pin van de OC32 module (de nummering is dus absoluut vanaf nul).
Let op (1): Het "Fade" bit heeft een afwijkende betekenis:
Indien het Fade bit NIET actief is wordt de acceleratieparameter meegekopieerd van bron naar doel.
Indien de doel-Pin, dus de pin aangegeven door het Pin nummer + Pin Offset, NIET geconfigureerd is voor PWM (dus in automatische mode staat) wordt de acceleratieparameter meegekopieerd van bron naar doel.
Indien het Fade bit WEL actief is EN de doel-Pin WEL geconfigureerd is voor PWM wordt de voor die pin geconfigureerde acceleratie-parameter gebruikt. In dit geval wordt deze dus niet mee gekopieerd.

Off: Zet uitgang "uit" gedurende tijd = <T/60> in eenheden van 1/60 sec met een maximum van 127. <T/60> = 0 is permanent uit, elke andere waarde geeft dus een "puls laag".

On: Zet uitgang "aan" gedurende tijd = <T/60> in eenheden van 1/60 sec met een maximum van 127. <T/60> = 0 is permanent aan, elke andere waarde geeft dus een "puls hoog".

Invert: Inverteer uitgang na elke opgegeven tijd = <T/60> in eenheden van 1/60 sec met een maximum van 127. <T/60> = 0 is éénmalig, elke andere waarde zorgt dus voor knipperen met <T/60> = halve periode tijd.

Random: Geef uitgang periodiek, na elke tijd = <T/60> een willekeurige waarde (aan of uit). De tijd is in eenheden van 1/60 sec met een maximum van 127. <T/60> = 0 is éénmalig, elke andere waarde zorgt dus voor een willekeurig patroon met tijd = <T/60> als tijdsinterval.

Multibit (n): Zet (n) opeenvolgende uitgangen, startend bij de gespecificeerde uitgang volgens het in <Patt> opgegeven patroon. Het minst significante bit van

<Patt> vertegenwoordigt de eerste uitgang (m.a.w: de bits 0 t/m n-1 in <Patt> worden van rechts naar links afgewerkt voor de opeenvolgende uitgangen). Dit mechanisme werkt modulo 32, dus bv een Multibit 3 op uitgang 32 zet de uitgangen 32, 1 en 2 volgens het patroon in <Patt>. Bij het wijzigen van het patroon worden eerst alle bits die uit moeten gaan uit gezet, vervolgens worden na 1/3 seconde alle bits die aan moeten gaan aan gezet.
Bij Multibit 2 t/m 5 worden de bits 5 en 6 van het <Patt> niet gebruikt voor definitie van het patroon, maar hebben een speciale, aanvullende functie:

- Bit 5 = 1 (alldark): Het beeld wordt eerst volledig gedoofd, dus ook de bits die aan moeten blijven.
- Bit 6 = 1 (nodark): Het beeld wordt ingesteld zonder de extra vertraging: er is dus geen donkerfase.

Bij Multibit 6 en 7 zijn deze bits nodig voor de patroondefinitie en zitten deze opties verpakt in het commando-byte zelf.
Je hoeft het patroon niet zelf binair te berekenen . Dubbelklik op "Patt" en je krijgt een invulhulp.

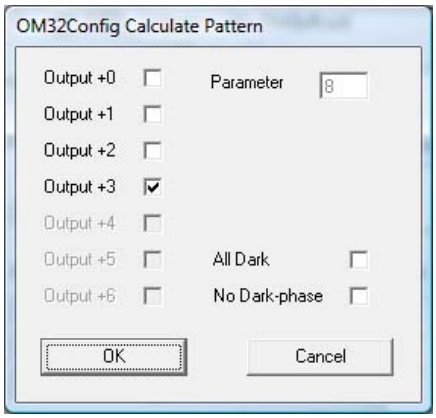


Fig 13: Hulpscherm voor instellen van het patroon

Set Level: Stelt de uitgang <Pin + Pin Offset> in op de waarde <Level>. In tegenstelling tot de Level Log/Lin en Level-Pulse Log/Lin kun je hier de volledige range van 32 stappen gebruiken (0..31).

Set Servo: Zet de de servo op uitgang <Pin + Pin Offset> in stand <Pos> (-64..+63). Voorwaarde is dat de Pin geconfigureerd is als Servo.

Multiblink (n): Test de (n) opeenvolgende uitgangen of deze aan of uit zijn. Indien de uitgang aan is wordt deze op knipperen gezet met het in <T/60> opgegeven tijdsinterval (halve periodetijd) in eenheden van 1/60 sec (max 127). Indien de uitgang uit is blijft deze uit. Deze functie wordt primair gebruikt om meerdere lampen van een seinbeeld synchroon op knipperen te zetten, zoals bij Belgische seinen het geval is.
Alleen indien de uitgang volledig uit is (zonder actieve Periode-Timers, zie paragraaf 3.6) wordt deze bij de test als "uit" gezien, in alle andere gevallen als "aan".

Event Input: Hiermee kun je diverse acties uitvoeren die te maken hebben met Event Trigger Inputs. Dubbelklik op het invulveld onder het label ETImap voor een invulhulp.
De volgende mogelijkheden worden geboden:

- Het enablen (toestaan) van meerdere on-events
- Het disablen (uitschakelen) van meerdere on-events
- Het enablen van meerdere off-events
- Het disablen van meerdere off-events
- Het uitvoeren van meerdere on-events
- Het uitvoeren van meerdere off-events

Het uitvoeren van een on/off-event, ook wel een "Soft-Event" genoemd, betekent dat alle acties worden uitgevoerd alsof het event daadwerkelijk zou hebben plaatsgevonden. Het is een bijzonder krachtige manier om in de OC32 veel acties tegelijk op te starten teneinde de module in een bepaalde toestand te

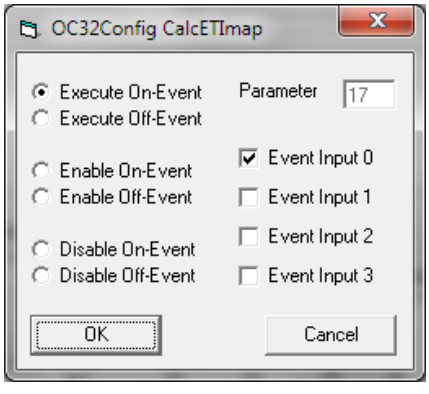


Fig 14: Hulpscherm ETImap

	krijgen zonder dat daarvoor heel veel instructies nodig zijn. Voor het uitvoeren van een "Soft-Event" is het niet nodig dat de ETI inputs ook fysiek aanwezig zijn. Het enablen en disablen van events heeft alleen invloed op de echte, elektrische event-inputs. Ale een event disabled is kan het door middel van een Soft-Event dus nog gewoon worden uitgevoerd. Voordat een Soft-Event daadwerkelijk wordt uitgevoerd wordt eerst de lopende uitvoering van instructies afgemaakt.
SendSerial:	Zet één of twee bytes (0..255) in het verzendbuffer voor de Seriele Accessoire Poort. Als er reeds bytes in het buffer staan worden de nieuwe er gewoon achter gezet. Verzending vindt alleen plaats als de SAP enabled is.
Feedback:	Stuurt een bericht naar de host (Dinamo systeem, PC programma) dat een Aspect van een bepaalde Pin is "afgevuurd". Het verdient voor enige overzichtelijkheid sterk aanbeveling P.O. op 0 te laten staan en Aspect gelijk te houden aan de Aspect waaronder de instructie is opgenomen, maar voor creatieve oplossingen kun je hiervan afwijken. De Feedback instructie wordt typisch gebruikt om een Input terug te melden aan de besturingssoftware. Feedback-berichten worden altijd in het verzend-buffer gezet voor de RS485 poort van de module. De uitvoering van de instructies gaat direct door bij de volgende instructie.
MXpulse:	MX staat hier voor MultipleX. Deze instructie wordt gebruikt om de wisselspoel-multiplexer aan te sturen. Voor wisselspoel-multiplexing is een separate handleiding beschikbaar.

3.4 Aspect Configuratie - Timed Instructies

Let op: Timed Instructies en Sequence Instructies maken gebruik van een "Event Timer". Zie paragraaf 3.6 voor een nadere uitleg en de beperkingen!

Een Timed Instructie voert nooit direct een actie uit, maar zet slechts de Event Timer. Direct volgend op het zetten van de Event Timer wordt de volgende instructie uitgevoerd. Een Timed Instructie met Time = 0 heeft als effect dat de Event Timer van de betreffende Pin wordt gereset en een eventueel wachtende instructie wordt gewist.

SetAspect(n):	Voer Aspect (n) uit van Pin <huidige Pin + Pin Offset> na <Time>*<Base>.
Suspend Servo:	Stopt de aansturing van de servo op <huidige Pin + Pin Offset> na <Time>*<Base>.

3.5 Aspect Configuratie - Sequence Instructies

Let op: Timed Instructies en Sequence Instructies maken gebruik van een "Event Timer". Zie paragraaf 3.6 voor een nadere uitleg en de beperkingen!

Sequence Instructies zijn instructies waarmee LichtOrgel functies kunnen worden uitgevoerd plus nog heel wat meer. Deze "Sequence Instructies" voeren "iets" uit en wachten vervolgens een te definiëren tijd alvorens de volgende instructie in de lijst aan te roepen.
Sequence Instructies en "normale" instructies kunnen gewoon willekeurig door elkaar worden gebruikt. Juist dit geeft veel meer creatieve mogelijkheden dan wat er mogelijk is met LichtOrgel.

Voor het wachten tussen de instructies wordt de "Event Timer" van de betreffende pin gebruikt. M.a.w.: Sequence Instructies overschrijven een eventueel reeds actieve Event Timer voor de betreffende pin.

Sequence Instructies werken alleen op de eigen pin, m.a.w. je kunt met een Sequence Instructie geen andere pin aansturen dan die waaronder de instructie zelf is opgeslagen

Set Level & Wait (A,L,T,B):
Stel Niveau L (0..31) in en wacht tijd <Time>*<Base> tot de volgende stap.
Bij het wijzigen van het niveau wordt Accelaratie-Index A gebruikt (-1..31). A = -1 betekent dat de Default Acceleration Index gebruikt wordt, zoals gedefinieerd bij de Pin Configuratie (PWM tab). Andere waarden (0..31) geven een expliciete Accelartion Index.

Set Random & Wait (A,C,T,B):
Zet uitgang aan met kans C (0..31)/32 of uit met kans (1-C) en wacht tijd <Time>*<Base> tot de volgende stap.
Let op: Kans 0 = 0%, kans 31 = 97%. Wil je 100% kans gebruik dan Set Level & Wait.
Als de kansberekening uitwijst dat de uitgang “aan” moet zijn wordt het niveau op het “On-Level” gezet dat bij de Pin Configuratie is ingesteld (PWM tab).
Als de kansberekening uitwijst dat de uitgang “uit” moet zijn wordt het niveau op het “Off-Level” gezet dat bij de Pin Configuratie is ingesteld (PWM tab)
Bij het wijzigen van het niveau wordt Accelaratie-Index A gebruikt (-1..31). A = -1 betekent dat de Default Acceleration Index gebruikt wordt, zoals gedefinieerd bij de Pin Configuratie (PWM tab). Andere waarden (0..31) geven een expliciete Accelartion Index.

Set Servo & Wait (P,T,B):
Stel servo in op positie P (-64..63) in en wacht tijd <Time>*<Base> tot de volgende stap.

Wait (T,B): Doe niets en wacht tijd <Time>*<Base> tot de volgende stap.

Set Level & Wait Random (A,L,T,B):
Set Random & Wait Random (A,C,T,B):
Set Servo & Wait Random (P,T,B):
Wait Random (T,B):
Als bovengenoemde instructies. Alleen is de wachttijd tot de volgende stap nu geen vaste tijd, maar een willekeurige tijd tussen 20ms en de tijd <Time>*<Base>.
Het aantal stappen waarmee de willekeurige tijd berekend wordt is 256. Er zijn dus 256 mogelijke uitkomsten van de random berekening.
De berekende tijd is in eenheden van 20ms. Als <Time>*<Base> erg klein is zijn er dus minder mogelijke uitkomsten. In een extreem geval: Stel dat <Time>*<Base> 40ms zou zijn, dan heb je slechts 2 mogelijke uitkomsten: 20ms of 40ms

Als de laatste instructie (afhankelijk van de "Aspect Mode" dus instructie 2 of 8) een Sequence Instructie is, zal de eerste instructie die na de "wait" wordt uitgevoerd Instructie 1 zijn (dus niet instructie 0). Dit is vergelijkbaar met LichtOrgel waar stap 0 een initialisatiestap is en de overige stappen cyclisch.
Een sequence instructie met T=0 reset de Event Timer voor de Pin. Het staat dus gelijk aan een "Halt" na die instructie, maar zo'n instructie (bv Wait(0)) kan ook gebruikt worden om zeker te stellen dat een lopende Event Timer gereset wordt. Als je dit zo gebruikt moet het dus wel de laatste instructie zijn, want daarna stopt de uitvoering van de reeks.

3.6 Event Timers en Periode Timers

Timed Instructies en Sequence Instructies maken gebruik van een “Event Timer”.

Er is één (1) Event Timer per Pin. Dat wil zeggen dat er per Pin slechts 1 gebeurtenis kan zijn die “in de wacht” staat. Als de Event Timer van een Pin actief is en er wordt in de tussentijd een (Sequence of Timed) Instructie uitgevoerd die de Event Timer van diezelfde Pin gebruikt, dan wordt de timer overschreven en wordt de vorige timer-actie niet meer uitgevoerd.

Naast de Event Timer is er ook één (1) Periode Timer per pin. De Periode Timer wordt gebruikt voor cyclische acties, zoals het laten knipperen van een Pin.

De “Event Timer” en de “Periode Timer” zijn verschillende timers. Dus je kunt bv gerust een Pin op knipperen zetten en daarachter een Wait. Het knipperen gaat dan gewoon door tijdens de wachttijd.

Het verschil is gemakkelijk te zien aan de plaats waar je de tijd invult. Daar waar je iets invult in de velden Time/Base betreft het de Event Timer. Daar waar je iets invult in het veld T/60 betreft het een Periode Timer.

4 Apparaatdefinities maken

Vóór OC32 release 3.0 was het gebruik van apparaatdefinities beperkt tot voorgedefinieerde apparaten. Wat je zelf configureerde was gewoon een verzameling losse Pinnen en Aspect configuraties.
Vanaf 3.0 is in principe elk onderdeel dat je aanstuurt een apparaat. Het configureren wordt hiermee eenduidiger en uiteindelijk ook gemakkelijker.

Een apparaat kan bestaan uit één of meerdere Pinnen. Eén van die Pinnen, in het algemeen de eerste Pin die toegekend is aan het apparaat, is de "Device Control Pin". Onder elke Pin wordt aangegeven bij welke Device Control Pin die Pin hoort.

In het tabblad OC32 Device Configuration heet de Device Control Pin eenvoudigweg "Device Pin". Een Device heeft ook een naam. Die naam is altijd opgeslagen onder de Device Control Pin. Andere Pinnen die horen bij hetzelfde Device wijzen naar dezelfde Device Control Pin en daarmee ook naar de naam van dat Device.
Dus als je zelf een apparaat definieert bestaande uit meerdere Pinnen moet je er zelf ook voor zorgen dat al die Pinnen wijzen naar dezelfde Device Pin. Dat doe je door bij al die Pinnen het betreffende "Device Pin" veld correct in te vullen. De naam van het Device kun je ook aanpassen, maar alleen als de Pin geselecteerd is die ook Device Control Pin is. Bij de andere Pinnen wordt de naam wel getoond, maar is deze grijs en geblokkeerd voor wijziging.

Je kunt de configuratie van alle Pinnen, die horen bij dezelfde Device Control Pin, in één actie wegschrijven naar de OC32 door te klikken op "Write Device". Je kunt ook een Device laden uit de OC32 door te klikken op "Read Device". Bij "Read Device" zal OC32Config eerst kijken wat de in de OC32 opgeslagen Device Control Pin is waar de op dat moment geselecteerde Pin naar verwijst. Vervolgens haalt OC32Config alle instellingen op van alle Pinnen die verwijzen naar diezelfde Device Control Pin. Merk op dat in de OC32 geen namen zijn opgeslagen, dus die blijven bij lezen leeg of ongewijzigd.

Als je zelf een apparaat hebt gemaakt of aangepast kun je dat apparaat ook opslaan in je computer om het later weer te kunnen gebruiken als apparaatdefinitie. Dat opslaan kun je doen door te klikken op de knop "Save Device". Er wordt dan een apparaatdefinitiebestand aangemaakt met één apparaat, dat je in OC32 vervolgens weer kunt laden als definitiebestand.
Als je wilt kun je meerdere apparaatdefinitiebestanden samenvoegen. Dat is op zich een vrij eenvoudig, maar handmatig proces dat je kunt uitvoeren met een eenvoudige tekst-editor zoals Windows kladblok (notepad). Een definitiebestand is namelijk gewoon leesbare tekst.

Open het definitiebestand van Duitsland eens met Notepad (klik met de rechtermuisknop op het bestand en kies voor openen met ….etc)

Je ziet dan iets als:

"OC32DeviceDefinitions DE 2013/03/05"

"Dev","(2)DE: Blocksignal"
….
"Dev","(6)DE: Blocksignal + Vorsignal"
….

…..
"End"

De eerste regel dient altijd te beginnen met "OC32DeviceDefinitions <iets>"
Het <iets> is de naam die in OC32Config getoond wordt als je het definitiebestand selecteert.

De regel die start met “Dev” kondigt een apparaatdefinitie aan. Alle regels tot aan de volgende “Dev” (of “End”) maken onderdeel uit van die definitie.

Je kunt een willekeurige definitie uit elk definitiebestand (het hele blok van “Dev” tot aan de volgende “Dev”) kopiëren en plakken tussen de definities in een willekeurig ander definitiebestand. Zorg er alleen voor dat de structuur van het bestand in tact blijft en zorg dat er maar één eerste regel is die start met “OC32DeviceDefinitions “ (inclusief een spatie) en één laatste regel die “End” bevat. De tekst achter “OC32DeviceDefinitions “ mag je naar believen aanpassen, bijvoorbeeld in “OC32DeviceDefinitions MijnEigenDefinitieBestand”

5 Toekennen adressen aan Pinnen

Vóór release 3.0 had elke Pin één serieel adres en één DCC adres. Met ingang van release 3.0.0.0 is dat aangepast. Aan elke Pin kun je nu toekennen:

- 0 t/m 6 seriële adressen
- 0 t/m 6 Basic DCC Accessory adressen (zuiver gezien zijn dit DCC poortnummers)
- 0 t/m 1 Extended DCC Accessory adressen

Het instellen van het aantal adressen doe je op het tabblad OC32 Device Configuration onder de beschrijving van de Pin terwijl "Show Details" is aangevinkt.
In heel veel gevallen kun je gewoon 1 adres per Pin laten staan. Als er geen duidelijke reden voor is adviseren we ook dat zo te laten. Alleen in bepaalde gevallen kan het aanpassen van de adressen handig en soms zelfs noodzakelijk zijn.

LET OP: Het aanpassen van het aantal adressen per Pin wijzigt de adressering van de apparaten die op de daaropvolgende Pinnen zitten! Wijzig dus nooit zo maar, maar doe dat alleen met beleid en nadat je een goed plan hebt gemaakt daarvoor.
De OC32 nummert de adressen in elke reeks doorlopend en aansluitend. Als de eerste Pin serieel adres 33 heeft en je kent aan de eerste Pin 2 seriële adressen toe, dan is het seriele adres van de tweede Pin dus 35. Als je aan die tweede Pin één adres toekent en aan de derde Pin nul adressen, dan heeft de derde Pin geen serieel adres en is het seriële adres van de vierde Pin 36.
Als je in dit voorbeeld het aantal adressen toegekend aan de eerste Pin wijzigt van 2 naar 3, dan worden de adressen van Pin 2 en 4 ook één hoger.

Aangezien je aan elke Pin meerdere adressen kunt toekennen kan een OC32 met ingang van release 3.0.0.0. dus ook meer dan 32 seriële adressen bevatten. De maximale adrescapaciteit per OC32 bedraagt:

- 128 seriële adressen
- 128 Basic DCC Accessory poorten
- 32 Extended DCC Accessory adressen

Adrestoewijzing op seriële adressen is uitsluitend van toepassing en ook alleen zinvol als je SetAspect commando's gebruikt en dus de module echt gebruikt als OC32. Bij gebruik van de "oude" OM32 commando's wordt altijd rechtstreeks de Pin aangestuurd.

Meer dan 32 seriële adressen per module kunnen alleen worden aangestuurd als de module wordt geadresseerd door middel van modulenummer en uitgang. Bij de zogenaamde lineaire adressering zijn er 512 seriële adressen die normaliter gelijk verdeeld worden over 16 modules. Welke methode er gebruikt wordt hangt af van je besturingssysteem en-software.

Meer dan 32 adressen kan noodzakelijk zijn, bijvoorbeeld als je een wisselmultiplexer gebruikt in een 8 x 8 matrix. In dat geval sturen 16 Pinnen 32 wissels aan en zou je anders geen adressen meer over hebben voor de resterende 16 Pinnen.

Het toekennen van meer dan 1 adres per Pin is in sommige gevallen erg nuttig. Stel dat je een vormsein hebt dat je aanstuurt met één servo, maar dat dat sein in 3 standen moet kunnen staan. Als je dat wilt aansturen met Basic DCC pakketten heb je maar 2 standen en moet je dus een Pin opofferen om de derde stand te kunnen adresseren. Door gewoon 2 DCC adressen toe te kennen aan de Pin los je dit efficiënter op.

Het toekennen van minder dan één adres per Pin kan nuttig zijn als je een tekort aan adressen hebt. Stel dat je een DCC systeem hebt met een toetsenbord dat slechts 32 aansluitende adressen kan besturen. Je kunt die 32 adressen dan verdelen over meerdere OC32's door per OC32 alleen die adressen te gebruiken die ook nuttig zijn.

Als je de OC32's gebruikt voor zowel treinen (DCC) als auto's (MCC/RS485) zou je zelfs nog verder kunnen gaan: ken alleen DCC adressen toe aan OC32 apparaten die door DCC bestuurd moeten worden en ken alleen seriële adressen toe aan apparaten die door MCC bestuurd moeten worden. Dan weet je zeker dat beide systemen elkaar nooit in het vaarwater kunnen zitten.

(Deze pagina is opzettelijk leeg)

(Deze pagina is opzettelijk leeg)