author: Gerrit van Dalfsen
EKV Genneper Molen
tel.: +31 10 458 4816
email: gdalfsen@wxs.nl
web: Ariadne

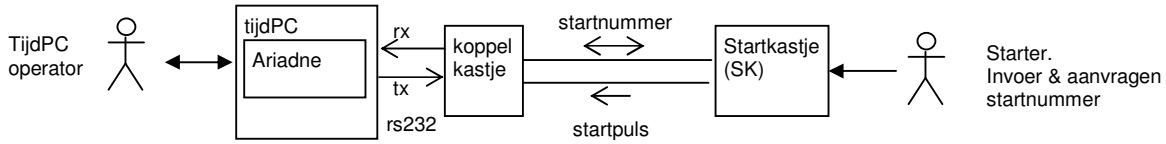
Date	Version	Change
08Nov2006	0.1	First draft
06Jan2007	0.2	Aanscherping specificaties
15Jan2007	0.3	remote alive check verwijderd
22Feb2007	0.4	Review Anton verwerkt
29Mei2007	0.5	Aanvullingen Iede verwerkt
22Nov2007	0.6	Nieuw koppelkastje verwerkt

Ariadne - Communicatie met Startkastje

INHOUDSOPGAVE

1. Inleiding

Dit document beschrijft de specificatie en het ontwerp van de startnummercommunicatie tussen de Ariadne tijdTPC en het GenneperMolen (GM) startkastje.



2. Beschrijving startnummercommunicatie

Met het startkastje vraagt de starter een te starten startnummer aan bij Ariadne op de tijdPC. Ariadne controleert of het startnummer in de betreffende loop al eerder gestart is. Als het startnummer niet eerder gestart is, wacht Ariadne tot het startinterval tot de vorige start verstreken is en geeft daarna een OK-signaal naar het startkastje. Het startkastje activeert daarop de fotocel startlijn.

Als het startnummer al reeds eerder gestart is, dan signaleert Ariadne dat naar de tijdPC-operator. Deze heeft daarop telefonisch contact met de starter. Om verder te kunnen moet het startkastje gereset worden. De tijdPC operator stuurt daarvoor via Ariadne een resetpuls naar het startkastje. Dat brengt het startkastje in de standby toestand. De starter kan nu opnieuw een startnummer aanvragen.

2.1. Startnummer communicatie protocol

Bij een startnummeraanvraag stuurt het startkastje ten behoeve van foutdetectie het startnummer twee keer achter elkaar naar de tijdPC. Het tweede gestuurde startnummer moet gelijk zijn aan het eerst gestuurd startnummer. Wanneer dat niet het geval is, maakt Ariadne melding van een 'Transmissiefout'.

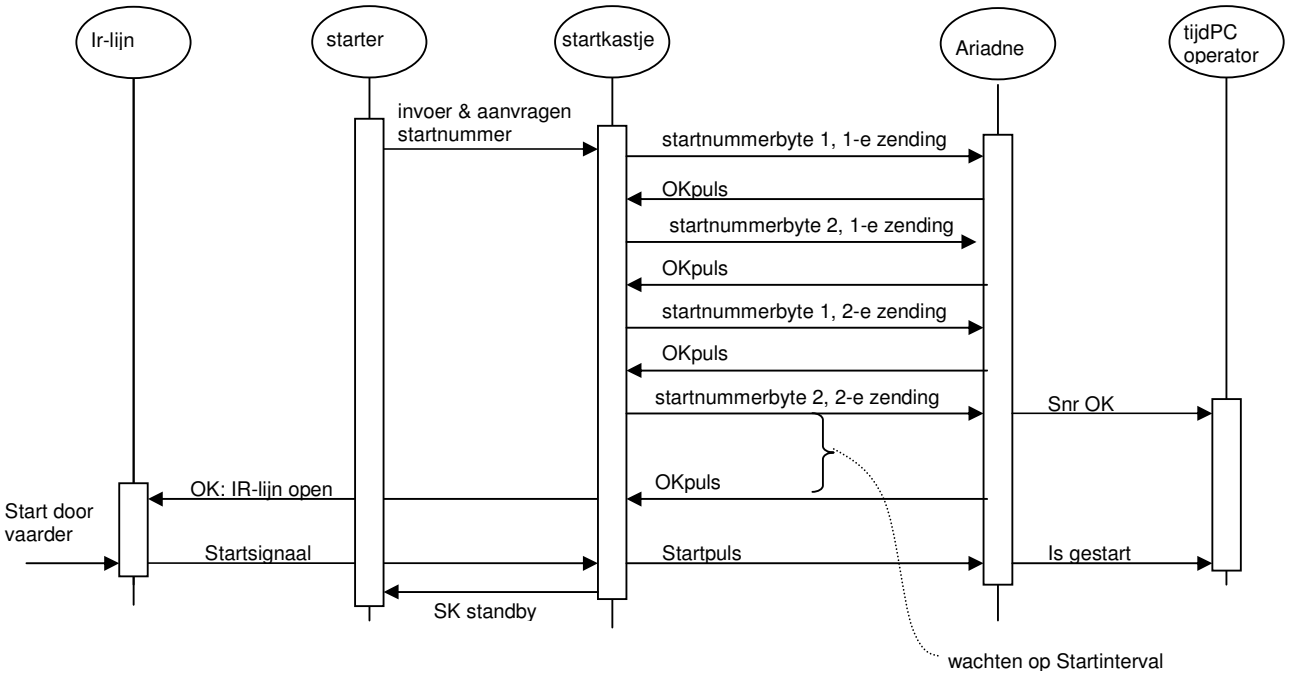
Startnummer OK

Als Ariadne het startnummer goed ontvangen heeft, en het startnummer is OK, dan wacht Ariadne nog tot de startintervaltijd[1] tot de vorige start is verstreken en stuurt dan een OK-signaal naar het startkastje. Het startkastje activeert daarop de fotocel startlijn. Na de onderbreking van de fotocel startlijn door de startende vaarder komt het startkastje weer in de standby toestand. De starter kan dan het volgende startnummer aanvragen.

Het volgende UML sequentiediagrammen toont de startnummercommunicatie schematisch:

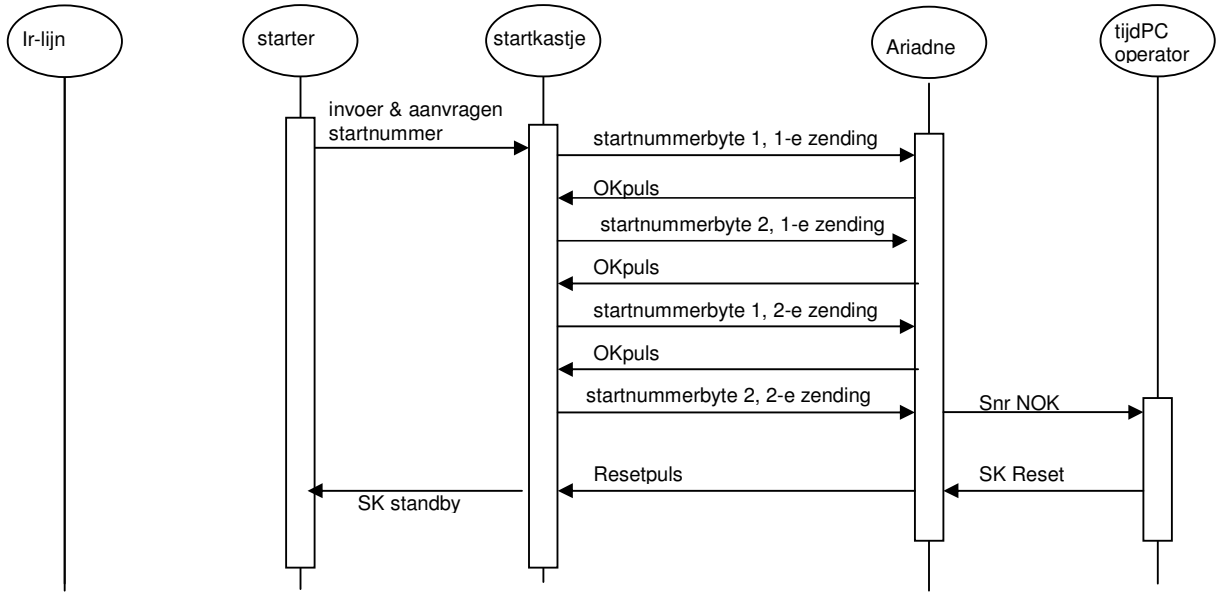
[1] De lengte van het startinterval is instelbaar op Ariadne.

Normale Startnummercommunicatie
Communicatiesequentie van het normaal aanvragen van een startnummer en het afronden van een start.



Startnummer transmissiefout
Wanneer bij een startnummeraanvraag de twee gestuurde startnummers niet aan elkaar gelijk zijn, meldt Ariadne een 'Transmissiefout'.

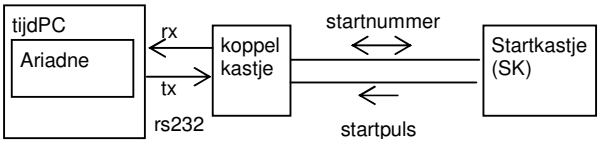
Communicatiesequentie bij het optreden van een startnummer transmissiefout:



De tijdPC-operator probeert uit te zoek wat er aan de hand is. Hij zal in ieder geval met de starter contact opnemen over welk startnummer zojuist is aangevraagd. De tijdPC-operator kan via Ariadne een resetpuls naar het startkastje sturen. Daardoor komt het startkastje in de standby toestand, waarna de starter opnieuw het startnummer kan aanvragen. Als de transmissiefout opnieuw optreedt, moet de hardware gecontroleerd worden (batterijen startkastje, connectors, bekabeling e.d.).

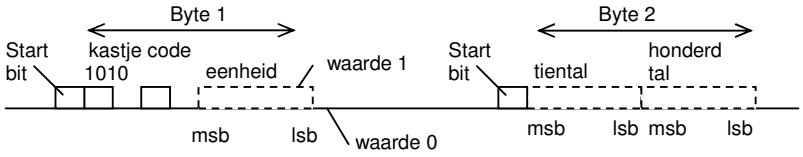
2.2. Startnummer transmissietechniek

In de afstandsoverbrugging (tot 200 mtr) tussen startkastje en tijdPC worden voor de startnummer communicatie en voor de startpulsdoorgifte een afzonderlijk aderpaar gebruikt, elk met een open collector verbinding met pull-up weerstand.



Over het aderpaar voor de startnummercommunicatie worden zowel de startnummer bytes als de OK-en resetpuls verstuurd. In normale werking zijn deze signalen niet conflicterend. Een OK-puls of een resetpuls wordt altijd verstuurd als er geen startnummerbyte verstuurd wordt. Deze situatie is niet beveiligd. Vanuit de tijdPC kan elk moment een resetpuls gestuurd worden, die kan conflicteren met een op dat moment gestuurd startnummerbyte.

De startnummer cijfers die het startkastje verstuurt, zijn gecodeerd in bytes, met een start- en stopbit overeenkomstig het rs232 protocol. Byte 1 bevat de eenheid van het startnummer plus een startkastje-identificatiecode, byte 2 het tiental en het honderdtal van het startnummer:



De startnummercommunicatie is te beschrijven in de volgende rs232 parameters:

Baudrate 67,65 bytes/sec
Startbits = 1
Stopbits = 1
Databits = 8
Parity = None

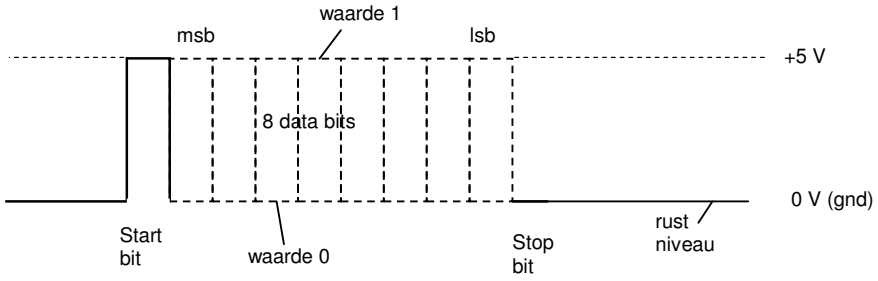
Het Startkastje verstuurt niet expliciet een stopbit. De stopbit van rs232 is een nul aan het einde van het byte. Deze nul is de facto aanwezig omdat na het versturen van het byte het startkastje wacht op een OKpuls van de tijdPC.

Het koppelkastje splitst de signalen op de startnummer-communicatielijn in een Rx (Read) en Tx (Transmit) lijn, die met een rs232-omzetter omgezet worden naar de juiste rs232 spanningsniveaus (+ en - 12 V).

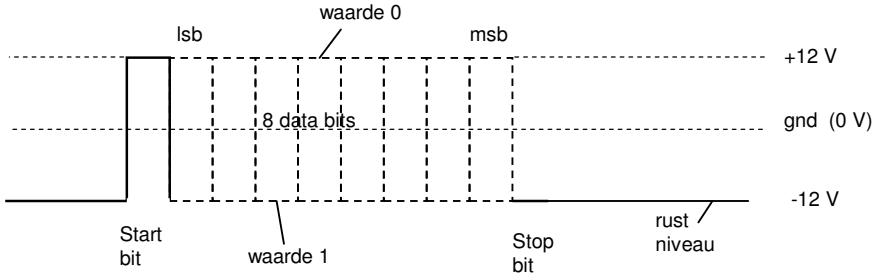
Het startnummerbyte wordt daardoor leesbaar op de serial port van de tijdPC.

Het startnummergetal dat de tijdPC op zijn serial port leest, is niet hetzelfde als het startnummergetal dat het startkastje wegstuurt. Dat komt doordat zowel de lsb-msb volgorde van de bits in de byte, als de signaalniveaus waaraan de 0 en 1 waarden worden toegekend, van het startkastje ten opzichte van het rs232 protocol omgekeerd zijn. (Het startnummercommunicatie protocol dateert van voor de tijd dat er PC's met rs232 poorten voor de tijdwaarneming gebruikt gingen worden).

De opbouw van het startnummerbyte dat het startkastje wegstuurt is als volgt:



Na de niveauaanpassing naar rs232 leest de tijdPC leest dit byte als volgt:



Een bit dat door het startkastje als een 1 wordt aangeboden, wordt door de tijdPC (conform rs232 specificatie) gelezen als een 0. En een bit dat in het startkastje-startnummer een msb is, is in rs232 specificatie een lsb. Daardoor is het nummer dat de tijdPC op de serial port leest, een ander nummer dan dat het startkastje heeft verzonden. Er zijn een paar getallen transformaties nodig om de gelezen bytes weer om te rekenen naar het verzonden startnummer.

Het terugrekenen van het startnummer gaat als volgt:

Op het verzenden van een startnummer door het startkastje leest de tijdPC twee bytes in:

Byte	**Startkastje stuurt**	**TijdPC leest (msb-lsb omkering)**
byte 1	kastje-code (C) + startnummer-eenheid (E)	EC
byte 2	startnummer-tiental (T) + startnummer-honderdtal (H)	HT

Vanwege de omkering van msb-lsb is de nibble-volgorde van de startnummer cijfers in de gelezen bytes omgekeerd ten opzichte van de door het startkastje verzonden volgorde. (Een nibble is een groep van 4 bits, bijv. 1011).

Het berekenen van het startnummercijfer uit het byte-nibble gaat als volgt:
- de tijdPC leest een nibble met bijv. hex waarde B
- schrijf het nibble binair uit. nibble hex B is binair 1011
- draai de volgorde van de bits om: 1011 wordt 1101
- inverteer het binaire getal: 1101 wordt 0010
- de (hexa)decimale waarde van dit laatste nibble is het gevraagde cijfer: 2

De omrekening kan met met een mooie rekenkundige operatie. De volgende vertaaltabel geeft de vertaalcombinaties:

Tijd PC nibble hex	0	**1**	2	**3**	4	**5**	**6**	**7**	8	**9**	A	**B**	C	**D**	**E**	**F**
startnummer cijfer	F	7	B	3	D	5	9	1	E	6	A	2	C	4	8	0

(Opmerking: Alleen de vette tijdPC hex nibbles zijn relevant omdat alleen deze naar decimale cijfers mappen.)

Voorbeeld:
Het startkastje stuurt nummer startnummer 033 weg. De TijdPC ontvangt de integer-bytes 58 en 243, omgerekend in hex: 3A en F3. De volgende tabel geeft aan hoe uit deze getallen het startnummer is af te leiden:

byte	nibble	Waarde	inhoud	vertaling	resultaat
1	1	3	startnummer eenheid	3 => 3	startnummereenheid = 3
1	2	A	startkastje code		
2	1	F	startnummer hondertal	F => 0	startnummer hondertal = 0
2	2	3	startnummer tiental	3 => 3	startnummer tiental = 3

Samengevoegd geeft dit startnummer 033.

tijdPC signalen: OK-puls en resetpuls
De signalen die de tijdPC geeft naar het startkastje moeten een omgekeerd transformatie ondergaan.

De **OK-puls** bestaat uit een puls van een half byte lang. In Startkastje-taal is dit hex F0. De tijdPC moet hiervoor hex F0 sturen. (inverteren en roteren).

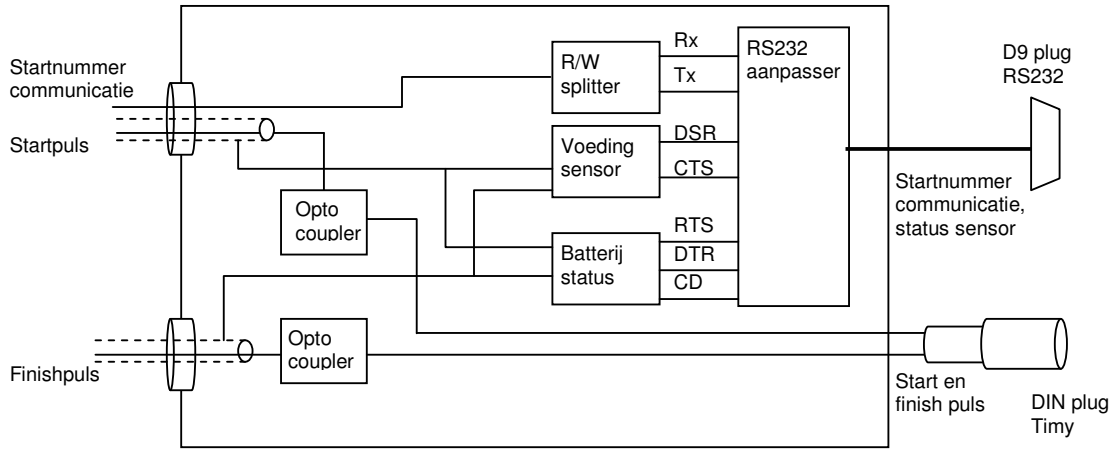
De **Resetpuls** bestaat uit een puls van een heel byte lang. In Startkastje-taal is dit hex FF. De TijdPC moet hiervoor hex 00 sturen.

De tijdPC signalen echoën terug naar het tijdPC leeskanaal. De bytes hex F0 en hex 00 kunnen niet door het Startkastje verstuurd worden. Deze bytes kunnen daarom direct weggegooid worden wanneer de tijdPC ze leest.

2.3. Het koppelkastje

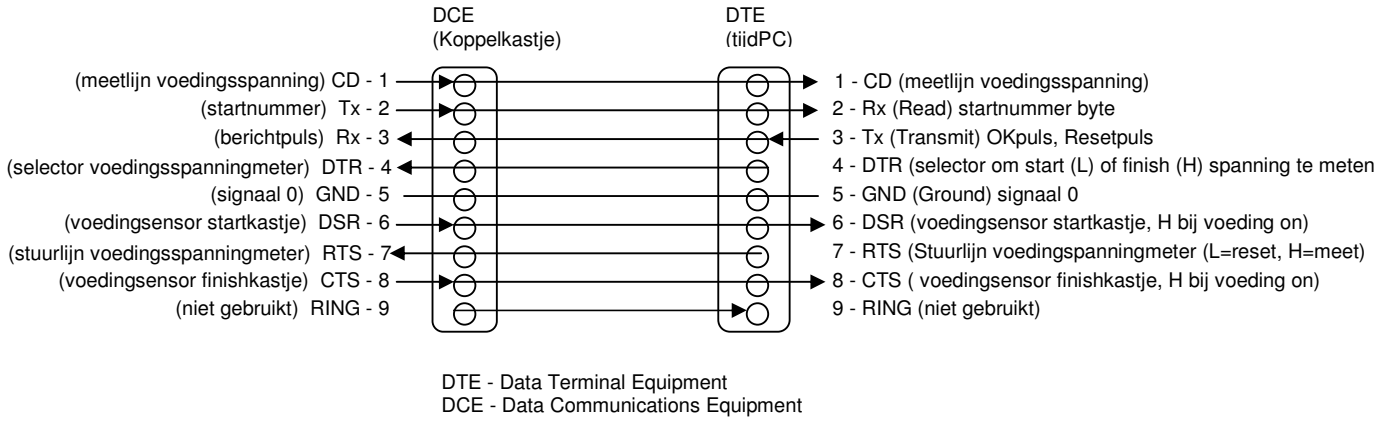
Het koppelkastje heeft de volgende functies:

1) Het bundelen van de tijdpulsleiding naar de start en naar de finish in één plug naar de Timy
2) Het splitsen/samenvoegen van de startnummer communicatiesignalen van en naar de startnummer communicatielijn en de Rx en Tx lijnen van de rs232 plug
3) Het aanpassen van het signaalnivo tussen startnummer communicatielijn (0..5 V) en de rs232 spanningsniveau's (-12,0,+12 V).
4) Het doorgeven een signalering naar de rs232 plug of het startkastje en of finishkastje is aangesloten. (De batterijspanning van het start- en finishkastje zijn via de Start/finish kabels waar te nemen in het koppelkastje)
5) Een functie om de hoogte van de spanning van het start- of finishkastje te meten en naar de tijdPC door te geven..



Voor het electronische schema van het koppelkastje zie: koppelkastjev2.gif

Het volgend schema toont welke signalen op de rs232 connector beschikbaar zijn:



DTE - Data Terminal Equipment
DCE - Data Communications Equipment

NB. Het koppelkastje krijgt zijn voeding uit de Timy. De Timy moet dus aangesloten zijn, wil de communicatie met het startkastje werken.

De DSR-lijn wordt gebruikt als voedingsensor voor het startkastje.
DSR = Hoog: startkastje is aangesloten
DSR = Laag: startkastje is niet aangesloten

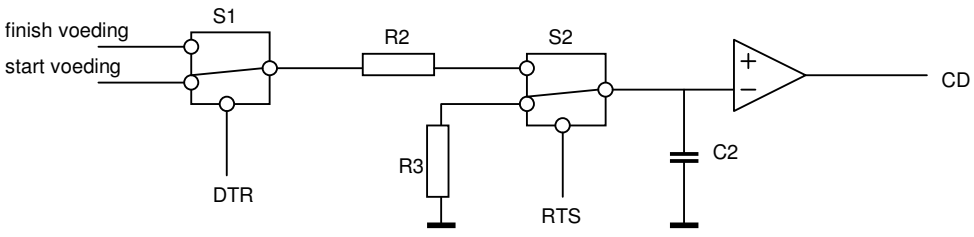
De CTS-lijn wordt gebruikt als voedingsensor voor het finishkastje.
CTS = Hoog: finishkastje is aangesloten
CTS = Laag: finishkastje is niet aangesloten

In de driver voor het koppelkastje kan de waarde van de DSR en CTS lijn opgevraagd worden. Daarmee is op de tijdPC waar te nemen of het startkastje resp. finishkastje wel of niet is aangesloten. Een wijziging van de DSR of CTS lijn is ook als een event beschikbaar. Dat houdt in dat de tijdPC een alarm kan geven als de verbinding met ht start- of finishkastje wegvalt.

In het koppelkastje is de voedingssensor zo ingesteld dat de DSR en CTS lijn hoog is bij een spanning boven 4,6 V, en laag bij een spanning beneden de 4,6 V. Deze spanning is zo ongeveer het minimum waarop het start- en finishkastje nog werken.

2.4. Het meten van de voedingsspanning van start- en finishkastje

In het koppelkastje is een voorziening (zie schema onder) opgenomen waarmee je de batterijspanning van het startkastje, resp. het finishkastje kunt meten.



Het meten van de voedingsspanning gaat als volgt:

1. Zet S1 met DTR op de voeding van het startkastje (DTR = L) of op de voeding van het finishkastje (DTR = H).
2. Ontlaadt C2 via S2 en R3 door RTS op L te zetten. CD wordt dan L.
3. Schakel S2 om door RTS op H te zetten. C2 gaat nu opladen uit de voedingsspanning van het start- of finishkastje.
4. Als C2 voldoende is opgeladen, dan zal CD omklappen naar H.
5. De tijd tussen het H maken van RTS en het H worden van CD moet op de tijdPC gemeten worden. De gemeten tijd is een maat voor de batterijspanning van start- of finishkastje.

De volgende formule wordt gebruikt om de tijd naar voltage om te rekenen:

Ui = Vc/(1-exp(-t/RC))

waarin:
Ui - de te meten spanning
Vc = 2,5 V (Dit is de helft van de voedingspanning van het koppelkastje (is 5V afkomstig uit de Timy)
t = gemeten tijd in seconden
R = 1 E06 Ohm (de weerstand van het RC-netwerk)
C = 1 E-06 F (de condensator van het RC netwerk)

Een tabel met berekende waarden binnen de praktische batterijspannings range is:

t (sec)	Ui (Volt)
0,45	6,90
0,5	6,35
0,55	5,91
0,6	5,54
0,65	5,23
0,7	4,97
0,75	4,74
0,8	4,54

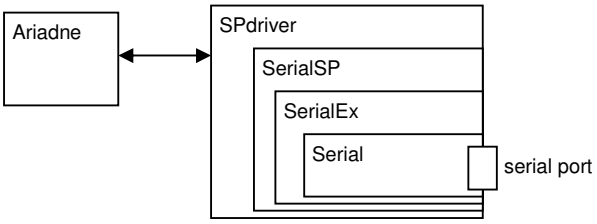
Het idee is dat de tijdPC op gezette tijden de batterijspanning meer van start- en finishkastje, en een alarm geeft wanneer de batterijspanning beneden een norm komt.

3. De StartpostDriver

De communicatie met de startpost (via het koppelkastje) is ondergebracht in een software service op basis van Microsoft COM, geschreven in C++ en gebruikmakend van ATL. De COM-component kan door Ariadne (MS-Access) aangeroepen worden. Zo'n COM-component kan gezien worden als een driver voor het startkastje. Vandaar de naam "StartpostDriver".

De StartpostDriver heeft richting Ariadne een interface met daarin eigenschappen (properties), functies (methods) en gebeurtenissen (events). Ariadne communiceert met de StartpostDriver via deze interface. De StartpostDriver communiceert via de serial port met het koppelkastje en het startkastje daarachter.

De SPdriver bestaat uit een aantal classes:

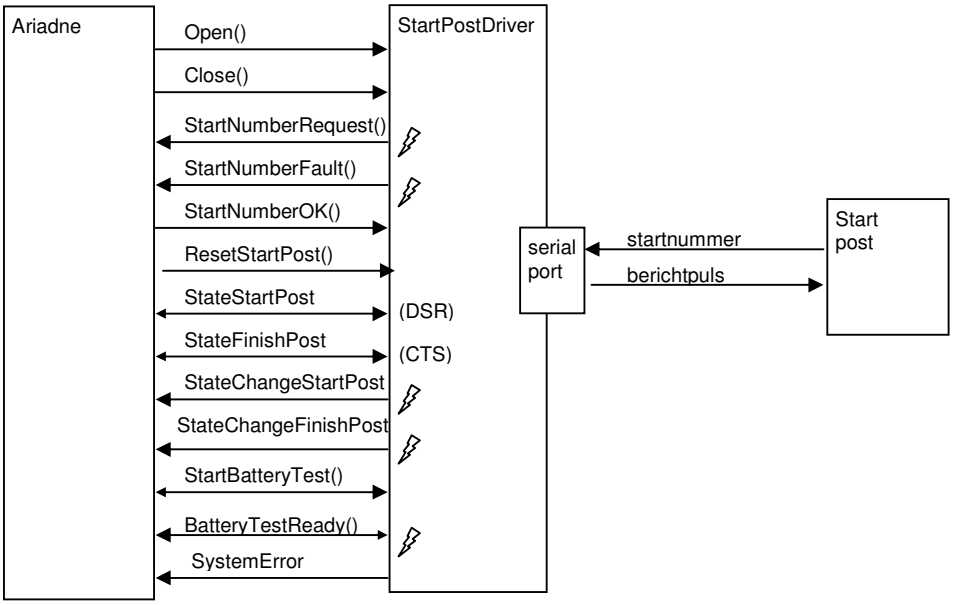


- SPdriver verzorgt de interface met Ariadne, doet de startnummer transformatie, controleert of startnummer bij eerste en tweede keer versturen hetzelfde is.
- SPserial is een uitbreiding op SerialEx waarin de communicatie met SPdriver wordt geregeld
- SerialEx zorgt ervoor dat Serial in een aparte thread draait.
- Serial handelt de interactie met de serial port af.

Voor het aansturen van de seriele poort wordt gebruik gemaakt van een open source bibliotheek: Serial Library for C++ Copyright (C) 1999-2003 Ramon de Klein (Ramon.de.Klein@ict.nl)

De StartPostDriver handelt een deel van de interactie met het startkastje zelf af, onder andere de check dat de tweemaal gestuurde startnummers gelijk moeten zijn. De startPostDriver geeft ofwel een goed ontvangen startnummeraanvraag door, ofwel een transmissiefout als de tweemaal gestuurde startnummers niet gelijk zijn. Verder zijn er functies om de seriële poort te openen en te sluiten, en functies voor het uitvragen van de voedingslijnen toestand, en voor het meten van de batterijspanning.

De interface van de startpostdriver met Ariadne ziet er als volgt uit:



Overzicht van de interface items:

Interface item	type	Omschrijving
Open()	method	opent serial port en stelt de poortparameters in
Close()	method	sluit serial port
StartNumberRequest()	event	startnummer aanvraag
StartNumberFault()	event	er is een fout opgetreden in de startnummer communicatie
StartNumberOK()	method	goedkeuring op startnummer aanvraag
ResetStartPost()	method	reset van de startpost
StateStartPost	property	informatie over of de startpost wel/niet is aangesloten
StateFinishPost	property	informatie over of de finishpost wel/niet is aangesloten
StartPostStateChange	event	startpost toestand is gewijzigd. (verbinding met startpost is opgekomen of weggevallen)
FinishPostStateChange	event	finishpost toestand is gewijzigd. (verbinding met finishpost is opgekomen of weggevallen)
StartBatteryTest()	method	Geeft opdracht de voedingsspanning van start- of finishkastje te meten
BatteryTestReady()	event	Geeft de gemeten de voedingsspanning van start- of finishkastje terug
SystemError()	event	Komt op wanneer een andere dan een gedefinieerde startpostdriver fout optreedt.

In de volgende paragrafen worden de interface items meer in detail bescheven.

3.1. Open (PortNumber, Status)

Opent een serial poort naar de Startpost met opgegeven PortNumber (1,2,3,4,5). De method geeft een Status terug.

Status: 0: COM[PortNumber] geopend
-1: COM[PortNumber] kan niet geopend worden

Als de poort niet geopend kan worden, wordt ook de event **SystemError** gezet. Deze bevat een aanwijzing over de aard van de fout. Zie daar.

Bij het openen van de poort worden de poortparameters als volgt ingesteld:

Baudrate	= 68
Startbits	= 1
Stopbits	= 1
Databits	= 8
Parity	= None
RTS	= Disable (zet batterijmeter in nulstand)

3.2. Close()

Sluit de communicatie naar de startpost en geeft de poort vrij.

3.3. StartNumberRequest (StartNumber)

Bij een startnummeraanvraag leest en controleert de StartPostDriver de vier (4) binnenkomende startnummerbytes. Wanneer de startnummeraanvraag goed is ontvangen (de twee gestuurde startnummers zijn gelijk) dan wordt het StartNumberRequest event gegenereerd, waarbij het startnummer als argument wordt meegegeven.

Als zich tijdens het verwerken van een startnummer aanvraag fouten voordoen (de twee gestuurde startnummers zijn niet gelijk, of er is een timeout) dan wordt het event **StartNumberFault** event gegenereerd (zie daar).

3.4. StartNumberFault(ErrorCode, ErrorMessage)

Wanneer het startkastje een startnummeraanvraag doet, en het aangevraagde startnummer komt niet goed over (het eerste keer en tweede keer verstuurde startnummer is niet hetzelfde, of er is een timeout, dan stuurt de Startpostdriver een Fault event met ErrorCode en ErrorMessage

ErrorCode	ErrorMessage
1	Illegal startpost id
2	1st byte <> 2nd byte
3	Inter character timeout

3.5. StartNumberOK(StartNumber)

Als reactie op een StartNumberRequest van de startpostdriver geeft Ariadne een StartNumberOK indien het startnummer in de betreffende serie niet al geweest is, en nadat de startintervaltijd is verstreken. Wanneer het aangevraagde startnummer in de betreffende serie al geweest is, geeft Ariadne geen StartNumberOK.

De startpostdriver controleert of het OK-gegeven startnummer hetzelfde is, als dat was aangevraagd. Zoja, dan stuurt de startpostdriver een OK-puls (hex F0) naar de startpost en geeft returnwaarde 0. Zonee, dan stuurt de startpostdriver geen OK-puls, en geeft een returnwaarde 1 op de StartNumberOK aanvraag.

3.6. ResetStartPost()

Stuurt een resetpuls (hex 00) naar de startpost. De startpost komt daarmee in de standby toestand. De tijdPC operator geeft via Ariadne een startpostreset wanneer de starter een onjuist startnummer heeft aangevraagd, of wanneer er een transmissiefout is opgetreden.

3.7. StateStartPost

Dit is een eigenschap van de startpostdriver die aangeeft of het startkastje is aangesloten. Als het startkastje via de kabel op het koppelkastje is aangesloten, dan is de voedingsspanning van het startkastje beschikbaar in het koppelkastje. Het koppelkastje vertaalt het al dan niet aanwezig zijn van de voedingsspanning van het startkastje naar

een hoog of laag op de pen DSR van de rs232 connector. Bij het uitlezen van de de StateStartPost eigenschap geeft de startpostdriver de toestand van de DSR-pin door:

StateStartPost	DSR	Omschrijving
on	enable (Hoog)	Startkastje aangesloten
off	disable (Laag)	Startkastje niet aangesloten

3.8. StateFinishPost

Dit is een eigenschap van de startpostdriver die aangeeft of het finishkastje is aangesloten. Als het finishkastje via de kabel op het koppelkastje is aangesloten, dan is de voedingsspanning van het finishkastje beschikbaar in het koppelkastje. Het koppelkastje vertaalt het al dan niet aanwezig zijn van de voedingsspanning van het finishkastje naar een hoog of laag op de pen CTS van de rs232 connector. Bij het uitlezen van de de StateFinishPost eigenschap geeft de startpostdriver de toestand van de CTS-pin door:

StateFinishPost	CTS	Omschrijving
on	enable (Hoog)	Finishkastje aangesloten
off	disable (Laag)	Finishkastje niet aangesloten

3.9. StateChangeStartPost()

De startpostdriver kan waarnemen of er een verandering op de DSR-pin van de rs232-connecter heeft plaatsgevonden. In de context van de startpostdriver houdt dit in dat het startkastje is aangesloten geworden, of dat de verbinding met met startkastje is verbroken. Tijdens een wedstrijd is het belangrijk om te weten dat de verbinding met het startkastje is weggevallen. Met de StateChangeStartPost event kan het wegvallen van de verbinding met het startkastje naar Ariadne gesignaleerd worden.

3.10. StateChangeFinishPost()

De startpostdriver kan waarnemen of er een verandering op de CTS-pin van de rs232-connecter heeft plaatsgevonden. In de context van de startpostdriver houdt dit in dat het finishkastje is aangesloten geworden, of dat de verbinding met met finishkastje is verbroken. Tijdens een wedstrijd is het belangrijk om te weten dat de verbinding met het finishkastje is weggevallen. Met de StateChangeFinishPost event kan het wegvallen van de verbinding met het finishkastje naar Ariadne gesignaleerd worden.

3.11. StartBatteryTest(DeviceSelector)

Het koppelkastje bevat een faciliteit om de batterijspanning van het startkastje of het finishkastje te meten. Zie par.2.4. Vanuit de startpostdriver wordt dit als volgt aangestuurd:

- De RTS pin staat normaliter op laag (disable) (Is zo ingesteld bij het openen van de serial poort).
- De waarde uit DeviceSelector wordt overgenomen naar de DTR-pin:

DeviceSelector	DTR	Omschrijving
StartPost	disable (Laag)	Startkastje voeding geselecteerd
FinishPost	enable (Hoog)	Finishkastje voeding geselecteerd

- RTS wordt vervolgens op hoog (enable) gezet, en tegelijk wordt de systeemtijd in milliseconden uitgelezen.
- Op het moment dat de CD-pin van waarde verandert, wordt de tijd opnieuw uitgelezen. De het tijdverschil (in milliseconden) wordt omgerekend naar een spanningswaarde. Zie verder BatteryTestReady.

3.12. BatteryTestReady(DeviceSelector, Voltage)

Met de method StartBatteryTest wordt de batterijmeter ingesteld en gestart. De batterijspanning wordt berekend op grond van de tijd die nodig is een condensator op te laden. Met StartBatteryTest wordt de begintijd vastgelegd. Op

het moment dat de condensator een bepaalde spanning heeft aangenomen, klapt de CD-lijn om. Op dat event wordt de eindtijd vastgelegd en de meettijd (in milliseconden) uitgerekend. Deze waarde wordt vervolgens omgerekend naar een spanning, (zie par. 2.4) die via het event wordt doorgegeven.

Na afloop van de meting wordt DTR weer laag gezet (dit zet de batterijmeter weer in de nulstand, de condensator wordt dan weer ontladen).

3.13. SystemError(ErrorMessage)

Indien in de startpostdriver een andere fout optreedt dan een expliciet afgevangen fout,dan wordt die met de system error event doorgegeven. De ErrorMessage bevat een omschrijving van de fout.

4. Startpostdriver testinterface

Ten behoeve van het testen van de startpostdriver zijn op een testinterface een aantal interne gegevens beschikbaar gemaakt.

4.1. StartNumberTest(StartNumber)

Het simuleren van een startnummeraanvraag door het startkastje vanuit de StartpostSimulator.

4.2. ReadyToSend(RTS)

Opvragen en instellen RTS status.

4.3. Parity(Parity)

Opvragen en instellen Parity parameter.

4.4. BaudRate(Baudrate)

Opvragen en instellen Baudrate parameter.

4.5. DataBits(Databits)

Opvragen en instellen Databits parameter.

4.6. StopBits(Stopbits)

Opvragen en instellen Stopbits parameter.

4.7. ParityCheck(Paritycheck)

Opvragen en instellen Paritycheck parameter.

4.8. WriteToPort(strToWrite)

Schrijft strToWrite naar de poort.

Events

4.9. SP_PortOpened()

Fired when port has been succesfully closed

4.10. SP_PortClosed()

Fired when port has been succesfully closed

4.11. SP_Reset()

Fired when port has been succesfully reset

4.12. InputReceived(Message)

Fired when input data from port has been received, message contains received data.

4.13. Get_StateDSR()

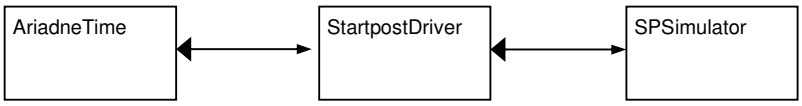
Leest de toestand van de DSR-pin uit (Hoog of Laag).

4.14. Get_StateCTS()

Leest de toestand van de CTS-pin uit (Hoog of Laag).

5. TestTooling

Ten behoeve van het testen van de startpostdriver zonder gebruik te maken van Ariadne en het startkastje, zijn er twee .Net hulpprogramma's beschikbaar:
- SPSimulator: simuleert het startkastje
- AriadneTime: simuleert Ariadne



SPSimulator werkt op de testinterface van de Startpostdriver. AriadneTime werkt op de Ariadne interface van de Startpostdriver.

6. Versiehistorie Startpostdriver

SPdriver operationele versie is v1.1 van 28Mei2007

Date	Version	Change
30Nov2007	1.3.0	Faciliteiten nieuw koppelkastje verwerkt. (online sensor, batterijspanningmeter)
28-05-2007	1.2.0	Door Iede aangepast. Pollt nu op Serial ipv callback.
28-05-2007	1.1.0	Door Gerrit aangepast. Op testinterface de datamessage opmaak vereenvoudigd.
23-04-2007	1.0.0	Eerste versie. Door Iede gecompileerd. Events vanaf Serial komen wel op .Net door maar niet op MSAccess

7. Wensen / verbeterlijst

Version	Date	Change
1.3	30Nov2007	Ook batterijmeting kunnen doen als de voedingssensor zegt dat het startkastje niet is aangesloten (zegt dat al bij 4,6 V.) Een timeout mechanisme moet zorgen dat een meting niet blijft hangen, in het geval er inderdaad geen startkastje is aangesloten. Je kunt dan iets meer zeggen over de situatie dat het startkastje wel is aangekoppeld, terwijl de voedingssensor dat niet ziet omdat de batterijspanning volgens het koppelkastje-door welke reden ook, maar bijv. een slechte verbinding - onder de 4,6 V ligt.