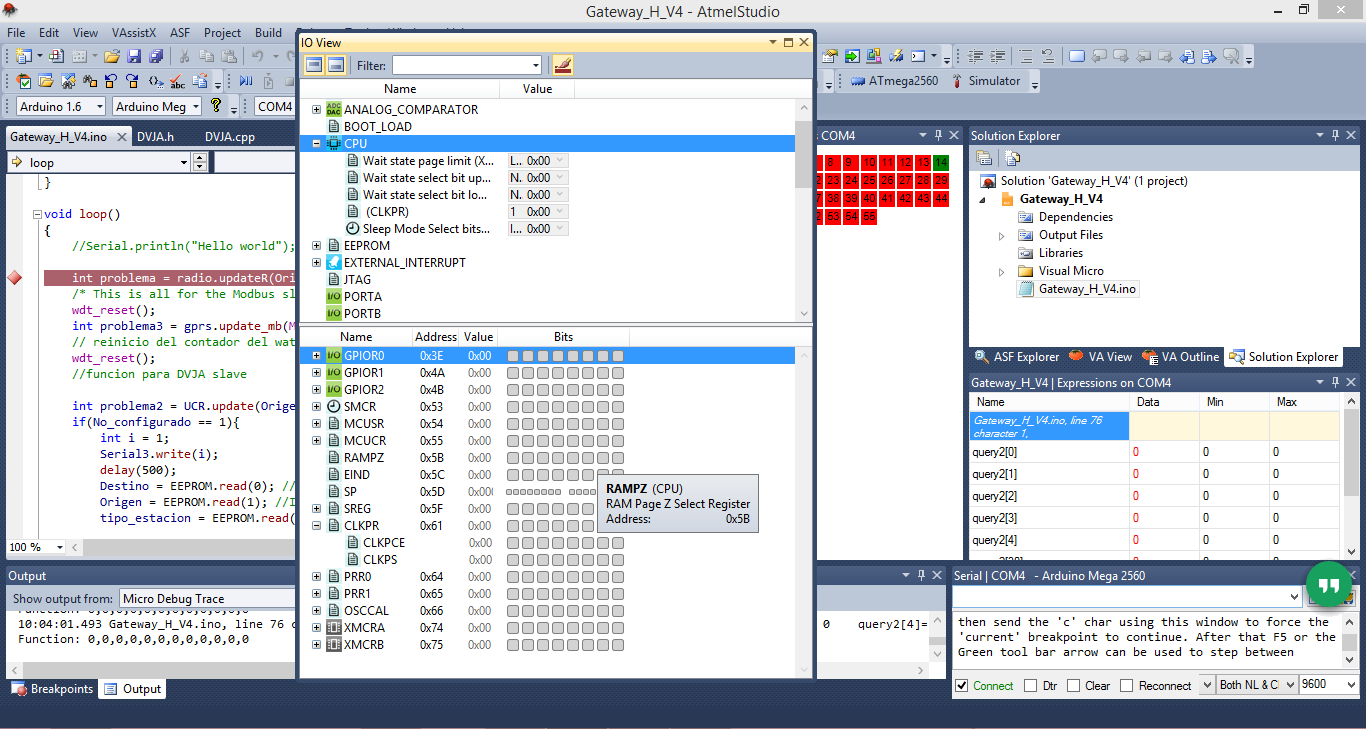


Academie IT en Mediadesign

Internet of Things – Techniek

Debug opdracht



4 oktober 2022

Arjen Huisman, Remco Veurink en Eddy Luursema

Inhoudsopgave

[1 Debugging 3](#_Toc115769022)

[1.1 Inleiding 3](#_Toc115769023)

[1.2 Proces van debuggen 3](#_Toc115769024)

[1.3 Manifesto 3](#_Toc115769025)

[1.4 Kennismaking debugger 4](#_Toc115769026)

[1.5 Kennismakingsopdracht 5](#_Toc115769027)

[1.6 Eindopdracht 5](#_Toc115769028)

[1.7 Prijs 6](#_Toc115769029)

[2 Sein- servodecoder 7](#_Toc115769030)

[2.1 Modelspoorbaanbesturing 7](#_Toc115769031)

[2.2 Loc-decoder 7](#_Toc115769032)

[2.3 Centrale unit 7](#_Toc115769033)

[2.4 Decoder voor accessoires 8](#_Toc115769034)

[2.5 Specificatie zelfbouwdecoder 8](#_Toc115769035)

# Debugging

## Inleiding

In deze opdracht oefent de student met de hardware debugger, die verbonden zit op een board met een Atmel 328. Dit wordt gecombineerd met de ontwikkelomgeving van Microchip genaamd Microchip Studio. Helaas is het de docenten niet gelukt deze debugger werkend te krijgen onder de standaard omgeving binnen IoT nl. Platform IO. De uiteindelijke opdracht is het debuggen van een door de docenten aangedragen stuk code.

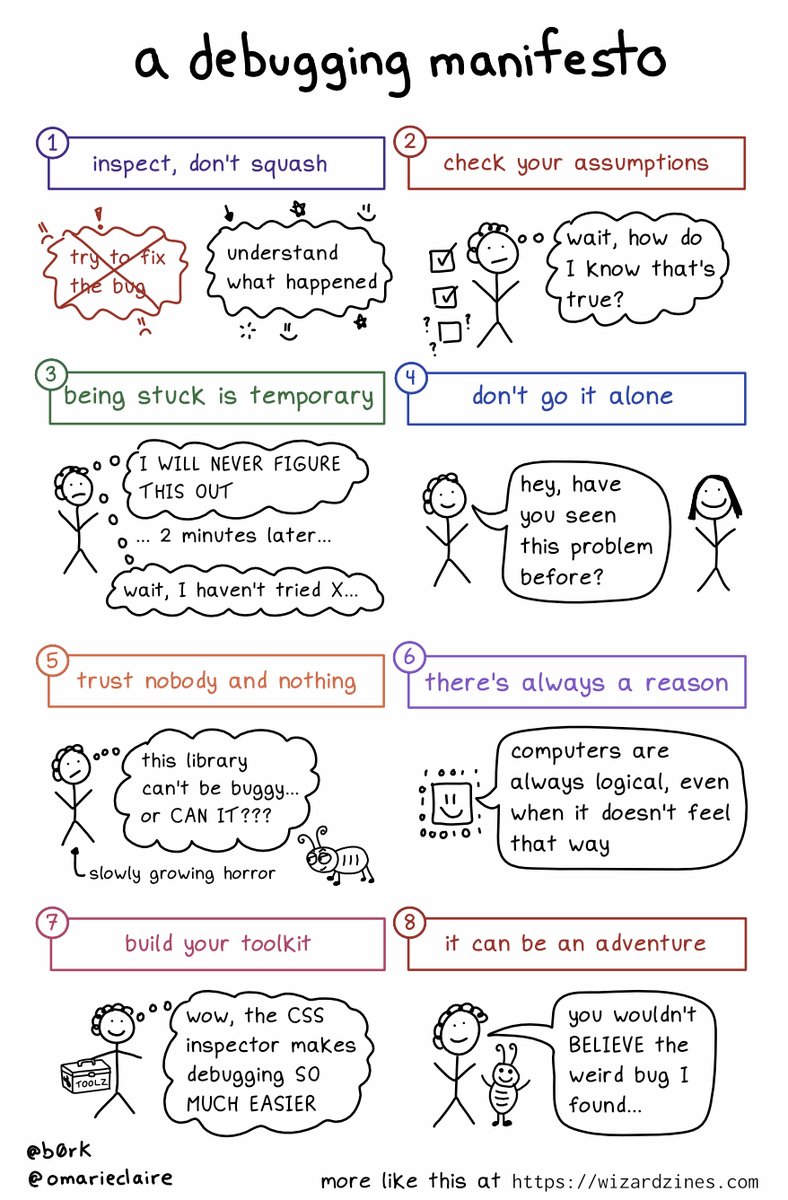
## Proces van debuggen

Om een willekeurig programma te kunnen debuggen met een professionele source-code-level debugger kun je de volgende functies uitvoeren:

* Zetten van breakpoints in de broncode. Slechtere debuggers kunnen alleen breakpoints zetten in de (door de compiler) gegenereerde machine-code
* Bekijken van alle registers, IO en geheugen tijdens een break
* In sommige gevallen wijzigen van registers, IO en geheugen tijdens een break

## Manifesto

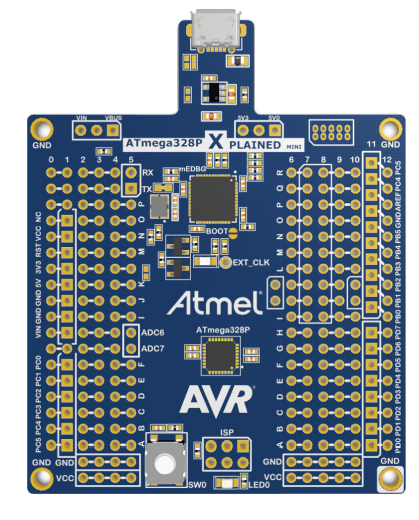
Grappig, maar zeer verhelderd een manifest (= lijst met standpunten en/of uitgangspunten) van Julia Evens:



Bron: https://twitter.com/b0rk/status/1570060516839641092?s=20&t=lrF25sPp1R6T3NIDMNbRrQ

## Kennismaking debugger

We gebruiken de ATMEGA328P XPLAINED MINI uit de doos. Installeer Microchip Studio en probeer de werking uit middels de code voor de beroemde 1s knipperende LED. Start de debugger en controleer de werking van het programma door op de juiste plek een breakpoint te plaatsen. Toon de IO in een apart venster.



Bron: <http://ww1.microchip.com/downloads/en/devicedoc/50002659a.pdf>

## Kennismakingsopdracht

Pak weer de avrlib-opdracht waarbij twee LED’s alternerend gaan branden zodra een button wordt ingedrukt. De afhandeling van de drukknop geschiedt via een interrupt met bijbehorende ISR. Om e.e.a. beter te bekijken bevat de main een zo snel mogelijke pin-toggle.

Maak een setje schermafdrukken als er vanuit het hoofdprogramma een interrupt plaatsvindt. Laat ook de aansturing van de IO duidelijk zien evenals het feit dat er een interrupt plaatsvindt.

## Eindopdracht

Zoals de studenten inmiddels weten is de docent geïnteresseerd in modelspoorbanen. Er is een gedeeltelijk prototype ontwikkeld voor een gecombineerde sein- en servo-decoder en deze decoder lijkt sterk op de eindopdracht van het vak Embedded Program Development uit de porpedeuse , maar helaas bevat de code een aantal fouten.

De studenten moeten aan de hand van een specificatie het product systematisch onderzoeken op fouten. Zodra één fout gevonden is wordt deze in een tabel genoteerd en wordt omschreven hoe de fout gevonden is. Ook moet een klein stukje van de oorspronkelijke code worden getoond en het verbetervoorstel.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fout | Omschrijving | Code regel | Originele code | Verbetervoorstel |
| 1 | Outputpoorten werken niet. PC springt naar andere routine. Afdrukken debugscherm. | 83 | Init\_LCD() | Init\_Ports() |
| 2 | LIGHTS-array wordt benaderd met [country][light][stop] maar de volgorde is verkeerd | 104 | LIGHTS[country][light][STOP] | LIGHTS[country] [STOP][light] |
| 3 | In de setup wordt setupGatepins nooit aangeroepen | 124 | leeg | setupGatePins(); |
| 4 | In de setup wordt TESTLIGHTINTERVAL = 1000 en dat kan niet wat de type is byte(max 255). In de opdracht moet die ook niet veranderd worden dus halen we die uit de setup weg en maken we het const | 54 | byte TESTLIGHTINTERVAL = 250; | Const byte TESTLIGHTINTERVAL = 250; |
| 5 | In de if conditie wordt er 1 = teken gebruikt, wat 2 moet zijn voor vergelijking | 251 | lightsRunTestStatus = LIGHTSTESTING | lightsRunTestStatus == LIGHTSTESTING |
| 6 | In moveGates en blinklights wordt millis in een long gestopt terwijl millis unsigned long terug geeft | 197, 182, | long currentMillis = millis();  long currentBlinkMillis = millis(); | Unsigned long currentMillis = millis();  Unsigned long currentBlinkMillis = millis(); |
| 7 | In de softwaretimer wordt de int -previousGateMillis gedaan met een unsigned long. | 56 | unsigned int previousGateMillis = 0; | unsigned long previousGateMillis = 0; |
| 8 | In testlight wordt een software timer gebruikt maar de previousmillis is lokaal gedeclareert wat niet hoort. Er bestaat al een globale variable Verder moet het unsigned zijn en moet je hem initiliaseren. | 155 | long previousTestBlinkMillis; | Verwijderen |
| 9 | Country is niet geintiliasieerd en is standaard op D terwijl in de specificatie op NL moet. | 47 | country t country; | country\_t country = NL; |
| 10 | Serial.begin is niet goed aangeroepen | 127 | Serial.begin(9600, SERIAL\_7E2); | Serial.begin(9600); |
| 11 | Input string ascii naar decimal gaat verkeerd | 230 | byte signalOrGateNumber = inputCommand[1] - 'O' - 1; | byte signalOrGateNumber = inputCommand[1] - '0' - 1; |
| 12 | For loop verkeerde condition | 174 | for (int light = 0; light <= NROFLIGHTSPERSIGNAL; light++) { | for (int light = 0; light < NROFLIGHTSPERSIGNAL; light++) { |
| 13 | Break in case vergeten | 236 |  | break; |
| 14 | Regel 213 zorgt voor underflow | 59 | byte gatePosition[NROFGATES]; | Signed int gatePosition[NROFGATES]; |
| 15 | Accolades vergeten. If else klopt niet | 206,  208, 217,  220 |  | } |
| 16 | De cases stonden verkeerd, nu moest je rijden als je A (voor afremmen) invoerde. BRAKE en DRIVE moeten dus omgewisseld worden. | 242, 244 | case 'R': setSignal(signalOrGateNumber, BRAKE);  break;  case 'A': setSignal(signalOrGateNumber, DRIVE);  break; | case 'R': setSignal(signalOrGateNumber, DRIVE);  break;  case 'A': setSignal(signalOrGateNumber, BRAKE);  break; |
| 17 | Hij werd hier eerst niet getoggled, maar dat wordt nu wel gedaan, omdat GATETESTING en GATERUNNING verkeerd om geplaatst waren. | 147 | gatesRunTestStatus = (gatesRunTestStatus == GATETESTING) ? GATETESTING : GATERUNNING; | gatesRunTestStatus = (gatesRunTestStatus == GATETESTING) ? GATERUNNING : GATETESTING; |
| 18 | De potentiometer moet een Analoge pin hebben ipv een digitale. | 45 | const byte POTENTIOPIN = 2; | const byte POTENTIOPIN = A2; |

Het is de bedoeling om in de opgeleverde software de fouten te zoeken m.b.v. de debugger en **niet met gebruik van print()**. Zodra een fout is gevonden mag deze pas worden verbeterd in overleg met de docent. Het is dus niet de bedoeling vanuit de specificatie een hernieuwd stuk software te schrijven.

## Prijs

De groep, die de meeste fouten vindt (en kan aantonen dat deze fouten gevonden zijn met de debugger) krijgt een gevulde koek.

# Sein- servodecoder

## Modelspoorbaanbesturing

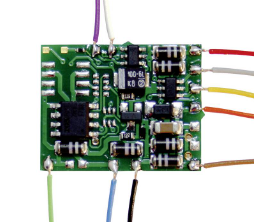
In een moderne modelspoorbaanbesturing wordt alles vanuit een digitale centrale aangestuurd:

* Rijrichting en -snelheid van de locomotieven
* Verlichting van locomotieven, wagons
* Seinen
* Wissels
* Spoorwegbomen, deuren van locomotiefloodsen etc.
* Straat- en huisverlichting, omgevingslicht inclusief dag/nacht en onweer
* Geluid in locomotieven, gebouwen etc.
* Camera’s geïnstalleerd op de modelspoorbaan en ingebouwd in locomotieven

Om uiteindelijk een modelspoorbaan geheel of gedeeltelijk automatisch te laten rijden zonder botsingen stuurt een computerprogramma op b.v. een laptop deze centrale aan.

## Loc-decoder

In een moderne modelspoorbaan is in alle locomotieven een decoder ingebouwd, die de rijrichting en rijsnelheid kan besturen alsmede de verlichting. Daarnaast hebben de nieuwste decoders geluiden zoals hoorn/fluit, stoom- en wielengeluiden, motorgeluid, gesproken oproepteksten etc. etc. Iedere decoder dus iedere locomotief heeft een uniek adres. Een commerciële decoder kost tussen de €20 en €60.



Bron: <https://www.conrad.nl/nl/p/tams-elektronik-41-05421-01-c-ld-w-42-mit-kabeln-locdecoder-met-kabel-2280294.html>

## Centrale unit

Een centrale unit stuurt een gecodeerde wisselspanning naar de rails. De locomotieven halen hieruit de energievoorziening van de motoren en de decoder. Daarnaast zorgen de gecodeerde signalen voor het aansturen van de decoder; een moderne centrale kan meer dan 100 locomotieven aansturen. Er bestaan grofweg twee standaarden voor gecodeerde wisselspanning DCC en het Märklin-protocol MFX. Ook zijn er helaas twee standaarden voor modelbaanrails: 2-spoor en 3-spoor. Een commerciële centrale kost tussen de €200 en €700.



Bron: <https://www.digikeijs.com/nl/dr5000-dcc-multi-bus-central.html>

## Decoder voor accessoires

Naast loc-decoders zijn er decoders voor alle andere te besturen onderdelen zoals seinen, wissels, deuren etc. Deze niet locomotieven heten accessoires. Een decoder voor accessoires kan specifiek alleen voor seinen zijn gemaakt, maar er bestaan ook gecombineerde decoders, die zowel seinen als b.v. servo’s kunnen aansturen. Een commerciële decoder kost tussen de €30 en €100.

Accessoire-decoders kunnen op verschillende manieren worden verbonden met de digitale centrale:

* Gecodeerde railsignaal
* CAN-bus
* Specifieke bus voor modelspoorbanen zoals Loconet, Bidib etc.



Bron: <https://www.digikeijs.com/nl/dr4018-schakel-decoder.html>

## Specificatie zelfbouwdecoder

De software in een decoder is ruwweg opgebouwd uit 2 delen: het ene deel verzorgt de communicatie met de b.v. DCC-bus; het andere deel stuurt de seinen, wissels of servo’s aan.

AANSTURING

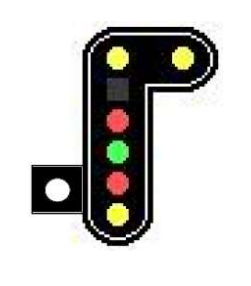
BUS

De hier beschreven specificatie betreft alleen het onderdeel aansturing.

Er kunnen meerdere seinen(signals) op de decoder worden aangesloten. Zo kan een decoder van tevoren geconfigureerd zijn voor 2 seinen van 3 lampen of 1 sein met 5 lampen. In Nederland bevatten de meeste seinen 3 lampen:



Er zijn echter landen met 4 of zelfs 8 lampen b.v. Frankrijk. De decoder zou dan één Frans sein kunnen besturen. De specificatie van een Frans sein is out of scope:



Een seinbeeld (aspect) is één combinatie van in-, uitgeschakelde of knipperende lampen, die de machinist één instructie geven b.v. afremmen, stoppen, rijden op zicht. Dit betekent dat bij één seinbeeld meerdere lampen (afhankelijk van het land) tegelijk kunnen branden/knipperen. Zie b.v. <https://nl.wikipedia.org/wiki/Spoorwegsein> voor een overzicht van de Nederlandse en Belgische seinbeelden.

De decoder kan (voorlopig) worden bediend door commando’s via de seriële poort. In een later ontwerp zal de decoder worden aangesloten op het DCC-signaal van de centrale; dit valt buiten de scope. De decoder kan seinbeelden voor meerdere landen tonen; het land is instelbaar via een commando. De huidige code is geconfigureerd voor 2 landen (NL en D) en 3 lampen per sein.

In Nederland zijn de seinbeelden Rood = stoppen, Geel = afremmen, Groen = rijden.

In Duitsland Rood = stoppen, Groen knipperen (1 Hz) + Geel = afremmen en Groen = rijden.

Er kunnen meerdere servo’s worden aangesloten om slagbomen van een overweg te sluiten. De slagbomen openen en sluiten gedurende een bepaalde instelbare tijd. De seinen en slagbomen werken volkomen onafhankelijk van elkaar zowel bij gebruik als bij testen. De huidige code is geconfigureerd voor 2 servo’s. De sluitings- en openingstijd is te beïnvloeden met een potentiometer en ligt tussen 1 en 10 sec. Het naspelen van een nazwaaiende slagboom is buiten scope.

Commando’s:

* N – Nederland
* D – Duitsland
* H1 – stopsignaal sein 1
* H2 – stopsignaal sein 2
* R1 – rijden signaal sein 1
* R2 – rijden signaal sein 2
* A1 – afremmen signaal sein 1
* A2 – afremmen signaal sein 2
* T – test seinen. Alle seinlampen branden afzonderlijk telkens 1 sec.; nogmaals T stop testen
* G – test slagbomen; nogmaals G stop testen
* O1 – open slagboom 1
* O2 – open slagboom 2
* S1 – sluit slagboom 1
* S2 – sluit slagboom 2

Het systeem start met als land NL en met het stopsignaal op alle seinen; de slagbomen staan in ruststand. Het is mogelijk dat de specificaties niet volledig zijn.

Het aansluitschema bij de gegeven code ziet er als volgt uit:

