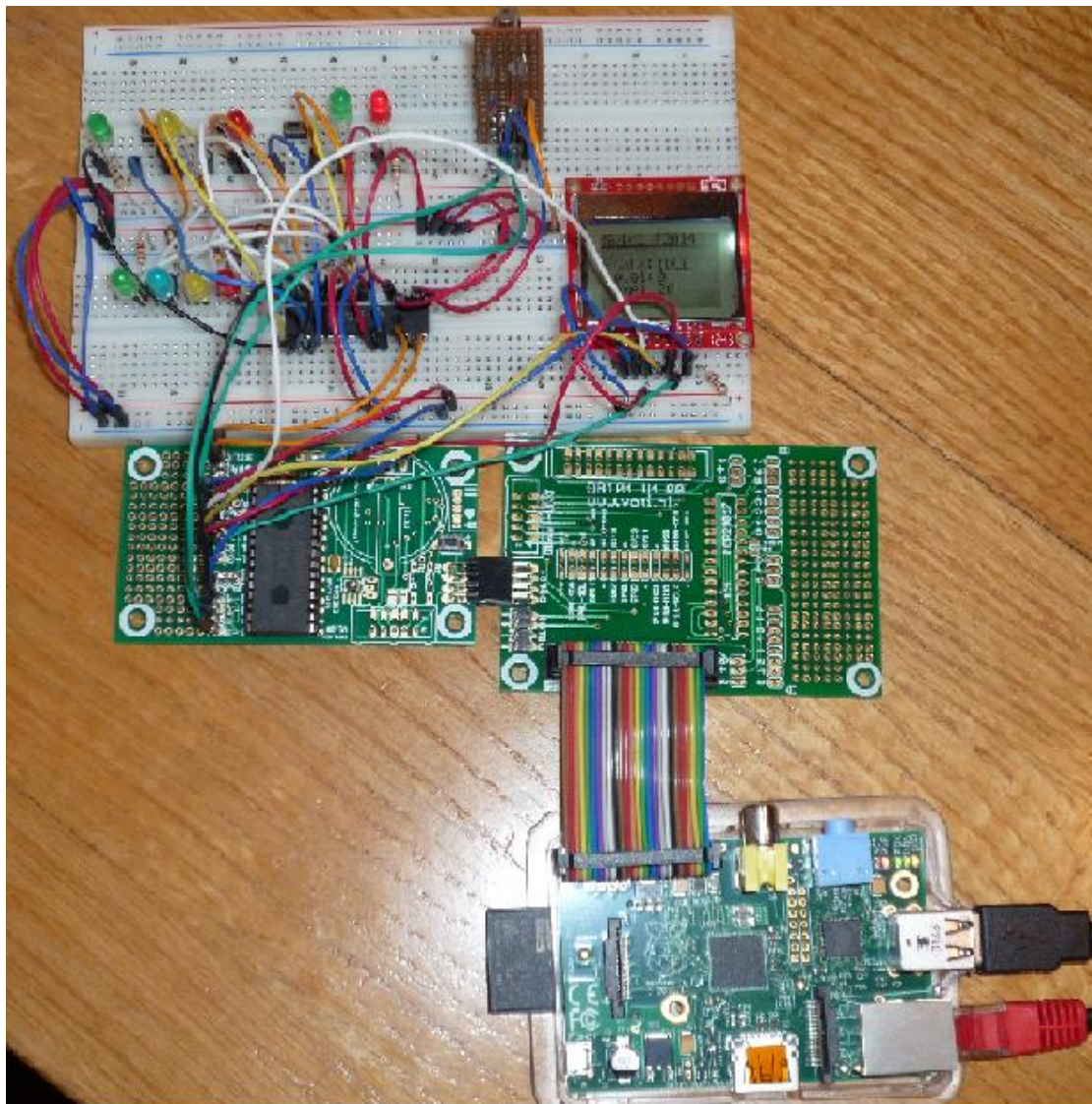


Themaopdracht 6: Domotica

Osiriscode: TCTI-V2THO6-14

Beschrijving wasmachine-emulator



| Auteurs | Datum | Versie |
|------------|-----------------|--------|
| M. Wensink | 5 november 2015 | 1.1 |

Inhoud

| | |
|---|----------|
| 1. INLEIDING..... | 2 |
| 2. DE GEËMULEERDE HARDWARE | 3 |
| 2.1 DE HARDWARE | 3 |
| 2.2 AANSLUITING VAN DE COMPONENTEN | 4 |
| 2.2.1 DE I/O-EXPANDER..... | 4 |
| 2.2.2 HET DISPLAY..... | 4 |
| 2.2.3 DE TRILMOTOR..... | 4 |
| 2.2.4 DE NOODKNOP..... | 5 |
| 2.2.5 DE DEURVERGRENDELING | 5 |
| 2.2.6 UITSCHAKELING STROOM..... | 5 |
| 2.2.7 DE OVERIGE LEDS | 5 |
| 2.3 FYSIEKE GEGEVENS WASMACHINE | 5 |
| 3. HET COMMUNICATIEPROTOCOL..... | 6 |
| 3.1 HET FORMAAT VAN BERICHTEN | 6 |
| 3.2 REQUESTS EN COMMANDS | 7 |
| 3.3 MACHINETOESTANDEN | 8 |
| 4. INSTALLATIE EN AANSTURING VAN DE EMULATOR | 9 |
| 4.1 INSTALLATIE..... | 9 |
| 4.2 CONFIGURATIE RASPBERRY PI | 9 |
| 4.3 COMMUNICATIE ARM - RASPBERRY PI..... | 9 |

1. Inleiding

In themaopdracht 6, Domotica, wordt voor de realisatie van een wasmachine die via het internet kan worden bediend, gebruik gemaakt van een emulator voor de hardware van de wasmachine. Deze emulator draait op een ARM Cortex-M0 processor op een DB103-bordje, met daaraan gekoppeld een breadboard met een MCP23017 I/O-expander, Nokia 5510 LCD, een trilmotor en een aantal knoppen en leds.

Dit document beschrijft de hardware componenten van de wasmachine die worden geëmuleerd, waarmee ze worden geëmuleerd en hoe ze moeten worden aangesloten op de ARM-processor, respectievelijk de I/O-expander.

Voorts wordt het protocol beschreven waarmee, via een seriële interface, commando's aan de emulator kunnen worden gestuurd.

Ook wordt beschreven hoe de emulator kan worden geïnstalleerd op de ARM-processor en hoe met een Raspberry Pi commando's naar de emulator kunnen worden gestuurd.

2. De geëmuleerde hardware

Dit hoofdstuk beschrijft over welke hardware componenten de geëmuleerde wasmachine beschikt en met welke middelen dit wordt gerealiseerd.

2.1 De hardware

De wasmachine beschikt over de volgende componenten:

- Een wasmachinemotor die wordt geëmuleerd door een trilmotor die links- en rechtsom kan draaien voor het uitvoeren van wasbewegingen en het centrifugeren. De motor heeft een instelbaar toerental van 0 tot en met 1600 rpm. De omwentelingssnelheid kan worden ingesteld met stappen van 25 rpm.
- Een deurslot met vergrendeling dat wordt geëmuleerd door een drukknop en twee leds. Als de deur open staat, de initiële toestand, brandt een van de leds. Nadat de geëmuleerde wasmachine is gestart, gaat na een druk op de knop de led uit, ten teken dat de deur dicht is. Hierna accepteert de emulator een commando om de deur te vergrendelen en vervolgens ontgrendelen, waarbij de tweede led aan, respectievelijk uit gaat. Als de deur ontgrendeld is, kan met de drukknop de deur weer worden geopend, waarbij de eerste led weer aan gaat.
- Een noodknop, die wordt geëmuleerd door een drukknop en een led. Als de geëmuleerde wasmachine is gestart, gaat na een druk op de knop de led branden. De wasmachine is dan gestopt en zal geen besturingscommando's meer accepteren.
- Een verwarmingselement, dat wordt geëmuleerd door een led. De led brandt als de verwarming aan is. Het verwarmen en afkoelen van het water wordt gesimuleerd in software.
- Een kraan, die wordt geëmuleerd door een led. De led brandt als de kraan open is. Het vullen met water wordt gesimuleerd in software.
- Een pomp, die wordt geëmuleerd door een led. De led brandt als de pomp is ingeschakeld. Het pompen wordt gesimuleerd in software.
- Een zeep toevoer, die wordt geëmuleerd door een led. De led brandt als de zeep toevoer open is en dus het water door het zeepbakje stroomt.
- Een signaal led die kan aangeven dat het wasprogramma is afgelopen.

Verder zijn er nog twee componenten die in feite geen onderdeel zijn van de wasmachine:

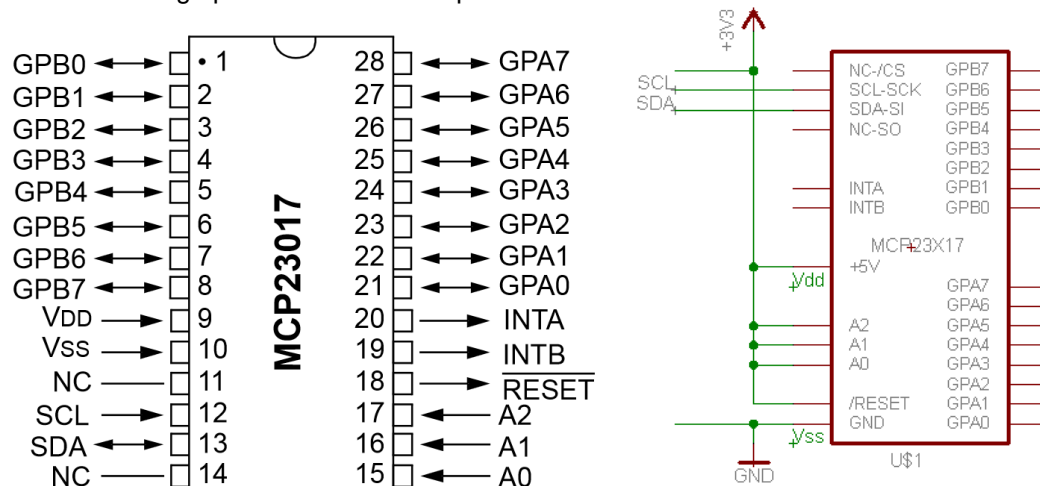
- Een knop met een led, waarmee het uitvallen van de stroom kan worden gesimuleerd. Na het drukken op de knop gaat de led aan, ten teken dat de stroom is uitgevallen. De geëmuleerde wasmachine stopt dan alle activiteit. Het opnieuw indrukken van de knop schakelt de stroom weer in, waarbij de wasmachine in een toestand komt waarin de deur is ontgrendeld, het water nog op peil is, maar wel afgekoeld. De emulator zal dan weer commando's uitvoeren.
- Een display, waarop de toestand waarin de emulator zich bevindt wordt weergegeven, de watertemperatuur in °C en voor hoeveel procenten van het totale volume de trommel is gevuld met water.

2.2 Aansluiting van de componenten

De software van de emulator is niet configureerbaar en werkt alleen goed als bovenstaande componenten zijn aangesloten op de pinnen zoals hieronder is beschreven.

2.2.1 De I/O-expander

De SCL en SDA pinnen van de MCP23017 moeten worden aangesloten op de overeenkomstige pinnen van de ARM-processor.



De pinnen A0, A1, A2 en RESET moeten worden aangesloten op de + van de voeding (zie de rechter afbeelding). De NC- en de INT-pinnen blijven vrij.
De VDD-pin moet worden aangesloten op de + van de voeding en de VSS-pin op de -.

2.2.2 Het display

Het Nokia display dient op de volgende manier te worden aangesloten op de GPIO-pinnen 1.0 t/m 1.4 van de ARM-processor:

GPIO1_0 : SCE
GPIO1_1 : RES
GPIO1_2 : D/C
GPIO1_3 : SDIN (MOSI)
GPIO1_4 : SCLK

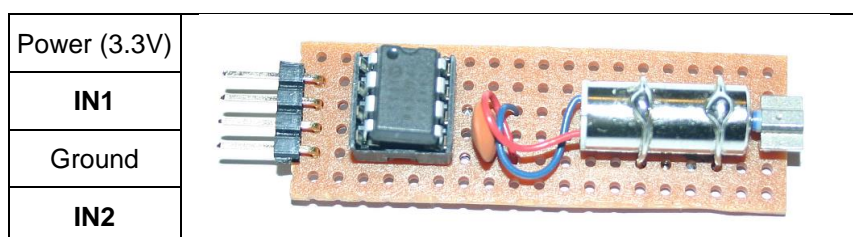
Andere LCD pinnen:

GND : 0V

VCC : 3V3

LED : ledverlichting, verbinden aan 3V3 via 100Ω weerstand.

2.2.3 De trilmotor



De pinnen IN1 en IN2 van de motor moeten worden aangesloten op de GPIO-pinnen 1.8 en 1.9 van de ARM-processor. Als de motor bij aansturing de verkeerde kant op draait, moeten de aansluitingen worden verwisseld.

2.2.4 De noodknop

De knop en de led voor de noodknop moeten worden aangesloten op respectievelijk de pinnen GPB0 en GPA7 van de I/O-expander.

2.2.5 De deurvergrendeling

De knop voor de deurvergrendeling moet worden aangesloten op pin GPB2 van de I/O-expander en de leds op respectievelijk de pinnen GPA0 en GPA1. De led op pin GPA0 geeft aan of de deur is gesloten of niet en de led op pin GPA1 geeft aan of de deur is vergrendeld of niet.

2.2.6 Uitschakeling stroom

De knop en de led voor het uitschakelen van de stroom dienen te worden aangesloten op respectievelijk de pinnen GPB1 en GPA6 van de I/O-expander

2.2.7 De overige leds

| Functie | Pin |
|-------------|-------------------------------|
| Verwarming | GPA2 van de I/O-expander |
| Kraan | GPA3 van de I/O-expander |
| Pomp | GPA4 van de I/O-expander |
| Zeeptoevoer | GPA5 van de I/O-expander |
| Signaalled | GPIO 1.5 van de ARM-processor |

2.3 Fysieke gegevens wasmachine

De geëmuleerde wasmachine heeft een trommel met een diameter van 50 cm en een diepte van 40 cm.

Het verwarmingselement van de wasmachine heeft een capaciteit van 3000 Joule/sec en het water koelt af met 100 Joule/sec.

Het water waarmee de wasmachine wordt gevuld heeft een temperatuur van 14°C. Bij de start heeft de machine de omgevingstemperatuur van 20°C.

Om de simulatie van het verwarmingsproces niet te lang laten duren, wordt het proces met een factor 8 versneld. Dat betekent dus in feite hetzelfde als dat de capaciteit van het verwarmingselement 24000 Joule/sec is.

3. Het communicatieprotocol

Via de seriële aansluiting (UART) van de ARM-processor kunnen commando's worden gestuurd naar de emulator. De emulator reageert altijd met een tegenbericht op een ontvangen bericht.

In dit hoofdstuk wordt het protocol voor deze communicatie beschreven alsmede een aantal bijzonderheden.

3.1 Het formaat van berichten

Ieder bericht dat naar de emulator wordt gestuurd bestaat uit twee bytes (type `uint8_t`).

| | |
|---------|---------|
| REQUEST | COMMAND |
|---------|---------|

De eerste byte is de Request-byte, die aangeeft voor welke hardware component het bericht bestemd is. De tweede byte, de Command-byte, bepaalt, afhankelijk van de Request-byte, het commando dat moet worden uitgevoerd, of representeert een bepaalde waarde, b.v. het in te stellen toerental, of heeft geen betekenis.

| | |
|----------|--------|
| RESPONSE | STATUS |
|----------|--------|

De emulator antwoordt altijd met een tegenbericht dat ook uit twee bytes bestaat. De Respons-byte bestaat uit de ontvangen Request-byte waarin het hoogste bit (0x80) is aangezet. De tweede byte bevat de status van de hardware component na uitvoering van het commando of de opgevraagde waarde, zoals b.v. het waterniveau.

Voorbeeld:

Een bericht om de status van de emulator op te vragen bestaat uit de bytes 0x01 en 0x01. De emulator zal antwoorden met de bytes 0x81 en 0x04 als deze is gestart.

Als een fout geformatteerd bericht wordt gestuurd naar de emulator, antwoordt deze met de bytes 0xEE en 0xEE. Bovendien verschijnt dan op het display de melding '>>ERROR<<'. Hierna zal de emulator niet meer reageren op berichten.

In het geval dat de knop, die het uitschalen van de stroom simuleert, wordt ingedrukt, antwoordt de emulator op ieder ontvangen bericht met de bytes 0xFF en 0xFF. Zodra de stroom weer wordt ingeschakeld, zal de emulator weer reageren op commando's.

Bij het uitvoeren van een commando dat geen effect heeft, b.v. het vergrendelen van de deur die al eerder is vergrendeld, wordt met dezelfde Status-byte geantwoordt als bij de eerste keer dat het commando is uitgevoerd. Als een commando niet kan worden uitgevoerd, b.v. het aanzetten van de verwarming, terwijl de machine nog niet is gestart, dan antwoordt de emulator met de Status-byte 0x00.

Bij de request om het toerental van de motor op een bepaalde waarde in te stellen, dient de Command-byte een van de waarden 0x00 t/m 0x40 te hebben (0 t/m 64 decimaal).

De waarde 0x00 stopt de motor en een waarde van 0x01 t/m 0x40 stelt de motor in op een toerental (rechtsom draaiend) van 1*25 t/m 64*25 rpm, dus van 25 t/m 1600 rpm.

Als de motor de andere kant (linksom) moet opdraaien, moet het hoogste bit van het byte voor het gewenste toerental worden gezet (OR operatie met 0x80).

De emulator antwoordt met het ingestelde toerental of het eerder ingestelde toerental als de waarde ongeldig is.

3.2 Requests en Commands

De volgende tabel geeft een overzicht van alle Request-bytes en de Command-bytes die in combinatie daarmee kunnen worden gebruikt. In de twee rechter kolommen van de tabel staan de mogelijke antwoorden die de emulator kan geven. Het antwoord van de emulator is mede afhankelijk van de toestand waarin de emulator verkeert (zie H.3.3).

| Request | Byte | Command | Byte | Antwoord | Byte |
|---------------------------|------|------------|------------------------|--------------------|------------------------|
| MACHINE_REQ | 0x01 | STATUS_CMD | 0x01 | HALTED | 0x01 |
| | | START_CMD | 0x10 | IDLE | 0x02 |
| | | STOP_CMD | 0x20 | RUNNING STOPPED | 0x04 0x08 |
| DOOR_LOCK_REQ | 0x02 | STATUS_CMD | 0x01 | OPENED | 0x01 |
| | | LOCK_CMD | 0x40 | CLOSED | 0x02 |
| | | UNLOCK_CMD | 0x80 | LOCKED | 0x04 |
| WATER_VALVE_REQ | 0x03 | STATUS_CMD | 0x01 | OPENED | 0x01 |
| | | OPEN_CMD | 0x10 | CLOSED | 0x02 |
| | | CLOSE_CMD | 0x20 | | |
| SOAP_DISPENSER_REQ | 0x04 | STATUS_CMD | 0x01 | OPENED | 0x01 |
| | | OPEN_CMD | 0x10 | CLOSED | 0x02 |
| | | CLOSE_CMD | 0x20 | | |
| PUMP_REQ | 0x05 | STATUS_CMD | 0x01 | ON | 0x08 |
| | | ON_CMD | 0x10 | OFF | 0x10 |
| | | OFF_CMD | 0x20 | | |
| WATER_LEVEL_REQ | 0x06 | - | - | Niveau in % | 0 - 0x64 |
| HEATING_UNIT_REQ | 0x07 | STATUS_CMD | 0x01 | ON | 0x08 |
| | | ON_CMD | 0x10 | OFF | 0x10 |
| | | OFF_CMD | 0x20 | | |
| TEMPERATURE_REQ | 0x08 | - | - | Temp in °C | 0 - 0x64 |
| SET_RPM_REQ | 0x0A | RPM | { 0x80 } 0 - 0x40 | RPM | { 0x80 } 0 - 0x40 |
| GET_RPM_REQ | 0x09 | - | - | RPM | { 0x80 } 0 - 0x40 |
| SIGNAL_LED_REQ | 0x0B | STATUS_CMD | 0x01 | ON | 0x08 |
| | | ON_CMD | 0x10 | OFF | 0x10 |
| | | OFF_CMD | 0x20 | | |

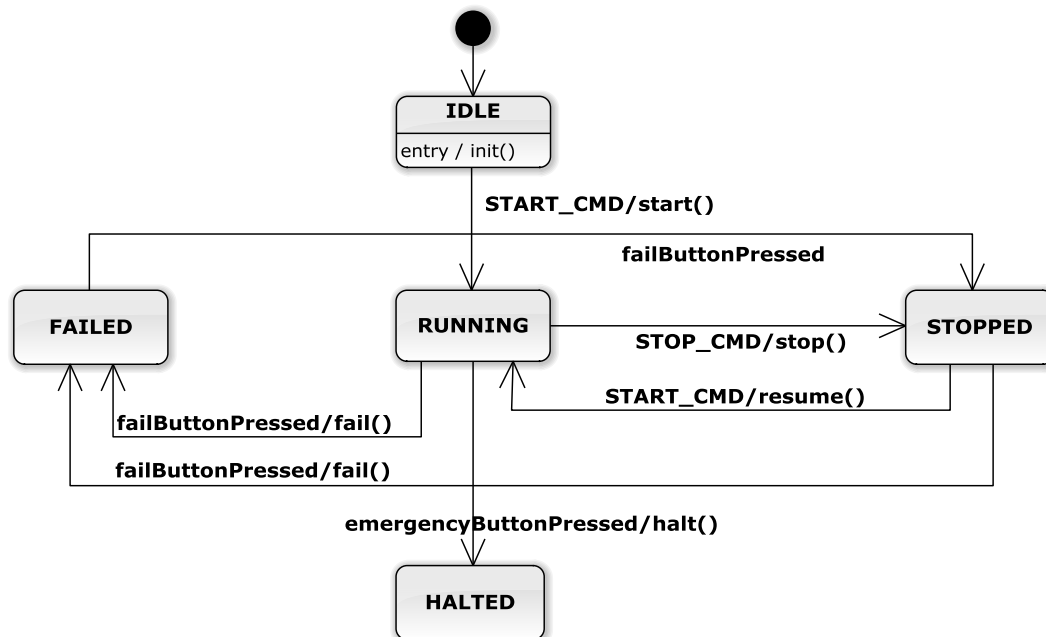
Zoals eerder beschreven antwoordt de emulator met de Status-byte 0x00 als een commando niet kan worden uitgevoerd.

Als een Request -of Command-byte wordt gestuurd die niet in de tabel voorkomt, dan antwoordt de emulator met de Response- en Status-bytes 0xEE en 0xEE.

Het sturen van een bericht, in de toestand dat op de emulator een stroomuitval wordt gesimuleerd, wordt beantwoord met de Response- en Status-bytes 0xFF en 0xFF.

3.3 Machinetoestanden

Zoals in het STD hieronder is te zien, kan de emulator kan zich in 5 toestanden bevinden.



Als de emulator wordt gestart, komt deze in de toestand IDLE. In deze toestand reageert de emulator alleen op berichten met de Request-byte MACHINE_REQ en een van de Commando-bytes STATUS_CMD of START_CMD. Ook op het indrukken van een van de knoppen zal de emulator niet reageren.

Na het startcommando, komt de emulator in de toestand RUNNING. In deze toestand worden alle commando's door de emulator beantwoord en wordt ook gereageerd op het indrukken van een van de knoppen.

De failButton veroorzaakt de simulatie van een stroomuitval, waarbij de emulator naar de toestand FAILED gaat. In deze toestand is het display leeg en stoppen de simulaties van het vullen, verwarmen en pompen. Na een tweede keer indrukken van de failButton, gaat de emulator naar de toestand STOPPED.

Ook als in de toestand RUNNING het bericht met de bytes MACHINE_REQ en STOP_CMD naar de emulator wordt gestuurd, gaat de emulator naar de toestand STOPPED.

In de toestand STOPPED blijft het waterniveau in de machine hetzelfde, maar koelt het water af tot minimaal kamertemperatuur (20°C). In de toestand STOPPED is de deur ontgrendeld en kan deze worden geopend door een druk op de deurknop.

Als de emulator in de toestand RUNNING is en op de noodknop wordt gedrukt, komt de emulator in de toestand HALTED en zal niet meer reageren op berichten anders dan het opvragen van de machine status.

4. Installatie en aansturing van de emulator

Dit hoofdstuk beschrijft hoe de emulator kan worden geïnstalleerd op de ARM-processor en hoe de Raspberry Pi moet worden geconfigureerd om via de UART met de ARM-processor te kunnen communiceren.

4.1 Installatie

Om de emulator te kunnen installeren is het programma `lpc21-97hr.exe` nodig. Dit is te vinden in de ZIP-file *Software.zip* dat beschikbaar is via de SharePoint-site van de thema-opdracht. De HEX-file van de emulator bevindt zich ook in de ZIP-file.

Na het uitpakken van de ZIP-file kan, nadat het ARM DB103-bordje m.b.v. een DB100 (zie het document *2015-2016-V2TH06-notes.pdf* dat eveneens is te vinden in de file *Software.zip*) connector is verbonden met een USB-port van een laptop of PC, worden geüpload met het commando

```
lpc21-97hr -control -verify -term Emulator.hex COM4 38400 12000
```

Als een ander COM-port nummer is toegewezen aan de USB-port, dan moet natuurlijk dit nummer worden gebruikt of zal de port een ander nummer moeten worden toegewezen.

4.2 Configuratie Raspberry Pi

Om de Raspberry Pi via de UART te kunnen laten communiceren met de ARM-processor, moet de configuratie van de Raspberry Pi worden aangepast.

Zie daarvoor het document *2015-2016-V2TH06 notes.pdf*.

4.3 Communicatie ARM - Raspberry Pi

Het besturingsprogramma van de wasmachine, dat draait op de Raspberry Pi, kan via de UART communiceren met de emulator die draait op de ARM-processor. In de ZIP-file *Software.zip* is een kleine C++-bibliotheek opgenomen, genaamd *libserial*, waarmee de UART-hardware kan worden geprogrammeerd. De bibliotheek bevat de klasse `LibSerial` die de nodige functionaliteit bevat om te schrijven en lezen. De ZIP-file bevat ook een met Doxygen gegenereerde help-file waarin de klasse wordt beschreven.

De UART op het DB103-bordje is ingesteld op een baud rate van 9600 en de instelling 8N1. Bij het openen van de UART door het besturingsprogramma dient daar rekening mee te worden gehouden.

Bij het ontwerpen van het besturingsprogramma van de wasmachine, een realtime systeem, moet er rekening mee worden gehouden dat de UART een wachttijd heeft van bijna 10 ms. D.w.z. na het schrijven van een commando naar de UART, duurt het ongeveer 10 ms voordat het antwoord van de emulator beschikbaar is.

Laat daarom de taak die zorgt voor de communicatie met de emulator niet nodeloos wachten op antwoord, maar zorg ervoor dat deze zijn beurt afgeeft aan een andere taak. Een eventuele andere taak mag dan natuurlijk geen gebruik maken van de UART!

Let er ook op dat het besturingsprogramma de emulator niet met een te hoge frequentie vraagt om het waterniveau of temperatuur, want dat zal de ARM-processor erg zwaar belasten. Het heeft bovendien geen zin om b.v. om de 50 ms de temperatuur op te vragen als deze maar eens in de 200 ms een graad hoger wordt.

De ZIP-file *Software.zip* bevat het Linux-programma *wm_tester* waarmee kan worden getest of alle hardware componenten goed zijn aangesloten en of de seriële communicatie werkt.