

Onderzoek Communicatieprotocollen

Verantwoordingsverslag Stage EVAbits

Friedel Schön

14-06-2024

Titel	Onderzoek Communicatieprotocollen
Ondertitel	Verantwoordingsverslag Stage EVAbits
Plaats en Datum	Assen, 14-06-2024
Auteur	Friedel Schön
Studentnummer	445455
E-mail	f.schon@st.hanze.nl
Bedrijf	EVAbits
Locatie	Skagerak 26 9723JR Groningen
Bedrijfsbegeleider	Jan Stegenga, Erik Krallen
Opleiding	Software Engineering, HBO-ICT
School	Hanze Groningen
Docentbegeleider	Kobus Bijker Nienke van der Spek

Voorwoord

Voorwoord

Samenvatting

Samenvatting

Inhoudsopgave

Voorwoord	3
Samenvatting	4
1 Hoofdstuk Introductie	6
2 Hoofdstuk Opdrachtbeschrijving	7
2.1 Probleem	7
2.2 Oplossing	7
2.3 Doel	7
2.4 Onderzoek	7
2.5 Stakeholders	8
2.6 Beroepsproducten	8
2.6.1 Demonstratie Zephyr OS op Efinix' Sapphire Core	8
2.6.2 Demonstratie communicatie tussen de FPGA en een computer via EVAbus (RISC-V Core)	8
2.6.3 Demonstratie communicatie tussen de FPGA en RISC-V Core (VHDL)	8
2.6.4 Implementatie van een driver in Zephyr OS voor het aansturen van de FPGA	9
2.6.5 Onderzoeksrapport communicatieprotocollen	9
3 Hoofdstuk Onderzoek	10
3.1 Protocollen	10
3.1.1 Inter-Integrated-Circuit Protocol	10
3.1.2 Serial Peripheral Interface	10
3.1.3 Universal Asynchronous Receiver/Transmitter	11
3.2 Toepassing op EVAjig	11
3.3 Conclusie	11
3.4 Aanbevelingen	11
4 Hoofdstuk Projectaanpak en Risico's	12
4.1 Planning	12
4.1.1 Requirements	12
4.2 Risico's	12
5 Hoofdstuk HBO-I competenties	13
5.1 Onderzoeken	13
5.2 Projectmatig Werken	13
5.3 Analyseren	13
5.4 Ontwerpen	13
5.5 Realiseren	13
5.6 Schriftelijke Vaardigheden	13
5.7 Adviseren	13
Begrippenlijst	14
Literatuurlijst	15

1 Hoofdstuk Introductie

EVAbits is een klein bedrijf gevestigd in Euvelgunne te Groningen. Het bedrijf is gespecialiseerd in het schrijven en onderhouden van software en firmware in de kader van Embedded Programming en Architecture Design. Het bedrijf is begonnen met opdrachtprogrammering, maar is ook sinds een aantal jaren bezig met hun eigen product: de EVAjig. De EVAjig is een testapparaat dat speciaal is ontworpen om het proces van programmeren, instellen en testen van printplaatasmblages (PCBA) te vergemakkelijken. Dit apparaat is uitgerust met een dashboard en een online portaal. Een van de opvallende kenmerken van de EVAjig is de mogelijkheid om cassettes met PCBA's gemakkelijk te verwisselen. Dit betekent dat je met hetzelfde apparaat verschillende soorten PCB ontwerpen kunt testen zonder veel tijd kwijt te zijn aan het aanpassen van de testopstelling. Daarnaast biedt de EVAjig een breed scala aan algemene functies, waardoor het een veelzijdig hulpmiddel is voor verschillende testbehoeften. Dit maakt het apparaat bijzonder geschikt voor bedrijven en technici die regelmatig met verschillende PCB ontwerpen werken. De EVAjig is geïmplementeerd door middel van een microcontroller. Een microcontroller is een klein circuit dat als een complete computer op een chip fungeert. Het bevat alle essentiële componenten die een computer nodig heeft om te werken, zoals een processor (de 'hersenen' van de computer), geheugen (om gegevens op te slaan), en invoer-/uitvoerpoorten (om te communiceren met andere apparaten). Microcontrollers worden vaak gebruikt in elektronische apparaten en systemen om specifieke taken uit te voeren. Wat microcontrollers bijzonder maakt, is hun vermogen om geprogrammeerd te worden voor specifieke taken. Dit betekent dat je de microcontroller instructies kunt geven om bepaalde acties uit te voeren op basis van de informatie die het ontvangt van sensoren of andere invoerapparaten. Dit programmeerbare karakter maakt het flexibel en toepasbaar in een breed scala van toepassingen, van eenvoudige huishoudelijke apparaten tot complexe industriële machines.

2 Hoofdstuk Opdrachtbeschrijving

2.1 Probleem

Een nadeel van de huidige implementatie met een microcontroller is, dat communicatie wat vaak door middel van extra chips wordt afgehandeld, niet flexibel is. Communicatieprotocollen werken heel snel, vaak in de miljoenen bytes per second. Een microcontroller is niet snel genoeg om dit in software op te lossen en dus laat dit over aan een extern chip. Op de microcontroller is de processor vast aan de communicatie chipjes gekoppeld, zonder de mogelijkheid iets aan deze constructie te veranderen. Meestal is dit geen probleem voor de applicatie van een microcontroller, maar voor de EVAjig is dit een beperkend criteria door de brede schaal aan PCBA's die moeten getest kunnen worden.

2.2 Oplossing

EVAbits heeft dit probleem herkent en heeft een opdracht aangemaakt om deze beperking te verhelpen. Het bedrijf heeft kennis in field-programmable gate-array's (FPGA). Een FPGA is een type circuit door de gebruiker geconfigureerd kan worden om specifieke taken uit te voeren. Deze configuratie gebeurt door het programmeren van de logische blokken en verbindingroutes binnen de FPGA. Hierdoor kan een FPGA worden aangepast aan een breed scala van toepassingen, van eenvoudige logische functies tot complexe digitale systemen. De kern van een FPGA bestaat uit duizenden tot miljoenen programmeerbare logische blokken en een netwerk van interconnecties. Deze logische blokken kunnen worden geconfigureerd om elementaire functies uit te voeren, zoals AND, OR, en NOT-operaties, en ze kunnen worden gecombineerd om complexere digitale circuits te vormen. De interconnecties zorgen ervoor dat deze logische blokken op verschillende manieren met elkaar kunnen worden verbonden, afhankelijk van de vereisten van de toepassing. Een van de grootste voordelen van een FPGA is zijn flexibiliteit. Omdat het apparaat kan worden geherprogrammeerd, kunnen ontwerpers hun hardwareontwerpen aanpassen en optimaliseren zonder dat nieuwe fysieke chips nodig zijn. Daarnaast kunnen FPGAs meerdere taken parallel uitvoeren, wat resulteert in hoge verwerkingssnelheden. Dit maakt ze ideaal voor gebruik in toepassingen zoals signaalverwerking, communicatiesystemen, en complexe rekenintensieve taken. In vergelijking met een microcontroller biedt een FPGA veel meer flexibiliteit. Waar een microcontroller vast gekoppeld is aan bepaalde communicatiechips, kan een FPGA deze functionaliteit emuleren door de hardware intern opnieuw te configureren. Dit betekent dat een FPGA verschillende soorten communicatieprotocollen kan nabootsen zonder de beperkingen van vooraf vastgestelde hardware. Hierdoor kan een FPGA meerdere functies en communicatieprotocollen tegelijkertijd afhandelen, wat ideaal is voor complexe en veelzijdige toepassingen. De configuraties binnen de FPGA zijn beschrijven in een Hardware Description Language (HDL), in deze taal worden signalen en primitieve logica's beschreven.

2.3 Doel

Het doel van de opdrachtgever is om binnen zes maanden een proof-of-concept te implementeren voor de EVAjig, waarbij de huidige microcontroller gebaseerde implementatie wordt vergeleken met een nieuwe FPGA-gebaseerde oplossing. Deze proof-of-concept moet aantonen dat de FPGA-oplossing verbetering biedt in flexibiliteit en communicatiecapaciteit vergeleken met de huidige implementatie.

2.4 Onderzoek

De concrete onderzoeksvraag is *“Welke communicatiemiddelen tussen FPGA en computer zijn beschikbaar, wat zijn voor- en nadelen ten opzichte van snelheid, betrouwbaarheid en kosten en hoe dit*

communicatiemiddel gebruikt zou moeten worden?”

De huidige implementatie op de microcontroller, de software op de microcontroller, hoort bijna onveranderd op de FPGA-implementatie te kunnen draaien. Wel moet de software met de FPGA kunnen praten, dit vereist een vast communicatieprotocol tussen de software en de firmware architectuur. Er zijn vele communicatieprotocollen beschikbaar met elk hun eigen use-case. In dit verslag zou onderzocht worden, welk protocol het beste geschikt is voor communicatie tussen de firmware en de software ten opzichte van snelheid, flexibiliteit en kosten.

2.5 Stakeholders

Naam	Organisatie	Rol
Jan Stegenga	EVAbits	Stagebegeleiding
Erik Krallen	EVAbits	Stagebegeleiding
Kobus Bijker	Hanzehogeschool	Stagebegeleiding / Beoordelaar
Nienke van der Spek	Hanzehogeschool	Beoordelaar Communicatie

2.6 Beroepsproducten

Concrete beroepsproducten die van EVAbits werden gevraagd zijn:

2.6.1 Demonstratie Zephyr OS op Efinix' Sapphire Core

De opdrachtgever gebruikt voor de huidige implementatie op de microcontroller Zephyr OS. Zephyr OS is een Real-Time Operating System (RTOS), een besturingssysteem dat is ontworpen om applicaties te draaien waarbij het belangrijk is dat ze reageren op gebeurtenissen binnen een strikte tijdslimiet. Dit type besturingssysteem wordt vaak gebruikt in embedded systemen, waar tijdige en voorspelbare reacties op externe signalen cruciaal zijn. Zephyr OS is een open-source real-time operating system dat is ontwikkeld voor het draaien op embedded apparaten. Het is ontworpen om lichtgewicht, schaalbaar en veilig te zijn, waardoor het geschikt is voor een breed scala aan toepassingen, van eenvoudige sensoren tot complexe industriële besturingssystemen. Efinix is een fabrikant van FPGA's inclusief een reeks handige core-modules. De Sapphire Core is een processor logicamodule geschreven en geoptimaliseerd voor hun FPGA, daarmee is het mogelijk minder primitieve operatie in Zephyr OS te draaien. Deze core gebruikt de RISC-V architectuur. Voortaan wordt Sapphire core ook RISC-V core genoemd om verwarring met Zephyr OS te voorkomen.

2.6.2 Demonstratie communicatie tussen de FPGA en een computer via EVAbus (RISC-V Core)

EVAbits heeft hun eigen datastructuur bedacht voor het effectieve aansturen van de EVAjig. Dit is geïmplementeerd met behulp van Google's Protobuffers (Protobuf), Protobuf is een efficiënt en taalneutraal data-serialisatieformaat ontwikkeld door Google, het wordt gebruikt voor het uitwisselen van gestructureerde informatie tussen verschillende programma's en systemen. Het maakt gebruik van een eenvoudig en uitgebreid berichtformaat om data in een compacte, snelle en makkelijk te lezen manier te coderen en decoderen.

2.6.3 Demonstratie communicatie tussen de FPGA en RISC-V Core (VHDL)

Als volgt moet er gecommuniceerd worden tussen de primitieve FPGA Core en de RISC-V Core om data zoals configuratie uit te wisselen. Dit moet van beide kanten ontwikkeld worden.

2.6.4 Implementatie van een driver in Zephyr OS voor het aansturen van de FPGA

Zephyr OS is modulair ontwikkeld om een brede schaal aan microcontrollers te ondersteunen. Hiervoor zijn er verschillende drivers beschikbaar. De FPGA hoort zelf geconfigureerd te worden, hiervoor is geen driver beschikbaar en deze dient zelf ontwikkeld te worden. EVAbits wil graag dat er een driver voor Zephyr OS wordt geschreven om makkelijk de FPGA aan te sturen.

2.6.5 Onderzoeksrapport communicatieprotocollen

De communicatie tussen de FPGA en de RISC-V Core kan op verschillende manieren geïmplementeerd worden. Hiervoor zijn vele protocollen beschikbaar en toepasbaar. De opdrachtgever wil graag een vergelijking hebben tussen deze protocollen om beter te kunnen kiezen welke voor de EVAjig het beste is.

3 Hoofdstuk Onderzoek

3.1 Protocollen

In dit hoofdstuk worden verschillende protocollen toegelicht. Een protocol is een afspraak tussen apparaten hoe gecommuniceerd wordt. Een voorbeeld is een spreektaal, er is afgesproken hoe bepaalde dingen worden genoemd en dat er regels zijn in de taal. Als twee personen er niet over eens zijn, welke taal wordt gesproken of hoe zij de taal spreken, kunnen zij niet met elkaar communiceren. In communicatieprotocollen is gedefinieerd welke signalen (draadjes/lijnen) er zijn en hoe deze moeten worden aangestuurd; wanneer gaat stroom over welk signaal, hoeveel tijd zit er tussen etc.

Protocollen hebben meestal een clock-sigitaal, die de snelheid van communicatie bepaald. Bijvoorbeeld wanneer het signaal stijgend is (oftewel van een lage waarde naar een hoge waarde gaat) worden de data signalen op een bepaalde manier uitgelezen. Als het signaal van een hoge waarde naar een lage waarde gaat, is het een vallend signaal.

In de komende beschrijvingen worden VDD en GND niet genoemd of meegeteld. VDD is de stroomvoering van desbetreffende chip en GND is de aarding van de chip.

Vele protocollen werken naar het bus- of master-slave principe. Bij een bus zijn meerdere apparaten, zoals chips en processoren, parallel aan de signalen aangesloten. Er is één master is gedefinieerd, die controle heeft over het gebeuren op de bus en één of meerdere slaves, die op aanvraag van de master kunnen lezen van de bus of schrijven op de bus. Een aangesloten chip zou voortaan *node* genoemd worden.

3.1.1 Inter-Integrated-Circuit Protocol

Inter-Integrated-Circuit (I²C) is één van de meest makkelijke protocollen. I²C is ontwikkeld in 1982 door Philips Semiconductor.^{1(Chapter 1.1)} I²C is een bus-systeem, die minstens één master en één slave eist. I²C bestaat uit twee signalen, een voor de data die moet worden verstuurd en een voor de clock.^{1(Chapter 2.1)}

Elke node heeft een adres waarop het wordt aangestuurd, er is alleen communicatie mogelijk tussen de master en een slave, het is niet mogelijk dat slaves onderling kunnen communiceren. Als standaard is 100kbit/s gedefinieerd als snelheid van communicatie, maar I²C is compatibel tot een snelheid van 3,4Mbit/s genoemd *high speed mode*.^{1(Chapter 1.2)}

Voordelen I²C heeft een eenvoudige bedrading en is makkelijker en goedkoper te implementeren. Ook is hot-swapping met I²C mogelijk, dat betekent dat slaves verwijderd of toegevoegd kunnen worden aan de bus zonder dat de bus uitgeschakeld of herstart moet worden.²

Nadelen Doordat er maar één datalijn is, is I²C half-duplex. Half-duplex betekent, dat er of geschreven kan worden door de master of door een slave, er kan niet tegelijk gecommuniceerd worden.^{1(Chapter 2.1)} Ook is I²C relatief langzaam in vergelijking met bv. SPI.^{1(Chapter 1.2),3}

3.1.2 Serial Peripheral Interface

Het Serial Peripheral Interface (SPI) protocol is ook een bus-protocol. Het is complexer en flexibeler ten opzichte van I²C, het is een 3-sigitaal protocol. Er is een clock, geleverd door de master en een MOSI- en MISO-sigitaal. De MOSI-lijn (Master Out, Slave In) wordt gebruikt om data naar een slave te sturen, de slave ontvangt data. De MISO-lijn (Master In, Slave Out) wordt gebruikt om data vanuit een slave naar de master te sturen. Er is een optionele Chip-Select lijn bij het protocol om eenvoudig een node te selecteren.^{4(Chapter 2.2)}

3.1.3 Universal Asynchronous Receiver/Transmitter

Universal Asynchronous Receiver/Transmitter (UART) is een asynchrone protocol, dus er kan tegelijk geschreven en gelezen worden. In tegenstelling tot de bovenstaande protocollen is UART geen bus-protocol en werkt alleen tussen twee nodes. UART heeft twee datalijnen: RX en TX, RX staat voor *Receive Text* en TX staat voor *Transmit Text* en heeft geen clock. Door beide nodes wordt een clock-snelheid gedefinieerd, dat wordt *Baud* genoemd. Er wordt simpelweg de data geschreven met de gedefinieerde snelheid inclusief een start- en stop-bit en optioneel een parity-bit voor fout-detectie.

Voordelen Zoals bij I²C heeft UART een eenvoudige bedrading, dus is het goedkoop te implementeren. Ook is UART technisch makkelijk te implementeren.

Nadelen UART is niet geschikt voor meerdere nodes. Als er gecommuniceerd moet worden naar meerdere nodes, moeten er onderling UART-signals geplaatst worden.

3.2 Toepassing op EVAjig

Voor de geplande toepassing op de EVAjig is het gekozen protocol de bottleneck. De computer stuurt data naar de FPGA via één protocol om zo de data via verschillende protocollen naar de PCB's door te sluizen.

3.3 Conclusie

3.4 Aanbevelingen

4 Hoofdstuk Projectaanpak en Risico's

4.1 Planning

4.1.1 Requirements

4.2 Risico's

5 Hoofdstuk HBO-I competenties

Tijdens de stageperiode zijn er plannings gemaakt om de werkzaamheden gestructureerd aan te pakken.

5.1 Onderzoeken

Er is onderzoek gedaan naar welk communicatieprotocol het meest geschikt is voor de toepassing op de FPGA. Ook is onderzocht hoe het huidige systeem geïmplementeerd kan worden op de FPGA.

5.2 Projectmatig Werken

In de stage werd gebruik gemaakt van verschillende projectstructuren. Er wordt op een systematische wijze gewerkt aan het vraagstuk.

5.3 Analyseren

De huidige implementatie van het systeem is geanalyseerd om eventuele verbeterpunten te identificeren. Deze analyse maakte het mogelijk om gerichte aanpassingen te doen en het systeem efficiënter te maken.

5.4 Ontwerpen

Op basis van de analyse zijn nieuwe implementaties ontworpen voor de FPGA. Deze ontwerpen omvatten verbeterde logische schema's en configuraties die de prestaties van het systeem optimaliseren.

5.5 Realiseren

De nieuwe ontwerpen zijn getest op de FPGA om de functionaliteit te verifiëren.

5.6 Schriftelijke Vaardigheden

Gedurende de stage zijn documentaties en dit verslag geschreven. Deze documenten zijn handig voor de opdrachtgever en de Hanzehogeschool.

5.7 Adviseren

Op basis van de nieuwe implementatie zijn aanbevelingen gedaan voor verdere optimalisaties en verbeterpunten. Deze adviezen helpen het team om de prestaties en efficiëntie van de projecten te verbeteren.

(terugkoppeling met mijn activiteiten volgt nog)

Begrippenlijst

Begrip	Afkorting	Beschrijving
...

Literatuurlijst

1. Wu, J. (2022). *A basic guide to I²C*. <https://www.ti.com/lit/an/sbaa565/sbaa565.pdf>; Texas Instruments.
2. Nguyen, D. (Bobby). (2020). *I2C solutions for hot swap applications*. <https://www.ti.com/lit/an/scpa058a/scpa058a.pdf>; Texas Instruments.
3. *What is the maximum SPI clock speed?* (2021). <https://community.silabs.com/s/article/what-is-the-maximum-spi-clock-speed-x>; Silicon Labs.
4. *KeyStone architecture serial peripheral interface (SPI) user guide*. (2012). <https://www.ti.com/lit/ug/sprugp2a/sprugp2a.pdf>; Texas Instruments.