RTOS-analyse voor Themaopdracht Devices



|  |  |  |  |
| --- | --- | --- | --- |
| **Projectleden** |  | René de Kluis:  Brent Cranko:  Marten Butter:  Arco Gelderblom: | 1661627  1674186  1691576  1679531 |
| **Docent** |  | Marten Wensink  Jan Zuurbier |  |
| **Klas** |  | TI-ICT-V2B |  |
| **versie** |  | 1.0 |  |
| **Datum** |  | 25-10-2016 |  |

Inhoudsopgave

[Inleiding 3](#_Toc466536118)

[Onderzoeksmethoden 4](#_Toc466536119)

[Arduino RTOS-functionaliteiten 4](#_Toc466536120)

[Kandidaat RTOS-en 4](#_Toc466536121)

[Compatibiliteit kandidaat RTOS-en 4](#_Toc466536122)

[Niet ondersteunde functionaliteiten 4](#_Toc466536123)

[Implementatie niet ondersteunde functionaliteiten 5](#_Toc466536124)

[Resultaten 5](#_Toc466536125)

[Arduino RTOS-functionaliteiten 5](#_Toc466536126)

[Kandidaat RTOS-en 6](#_Toc466536127)

[Compatibiliteit kandidaat RTOS-en 6](#_Toc466536128)

[Niet ondersteunde functionaliteiten 7](#_Toc466536129)

[Implementatie niet ondersteunde functionaliteiten 7](#_Toc466536130)

[Conclusie 7](#_Toc466536131)

[Discussie 8](#_Toc466536132)

[Bronvermelding 9](#_Toc466536133)

# Inleiding

Voor de studieHBO-ICT, afstudeerrichting Technische Informatica,is er naar aanleiding van een projectopdracht een lasergamesysteemgemaakt wat uitgevoerd wordt op een Arduino Due. Dit lasergamesysteem maakt gebruik van een Real Time Operating System (RTOS) genaamd Arduino RTOS.

Naast de Arduino Due zijn er echter veel alternatieve embedded platforms(Systems on a Chip, SOC's). Ontwikkeling van softwareproducten is hierdoor een lastige opgave, want de vraag blijft of de software ook nog steeds werkt op andere hardware. Het RTOS is de basis waarop de rest van de software wordt gemaakt en niet ieder RTOS werkt op alle platformen en microcontrollers.

In dit onderzoeksrapport wordt gekeken naar meerdere alternatieve RTOS-en die verenigbaar zijn met het project. Hiervoor is een lijst opgesteld met functionele en non-functionele eisen waar een RTOS aan moet voldoen. Door deze opgestelde eisen zijn niet alle RTOS-en geschikt.

Daarnaast moet er duidelijk zijn welke functionaliteiten wel of niet aanwezig zijn in de RTOS-en die voldoen aan de non-functionele eisen. De kans is klein dat een RTOS specifiek alle functionaliteiten heeft die het Arduino RTOS ook biedt. Voor ontbrekende functionaliteiten moet worden bekeken hoe deze in het betreffende RTOS geïmplementeerd kunnen worden.

Om de verschillende eisen goed te kunnen onderzoeken heeft de opdrachtgever de volgende hoofdvraag opgesteld:

‘Met behulp van welk open source realtime operating system kunnen tasks en de concurrency mechanismen, pool, channel, flag(group), clock timer en mutex, zoals aangeboden door het Arduino RTOS, met zo weinig mogelijk overhead worden gerealiseerd?’

Om deze vraag zo gestructureerd mogelijk te beantwoorden zijn er de volgende deelvragen gegeven:

1. Wat zijn de kenmerkende eigenschappen van tasks en de concurrency mechanismen van het Arduino RTOS?
2. Welke open source RTOS-en zijn beschikbaar?
3. Welk van de beschikbare RTOS-en biedt de meeste van de concurrency mechanismen van het Arduino RTOS aan zonder enige modificatie en biedt dezelfde functionaliteit om taken de realiseren
4. Welke mechanismen van het Arduino RTOS worden niet ondersteund door de beschikbare RTOS-en?
5. Hoe kunnen de mechanismen ban het Arduino RTOS die niet direct worden ondersteund door de beschikbare RTOS-en worden gerealiseerd met behulp van deze RTOS-en?

Op basis van deze deelvragen zal in het hoofdstuk **“Onderzoeksmethoden”** het onderzoeksplan wordenopgesteld en in het hoofdstuk **“Resultaten”** zullen de deelvragen beantwoord worden. Tenslottezalin het hoofdstuk “**Conclusie”** de hoofdvraag worden beantwoord.

Het onderzoek zal hoofdzakelijk bestaan uit een literatuurstudie die wordt uitgevoerd op het internet, op de documentatie van het Arduino RTOS en op de documentatie van de RTOS-en die in dit rapport worden onderzocht.

# Onderzoeksmethoden

Er zijn vijf verschillende eisen waar rekening mee gehouden moet worden bij het vinden van een geschikt RTOS. Deze eisen zijn geformuleerd in de vorm van de vijf deelvragen die in de inleiding zijn genoemd. In dit hoofdstuk wordt beschreven hoe deze vragen worden beantwoord.

## Arduino RTOS-functionaliteiten

Als er een alternatief wordt gebruikt voor het huidige RTOS moet deze ondersteuning bieden voor de beschikbare functionaliteiten. De vraag om deze eis te onderzoeken is ***"Wat zijn de kenmerkende eigenschappen van tasks en concurrency mechanismen van het Arduino RTOS".***

Er is een beeld van welke beschikbare functionaliteiten dit zijn. Voor dit onderzoek is er echter nog een beter overzicht nodig. Daarom zal de documentatie van het Arduino RTOS bestudeerd worden en op basis hiervan zal er een functionaliteiten lijst worden opgesteld.

## Kandidaat RTOS-en

Er zijn veel verschillende RTOS-en, maar een duidelijk overzicht van welke beschikbaar zijn, is er niet. De deelvraag ***"Welke open-source RTOS-en zijn beschikbaar"*** wordt beantwoord door een lijst van RTOS-en op te stellen. Om de lijst van beschikbare systemen behapbaar**?!?** te maken is er op basis van het huidige systeem een lijst van 4 non-functionele eisen opgesteld:

* Beschikbaar op het ARM-platform;
* Ondersteuning voor C en C++;
* Binnen het afgelopen jaar geüpdatet;
* Beschikbare documentatie met beschrijving over geboden functionaliteiten.

Door middel van dit literatuuronderzoek zal er gezocht worden naar de verschillende RTOS-en die voldoen aan deze eisen.

## Compatibiliteit kandidaat RTOS-en

Het moment dat er een ingekorte lijst is gemaakt van alle RTOS-en die voldoen aan de non functionele eisen kunnen de functionaliteiten van deze systemen vergeleken worden. Om de vraag, ***“Welk van de beschikbare RTOS-en biedt de meeste van de concurrency mechanismen van het Arduino RTOS aan zonder enige modificatie en biedt dezelfde functionaliteit om taken de realiseren”****,*voor elk van de in deelvraag 2 gevonden RTOS-en te beantwoorden**,** wordt bekekenof ze voldoen aan de lijst van functionaliteiten die is opgesteld voor deelvraag 1. De resultaten worden samengevoegd in een tabel zodat er een duidelijk overzicht van de compatibiliteit van de verschillende RTOS-en is. Tabel 1 is een voorbeeld hiervan.

|  |  |  |
| --- | --- | --- |
|  | RTOS 1 | RTOS 2 |
| Functionaliteit 1 | V | X |
| Functionaliteit 2 | X | V |

Tabel 1: Voorbeeld structurering resultaten deelvraag 3

## Niet ondersteunde functionaliteiten

Het is belangrijk duidelijk te hebben welke eigenschappen niet worden ondersteund door een systeem. Door middel van de gegevens die komen uit de resultaten van deelvraag 3 wordt de vraag, ***“Welke mechanismen van het Arduino RTOS worden niet ondersteund door de beschikbare RTOS-en?”***, beantwoord. Het tabel dat in het hoofdstuk “Compatibiliteit kandidaat RTOS-en” staat geeft een duidelijk overzicht van ondersteunde mechanismen en welke hiervan nog moeten worden geïmplementeerd.

## Implementatie niet ondersteunde functionaliteiten

Hoogstwaarschijnlijk is er niet een RTOS die alle benodigde functionaliteiten biedt. Sommige van de mechanismen zullen dus moeten worden geïmplementeerd met op maat gemaakte code. Voor elk van de mechanismen gevonden door middel van deelvraag 4 zal worden gekeken hoe deze moeten worden geprogrammeerd. Hiermee wordt de laatste deelvraag, “***Hoe kunnen de mechanismen van het Arduino RTOS die niet direct worden ondersteund door de beschikbare RTOS-en worden gerealiseerd met behulp van deze RTOS-en?”***,beantwoord.

De resultaten worden in het hoofdstuk “Resultaten” samengevat. Dit zal eruitzien als een tabel waarin de functionaliteiten staan opgesomd tegenover de beschikbare RTOS-en. Na deze resultaten bij elkaar gezet te hebben is het mogelijk om een geïnformeerde keuze te maken voor een systeem dat voldoet aan de gestelde eisen. Ook kan hier gezien worden welke functionaliteiten er aangeboden worden per RTOS. Eveneens zal er een beschrijving gegeven worden over hoe missende functionaliteiten toch kunnen worden geïmplementeerd aan de hand van wel bestaande functionaliteiten.

# Resultaten

## Arduino RTOS-functionaliteiten

De Arduino RTOS-documentatie is doorzocht op de verschillende functionaliteiten en hun eigenschappen. Daarnaast is ook het boek van Hassan Gomaa erop nageslagen *(Gomaa, 2016; Ooijen, 2016)*. **<- deze dingen omdraaien om conform tekst te doen?**

|  |  |
| --- | --- |
| Functionaliteit | Kenmerkende eigenschappen |
| Mutex | * Synchroon; * Zelf niet een datastructuur; * Beperkt de toegang tot gedeelde data tot een enkele taak. |
| Eventflag | * Asynchroon; * Zelf niet een datastructuur; * Signaleert een interne of externe gebeurtenis. |
| Channel | * Asynchroon; * First in, first out; * Data overdracht. |
| Timer | * Niet een inter-taak mechanisme; * Kan op gewacht worden; * Geeft op een gezette tijd een signaal af. |
| Pool | * Construct waar data in bewaard blijft gedurende een lopend programma; * Voorkomt het onnodig opzoeken van data. |

Tabel 2: Overzicht Arduino RTOS-functionaliteiten

## Kandidaat RTOS-en

Het opstellen van de lijst van mogelijke RTOS-en is gedaan aan de hand van de lijst van RTOS-en op Wikipedia. Deze lijst heeft een beknopte samenvatting van kenmerken die gebruikt konden worden in het selectieproces. Daarna is deze lijst vergeleken met de lijst die gevonden is op osrtos.com. Deze tweede lijst geeft namelijk ook aan wanneer de laatste update voor de RTOS-en is uitgerold.

De lijst is eerst gefilterd op de beschikbaarheid op het ARM-platform. In de lijst op Wikipedia staat ook of een systeem actief wordt onderhouden. Alleen de actief onderhouden systemen zijn verder beoordeeld. De Wikipedia**-**lijst had geen informatie over de ondersteuning van C en C++ of over beschikbare documentatie. De documentatie werd echter op de site osrtos.com wel gevonden. Door middel van het onderzoeken van de overgebleven systemen is er een lijst opgesteld van potentiële RTOS-en. In tabel 3 staat hiervan een overzicht.

|  |
| --- |
| RTOS |
| EcosPro(“eCosPro Developer’s Kit,” n.d.) |
| uKOS(*The uKOS system*, n.d.) |
| cocoOS(“NuttX Configuration Options,” n.d.) |
| Distortos(“Documentation • distortos,” n.d.) |
| FreeRTOS(“FreeRTOS,” n.d.) |
| FunkOS (“FunkOS developers Wiki,” n.d.) |
| scmRTOS(“ScmRTOS developers Wiki,” n.d.) |
| NuttX(“NuttX Configuration Options,” n.d.) |
| StratifyOS(“Stratify OS Docs,” n.d.) |
| mbed OS(“mbed-rtos API documentation,” n.d.) |

Tabel 3: Overzicht van kandidaat RTOS-en (“Comparison of real-time operating systems - Wikipedia,” n.d.)

## Compatibiliteit kandidaat RTOS-en

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **RTOS naam** | **mutex** | **event flag** | **channel** | **timer** | **pool** |
| EcosPro | V | V | X | V | X |
| uKOS | V | V | V | V | X |
| Coco OS | V | V | V | V | V |
| Distort OS | V | V | X | V | X |
| FreeRTOS | V | V | X | V | X |
| Funk OS | V | V | X | V | X |
| scmRTOS | V | V | V | V | V |
| NuttX | V | V | V | V | V |
| Stratify OS | V | V | X | V | X |
| MbedOS | V | V | X | V | V |

Tabel 4: Vergelijking kandidaat RTOS-en met Beschikbare functionaliteiten

Van de tien onderzochte systemen zijn er drie die alle functies hebben. Dit zijn: Nuttx, scmRT OS en Coco OS. (“cocoOS,” n.d., “NuttX Configuration Options,” n.d., “ScmRTOS developers Wiki,” n.d.)

## Niet ondersteunde functionaliteiten

Aan de hand van de resultaten in tabel vier zijn er twee functionaliteiten gevonden die missen in sommige van de RTOS-en.   
In tabel 5 is te zien welke functionaliteiten dit zijn en welke systemen ze missen.

|  |  |
| --- | --- |
| Functionaliteit | Niet ondersteunende RTOS-en |
| Channel | EcosPro  FreeRTOS  Stratify OS  Distort OS  Funk OS  MbedOS |
| Pool | EcosPro  Distort OS  Funk OS  uKOS  FreeRTOS  Stratify OS |

Tabel 5: Overzicht missende functionaliteiten

## Implementatie niet ondersteunde functionaliteiten

De missende functionaliteiten kunnen geïmplementeerd worden door andere functionaliteiten te gebruiken. In onderstaande tabel staat beschreven hoe deze gerealiseerd kunnen worden. (Gomaa, 2016; Wensink & van Ooijen, 2011).

|  |  |
| --- | --- |
| Channel | Een queue die een producer blokkeert als deze vol zit en een consumer blokkeert als de queue leeg is. |
| Pool | Via een memory allocatie is dit met kleine moeite te realiseren. |

# Conclusie

Er zijn op dit moment veel verschillende RTOS-en. Veel hiervan voldoen niet aan de criteria van het project: ‘Themaopdracht Devices’. Een duidelijk overzicht van systemen is er niet en documentatie is regelmatig afwezig. Na de lijsten op wikipedia en osrtos te hebben bekeken, zijn er drie systemen**,** NuttX, scmRT OS en Coco OS, uitgekomen die voldoen aan alle benodigde functionaliteiten. Aan de hand van dit rapport kunnen wij concluderen dat het antwoord op de gegeven hoofdvraag, ***‘Met behulp van welk open source realtime operating system kunnen tasks en de concurrency mechanismen, pool, channel, flag(group), clock timer en mutex, zoals aangeboden door het Arduino RTOS, met zo weinig mogelijk overhead worden gerealiseerd?’***, NuttX is. Deze keuze is gemaakt omdat NuttX significant veel meer mogelijkheden heeft dan scmRT OS of Coco OS.   
Echter, Zoals in het verslag te lezen is, is mogelijk om missende de functionaliteiten te implementeren in een van de andere acht systemen en hoeft daardoor de keuze voor een RTOS niet alleen op deze drie te vallen.

# Discussie

In dit rapport is voornamelijk gekeken naar functionaliteiten. Een van de vervolgstappen die bij dit onderzoek gedaan kan worden is het uitbreiden van zowel de non-functionele eisen als benodigde functionaliteiten.

Met het oog op non-functionele eisen is de performance heel erg significant. NuttX is een uitgebreid systeem en de vraag is of de omvang van het pakket niet een negatieve invloed heeft op de werking van het real time systeem (“NuttX Configuration Options,” n.d.).

Qua functionele eisen is het interessant om te kijken hoe bepaalde concurrency mechanismen worden geïmplementeerd. Het is mogelijk dat bepaalde implementatie niet ideaal aansluiten op de door deze projectgroep geschreven code, of op de HWLIB van de hogeschool Utrecht. Het kan dat een systeem met meerdere missende functionaliteiten beter aansluit door andere functionaliteiten.

Daarnaast is het de vraag of de functionaliteiten die het Arduino RTOS biedt, optimaal zijn voor het opzetten van een lasergamesysteem. Een van de functionaliteiten van het NuttX is een spinlock. Binnen het te bouwen systeem is een spinlock een interessante toevoeging omdat het de een positief effect op performance, als er een taak veel en kort wordt aangeroepen. Een vervolgonderzoek is mogelijk om te kijken wat de functionaliteiten zijn die nodig zijn voor een lasergame systeem(IEEE & The Open group, 2016; “NuttX Configuration Options,” n.d.).

# Bronvermelding

cocoOS. (n.d.). Retrieved November 8, 2016, from http://www.cocoos.net/

Comparison of real-time operating systems - Wikipedia. (n.d.). Retrieved November 8, 2016, from https://en.wikipedia.org/wiki/Comparison\_of\_real-time\_operating\_systems

Documentation • distortos. (n.d.). Retrieved November 8, 2016, from http://distortos.org/documentation/

eCosPro Developer’s Kit. (n.d.). Retrieved November 8, 2016, from http://www.ecoscentric.com/ecos/ecospro.shtml

FreeRTOS. (n.d.). https://doi.org/2016-11-08

FunkOS developers Wiki. (n.d.). Retrieved November 8, 2016, from https://sourceforge.net/p/funkos/wiki/Home/

Gomaa, H. (2016). *Real-Time Software Design for Embedded Systems - Hassan Gomaa - Google Boeken* (1st ed.). New York: Cambridge University Press. Retrieved from https://books.google.nl/books?hl=nl&lr=&id=mZyKCwAAQBAJ&oi=fnd&pg=PR15&dq=hassan+gomaa&ots=YKcmHAMGEE&sig=FzbHb75cYrDfu\_U4DiMW3nrC19U#v=onepage&q=hassan gomaa&f=false

IEEE, & The Open group. (2016). Posix.1-2008.

mbed-rtos API documentation. (n.d.). Retrieved November 8, 2016, from https://developer.mbed.org/users/mbed\_official/code/mbed-rtos/docs/tip/

NuttX Configuration Options. (n.d.). Retrieved November 8, 2016, from http://nuttx.org/Documentation/NuttXConfigVariables.html

Ooijen, V. (2016). *DoxyGen documentation Arduino RTOS*.

ScmRTOS developers Wiki. (n.d.). Retrieved from http://scmrtos.sourceforge.net/ScmRTOS

Stratify OS Docs. (n.d.). Retrieved November 8, 2016, from https://stratifylabs.co/StratifyOS/html/group\_\_\_p\_o\_s\_i\_x.html

*The uKOS system*. (n.d.). Retrieved from http://www.ukos.ch/downloads/the-ukos-system.pdf

Wensink, M., & van Ooijen, W. (2011). Realtime System Programming.