RTOS systemen

Inhoudsopgave

[Inleiding 2](#_Toc465429654)

[1. Wat zijn de kenmerkende eigenschappen van tasks en de concurrency mechanismen van het Arduino RTOS? 3](#_Toc465429655)

[2. Welke open source RTOS-en zijn beschikbaar? 4](#_Toc465429656)

[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 te realiseren? 5](#_Toc465429657)

[4. Welke mechanismen van het Arduino RTOS worden niet ondersteund door de beschikbare RTOS-en? 6](#_Toc465429658)

[5. Hoe kunnen de mechanismen van het Arduino RTOS die niet direct worden ondersteund door de beschikbare RTOS-en worden gerealiseerd m.b.v. van deze RTOS-en? 7](#_Toc465429659)

[Conclusie 8](#_Toc465429660)

# Inleiding

# Wat zijn de kenmerkende eigenschappen van tasks en de concurrency mechanismen van het Arduino RTOS?

* Pool
* Mutex
* Channel
* Mailbox
* Timer/clock
* Flag

# Welke open source RTOS-en zijn beschikbaar?

Free-RTOS (<http://freertos.org>)

Distortos (<http://distortos.org>)

# 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 te realiseren?

Free-RTOS beschikt over queues(channels), event-bits(flags), mutexes, timers, clocks, semaphores.

Distortos beschikt over queues(channels), flags, mutexes, timers, clocks, semaphores.

# Welke mechanismen van het Arduino RTOS worden niet ondersteund door de beschikbare RTOS-en?

Free-RTOS en Distortos missen beide een pool en een mailbox.

# Hoe kunnen de mechanismen van het Arduino RTOS die niet direct worden ondersteund door de beschikbare RTOS-en worden gerealiseerd m.b.v. van deze RTOS-en?

Een mailbox kan worden nagebootst door middel van channel en een binary semaphore. Een mailbox houdt in dat er data in kan worden gezet en kan worden uitgelezen. Een binary semaphore kan worden gezien als een mutex met een counter. Deze counter wordt op een nummer gezet, de task die iets wilt doen, moet de semaphore daarom vragen. Elke keer dat er iets aan de semaphore wordt gevraagd, wordt deze counter met 1 verlaagd. Dit kan als mailbox worden gebruikt door de channel te “locken” wanneer erin is geschreven. De lezer zal dan eerst meerdere keren moeten proberen het te kunnen lezen. Op de manier wordt het systeem, net als bij een mailbox, geblokt voor een bepaalde periode.

Een pool kan worden geïmiteerd door middel van een globale variabele en een mutex. Een pool kan worden gezien als een variabele die kan gelezen en overschreven kan worden door meerdere tasks, maar dit niet tegelijk. Door middel van een globale variabele en een mutex kan dit ook. Wanneer een task de variabele wilt schrijven, locked hij de mutex, wanneer hij klaar is, geeft hij hem vrij. Hetzelfde geldt voor lezen. Dit is zo zodat de variabel niet kan worden veranderd wanneer hij wordt gelezen.

# Conclusie

Na 8 verschillende RTOS-en te hebben onderzocht, zijn wij tot conclusie gekomen dat er verschillende RTOS-en ongeschikt zijn. Ook hebben wij een aantal RTOS-en gekozen welke wel de moeite waard zijn. Wij hebben ook de Arduino RTOS betrokken in deze conclusie.

Als eerste hebben we Free-RTOS, zoals alle anderen is dit RTOS open-source en beschikbaar voor ARM-processoren. Het opvallende van dit RTOS is