

**VOORBLAD SCHRIFTELIJKE TOETSEN**

<b>OPLEIDING</b>	<b>: ELEKTROTECHNIEK</b>
<b>TOETSCODE</b>	<b>: RTSYST-co1</b>
<b>GROEP</b>	<b>: ECV</b>
<b>TOETSDATUM</b>	<b>: 26 JUNI 2012</b>
<b>TIJD</b>	<b>: 15.00 – 16.30 uur</b>
<b>AANTAL PAGINA'S (incl. voorblad)</b>	<b>: 4</b>
<b>DEZE TOETS BESTAAT UIT</b>	<b>: 4 open vragen</b>
<b>GEBRUIK HULPMIDDELEN</b>	<b>: JA</b>
<b>TOEGESTANE HULPMIDDELEN</b>	<b>: Tijdens dit tentamen mogen alle boeken, dictaten, aantekeningen enz. worden gebruikt.</b>
<b>OVERIGE OPMERKINGEN</b>	<b>: Toetsopgaven na afloop inleveren!</b>
<b>OPSTELLER VAN DEZE TOETS</b>	<b>: Harry Broeders</b>
<b>TWEDE LEZER VAN DEZE TOETS</b>	<b>: Jesse op den Brouw</b>

**BELANGRIJKSTE PUNTEN UIT ARTIKEL 12 VAN DE ONDERWIJS- EN EXAMENREGELING:**

- je dient je via Osiris ingeschreven te hebben voor deze toets
- schrijf je naam, je studentnummer, de toetscode en de naam van de docent meteen op het tentamenpapier
- leg je identiteitsbewijs op de hoek van de tafel
- zet alle elektronische communicatiemiddelen (mobiele telefoon, PDA, etc.) uit en stop deze in je tas; deze mogen niet als calculator of klok worden gebruikt
- je mag het lokaal het eerste halfuur niet verlaten
- volg de instructies op het toetsvoorblad
- steek je hand op als je een vraag hebt

---

KLAS(SEN) : ECV	BLAD : 1 van 3 BLADEN
TOETS : real-time systemen (TOETS01)	DOCENT : Harry Broeders
CODE : <b>RTSYST-co1</b>	DATUM : 26 juni 2012
KWARTAAL : 4	TYPE : hertentamen
	TIJD : 15:00 – 16:30

---

Tijdens dit tentamen mogen **alle** boeken, dictaten, aantekeningen enz. worden gebruikt.

---

Bij elke opgave staat tussen haakjes het maximale aantal te behalen punten vermeld.  
Eindcijfer = (aantal behaalde punten + 10) / 10.

- 1.** In een multi-threaded POSIX C programma maken verschillende threads gebruik van een ADC (Analog Digital Converter). Om deze ADC te kunnen gebruiken zijn verschillende functies beschikbaar (in een library). De ADC heeft meerdere kanalen. De code om de analoge spanning op kanaal 3 in te lezen in de variabele `res` met behulp van de beschikbare functies uit de library is als volgt:

```
AdcSelectChannel(3);  
res = AdcGetSample();
```

Deze code werkt niet goed in een multi-threaded programma omdat het selecteren van het kanaal en het uitvoeren van de conversie onderbroken kan worden door een andere thread die een ander kanaal kan selecteren.

- A. (15) Geef aan hoe de code aangepast moet worden zodat het selecteren van het kanaal en het uitvoeren van de conversie niet onderbroken kan worden door een andere thread.
- B. (10) Pas het programma uit opgave A zodanig aan dat nooit langer dan 100 ms op het beschikbaar komen van de ADC converter wordt gewacht. Maak gebruik van een POSIX system call met time-out. Als de ADC niet binnen 100 ms beschikbaar komt moet de variabele `res` de waarde `-1` krijgen.

- 2.** POSIX message queues worden meestal gebruikt om **asynchrone** messages te versturen. Als we met behulp van POSIX message queues **synchrone** messages willen versturen dan is dat ook mogelijk.

- A. (5) Geef aan welke **asynchrone** berichten verstuurd moeten worden om 1 **synchrone** message te versturen van proces A naar proces B. Verklaar je antwoord!
- B. (5) Hoeveel message queues zijn hierbij nodig. Verklaar je antwoord!
- C. (10) Als zowel proces X als proces Y **synchrone** messages moeten kunnen versturen naar proces Z geef dan aan hoe dit met behulp van POSIX message queues geïmplementeerd kan worden. Geef een beschrijving van het te gebruiken protocol en leg duidelijk uit hoeveel message queues **minimaal** nodig zijn.

KLAS(SEN) : ECV	BLAD : 2 van 3 BLADEN
TOETS : real-time systemen (TOETS01)	DOCENT : Harry Broeders
CODE : <b>RTSYST-co1</b>	DATUM : 26 juni 2012
KWARTAAL: 4	TYPE : hertentamen TIJD : 15:00 – 16:30

Tijdens dit tentamen mogen **alle** boeken, dictaten, aantekeningen enz. worden gebruikt.

3. De besturing van een robot wordt geïmplementeerd als een multi-threaded C++ programma met behulp van de boost library. De robot houdt zijn positie bij aan de hand van een X en Y coördinaat. Deze coördinaten worden opgeslagen in de globale variabelen `x` en `y` en beveiligd met behulp van de mutex `m`:

```
int x = 0, y = 0;
boost::mutex m;
```

De code waarin `x` en `y` een nieuwe waarde krijgen ziet er als volgt uit:

```
{
    boost::lock_guard<boost::mutex> guard(m);
    x = /* code niet gegeven */;
    y = /* code niet gegeven */;
}
```

Om veiligheidsredenen moet de robot meteen stopgezet worden als de X of Y coördinaat kleiner dan 0 of groter dan 100 wordt. Er wordt een aparte thread gemaakt die de robot **zo snel mogelijk** moet stoppen (door het aanroepen van de functie `stop`) als de X of Y coördinaat kleiner dan 0 of groter dan 100 wordt.

Een onervaren programmeur heeft deze thread als volgt geprogrammeerd:

```
void alarm() {
    while (1) {
        bool ok;
        do {
            boost::lock_guard<boost::mutex> guard(m);
            ok = x >= 0 && x <= 100 && y >= 0 && y <= 100;
        }
        while (!ok);
        stop();
    }
}

// in main:
boost::thread t(alarm);
```

- A. (5) Als de robot zich in een veilige positie bevindt dan wordt de processor waarop thread `t` draait altijd volledig belast. **Verklaar hoe dit komt?**
- B. (20) Verbeter de code van de onervaren programmeur en zorg er daarbij (met behulp van een conditionele variabele) voor dat de processor slechts **minimaal** belast wordt door de thread `t`. Geef ook aan hoe de code waarin `x` en `y` een nieuwe waarde krijgen moet worden aangepast.

**Zie volgende blad →**

---

KLAS(SEN) : ECV  
TOETS : real-time systemen (TOETS01)  
CODE : **RTSYST-co1**  
KWARTAAL : 4

BLAD : 3 van 3 BLADEN  
DOCENT : Harry Broeders  
DATUM : 26 juni 2012  
TIJD : 15:00 – 16:30

---

Tijdens dit tentamen mogen **alle** boeken, dictaten, aantekeningen enz. worden gebruikt.

---

**4.** In een multitasking systeem zijn de volgende taken gedefinieerd:

<b>taak</b>	<b><i>T</i></b>	<b><i>D</i></b>	<b><i>C</i></b>
A	8	5	4
B	20	14	5
C	18	16	2

$T$  = periodetijd.  $D$  = deadline.  $C$  = calculation time.

- A. (5) Welke prioriteit moet aan deze taken worden toegekend om een optimale scheduling te garanderen? Er zijn 3 verschillende prioriteiten beschikbaar: 1, 2 en 3, waarbij 3 de hoogste en 1 de laagste prioriteit is.
- B. (15) Bereken de response tijden van de 3 taken bij de in antwoord op opgave A gegeven prioriteiten.