

ECS les 7 **Antwoorden/Solutions**



Docent : [ir drs E.J Boks](#)

Opdrachten om zelfstandig uit te voeren tijdens het derde lesuur

Assignments for independent execution during the third course hour

Voer de onderstaande opdrachten zelfstandig uit tijdens het derde lesuur ECS. De opdrachten zijn een test om in te schatten hoe de student de theoriestof van het vak ECS beheerst.

De uitwerkingen en antwoorden worden gepresenteerd in de volgende lesweek.

Complete the assignments below independently during the third ECS class. The assignments are a test to estimate how the student has mastered the theory of ECS.

Elaborations and answers are presented in the following lesson week.

1. Beschrijf in eigen woorden de volgende RGT begrippen / *Describe in your own wordings the following CS concepts :*
 - DAS
 - Controller
 - Actuator

antwoord/solution : Zie/See §9.1

2. Wat is het verschil tussen een sensor en een "transducer" ? / *What is the difference between a sensor and a "transducer"?*

antwoord/solution : Zie/See §9.2.1 . Sensor = meetinstrument, Transducer = omzetter van het sensorsignaal naar een formaat dat de controller kan gebruiken / *Sensor = measuring instrument, Transducer = converter of the sensor signal to a format that the controller can use.*

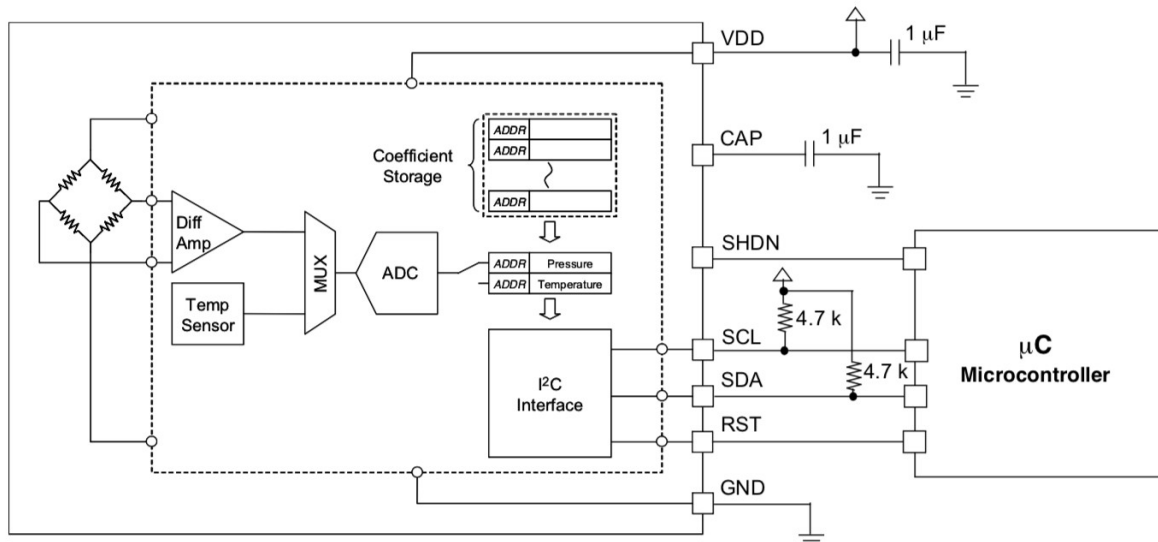
3. Wat is een SCADA systeem / *What is a SCADA system?*

antwoord/solution : Zie/See §9.2.3

4. Geef aan hoe het [MPL3115A2](#) IC van Freescale werkt / *State how the MPL3115A2 IC from Freescale works.. Legt uit/ Explain :*

- Hoe de sensor werkt / *How the sensor functions.*
- Wat de transducer is / *What the transducer is.*
- Wat de operationele grenzen van het IC zijn / *What the operational limits of the IC are.*

antwoord/solution : De MPL115A2 maakt gebruik van een MEMS-druksensor met een conditionerings-IC voor nauwkeurige drukmetingen van 50 tot 115 kPa. Geïntegreerde ADC biedt gedigitaliseerde temperatuur- en druksensoruitgangen via een I2C-poort. Kalibratiegegevens worden opgeslagen in interne ROM. Met behulp van onbewerkte sensoruitvoer voert de host-microcontroller een compensatiealgoritme uit om gecompenseerde absolute druk te produceren met een nauwkeurigheid van 1 kPa. / *The MPL115A2 employs a MEMS pressure sensor with a conditioning IC to provide accurate pressure measurement from 50 to 115 kPa. An integrated ADC provides digitized temperature and pressure sensor outputs via an I2C port. Calibration Data is stored in internal ROM. Utilizing raw sensor output, the host microcontroller executes a compensation algorithm to render Compensated Absolute Pressure with 1 kPa accuracy.*



- De sensor is een piezo-resistief element dat de druk oppikt / *The sensor is a piezo-resistive element that detects the pressure.*
- De transducer bestaat uit een Diff Opamp en een tweedegraads polynoom (datasheet, paragraaf over “Compensation”), en een I2C interface / *The transducer consists of a Diff Opamp and a second-degree polynomial (datasheet, section on “Compensation”), and an I2C interface.*
- De operationele grenzen zijn te vinden in het hoofdstuk “operating characteristics” / *The operational limits can be found in the chapter “operating characteristics” :*

Maximum Ratings

Voltage (with respect to GND unless otherwise noted)	
V _{DD}	-0.3 V to +5.5 V
SCLK, CS, D _{IN} , D _{OUT}	-0.3 V to V _{DD} +0.3 V
Operating Temperature Range	-40°C to +105°C
Storage Temperature Range	-40°C to +125°C
Overpressure	1000 kPa

Operating Characteristics

(V_{DD} = 2.375 V to 5.5 V, T_A = -40°C to +105°C, unless otherwise noted. Typical values are at V₊ = 3.3 V, T_A = +25°C.)

Ref	Parameters	Symbol	Conditions	Min	Typ	Max	Units
1	Operating Supply Voltage	V _{DD}		2.375	3.3	5.5	V
2	Supply Current	I _{DD}	Shutdown (SHDN = GND)	—	—	1	µA
			Standby	—	3.5	10	µA
			Average – at one measurement per second	—	5	6	µA
Pressure Sensor							
3	Range			50	—	115	kPa
4	Resolution			—	0.15	—	kPa
5	Accuracy		-20°C to 85°C	—	±1	—	kPa
6	Power Supply Rejection		Typical operating circuit at DC	—	0.1	—	kPa/V
			100 mV p-p 217 Hz square wave plus 100 mV pseudo random noise with 10 MHz bandwidth.	—	0.1	—	kPa
7	Conversion Time (Start Pressure Convert)	t _{CP}	Time between start convert command and data available in the Pressure register	—	0.6	0.7	ms
Temperature Sensor							
8	Range			-40	—	105	°C
9	Conversion Time (Start Temperature Convert)	t _{CT}	Time between start convert command and data available in the Temperature register	—	0.6	0.7	ms
10	Conversion Time (Start Both Convert)	t _{CB}	Time between start convert command and data available in the Pressure and Temperature registers	—	0.8	1	ms
11	Resolution		Temperature ADC is 472 counts at 25°C	—	-5.35	—	counts/°C
I²C I/O Stages: SCL, SDA							
12	SCL Clock Frequency	f _{SCL}		—	—	400	KHz
13	Low Level Input Voltage	V _{IL}		—	—	0.3V _{DD}	V
14	High Level Input Voltage	V _{IH}		0.7V _{DD}	—	—	V
I²C Outputs: SDA							
15	Data Setup Time	t _{SU}	Setup time from command receipt to ready to transmit	100	—	—	ns
I²C Addressing							
MPL115A2 uses 7-bit addressing, does not acknowledge the general call address 0000000. Slave address has been set to 0x60 or 1100000.							

5. Schrijf een software functie die een hysteresis implementeert zoals voorgesteld in afbeelding 9.8 uit het boek / Write a software function that implements a hysteresis as presented in Figure 9.8 from the book.

antwoord/solution : Dit is een voorbeeld (Nederlandstalige software)/ This is an example (Dutch language mplementation):

```
class AanUitController
{
public :
    /*! @brief De constructor voor de Aan/Uit controller.
    */
    AanUitController() = default;
    ~AanUitController() = default;

    enum class Eigenschap
    {
        Standaard,
        DodeZone,
        Hysteresis
    };

    void zetEigenschap(const Eigenschap, const Spanning param=0.0f);

    /*! @brief De student moet deze functie implementeren / The student must
    implement this function.
    * @param De setpoint waarde / the setpoint value.
    * @param de proces meetwaarde / the proces measurement value.
    * @return De controle waarde die het proces in gaat / the control value
    that will be put into the process.
    * @note Spanning = Voltage ***/
    Spanning aanUitFunctie(const Spanning, const Spanning);
private:

    /* Grenswaarden voor het systeem. Gebruik deze grenzen om de integratie
    te temmen. */
    static constexpr auto dacLowerLimit = 0.0f, dacUpperLimit=3.0f;
    static constexpr auto minIGrens = 10*dacUpperLimit;
    static constexpr auto iGrens = 10*dacUpperLimit;
    Eigenschap eigenschap = Eigenschap::Standaard;
    Spanning dzWaarde=0.0f;
    Spanning minDzWaarde=0.0f;

    enum class HysteresisPad
    {
        Hoog,
        Laag,
        Onbepaald
    } hysteresis = HysteresisPad::Onbepaald;
};
```



```

void AanUitController::zetEigenschap(const AanUitController::Eigenschap e,
                                   const Spanning param)
{
    eigenschap = e;
    if (Eigenschap::Standaard == eigenschap)
        dzWaarde=minDzWaarde=0.0f;
    else
    {
        dzWaarde = param;
        minDzWaarde= -1.0f*param;
    }
}

Spanning AanUitController::aanUitFunktie(const Spanning sp,
                                         const Spanning mw)
{
    const auto verschil = sp-mw;
    auto resultaat = 0.0f;
    switch(eigenschap)
    {
        case Eigenschap::Standaard:
        {
            resultaat = (verschil >= 0) ? dacUpperLimit : dacLowerLimit;
        }
        break;
        case Eigenschap::DodeZone:
        {
            resultaat = (verschil >= dzWaarde) ? dacUpperLimit :
                ((verschil < minDzWaarde) ? dacLowerLimit : 0.0f);
        }
        break;
        case Eigenschap::Hysterese:
        {
            switch (hysterese)
            {
                case HysteresePad::Laag:
                {
                    if (verschil >= dzWaarde)
                    {
                        hysterese = HysteresePad::Hoog;
                        resultaat = dacUpperLimit;
                    }
                    else
                        resultaat = dacLowerLimit;
                }
                break;
                case HysteresePad::Hoog:
                {
                    if (verschil < minDzWaarde)
                    {
                        hysterese = HysteresePad::Laag;
                        resultaat = dacLowerLimit;
                    }
                    else
                        resultaat = dacUpperLimit;
                }
                break;
                case HysteresePad::Onbepaald:
                {
                    if (verschil >= dzWaarde)
                    {
                        hysterese = HysteresePad::Hoog;
                        resultaat = dacUpperLimit;
                    }
                    else if (verschil < minDzWaarde)
                    {
                        hysterese = HysteresePad::Laag;
                        resultaat = dacLowerLimit;
                    }
                    else
                        resultaat = 0.0f;
                }
                break;
            }
        }
    }
    return (resultaat);
}

```

6. Een PID regelaar wordt ingezet in de industrie om talloze taken te vervullen. Geef aan wat de betekenis is van de termen P, I en D / *A PID controller is used in industry to fulfill countless tasks. Indicate the meaning of the terms P, I and D.*

antwoord/solution : Zie/See §9.3.3

7. Bij het gebruik van een PID regelaar hoeven niet alle acties ingezet te worden. Geef aan wanneer een ontwerper de P, I en D acties toepast / *When using a PID controller, not all actions need to be used. Indicate when a designer applies the P, I and D actions.*

antwoord/solution : Zie/See §9.3.3

Een te regelen proces heeft een overdrachtfunctie $H(s)$ /

A process to be controlled has a transfer function $H(s)$:
$$H(s) = \frac{1}{s^2 + 0,4s + 1}$$

8. Bereken of het proces ondergedempt, kritisch gedempt over overgedempt is / *Calculate whether the process is underdamped, critically damped or overdamped.*

antwoord/solution : $H(s)$ is gelijk aan / *is equal to :*
$$H(s) = \frac{1}{\left(\frac{s}{\omega_0}\right)^2 + 2\beta\frac{s}{\omega_0} + 1}$$
 Zie lesvragen

week vier / See assignments week 4.

ω_0 : de ongedempte hoekfrequentie / *the natural frequency.*

β is de dempingsconstante / *the damping constant.*

thus / therefore : $\left(\frac{1}{\omega_0}\right)^2 = 1 \rightarrow \omega_0 = 1 \text{ rad s}^{-1}$

Toepassing van het vorige resultaat leidt tot / *Application of the previous result :*

$$2\beta = 0,4 \rightarrow \beta = 0,2$$

De grens tussen kritische gedempt en ondergedempt is / *The boundary between critically damped and underdamped is* $\beta = \frac{1}{2}$. Dit betekent dat het systeem ondergedempt is / *this implies that the system is underdamped.*

De gekozen regelaar is een PID regelaar in een negatieve eenheidsterugkoppeling. Neem voor de waarde $K_p = 4$ / *The selected controller is a PID controller in a negative unity feedback loop. Apply the value $K_p = 4$.*

Een PID regelaar heeft de overdrachtfunctie / *A PID controller has the transfer function :*

$$H_{PID}(s) = K_p + K_d s + \frac{K_i}{s}$$

Het systeem heeft om deze reden de volgende teruggekoppelde overdrachtfunctie / The system therefore has the following feedback transfer function :

$$H_{tk}(s) = \frac{K_p + K_d s + \frac{K_i}{s}}{s^2 + 0,4s + 1} \cdot \frac{1}{1 + \frac{K_p + K_d s + \frac{K_i}{s}}{s^2 + 0,4s + 1}}$$

$$H_{tk}(s) = \frac{K_p + K_d s + \frac{K_i}{s}}{s^2 + (0,4 + K_d)s + (1 + K_p) + \frac{K_i}{s}} = \frac{K_p + K_d s + \frac{K_i}{s}}{\frac{s^2}{(1 + K_p)} + \frac{(0,4 + K_d)}{(1 + K_p)}s + \frac{K_i}{(1 + K_p)s} + 1}$$

9. Welke ongedempte hoekfrequentie had het systeem zonder regelaar, en wat is deze nu geworden / What undamped angular frequency did the system have without a controller, and what has it become now ?

antwoord/solution : In open loop, zie vraag 8. In gesloten lus met $K_p = 4$ / In open loop, see question 8. In Closed loop with $K_p = 4$: $(\frac{1}{\omega_0})^2 = \frac{1}{1 + K_p} \rightarrow \omega_0 = \sqrt{1 + K_p} = \sqrt{5} \text{ rad s}^{-1}$

10. Het proces moet kritisch gedempt worden afgeregeld. Welke waarde voor K_D moet hier voor gekozen worden / The process needs to be damped critically. Which value for K_D should be chosen for this?

antwoord/solution : kritische demping / critically damped $\Rightarrow \beta = \frac{1}{2}$

$$2 \frac{\beta}{\omega_0} = \frac{(0,4 + K_d)}{(1 + K_p)}$$

$$\beta = \frac{\omega_0 (0,4 + K_d)}{2 (1 + K_p)} = \frac{1}{2}$$

Bereken de waarde van K_d die deze vergelijking oplost / Calculate the value for K_d that solves this equation :

$$\frac{\omega_0 (0,4 + K_d)}{2 (1 + K_p)} = \frac{1}{2}$$

$$(0,4 + K_d) = \frac{(1 + K_p)}{\omega_0}$$

$$K_d = \frac{(1 + K_p)}{\omega_0} - 0,4 = \frac{5}{\sqrt{5}} - 0,4 \approx 1,836$$

11. Door gebruik van K_p zal het systeem met een statische fout in de einwaarde blijven zitten. Hoe groot is deze statische fout / Because of using K_p , the system will have a static error in the final value. How large is this static error?

antwoord/solution : Schakel de I actie uit om de statische fout te zien \Rightarrow Stel K_i gelijk aan 0 / Switch off the I action to see the static error \Rightarrow Make K_i equal to zero.

Statische eindwaarde \Rightarrow substitueer $s=0$ / Static end value \Rightarrow substitute $s=0$. De fout/error is

dan: $(1 - \frac{K_p}{1 + K_p}) = \frac{1 + K_p - K_p}{1 + K_p} = \frac{1}{1 + K_p} = \frac{1}{1 + 4} = \frac{1}{5} = 20\%$

12. Schat een waarde voor K_i waar mee het probleem uit de vorige vraag kan worden opgelost /

Estimate a value for K_1 that can solve the problem from the previous question.

***antwoord/solution** : Dit is moeilijk om te doen zonder tijdaanduiding die aangeeft hoe snel de fout moet worden geëlimineerd. Begin met een waarde van 0,1 en evalueer in een simulatie of testopstelling hoe snel de oplossing convergeert naar de eindwaarde 1 / *This is hard to do without a time boundary that indicates how fast the static error must be eliminated* . Start with a value of 0.1 and evaluate in a simulation or test setup how quickly the solution converges to the final value 1.*