

# Lab: emedded system design 1

Thomas Feys      Jona Cappelle

9 december 2019

## Inhoudsopgave

<b>1</b>	<b>Inleiding</b>	<b>1</b>
<b>2</b>	<b>Sensor</b>	<b>1</b>
<b>3</b>	<b>Systeem</b>	<b>2</b>
3.1	overzicht . . . . .	2
3.2	I2C . . . . .	4
<b>4</b>	<b>Code</b>	<b>4</b>
4.1	Functionaliteiten . . . . .	4
4.2	Flow van de code . . . . .	4
<b>5</b>	<b>Power consumptie</b>	<b>4</b>
5.1	Principe . . . . .	4
5.2	Metingen . . . . .	6
<b>6</b>	<b>Besluit</b>	<b>6</b>

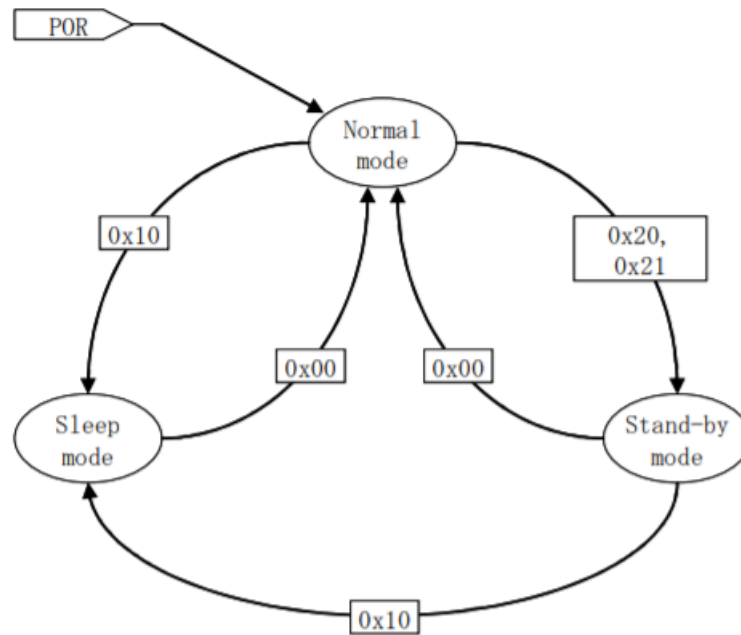
## 1 Inleiding

Ons doel is om de wachtrij in de rabotaria te monitoren. Dit zullen we doen aan de hand van een IR grid sensor (AMG8833). Aan de hand van de uitgelezen waarden zullen we in het tweede semester bepalen hoeveel mensen er in de wachtrij staan. Deze informatie zullen we vervolgens via een app of website verspreiden.

## 2 Sensor

De sensor die we gebruiken om de wachtrij te monitoren is de AMG8833. Deze IR grid sensor heeft 8x8 pixels die de temperatuur weergeven. Volgens de datasheet kan a.d.h.v. de temperatuur een mens waargenomen worden vanop een afstand van 5 meter. De sensor kan gebruikt worden in 3 verschillende modes:

normal, sleep en stand-by. In de normale mode heeft de sensor een verbruik van 4.5 mA. De stand-by mode heeft twee opties; de waardes kunnen geüpdatet worden om de 60 seconden of om de 10 seconden. In deze mode is er een verbruik van 0.8 mA. Als laatste is er de sleep mode deze verbruikt 0.2 mA. Een overzicht van alle modes en de commando's die verstuurd moeten worden om in deze modes te raken wordt weergegeven in figuur 1. Tijdens het testen van de sensor hebben we periodiek geswitched tussen de verschillende power modes. Het resultaat van deze test is te zien in figuur 2. De sensor heeft ook een interrupt pin. Deze pin genereert een interrupt als een van de pixels over of onder een bepaalde waarde gaat. Deze waarde is instelbaar.

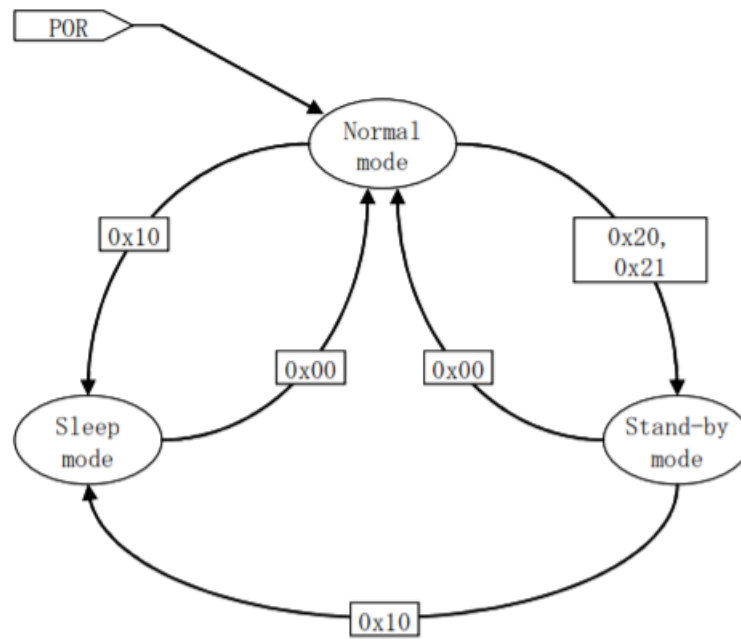


Figuur 1: Overzicht operating modes

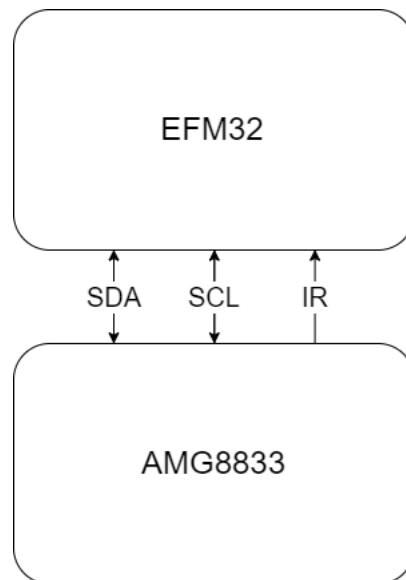
## 3 Systeem

### 3.1 overzicht

De AMG8833 communiceert met de EFM32 via I2C. Naast de I2C communicatie is er ook een interrupt pin voorzien op de sensor. Deze pin genereert een interrupt als er een van de pixels een bepaalde, instelbare waarde overschrijdt. Een volledig overzicht van het systeem is te zien in figuur 3.



Figuur 2: Test van de verschillende powermodes



Figuur 3: Overzicht van het systeem

## 3.2 I2C

# 4 Code

## 4.1 Functionaliteiten

Om gemakkelijk met de sensor te werken werd er een library geschreven voor de AMG8833. Er werden verschillende methodes geschreven om vlot te kunnen omgaan met de sensor. Er werd een functie geschreven om alle pixels uit te lezen. Naast de pixels is er ook een thermistor aanwezig in de sensor, ook hiervoor werd een functie geschreven. Er werden ook verschillende functies geschreven om makkelijk tussen de verschillende powermodes te kunnen wisselen. Een volledig overzicht van deze library is te vinden in .....

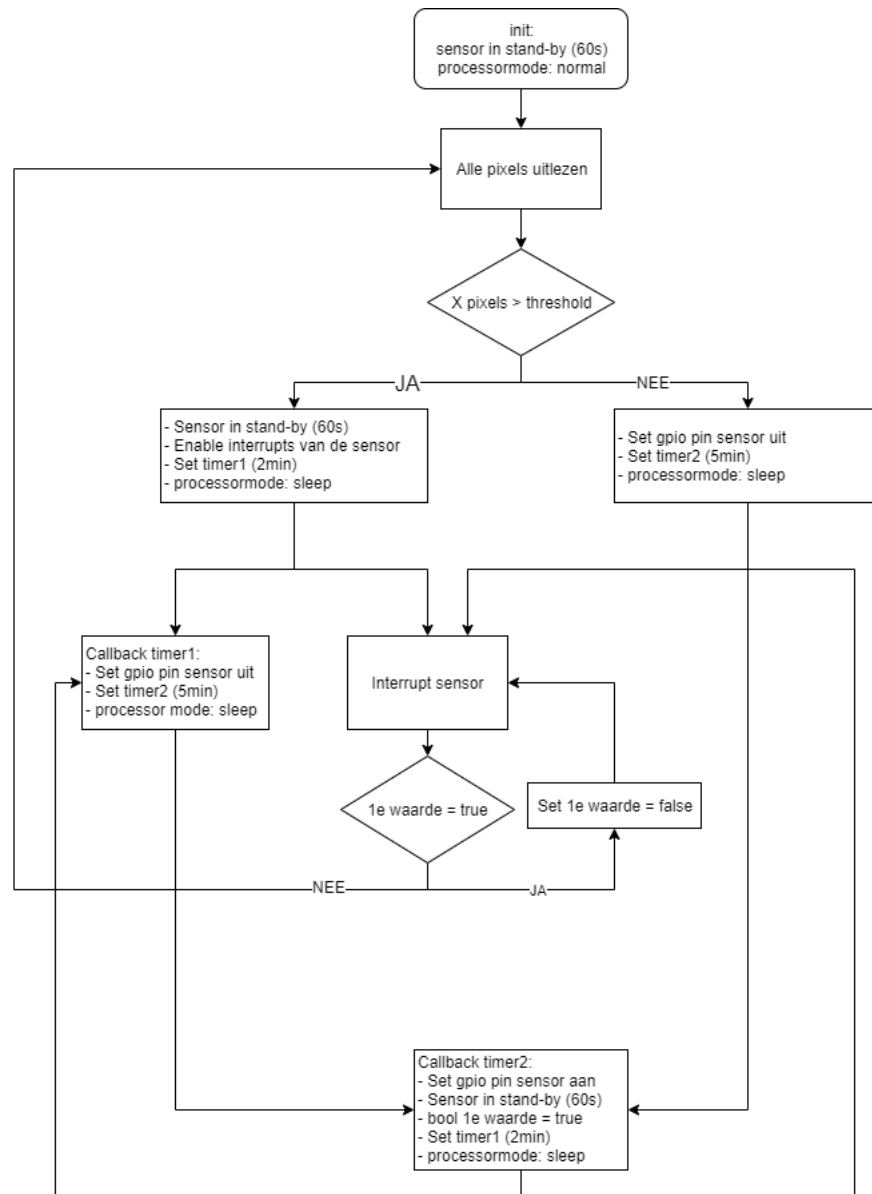
## 4.2 Flow van de code

Om het verbruik te minimaliseren wordt er maar om de 5 minuten gecontroleerd of er mensen aanwezig zijn. Omdat de sensor in sleep 0.2 mA verbruikt wordt de sensor volledig uit gezet door de gpio pin uit te zetten. Om de 5 minuten wordt de gpio pin aangezet en de sensor in de stand-by mode gezet waar de waardes om de 60 seconden ververs worden. Aangezien de output van de sensor 20 seconden nodig heeft om te stabiliseren wordt de eerste uitgelezen waarde genegeerd. De waardes worden uitgelezen als de sensor een interrupt genereert. Dit gebeurt als er 1 van de pixels boven de threshold ligt. Indien er na 2 minuten geen interrupt gegenereerd wordt, wordt de gpio pin van de sensor weer uitgezet. Indien er wel een interrupt gegenereerd wordt, wordt er gecontroleerd of er X aantal waarden boven de threshold zijn. Als dit het geval is, dan blijft de sensor in stand-by en wordt ze uitgelezen telkens als er een interrupt gegenereerd wordt. Indien er 2 minuten geen interrupt gegenereerd wordt, wordt de gpio pin weer uitgezet. Indien er niet voldoende pixels boven de threshold liggen dan wordt de gpio pin weer uitgezet. Een volledig overzicht van de code is terug te vinden in figuur 4.

# 5 Power consumptie

## 5.1 Principe

**Sensor:** Door de methode toe te passen die in besproken werd in section 4.2 wordt de power consumptie sterk verminderd. Wanneer er geen mensen aanwezig zijn, wordt het verbruik met 60 procent verminderd aangezien de sensor om de 5 minuten, 2 minuten aangezet wordt. Verder is het verbruik afhankelijk van hoe frequent er mensen in de wachtrij staan. In de berekeningen gaan we er van uit dat er enkel mensen aan het wachten zijn tussen 11u en 14u. In dit geval zal de sensor 3u per dag constant in stand-by staan aangezien er 3u lang mensen aanwezig zijn. Verder zal de sensor nog 21u, 24 minuten per uur aanstaan om periodiek te kijken of er iemand aan het wachten is. Als we dit na



Figuur 4: flowchart van de code

rekenen bekomen we dat de sensor 11.4 uur in stand-by staat. Dit komt uit op een stroom verbruik van:

$$\frac{11.4 \text{ h} \cdot 0.8 \text{ mAh}}{24 \text{ h}} = 0.38 \text{ mAh} \quad (1)$$

Indien deze methode niet toegepast wordt is er een verbruik van 0.8 mAh. We bekomen dus een besparing van 47.5 procent.

**Processor:**

## 5.2 Metingen

**Geen aanwezigheid:** meting doen met geen aanwezigheid: sensor constant aan en uit

**Aanwezigheid:** meting doen met constant x waarden  $>$  threshold

## 6 Besluit