Lab: Embedded system design 1

Thomas Feys Jona Cappelle

16 december 2019

Inhoudsopgave

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 gaan, 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 geneert een interrupt als een van de pixels over of onder een bepaalde waarde gaat. Deze waarde is instelbaar.

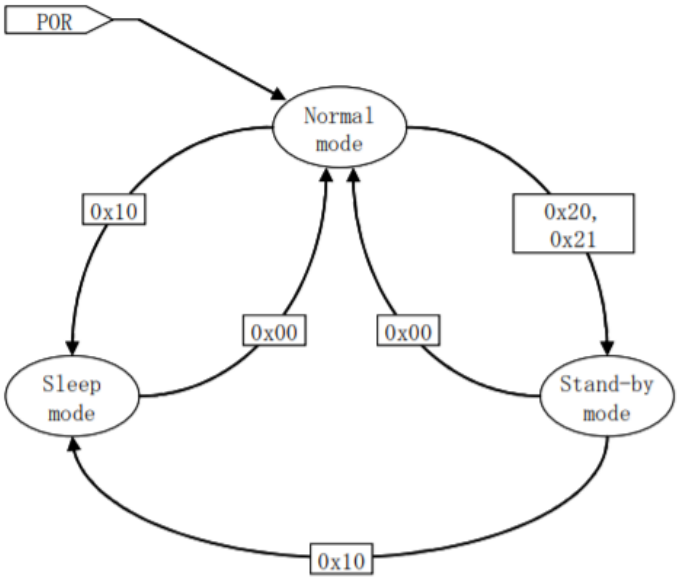
3 Systeem

3.1 Overzicht

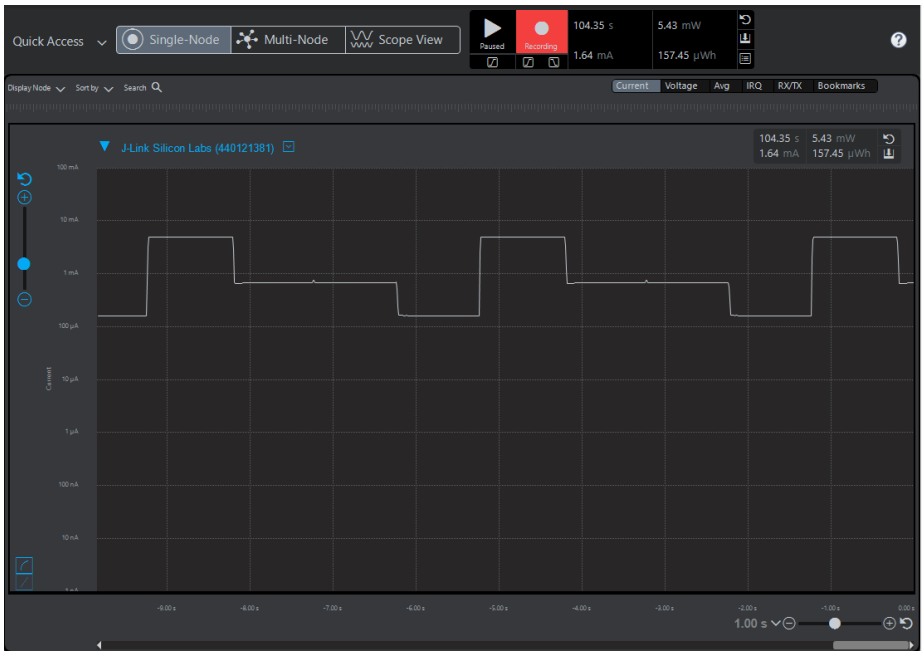
De AMG8833 communiceert met de EFM32 via I2C. Naast de I2C communicatie is er ook een interrupt pin voorzien op de sensor. Deze pin genereert een interrupt als er een van de pixels een bepaalde, instelbare waarde overschrijdt. Een volledig overzicht van het systeem is te zien in figuur 3.

3.2 I2C

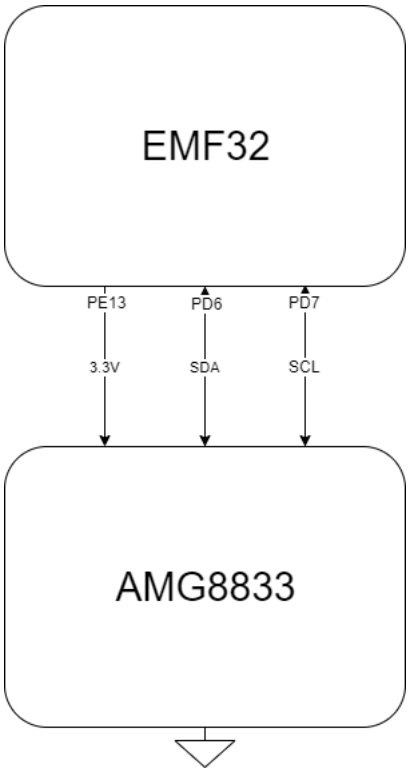
De communicatie met de sensor gebeurt via I2C. Het adres van de sensor is een 7 bit adres namelijk: 1101001. Aangezien we met een 7 bit adres zitten moet deze 1 bit geshift worden naar links. Dit adres moet telkens naar de sensor verstuurd worden vooraleer er iets gelezen of geschreven kan worden. De code die gebruikt wordt om de sensor uit te lezen, in andere powermodus te zetten en te interrupten wordt in de volgende sectie besproken.



Figuur 1: Overzicht operating modes



Figuur 2: Test van de verschillende powermodes



Figuur 3: Overzicht van het systeem

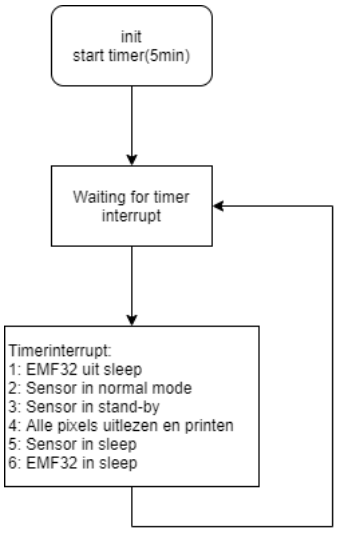
4 Code

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 en de rest van de code is terug te vinden op github: `https://github.com/jonacappelle/Embedded_Project`.

4.2 Flow van de code

Aangezien de sensor in stand-by een verbruik van 0.8 mA heeft wordt deze altijd in sleep gezet. De sensor wordt om de 5 minuten uit sleep gehaald om alle pixels uit te lezen. Hierna wordt ze weer in sleep gezet. Een volledig overzicht van de code is terug te vinden in figuur 4.



Figuur 4: flowchart van de code

5 Power consumptie

5.1 Sensor

De sensor heeft 3 verschillende powermodes, deze zijn te zien in figuur 5. Deze hebben we ook zelf opgemeten zoals te zien is in figuur 2. De gemeten waarden liggen in dezelfde grote orde als deze die opgegeven zijn in de datasheet, maar wijken toch een klein beetje af. In normal mode gebruikt onze sensor 4.95 mA, in stand-by 0.66 mA en in sleep 0.156 mA.

Energy Mode	Current
Normal	4.5 mA
Stand-by	0.8 mA
Sleep	0.2 mA

Figuur 5: powermodes van de sensor

5.2 Principe

Sensor: Door de methode toe te passen die besproken werd in section 4.2 wordt de power consumptie sterk verminderd. De sensor zit enkel in stand-by tijdens het uilezen van de sensor waarde. De sensor zit ook 105 ms in normal mode, omdat je altijd naar de normal mode moet gaan als je uit sleep komt. Er werd een meting uitgevoerd om dit verbruik te controleren, dit is terug te vinden in figuur **??**. Uit deze meting is te zien dat de sensor 105 ms in normal mode zit met een verbruik van 4.8 mA, 50ms in stand-by met een verbruik van 2.83 mA[1] en verder zit de sensor in sleep met een verbruik van 0.156 mA. Hierdoor bekomen we volgend verbruik per meting:

$$3.3\text{ V}\cdot\big(105\text{ ms}\cdot 4.8\text{ mA}+50\text{ ms}\cdot 2.83\text{ mA}+299.845\text{ s}\cdot 0.156\text{ mA}\big) = 622.973\text{ J/meting} \tag{1}$$

Een overzicht is te zien in tabel **??**. Als we om de 5 minuten een meting uitvoeren, bekomen we volgend verbruik:

$$12 \cdot 622.973\text{J/meting} = 7475.68\text{J} \tag{2}$$

$$\rightarrow 7475.68\text{ J}/3600\text{ s} = \tag{3}$$

In plaats van een constant verbruik van 0.8 mA verbrverbruikt de sensor maar 0.2 mA aangezien hij bijna constant in sleep zit. Het verbruik zal een klein beetje hoger zijn aangezien de sensor voor een kleine tijd weer in normal en stand-by mode gezet wordt.

Processor: De processor wordt enkel uit sleep gehaald als er een timerinterrupt afgaat. Hierdoor zit de processor voornamelijk in EM2 dit geeft een verbruik van 0.9 μA.

5.3 Metingen

Om het verbruik van het systeem te controleren werd een meting uitgevoerd. Het resultaat van deze meting wordt weergegeven in figuur **??**.

6 Besluit

[1] Tijdens het uitlezen van de sensor ligt het verbruik hoger, hierdoor verbruikt de sensor 2.83 mA in stand-by ipv 0.8 mA