**Inleiding**

**<Klaas>**

Stel je hebt een stuk software geschreven voor de Arduino. Tijd speelt en belangrijke rol en je wilt bijvoorbeeld de data van je weerstation elke maandag van de week om 12 uur ’s middags en 12 uur ’s avonds verzenden naar je Processing applicatie. Normaal gesproken maak je gebruik van een unsigned long waarin je de startijd en de huidige tijd bijhoudt en op basis van een interval de data verstuurt. Dit kan maar is nogal vereist nogal wat reken werk mocht je dit op een specifieke tijd op een specifieke dag/datum willen doen, aangezien je gebruikt maakt van de millis() functie die milliseconden bijhoudt. Bovendien wordt de startijd en huidige tijd weer op 0 gezet als je je Arduino loskoppelt van je computer en weer opnieuw aansluit op je computer.

Gelukkig is hiervoor een oplossing namelijk ons hardware component: De RTC.

**Persoonlijke leerdoelen**

**<Ieder z’n eigen leerdoel>**

Bart: Meer structuur, minder afwijken van het onderwerp en minder frunniken.

Klaas: Minder snel praten.

**Voorbeelden van toepassingen van de hardware**

**<Klaas>**

Een real-time klok is een klok die de huidige tijd bijhoudt en gebruikt kan worden om acties op een bepaalde tijd te laten uitvoeren . De meeste RTC’s gebruiken een **kristaloscillator** met een frequentie van 32.768 khz. (deze frequentie wordt ook gebruikt in klokken en horloges van kwarts).

Deze frequentie staat gelijk aan 2 tot de macht 15 cycles per seconde wat een goede snelheid is voor binaire tel circuits.

Daarnaast kan de RTC ook worden gebruikt in slaapmodus zodat het kan worden geprogrammeerd om een apparaat op een bepaalde tijd uit slaapmodus te halen. Wanneer je de Arduino opnieuw aansluit op je computer dan weet de RTC nog steeds welke tijd het is. De RTC heeft namelijk een eigen batterijtje en gaat dus door met doortellen ook als de Arduino geen stroom ontvangt..

**<Bart>**

Een van de voorbeelden waarin een RTC wordt gebruikt is een personal computer. Dit zorgt ervoor dat de tijd op je computer altijd wordt bijgehouden ook als deze uitstaat. Vroeger waren RTC modules op computermoederborden duidelijk zichtbaar zoals in het plaatje te zien is. Tegenwoordig zijn ze bijna altijd geïntegreerd in de chipset van het moederbord.

Sommige microcontrollers maken tevens gebruik van een RTC. Meestal zijn dit microcontrollers die over veel andere kenmerken beschikken die te maken hebben met informatie uit een computer halen en in een computer stoppen. Ook is een RTC voor sommige microcontrollers los aan te schaffen, mocht deze er geen hebben ingebouwd, zoals bij de Arduino en de Raspberry Pi 3.

**Alternatieve hardware/techniek**

**<Bart>**

Als alternatief van de DS1307 zou je een DS3231 of een WiFi-shield kunnen gebruiken, hieronder zie je een vergelijking:

|  |  |  |  |
| --- | --- | --- | --- |
| Product | DS1307 | DS3231 | WiFi-shield |
| Prijs € | ~2,10 | ~2,25 | ~80,00 |
| Formaat | 2,8cm X 2,7cm | 3,8cm X 2,2cm | 8,0cm X 6,0 cm |
| Geheugen | 56 bytes | 32K | Uitbreidbare SD-kaart slot. |
| Max jaar | 2100 | 2100 | nvt |
| Snelheid | 100 kHz | 400 kHz | 802,11b/g netwerk |
| Aanvullend | nvt | Geïntegreerde temperatuur gecompenseerd kristaloscillator (TCXO) aanboort | Kan gebruikt worden voor andere doeleinden behalve tijd. |

Voordeel van de DS3231 tegenover de DS1307 is dat hij nauwkeuriger is, waardoor je minder snel een controle hoeft uit te voeren, daarnaast heeft hij een TCXO aanboort (dit zorgt onder andere voor de betere prestatie).

Daarin tegen heeft de DS1307 een kleiner formaat en is gemiddeld gezien goedkoper. Daarvoor lever je wel nauwkeurigheid in waardoor hij minder geschikt wordt voor projecten waar je op de seconde nauwkeurig je meting wilt lezen, dit verschil is enkel een paar seconde op dag basis.

Voor projecten waar formaat belangrijk is kan je beter voor de DS1307 gaan.

Daarnaast kun je ook nog met een WiFi-shield de tijd opvragen, dit kan dan via een site die de tijd geeft. Het voordeel hiervan is dat dit uiterst nauwkeurig is aangezien deze sites vaak gebruik maken van het nationale tijd zones. Je kan het shield ook gebruiken voor andere doeleinden zoals verbinden met je eigen thuis netwerk, en aangestuurd worden op afstand, voor bijvoorbeeld je werker van je mobiel instellen.

Het WiFi-shield heeft wel de grootste afmetingen en is een stuk duurder, dus als je hem alleen wilt gebruiken voor de tijd kan je beter gaan voor een van de andere opties.

In plaats van deze alternatieven kun je ook de Arduino zelf gebruiken om de tijd bij te houden. De voordelen hiervan zijn :

* Bij het gebruik van een RTC ben je een paar pinnen op je bordje kwijt. In het geval van het gebruiken van de Arduino voor tijd heb je deze pinnen beschikbaar voor andere hardware
* Je hoeft geen extra hardware component te kopen (RTC)
* Je hoeft geen extra hardware component te programmeren

De nadelen:

* Een RTC is preciezer dan de Arduino zelf.
* Een RTC heeft zelf een batterij of kan op een batterij worden aangesloten. Hierdoor kan de RTC worden gebruikt terwijl de Arduino geen stroom krijgt
* Een RTC blijft de tijd onthouden als deze niet stroom krijgt via de Arduino wanneer de Arduino crasht of wordt gereset
* Een RTC houdt zelf de dag, maand en het jaar bij. De Arduino doet dit niet.

**Hardware architectuur**

**<Klaas>**

Dit hoe je een RTC aansluit op je Arduino. De aansluiting lijkt erg veel op de theorie van de i2c bus van Philips die je ook voor de ESD toets moet kennen. Hierbij wordt namelijk gecommuniceerd via twee draden namelijk een datadraad (de SDA genoemd en een klokdraad (de SCL genoemd). De klok zorgt er hierbij voor dat de ontvanger weet wanneer er een nieuwe bit uit de data-draad gelezen kan worden. Bij de Arduino Uno sluit je de SDA-lijn (data lijn) op de A4 pin van de Arduino aan. Dit is op het plaatje de gele lijn. De kloklijn sluit je aan op de A5 pin wat in het plaatje de groene lijn is. De ground en de 5v sluit je natuurlijk op de 5v en gnd pinnen van de Arduino aan.

**Uitleggen code <Bart + Klaas>**

**Slot**

**<Bart>**

Kort samengevat is de RTC DS1307 een goed alternatief op de interne klok van de Arduino zelf. Dit komt voornamelijk doordat de DS1307 de tijd nauwkeuriger meet dan de Arduino en ook gegevens als de dag, de maand en het jaar bijhoudt. Ten opzichte van alternatieve merken van de RTC is de DS1307 wat minder nauwkeurig op de lange termijn. Daartegen is de DS1307 wel goedkoper en compacter waardoor deze makkelijk is mee te nemen en minder ruimte inneemt in je opstelling.

**Bronnen**

# Bibliografie

*http://domoticx.com/arduino-rtc-tijdklok-ds1307/*. (sd). Opgehaald van Domoticx.

*http://www.best-microcontroller-projects.com/real-time-clock-ic.html*. (sd). Opgehaald van best-microcontroller-projects.

*http://www.instructables.com/id/TESTED-Timekeeping-on-ESP8266-Arduino-Uno-WITHOUT-/*. (sd). Opgehaald van Instructables.

*http://www.reuk.co.uk/wordpress/accurate-ds3231-real-time-clock-as-alternative-to-ds1307/*. (sd). Opgehaald van reuk.co.

*https://hackerstore.nl/Artikel/233*. (sd). Opgehaald van Hackerstore.

*https://learn.adafruit.com/ds1307-real-time-clock-breakout-board-kit/what-is-an-rtc* . (sd). Opgehaald van Adafruit.

*https://www.arduino.cc/en/Reference/RTC*. (sd). Opgehaald van Arduino Reference.

*https://www.distrelec.nl/nl/arduino-wifi-shield-a000058-arduino-a000058/p/11038916?channel=b2c&price\_gs=83.49&source=googleps&ext\_cid=shgooaqnlnl-na&pup\_e=1&pup\_cid=36007&pup\_id=11038916&ext\_cid=shgooaqnlnl-na-&kw=&gclid=CjwKCAjw4sLVBRAlEiwASblR-7KtwCLBQv*. (sd). Opgehaald van Distrelec.

*https://www.vanallesenmeer.nl/Real-Time-Clock-DS3231-RTC-Klok-module*. (sd). Opgehaald van vanallesenmeer.

*Real-Time-Clock*. (sd). Opgehaald van Wikipedia: https://en.wikipedia.org/wiki/Real-time\_clock