Auteur: Max Huiskes (2151960)

Klas: 32022INF1

Vak Beroepsproduct 5/6

School: Avans Stratenlaan te Den Bosch

Datum: 25-05-2023

Versie: 1.0

Sensorisch computersysteem

Inhoudsopgave

[Inleiding 2](#_Toc146529925)

[Data van The Thinngs Network 3](#_Toc146529926)

[Beschrijving arduino processor 4](#_Toc146529927)

[Configuratie van de Atmega 328P 6](#_Toc146529928)

[Lora Network 6](#_Toc146529929)

[Arduino 7](#_Toc146529930)

[Samenvatting 8](#_Toc146529931)

# Inleiding

Dit artikel is er om mensen te informeren over de computerarchitectuur die tijdens het faunaproject wordt gebruikt en welke componenten hier geschikt zijn. Het werk met de Arduino UNO Ref 3 met processor Atmega 328P is gepland om te beginnen. Later zal er meer informatie worden verstrekt over de inhoud van deze processor en hoe de gegevens van de gebruikte sensoren via het TTN-netwerk zullen worden behandeld. De sensoren die de gegevens leveren, zijn onder meer een temperatuursensor (DS18B20), een gewichtssensor (HX711), een infrarood breeksensor (IR Breakbeam-sensor) en een radarbewegingssensor (RCWL-0516). Op de laatste pagina wordt een samenvatting gegeven van de meest cruciale details van het document.

# Data van The Thinngs Network

Aan de rand van het netwerk staan ​​sensoren die data verzamelen binnen de Arduino Uno Rev3. De Arduino Uno Rev3 verzendt draadloos gegevens naar de dichtstbijzijnde TTN-gateway via de Lora-radiofrequentie.

Een TTN-gateway is een hardware apparaat dat gegevens van sensorapparaten ontvangt en deze doorstuurt naar het TTN-netwerk. Ze fungeren als tussenstations tussen sensorapparaten en de TTN-netwerkkern. Deze gateways ontvangen gegevens op verschillende radiofrequenties en verzenden deze via een internetverbinding (zoals Ethernet of mobiele data) naar het TTN-netwerk.

De TTN-netwerkserver is een belangrijk onderdeel van de architectuur. Het ontvangt gegevens van de gateway en verwerkt deze. Dit omvat het identificeren van het apparaat, het valideren van de ontvangen gegevens en het doorsturen van de gegevens naar het juiste applicatie-integratiepunt.

Nadat de gegevens door de TTN-netwerkserver zijn verwerkt, worden deze naar een specifiek applicatie-integratiepunt gerouteerd. Dit kunnen cloud gebaseerde services, servers of applicaties zijn die zijn ontwikkeld om IoT-gegevens te verwerken. Hier kunnen ontwikkelaars gegevens analyseren, visualiseren en gebruiken voor een verscheidenheid aan toepassingen, zoals smart city-projecten, industriële monitoring of toepassingen in de gezondheidszorg.

In TTN-netwerken wordt beveiliging ook serieus genomen. Gegevens worden vaak gecodeerd om de vertrouwelijkheid en integriteit ervan te garanderen. Dit is vooral belangrijk omdat IoT-apparaten gevoelige informatie kunnen verzenden.

In wezen worden in een TTN-netwerkarchitectuur gegevens van een sensorapparaat via een gateway naar een TTN-netwerkserver verzonden, waar ze worden verwerkt en doorgestuurd naar een specifiek applicatie-integratiepunt voor verdere verwerking en analyse. Deze end-to-end infrastructuur maakt het mogelijk om IoT-gegevens op schaal te verzamelen en te gebruiken voor een verscheidenheid aan toepassingen.

Bij het versturen van data over het Lora netwerk mag je maar een duty cycle tussen de 1% of 0,1% hebben. Deze vermenigvuldig je me “consument\_airtime” zoals dat heet binnen het verzonden pakket. Met deze uitkomst krijg je hoe lang je moet wachten tot je weer een bericht mag sturen. Wij zijn uitgegaan van een duty cycle van 1% en onze consument\_airtime was 0.061696 seconde. Dus onze berekening is als volgt:

1%

99%

Dus wij mogen elke 7 seconde een bericht sturen met een duty cycle van 1%.

# **Afbeelding met diagram, Plan, schematisch, Technische tekening Automatisch gegenereerde beschrijving**Beschrijving arduino processor

Voor dit project is gebruik gemaakt van de Arduino Uno Rev. Dit is een microcontrollerbord dat u kunt programmeren om lichten, motoren en andere apparaten te bedienen. De Atmega328P dient als de on-board processor van de controller board. Ik zal de algemene werking van de kritieke hardwarecomponenten op de Atmega328P hieronder beschrijven.

**CPU of centrale verwerkingseenheid:** De belangrijkste verwerkingseenheid van de microcontroller is de CPU. Dit geeft de instructies die op het flashgeheugen zijn geschreven. De CPU kan vaak redeneren en logische bewerkingen uitvoeren, gegevens opslaan en besturingsinstructies uitvoeren.

**Flashgeheugen:** Het Flash-geheugenapparaat heeft een capaciteit van 32 kilobyte. Dit apparaat wordt gebruikt om de programmeerinterface (de Arduino-schets) in beweging te zetten. Deze haag zal blijven bestaan, zelfs nadat de kamer van de Arduino is gereset.

**SRAM (Statisch ram):** De SRAM heeft een geheugencapaciteit van 2 kilobyte (KB). Dit wordt gebruikt voor het tijdig opslaan van variabelen en informatie terwijl de toepassing wordt uitgevoerd. Sinds de Arduino's stroom is verwijderd, worden deze gegevens daar niet meer opgeslagen.

Figuur 1 (Microship, 2015)

**EEPROM voor elektrisch programmeerbaar geheugen:** Deze poort is voor het voortdurende behoud van informatie die privé moet worden gehouden, zelfs als de kamer wordt opgeruimd. Dit is handig voor het toevoegen van configuratie-informatie en kalibratiestandaarden. Dit bestand heeft een gegevensopslag van 1 kilobyte.

**I/O-modules (input/output):** De microcontroller kan worden gekoppeld aan externe apparaten dankzij de 14 digitale en 6 analoge I/O-pinnen. Om logische signalen te lezen of te maken, kunt u de digitale pinnen installeren als invoer- of uitvoerpennen. U kunt analoge signalen meten met behulp van de analoge pinnen en de ingebouwde ADC (Analoog naar Digitaal Omzetter).

**Interrupt Unit**: De interrupt unit rust de microcontroller uit om onmiddellijk te reageren op externe gebeurtenissen zonder eerst te controleren of er een wijziging is aangebracht. Het apparaat kan verschillende soorten interrupts aan, waaronder interne software-interrupts en hardware-interrupts.

**Interfaces voor communicatie:** De microcontroller ondersteunt beveiligde communicatie via UART (ook bekend als het beveiligde kanaal of de beveiligde pijp), die wordt gebruikt om beveiligde gegevens te verzenden en te ontvangen. Daarnaast heb je de SPI (Serieel Perifeer Interface of SPI-interface) voor seriële communicatie met andere SPI-compatibele apparaten, evenals de I2C (Inter-Gemeente Schakelcircuit of I2C-Circuit) voor serieuze communicatie van de tweede generatie met I2C-compatibele apparaten.

**Tellers en timers**: De microcontroller heeft timers en tellers die kunnen worden gebruikt om timers te genereren, tijd te meten of dingen bij te houden die al zijn gebeurd.

**Analog-to-Digital Converter (ADC):** De ADC zorgt ervoor dat analoge signalen worden omgezet in digitale signalen, zodat de microcontroller de signalen van analoge sensoren, zoals temperatuur, kan lezen.

**Klok- en Voedingsmanagement**: De Atmega328P ontstaat via een elektrische swapping. Dit heeft een standaard kloksnelheid van 16MHz, maar het kan ook worden gewijzigd om energie te besparen of de microcontroller met een hoge snelheid te laten werken. Dit is handig voor toepassingen op batterijen. Er zijn ook externe klokbronnen zoals krisallen, zodat u meer precisie kunt verkrijgen. Bovendien werkt het energiebeheer van de microcontroller met een vermogensbereik tussen 1,8 en 5,5 volt. Het is gebruikelijk om deze in de batterijmodus te plaatsen, zoals de slaapstand, wanneer de toepassing voor batterijen begint toe te passen.

Dit waren de kritieke hardwarecomponenten die samenwerkten om de Atmega328P voor de Arduino in staat te stellen programma's uit te voeren, gegevens te verwerken, te communiceren met andere apparaten en invoer-/ uitvoeracties uit te voeren. Door deze hardwarecomponenten te programmeren en te gebruiken, kunnen we ons project maken met behulp van de Arduino.

# Configuratie van de Atmega 328P

## Lora Network

Deze code is ontworpen voor het gebruik van een Arduino in combinatie met LoRaWAN-technologie om real-time communicatie met het Internet of Things (IoT) mogelijk te maken. Laten we de specifieke configuratie van de hardwarecomponenten beschrijven en hoe deze zich verhouden tot de softwaretoepassing:

1. Processor (Arduino): De code wordt uitgevoerd op een Arduino-microcontroller, die dient als de hardware basis van het apparaat. Vanwege de programmeerflexibiliteit en gebruiksvriendelijkheid zijn Arduino's behoorlijk populair.

2. LoRaWAN-bibliotheek: De code maakt gebruik van de LoRaWAN-bibliotheek, die communicatie via het LoRaWAN-protocol mogelijk maakt. Deze bibliotheek is nodig voor het verzenden en ontvangen van gegevens via de LoRa-radio.

3. SPI-bibliotheek: De SPI-bibliotheek wordt gebruikt voor serieuze communicatie met de SPI-interface, die nodig is om te communiceren met LoRa-apparaten.

4. Pinconfiguratie: Er wordt een bepaalde pinconfiguratie opgegeven voor communicatie met de LoRa-module. De meest kritische punten zijn:

- ".nss": Pin 10, die wordt gebruikt voor chipselectie tijdens SPI-communicatie.

'.rxtx': deze pin wordt niet gebruikt (LMIC\_UNUSED\_PIN).

- '.rst': de resetpin van de LoRa-module (9).

- '.dio': Een set pinnen die wordt gebruikt voor digitale in- en uitvoer, zoals interne interruptpinnen voor het ontvangen van nieuws (pinnen 2, 6 en 7).

5. Callback-functies: Callback-functies zijn gedefinieerd om te reageren op gebeurtenissen in het LoRaWAN-protocol. "onEvent" behandelt bijvoorbeeld een verscheidenheid aan gebeurtenissen, waaronder netwerksluiting, de ontvangst van gegevens en meer.

6. LoRaWAN-sleutels omvatten "NWKSKEY" (Network Session Key) en "APPSKEY" (Application Session Key) voor veilige communicatie met het LoRaWAN-netwerk. Deze sleutels worden gebruikt om de netwerkverbinding te inspecteren.

7. Transmissieschema: Er is een opzet voor gegevensoverdracht waarbij de Arduino periodiek gegevens (in dit geval "Hallo, wereld!") over het LoRaWAN-netwerk verzendt. De 'TX\_INTERVAL'-constante, die na 60 seconden wordt ingenomen, regelt het verzendschema.

8. Setup-functie: Hier wordt de basisconfiguratie uitgevoerd, inclusief de installatie van de seriële communicatie, de installatie van pinmodifiers en de initialisatie van de LoRaWAN-module. Daarnaast worden de sessie-instellingen en kanaal-configuraties ingevoerd.

9. Loop-functie: Deze functie wordt continu uitgevoerd en zorgt ervoor dat de LoRaWAN-stack wordt beheerd en bewerkt.

Over het algemeen gebruikt deze code bepaalde hardwarecomponenten, waaronder de LoRa-module en Arduino, om LoRaWAN-communicatie te bieden. Om ervoor te zorgen dat deze hardware gegevens via het LoRaWAN-netwerk verzendt en reageert op communicatie gerelateerde gebeurtenissen, configureert en beheert de software deze.

## Arduino

Deze code is een Arduino-applicatie die verbinding maakt met verschillende sensoren en sensorgegevens uitvoert via een realtime monitor. De code is verdeeld over een aantal situaties, waarbij elke sensor zijn eigen klasse heeft. Laten we de code en gerelateerde klassen eens nader bekijken:

('main.ino') header informatie:

Vier sensoren worden in dit bestand geïnitialiseerd en gebruiksklaar gemaakt bij logging operations door de 'setup()'-functie.

De parameters van elke sensor worden geëvalueerd in de functie "loop()" en vervolgens naar de serieuze monitor gestuurd.

- Hoewel 'LOOP\_DELAY' is gedefinieerd, lijkt het er momenteel op dat de code er geen gebruik van maakt.

('GewichtSensor.h' in 'GewichtSensor.cpp'):

- Deze klasse vereist een gewichtssensor, die waarschijnlijk is gebaseerd op een HX711-chip.

- De pinnen en meetinterval worden geïnitialiseerd in de constructor.

- De functie 'setup()' stelt de kalibratiefactor in en initialiseren we de gewichtssensor.

- De functie 'getWeight()' leest de gewichtsstandaard en retourneert deze.

('RadarSensor.h' en 'RadarSensor.cpp'):

Deze categorie vereist een radarsensor.

- De pin van de radarsensor wordt geïnitialiseerd door de constructeur.

- De pin wordt geconfigureerd als een input pin door de "setup()" functie.

- De functie 'getReading()' controleert het bereik van de radarsensor en retourneert 'true' wanneer een item wordt gevonden, anders 'false'.

('TemperatuurSensor.h' en 'TemperatuurSensor.cpp'):

Deze klasse vereist een temperatuursensor die (duidelijk) is gebaseerd op de Dallas Temperature Library.

- De pin en meetinterval worden geïnitialiseerd door de constructor.

- De temperatuursensor wordt geïnitialiseerd via de functie "setup()".

- De functie 'getTemperature()' leest het temperatuurbereik en retourneert het in graden Celsius.

('InfraSensor.h' en 'InfraSensor.cpp'):

- Deze klasse vereist een infraroodsensor.

- De pin, de drempelwaarde en het meetinterval worden geïnitialiseerd door de constructeur.

- De pin wordt geconfigureerd als een input pin door de "setup()" functie.

- De functie 'getValue()' leest de analoge waarde van de infraroodsensor en retourneert deze.

- De input pin moet een opschakel weerstand hebben om te kunnen "lezen".

Deze code dient als een voorbeeld van hoe verschillende sensoren kunnen worden geïntegreerd en gebruikt met Arduino in het algemeen. Het combineert de informatie van alle sensoren en stuurt deze terug via de officiële monitor. Verder kan deze code worden verbeterd om informatie naar andere systemen te verzenden of om bepaalde acties uit te voeren op basis van sensorgegevens.

# Samenvatting

Dit artikel bespreekt de computerarchitectuur die wordt gebruikt in het faunaproject en de gebruikte componenten. Het project maakt gebruik van de Arduino Uno Ref 3 met de Atmega 328P-processor, die is ontworpen om te beginnen. De gebruikte sensorapparaten omvatten temperatuur-, gewichts-, infrarood- en radarsensoren. Het TTN-netwerk wordt gebruikt om gegevens te verzenden tussen sensorapparaten en het netwerk, dat functioneert als tussenstations. De TTN-netwerkserver is een cruciaal onderdeel van de architectuur, waardoor gegevensanalyse, visualisatie en gebruik voor verschillende doeleinden mogelijk is. De Arduino Uno Rev wordt gebruikt als het microcontrollerbord, dat kan worden geprogrammeerd om lichten, motoren en andere apparaten te bedienen. De CPU is het belangrijkste werkingsprincipe en het flashgeheugen wordt gebruikt voor gegevensregistratie en instructies. Het SRAM wordt gebruikt voor datalogging en informatieopslag. De EEPROM wordt gebruikt voor gegevensopslag en configuratie-informatie. De microcontroller kan worden aangesloten op externe apparaten via 14 digitale en 6 analoge I/O-pinnen. De interrupt-eenheid wordt gebruikt om te reageren op externe gebeurtenissen zonder een signaal te activeren. De microcontroller ondersteunt ook veilige communicatie via UART- en SPI-interfaces.

De Arduino-code biedt een communicatieprotocol voor LoRaWAN-communicatie, waardoor gegevensoverdracht via het netwerk mogelijk is. Het bevat installatie-, initialisatie- en loopfuncties, die ervoor zorgen dat hardwarecomponenten correct zijn geconfigureerd en functioneren. De code bevat ook hardwarecomponenten zoals LoRa-modules en Arduino voor LoRaWAN-communicatie. Het laat zien hoe verschillende sensoren kunnen worden geïntegreerd en gebruikt met Arduino, waarbij gegevens van alle sensoren worden gecombineerd en via de officiële monitor worden teruggestuurd.