



Project 3-it

Monitoring van silo's

Brent De Vos

Brent Belliën

Dieter Janssens

Guylian De Wit

Alessandro Dierickx

Davy Van Roey

CAMPUS

Geel

IT Factory

2 CCS A



Inhoudsopgave

1	Inleiding	2
2	Opdracht	2
3	Plan van aanpak.....	2
3.1	Aanleiding en achtergrond	2
3.2	Verwacht resultaat	2
3.3	Business case en doelgroepen	3
3.4	Planning	3
3.5	Projectafbakening en risicoanalyse	4
3.6	Informatie en rapportering	4
3.7	Projectteam	5
4	Lora	6
4.1	Inleiding	6
4.2	Het LoRaWAN-netwerk	6
4.3	Beveiliging	8
5	Opstelling op afstand	9
6	Gebruikte componenten	10
6.1	Raspberry pi 3b+	10
6.2	Ultrasonische sensor	10
7	Data opslaan in database.....	11
8	Data visualiseren	11
9	Besluit	14

1 Inleiding

Dit document beschrijft de loop van ons project waarin we aan volumemeting van silo's doen. Dit is gebeurd via een raspberry pi-opstelling die we op afstand plaatsen. Het is belangrijk aan te halen dat het om een simulatie gaat van een silo, wat betekent dat dit een virtueel project is met een virtueel resultaat dat gevisualiseerd zal worden via een site.

2 Opdracht

Dit project heeft als opdracht het monitoren van de inhoud van silo's. Dit komt neer op een volumemeting van de silo's waardoor men via alerts te weten kan komen wanneer een silo bijgevuld moet worden. Dit zou op verschillende locaties op eenzelfde moment mogelijk moeten zijn, wat betekent dat we met een opstelling op afstand zullen moeten werken.

3 Plan van aanpak

3.1 Aanleiding en achtergrond

De opdracht komt vanuit 3-it, het bedrijf van docent Dominik Verrydt, maar het project zal gebruikt worden in de bouwsector. Deze sector heeft een probleem met monitoring van mortelsilo's waarvoor wij een oplossing moeten bedenken.

De situatie bestaat uit een silo, die gevuld is met cementpoeder en aangesloten is op een waterleiding. Wanneer men op een knop drukt, start het mengproces en kan men mortel uit de silo laten lopen. Het probleem hiermee is dat men niet kan zien wanneer de silo bijgevuld moet worden. Hierdoor kan men tijdverlies lijden wanneer de silo onverwacht leeg blijkt te zijn.

3.2 Verwacht resultaat

Van ons wordt er verwacht dat wij met behulp van een Pi 3b+ een oplossing bedenken waarbij men de silo op tijd kan bijvullen en er hierdoor tijd en geld bespaard kan worden.

Hiervoor zullen we dus, zoals gezegd, een Pi 3b+ met een ultrasoon sensor gebruiken om leegte in de silo te meten. Wanneer we hiermee ondervinden dat de silo nog maar 25% of minder gevuld is, komt er een e-mail alert naar de werfleider of bevoegde om een nieuwe bestelling van mortel te plaatsen.

Doordat er na vulling stofdeeltjes aanwezig zijn in de silo, kunnen deze deeltjes de ultrasoon sensor misleiden. Hierdoor zal men na levering een dertigtal minuten moeten wachten om een lezing van het volume te doen. Dit tijdslot van dertig minuten zal in rekening gebracht moeten worden in het systeem dat wij ontwikkelen.

Verder zullen wij PubNub gebruiken om de gegevens van de Pi 3b+ uit te lezen.

Deze gegevens zullen terechtkomen in een database via phpmyadmin waar ze opgeslagen zullen worden. Deze gegevens zullen gaan over de datum van vulling, tijd sinds laatste vulling, het huidige percentage van volume mortel, De informatie in de database zal getoond worden op een grafische interface, meer bepaald een website, in de vorm van statistieken, grafieken, alerts,

3.3 Business case en doelgroepen

Onze oplossing zal ervoor zorgen dat bouwbedrijven, zoals eerder aangehaald, geen tijdverlies, en dus geldverlies, zullen lijden doordat de silo's niet meer onverwacht leeg zullen zijn.

Het nut voor de organisatie, in dit geval 3-it, is dat wij voor hen een bruikbaar en efficiënt systeem zullen creëren dat zij kunnen verkopen aan bedrijven in de bouwsector.

Het project is dus eigenlijk bestemd voor deze bedrijven in de bouwsector, aangezien zij er het meeste mee zullen zijn.

3.4 Planning

Eerst moeten we een plan van aanpak hebben, zodat we weten wat we gaan doen.

Daarna moeten we ervoor zorgen dat we bij iemand een Pi 3b+ kunnen installeren, zodat deze continu beschikbaar kan zijn. We zullen de opstelling moeten bepalen en deze testen in de les. Hierbij zullen we ook code moeten schrijven om uiteindelijk een volledige test te kunnen doen.

Tegelijkertijd zullen we een onderzoek moeten doen omtrent de beschikbaarheid van internet, voor ons is dit via LoRa. Ook moeten we de database en website aan de praat krijgen. Wanneer dit gedaan is, kunnen we ons gaan voorbereiden op de demo door de silo te simuleren. Daarna zullen we alles in de projectbundel moeten samenbrengen en de PowerPoint presentatie voorbereiden.

3.5 Projectafbakening en risicoanalyse

Het project bestaat uit het creëren van een meetsysteem met toegankelijke interface. Dit betekent dat het onze verantwoordelijkheid is dat men ten eerste kan vertrouwen op de gemaakte metingen en dat men ten tweede altijd toegang heeft tot de interface. Zaken zoals het welzijn van de silo's en het onderhoud van het meetapparatuur valt niet onder onze verantwoordelijkheid.

De enige risico's die onder onze verantwoordelijkheid vallen zijn de risico's die vasthangen aan het vertrouwen dat men in ons systeem zal hebben. Als ons systeem faalt, of niet werkt zoals het moet, kan dit financiële gevolgen hebben voor de klant, wat voor ons een risico kan zijn. Risico's die niet voor ons zijn bevatten vooral situationele risico's waaraan wij, als makers van het systeem niet op voorbereid konden zijn, zoals bijvoorbeeld natuurrampen.

Maatregelen die genomen kunnen worden door de klant om problemen te vermijden bestaan vooral uit onderhoudsmogelijkheden en extra fysieke beveiliging van het systeem in de silo.

3.6 Informatie en rapportering

Alle informatie zal verzameld worden in de database, met de mogelijkheid om het te tonen via de website. De rapportering zal gebeuren via een projectbundel waarin het hele project in detail beschreven zal worden, met de nodige extra studies erin verwerkt. Ook zullen wij een Powerpointpresentatie geven waarin we de loop van het project uitleggen en meer informatie geven over ons projectteam.

3.7 Projectteam

	Opstelling	Remote maken	Code	Weergave	Projectbundel / papierwerk
Brent De Vos		✗	✗		
Dieter Janssens	✗		✗		
Guylian De Wit		✗	✗		
Brent <u>Belien</u>	✗			✗	
Davy Van <u>Roey</u>		✗			✗
Alessandro Dierickx				✗	✗

4 Lora

4.1 Inleiding

LoRa (Long Range) is de technologie die gebruikt wordt om te communiceren tussen 2 punten over een lange afstand. Het netwerkprotocol dat gebruikt wordt om deze technologie te ondersteunen is LoRaWAN.

LoRa gebruikt als medium voor de communicatie niet het internet, maar de ISM-radiofrequentieband. ISM is afkorting voor "Industrial, Scientific and Medical". Deze frequenties zijn gratis. Men heeft dus geen licentie nodig om van deze frequenties gebruik te maken. De enige voorwaarde is dat maar maximum 1% van de bandbreedte gebruikt mag worden, wat ongeveer overeen komt met 140 berichten van 12 bytes per dag.

LoRaWAN-netwerken behoren tot de groep van LPWAN (Low Power Wide Area Network). Deze groep van netwerken wordt speciaal gebruikt om IoT-sensoren over een grote afstand te kunnen monitoren. Het grote voordeel van een LPWAN is dat de sensoren gekoppeld aan het openbaar netwerk zeer weinig energie verbruiken, waardoor ze op gewone batterijen tot 5 jaar lang kunnen meegaan. Ook is het netwerk geoptimaliseerd om het dataverbruik zeer laag te houden, waardoor de kostprijs dat ook is.

Van alle LPWAN protocollen is LoRaWAN de populairste. Ze zijn namelijk de enige die de mogelijkheid geven om privé-netwerken op te stellen. In België worden deze netwerken ondersteund door Proximus.

4.2 Het LoRaWAN-netwerk

Voor het implementeren van een LoRa-netwerk zijn de volgende onderdelen nodig:

- Sensoren die de data versturen over radiofrequenties
- Gateways die de radiofrequenties ontvangen en naar de server sturen
- De server die de communicatie tussen de LoRa-toestellen beheert en beveiligt.
- Uw applicatie die de data interpreteert en presenteert

Voor sensoren bestaan er 3 klassen:

- Klasse A: sensoren van deze klassen zullen alleen maar data versturen en niets ontvangen. Deze sensoren moeten enkel actief zijn als de data moet verstuurd worden, tussen 2 zendingen staat de sensor in slaapstand. Op deze manier verbruiken deze sensoren zeer weinig energie, waardoor ze op batterijen jaren kunnen meegaan.
- Klasse C: sensoren van deze klasse dienen enkel om commando's te ontvangen, om deze door te sturen naar het toestel dat met de sensor is verbonden. Ze sturen zelf niets terug. Deze sensoren moeten altijd aan staan om op eender welk moment commando's te kunnen ontvangen. Daardoor worden ze meestal aangesloten op het elektriciteitsnet.
- Klasse B: deze klasse is een tussenweg tussen de 2 andere klassen.

LoRa-gateways verzorgen de radiocommunicatie met de LoRa-sensoren. Een gateway is verbonden met de server via Ethernet, WiFi of 3G/4G. Een gateway bevat een zekere hoeveelheid kanalen. Het aantal kanalen bepaalt hoeveel IoT-sensoren er tegelijk hun data naar de server kunnen sturen. Omdat de sensoren weinig tijd nodig hebben om de data te versturen, kunnen er over een periode van een uur enorm enkele duizenden sensoren verbonden worden met het netwerk.

De netwerkserver beheert de gateways die verbonden zijn met het LoRa-netwerk. Vanuit de server kan bepaald worden over welke frequentie uit de ISM-band de sensoren hun informatie versturen en de gateways moeten luisteren om deze informatie te ontvangen. De server verzorgt ook de beveiliging zodat de data die de sensoren versturen versleuteld wordt.

De applicatie die de eindgebruiker raadpleegt slaat de informatie op en zet deze om in een visuele presentatie. Via de app kan u ook communiceren met de sensoren en gateways binnen uw netwerk. Deze applicaties worden meestal op maat gemaakt door IT-bedrijven.

4.3 Beveiliging

LoRaWAN is 1 van de weinige IoT-protocollen dat zijn data encrypted vanaf end-device helemaal tot aan de server. Normaal wordt alleen het gedeelte tussen end-device en gateway geëncrypteerd en wordt het tussen gateway en server als plain text verstuurd. Van de eindgebruiker wordt dan verwacht om een extra security layer te implementeren, maar deze extra laag vereist meer energie en complexiteit dan voor LPWAN-netwerken nodig is.

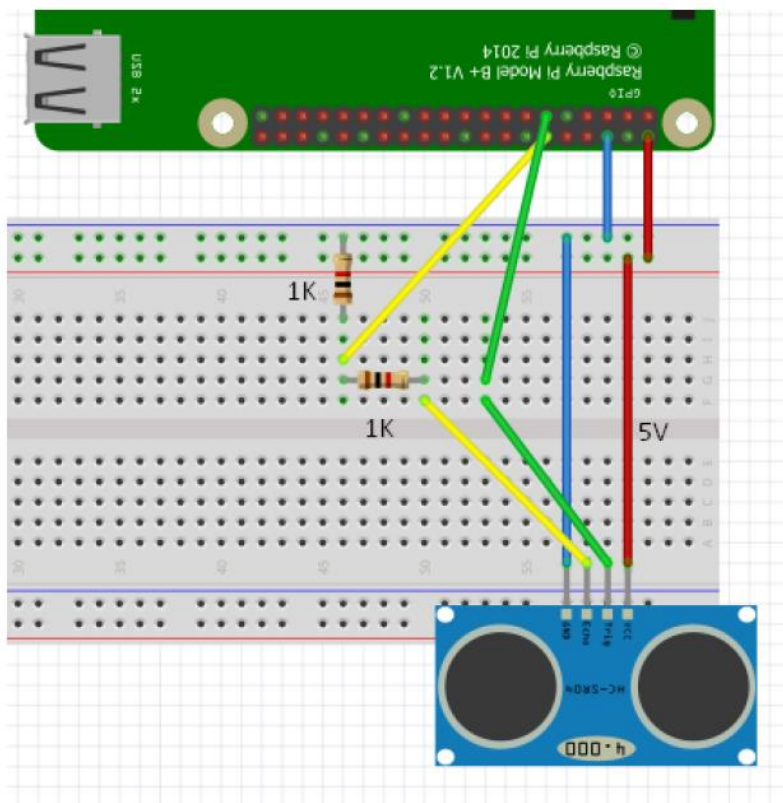
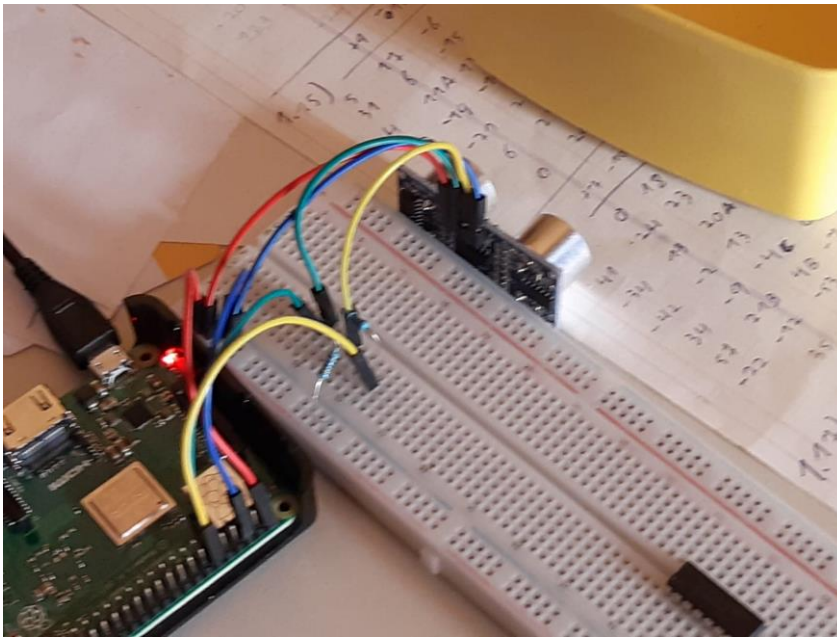
LoRaWAN gebruikt AES als basis voor zijn encryptie-algoritmes om alle data te versleutelen. Elk apparaat heeft een eigen 128-bit sleutel en een 24-bit unieke ID-code voor authenticatie.

Om authenticatie tussen end-device en server te beveiligen, wordt er een gemeenschappelijke sleutel afgesproken tussen beiden: de AppKey. Vanuit deze AppKey worden 2 nieuwe sessie-sleutels berekend die allebei de data zullen versleutelen. In elk pakket zit ook een teller (om replay-aanvallen te voorkomen) en een Message Integrity Code (zodat het pakket niet kan aangepast worden door hackers).

Standaard worden alle sleutels bijgehouden op het toestel zelf, wat betekent dat gebruikers in acht moeten nemen om hun end-devices zelf nog fysiek te beveiligen.

5 Opstelling op afstand

Om een silo te kunnen simuleren hebben we een opstelling met een raspberry pi geconnecteerd aan een breadboard zoals hieronder te zien is. Om de silo te simuleren moet men dus contact maken met de pi om zo het volume aan te passen. Hierdoor kan men alle volumes nabootsen en zo ook testen of alles inderdaad werkt.



6 Gebruikte componenten

6.1 Raspberry pi 3b+



6.2 Ultrasonische sensor



7 Data opslaan in database

De verkregen data wordt opgeslagen in een database op phpMyAdmin via onderstaande code. Hierdoor kunnen we de data visualiseren door de gegevens uit deze database op een site uit te lezen. Onderstaande code toont hoe wij dit mogelijk gemaakt hebben.



Bringing MySQL to the web

```
def zendNaarDb():
    locatie = 'Kleinhoefstraat 4, 2440 Geel'
    tijdstip = datetime.now()
    afstand = round(readAfstand(),2)
    percentage = round((afstand/20)*100,1)
    if percentage > 100:
        percentage = 100
    cursor.execute("INSERT INTO SiloVolumeMeting(tijdstip, volume, locatie) VALUES(%s, %s, %s)", (tijdstip,percentage,locatie))
    database.commit()

#Verbinden met de database
database = MySQLdb.connect(host="localhost", user="pi", passwd="ArbeitenBitte!",db="mydb")

#database select
cursor = database.cursor()

schedule.every(1).minutes.do(zendNaarDb)
```

8 Data visualiseren

De visualisatie gebeurt aan de hand van een site waarop men een horizontale representatie te zien krijgt van het volume van de silo. Dit is te bekijken op <https://r0743766.sinners.be/index.html>. Hieronder kan u de gebruikte code bekijken.

```

import RPi.GPIO as GPIO
import time
from datetime import datetime
from email.mime.text import MIMEText
import time
import sys
from pubnub.pnconfiguration import PNConfiguration
from datetime import datetime
from pubnub.pubnub import PubNub
from pubnub.callbacks import SubscribeCallback
import cgitb ; cgitb.enable()
import spidev
import os
import sys
import MySQLdb
import smtplib
from datetime import timedelta
from datetime import datetime
import schedule
import time

GPIO.setmode(GPIO.BCM)
GPIO.setup(18,GPIO.IN)
GPIO.setup(17,GPIO.OUT)
|
pnconfig = PNConfiguration()
pnconfig.subscribe_key = 'sub-c-93f51b64-e111-11e9-87e1-4690e5f445a1'
pnconfig.publish_key = 'pub-c-262e04e4-44a9-4e58-b15e-a3725d2a282a'
pubnub = PubNub(pnconfig)

def readAfstand():
    tijdstip = datetime.now()
    GPIO.output(17, 1)
    time.sleep(0.00001)
    GPIO.output(17, 0)
    while (GPIO.input(18) == 0):
        pass
    signalhigh = time.time()

    while (GPIO.input(18) == 1):
        pass
    signallow = time.time()
    timepassed = signalallow - signalhigh
    distance = timepassed * 17000
    return distance

```

```

while True:

    locatie = 'Kleinhoefstraat 4, 2440 Geel'
    tijdstip = datetime.now()
    maxdistance = 20
    afstand = round(readAfstand(),2)
    percentage = round((afstand/20)*100,1)
    emailSend = False

    if percentage > 100:
        percentage = 100

    schedule.run_pending()

    while percentage < 30:
        schedule.run_pending()
        if percentage <= 20:
            Alert = 'Silo volume kleiner dan 20% !'
            pubnub.publish().channel("alert").message(Alert).sync()

        if emailSend == False:
            Alert = 'Silo volume kleiner dan 30% !'
            bericht = 'mail verstuurt naar centrale!'
            pubnub.publish().channel("alert").message(Alert).sync()

            msg = MIMEText(Alert)
            msg['Subject'] = 'RPi Python test'
            msg['From'] = 'raspberrypie@gmail.com'
            msg['To'] = 'brent.devos1999@gmail.com'

            # send the email via Gmail server
            username = 'raspberrypie@gmail.com'
            password = 'Rasperry888'
            server = smtplib.SMTP('smtp.gmail.com:587') # Gmail rewriting port 25 to port 587
            server.starttls()
            server.login(username,password)
            server.sendmail(msg['From'], msg['To'], msg.as_string())
            server.quit()
            emailSend = True
            pubnub.publish().channel("bericht").message(bericht).sync()

        afstand = round(readAfstand(),2)
        percentage = round((afstand/20)*100,1)
        pubnub.publish().channel("adc").message(percentage).sync()
        time.sleep(0.5)

    if percentage <= 20:
        Alert = 'Silo volume kleiner dan 20% !'
        pubnub.publish().channel("alert").message(Alert).sync()
        print('alert send')

    if percentage > 30:
        Alert = ''
        bericht = ''
        pubnub.publish().channel("alert").message(Alert).sync()
        pubnub.publish().channel("bericht").message(Alert).sync()

    print(percentage)
    print(afstand)

    pubnub.publish().channel("adc").message(percentage).sync()

    time.sleep(0.5)

GPIO.cleanup()

```

9 Besluit

In dit verslag hebben wij aangetoond hoe wij via IOT een volumemeting van een silo gesimuleerd hebben. Dit project toont aan dat IOT niet enkel zijn nut kent in IT-oplossingen, maar ook in een bredere context handig kan zijn. Zo zou dit project, indien echt uitgewerkt, een zeer handige tool zijn voor de bouwsector.