Practice Enterprise

House Alarm System

1 Inleiding

Allereerst zouden we willen de school bedanken voor dit afgelopen jaar. Het heeft ons de kans gegeven om ons te verdiepen in programmeertechnieken. Via dit project willen we tonen wat we hebben kunnen realiseren met de opgedane kennis.

Daarnaast willen we ook ouders en vrienden bedanken voor de ondersteuning tijdens dit, toch wel woelige jaar.

Het project dat we voorstellen is zeker nog voor verbetering en uitbreiding vatbaar maar het heeft ons toch wat tijd en moeite gekost om de verschillende onderliggende technieken onder de knie te krijgen. Veel code werd enkele keren herschreven om tot een werkend geheel te komen.

We zijn erin geslaagd een minimaal, maar werkend, alarmsysteem op te zetten. Op basis van de toegepaste technieken zou dit alarmsysteem nog heel wat uitgebreid kunnen worden, maar de minimaal vereiste bouwstenen zitten erin vervat.

Misschien een kort woordje over onszelf. Wij zijn Siebe Demetser (siebedemetser@gmail.com) en Arnoud Serruys (arnoud.serruysotmail.com) en wij hebben als project gekozen om een huisalarm te maken. Het alarm is gebaseerd op een Raspberry Pi 4 en er werd webtechnologie en databank technologie gebruikt voor zijn opzet.

Het project zou in staat moeten zijn om te functioneren als een modern alarmsysteem. Door middel van een web interface kan je gegevens vragen van de alarmgeschiedenis via smartphone of webbrowser. Je kan ook via de website zonder veel moeite bepaalde sensoren en de configuratie van het systeem aanpassen.

2 Hardware

2.1 Hardware

Het project bestaat uit een raspberry pi en enkele sensoren. We hebben een bewegingsdetector, een geluid sensor en een reedcontact. We hebben als voorbeeld van elke sensor 1 aangesloten. Maar er kunnen natuurlijk meer sensoren bijgeplaatst worden zolang er I/O pinnen vrij zijn.

2.2 Raspberry pi

Voor het project hebben wij gekozen voor een raspberry pi 4 met 2 Gigabyte RAM. Deze hebben we voorzien van een aluminium behuizing met ventilatoren. Dit was nodig aangezien we enkele warmte problemen hadden tijdens het aanmaken van de database.

2.3 bewegingsdetector

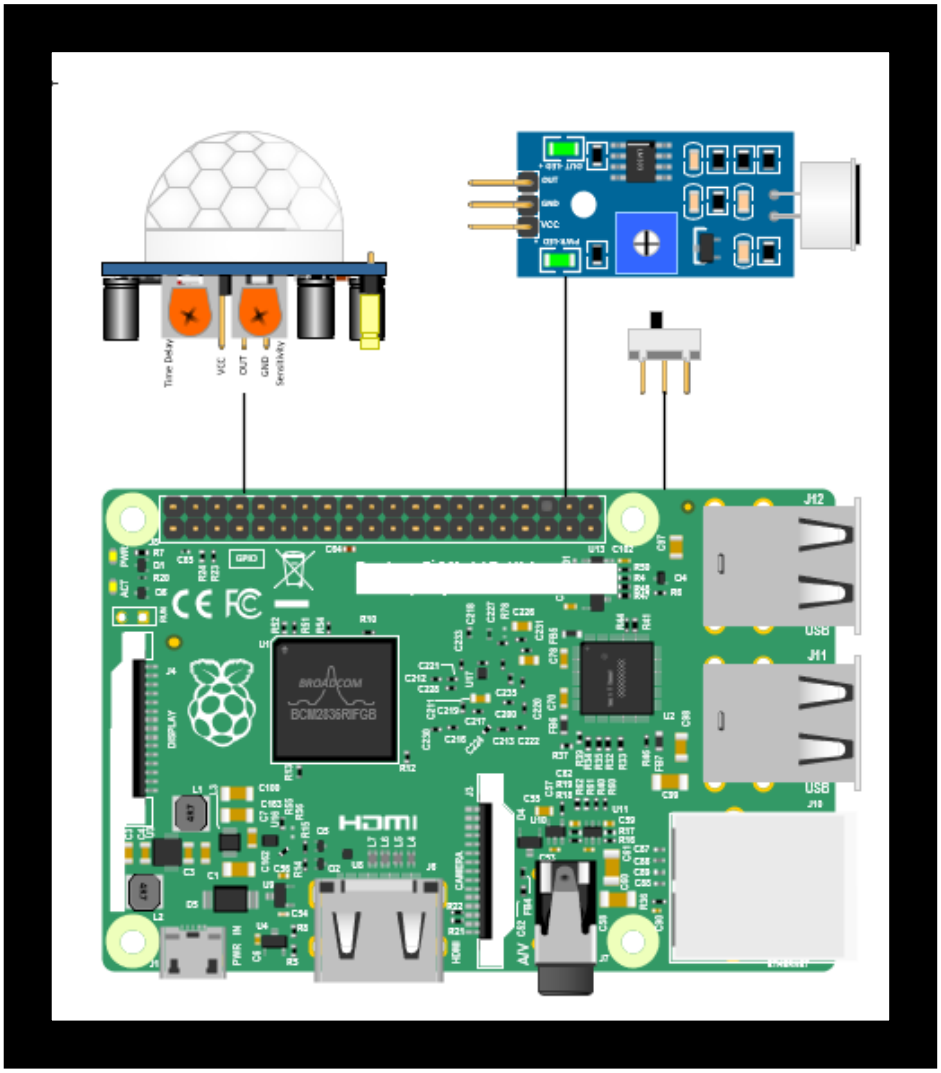
We hebben een PIR (Passief InfraRood) bewegingsdetector voorzien. Deze kan veranderingen in de omgeving waarnemen. Door het detecteren van de infra rood stralen die worden uitgezonden door de omgeving. Elk voorwerp met een temperatuur boven het absolute nulpunt zend infrarood stralen uit dus ook wij mensen. Hoe warmer iets is hoe sterker de straling. Deze straling kunnen wij niet zien met het blote oog maar wel met een infra rood camera. Onze sensor is slechts een simpele sensor en kan dus enkel verandering detecteren en niet de straling in kaart brengen.

2.4 Glasbreuk sensor

Dit is slechts een simpele geluidssensor. Deze werkt op ongeveer dezelfde manier als een microfoon. Door de trillingen van het geluid om te zetten in een elektrisch signaal. We hebben deze sensor ingesteld zodat deze enkel signaal geeft bij een luide slag.

2.5 Reedcontact

Tot slot hebben we ook nog ons reedcontact (wordt op het schema weergegeven als een schakelaar). Het reed contact bestaat uit 2 delen. Het eerste deel dat simpel weg bestaat uit een magneetje dat wordt gemonteerd op de deur. Het 2de deel is een NO/NC contact dat op de kader van de deur wordt geplaatst. Als het magneetje in de buurt van ons contact komt zal deze schakelen en een signaal doorlaten.


Time Delay
VCC
OUT
GND
Sensitivity
OUT
GND
VCC
BCM2836RIFGB
HDMI
USB
USB

3 Software

De sofware voor dit project kunnen we functioneel opsplitsen in twee delen: deel 1 verzorgt de afhandeling van de alarmen die via de GPIO van de raspberry pi verzameld worden, deel 2 omvat een web applicatie om aan de gebruikers van het alarmsysteem een interface aan te bieden.

Het project van het alarm is geschreven in de 'python programming language' en ook een stukje javascript voor de web applicatie. We gebruiken voor het project enkele bibliotheken die we in de volgende paragrafen toelichten. De voornaamste bibliotheek is Flask, een framework voor de ontwikkeling van web applicaties en API's. Omdat Flask standaard Sqlite3 als database ondersteunt hebben we ook voor het alarmsysteem voor deze lightweight database gekozen.

Voor de aanmaak van de structuur van de database hebben we beroep gedaan op een tool 'DB Browser'. Dit laat toe om via een grafische interface de database te structuren te analyseren.

3.1 Flask

Flask is een micro web framework voor python, hiermee wordt er bedoelt dat Flask een library is waarmee je in python makkelijker een web applicatie of web API kan aanmaken. De library heeft vele functies waarmee je, met gebruik van diverse html templates, een eigen webpagina maakt. Flask heeft ook functies voor de veiligheid van je web applicatie zoals gebruikers en gebruikersrollen. Deze rollen laten toe web pagina's of API oproepen selectief te beantwoorden. Als je als gebruiker een bepaalde rol voor een functie ontbreekt wordt de functie niet aangeroepen.

3.2 SQLite 3

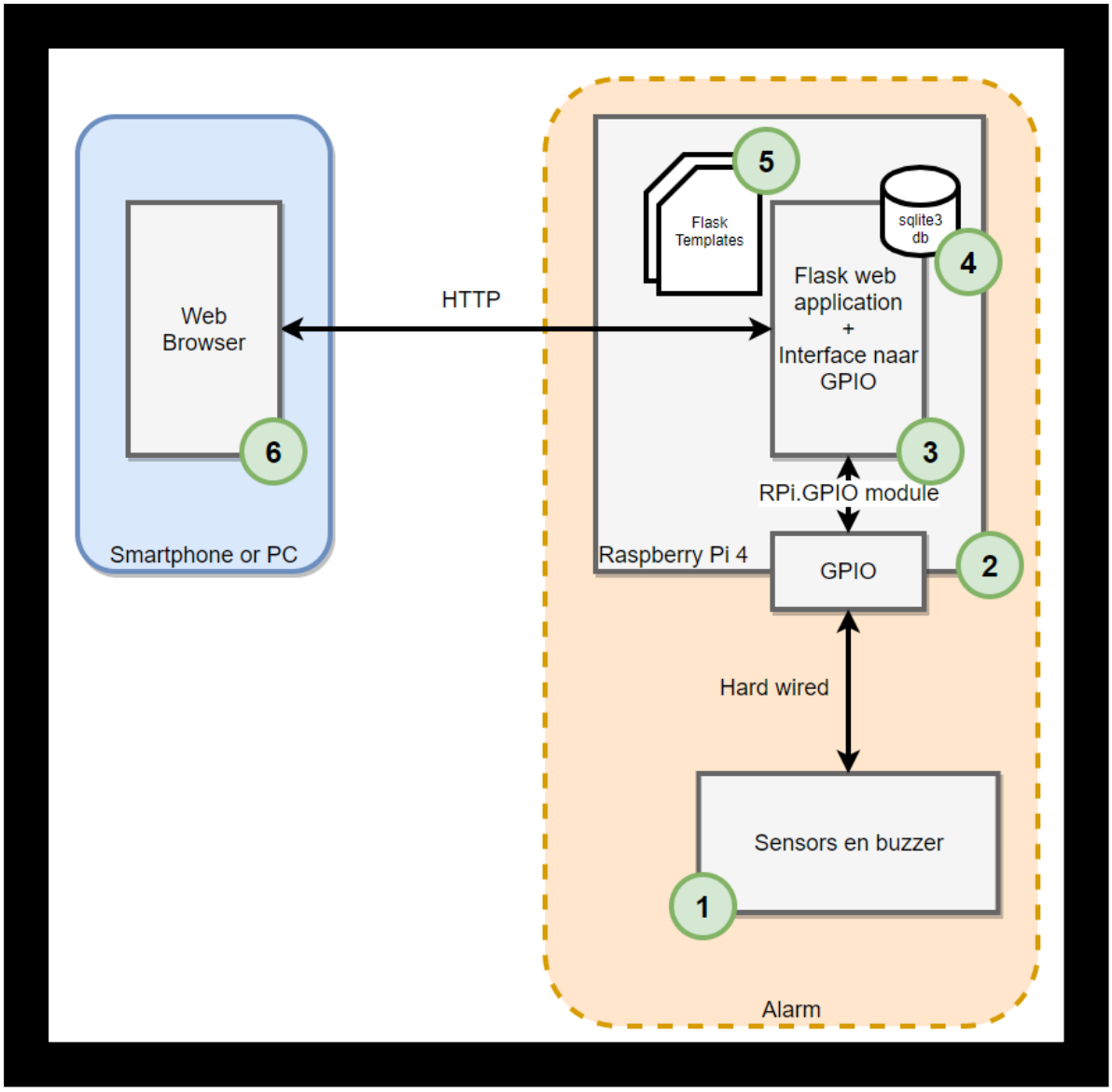
SQLite 3 is een databank library die standaard in python aanwezig is. Het behoeft geen engine zoals andere database varianten (MySQL …). De complete database wordt ondergebracht in een enkelvoudig bestand op het filesysteem. De werking vanuit de code is gebaseerd op standaard SQL wat het ook zou mogelijk maken om via enkele eenvoudige aanpassingen andere database technologie te gebruiken. In ons project gebruiken we de database voor twee doeleinden. Vooreerst worden hier de alarmen, afkomstig van de sensoren, geplaatst. Maar ook gebruikers en rollen worden in aparte tabellen in dezelfde database ondergebracht. Bij dit project is de databank op de raspberry pi het centrale gedeelte van het project. De code voor de afhandeling van de sensor informatie schrijft naar de database terwijl de functies vanuit de web interface via dezelfde database uitgelezen worden.

3.3 DB Browser

DB Browser is de software die we hebben gebruikt om makkelijk een databank te maken in sqlite3 vorm. Deze databank hebben we dan overgezet naar de Raspberry pi, de databank heeft 6 tabellen: alarms, config, sensors, rooms, users en roles. In deze tabellen staat alle data over de alarm installatie.

3.4 Het Alarm

3.4.1 Blokdiagram



Dit is het blokdiagram van het alarm. We overlopen even de verschillende onderdelen.

1) Sensoren en buzzer

Hierin is de bedrading van het alarm zelf, alle sensoren en ook de buzzer, er zijn drie soorten sensoren op dit moment: movement sensor, sound sensor en reed sensor. Het aantal sensoren kan uitgebreid worden. De web interface voorziet de mogelijkheid om sensoren aan de configuratie toe te voegen.

2) De GPIO interface van de Raspberry pi

In de GPIO worden alle pinnen van de sensoren en de buzzer uit de databank gelezen om dan te initializeren. De GPIO setup die we gebruiken is de BOARD mode waarin we de exacte pin nummering gebruiken van 1-40 i.p.v. de BCM mode die namen gebruikt (zoals GPIO16,…).

3) Het centrale proces

Dit is waar de main uitgevoerd wordt die de GPIO linkt aan de web application, de base functies van het opstellen van het alarm en de data wegschrijven naar de databank als het alarm afgaat. Dit is ook de plaats waar de API webrequests toekomen die dan een functie opstarten of een aanvraag doen om naar een volgende webpagina/template te gaan. De functies zijn er om bevoorbeeld: extra sensoren toe te voegen, het alarm aan en uit te zetten, een lijst van vorige alarm activaties op te vragen, enzovoort.

4) De Sqlite3 database

De database is de plaats waar alle data wordt bewaard: sensoren, kamers, de samenhang tussen sensor en kame ren de lijst van alle alarm activaties maar ook de users en roles van de web applicatie, de bedoelingen van roles zijn om authorisatie te verdelen. De main code gebruikt de databank constant om data uit te wisselen.

5) De HTML templates voor de web interface

De templates zijn html files die de opmaak bevatten van de web applicatie, bij elke request wordt er een template voorzien met de juiste informatie van de databank die werd opgevraagt of een template met velden om in te vullen zodat je in de databank kan schrijven. De templates worden dynamisch opgebouwd door Flask, er is een base.html file waar alle andere templates van afgeleid worden. Deze templates voegen toe aan de pagina enkel wat er specifiek is voor die pagina.

6) De browser op PC of smartphone

Dit is de web interface die altijd opgemaakt is uit base.html en een uitbreidende template. In de base.html zijn er altijd 6 knoppen aanwezig: index, config, profile, register, login en logout, deze laten u toe om te navigeren door de applicatie. Een stukje client-side javascript zal om de 2 seconden de toestand van het alarm ondervragen en de status van het alarm in de webpagina weergeven. Zowel de armed/disarmed informatie als de informatie dat het alarm geactiveerd werd staat onderaan iedere webpagina. Er wordt eveneens een audio signaal weergegeven. (de audio aanpak wordt jammer genoeg niet ondersteund in safari mobile).

3.4.2 De code

Het project bestaat uit verschillende onderdelen. Ieder onderdeel werd ondergebracht in een aparte python module (auth.py, app.py, main.py, raspberry.py). Elke module implementeert 1 stuk functionaliteit van het alarmsysteem.

3.4.2.1 De startmodule: App.py

App.py is de hoofdmodule. Wanneer Flask opstart is dit de eerste module die wordt uitgevoerd. Het staat in voor de opzet van de basis klassen, zoals user en rol. Het zal eveneens de GPIO en de sensoren initialiseren op basis van de gegevens die uit de database opgehaald worden.

De sleutels voor het veilig opslaan van gebruikerspaswoorden wordt ook hier geïnitialiseerd.

De database wordt hier opgezet en zal globaal ter beschikking staan voor de andere modules

Flask gebruikt een systeem van HTML templates en blueprint modules. De registratie van deze gebeurt eveneens in de app.py module.

3.4.2.2 De module voor authenticatie en gebruikersregistratie: Auth.py

In deze module worden de verschillende web url's geïmplementeerd voor het afhandelen van de security aspecten. Registratie van nieuwe users, login en logout en een zeer summiere gebruikers profiel pagina. Het login scherm is tevens het eerste scherm waarnaar de applicatie verwijst omdat alle ander pagina's slechts toegankelijk zijn voor geregistreerde gebruikers.

3.4.2.3 De module voor de implementatie van de web interface en API: Main.py

Main.py heeft alle andere web url's van de web applicatie, hierin zijn er de routes om de sensorconfiguratie aan te passen of om de alarm historiek uit te lezen. In Main.py staat er ook een getstate() functie die om de 2 seconden aangeroepen wordt door de web applicatie. Dit om in de webinterface de huidige toestand van het alarm te kunnen visualiseren. Ieder scherm van de webinterface toont trouwens twee iconen met de hoofdtoestand van het alarm, namelijk of het alarm actief staat ('armed') en of het alarm getriggered werd door een sensor. Dit laatste resulteert eveneens in een hoorbaar sirene signaal.

3.4.2.4 De module voor het afhandelen van de sensoren via GPIO

Hierin worden alle pin activaties/deactivaties gecodeerd, alle pinnen die worden gebruikt zijn uit te lezen uit de databank. Elke pin wordt in een event_state gezet waardoor, als ze geactiveert worden, wordt er een bepaalde functie uitgevoerd. Deze functie wordt alleen maar gebruikt als het alarm gearmed is anders worden de sensoren genegeerd. Bij de gebeurtenis van een gearmed alarm zal deze functie ervoor zorgen dat alle data van de activatie worden verzonden naar de databank, nadat dit gebuirt is zal het alarm beginnen biepen en ook de website zal geluid maken als je het open hebt staan, om het alarm af te zetten moet je enkel de arm knop indrukken op de website. In deze code staan ook alle functies in verband met het aanmaken en wissen van de sensoren of kamers of het verband te veranderen tussen de kamers en sensoren. De functies, om alles uit de databank te lezen, staan hier ook in.

3.4.3 De templates

De templates van de web applicatie zijn allemaal gemaakt in html. De base.html is de basis van alle andere templates, in dit template staan de 6 basis optie knoppen, en ook de twee iconen om te laten zien of het alarm gearmed is en/of actief staat. De andere templates worden opgeroepen door bepaalde knoppen in te drukken op de webpagina. De opmaak van deze templates zit in een constante css file die we niet zelf hebben gemaakt, maar van Github hebben gedownload (! bulma.io v0.7.2 | MIT License | github.com/jgthms/bulma). De andere templates hebben we zelf gemaakt en ons gebaseerd op een basis voorbeeld dat we hadden gevonden

4 Conclusie

Ondanks al onze kopzorgen, verzuchtingen, kleine en minder kleine tegenslagen, vinden we dat we in onze opzet geslaagd zijn. Ook al hadden we oorspronkelijk iets meer functionaliteit willen inbouwen in het alarmsysteem denken we toch de basis gelegd te hebben van een systeem die zou kunnen uitgebouwd worden tot een volwaardig alarm. User en rolbeheer zit er bijvoorbeeld in maar de configuratie via de web interface is er niet meer in geraakt.

5 Bijlagen

De code van het project werd op GitHub geplaatst. Volgend link brengt je naar de correcte repository.

Github House Alarm Link