

Groep: Covfefe

Kadeem Vers

Marc Zoomers

Niels Risseeuw

Max Bosch

Zoheb Javaid

Verslag CSN

Hogeschool Utrecht | V1O

2017

Inhoudsopgave

[Inleiding 2](#_Toc496886504)

[Deelvragen 2](#_Toc496886505)

[Hoofdvraag 2](#_Toc496886506)

[Werkverdeling 3](#_Toc496886507)

[Diagrammen 4](#_Toc496886508)

[Technische beschrijving van het systeem 6](#_Toc496886509)

# Inleiding

Voor dit project is het de bedoeling dat onze groep een inbraakalarmsysteem creëert doormiddel van meerdere RPI’s met elkaar te verbinden. De ene Pi heeft als taak een alarmsysteem na te bootsen en de ander een alarmcentrale. De functionele code wordt geprogrammeerd in in Python met behulp van Tonny.

In dit verslag wordt uitgelegd hoe wij planmatig te werk zijn gegaan met het ontwerpen en realiseren van het inbraakalarmsysteem, welke taken er door wie is uitgevoerd en wanneer deze zijn uitgevoerd.

# Deelvragen

* **Welke hardware is beschikbaar?**

De beschikbare hardware bestaat uit twee Raspberry Pi 3B modellen, UTP kabels, een USB webcam, HDMI kabel, Twee Monitors, Speaker, Led lichtjes en knopjes.

* **Welke functies hebben verschillende onderdelen?**

USB Camera om foto’s te maken, Rapsberry client als alarm, Raspberry Server als alarmcentrale, LED voor licht, Speaker om geluid af te spelen van het alarm.

* **Op welke wijze kan er een koppeling worden gelegd door hardware en software?**

De hardware legt een koppeling met de software doormiddel van een Python code en verbinding tussen beide Pi’s wordt uitgewisseld doormiddel van een UTP kabel.

* **Wat is het hardware ontwerp dat nodig is om het proof-of-concept te realiseren?**

Het hardware ontwerp is gemaakt met Fritzing.

* **Wat is het software ontwerp dat nodig is om het proof-of-concept te realiseren?**

Het software ontwerp is gemaakt in een flow chart diagram.

* **Op welke wijze kunnen een client en een server (een gedistribueerd systeem) met elkaar communiceren?**

De client en server zijn beiden op twee aparte Raspberry pi’s geprogrammeerd om met elkaar te comminuceren via een UTP kabel connectie.

* **Op welke wijze kan worden aangetoond dat het proof-of-concept laat zien dat er een alarmsysteem kan worden gerealiseerd?**

Het feit dat je het met RPI’s een alarmsysteem kan maken, laat zien dat je het ook op grotere schaal kan toepassen.

# Hoofdvraag

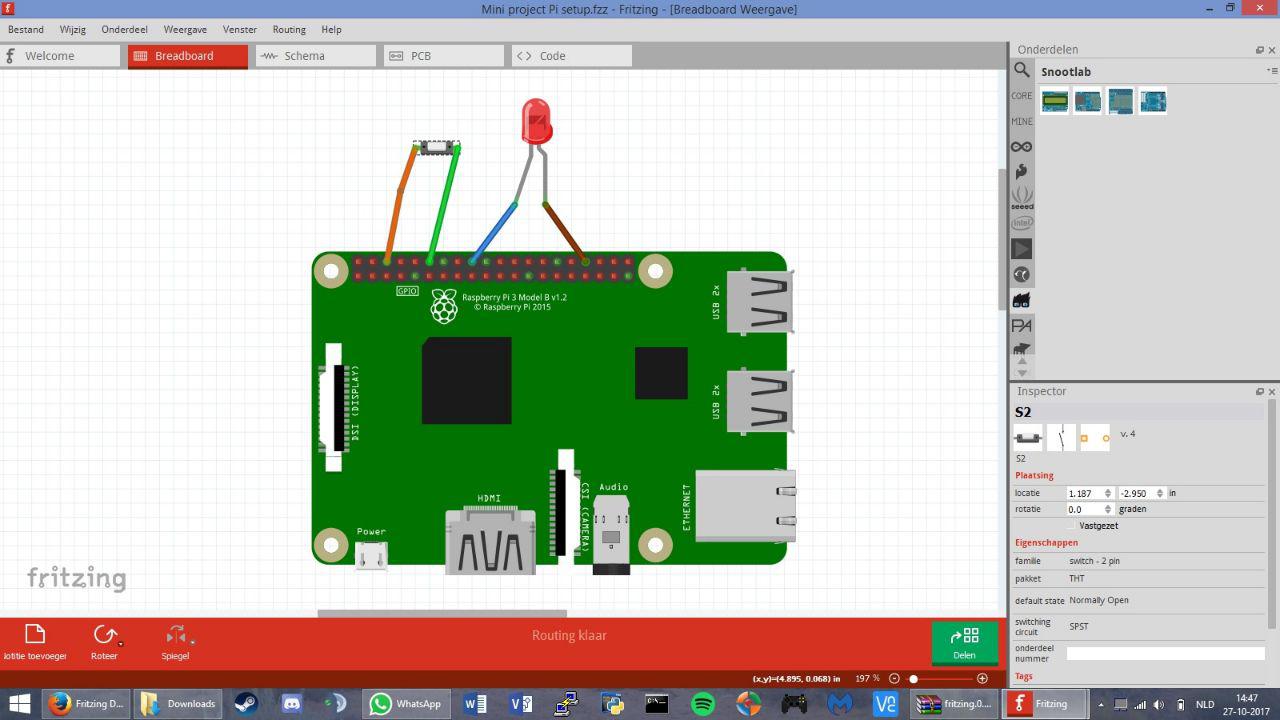
**Welk proof-of-concept van een hard- en software oplossing dient te worden geïmplementeerd om te laten zien dat door middel van de geleverde hardware een gedistribueerd alarmsysteem gemaakt kan worden?**

Het alarmsysteem en alarmcentrale(client-to-server) moet een werkende detectie en afslaan mode hebben. Met verificatie/communicatie van de centrale.

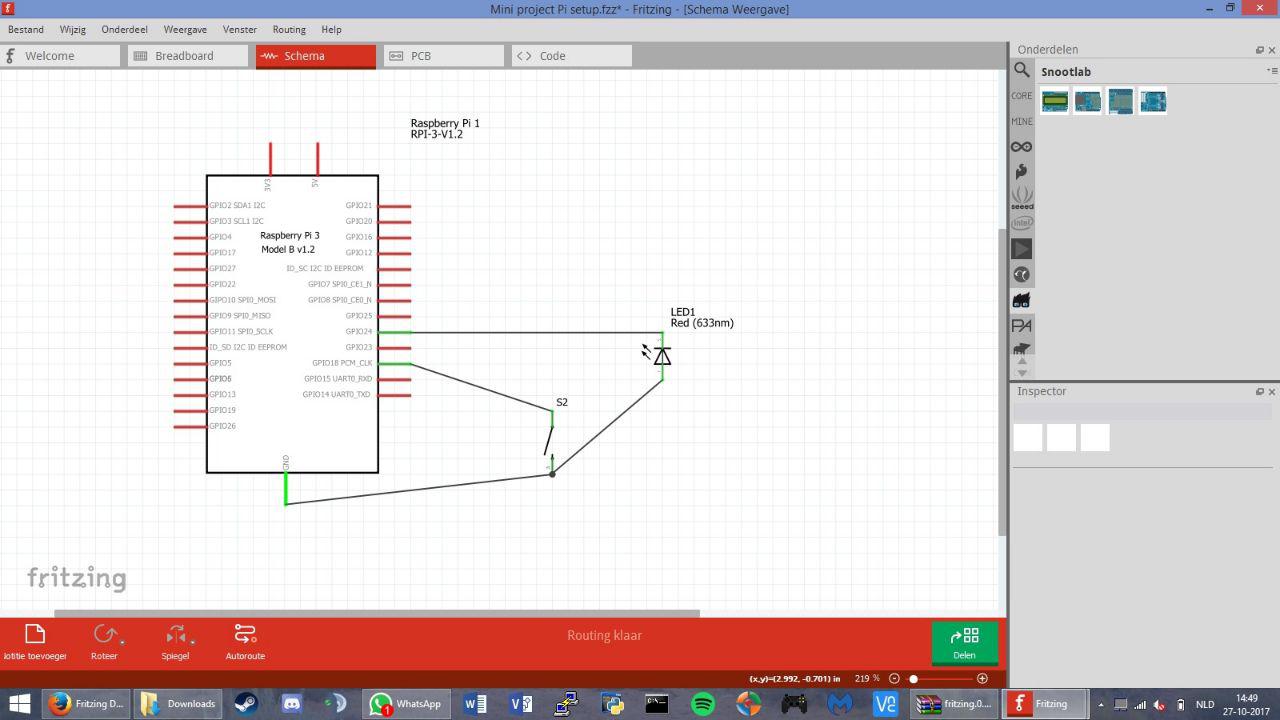
# Werkverdeling

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 23-10-17 | 24-10-17 | 25-10-17 | 26-10-17 | 27-10-17 |
| Marc Zoomers | Start cameracode,  Projectgroep mail gemaakt,  Bashscript.sh gemaakt | Implementeren van geluidscode,  alle mail berichtten en andere meldingen vertalen. Fixen van taalkeuze loop | Camerascript.py op andere pi laten werken | Camerscript.py met het client.py script combineren | De schematische lay-out van de GPIO maken. |
| Zoheb Javaid | Research pi 2 pi communicatie | Aanmaken van server en client | Communicatie tussen client en server | Client en server werkend krijgen met gehele code | Documentatie |
| Niels Risseeuw | Research pi 2 pi communicatie en hulp met coderen. | Research, coderen op Python en implementeren van geluidscode | Camerascript.py op andere pi laten werken en aan client code werken. | Maken en editen van video, afmaken code van de client. | Editten video toevoegen extra materiaal. |
| Kadeem Vers | Github repo gemaakt | Aanmaken van server en cliënt | Communicatie tussen cliënt en server | Client en server werkend krijgen met gehele code |  |
| Max Bosch | Datum bugfix |  | Geraamte van verschillende talen coderen in Python | Bugfixes in talen verbeteren, cliënt script bugfix, Video records gemaakt | Master code opschonen, video records gemaakt |

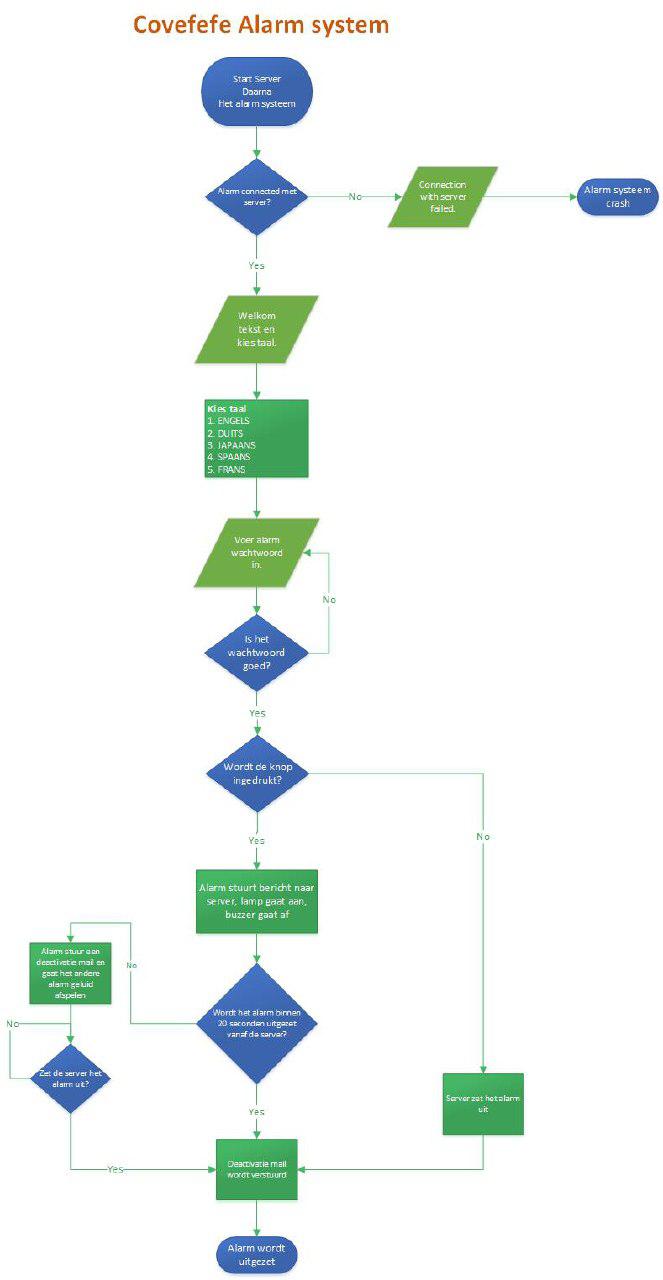
# Diagrammen



Schematische tekening van de RPI



Technische tekening van de RPI



Flow chart diagram van alle statussen.

# Technische beschrijving van het systeem

Refereer naar de readme in Github.