Handleiding project Computer Systems and Networks

Klas TICT-ICT-V1O, 11-11-2016

Siham Bouaabasslam, Koen van Burken, Mike de Jong en Benthe Klaassen

Wij hebben voor het vak Computer Systems and Networks een project gemaakt en we hebben de opdracht over het inbraakalarm op onze PI3 gemaakt. De bedoeling van deze opdracht is om ervoor te zorgen dat er een inbraakalarm wordt gerealiseerd op de Raspberry Pi met hulp van het experimenteer bord. Zodra er wordt ingebroken moet het alarm afgaan en die blijft net zo lang afgaan tot dat het juiste wachtwoord wordt ingevoerd in het systeem waarmee het alarm ongedaan kan worden gemaakt. Hierna volgt de vraag aan de eigenaar of de snelheid kan worden aangepast van het alarm (knipperen lang of kort). In volgende tekst beschrijven wij hoe we dat samen hebben gedaan.

Er zijn een paar benodigdheden om dit project zowel uit te kunnen voeren als testen, namelijk:

1. **De Raspberry Pi.** Wij hebben gebruik gemaakt van de Pi 3B. De Raspberry moet beschikken over een aantal **softwareprogramma’s**:

* Python
* Nano tekstbewerker
* Putty

Maar ook over een aantal **hardware onderdelen**:

* SD Kaart
* Voeding
* Ethernetkabel

1. Daarnaast hebben we ook gebruik gemaakt van de onderdelen van het **‘experimenteer bord’**:

* Een rood LED-lichtje
* Een geel LED-lichtje
* Buttons
* Kabeltjes
* Weerstand

1. Ook moet je in het bezit zijn van een scherm/laptop. Als je gebruik maakt van een laptop sluit je de ethernetkabel ook aan, om zo gebruik te kunnen maken van het beeldscherm en zo alles te kunnen bedienen.

Doordat we een inbraakalarm moesten maken hebben wij gebruik gemaakt van LED-lichtjes om op deze manier een realistisch ‘inbraak-idee’ te creëren. De bedoeling van de opdracht was dan ook ervoor te zorgen dat zodra er wordt ingebroken, er een alarm afgaat waar de dief van schrikt. Maar het kan ook zo zijn dat de eigenaar van het alarm het per ongeluk heeft laten afgaan, dan moet er natuurlijk wel de mogelijkheid zijn om deze weer uit te zetten. Het rode LED-lichtje zal blijven knipperen tot dat dit is gebeurd.

We hebben gebruik gemaakt van de buttons, omdat deze gebruikt konden worden als denkbeeldige deur. Zodra een knop wordt ingedrukt voelt dat, door middel van onze code, voor het systeem alsof er een deur wordt opengetrokken en hierna zal het alarm afgaan. Het andere knopje wordt gebruikt om de mogelijkheid op te vragen waar men zijn wachtwoord kan invullen om te bewijzen dat hij/zij de eigenaar is.

We hebben deze verbinding ook echt kunnen laten werken door een code te programmeren in Python op de Raspberry Pi. Dat is op zo’n manier gebeurd dat het lampje en de buttons los stonden van elkaar, tot dat er een knop wordt ingedrukt van de buttons. Doormiddel van kabeltjes en de weerstandjes hebben we de buttons en het lampje met elkáár verbonden, maar ook met de Raspberry Pi.

**Er zit weldegelijk een verbinding tussen de hardware- en de softwarecomponenten:**

De code die wij hebben geschreven in Python op de Raspberry Pi zorgt ervoor dat zodra deze gerund wordt dat de stroom verbinding maakt met de 40-pin header en dat deze stroom kan doorgeven aan de kabeltjes die daaraan verbonden zijn. In de code is gebruikt gemaakt van een while-loop, dit wil zeggen dat zodra we hem laten lopen, hij continu blijft controleren of het allemaal goed gaat tot het moment dat er iemand inbreekt. Zodra dat gebeurd, is er voorheen op de knop gedrukt. Met het indrukken van deze knop wordt er stroom doorgegeven aan het lampje. In de code staat geprogrammeerd dat het lampje na het indrukken aan moet blijven staan en moet blijven knipperen. Net zolang tot het goede wachtwoord is ingevuld.

Wanneer de mogelijkheid is geweest om het goede wachtwoord in te voeren en het is fout, staat er in de code dat het LED-lichtje weer moet door blijven gaan met knipperen om zo ‘paniek en stress bij de inbreker’ te veroorzaken en om aan te geven dat het alarm nog niet gedeactiveerd is.

Wanneer het wachtwoord goed is ingevoerd zal het rode knipperende LED-lichtje plaatsmaken voor een geel LED-lichtje (wat eigenlijk een groene moest zijn). Om zo aan te geven aan de gebruiker dat het wachtwoord goed is ingevuld en het alarm is gestopt.

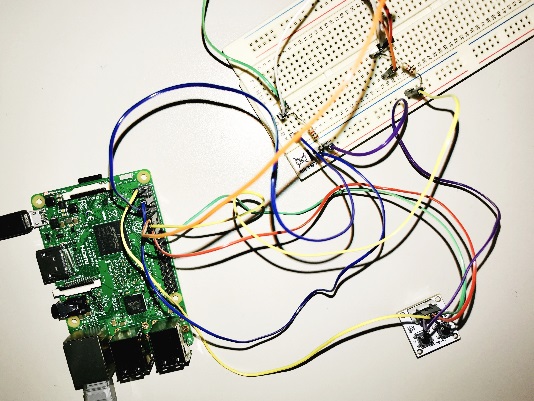
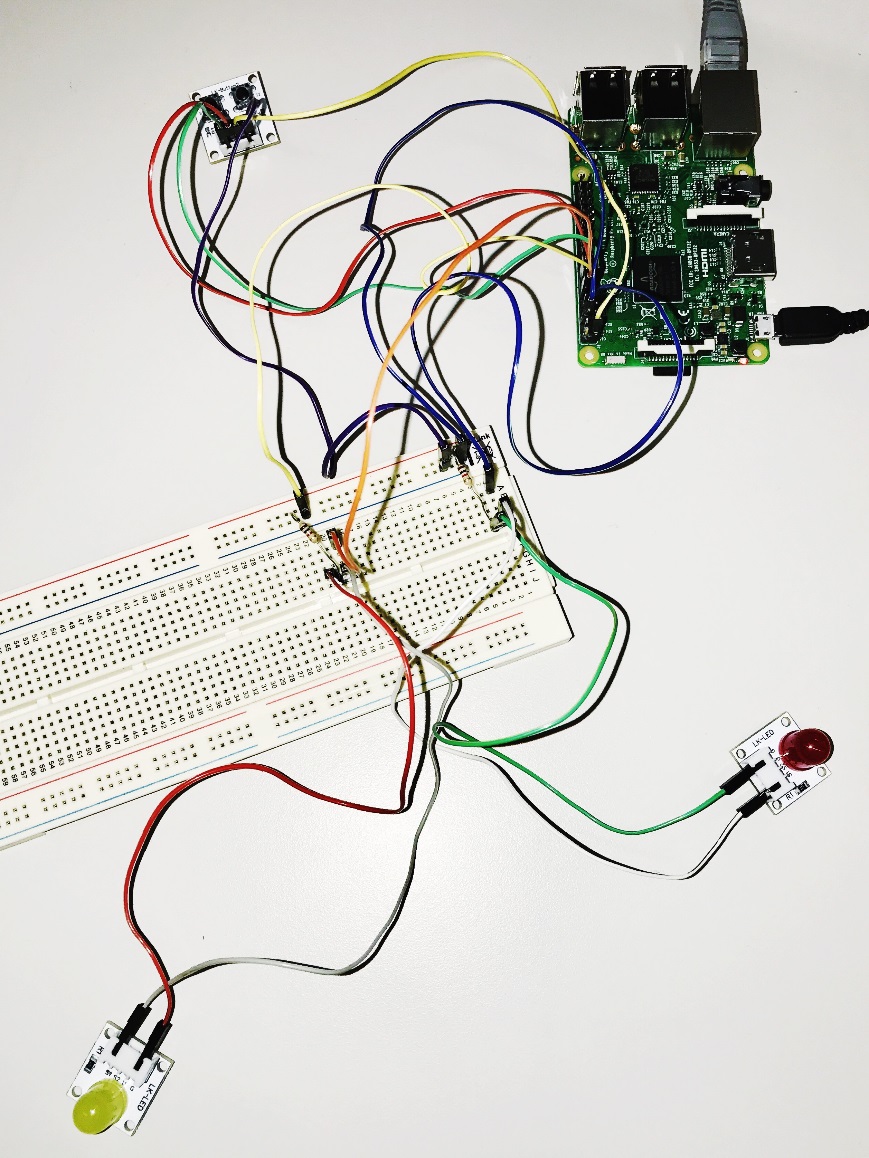
Hierna zal het programma klaar zijn en is het mogelijk om in dezelfde run weer in te breken. Het is tenslotte in het echte leven ook niet zo dat je elke keer het alarm aan en uit hoeft te zetten om het weer te activeren.

**Uitvoeren van het project**

U kunt de benodigde bestanden van onze Github halen, in onze map over dit project vanaf de volgende link: [www.github.com/koenvanburken/Mini-Project-CSN](http://www.github.com/koenvanburken/Mini-Project-CSN).

In de map Mini-Project-CSN vindt u het bestand alarm.py. Door deze te kopiëren en te plakken in een python bestand op de Pi kunt u dit project ook uitvoeren. Mits deze Raspberry Pi zit aangesloten met het experimenteer bord.

**Ons ontwerp op het experimenteer bord:**



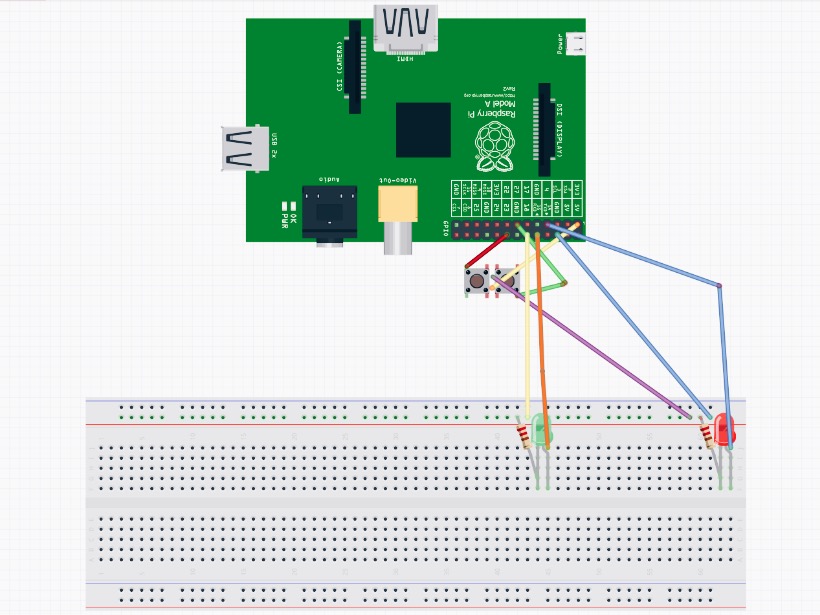
Wij hebben zeven kabeltjes vastgemaakt aan de 40-pin header.

Van boven naar beneden (beginnend bij 1, naar 40) zitten deze als volgt verbonden:

* 1: de Power, geel en is verbonden met buttons (VCC)
* 6: Ground, blauw en is verbonden met bord
* 7: GPIO 7, blauw en is verbonden met bord
* 12: GPIO 1, oranje en is verbonden met bord
* 13: GPIO 2, groen en is verbonden met buttons (S1)
* 14: Ground, geel en is verbonden met bord
* 16: GPIO 4, rood en is verbonden met buttons (S2)

De Ground van de buttons is verbonden met het bord. Doordat geen enkele kabel direct is verbonden met een lampje wordt dus duidelijk dat het lampje echt alleen kan branden/knipperen wanneer het geactiveerd wordt door ‘de deur’ of het knopje indrukken.

Hieronder ziet u ook nog een schematisch plaatje van onze opbouw van het bord. Ik heb hierboven de kabeltjes die met de pin header zijn geconfigureerd al beschreven en waartoe ze zijn verbonden. Hieronder is dat nog eens te zien samen met de kabeltjes die tot de lampjes horen.



Ik hoop dat door deze drie afbeeldingen en de nodige uitleg duidelijk is hoe wij het systeem hebben opgebouwd. Helaas waren er op het programma van het schematische plaatje geen dubbele buttons beschikbaar, daarom heb ik uitgelegd op welke plek de kabeltjes in de buttons terecht zijn gekomen.

**Gebruikte bronnen**

Wij hebben van de volgende bronnen gebruikgemaakt:

* <https://electrosome.com/led-blinking-raspberry-pi/>
* <https://www.jeremymorgan.com/tutorials/raspberry-pi/how-to-blink-led-raspberry-pi-2/>
* <https://www.youtube.com/watch?v=HhfzRn2WcDQ>
* <https://www.youtube.com/watch?v=9uot9uIC5CA>
* <http://helloraspberrypi.blogspot.nl/2015/03/readwrite-gpio-of-raspberry-pi-2-using.html>

**Mening**

Wij vonden het een uitdagend project, maar zodra we het onder de knie hadden ging het heel snel en waren we snel klaar. Het is dus een kwestie van het kwartje laten vallen. Het samenwerken ging ook goed en we hebben lange tijden op school gezeten om er samen aan te werken. Het was leuk dat er van verschillende aspecten sprake was zodat je met hard- en software tegelijkertijd bezig was.