

Smart air freshener

Vince van Noort - 6187021

Niek Geijtenbeek - 6214096

Video: <https://youtu.be/pbFeRx4UYts>

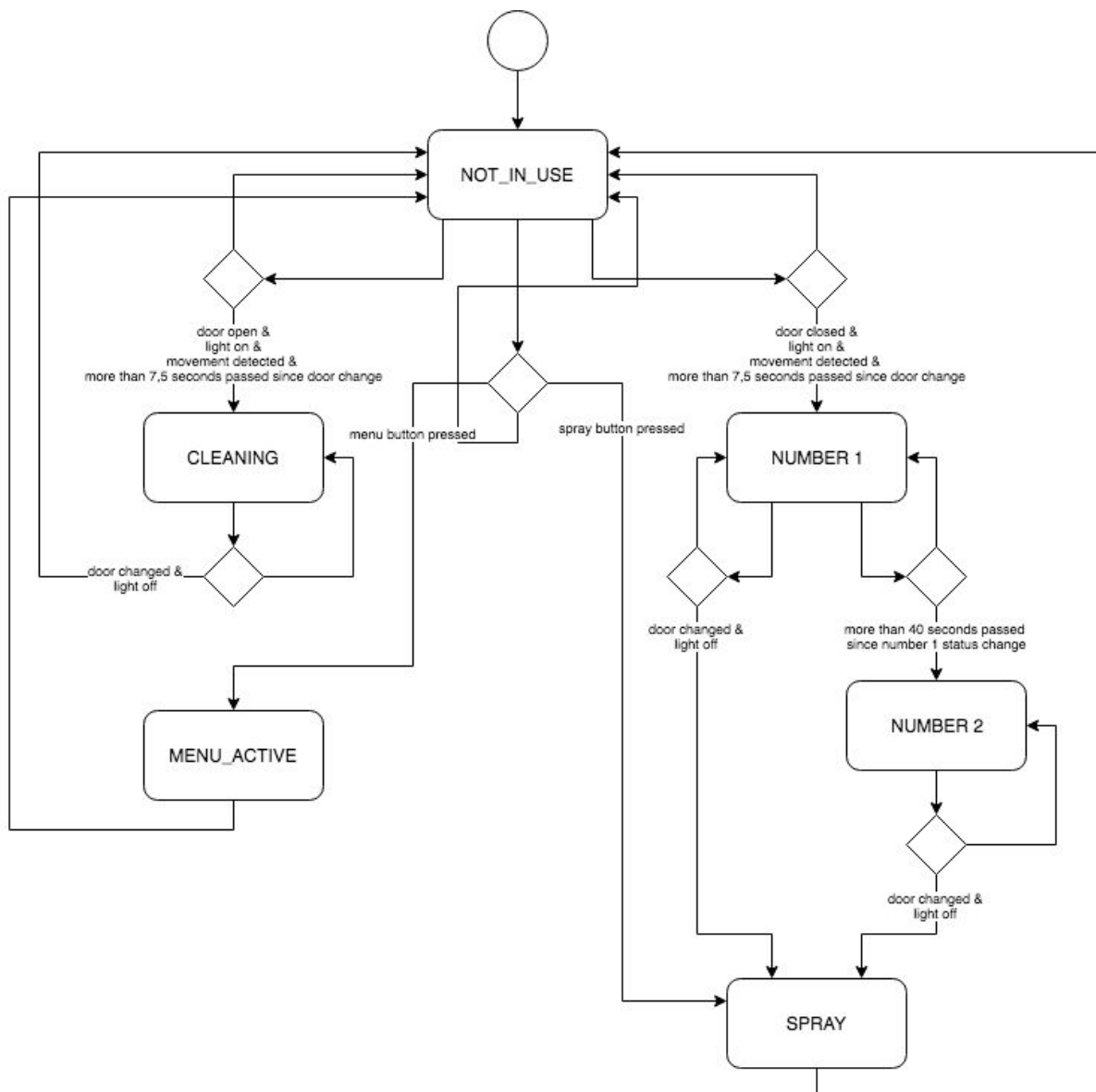
Introductie

Onze opdracht is om van de huidige airwick freshmatic, wat momenteel een 'dom' systeem is, te transformeren naar een 'slim' systeem. Wij mogen zelf bedenken hoe en wat we gebruiken (wat ons ter beschikking is gesteld in de kit) om dit doel te verwezenlijken.

Flow

Toen we begonnen met het project, zijn we begonnen met het maken van een flow diagram. Dit deden wij omdat we dachten dat dit ons een goede start zou geven voor het project. Tijdens het project hebben we veel verschillende configuraties van sensoren die de flow bepalen. Uiteindelijk kozen we voor de configuratie die te zien is op het flow diagram op de volgende pagina.

De flow is grotendeels tot zijn definitieve stand gekomen nadat we definitieve besluiten hebben genomen over welke sensoren we waar voor hebben gebruikt. Het spreekt voor zich dat dit niet het eerste diagram is geweest dat we voor deze opdracht hebben gemaakt.



Functionaliteit

De functionaliteit van het systeem is zoals je kan verwachten een stuk 'slimmer' dan voordat de sensoren zijn toegevoegd. We gebruiken verschillende sensoren om de status van gebruik te detecteren, we onderscheiden drie verschillende statuses: number 1, number 2 en cleaning. Daarnaast kun je direct een keer sprayen door gebruik te maken van een knop.

De verschillende statuses kunnen als volgt worden bereikt:

- Number 1:

- Gesloten deur
 - Licht aan
 - Beweging gedetecteerd
 - Er is een tijd van 7,5 seconden verstreken sinds de deur gesloten is
- Number 2:
 - Voortkomend uit "Number 1"
 - Een tijd van 40 seconden is verstreken sinds de status is veranderd
- Cleaning:
 - Open deur
 - Licht aan
 - Beweging gedetecteerd

Natuurlijk gaan deze statussen gepaard met een gepaste reactie. Na Number 1 kun je een enkele spray verwachten, na number 2 twee. Tijdens cleaning natuurlijk geen.

Hardware

We hadden volgens de opdracht complete vrijheid om de sensoren in te zetten zoals we zelf wilden. Uiteindelijk hebben we elke sensor weten te benutten behalve de ultrasonische afstandsensor. Elke andere sensor is ingezet om de huidige status van het programma te bepalen.

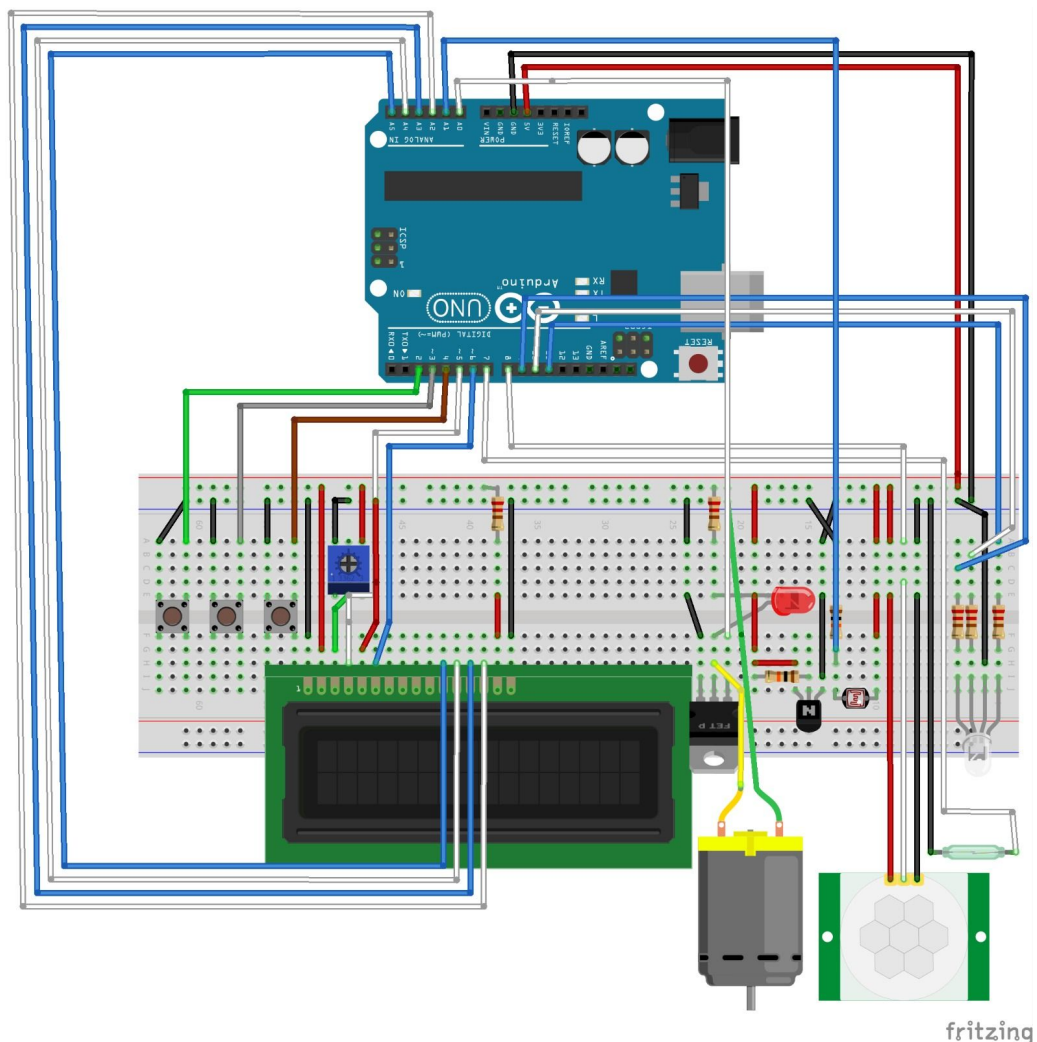
Naast de geleverde sensoren hebben we onze luchtverfrissers gemodificeerd op een manier waarop er geen delay ontstaat. Op deze manier hoefden we geen rekening te houden met enige opstartvertraging. Daarnaast hebben we voor de interactie tussen de gebruiker en het systeem gekozen om het LCD-schermje te gebruiken voor het verstrekken van informatie zoals bijvoorbeeld de temperatuur.

Tijdens het testen van ons systeem hebben we veel gebruik gemaakt van de overige hardware, vooral de verschillende LEDs kwamen erg van pas tijdens het testen en ontwerpen van het systeem. We hebben uiteindelijk ook nog ervoor gekozen om de

RGB-LED in te zetten als een status indicator zodat de gebruiker ook nog enige inzicht heeft in het systeem naast de informatie die weergegeven wordt op het LCD-schermje.

Zoals eerder genoemd is de ultrasonische afstands sensor niet gebruikt, we hebben wel meerdere pogingen gedaan om de sensor te integreren in het systeem maar kwamen al snel tot de conclusie dat de sensor weinig doorslaggevende informatie over de omgeving meebracht die we konden gebruiken in het systeem. Om deze reden besloten wij dat integratie hiervan onnodig was.

We waren nog enige tijd overtuigd dat we ook andere hardware die niet meegeleverd werd met de KIT konden gebruiken. We hebben nog andere hardware willen integreren zoals een precision timer, en een ander LCD scherm (kleiner). Maar we kwamen er gelukkig al relatief snel achter dat dit niet de bedoeling was dus hebben we dit maar losgelaten.



Software

Alle onderstaande code is tevens te zien op onze github repository:

<https://github.com/vincevannoort/interaction-technology-air-refreshener/blob/master/index.ino>

Tijdens het schrijven van het systeem, hebben we we ermee rekening gehouden dat de knoppen altijd direct moeten kunnen reageren. Daarvoor hebben we nergens een delay gebruikt en de knoppen altijd aan het begin van de loop gehouden. Hierdoor reageert het systeem altijd direct op het indrukken van knoppen.

Statussen

Voor het beheren van statussen hebben wij een enumeration opgezet. Door het gebruiken van een enumeration is het gebruik van statussen in de code een overzichtelijker dan het gebruik van integers op zichzelf.

```
enum State {  
    SETUP,  
    NOT_IN_USE,  
    ANALYSING,  
    IN_USE_NUMBER1,  
    IN_USE_NUMBER2,  
    IN_USE_CLEANING,  
    SPRAYING,  
    MENU_ACTIVE  
};
```

Status wijzigen

Daarnaast hebben we voor de statussen een functie opgezet die verantwoordelijk is voor het veranderen van huidige status. Deze zorgt ervoor dat de status niet veranderd wordt als de huidige status ook de nieuwe status wordt. Tevens is deze functie het centrale punt voor het wijzigen van statussen. Altijd als er een status wordt gewijzigd, zal deze functie worden aangeroepen.

```

static void switch_status(int new_state) {
    /** prevent double switching */
    if (current_state == new_state) { return; }
    previous_state = current_state;
    current_state = new_state;

    /** switch led color */
    switch(current_state) {
        case State::NOT_IN_USE: { Sensors::set_rgb_led_color(255, 0, 0); break; }
        case State::ANALYSING: { Sensors::set_rgb_led_color(0, 255, 0); break; }
        case State::IN_USE_NUMBER1: { Sensors::set_rgb_led_color(0, 0, 255); break; }
        case State::IN_USE_NUMBER2: { Sensors::set_rgb_led_color(255, 0, 255); break; }
        case State::IN_USE_CLEANING: { Sensors::set_rgb_led_color(255, 255, 0); break; }
        case State::SPRAYING: { Sensors::set_rgb_led_color(255, 255, 255); break; }
        case State::MENU_ACTIVE: { Sensors::set_rgb_led_color(0, 0, 0); break; }
    }
    Sensors::reset_time_passed();
}

```

Voor het gebruik van alle sensoren hebben we gekozen om een globale Sensor klasse te maken (dit had tevens een namespace kunnen zijn aangezien er alleen maar statische methoden in staan).

Status flow

In onze main loop hebben we een switch die kijkt in welke status het systeem momenteel is. Hieronder is een voorbeeld hiervan te zien waar alle overbodige code is verwijderd.

```

/**
 * States
 */
switch(current_state) {
    case State::NOT_IN_USE: { break; }
    case State::ANALYSING: { break; }
    case State::IN_USE_NUMBER1: { break; }
    case State::IN_USE_NUMBER2: { break; }
    case State::IN_USE_CLEANING: { break; }
    case State::SPRAYING: { break; }
    case State::MENU_ACTIVE: { break; }
}

```

Statuswijziging vereisten

We hebben ervoor gekozen om functies te maken die makkelijk begrijp- en leesbaar zijn voor mensen. Daardoor is het makkelijk om in een ogenblik te zien wat de vereisten zijn voor het veranderen van een status. We merkten snel dat dit voor ons heel erg handig was met testen, omdat het makkelijk was om waardes en vereisten voor het veranderen van een status te kunnen wijzigen. Het beste voorbeeld hiervan is in de 'analysing' status, hieronder te zien (overbodige code is verwijderd).

```
/**
 * Analysing
 * @state-change to number 1
 * @state-change to number 2
 * @state-change to cleaning
 */
case State::ANALYSING:
{
  /**
   * Number one requires:
   * - door to be shut
   * - light to be on
   * - is movement detected
   */
  if (
    Sensors::is_door_closed() &&
    Sensors::is_light_on() &&
    Sensors::is_movement_detected() &&
    Sensors::is_time_passed_since_door_change(Variables::TIME_AFTER_DOOR_CHANGE)
  ) {
    Sensors::reset_door_status();
    Sensors::switch_status(State::IN_USE_NUMBER1);
  }
  break;
}
```

Opslaan hoeveelheid sprays in EEPROM

Voor het opslaan van de hoeveelheid sprays hebben we de ingebouwde EEPROM gebruikt van de Arduino.

Analyse

Systeem evaluatie

Voor de analyse hebben we gebruik gemaakt van twee verschillende configuraties, deze configuraties hebben beide een aantal tests doorlopen. We hebben over al deze tests informatie opgevangen over de waarheid, testwaarheid en tijd. Met deze gegevens konden we meer inzicht krijgen in het systeem en hebben we de gegevens van beide configuraties tegen elkaar uiteengezet. Gegevens over deze evaluatie zijn verder in dit verslag te vinden.

Samplenummer	Testcase	Time	Population 1	Correct-rate 1	Population 2	Correct-rate 2
1	Pee	0:20	Pee	1	Pee	1
2	Poo	0:50	Poo	1	Poo	1
3	Poo	1:15	Poo	1	Poo	1
4	Poo	0:43	Poo	1	Poo	1
5	Cleaning	4:21	Cleaning	1	Cleaning	1
6	Pee	0:27	Pee	1	Pee	1
7	Cleaning	2:34	Cleaning	1	Cleaning	1
8	Poo	1:32	Poo	1	Poo	1
9	Pee	0:28	Pee	1	Pee	1
10	Cleaning	3:16	Cleaning	1	Cleaning	1
11	Pee	0:24	Pee	1	Pee	1
12	Poo	2:11	Poo	1	Poo	1
13	Pee	0:33	Poo	0	Pee	1
14	Cleaning	2:58	Cleaning	1	Cleaning	1
15	Poo	1:43	Poo	1	Poo	1
16	Poo	1:06	Poo	1	Poo	1
17	Pee	0:29	Pee	1	Pee	1
18	Poo	3:34	Poo	1	Poo	1
19	Cleaning	8:39	Cleaning	1	Cleaning	1
20	Pee	0:16	Pee	1	Pee	1
21	Poo	1:44	Poo	1	Poo	1
22	Pee	0:25	Pee	1	Pee	1
23	Poo	1:32	Poo	1	Poo	1
24	Poo	2:02	Poo	1	Poo	1
25	Pee	0:15	Pee	1	Pee	1
26	Cleaning	5:23	Cleaning	1	Cleaning	1
27	Cleaning	7:08	Cleaning	1	Cleaning	1

28	Pee	0:31	Poo	0	Pee	1
29	Poo	1:55	Poo	1	Poo	1
30	Pee	0:24	Pee	1	Pee	1

<i>Testcases</i>	Testcases count	Population 1 Correct	Correct-rate percentage	Population 2 Correct	Correct-rate percentage
Cleaning	7	7	1	7	1
Pee	11	9	0,8181818182	11	1
Poo	12	12	1	12	1
Eindtotaal	30	28	0,9333333333	30	1

μ (population1) = 0,93333

μ (population2) = 1

In onze eerste test ondervonden we dat het systeem het over het algemeen goed deed. We zagen ook direct de limitatie van ons systeem terug, we gingen ervan uit dat plassen een maximale tijd heeft. Aangezien de gevallen die niet correct waren, voort kwamen uit de maximale tijd van plassen komt, hebben we ervoor gekozen om bij de tweede test variatie deze maximale tijd te verhogen. Een betere oplossing zou zijn om een andere techniek te gebruiken om te bepalen wanneer iemand een grote boodschap doet. We hebben overwogen om de bewegingssensor te gebruiken om te kijken of er toiletpapier wordt gebruikt, maar over het algemeen gebruiken vrouwen altijd toiletpapier, bij een een kleine boodschap en een grote boodschap.

Gebruikers evaluatie

Natuurlijk is er ook ruimte voor een gebruikersevaluatie, want cijfers zeggen soms niet alles. De ruimte waarin we onze systeemconfiguraties hebben getest was een relatief grote badkamer. Nadat we onze systeemevaluatie hebben afgemaakt hebben we besloten om het systeem een periode actief te laten zijn in de eerder genoemde badkamer. Het product werd door alle inwonenden als prettig ervaren, al waren er wel enige klachten over de parfumkeuze.

Resultaten

Persoonlijke ervaring

Niek: Ik heb de opdracht als leuk ervaren, vooral vanwege het feit dat ik veel van mijn kennis heb kunnen gebruiken die ik heb opgedaan tijdens mijn vorige studie. Het was ook een prettige afwisseling van het reguliere theoretische onderwijs. Verder vond ik de praktische ervaring van het bouwen van de stickuino een echte verrijking.

Vince: Ik vond het een interessant opdracht, voornamelijk door de hoeveelheid sensoren die gecombineerd moesten worden. Dit was de eerste keer dat ik heb gesoldeerd, iets wat denk ik later nog wel van pas kan komen. Het leukste gedeelte van het project vond ik, zelf bedenken hoe de code gaat werken en dat uitschrijven. Ik merkte tijdens het schrijven van het verslag dat ik het analyse gedeelte nog niet helemaal goed snap qua toepassing voor dit project. De online cursus was duidelijk, maar het zelf goed toepassen vergt nog veel werk, misschien interessant om een vak te kiezen in statistiek.

Conclusie

In de tijd die beschikbaar was, hebben we een mooi project opgezet. Tijdens dit project hebben we veel geleerd over solderen, microcontroller en c/c++ zelf. Ook de praktische problemen die je tegenkomt in een project met hardware waren naast frustrerend, ook uiterst leerzaam voor ons beide. We kijken dan ook uit naar het volgende project van dit vak!