

Flipperkast en betaalmodule

6e jaar Elektriciteit-elektronica

Schooljaar 2020-2021

Robin Monseré

Begeleiders: Dhr. K. Hertens

Voorwoord

Voor onze geïntegreerde proef kregen we de opdracht om zelf een project te kiezen en dit te realiseren. Ik was dan ook best blij dat we zelf die keuze mochten maken. Ik ben voor een flipperkast gegaan, en er mijn eigen draai aan gegeven. Mijn focus ligt niet enkel op de flipperkast zelf maar ook de betaalmodule die ik er heb bijgemaakt. Deze werkt met RFID tags. Wil je de flipperkast gebruiken? Zet wat Credits op je RFID tag en je kan aan de slag! De gedachte hierachter is dat andere arcade games dan ook gespeeld worden met diezelfde badge. Om dit te realiseren heb ik gebruik gemaakt van meerdere arduinos, solanoids, stuurschakelingen en andere elektronica zoals een I2C LCD, RFID reader, etc.

Graag zou ik nog enkele mensen bedanken die me dit jaar geholpen hebben met mijn GIP, zonder hun zou deze GIP er niet geweest zijn. Eerst en vooral meneer Hertens, die doorheen het jaar tips en uitleg gaf over wat er goed en minder goed ging. Ook Gauthier Vanhove die met zijn 3D printer mij geholpen heeft en stukjes heeft gemaakt. Ook de rest van de klas die kwam testen en hun oprechte mening gaf.

Abstractum

Deze GIP was geen simpele opdracht. Ik heb er zeker veel van geleerd, niet enkel over de elektronica en programmatie maar ook over planning, testen, 3D printen, prototyping, je werk bijhouden en documenteren. Voor mij is het zeker geslaagd als ik kijk naar de hoeveelheid die ik hier heb uit geleerd.

Inhoudsopgave

[1 Inleiding 7](#_Toc65673969)

[2 Betaalmodule 8](#_Toc65673970)

[2.1 Werking 9](#_Toc65673971)

[2.2 Toetsenbord 10](#_Toc65673972)

[2.2.1 Aansluitingen 10](#_Toc65673973)

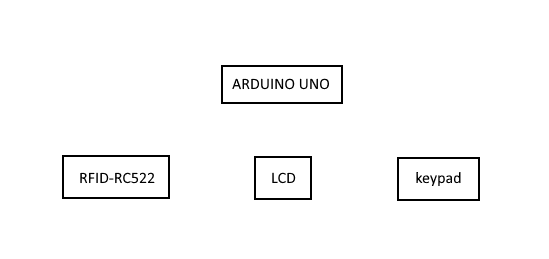
[2.2.2 werking 10](#_Toc65673974)

[2.2.3 Programma 12](#_Toc65673975)

# Inleiding

Mijn GIP valt op te delen in 2 stukken, de flipperkast en de betaalmodule. Ik begin bij de betaalmodule, deze is volledig ge-3D-print. Hier wordt een Arduino Uno gebruikt, een LCD, keyboard en RFID lezer.

Schematische voorstelling:



Figuur 1. Schematische voorstelling betaalmodule

Daarna is het aan de flipperkast.

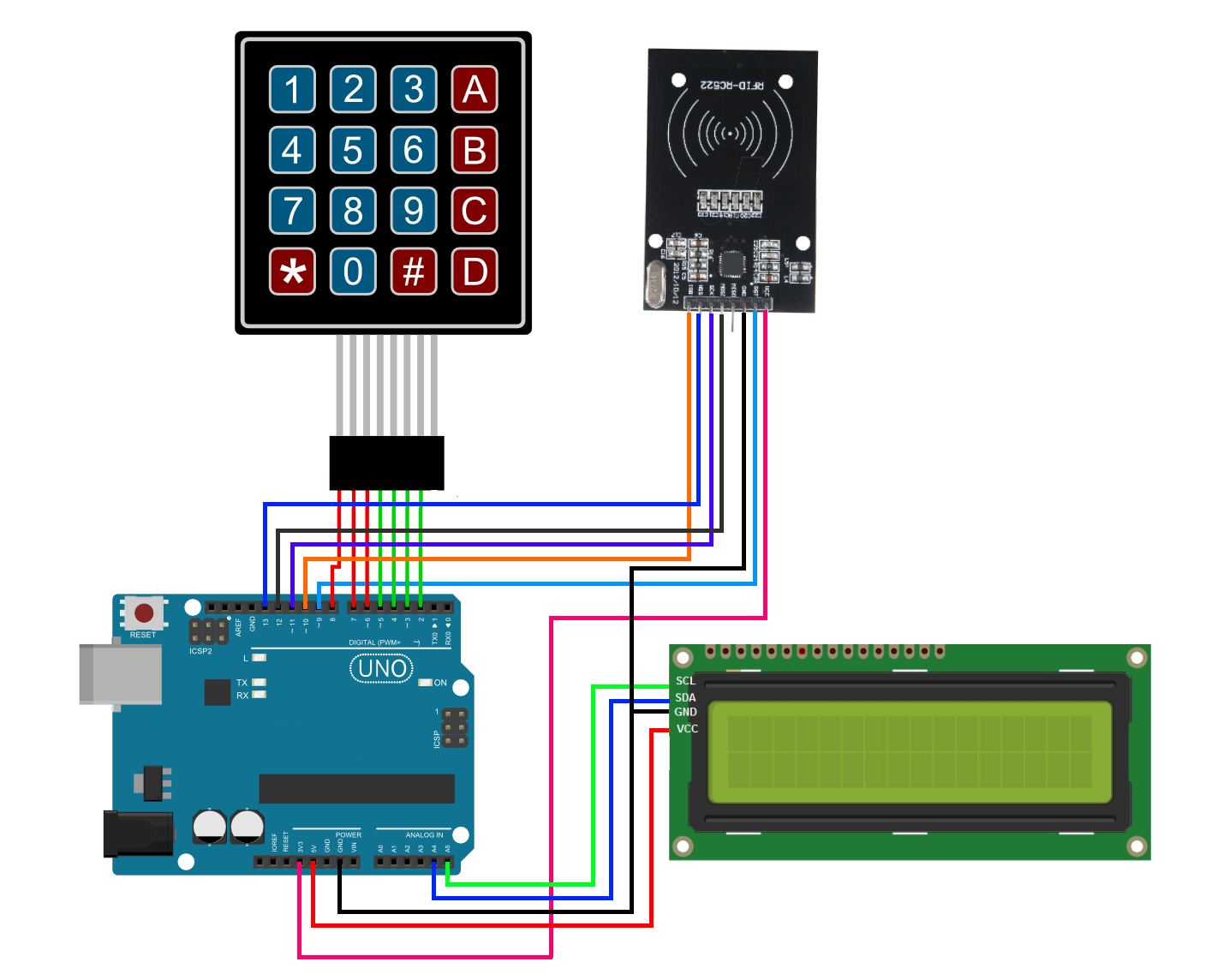
# 125KHz RFID Tag – 5pcsBetaalmodule

Figuur 2. RFID badge

De “betaalmodule” wordt gebruikt om “Credits” op een RFID badge te zetten, dit kan je vergelijken met een bankkaart of de Prizma kaart waarmee je je maaltijd betaalt.

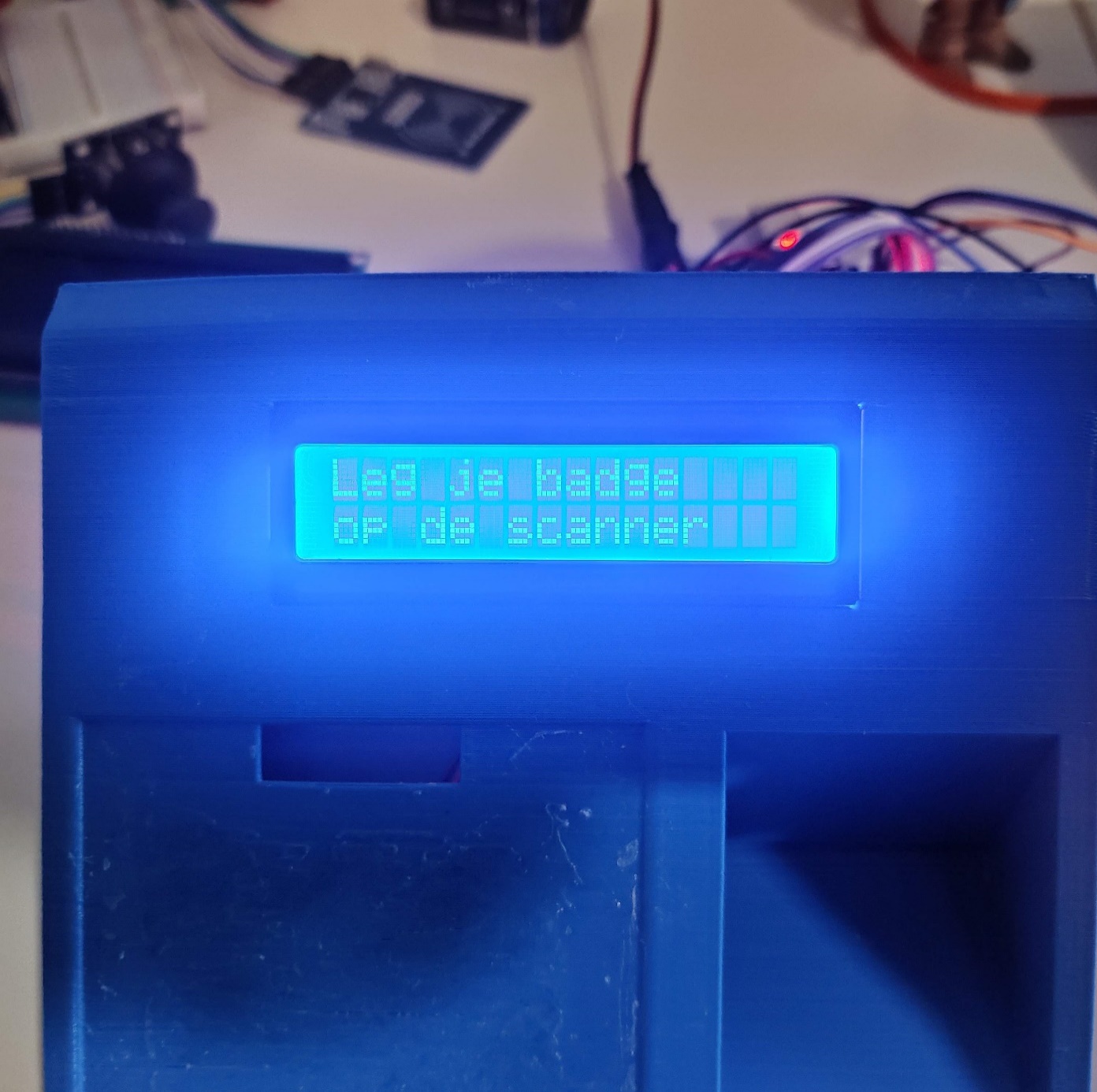
De badge wordt ingelezen via de RFID-RC552, en via de toetsenbord kan je navigeren op de LCD en ingeven hoeveel credits je op je badge wilt zetten.

De betaalmodule bestaat uit een Arduino Uno, een LCD, een membraan toetsenbord en RFID lezer. De Arduino Uno had net genoeg digitale ingangen om alle componenten aan te sluiten.



Figuur 3. verbindingen voor de betaalmodule.

## Werking

 De betaalmodule wacht tot er een badge gedetecteerd wordt, dan staat er “leg je badge op de scanner” op de LCD, zoals in figuur 4. Als die een badge detecteert, zal er “welkom [naam badge]” op de LCD komen, dit komt omdat elke badge een specifieke UID heeft (vb: 5D 68 BD 02). Deze UID wordt ingelezen en vergeleken met UID’s die in het programma zitten. Zo weet het programma welke badge er op de scanner ligt. Het is de bedoeling dat de gebruiker de badge laat liggen tot alles klaar is. Daarna kan de gebruiker kiezen uit 2 dingen, Credits storten, of kijken wat de balans is van de badge (zie figuur 5). Door op \* of # te drukken op het toetsenbord kom je in het corresponderend menu.

Figuur 4. LCD, passief

Figuur 5. LCD, keuzemenu

Druk je op \*, dan krijg je te zien hoeveel credits op de badge staan, door op # te duwen kom je terug in het keuzemenu. Als je kiest om credits te storten wordt je gevraagd hoeveel er moeten bijkomen. Door in te geven op het toetsenbord, met een maximum van 999, komt de waarde op het scherm. Je wordt gevraagd om te bevestigen door op \* te duwen. Als het programma de credits succesvol heeft gestort, komt dit op het scherm om te tonen aan de gebruiker (figuur 6).

Figuur 6. LCD, storting geslaagd

Figuur 7. LCD, storting

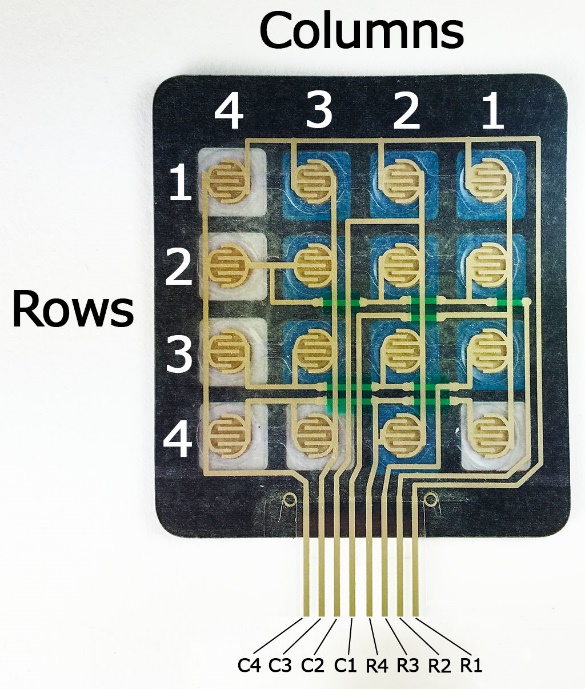
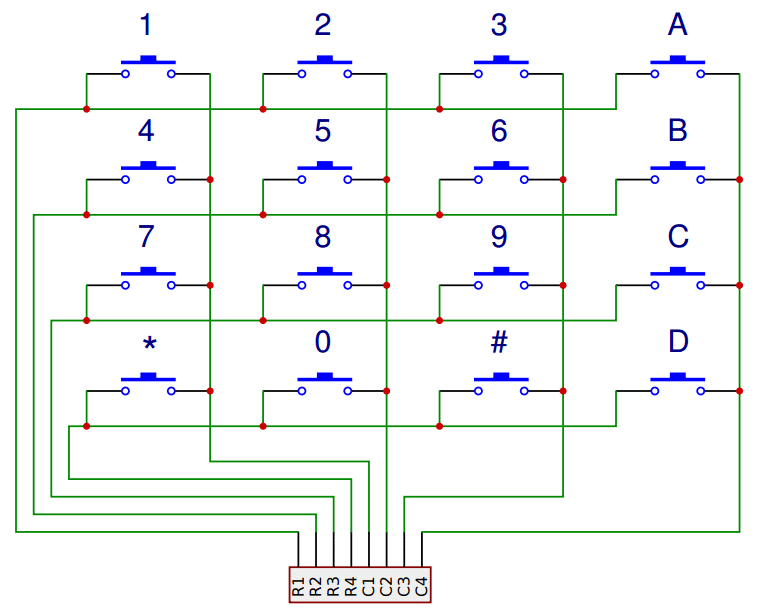
Als dit gedaan is keert het programma terug naar het keuzemenu. Wanneer er plots een badge van de scanner wordt gehaald, dan zal de LCD terug naar het eerste scherm springen. En een andere persoon kan zijn badge op de scanner leggen.

## Toetsenbord

### Aansluitingen

Het aansluiten van dit membraan toetsenbord aan een Arduino is zeer simpel, er zijn 8 pinnen die moeten aangesloten zijn, deze kunnen rechtsreeks aan de digitale ingangen van de Arduino verbonden worden. Op het schema (figuur 3) kan je zien dat er echter maar 7 verbindingen gemaakt worden, dit komt doordat de kolom met de knoppen A, B, C en D niet gebruikt worden. En die hoef ik dan ook niet aan te sluiten.

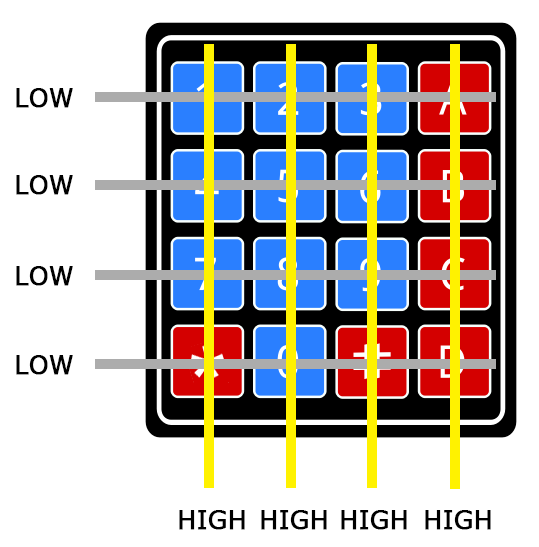
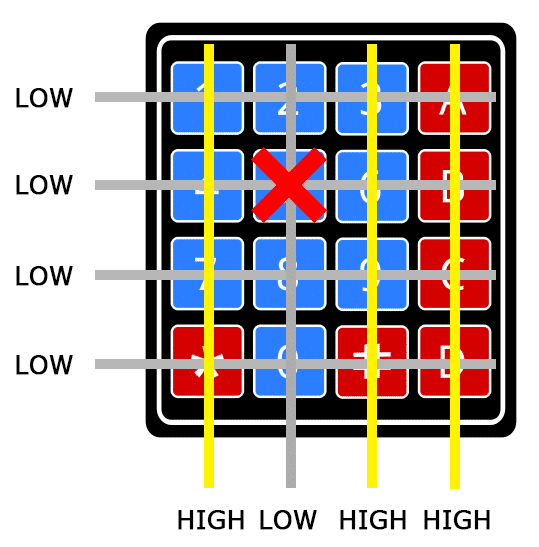
### werking

De knoppen op het toetsenbord zijn gerangschikt in rijen en kolommen. Onder elke knop zit een membraanschakelaar. Elke schakelaar in een rij is verbonden met de andere schakelaars in die rij, en elke schakelaar in een kolom is ook verbonden met elkaar (zie figuur 8 en figuur 9).

Figuur 8. schema toetsenbord

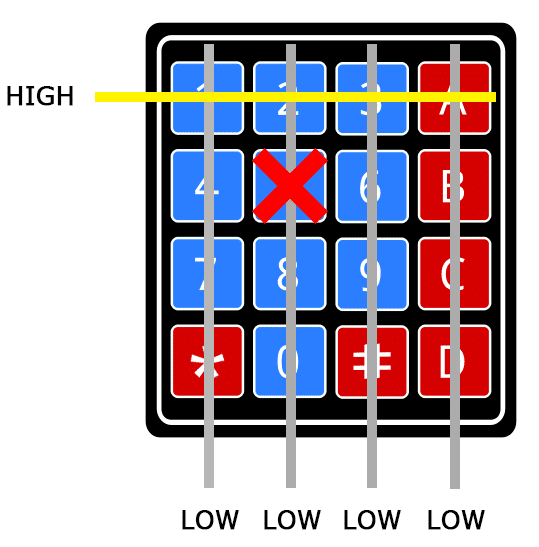
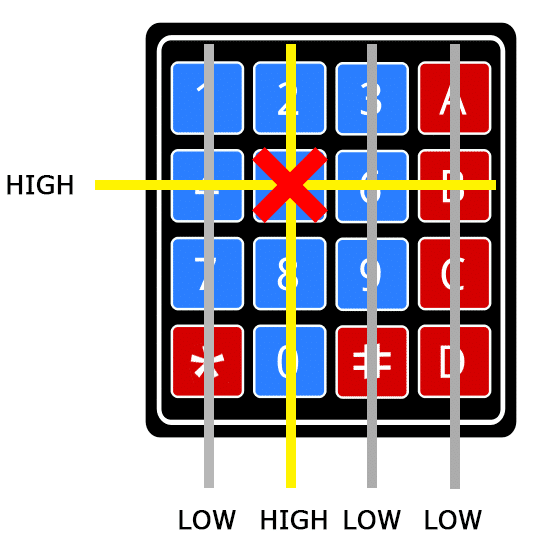
Figuur 9. binnenkant toetsenbord

Dit toetsenbord heeft 16 knoppen, en toch maar 8 aansluitingen. Het bepalen van welke knop er ingeduwd wordt gebeurt in het programma. Dit gebeurt in 4 stappen. Ten eerste is elke kolom pin hoog, en elke rij pin laag. (zie figuur 10). Wanneer er een knop wordt ingedrukt, wordt de kolom pin daarbij laag getrokken (zie figuur 11). De kolom van de ingedrukte knop is nu bepaald, de Arduino weet nu wel nog niet welke knop het precies is, het moet nu enkel de rij nog bepalen. Dit gebeurt door elke kolom pin nu laag te trekken en 1 voor 1 de rij pinnen kort hoog te maken. (zie figuur 12) De Arduino leest de kolom pinnen in en wanneer die hoog komt (zie figuur 13 )weet die nu ook in welke kolom de knop zit. Met die kennis beslist de Arduino welke knop je net ingedrukt hebt. Welk cijfer of letter waar staat op het toetsenbord moet je in het programma declareren.



Figuur 10, “5” ingedrukt en kolom pin laag

Figuur 11, geen knop ingedrukt.



Figuur 12, rij pin trekt kolom pin hoog.

Figuur 13, kolom pinnen laag, rij pinnen 1 voor 1 hoog.

### Programma

#### Bibliotheek

Het programmeren van een Arduino toetsenbord is zeer gemakkelijk. Door gebruik te maken van een bibliotheek hoef je zelf niet veel te programmeren maar kan je een functie oproepen die standaard in de bibliotheek zit. Voor zowat elke sensor/actuator kan je bibliotheek vinden en gebruiken om het programma makkelijker te maken. Alle informatie over de bibliotheek kan je hier vinden: https://playground.arduino.cc/Code/Keypad/

#### Declareren

Je begint in het programma met de nodige variabelen te declareren, ook wordt er een matrix aangemaakt met de karakters van het toetsenbord. Hier worden de karakters A, B, C en D ook niet gedeclareerd, ik heb die toch niet nodig.

|  |
| --- |
| const int ROW\_NUM = 4;  const int COLUMN\_NUM = 3;  char keys[ROW\_NUM][COLUMN\_NUM] =  {   {'1', '2', '3'},   {'4', '5', '6'},   {'7', '8', '9'},   {'\*', '0', '#'}  };  byte pin\_rows[ROW\_NUM] = {8, 7, 6, 5}; // de rijen van het keyboard  byte pin\_column[COLUMN\_NUM] = {4, 3, 2}; // de kolommen van het keyboard  **Keypad** keypad = **Keypad**( makeKeymap(keys), pin\_rows, pin\_column, ROW\_NUM, COLUMN\_NUM ); |

Daarna maak je een “Keypad” object aan, dit gebeurt in rij 36 (figuur 14). Als je meerdere toetsenborden gebruikt voor je project dan maak je meerdere objecten aan en wijs je de correcte pinnen toe aan het juiste object. Voor het oproepen van een functie die in de bibliotheek zit, plaats je altijd het object ervoor. Zo weet de Arduino over welk toetsenbord je het hebt. Bij mijn GIP kan er geen verwarring zijn want ik gebruik maar 1.

#### keypad.getkey()

Om te weten welke toets er op een bepaald moment is ingedrukt roep je de functie “keypad.getKey()” op, deze geeft een karakter terug. Wanneer er geen knop wordt ingedrukt op het moment, dan keert de “getKey” functie ‘NO\_KEY’terug. Dit karakter steek ik in een variabele die ik “keypressed”noem. Nu gaat het programma door 3‘if’en‘else if’ statements. Deze controleren of de variabele een ‘#’ of een‘\*’is, dan gaat het programma verder in de correcte functies. Als er echter op een andere knop gedrukt is, bv: 1 of 5, dan zal het programma in de laatste if else. Er komt een tekst op het scherm die de gebruiker laat weten dat je op ‘#’ of ‘\*’moet duwen. En dan gaat het programma verder.

|  |
| --- |
| char keypressed = keypad.getKey();     if (keypressed == '#')     {       KeuzeMenu = false;       STORT();     }     else if (keypressed == '\*') // \* voor naar balans menu te gaan.     {       KeuzeMenu = false;       BALANS();     }     else if (keypressed != NO\_KEY)     {       lcd.clear();       lcd.setCursor(0, 0);       lcd.print(F("Druk op # of \*"));       delay(1000);       lcd.clear();       lcd.print(F("#: Stort Credits"));       lcd.setCursor(0, 1);       lcd.print(F("\*: Bekijk balans"));     } |

#### keypad.waitForKey()

|  |
| --- |
| char KeyPressed = ' ';     while (KeyPressed != '#')  {       KeyPressed = keypad.waitForKey();     }     KeuzeMenu = true;     KEUZEMENU(); |