Labview Project Lucas en Warre

Dit project stelt een automatisch slot voor dat te vinden kan zijn op bijvoorbeeld een hoteldeur. De deur kan geopend worden met een RFID kaart of met een 4-cijferige code. Bij het openen van de deur zal er bijgehouden worden op welke manier, de toegangscode of RFID-kaart en het tijdstip.

# Opbouw Hardware

Het project bestaat uit verschillende componenten namelijk:

* Arduino UNO R3
* 3x4 matrix keypad
* RFID-lezer module
* Adresseerbare LED-ring
* Led (als simulatie van het deurslot)

Afbeelding met diagram, Plan, schets, Technische tekening

Automatisch gegenereerde beschrijving

Figuur 1: elektronisch schema van het project. (Fritzing, sd)

Zoals weergegeven in figuur 1 is de keypad verbonden met de analoge ingangen en één digitale ingang. Deze keuzen is gemaakt omdat er niet voldoende digitale pinnen over zijn en de RFID een vooraf opgelegde aansluiting heeft.

De RGB LED-ring is verbonden met één poort, namelijk digitale poort 8.

De aansluiting van de RFID-module is vastgelegd door zijn bijhorende bibliotheek en zijn nodige protocollen. In dit project gebruiken we specifiek da Wadda variant van Velleman (Velleman, sd)

Als laatst is er nog een LED verbonden aan digitale poort 3 van de Arduino. Deze LED is echter alleen maar ter demonstratie. In werkelijkheid zou deze vervangen moeten worden door een magnetisch slot of ander vergrendeling middel.

# Werking Arduino Code

De Arduino code die gebruikt wordt is ontworpen om zo weinig mogelijk taken uit te voeren en enkel een communicatie brug te zijn voor de hardware. Om het programma overzichtelijk te houden is het opgesplitst in meerde bestanden en meerdere functies.

In de header file zijn alle bibliotheken geïnstantieerd en zijn alle declaraties opgesplitst in zijn categorie.

Het hoofdbestand is LabvieuwProject.ino, deze bevat enkel de setup en de loop functie. Als een module een setup nodig heeft van meer dan een regel krijgt deze zijn eigen functie in zijn eigen bestand. In de loop functie wordt er gecheckt of Labview via de seriële communicatiedata heeft doorgestuurd, als dat zo is zal er op basis van de waarde de juiste output aangestuurd worden. De RFID-module en de keypad sturen op hun buurt zelf data naar Labview met informatie over de input.

Voor meer informatie over de overige functies verwijs ik naar de Arduino code.

# Werking Labview Code

Bij het starten van het project zal er eerst alle geautoriseerde sleutels ingelezen worden. Dit gebeurt in de status *update\_array*. Van zodra deze zijn ingelezen uit het CSV-bestand is het programma klaar voor gebruik en zit nu in de *idle\_checking* status. Er start ook een wacht animatie op de LED-ring.

Bij een input van een toegangsmogelijkheid zal er eerst gecheckt worden of dit RFID is of een cijfer van de keypad. In het eerste geval is dit RFID, hierbij wordt eerst de seriële date gemodificeerd tot een bruikbare string. Vervolgens wordt er gebruik gemaakt van een zelfgemaakte subVI die gaat checken op basis van een array en een string of de string in het array zit en geeft vervolgens een boolean terug. Het checken van de aanwezigheid van de string gebeurt door de array te doorlopen op basis van zijn indexen en vervolgens elke waarde achter elke index te vergelijken met de string. Bij een overeenkomst stopt deze for-lus en wordt er een boolean teruggegeven. Op basis van deze boolean komt het project in de *noAccess* of *access\_granted* status.

Een tweede input optie is door gebruik te maken van de keypad. Een Cijfercode is vier getallen maar met de keypad kan er maar één tegelijkertijd ingegeven worden. Bij het ontvangen van de keypad data in Labview zal er eerst gecheckt worden of de vorige ingegeven data bij de code hoort (vooraf ingegeven getallen) of dat dit nog andere data was. De vorige data wordt bijgehouden door een shift register. Als dit bijhorende data is zal de nieuwe data bewerkt worden tot een bruikbare string en vervolgens samengevoegd worden met deze data. Vanaf dat de data string vier karakters bevat wordt er net zoals bij de RFID-kaart vervolgens gecheckt door de subVI of de code gekend is en vervolgens op basis van de boolean verder gegaan naar een andere status.

Stemt de data niet overeen met iets uit de array zal er overgeschakeld worden naar de noAccess status. In deze status gebeurt er niets, de deur zal gesloten blijven. Er speelt enkel een animatie af op de LED-ring om de gebruiker te laten weten dat deur niet geopend wordt.

Stemt de date echter wel overeen met een waarde in de array, dan komt het systeem terecht in de access\_granted status. In deze status wordt er sequentieel een commando gestuurd om de deur te openen. Vervolgens wordt er een animatie gespeeld op de LED-ring en gewacht. Als laatst wordt de deur terug vergrendeld. In parallel met deze commando’s wordt er een opgeslagen wie er de deur heeft geopend en op welk tijdstip. Dit gebeurt door een zelfgemaakte subVI die als input een string en een bestandslocatie krijgt. De string wordt gecombineerd met de datum en tijd gescheiden door komma’s en vervolgens geschreven naar het log.csv bestand.

Er kunnen ook nieuwe toegangscodes en RFID-kaarten toegevoegd worden. Dit gebeurt door na het lezen en omvormen van de data tot een string te checken of een knop in Labview is ingedrukt. Opmerking, bij een cijfercode moet deze wel vier getallen zijn. Als de knop ingedrukt is, zal er overgeschakeld worden naar de *add\_new\_access\_token* status. Hierbij wordt er opnieuw een animatie afgespeeld op de LED-ring. In parallel wordt er gecheckt of de data die doorgegeven werd vanuit de vorige status geldig is. Als dat het geval is wordt deze als tekst toegevoegd onderaan in het bestand.

Na het opslaan van de data wordt het bestand terug ingelezen door de status update\_array. Nu is deze data gekend in het programma en kan gebruikt worden om de deur te openen.

# Bibliografie

Fritzing. (sd). *Fritzing*. Opgehaald van Fritzing: https://fritzing.org/

Velleman. (sd). *RFID schrijf en leesmodule*. Opgehaald van Velleman: https://www.velleman.eu/products/view?id=459146