Eindwerk Arduino file

Wat moet de Arduino juist doen ?

1) de Arduino start op

-> bij de initialisatie

a) wordt een bestand gelezen (vaste naam) van de SD-kaart (beschikbaar op de Ethernet-shield).

Dit bestand bevat:

- de basis-URL van de AVICENNA rest-endpoint

- de id van de machine waarmee de Arduino is verbonden

- eventueel het type machine (wanneer we in de toekomst meerdere machine-types van verschillende fabrikanten ondersteunen)

b) de Arduino roept de rest-endpoint aan om aan te geven dat de initialisatie succesvol was

-> wanneer dit lukt is de Arduino klaar -> groene onboard-LED wordt aangezet

-> wanneer dit niet lukt -> eind programma (LED blijft uit)

2) Arduino main applicatie-loop

De Arduino detecteert verschillende signalen van de machine (te bekijken).

Bedoeling is om het starten/stoppen van de machine te detecteren.

Detectie dmv event die wordt aangeroepen wanneer de aangesloten pin van laag naar hoog gaat.

Deze event roept de rest-endpoint van AVICENNA op.

Een extra LED kan eventueel worden gebruikt om aan te duiden dat de aangesloten machine 'loopt'?

3) Aanroep AVICENNA end-point

-> parameters end-point:

ID machine (uitgelezen van de SD-kaart)

EVENTCODE (bv INIT bij opstarten Arduino, START bij start machine, STOP bij stoppen machine,..)

Logging 0=nee, 1=ja

Meer over de werking van de arduino

Arduino unoR3 development board

* Atmega328 microcontroller
* 14 digital IO pinnen ( waarvan 6 PWM outputs kunnen zijn )
* 16 MHz klok
* 5V operating voltage
* Input voltage limits 6-20V
* DC current per I/O pin 40mA
* DC current per 3.3V pin 50 MA

External Interrupts pin 2 en 3 : kunnen worden geconfigureerd om een interrupt te triggeren bij een lage waarde , rising of falling edge, of een verandering in waarde. ( attachInterrupt() function )

Serial : 0 ( RX ) en 1 ( Tx ) : worden gebruikt om te seriele data te krijgen en te verzenden.

PWM : 3,5,6,9,10,11 : geven een 8 bit PWM output signaal analogwrite() function

SPI : 10 (SS),11 (MOSI) ,12 (MISO) ,13 (SCK) voor SPI communicatie door SPI library

LED : aan pin 13 hangt een LED

Arduino ethernet shield

* Micro SD kaart slot
* Ethernet W5100 chip
* SPI communicatie via de pinnen : 13 SCK , 12 MISO , 11 MOSI, 10 SS for ethernet, 4 SS for SD card
* Reset knop op de shield

Libraries nodig

SPI.h ( communicatie )

Ethernet.h ( communicatie )

SD.h ( SD kaart )

* Mac addres kan gedaan worden in de code

Deuren inlezen voltage ? 🡪 2 IO pinnen voor instellen ( met pull downs geactiveerd )

Arduino ondersteund 0 en 5 v logica

Lucas wou zien dat we konden pingen naar de arduino om te zien dat de communicatie oke is en dat het bordje nog draait.

* Iedere Arduino een Fixed IP geven zodat ze hier naar kunnen pingen

IN VERBAND MET CS VOOR SD EN VOOR ETHERNET CONTROLLER

DE CS voor de SD kaart op pin 4 !

DE CS voor de ethernet controller op pin 10 !

SD kaart uitleg

Wat is een SD kaart en micro SD kaart module

De SD en micro SD kaart module laat ons toe om te communiceren met de geheugen kaart en om hier naar te schrijven of de informatie dat er op staat te lezen. De modules gebruiken het SPI protocol.

Om deze te gebruiken kunnen we simpel weg de arduino SD library gebruiken. ( deze staat standaard geinstalleerd op de arduino applicatie.

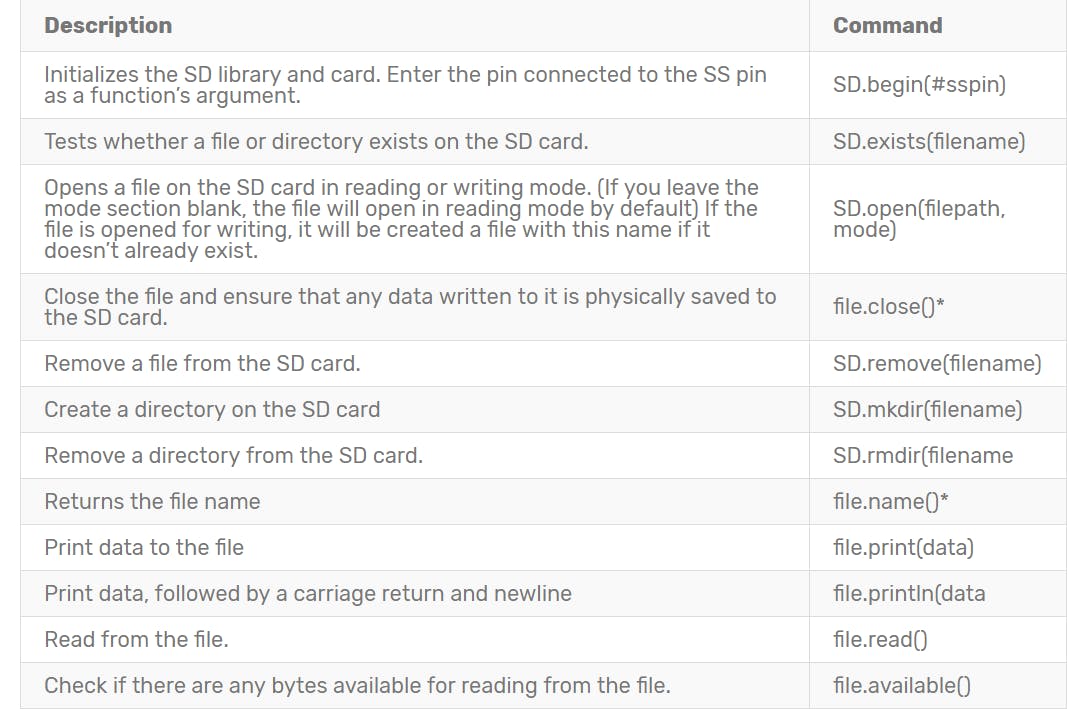
Opmerking deze modules werken vaak niet goed met hoge opslag geheugen kaarten. Om een goede werking te garanderen wordt er best gewerkt met een 2Gb SD kaart en een 16GB micro SD kaart.

Handige link : <https://www.arduino.cc/en/reference/SD>

Libraries nodig om te werken met de SD kaarten :

* SPI.h ( aangezien het SPI protocol gebruikt wordt om te communiceren tussen microcontroller en SD kaart )
* SD.h ( een library gemaakt voor SD kaarten )

Belangrijke SD module library commands.



Data lezen van de SD kaart voor te gebruiken met een ethernet shield, deze heeft een onboard SD kaart slot.

De SD kaart gebruikt de SPI bus dus volgende pinnen zijn nodig.

MOSI : pin 11

MISO : pin 12

CLK : pin 13

CS : pin 4

Pin 4 gaan we de naam ChipSelect geven.

We hebben een variabele nodig van het type File.

In de setup gaan we SD.begin () callen, deze initialiseerd de SD library en SD kaart en we geven de Chipselect hiermee mee.

Als dit gelukt is gaan we de file datalog.txt openen.

Dit doen we door de file variabele te gebruiken en de functie SD.open

( Opmerking er kan maar 1 file te gelijk open zijn. )

We kunnen nu of lezen of schrijven

We hebben een var van het type file nodig

Vb file datafile;

Lezen doen we via SD.read

Vb : datafile = SD.open(“datalog.txt”);

Schrijven doen we via .println of .print

Datastring = “This is a test”;

Vb datafile.println(datastring);

Of vb datafile.println(“this is a test”);

Ethernet

De ethernet controller maakt ook gebruik van de SPI bus en gebruikt volgende pinnen

MOSI : pin 11

MISO : pin 12

CLK : pin 13

CS : pin 10