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.

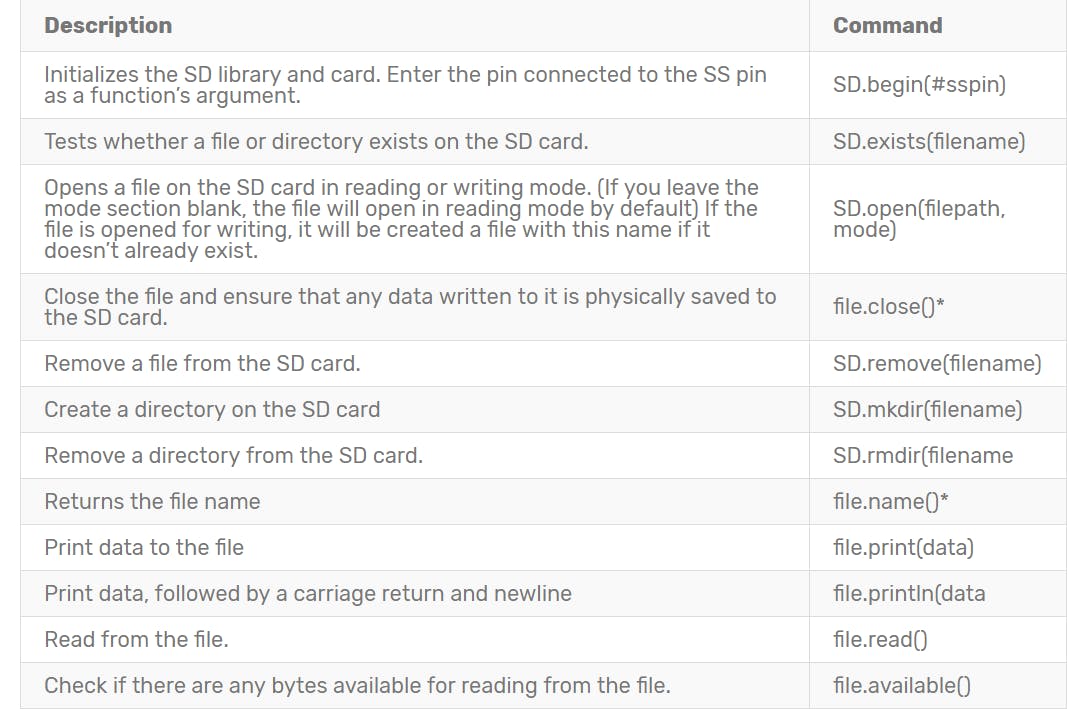
Opmerking deze modules werken vaak niet goed met hoge opslag geheugen kaarten. Om een goede werking te garanderen wordt er best gewerkt met een 2Gb SD kaart en een 16GB micro SD kaart.

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 de functie SD.begin () callen, deze initialiseerd de SD library en SD kaart en we geven de Chipselect waarde 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

Hardware.

MOSI : pin 11

MISO : pin 12

CLK : pin 13

CS : pin 10

Software

Voor de ethernet library te kunnen gebruiken hebben we ook de spi library nodig. Aangezien de ethernet controller in hardware SPI gebruikt.

Aangezien alles op een netwerk een IP adres en een MAC adres nodig heeft gaan we deze in software geven dit kan via de library.

Byte mac[] = ….. een array van 6 bytes deze moeten uniek zijn binnen het netwerk

IPAddress is een soort private variabele dat we kunnen gebruiken voor een fixed ip te geven.

We maken een variabele ip aan van het type IPAddress ( 4 bytes )

IPAddress Ip (192, 168,0,177);

IPAddress myDns(192, 168, 0, 1);

We hebben ook een constante nodig voor de chipselect van de SPI communicatie.

We moeten een ethernet client aan maken via de functie EthernetClient() deze client kan dan connecteren op een specifiek IP adres en poort via de functie client.connect().

Om alles te initialiseren van de ethernet library en de netwerk settings gebruiken we de functie Ethernet.begin () we gaan deze functie ook het vaste mac en vaste ip adres meegeven

Ethernet.begin(mac , ip , myDns); ( zonder DNS )

Ethernet.begin(mac) ( met DNS )

if (Ethernet.begin(mac) == 0) {

Serial.println("Failed to configure Ethernet using DHCP");

Ethernet.hardwareStatus() is een functie dat kijkt welke ethernet controller IC gebruikt wordt, als deze niets vind dan is er een probleem met de ethernet shield

Ethernet.linkStatus() detecteerd of een ethernet kabel aanwezig is of niet.

Ethernet.localIP() krijg het IP adres van de ethernet shield, ( kan handig zijn als je gebruik maakt van een DHCP om te zien welk IP adres je hebt )

Client.Connect(ip,port) ( ip array van 4 bytes )

Client.Connect(URL,port) (URL van het type string )

Returns an int :

1 Succes

-1 timed out

-2 Invalid server

-3 truncated

-4 invalid response

* Ik weet ni of er een DNS server op de intranet zit dus dan hebben we een IP address nodig ipv van een weblink

Bv www.google.be ip : [2a00:1450:400e:80d::2003]

// give the Ethernet shield a second to initialize:

delay(1000);

Serial.print("connecting to ");

Serial.print(server);

Serial.println("...");

// if you get a connection, report back via serial:

if (client.connect(server, 80)) {

Serial.print("connected to ");

Serial.println(client.remoteIP());

// Make a HTTP request:

client.println("GET /search?q=arduino HTTP/1.1");

client.println("Host: www.google.com");

client.println("Connection: close");

client.println();

} else {

// if you didn't get a connection to the server:

Serial.println("connection failed");

}

The domain name system links the website addresses you type into your Web browser to the numeric IP addresses by which servers identify each other on the Internet. Domain names are easier for humans to use, but as part of a small business you can encounter situations where, for technical reasons, you need to know a website's IP address instead of just its domain name. You can find the IP address of any website using Ping, a simple command line application that comes with every Windows installation.

**1.**

Click on the Start menu at the bottom left corner of the screen. Type "Command Prompt" into the search bar and then select "Command Prompt" from the list of results.

**2.**

Type "ping http://website-address.com" where "website-address.com" is the website's address into the command prompt.

**3.**

Write down the IP address that appears in the lines below your "ping" command on the command prompt. This will be the IP address of the website's server.

We zitten op een intranet in een ziekenhuis. Wat is dit juist

Intranets worden gebruikt door veel bedrijven als een manier om hun data te beschermen van de buiten wereld en om de data met andere te delen binnen hun netwerk.

Om een intranet te kunnen gebruiken moet je verbonden zijn op de LAN.

Een intranet gebruikt de communicatie standaard TCP/IP

Om op de intranet te komen heb je geen passwoord of login nodig ?

/\*

Web client

This sketch connects to a website (http://www.google.com)

using an Arduino Wiznet Ethernet shield.

Circuit:

\* Ethernet shield attached to pins 10, 11, 12, 13

created 18 Dec 2009

by David A. Mellis

modified 9 Apr 2012

by Tom Igoe, based on work by Adrian McEwen

\*/

#include <SPI.h>

#include <Ethernet.h>

// Enter a MAC address for your controller below.

// Newer Ethernet shields have a MAC address printed on a sticker on the shield

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };

// if you don't want to use DNS (and reduce your sketch size)

// use the numeric IP instead of the name for the server:

//IPAddress server(74,125,232,128); // numeric IP for Google (no DNS)

char server[] = "www.google.com"; // name address for Google (using DNS)

// Set the static IP address to use if the DHCP fails to assign

IPAddress ip(192, 168, 0, 177);

IPAddress myDns(192, 168, 0, 1);

// Initialize the Ethernet client library

// with the IP address and port of the server

// that you want to connect to (port 80 is default for HTTP):

EthernetClient client;

// Variables to measure the speed

unsigned long beginMicros, endMicros;

unsigned long byteCount = 0;

bool printWebData = true; // set to false for better speed measurement

void setup() {

// Open serial communications and wait for port to open:

Serial.begin(9600);

while (!Serial) {

; // wait for serial port to connect. Needed for native USB port only

}

// start the Ethernet connection:

Serial.println("Initialize Ethernet with DHCP:");

if (Ethernet.begin(mac) == 0) {

Serial.println("Failed to configure Ethernet using DHCP");

// Check for Ethernet hardware present

if (Ethernet.hardwareStatus() == EthernetNoHardware) {

Serial.println("Ethernet shield was not found. Sorry, can't run without hardware. :(");

while (true) {

delay(1); // do nothing, no point running without Ethernet hardware

}

}

if (Ethernet.linkStatus() == LinkOFF) {

Serial.println("Ethernet cable is not connected.");

}

// try to congifure using IP address instead of DHCP:

Ethernet.begin(mac, ip, myDns);

} else {

Serial.print(" DHCP assigned IP ");

Serial.println(Ethernet.localIP());

}

// give the Ethernet shield a second to initialize:

delay(1000);

Serial.print("connecting to ");

Serial.print(server);

Serial.println("...");

// if you get a connection, report back via serial:

if (client.connect(server, 80)) {

Serial.print("connected to ");

Serial.println(client.remoteIP());

// Make a HTTP request:

client.println("GET /search?q=arduino HTTP/1.1");

client.println("Host: www.google.com");

client.println("Connection: close");

client.println();

} else {

// if you didn't get a connection to the server:

Serial.println("connection failed");

}

beginMicros = micros();

}

void loop() {

// if there are incoming bytes available

// from the server, read them and print them:

int len = client.available();

if (len > 0) {

byte buffer[80];

if (len > 80) len = 80;

client.read(buffer, len);

if (printWebData) {

Serial.write(buffer, len); // show in the serial monitor (slows some boards)

}

byteCount = byteCount + len;

}

// if the server's disconnected, stop the client:

if (!client.connected()) {

endMicros = micros();

Serial.println();

Serial.println("disconnecting.");

client.stop();

Serial.print("Received ");

Serial.print(byteCount);

Serial.print(" bytes in ");

float seconds = (float)(endMicros - beginMicros) / 1000000.0;

Serial.print(seconds, 4);

float rate = (float)byteCount / seconds / 1000.0;

Serial.print(", rate = ");

Serial.print(rate);

Serial.print(" kbytes/second");

Serial.println();

// do nothing forevermore:

while (true) {

delay(1);

}

}

}