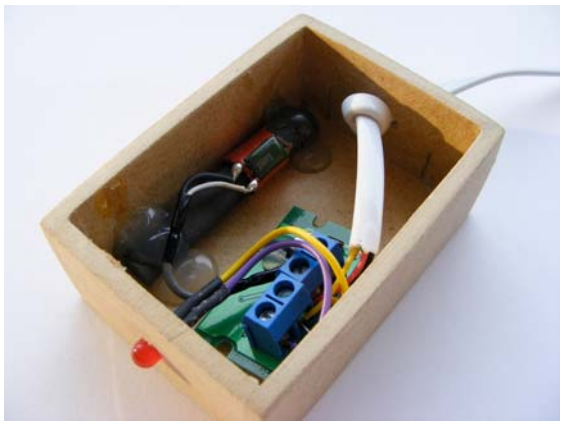
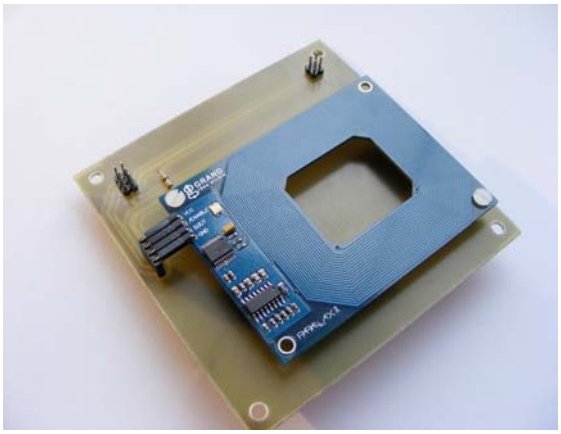
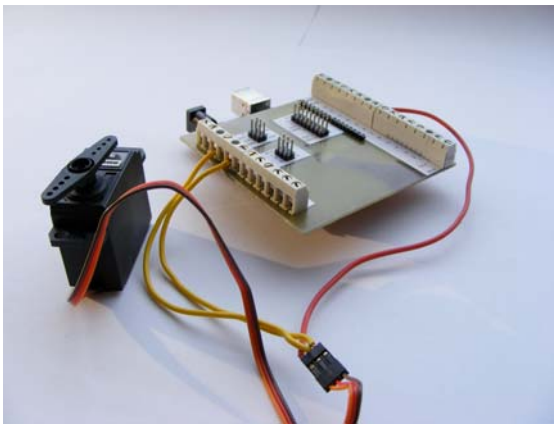
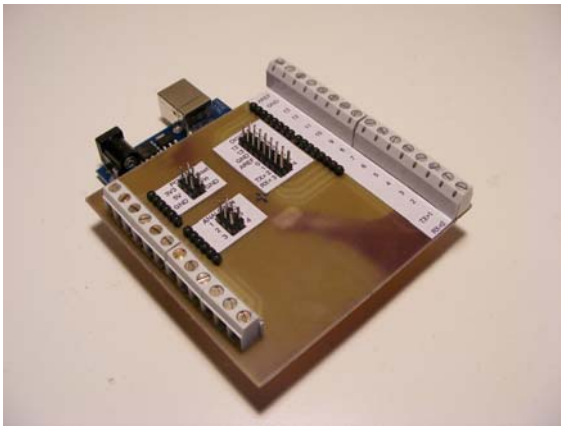
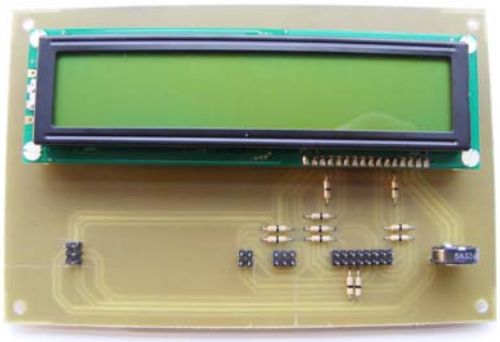
Verslag project
Remco Pel en Jorick de Wit
VTET3D4









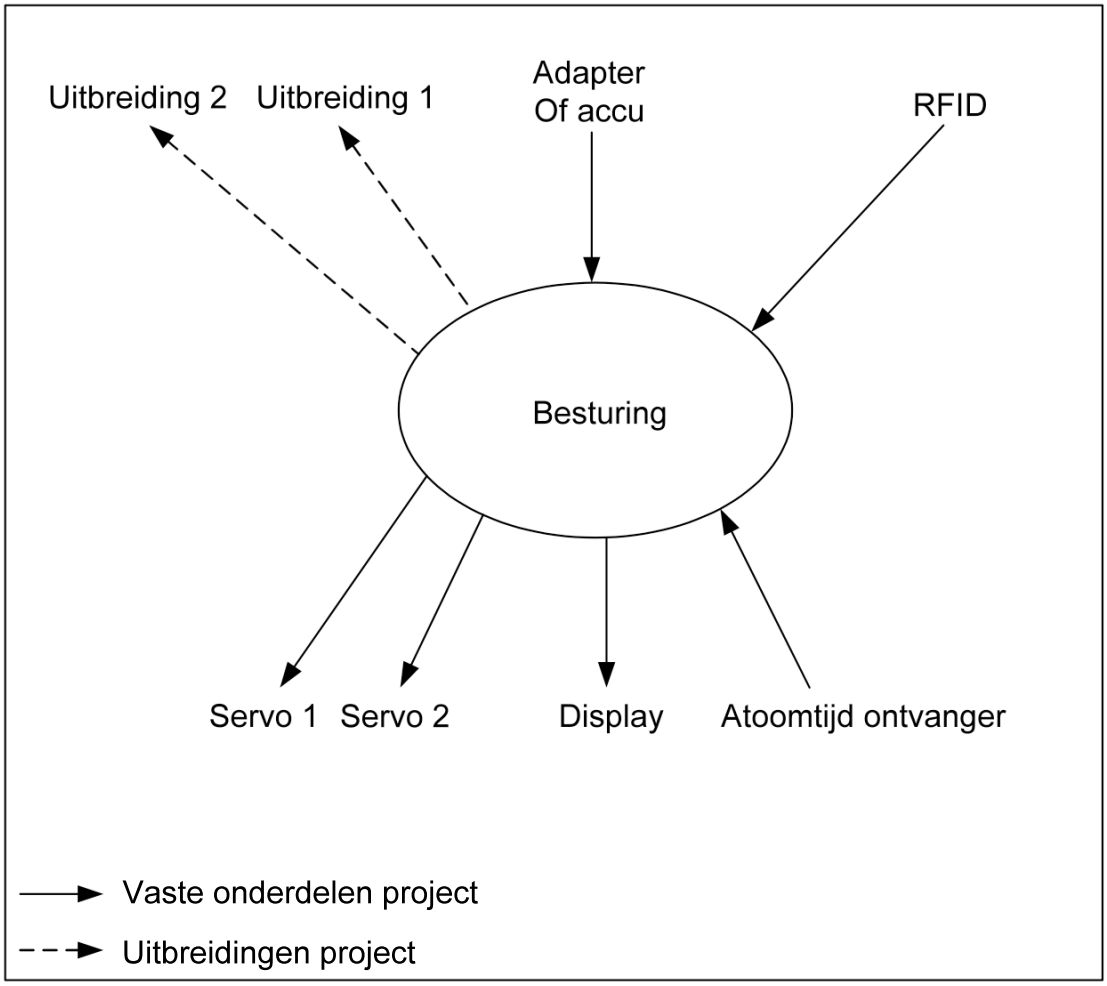


Inhoudsopgaven

Omschrijving project
Organisatie
Planning
Functionele eisen
Niet haalbare project onderdelen / aanpassingen
Problemen project
Ontwerp fase
Bouw fase
Gebruikte onderdelen
Programma’s / flowchart

Bijlage 1 Verantwoording

Omschrijving project
We willen meer kennis gaan opdoen van het programmeren van microcontrollers en het werken met RFID. Dus daarom dit project.



Onze bedoeling van het project is dat we door middel van RFID (Radio Frequency Identification) schakelingen kunnen aansturen.
De schakelingen die we gaan aansturen door middel van RFID zijn twee Servo's die een elektrisch slot van een deur moet voorstellen.
Op de display is het de bedoeling om:

- Naam gebruiker.
- Tijd.
- Toegang geweigerd, door missen van een signaal of een defect.
- Power alert, als de stroom toevoer van af het net onderbroken wordt zal deze verder werken op zijn accu.

Tijdweergave:
We willen met behulp van een dcf ontvanger de tijd ontvangen om weer te geven op het display.

RFID:

- doormiddel van een pasje inloggen.
- Na +/-10 seconde automatisch uitloggen/Vergrendelen van de deur.

Uitbreiding 1:

We willen een verbinding tussen de pc en de besturing van het project. Om door middel van een pasje de computer te laten blokkeren of deblokkeren. Als je bijvoorbeeld wel gemachtigd bent om een ruimte te betreden maar niet de pc te gebruiken zal deze geblokkeerd worden.

Uitbreiding 2:

Registratie van inlogtijden op de pc. De pc zal dan een lijst aanmaken met daarin de tijd inlognaam.

Deze uitbreidingen zijn bedoelt als het RFID systeem helemaal werkt en we tijd over hebben om deze uitbreidingen te realiseren.

Organisatie

Dit project wordt door de volgende personen gemaakt

Remco Pel	projectleider
Jorick de Wit	lid

Begeleider Technisch = Dhr. Kompanje
Begeleider Inhoudelijk proces = Dhr. Van Der Veen

Planning

Uit te voeren activiteiten

week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
	4-3-2009	11-3-2009	18-3-2009	25-3-2009	1-4-2009	8-4-2009	15-4-2009	22-4-2009	29-4-2009	6-5-2009	13-5-2009	20-5-2009	27-5-2009	3-6-2009	10-6-2009	17-6-2009	24-6-2009	1-7-2009
Verslag	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
project uitwerken	X																	
project plan uitwerken	X	X																
offerte maken		X	X															
schema's en documentatie uitwerken				X	X													
sapfabriek						X	X	X										
ontwerpen en testen schakelingen					X				X	X	X	X						
ontwerpen printplaatjes											X	X	X					
testen printplaatjes													X	X				
buffer															X	X	X	
project opleveren																		X

In bovenstaand planning hebben we aangegeven in welke week we wat moeten gaan doen.
(zie bijlage 1 voor de verantwoording)

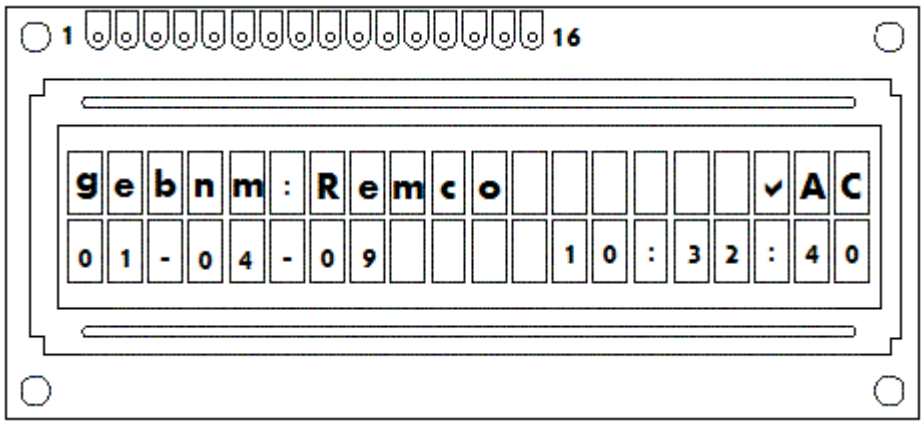
Functionele eisen

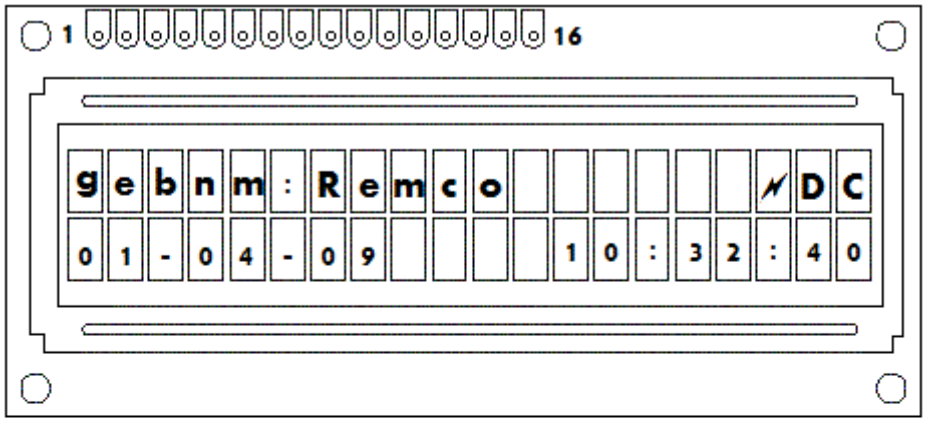
De functionele eisen van het project zijn:

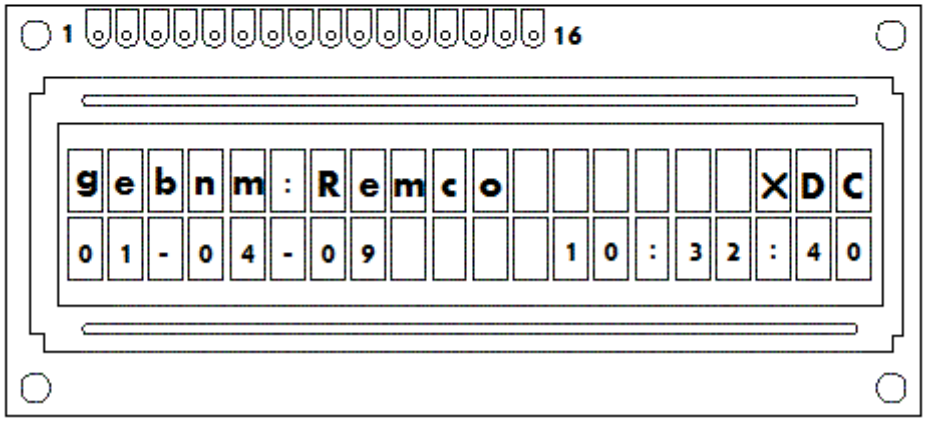
- Tijd en datum weergaven doormiddel van dcf ontvanger
- RFID

Onze bedoeling van het project is dat we door middel van RFID (Radio Frequency Identification) schakelingen kunnen aansturen. De schakelingen die we gaan aansturen door middel van RFID zijn twee Servo's die een elektrisch slot van een deur moet voorstellen. Ook zal er op het display verschijnen welke gebruiker inlogt.
Op het display is het de bedoeling om het volgende neer te zetten:

- Naam gebruiker.
- Tijd.
- Toegang geweigerd, door missen van een signaal of een defect.
- Power alert, als de stroom toevoer van af het net onderbroken wordt zal deze verder werken op zijn accu.







Op bovenstaande afbeeldingen zijn voorbeelden te zien met wat voor tekst we allemaal op het display willen laten zien.

We willen het zo gaan maken dat zodra de stroom uitvalt hij overgaat op een accu. Dit geven we weer met de symbolen:
✔ AC: adapter is aangesloten op het net
🗲/✕ DC: symbolen knipperen systeem werkt op nood accu.

Niet haalbare project onderdelen / aanpassingen

Tijdens ons project zij we tegen een aantal problemen aangelopen. Sommige hebben we kunnen oplossen maar ander problemen hebben we niet kunnen oplossen.

We hadden bedacht om een schakeling te maken die zou schakelen tussen netspanning en een accu. Dit schema hebben we helemaal uitgewerkt in multisim dit werkte ook, maar we hadden daar te weinig tijd voor om dit uit te testen en we het belangrijker vonden om de DCF en RFID programma aan de gang te krijgen.

Omdat we met het samen voegen van het RFID en het DCF programma problemen kregen. Het grootste probleem dat we hadden was dat de twee programma's niet samen kunnen werken. We hebben op internet lopen zoeken naar een oplossing. We kregen op een forum nog wel een aantal tips maar dit hebben we door tijdgebrek niet verder kunnen uitproberen. We hebben dus besloten om die twee programma's apart te demonstreren.

Ook hadden we twee uitbreidingen bedacht (zie omschrijving project **)**

Uitbreiding 1

> We willen een verbinding tussen de pc en de besturing van het project. Om door middel van een pasje de computer te laten blokkeren of deblokkeren. Als je bijvoorbeeld wel gemachtigd bent om een ruimte te betreden maar niet de pc te gebruiken zal deze geblokkeerd worden.

Deze uitbereiding hebben we kunnen realiseren door middel van een programma die we hebben geschreven in Visual Basic

Uitbreiding 2:

> Registratie van inlogtijden op de pc. De pc zal dan een lijst aanmaken met daarin de tijd inlognaam.

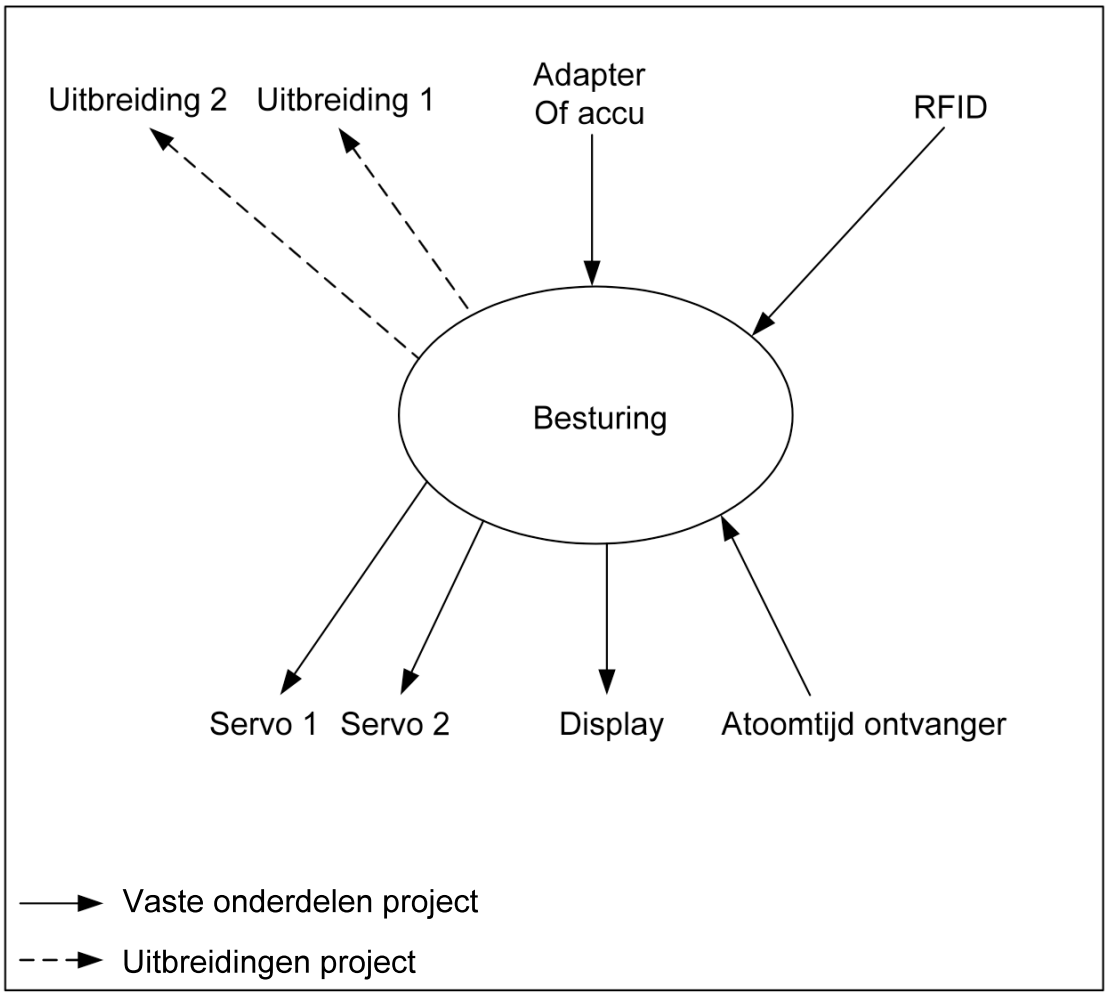
Deze uitbreiding hebben we niet kunnen doen omdat we er geen tijd meer voor hadden, en een ander probleem wat we daar bij ook hadden is dat we het RFID en DCF programma niet samen kunnen laten werken (zoals hierboven is uitgelegd).

Bovengenoemde problemen hebben er voor gezorgd dat we een aantal aanpassingen hebben gedaan. Zoals:
- DCF en RFID werken apart van elkaar in plaats van met elkaar.
- Accu schakeling is niet gerealiseerd.

Problemen project
Ons eerste probleem waar we tegen aanliepen was dat we de DCF ontvanger niet aan de gang konden krijgen. Na een lange tijd van testen hebben we een tweede ontvanger besteld en deze hebben we vervolgens uitgetest we zijn er achter gekomen dat we de DCF ontvanger bij het raam moeten leggen om signaal te kunnen ontvangen.
Nu we het signaal kunnen ontvangen hebben we het ook voor elkaar gekregen om de tijd op het display te krijgen.

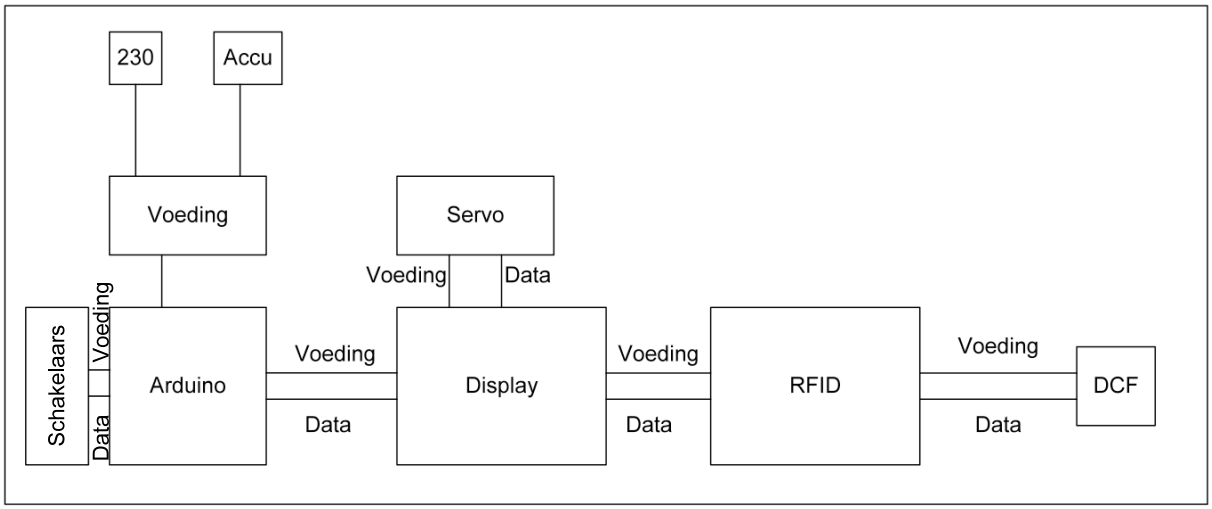
Zie ook vorige hoofdstuk.

Ontwerp fase



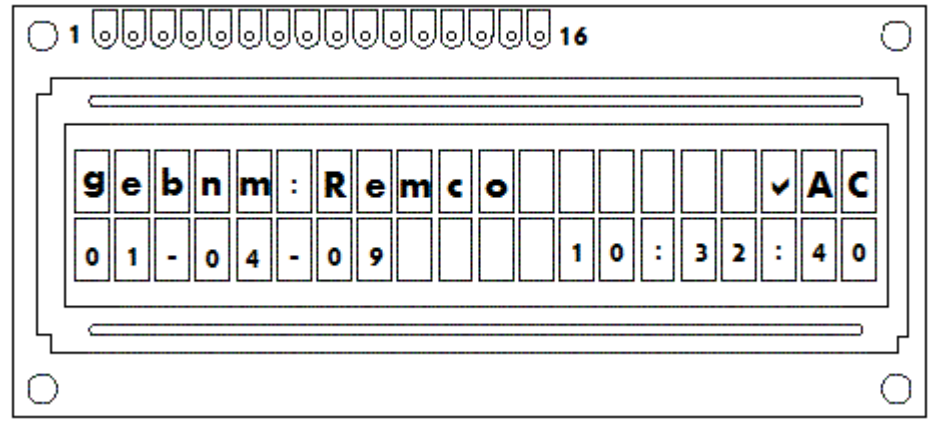
Context diagram
In bovenstaande contextdiagram hebben we aangeven hoe ons project gaat werken. Op de volgende bladzijde zullen we het project verder uitleggen in een blokdiagram.

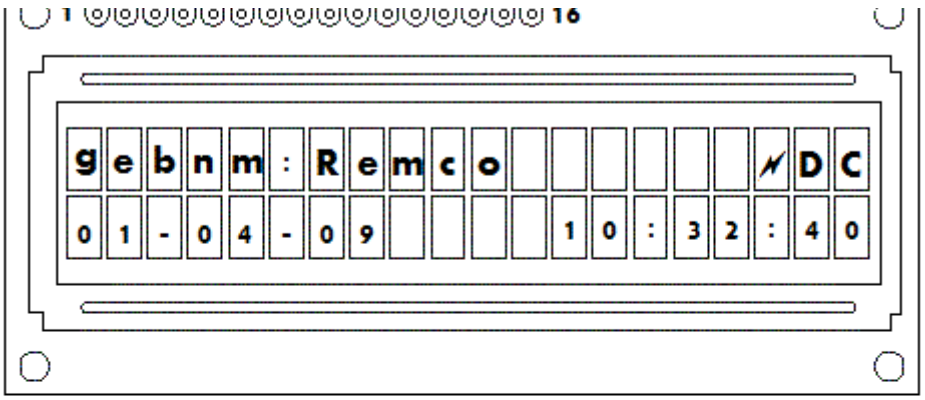
Blokdiagram

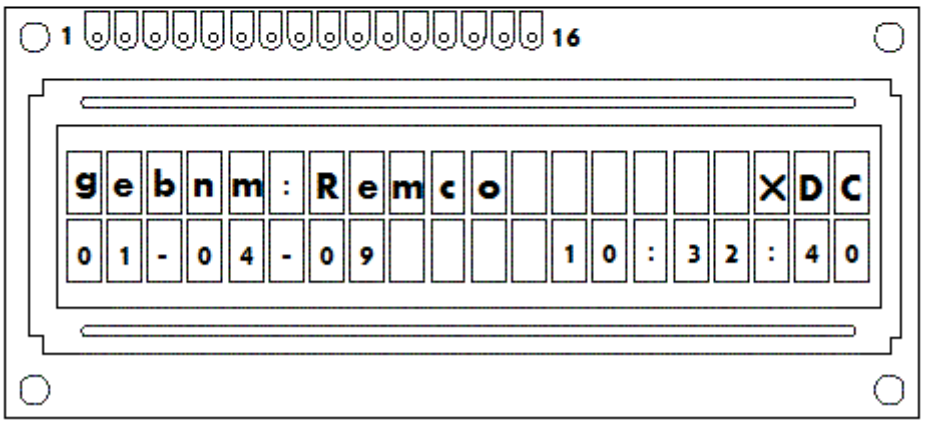


In bovenstaand blokdiagram hebben we weergegeven hoe we alle onderdelen gaan aansluiten.

Technische specificaties display.







Op het display is het de bedoeling om het volgende neer te zetten:

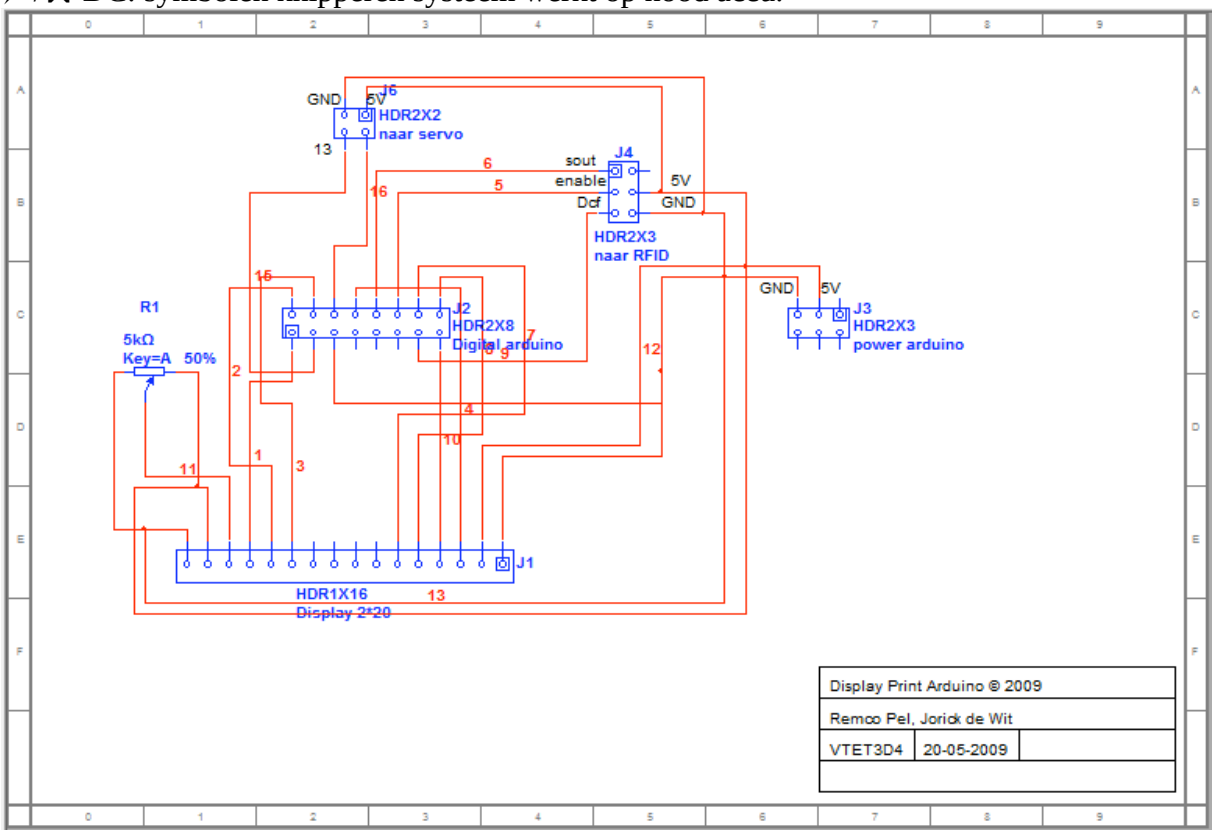
- Naam gebruiker.
- Tijd.
- Toegang geweigerd, door missen van een signaal of een defect.
- Power alert, als de stroom toevoer van af het net onderbroken wordt zal deze verder werken op zijn accu.

Op bovenstaande afbeeldingen zijn voorbeelden te zien met wat voor tekst we allemaal op het display willen laten zien.

We willen het zo gaan maken dat zodra de stroom uitvalt hij overgaat op een accu. Dit geven we weer met de symbolen:

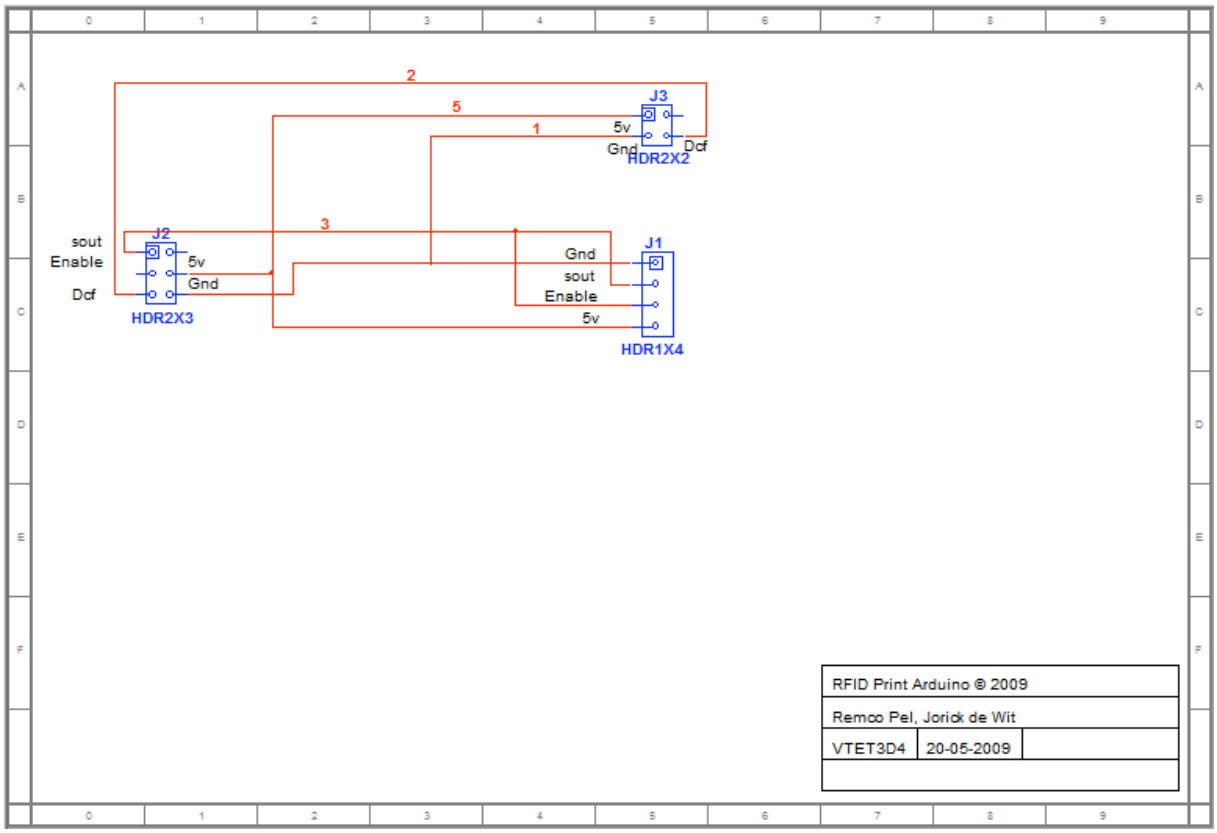
✔ AC: adapter is aangesloten op het net

🗲/✖ DC: symbolen knipperen systeem werkt op nood accu.



Display Print:

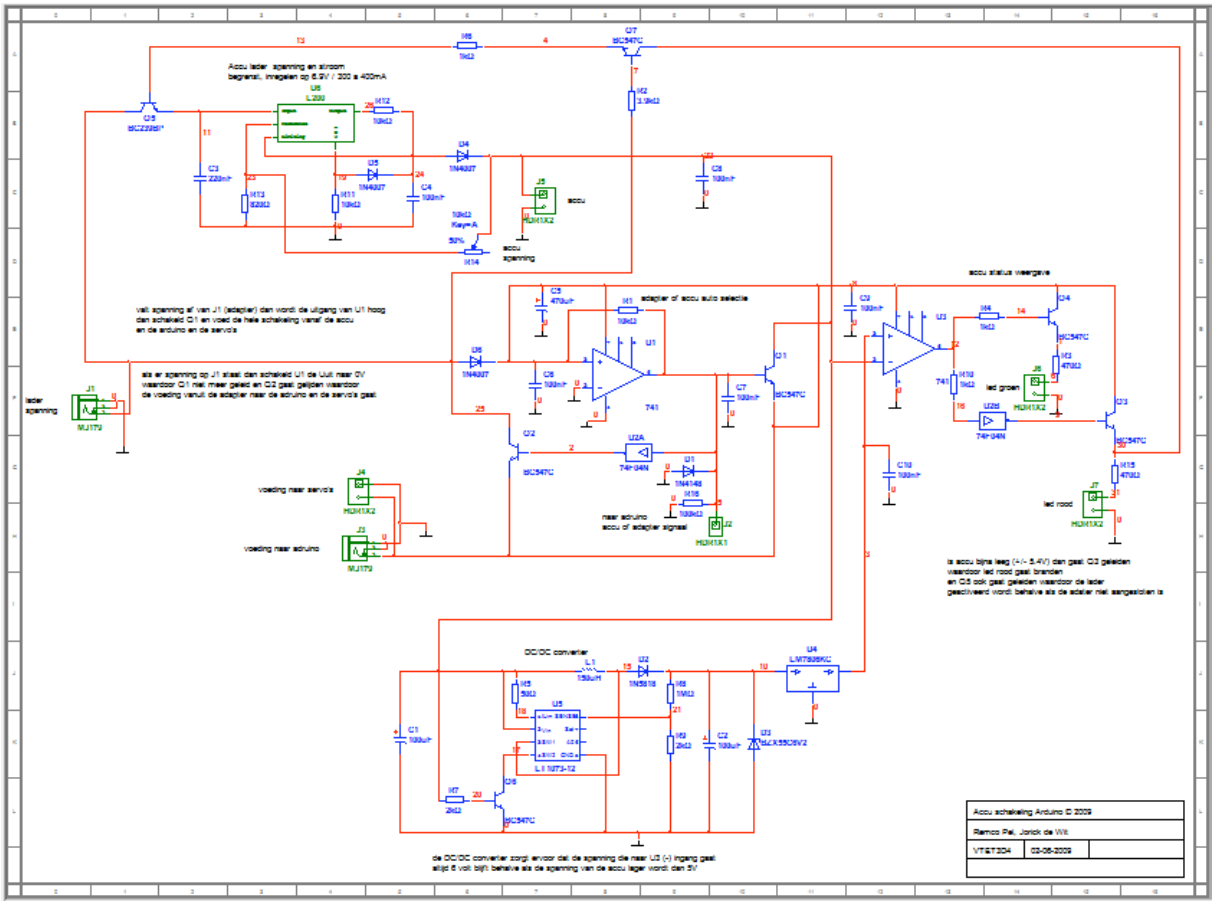
Deze schakeling is bedoeld om het display aan te kunnen sluiten op de arduino.



RFID schema:

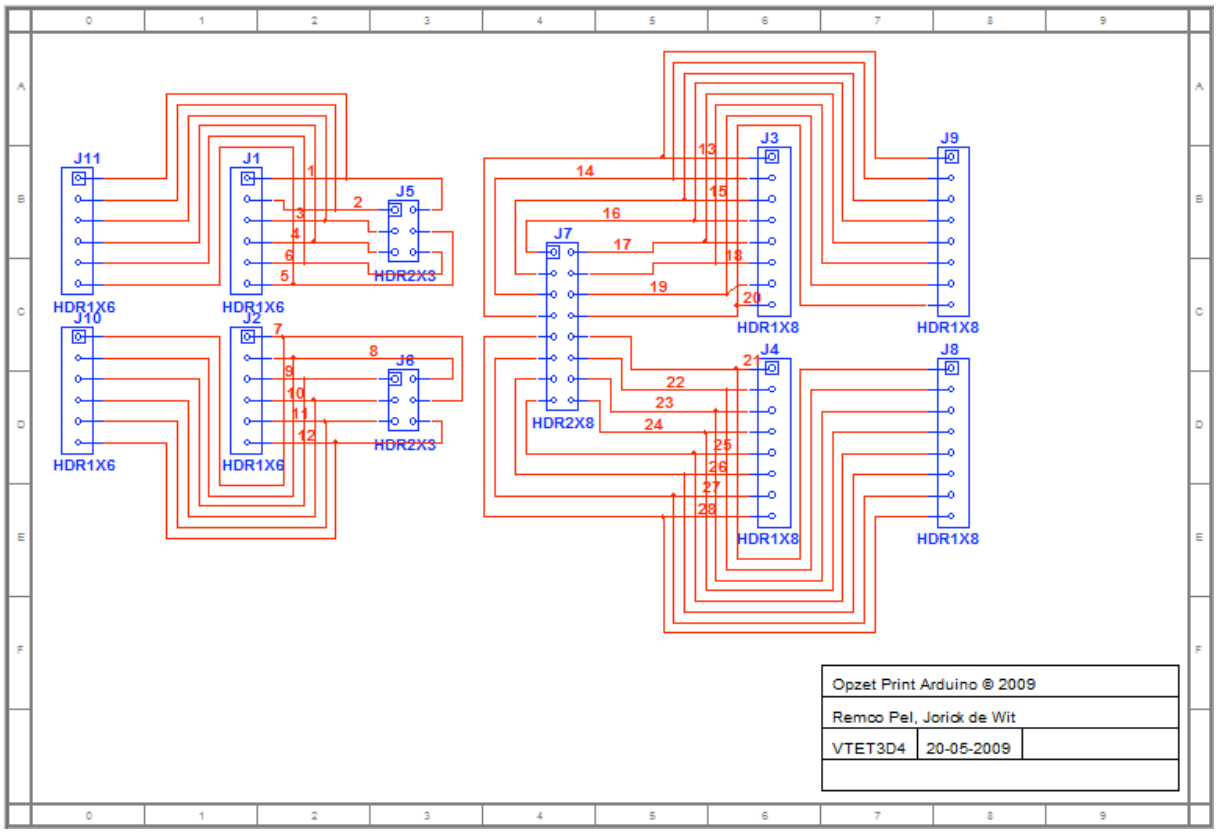
Deze schakeling is bedoeld om de RFID reader makkelijk aan te sluiten.

Accu schema:

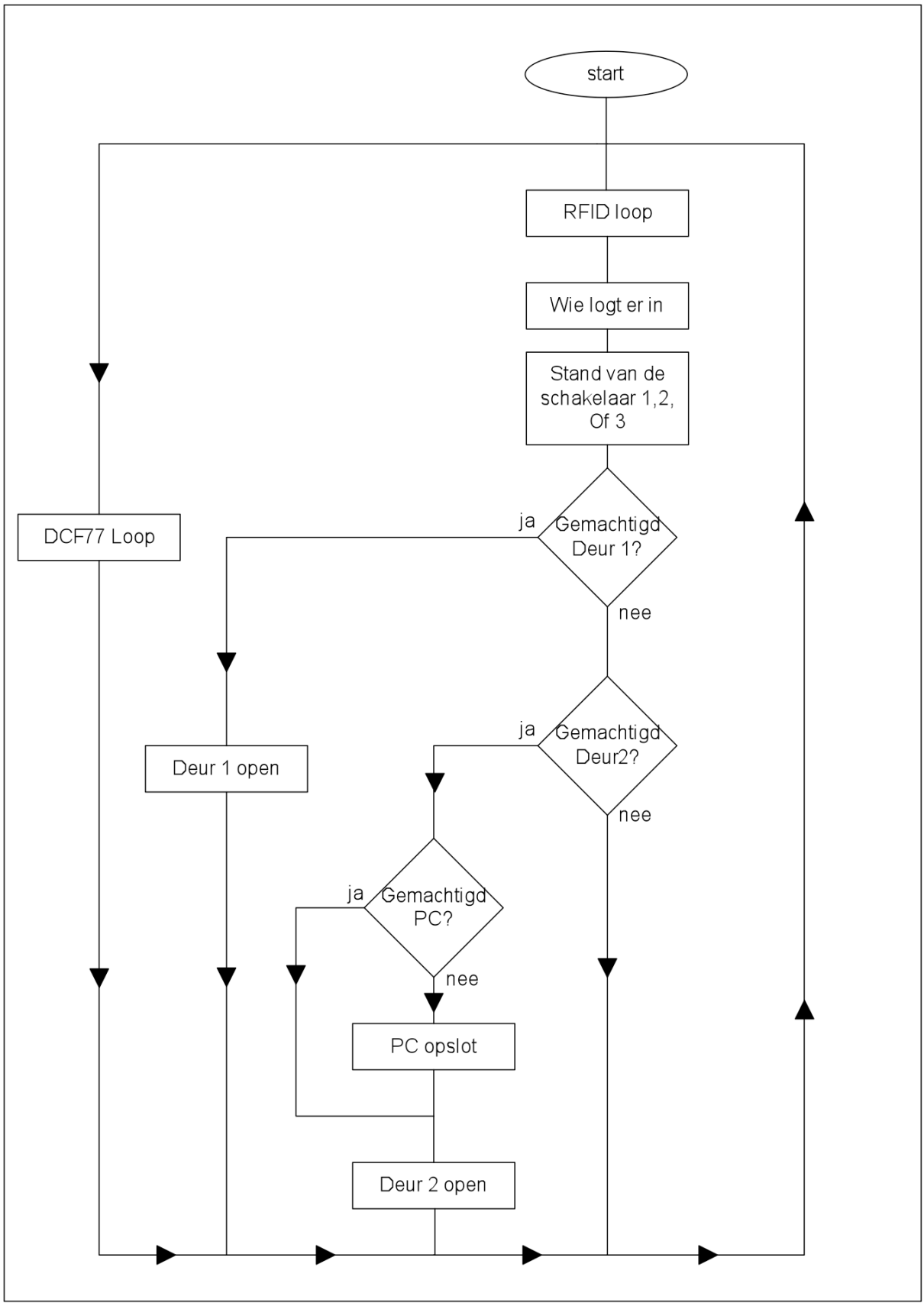


Dit schema is bedoeld voor het opladen van de accu, tevens zorgt deze schakeling er voor dat als de netspanning afvalt zal de schakeling zorgen dat die overschakelt op de accu.
(zie cd rom voor Multisim schema’s)

Door tijdgebrek hebben we dit niet kunnen testen.



Opzet print:
Deze schakeling is bedoeld voor een opzetprint die we op de arduino kunnen zetten. Met deze print kunnen we dan heel makkelijk verbinding met een flatcable maken met de rest van de printen.

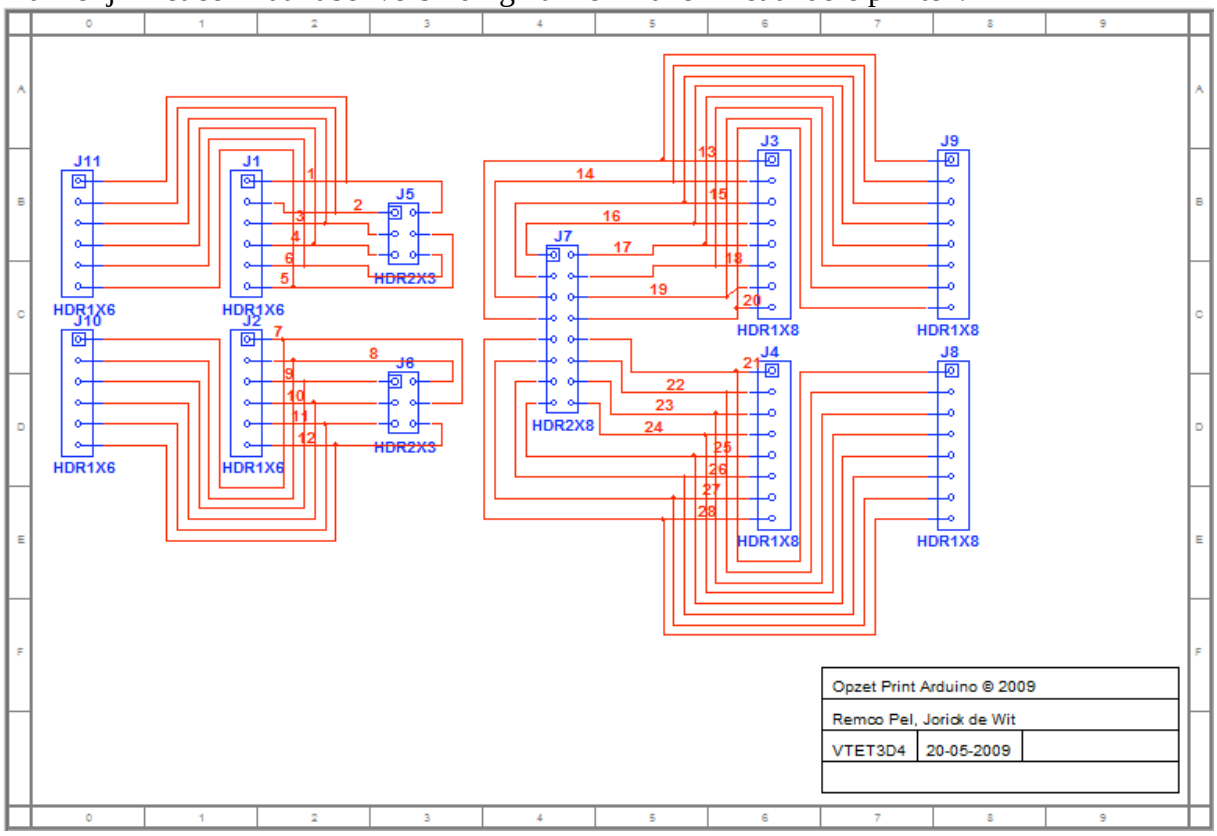


Op bovenstaande flowchart hebben we aan gegeven hoe ons programma gaat werken.

Bouw fase

De volgende schema's die we hebben bedacht hebben we voor zover mogelijk eerst getest of deze werken. En vervolgens uitgewerkt tot een printje. Nu gaan we alle schema's en bijbehorende printjes uitleggen.

We zijn begonnen met het maken van een opzet print voor op de arduino zodat we heel makkelijk met een flat kabel verbinding kunnen maken met andere printen.



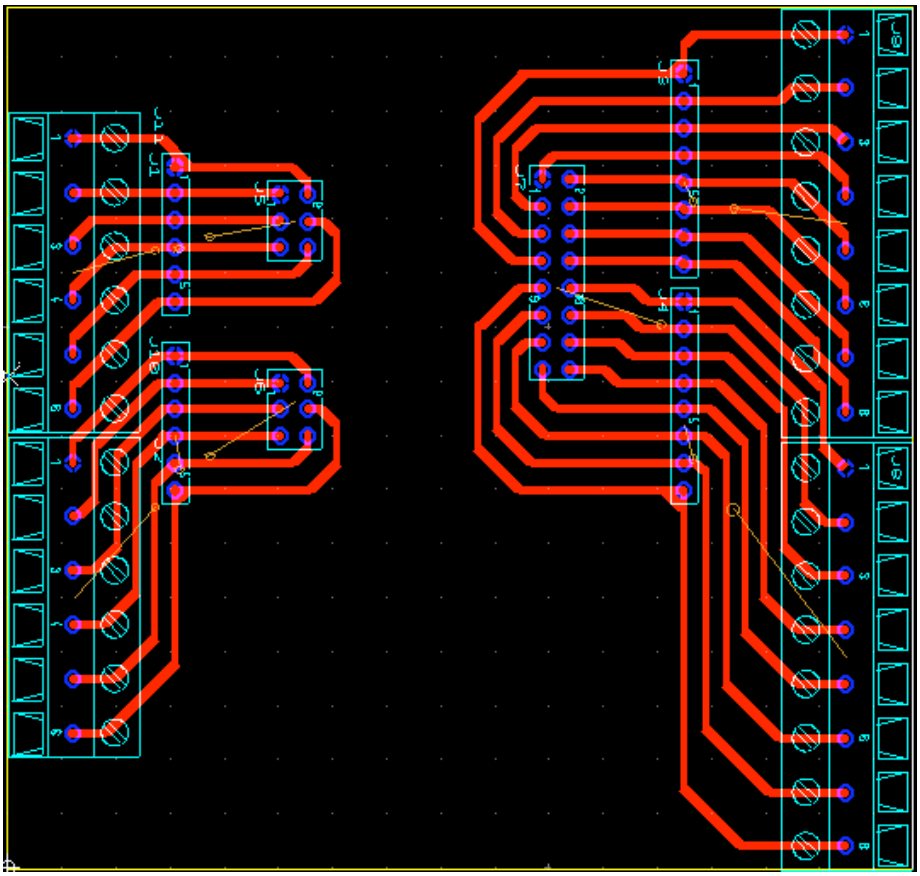
Op de volgende bladzijde zien we dus het ontwerp van het opzet printje. Deze hebben we laten vrezen en vervolgens in elkaar gesoldeerd.

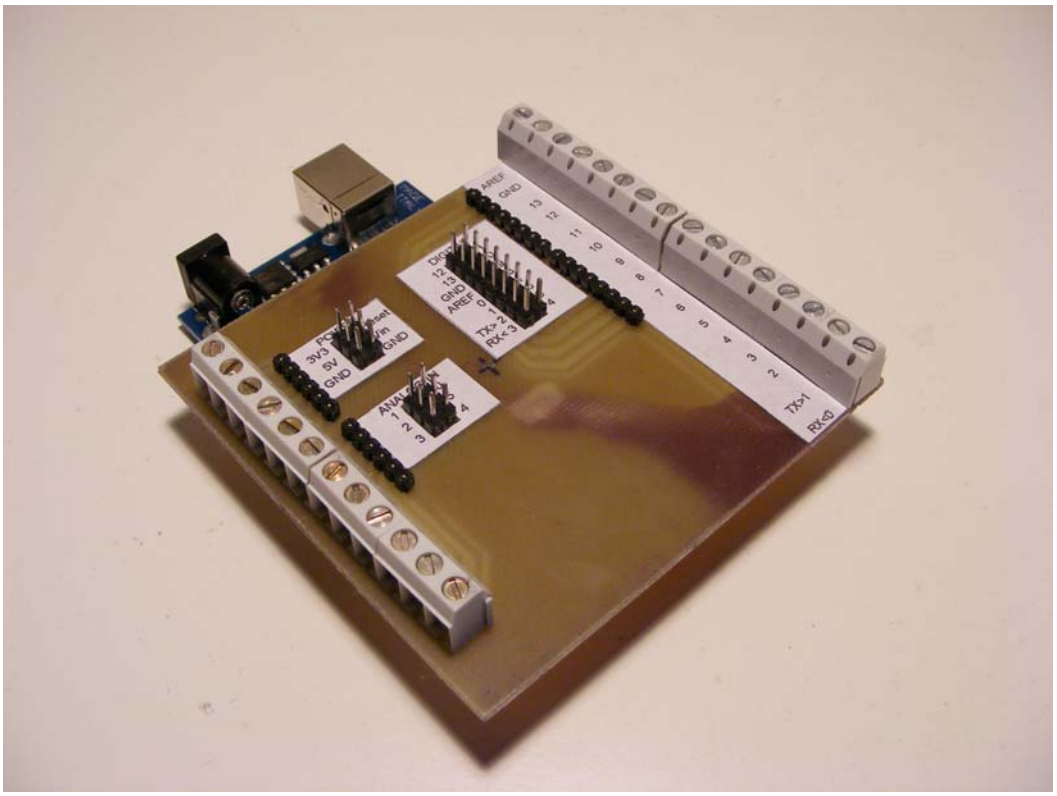
J1, J2, J3 en J4 zijn hdr strips die in de female connector van de arduino zullen schuiven.

J8, J9 J10 en J11 zullen print kroon steentjes worden zodat je makkelijk een paar draden kunt aansluiten. Op deze manier weet je altijd dat je draden goed vast zitten en niet los kunnen schieten.

J7 is een 2*8 header deze hebben we er opgezet zodat we met een flatcable verbinding kunnen maken met andere printjes.

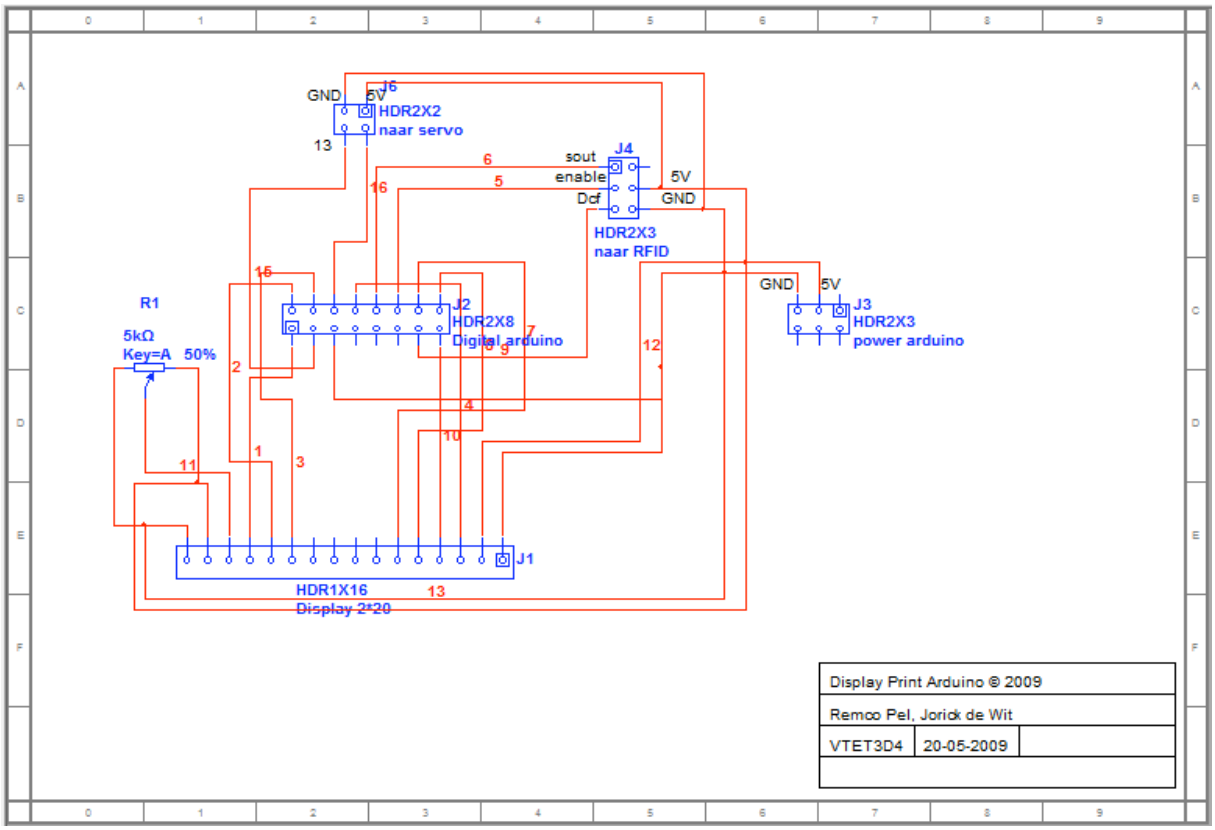
J1 en J2 zijn 2*3 header, op J1 kunnen we 5V en de GND gebruiken. J2 is voor de analoge ingangen en voor Ic2



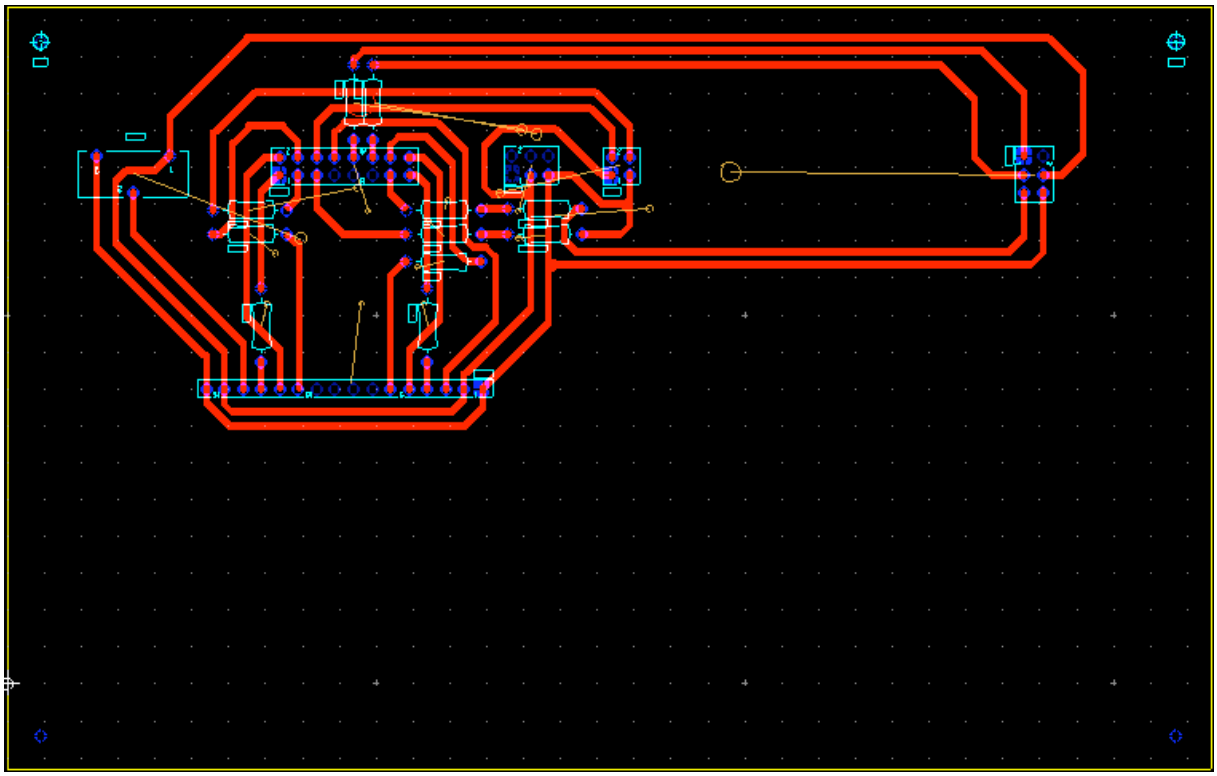


Op bovenstaande afbeelding zie je dus het opzet printje die we hebben gemaakt. Ook hebben we hiervoor stickers gemaakt zodat je makkelijk kan zien waar je wat op aan moet sluiten.

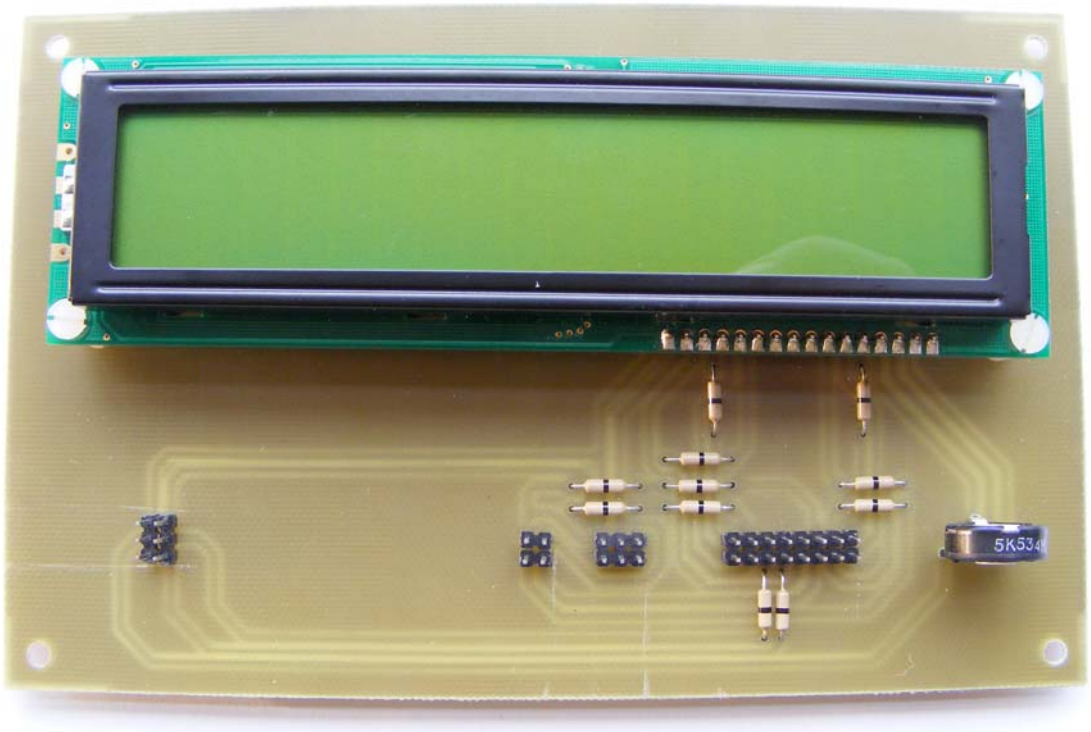
Display print



Het volgende schema is gemaakt om het display aan te sluiten.
J1 hier wordt het display op aan gesloten.
J2 hier komt de data binnen via een flatcable van af de arduino.
J3 hier komt de 5V en de GND van de arduino af.
J4 deze connector verbind de display print met de RFID print.
J5 deze connector gaat naar de servo’s toe.

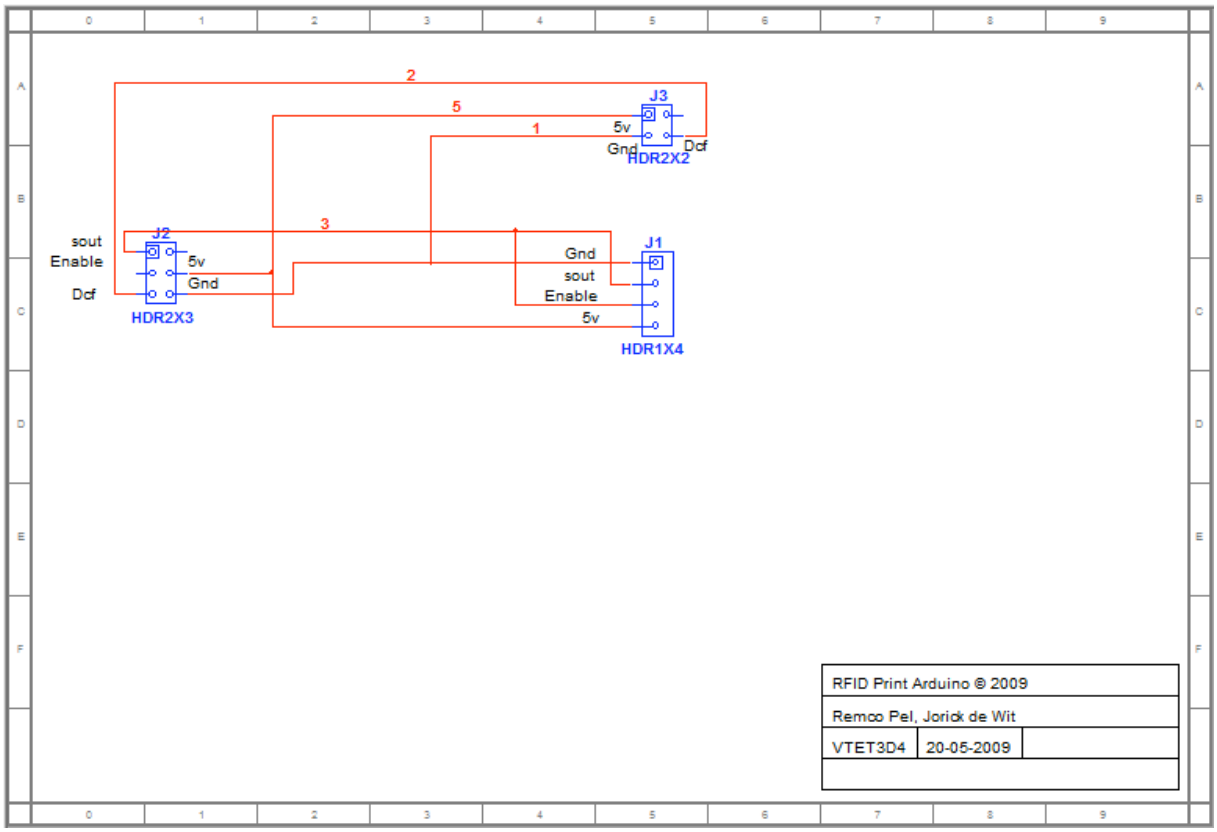


Hier boven zie je de display print (zie vorige blz voor uitleg connectors)



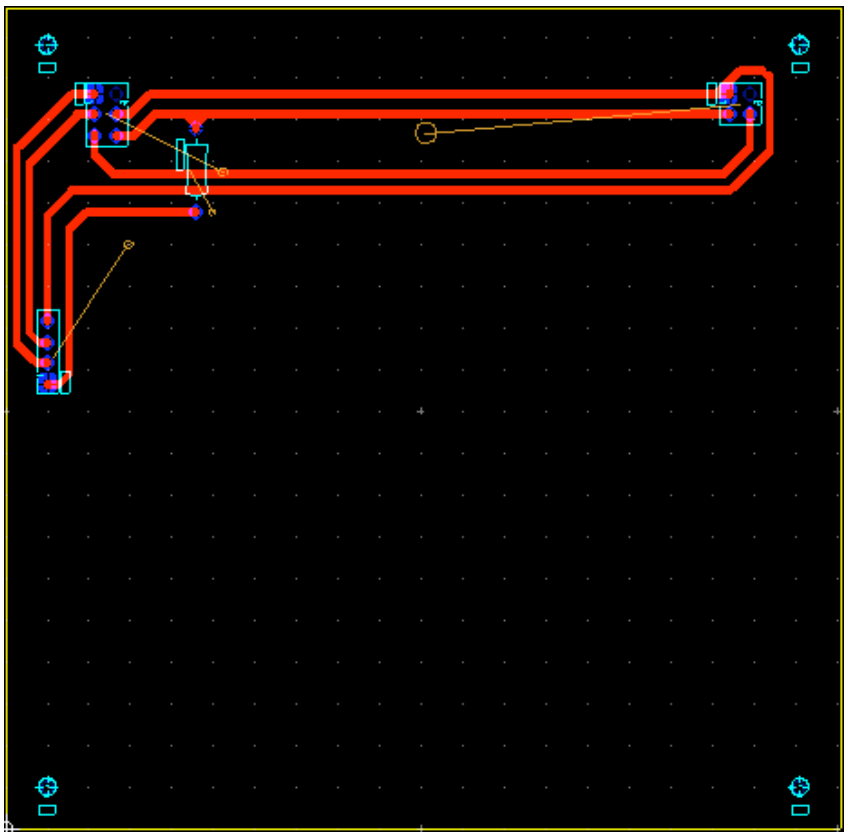
Op bovenstaande afbeelding is de display print te zien.

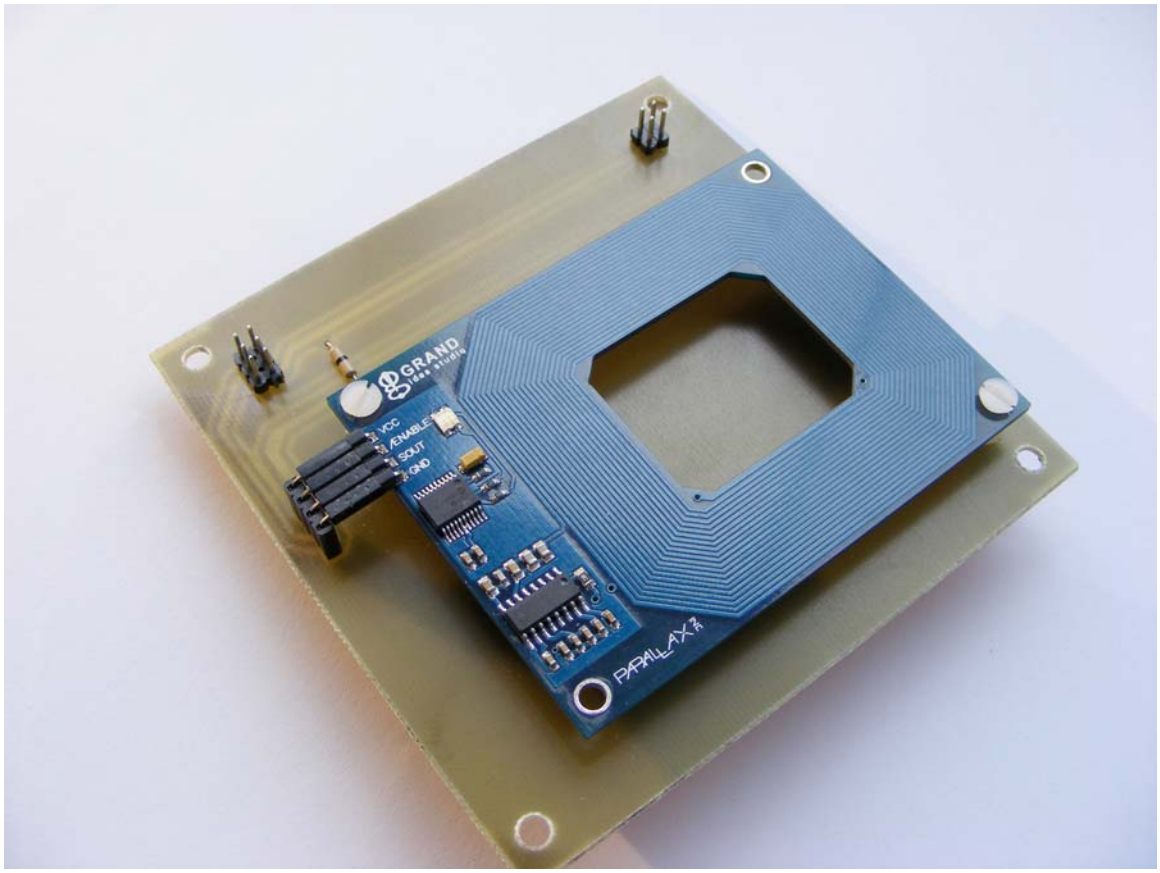
RFID print



Bovenstaande afbeelding is het schema voor de RFID print deze stelt niet zo veel voor omdat er alleen maar een aantal connectors op zitten

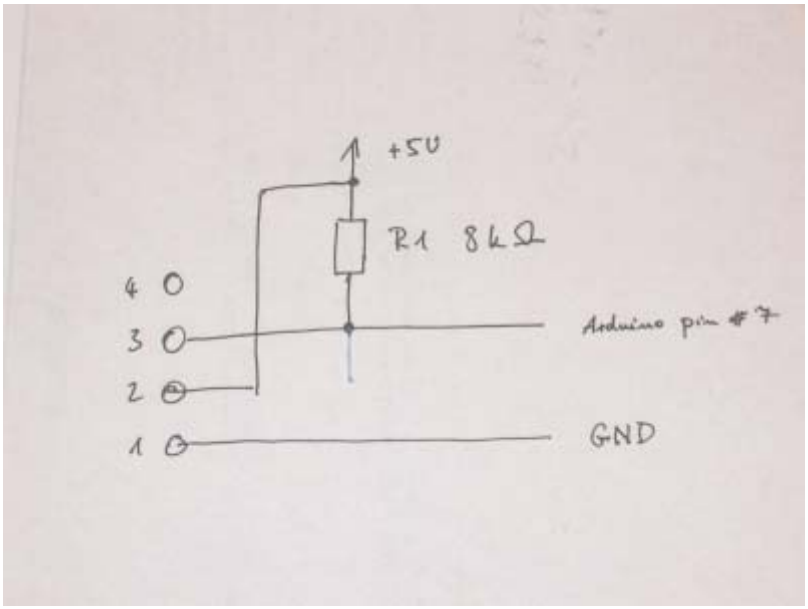
J1 hier zal de RFID reader op aangesloten worden.
J2 komt van de display print af
J3 gaat naar de DCF ontvanger toe.



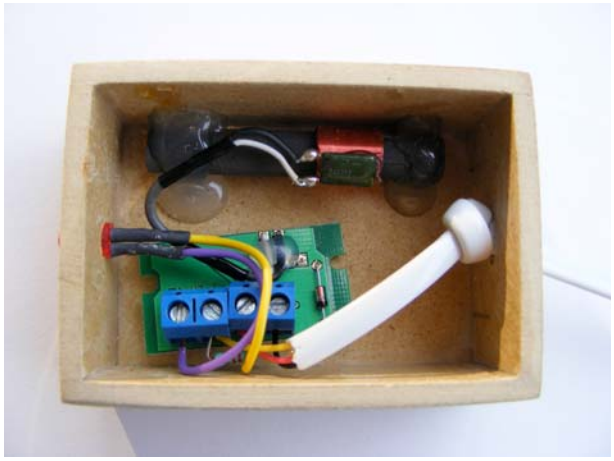


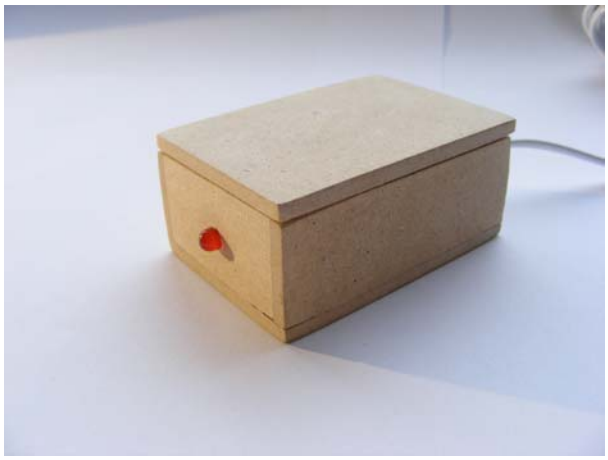
Op bovenstaande afbeeldingen zie je het Ultiboard ontwerp en de uitgewerkte verzie van de RFID print.

DCF ontvanger
Voor de DCF ontvanger hebben we geen printje ontworpen omdat dat niet nodig is.
We hebben de ontvanger in een kastje geplaatst met 5 meter kabel er aan, dit hebben we gedaan zodat de ontvanger niet dicht bij de rest van het project licht.



Op de bovenstaande afbeelding is te zien hoe we de dcf ontvanger hebben aangesloten.





Op bovenstaande afbeeldingen is het kastje te zien met daar in de DCF ontvanger

Gebruikte onderdelen

Voor dit project hebben we een aantal componenten gebruikt. Voor de aansturing maken we gebruik van een arduino.

Opzet print

Voor deze print hebben we de volgende componenten gebruikt.

Hdr kroonsteen 1*3	4 stuks
Hdr kroonsteen 1*2	8 stuks
Hdr strip	56 pins

Display print

Voor deze print hebben we de volgende componenten gebruikt.

Hdr strip	32 pins
Hdr strip Female	16 pins
Pot meter 5K	1 stuks
Display 2*20 regels	1 stuks

RFID print

Voor deze print hebben we de volgende componenten gebruikt.

Hdr strip	10 pins
Hdt strip 1*4 female	1 stuks
Hdr strip 1*4 female haaks	1 stuks
RFID reader Paralex	1 stuks
Tags	5 stuks

DCF ontvanger

Dcf ontvanger conrad	1 stuks
5 Meter telefoon kabel	1 stuks

Programma’s

RFID Programma

```
//LCD
#include <LiquidCrystal.h>                          // libary voor een LCD sherm toevoegen

LiquidCrystal lcd(12, 11, 10, 5, 4, 3, 8);          // declareren van de pinnen die voor het lcd scherm worden gebruikt

#include <SoftwareSerial.h>                         // libary om virtueel een RX en TX te maken i.p.v pin 0 en 1
#include <string.h>                                 // libary om strings te kunnen gebruiken voor RFID
#include <Servo.h>                                  // libary om servo's te kunnen aansturen

//RFID variabelen
int  val = 0;                                       // declareer val en maak hem 0
int bytesread = 0;                                  // declareer bytesread en maak hem 0
int result;                                         // uitkomst van de vergelijking a_alex - a_code
int result1;                                        // uitkomst van de vergelijking a_remco - a_code
int result2;                                        // uitkomst van de vergelijking a_jorick - a_code
int result3;                                        // uitkomst van de vergelijking a_kompanje - a_code

char a_code[10];
char a_remco[] = "38002124FA";                      // de code van de tag die ontvangen wordt door de RFIDreader
char a_alex[] = "17007E7ED1";                       // de code van de tag die ontvangen wordt door de RFIDreader
char a_jorick[] = "17007E6EF4";                     // de code van de tag die ontvangen wordt door de RFIDreader
char a_kompanje[] = "38002247C5";                   // de code van de tag die ontvangen wordt door de RFIDreader


#define rxPin 7                                     // RFID reader SOUT pin verbonden met Serial RX pin op 2400bps aan
pin7
#define txPin 6                                     // Wordt niet gebruikt maar moet wel gedeclareed worden


//SERVO variabelen
Servo servo1;                                       // maak een servo object om een servo te kunnen aansturen
int waarde;                                         // waarde voor een vergelijking bij de servo loops
int graden;                                         // waarde die de graden aangeeft die de servo moet draaien


void setup()
{
 Serial.begin(9600);                                // Hardware serial voor Monitor 9600bps

waarde = 0;                                         // zet waarde naar 0;
graden = 0;                                         // zet graden naar 0;

lcd.setCursor( 0, 0);                               // zet de cursor naar rij 0 kolom 0 op de het lcd scherm
lcd.print("Gebrnm:");                               // print op rij 0 Gebrnm:
servo1.attach(9);                                   // koppeld de servo op pin 9 aan het servo object
}


void loop()
{
 SoftwareSerial RFID = SoftwareSerial(rxPin,txPin); // het aanmaken van de variabele RFID (aan de virtuele rx en tx pin)
 RFID.begin(2400);                                  // begin met het uitlezen van de virtuele rx en tx pin

 if((val = RFID.read()) == 10)
 {                                                  // zoek naar header
  bytesread = 0;
  while(bytesread<10)
  {                                                 // lees de 10 cijferige code
   val = RFID.read();
   if(val == 10)
   {                                                // als de tag geen 10 cijferige code heeft voer dit uit
    break;                                          // stop reading
   }
   a_code[bytesread] = val;                         // voeg de cijfer/letter toe aan a_code
   bytesread++;                                     // klaar om de volgende byte te lezen
  }

  if(bytesread == 10)
  {                                                 // als alle 10 bytes zijn gelezen voer dit uit

  waarde = 0;                                       // maak "waarde" 0 voor het gebruik van de servo's
  graden = 0;                                       // maak "graden" 0 voor het gebruik van de servo's
```

```
   }


  result = memcmp(a_alex, a_code, 10);                  // vergelijk in het geheugen a_alex met a_code alle 10 de bytes
  if( result == 0 ){                                    // als de waarde gelijk is dan wordt result 0
      Serial.println("alex");                           // als result 0 is dan print "alex" naar de pc
      lcd.setCursor( 7, 0);                             // zet de cursor op rij 0 kolom 7
   lcd.print("Alex");                                   // print op die positie "Alex"
   delay(100);                                          // wacht 100ms
   lcd.setCursor(0,1);                                  // zet de cursor op rij 1 kolom 0
   lcd.print("Geen Toegang");                           // print op die positie "Geen Toegang"
  }

  result1 = memcmp(a_code, a_remco, 10);                // vergelijk in het geheugen a_remco met a_code alle 10 de bytes
  if( result1 == 0){                                    // als de waarde gelijk is dan wordt result 0
    Serial.println("remco");                            // als result 0 is dan print "remco" naar de pc
   lcd.setCursor( 7, 0);                                // zet de cursor op rij 0 kolom 7
   lcd.print("Remco");                                  // print op die positie "Remco"
   lcd.setCursor(0,1);                                  // zet de cursor op rij 1 kolom 0
   lcd.print("Welkom");                                 // print op die positie "Welkom"

while (waarde <90){                                     // zolang waard kleiner is dan 90 doe onderstaande
 if (waarde < 90){                                      // als waarde kleiner is dan 90 doe onderstaande
 if(graden >= 0){                                       // als graden gelijk of groter is dan 0 doe onderstaande
   graden++;                                            // graden = graden + 1
   servo1.write(graden);                                // schijf naar de servo de "graden" die hij moet draaien
   delay(50);                                           // wacht 50ms
   waarde = graden;                                     // waarde is gelijk aan graden
  }
  }
  if(graden == 90){                                     // als graden gelijk is aan 90 dan doe onderstaande
   delay(500);                                          // wacht 500ms
 }

 if (waarde == 90){                                     // als waarde gelijk is dan 90 doe onderstaande
   graden--;                                            // graden = graden - 1
    servo1.write(graden);                               // schijf naar de servo de "graden" die hij moet draaien
    delay(50);                                          // wacht 50ms
   }
    if (graden == 0){                                   // als graden is gelijk aan 0 dan doe onderstaande
     waarde = 89;                                       // geef "waarde" de waarde 89
 }
 if (graden == 0){                                      // als graden gelijk is aan 0 dan doe onderstaande
  delay(500);                                           // wacht 500ms
 }
  }
  }

  result2 = memcmp(a_code, a_jorick, 10);               // vergelijk in het geheugen a_jorick met a_code alle 10 de bytes
   if( result2 == 0){                                   // als de waarde gelijk is dan wordt result 0
     Serial.println("Jorick");                          // als result 0 is dan print "jorick" naar de pc
     lcd.setCursor( 7, 0);                              // zet de cursor op rij 0 kolom 7
     lcd.print("Jorick");                               // print op die positie "Jorick
     lcd.setCursor(0,1);                                // zet de cursor op rij 1 kolom 0
     lcd.print("Welkom");                               // print op die positie "Welkom"

while (waarde <90){                                     // zolang waard kleiner is dan 90 doe onderstaande
 if (waarde < 90){                                      // als waarde kleiner is dan 90 doe onderstaande
 if(graden >= 0){                                       // als graden gelijk of groter is dan 0 doe onderstaande
   graden++;                                            // graden = graden + 1
   servo1.write(graden);                                // schijf naar de servo de "graden" die hij moet draaien
   delay(50);                                           // wacht 50ms
   waarde = graden;                                     // waarde is gelijk aan graden
  }
  }
  if(graden == 90){                                     // als graden gelijk is aan 90 dan doe onderstaande
   delay(500);                                          // wacht 500ms
 }

 if (waarde == 90){                                     // als waarde gelijk is dan 90 doe onderstaande
   graden--;                                            // graden = graden - 1
    servo1.write(graden);                               // schijf naar de servo de "graden" die hij moet draaien
    delay(50);                                          // wacht 50ms
   }
    if (graden == 0){                                   // als graden is gelijk aan 0 dan doe onderstaande
     waarde = 89;                                       // geef "waarde" de waarde 89
```

```
 }
 if (graden == 0){                                          // als graden gelijk is aan 0 dan doe onderstaande
  delay(500);                                               // wacht 500ms
 }
  }
  }

    result3 = memcmp(a_code, a_kompanje, 10);               // vergelijk in het geheugen a_kompanje met a_code alle 10 de bytes
  if( result3 == 0){                                        // als de waarde gelijk is dan wordt result 0
   Serial.println("Kompanje");                              // als result 0 is dan print "Kompanje" naar de pc
   lcd.setCursor( 7, 0);                                    // zet cursor op rij 0 kolom 7
   lcd.print("Kompanje");                                   // print op die positie "Kompanje"
   Serial.println(101,BYTE);                                // code voor het aansturen van logoff van de laptop
   lcd.setCursor(0,1);                                      // zet cursor op rij 1 kolom 0
   lcd.print("Geen Toegang tot PC");                        // print op die positie "Geen Toegang tot PC"
   lcd.setCursor(0,1);                                      // zet cursor op rij 1 kolom 0
   delay(2500);                                             // wacht 2500ms
   lcd.print("                ");                          // maak de 2 karakters breede lcd scherm leeg (rij 1, kolom 0 - 20)
   lcd.setCursor(0,1);                                      // zet cursor op rij 1 kolom 0
   lcd.print("Welkom");                                     // print op die positie "Welkom"

while (waarde <90){                                         // zolang waard kleiner is dan 90 doe onderstaande
 if (waarde < 90){                                          // als waarde kleiner is dan 90 doe onderstaande
 if(graden >= 0){                                           // als graden gelijk of groter is dan 0 doe onderstaande
   graden++;                                                // graden = graden + 1
   servo1.write(graden);                                    // schijf naar de servo de "graden" die hij moet draaien
   delay(50);                                               // wacht 50ms
   waarde = graden;                                         // waarde is gelijk aan graden
  }
  }
  if(graden == 90){                                         // als graden gelijk is aan 90 dan doe onderstaande
   delay(500);                                              // wacht 500ms
 }

 if (waarde == 90){                                         // als waarde gelijk is dan 90 doe onderstaande
   graden--;                                                // graden = graden - 1
    servo1.write(graden);                                   // schijf naar de servo de "graden" die hij moet draaien
    delay(50);                                              // wacht 50ms
   }
    if (graden == 0){                                       // als graden is gelijk aan 0 dan doe onderstaande
     waarde = 89;                                           // geef "waarde" de waarde 89
 }
 if (graden == 0){                                          // als graden gelijk is aan 0 dan doe onderstaande
  delay(500);                                               // wacht 500ms
 }
  }
  }

  if ((result != 0) && (result1 != 0) && (result2 != 0) && (result3 != 0)){    // als bij result 0 t/m 3 overal geen 0 uit komt dan doe
onderstaande
    Serial.println("Geen Toegang");                         // print "Geen Toegang" naar de pc
   lcd.setCursor( 7, 0);                                    // zet cursor op rij 0 kolom 7
   lcd.print("Ongeldige Pas");                              // print op die positie "ongeldige pas"
   lcd.setCursor(0,1);                                      // zet cursor op rij 1 kolom 0
   lcd.print("Geen Toegang");                               // print op die positie "Geen Toegang"
  }
   bytesread = 0;                                           // maak bytesread 0
   delay(2500);                                             // wacht 2500ms
   lcd.setCursor( 7, 0);                                    // zet cursor op rij 0 kolom 7
   lcd.print("            ");                               // maak rij 0 leeg van de LCD vanaf kolom 7
   lcd.setCursor(0,1);                                      // zet cursor op rij 1 kolom 0
   lcd.print("                ");                          // maak rij 1 helemaal leeg
 }
}
```

Dcf programma

```
#include <LiquidCrystal.h>

#define DCF77PIN 2                    // Input for the DCF receiver
#define BLINKPIN 13                   // LED indicator output
#define DCF_split_millis 140          // Number of milliseconds before we assume a logic 1
#define DCF_sync_millis 1200          // No signal at second 59

// Definitions for the timer interrupt 2 handler
// The Arduino runs at 16 Mhz, we use a prescaler of 64 -> We need to
// initialize the counter with 6. This way, we have 1000 interrupts per second.
// We use tick_counter to count the interrupts.
LiquidCrystal lcd(12, 11, 10, 5, 4, 3, 8);
#define INIT_TIMER_COUNT 6
#define RESET_TIMER2 TCNT2 = INIT_TIMER_COUNT
int tick_counter = 0;

                                       // DCF time format struct
struct DCF77Buffer {
  unsigned long long prefix   :21;
  unsigned long long Min      :7;    // minutes
  unsigned long long P1               :1;    // parity minutes
  unsigned long long Hour     :6;    // hours
  unsigned long long P2               :1;    // parity hours
  unsigned long long Day      :6;    // day
  unsigned long long Weekday  :3;    // day of week
  unsigned long long Month    :5;    // month
  unsigned long long Year     :8;    // year (5 -> 2005)
  unsigned long long P3               :1;    // parity
};

                                          // Parity struct
struct {
  unsigned char parity_flag   :1;
  unsigned char parity_min    :1;
  unsigned char parity_hour   :1;
  unsigned char parity_date   :1;
} flags;

                                          // Clock variables
volatile unsigned char DCFSignalState = 0;
unsigned char previousSignalState;
int previousFlankTime;
int bufferPosition;
unsigned long long dcf_rx_buffer;

                                          // Time variables
volatile unsigned char ss;
volatile unsigned char mm;
volatile unsigned char hh;
volatile unsigned char day;
volatile unsigned char mon;
volatile unsigned int year;

unsigned char previousSecond;

                                          // Initialize the DCF77 routines: initialize the
variables,
                                          // configure the interrupt behaviour.
void DCF77Init() {
  previousSignalState=0;
  previousFlankTime=0;
  bufferPosition=0;
  dcf_rx_buffer=0;
  ss=mm=hh=day=mon=year=0;
  pinMode(DCF77PIN, INPUT);
                                          //Timer2 Settings: Timer Prescaler /64,
  TCCR2B |= (1<<CS22);                    // turn on CS22 bit
  TCCR2B &= ~((1<<CS21) | (1<<CS20));     // turn off CS21 and CS20 bits
                                          // Use normal mode
  TCCR2A &= ~((1<<WGM21) | (1<<WGM20));   // turn off WGM21 and WGM20 bits
  TCCR2B &= ~(1<<WGM22);                  // turn off WGM22
                                          // Use internal clock - external clock not used in
Arduino
  ASSR |= (0<<AS2);
```

```
  TIMSK2 |= (1<<TOIE2) | (0<<OCIE2A);     //Timer2 Overflow Interrupt Enable
  RESET_TIMER2;
  attachInterrupt(0, int0handler, CHANGE);
}

// Append a signal to the dcf_rx_buffer. Argument can be 1 or 0. An internal
// counter shifts the writing position within the buffer. If position > 59,
// a new minute begins -> time to call finalizeBuffer().
void appendSignal(unsigned char signal) {
  dcf_rx_buffer = dcf_rx_buffer | ((unsigned long long) signal << bufferPosition);
  // Update the parity bits. First: Reset when minute, hour or date starts.
  if (bufferPosition ==  21 || bufferPosition ==  29 || bufferPosition ==  36) {
       flags.parity_flag = 0;
  }
  // save the parity when the corresponding segment ends
  if (bufferPosition ==  28) {flags.parity_min = flags.parity_flag;};
  if (bufferPosition ==  35) {flags.parity_hour = flags.parity_flag;};
  if (bufferPosition ==  58) {flags.parity_date = flags.parity_flag;};
  // When we received a 1, toggle the parity flag
  if (signal == 1) {
    flags.parity_flag = flags.parity_flag ^ 1;
  }
  bufferPosition++;
  if (bufferPosition > 59) {
    finalizeBuffer();
  }
}

// Evaluates the information stored in the buffer. This is where the DCF77
// signal is decoded and the internal clock is updated.
void finalizeBuffer(void) {
  if (bufferPosition == 59) {
    struct DCF77Buffer *rx_buffer;
    rx_buffer = (struct DCF77Buffer *)(unsigned long long)&dcf_rx_buffer;
    if (flags.parity_min == rx_buffer->P1  &&
        flags.parity_hour == rx_buffer->P2  &&
        flags.parity_date == rx_buffer->P3)
    {
      //convert the received bits from BCD
      mm = rx_buffer->Min-((rx_buffer->Min/16)*6);
      hh = rx_buffer->Hour-((rx_buffer->Hour/16)*6);
      day= rx_buffer->Day-((rx_buffer->Day/16)*6);
      mon= rx_buffer->Month-((rx_buffer->Month/16)*6);
      year= 2000 + rx_buffer->Year-((rx_buffer->Year/16)*6);
    }
  }
  // reset stuff
  ss = 0;
  bufferPosition = 0;
  dcf_rx_buffer=0;
}

// Evaluates the signal as it is received. Decides whether we received
// a "1" or a "0" based on the
void scanSignal(void){
    if (DCFSignalState == 1) {
      int thisFlankTime=millis();
      if (thisFlankTime - previousFlankTime > DCF_sync_millis) {
        finalizeBuffer();
      }
      previousFlankTime=thisFlankTime;
    }
    else {
      /* or a falling flank */
      int difference=millis() - previousFlankTime;
      if (difference < DCF_split_millis) {
        appendSignal(0);
      }
      else {
        appendSignal(1);
      }
    }
}

// The interrupt routine for counting seconds - increment hh:mm:ss.
ISR(TIMER2_OVF_vect) {
  RESET_TIMER2;
```

```
  tick_counter += 1;
  if (tick_counter == 1000) {
    ss++;
    if (ss==60) {
      ss=0;
      mm++;
      if (mm==60) {
        mm=0;
        hh++;
        if (hh==24)
          hh=0;
      }
    }
    tick_counter = 0;
  }
};

// Interrupthandler for INT0 - called when the signal on Pin 2 changes.
void int0handler() {
  // check the value again - since it takes some time to
  // activate the interrupt routine, we get a clear signal.
  DCFSignalState = digitalRead(DCF77PIN);
}

//LCD routines
void clearLCD()

{
  lcd.print(12, BYTE);
}

void startBigChars()
{
 lcd.print(2, BYTE);
}

// Dump the time to the serial LCD

void DumpTime(void){

/*  if (year == 0) {
   lcd.print(bufferPosition);
    lcd.setCursor(0,1);
    lcd.print("  ");
    lcd.setCursor( 0, 0);
  } else {
    lcd.print("  ");
    lcd.setCursor( 0, 0);
    if (day < 10) {
      lcd.print(" ");
      lcd.setCursor( 0,0);
    }
 }*/

// de dagen wegschijven naar het display
  if (day == 0) {
    lcd.setCursor( 0,1);
    lcd.print("00");

// tussen streepje plaatsen op display
    lcd.setCursor(2,1);
    lcd.print("-");
    }else{
      if (day <= 9){
        lcd.setCursor( 0,1);
        lcd.print("0");
        lcd.setCursor( 1,1);
        lcd.print(day, DEC);
      }else{
        lcd.setCursor( 0,1);
        lcd.print(day, DEC);
      }

// tussen streepje plaatsen op display
  lcd.setCursor(2,1);
  lcd.print("-");
  }
```

```
// de maanden wegschijven naar het display
    if (mon == 0) {
    lcd.setCursor(3,1);
    lcd.print("00");

// tussen streepje plaatsen op display
    lcd.setCursor(5,1);
    lcd.print("-");
  }else{
      if (mon <= 9){
        lcd.setCursor( 3,1);
        lcd.print("0");
        lcd.setCursor( 4,1);
        lcd.print(mon, DEC);
      }else{
        lcd.setCursor( 3,1);
        lcd.print(mon, DEC);
      }

// tussen streepje plaatsen op display
  lcd.setCursor(5,1);
  lcd.print("-");
  }

// de jaren wegschijven naar het display
  if (year == 0) {
    lcd.setCursor(6,1);
    lcd.print("0000");
  }else{
  lcd.setCursor(6,1);
  lcd.print(year, DEC);
  }


// de uren weergeven op het display
  if (hh == 0) {
    lcd.setCursor( 12, 1);
    lcd.print("00");
  }
    lcd.setCursor( 12, 1);
  if (hh <= 9) {
    lcd.print("0");
    lcd.setCursor( 13, 1);
    lcd.print(hh, DEC);
  }else{
    lcd.setCursor( 12, 1);
    lcd.print(hh, DEC);
  }

 // per seconde de : laten knipperen
  if ((ss % 2) == 0) {
    lcd.setCursor( 14, 1);
    lcd.print(":");
  } else {
    lcd.print(" ");
  }

// de minuten weergeven op het display
  if (mm == 0) {
    lcd.setCursor( 15, 1);
    lcd.print("00");
  }
    lcd.setCursor( 15, 1);
  if (mm <= 9) {
    lcd.print("0");
    lcd.setCursor( 16, 1);
    lcd.print(mm, DEC);
  }else{
    lcd.setCursor( 15, 1);
    lcd.print(mm, DEC);
  }

// per seconde de : laten knipperen
  if ((ss % 2) == 0) {
    lcd.setCursor( 17, 1);
    lcd.print(":");
```

```
  } else {
    lcd.print(" ");
  }


// secondes weergeven op display
  if (ss == 0) {
    lcd.setCursor( 18, 1);
    lcd.print("00");
  }
    lcd.setCursor( 18, 1);
  if (ss <= 9) {
    lcd.print("0");
    lcd.setCursor( 19, 1);
    lcd.print(ss, DEC);
  }else{
    lcd.setCursor( 18, 1);
    lcd.print(ss, DEC);
  }

// alles weergeven in arduino programma op pc
  Serial.print(day,DEC);
  Serial.print("-");
  Serial.print(mon,DEC);
  Serial.print("-");
  Serial.print(year,DEC);

  Serial.print("   ");

  Serial.print(hh,DEC);
  Serial.print(":");
  Serial.print(mm,DEC);
  Serial.print(":");
  Serial.print(ss,DEC);

  Serial.print("   ");

  Serial.println(digitalRead(DCF77PIN));
}

// Standard Arduino methods below.
void setup(void) {
  // We need to start serial here again,
  // for Arduino 007 (new serial code)
  Serial.begin(9600);
  clearLCD();
  DCF77Init();
  lcd.setCursor(0,0);
  lcd.print("Atoomtijd:");

}

void loop(void) {

  if (ss != previousSecond) {
    DumpTime();
    previousSecond = ss;
  }
  if (DCFSignalState != previousSignalState) {
    scanSignal();

    if (DCFSignalState) {
      digitalWrite(BLINKPIN, HIGH);
    } else {
      digitalWrite(BLINKPIN, LOW);
    }
    previousSignalState = DCFSignalState;
  }
}
```

Bovenstaand DCF programma hebben we grotendeels van internet gehaald. Deze hebben we aan moeten passen voor ons display.

Visual Basic Programma

De lay-out:

Het programma:

```
Public Class Form1
    Dim run As Boolean
    Dim Str As Byte
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

        OVHverb.BackStyle = PowerPacks.BackStyle.Opaque
        SerialPort1.Open() 'open een seriële Port.
        OVHverb.BackColor = Color.Green
        Timer1.Enabled = True

    End Sub
    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Timer1.Tick

        Str = SerialPort1.ReadByte() 'lees wat er verzonden wordt

        If Str = 101 Then 'als de gelezen byte 101 is dan:

            OVHverb.BackColor = Color.Red
            SerialPort1.Close() 'sluit de communicatie met com poort

            Dim RetVal
            RetVal = Shell("C:\LogOff.bat", AppWinStyle.Hide) ' open .bat file
            Me.Close() ' sluit de form

        End If
    End Sub
End Class
```

We hebben een visual basic programma gemaakt waarmee we de computer kunnen uitloggen. dat doen we doormiddel van een code die we vanuit de arduino sturen naar de computer met het commando Serial.print(). Dat lezen we uit in visual basic met de code: **SerialPort1.ReadByte().** Vanaf de arduino sturen we de code 101 (in ascii code een “e”) het is een willekeurig gekozen code, die code lezen we dus uit met visual basic en als die code ontvangen is dan sluiten we de communicatie met de com poort om zo overige codes te blokkeren en error meldingen te voorkomen als de computer zich afmeld door het commando: **Shell("C:\LogOff.bat", AppWinStyle.Hide).** Hier mee openen we een .bat file die we hebben aangemaakt met de code: “**shutdown\l**” als die code wordt uitgevoerd wordt de laptop afgemeld en alle draaiende programma’s afgesloten. En de from sluit zichzelf ook meteen af als die file wordt geopend om een error code te voorkomen als de form geforceerd af moet sluiten.

Bijlage 1 Verantwoording