CAMPUS DE NAYER
Industriële ingenieurswetenschappen

KU LEUVEN

Katten in Katelijne

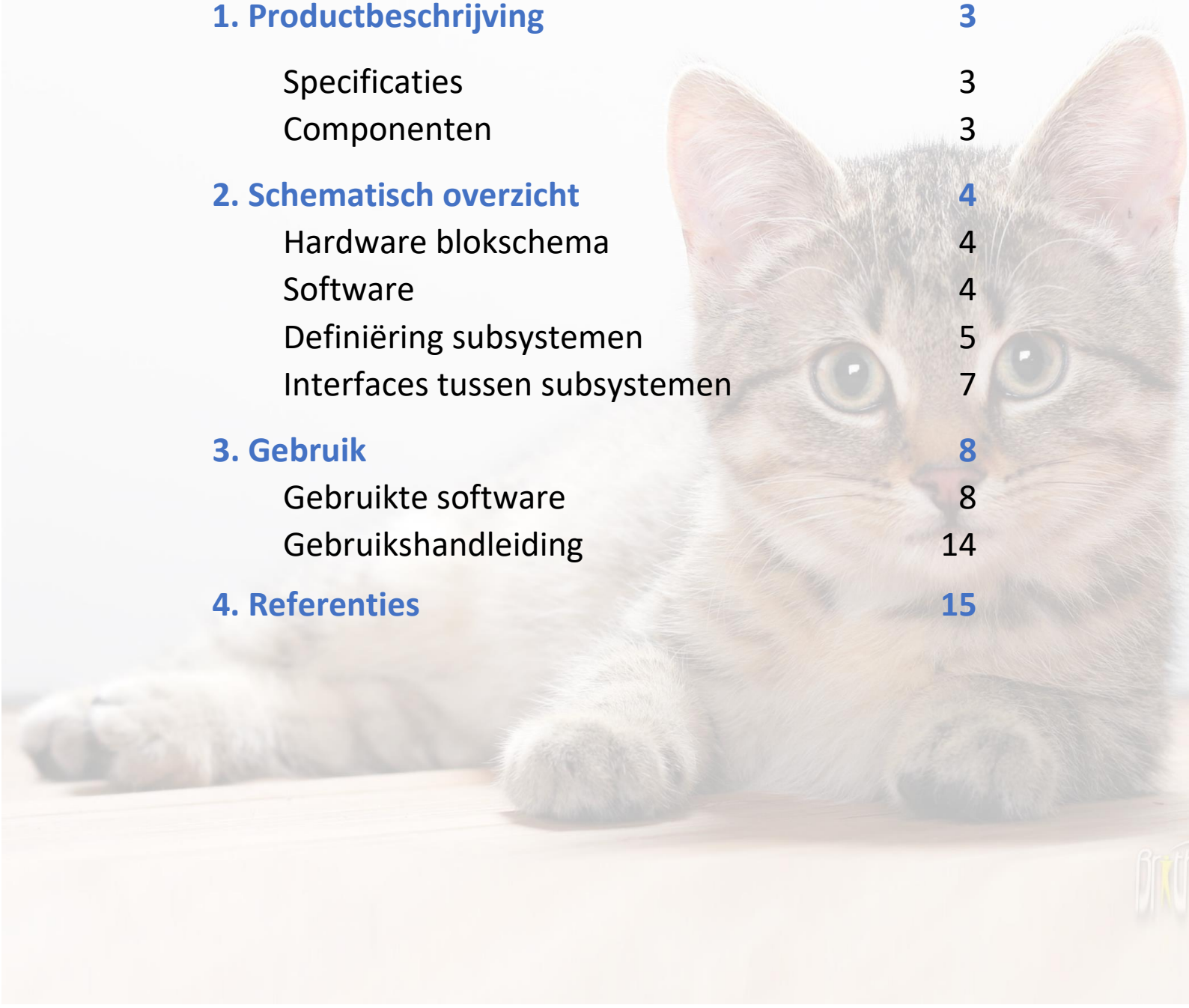
Bachelorproef E-ICT

Sofie Geens
Maurits Van Mossevelde

Mentor:
Danny Pauwels

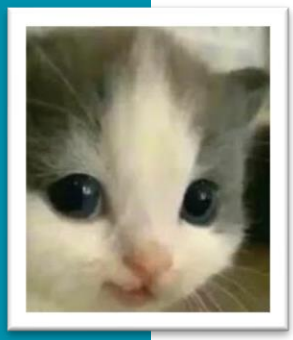
Februari - Juni 2022

Inhoudsopgave



KU LEUVEN

1. Productbeschrijving



We maken een kooi om zwerfkatten te vangen slimmer. Bij het gebruik van ordinaire draadkooien gaan veel tijd en middelen verloren aan het controleren of de kooi een dier heeft gevangen en kunnen gevangen dieren soms veel langer gevangen zitten dan wenselijk is. In dit project maken we een slimme kooi die de gemeente verwittigt als een dier gevangen wordt.

Als de kooi sluit wordt automatisch een SMS verstuurd. Het systeem is ook uitgerust met een zonnepaneel en een batterij om 's nachts en op bewolkte dagen te kunnen functioneren.

<u>Specificaties:</u>

Detectie:
- Aantal sensoren: 2
- Detectie: Magnetisch veld

Stroomvoorziening:
- Zonnecelspanning: 5V
- Zonnecelvermogen: 6W
- Batterijspanning: 3.7V
- Batterijcapaciteit: 2000mAh

Communicatie:
- Medium: 2G SMS netwerk
- Instructieset: AT

<u>Componenten:</u>

Microcontroller:
- Sparkfun edge board[1]

GSM-module:
- Adafruit Fona 808[2]

Batterij:
- Lithium ion batterij[3]

Batterij-oplader:
- Lithium ion charger[4]

Magnetische sensoren:
- Magnetic sensor[5]

Analoge AND-port:
- Type 74HC08N[6]

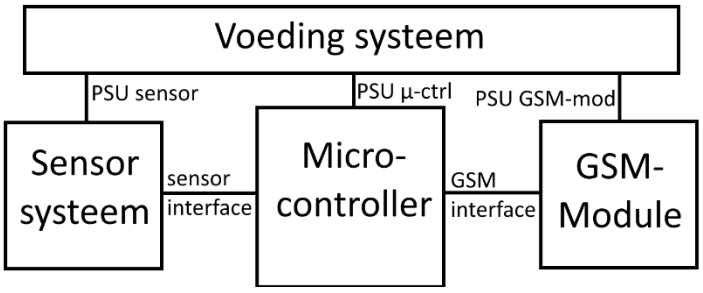
Microcontroller interface:
- Sparkfun serial basic breakout[7]

Zonnepaneel:
- 5V 6W solar panel[8]

2. Schematisch overzicht

Hardware blokschema:

De blokken zijn de verschillende subsystemen en de connecties tussen de blokken zijn de verschillende interfaces tussen subsystemen.
Ons systeem bevat 4 subsystemen en 5 interfaces.



Software:

Ons software systeem bestaat slechts uit 1 programma op de microcontroller. Dit programma luistert naar een SMS-input van de GSM-module en naar een actief laag signaal van de sensoren. Instructies kunnen aan het systeem worden gegeven door bepaalde berichten te sturen naar de GSM-module:

- ‘v’ of ‘V’: voegt je nummer toe aan de lijst van verwittigden
- ‘d’ of ‘D’: verwijdert je nummer uit de lijst van verwittigden
- ‘g’ of ‘G’: vraagt de GPS-locatie van de kooi op

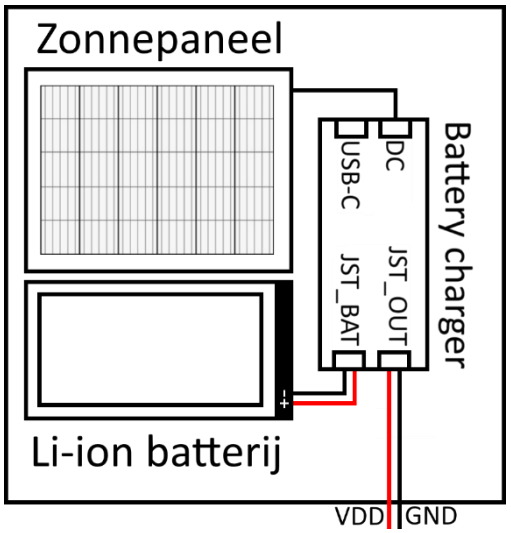
Bij het detecteren van de sensoren stuurt de microcontroller via de GSM-module dan een bericht naar alle toegevoegde nummers in de lijst.

Bij het opstarten van het systeem wordt de simkaart ontgrendeld op voorwaarde dat de PIN code is ingesteld op “1234” en worden alle ongelezen berichten verwijderd zodat geen berichten van toen de kooi uit stond invloed hebben.

Een software listing bevindt zich achteraan bij gebruikte software.

Definiëring subsystemen:

Voeding systeem

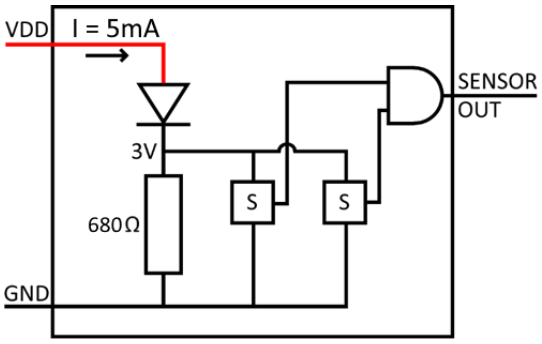


- Dit systeem simuleert een spanningsbron van 3.7V om alle andere subsystemen te kunnen voorzien van stroom. Het bestaat uit een zonnepaneel-gevoede batterij met een regulator ertussen om de batterij te beschermen van overspanning en extern ook te kunnen opladen.
- Het zonnepaneel levert stroom aan het oplaadcircuit met behulp van een DC adapter, en de batterij neemt en levert stroom aan het oplaadcircuit met een JST connector. Op het oplaadcircuit staat ook nog een extra JST connector om stroom naar buiten te brengen.
- Het systeem heeft dus geen input en een positieve spanning en ground als output.
- Het zonnepaneel levert een maximaal vermogen van 6W op een spanning van 5V, en de batterij werkt op 3.7V met een capaciteit van 2000mAh. Het systeem verbruikt dus zelf geen energie. Het levert en slaat enkel op.

Sensor systeem

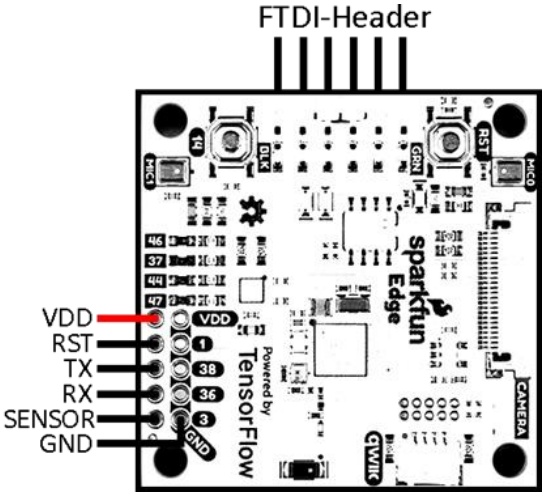
- Dit systeem stuurt een logische 0 door in de nabijheid van een magnetisch veld, en een logische 1 in de afwezigheid ervan (Actief laag). Het systeem wordt gevoed op 3.7V waardoor de logische 1 zich ook bevindt op 3.7V.
- Er wordt voor de sensor een diode met een weerstand als instelpunt voorgeschakeld om de spanning over de sensor lichtjes te reduceren om binnen de toegelaten grenswaarden te blijven. Om zeker te zijn dat de sensoren zouden afgaan bij het sluiten van de kooi gebruiken we meerdere sensoren op verschillende plaatsen gelinkt aan een AND-poort. Op deze manier zal, als slechts 1 sensor laag valt, het uitgangssignaal ook laag vallen.
- Om zo energie efficiënt mogelijk te zijn, kiezen we het instelpunt voor de diode op 1mA, waarbij 0.7V over de diode staat. Hiervoor hebben we een weerstand nodig waardoor 1mA stroomt bij de overige 3V. We krijgen een weerstand van 300Ω. Door de sensoren stroomt 6µA, dus is deze verwaarloosbaar voor het bepalen van ons instelpunt.

- Het systeem heeft een voedingsspanning en een ground als input en een digitaal signaal als output die de sensoren representeren.
- Door de schakeling stroomt te allen tijde een stroom van 5mA over een spanning van 3V. Het volledig circuit verbruikt dus slechts 15mW. De sensoren zelf vergen slechts 6µA aan stroom op 3V waardoor elke aparte sensor maar 18µW verbruiken. Het toevoegen van sensoren als redundantie veroorzaakt dus nagenoeg geen stijging in vermogen.



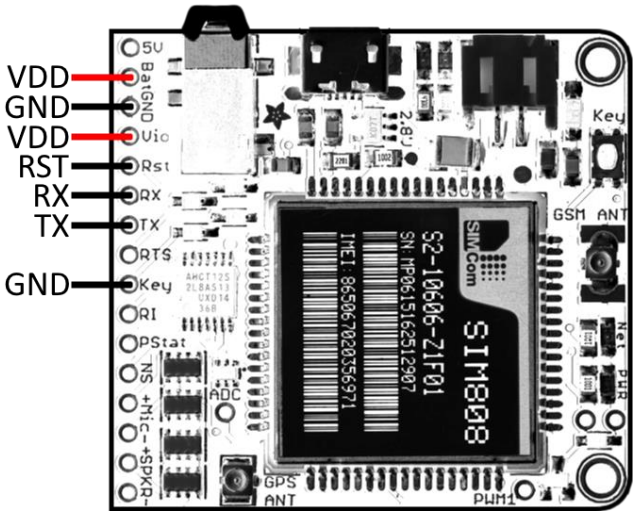
Microcontroller:

- Dit systeem is verantwoordelijk voor het verwerken van het sluiten van de kooi. Bij het afgaan van de sensoren zal het de GSM-module aansturen om een SMS te versturen dat de kooi is gesloten. Het luistert ook naar berichten die de GSM-module aankrijgt, waarmee de lijst van verwittigden kan worden aangepast of geverifieerd kan worden of de kooi nog functioneel is.



- Het subsysteem bestaat enkel uit de microcontroller zelf en een eventuele serial breakout die geplaatst kan worden op de FTDI-header zodat met een USB-C kabel programma's kunnen worden opgeladen.
- Als input heeft dit systeem een voedingsspanning en een ground, de sensor, en de RX van de UART. Als output is er de TX en de RST van de UART en bijkomend is er een FTDI-Header die gebruikt wordt voor het programmeren van de microcontroller. De RX, TX en RST vormen de UART die de microcontroller verbindt met de GSM-module. Het systeem verbruikt 1.6mA op 3.7V in actieve modus (6mW) en in slaapmodus slechts 1µA waardoor dit verbruik verwaarloosbaar is.

GSM-Module:



- Deze module maakt de connectie met het GSM-netwerk om sms'en te kunnen versturen en ontvangen.
- Dit systeem heeft 3 componenten: De GSM-module zelf, de GSM-antenne die wordt verbonden op de ingebedde connector op het bord, met het label GSM ANT, en de GPS-antenne, verbonden met de connector met het label GPS ANT.
- Als input heeft het systeem een voedingsspanning en een ground, en een RX en RST van de UART. Als output is er enkel de TX van de UART. De UART bestaande uit de RX, TX, en RST wordt gebruikt om te communiceren met de microcontroller.
- Dit systeem verbruikt veruit het meeste van alle subsystemen, namelijk ongeveer 1W tijdens gebruik. De module heeft echter ook een slaapmodus waarvan vaak gebruikt gaat moeten worden gemaakt om dit verbruik te kunnen compenseren. In ons design is de slaapmodus echter niet geïmplementeerd.

Interfaces tussen subsystemen:

Er zijn 5 interfaces in totaal tussen de 4 componenten, waarvan 3 interfaces voeding zijn van de componenten.

- De interfaces tussen voeding en sensoren, microcontroller en GSM-module zijn alle drie een positieve spanning en een ground. Er vindt geen communicatie plaats over deze interfaces.
- De interface tussen sensoren en microcontroller is een signaaldraad waarop een logische 1 staat als de sensoren niets detecteren, en naar een logische 0 valt als slechts 1 van de sensoren een magnetisch veld detecteert.
- De interface tussen microcontroller en GSM-module is een UART bestaande uit een receive (RX), transmit (TX), en een reset (RST). De communicatie is dus full-duplex, en de reset wordt gebruikt door de microcontroller om de GSM-module te kunnen resetten.

Gebruikte software:

```
#include "Adafruit_FONA.h"
#include <SoftwareSerial.h>
//used pinns on the sparkfun edge
#define FONA_RST 36
#define FONA_RX 38
#define FONA_TX 1
#define SENSOR 3
#define LED LED_BUILTIN

#define LISTLEN 10  //length of list for the phone numbers

//declaration of all functions
void unlockPin();
void deleteFromListAllSMS();
void checkNetworks();
char* readSMS();
void sendSMS(char sendto[],char message[]);
void cageClosed();
void add(char number[]);
void deleteFromList(char number[]);
void readGps(char sendto[]);
void flushSerial();
char readBlocking();
uint16_t readnumber();
void sendNumbers(char number[]);

//declaration of all variables
char gsmnrs[LISTLEN][21];
int indexList=0;
boolean currentFlag=true;
boolean previousFlag=true;
boolean receivedSMS=false;
uint8_t type;
uint8_t readline(uint16_t timeout, bool multiline);

//needed for the connection between microporcessor and gsm module
SoftwareSerial mySerial = SoftwareSerial(FONA_TX, FONA_RX);
SoftwareSerial *fonaSerial = &mySerial;

// this is a large buffer for replies
char replybuffer[255];

Adafruit_FONA fona = Adafruit_FONA(FONA_RST);
```

```
void setup() {
  //needed for communication with commandline and with Fona
  while (!Serial);
  Serial.begin(115200);
  mySerial.begin(4800);
  if (! fona.begin(*fonaSerial)) {
    Serial.println("Couldn't find FONA");
  }
  type = fona.type();

  unlockPin();
  delay(5000);  //make sure the pin has enough time to unlock

  pinMode(LED,OUTPUT);
  pinMode(SENSOR,INPUT);
  deleteFromListAllSMS();
  delay(10000);//make sure the sim has enough time to delete all the sms on it
}

void loop() {
  //positive edge detector for the sensor
  if(!digitalRead(SENSOR)){
    currentFlag=true;
  } else {
    currentFlag=false;
  }
  if(currentFlag){
    digitalWrite(LED,HIGH);
  } else{
    digitalWrite(LED,LOW);
  }
  if(!currentFlag and previousFlag){
    Serial.print("kat gevangen\n");
    cageClosed();
  }
  previousFlag=currentFlag;

  if(!receivedSMS){
    readSMS();
  }

  flushSerial();
  while (fona.available()) {
    Serial.write(fona.read());
    //Serial.write("vast in deze loop");
  }
  delay(1000);//only read every 1 second to save battery
}
```

```
void unlockPin(){
  char PIN[6]="1234\0"; // Only unlocks if pin code is "1234"
  flushSerial();
  Serial.print(F("Unlocking SIM card: "));
  if (! fona.unlockSIM(PIN)) {
    Serial.println(F("Failed"));
  } else {
    Serial.println(F("OK!"));
  }
}

void deleteFromListAllSMS(){  //deletes all sms so that sms number (needed for
readSMS()) is always 1
  int8_t smsnum=fona.getNumSMS();
  if(smsnum<0){
    return;
  }
  for(int i=1;i<=smsnum;i++){
    fona.deleteSMS(i);
  }
}

void checkNetworks() {    //checks if there are any 2G networks available and
prints them in the console
  DEBUG_PRINT("\t---> ");
  DEBUG_PRINTLN("AT+COPS=?");

  fonaSerial->println(F("AT+COPS=?"));

  DEBUG_PRINT("\t<--- ");
  DEBUG_PRINTLN(replybuffer);
}

char* readSMS(){
  uint16_t smslen;
  char sms[255];
  char sendto[21]="";
  if (! fona.readSMS(1, replybuffer, 250, &smslen)) { // sms number is always 1,
because after processing an sms, we delete it again
    Serial.println("Failed to read sms");
    return("\0");
  }
  receivedSMS=true;
  for(int i=0;i<strlen(replybuffer);i++){
    sms[i]=replybuffer[i];
  }
  Serial.println(sms);
  fona.getSMSSender(1,replybuffer,250);
```

```
 for(int i=0;i<strlen(replybuffer);i++){
    sendto[i]=replybuffer[i];
  }
  switch(sms[0]){ //add number to list
    case 'v':{}
    case 'V':{
      int i=0;
      Serial.println("nummers:");
      for(int i=0;i<LISTLEN;i++){
        Serial.println(gsmnrs[i]);
        //sendSMS(gsmnrs[i],message);
      }
      add(replybuffer);
      break;
    }
    case 'd':{}  //delete number from list
    case 'D':{
      deleteFromList(replybuffer);
      char message[141]="Uw nummer werd uit de lijst verwijderd\0";
      sendSMS(sendto,message);
      break;
    }
    case 'g':{} //gps
    case 'G':{
      readGps(sendto);
      break;
    }
    case 'n':{} //get list of numbers
    case 'N':{
      sendNumbers(sendto);
      break;
    }
    default:{
      Serial.println("commando niet herkend");
      char message[141]="Stuur 'v', 'd', 'g' of 'n' als u 1 van deze letters
stuurde is het probleem door een storing, probeer opnieuw\0";
      sendSMS(sendto,message);
      break;}
  }
  delay(10000); //make sure there is enough time to read the sms properly
  fona.deleteSMS(1);
  delay(10000); //make sure there is enough time to delete the sms
  receivedSMS=false;
  return(sms);
}

void sendSMS(char sendto[],char message[]) {
  flushSerial();
```

```
  if (!fona.sendSMS(sendto, message)) {
    Serial.println(F("Failed"));
  } else {
     Serial.println(F("Sent!"));
  }
}

void cageClosed(){    //sends sms to all numbers in the list
  //char sendto[21]="0032473483449\0";
  char message[141]="De kooi is gesloten, er is misschien een kat gevangen\0";
  for (int i = 0; i <= indexList; i++) {
    Serial.println(gsmnrs[i]);
    Serial.println("\n");
    sendSMS(gsmnrs[i],message);
    delay(5000);
  }
}

void add(char nummer[]){    //adds a new number to the list
  for(int i=0;i<LISTLEN;i++){
    if(strcmp(nummer,gsmnrs[i]) == 0){
      sendSMS(nummer,"Uw nummer staat reeds in de lijst");
      return;
    }
  }
  char message[141]="Uw nummer werd toegevoegd aan de lijst\0";
  sendSMS(nummer,message);
  for(int i=0;i<13;i++){
    gsmnrs[indexList][i]=nummer[i];
  }
  gsmnrs[indexList][13]='\0';
  indexList++;
}

void deleteFromList(char nummer[]){   //deletes a given number from the list
  for(int i=0;i<indexList;i++){
    if(strcmp(nummer,gsmnrs[i])==0){
      for(int j=i;j<10;j++){
        strcpy(gsmnrs[i],gsmnrs[i+1]);
      }
      strcpy(gsmnrs[9],"\0");
      break;
    }
  }
  indexList--;
}
```

```
void readGps(char sendto[]){   //starts gps, reads it, sends location to person
who asked for it, turns it off again
  if (!fona.enableGPS(true)){
    Serial.println(F("Failed to turn on"));
    sendSMS(sendto,"De gps kon niet opgestart worden");
    return;
  }
  sendSMS(sendto,"De locatie wordt opgevraagd, dit kan enkele minuten duren.");
  int8_t stat;
  stat = fona.GPSstatus(); // check GPS fix
  while(stat<=2){
    delay(1000);
    stat = fona.GPSstatus();
  }
  char gpsdata[120];
  fona.getGPS(0, gpsdata, 120);
  if (type == FONA808_V1)
    Serial.println(F("Reply in format:
mode,longitude,latitude,altitude,utctime(yyyymmddHHMMSS),ttff,satellites,speed,cou
rse"));
  else
    Serial.println(F("Reply in format:
mode,fixstatus,utctime(yyyymmddHHMMSS),latitude,longitude,altitude,speed,course,fi
xmode,reserved1,HDOP,PDOP,VDOP,reserved2,view_satellites,used_satellites,reserved3
,C/N0max,HPA,VPA"));
   Serial.println(gpsdata);
   sendSMS(sendto,gpsdata);
   if (!fona.enableGPS(false))
    Serial.println(F("Failed to turn off"));
}

void sendNumbers(char number[]){
  String str="Nummers: ";
  String str2=String(gsmnrs[0]);
  str2.trim();
  str+=str2/*.trim()*/;
  int i=1;
  while(gsmnrs[i][0]!='\0'){
    str+=", ";
    str2=String(gsmnrs[i]);
    str2.trim();
    str+=str2;
    i++;
  }
  char numbers[21*LISTLEN];
  str.toCharArray(numbers,21*LISTLEN);
  sendSMS(number,numbers);
}
```

```
void flushSerial() {
  while (Serial.available())
    Serial.read();
}

char readBlocking() {
  while (!Serial.available());
  return Serial.read();
}

uint16_t readnumber() {
  uint16_t x = 0;
  char c;
  while (! isdigit(c = readBlocking())) {
    //Serial.print(c);
  }
  Serial.print(c);
  x = c - '0';
  while (isdigit(c = readBlocking())) {
    Serial.print(c);
    x *= 10;
    x += c - '0';
  }
  return x;
}
```

Handleiding

- Schroef de doos open en schroef het bordje los.
- Haal de DC adapter uit de voltage regulator en haal het bordje uit de doos.
- Doe de simkaarthouder van de GSM-module los en steek de simkaart hierin
- Bevestig het bordje opnieuw aan de doos en verbind de DC adapter opnieuw.
- Na de voltage regulator staan 2 jumper pinnen. Plaats hierover een jumper om het systeem aan te zetten.
- Sluit de doos goed zodat deze terug waterdicht is.
- Wacht een halve minuut tot het systeem volledig is opgestart.
- De kooi kan nu worden gebruikt. Stuur ‘V’ naar het nummer van de simkaart in de kooi om uw nummer toe te voegen aan de lijst van verwittigden.
- Alternatief kan een ‘D’ verstuurd worden om uw nummer terug te verwijderen, en een ‘G’ om de GPS-locatie op te vragen.

4. Referenties

1. https://www.antratek.nl/edge-development-board-apollo3-blue
2. https://www.antratek.nl/fona-808-mini-cellular-gsm-gps-breakout
3. https://www.antratek.nl/polymer-lithium-ion-battery-2000mah
4. https://www.antratek.nl/bq24074
5. https://be.farnell.com/alps-alpine/hgdept021b/magnetic-sensor-0-0013-0-0027t/dp/3261855?st=magnetic%20sensor
6. https://datasheet.octopart.com/74HC08N-Philips-datasheet-8217261.pdf
7. https://www.antratek.nl/serial-basic-breakout-ch340g
8. https://www.antratek.be/solar-panel-6v-5w