

Sakari Järvelä

CAR ALARM SYSTEM ENGINEERED IN ARDUINO ENVIRONMENT

CAR ALARM SYSTEM ENGINEERED IN ARDUINO ENVIRONMENT

Sakari Järvelä
Bachelors Thesis
Autumn 2014
Information Technology
Oulu University of Applied Science

TIIVISTELMÄ

Oulun ammattikorkeakoulu

Tietotekniikka, Ohjelmistokehitys

Tekijä: Sakari Järvelä

Opinnäytetyön nimi: Auton varashälytinjärjestelmä toteutettuna Arduino-ympäristössä

Työn ohjaaja: Timo Vainio

Työn valmistumislukukausi ja -vuosi: Syksy 2014

Sivumäärä: 29+3 liitettä

Tavoitteena tässä opinnäytetyössä oli suunnitella ja toteuttaa autoon varashälytinjärjestelmän prototyyppi hyödyntäen Arduinon tarjoamaa ohjelmistokehitys- ja piirilevy-ympäristöä.

Kehitettävän järjestelmän on tarkoitus tarjota käyttäjälleen monipuolinen varkaudenestolaitteisto ja tarvittaessa varastetun ajoneuvon paikantamisen mahdollistavat toiminnot.

Opinnäytetyöllä ei tavoitella kaupallista hyötyä, vaan tutustutaan Arduinon, GPS:n NMEA 0183:n ja SIM900:n tarjoamiin kirjastoihin ja informaatioon puolueettomasta näkökulmasta. Suurin osa projektin lähteistä on yritysten julkaisemia, mutta ilman yritysten tarjoamia lähteitä ja kirjastoja ohjelmistoa olisi vaikea toteuttaa.

Opinnäytetyössä käydään läpi määritellyt toiminnalliset vaatimukset, ohjelmiston käyttöön tarkoitettu laitteisto ja suunniteltu ohjelmisto, joka pyritään toteuttamaan. Lopuksi perehdytään aikaansaannoksiin ja pohditaan tuloksia.

Resursseja kartoittaessa laadittiin suunnitelma siitä, miten järjestelmän tulisi toimia. Samoin ohjelmistoa ehdittiin jonkin verran toteuttaa. Ohjelmiston lisäksi GSM/GPRS Quadband -moduuli asennettiin Arduino Mega 2560:een ja ajonestorele juotettiin piirilevyyn.

Asiasanat: Varashälytin, Arduino, GPS, GSM, PIR-sensori, SIM900

ABSTRACT

Oulu University of Applied Sciences
Information Technology, Software Developing

Author: Sakari Järvelä
Title of thesis: Car Alarm System Engineered in Arduino Environment
Supervisor: Timo Vainio
Term and year when the thesis was submitted: Autumn 2014 Pages: 29+3 appendices

This thesis is targeting to develop and engineer a burglar alarm system prototype using the software and hardware environment Arduino, which is originally manufactured by Smart Projects. The system under development provides a versatile theft inhibition hardware configuration for the holder of the system. It also provides vehicle localization enabling features in case of theft.

In the final project, the aim is not to reach commercial benefit. The aim is to get familiar with software libraries and information offered by Arduino environment, GPS utilizing standard NMEA0183 and SIM900 with AT commands. This project examines burglar alarm system engineering from a commercially neutral point of view. Most of the sources have been published by companies, but without these essential resources it would be extremely complicated to engineer the software. In process of this final project, the functional requirements and hardware are defined. Then the embedded software is implemented. Finally, the achievements are explored and the final outcome is discussed.

With the available resources for the project, it was feasible to create a project plan. The project plan describes how the burglar alarm system should work. A partial software was engineered and it was ready to use few AT commands. Also, the hardware design was put together. The GSM/GPRS Quadband module was assembled into the Arduino Mega 2560, and an anti-theft immobilizer relay was soldered into a circuit board. The PIR-sensor was acquired and connected to the Arduino Mega 2560.

Keywords: Burglar alarm, GPS, GSM, PIR-sensor, SIM900

PREFACE

The information to create software came as a gift after the studies. The motivation to create is brought by friendships, family and hope for better future. This thesis was made feasible to be created by many instances, and it is impossible to show enough gratitude.

I would like to thank Mika Männistö for giving the idea in my thesis. I also want to thank Petri Nieminen for proofreading the thesis. In addition, I wish to thank my family for all the support which has been priceless. And finally, I would like to thank OUAS for offering me a share of information to create software.

18.11.2014 Sakari Järvelä

CONTENTS

| | |
|--|----|
| TIIVISTELMÄ | 3 |
| ABSTRACT | 4 |
| PREFACE | 5 |
| CONTENTS | 6 |
| SYMBOLS AND ABBREVIATIONS | 8 |
| 1 INTRODUCTION | 9 |
| 2 FUNCTIONAL REQUIREMENTS | 10 |
| 2.1 Main function of alarm system | 10 |
| 2.2 Alarm system setup | 10 |
| 2.3 Immobilization | 10 |
| 2.4 PIR-sensor state reporting via wiring | 10 |
| 2.5 Capability of fetching location using text message | 11 |
| 2.6 Automated low battery level detection | 11 |
| 3 HARDWARE | 12 |
| 3.1 Arduino Mega 2560 | 13 |
| 3.2 GPRS/GSM shield (GGS) | 13 |
| 3.3 GPS module | 13 |
| 3.4 Passive Infrared (PIR) sensor | 14 |
| 3.5 Self-made anti-theft relay | 14 |
| 3.6 Hardware installation location | 15 |
| 3.7 Alarm system battery | 15 |
| 3.8 Wiring | 16 |
| 3.9 Hardware testing | 18 |

| | |
|---|----|
| 4. SOFTWARE | 19 |
| 4.1 Structure | 19 |
| 4.2 PIR sensor decoding | 21 |
| 4.3 GPRS and GSM shield controlling | 21 |
| 4.4 SMS commands | 21 |
| 4.5 GPS controlling and NMEA reading | 22 |
| 4.6 Testing | 22 |
| 5. OUTCOME | 23 |
| 5.1 Hardware installations and electrical solutions | 23 |
| 5.2 Software engineering | 23 |
| 6. SPECULATION | 26 |
| 6.1 Achievements | 26 |
| 6.2 Learnings | 26 |
| 7. CONCLUSION | 28 |
| REFERENCES | 29 |
| APPENDICES | 31 |

SYMBOLS AND ABBREVIATIONS

| | |
|--------------------|--|
| AM2560 | Arduino Mega 2560 board |
| ANT | Antenna (Internal 4G/3G/GPRS/GSM) |
| AT-M | (Self-made) Anti-Theft Module |
| GGQM | GPRS/GSM Quadband Module (for Arduino/Raspberry PI) |
| GGs | GPRS/GSM Shield |
| GPS-M | GPS Module (Navilock GPS Engine Board EM-406A) |
| HPN | Holder's Phone Number (char type parameter in software) |
| INPUT MODE message | Message send from PIR-sensor to AM2560 |
| SIM900 | Communication operating chip |
| SPN | System Phone Number (telephone number of the burglar alarm system) |
| TXT-M | Text Message |

1 INTRODUCTION

This thesis was started as a result of a small talk at a school cafeteria. Mika Männistö, an engineer, had software and hardware needs for his current vehicle. The most interesting project during the discussion was an excellent burglar alarm system. It consists of versatile benefits, for example calling an alarm call to holder of the system (referred as *holder* later in the thesis) and allowing to locate the vehicle if it has been stolen. After the idea of the burglar alarm system appeared, there was weeks of studying and it was brought up that there is information enough to engineer such a system.

Mika Männistö is an engineer of Mechanical and Production Technology, the option of Vehicle and Transportation Technology and the target vehicle of the burglar alarm system is the same as used in his thesis (1).

While studying the hardware needs for the project, it was found out that Arduino Mega 2560 (AM2560) is in the budget range of the project. Arduino offers multiple hardware solutions and wide software libraries for use with hardware.

For communication purposes, the GPRS/GSM shield (GGS) is defined as follows: GGS consists of GPRS/GSM Quadband module for Arduino/Raspberry PI (GGQM), communication operating chip SIM900 (SIM900) and Antenna module (ANT).

The Passive Infrared (PIR) sensor is a simple sensor which is monitoring the surrounding area. If there are movements seen in the surrounding area, a sensor changes the alarm pin state to HIGH after calibrating the sensor.

For a location requirement, there was seen a clear need for a GPS chip. There is enough information to utilize a Navilock GPS Engine Board EM-406A with NMEA 0183 standard and it was reason to choose it for the project purpose.

Along the hardware plan the whole project was planned as introduced in appendix 3.

2 FUNCTIONAL REQUIREMENTS

In this chapter, it is described how the software was defined and planned to operate. Every function is explained in detail containing a description on how it runs in the software. The software variables and constants are introduced for the intended purpose.

2.1 Main function of alarm system

The main function of the alarm system is to inform the holder of the system about any possibilities of a theft. In case of a burglary, the alarm informing is executed via the GGS and the system dials to the HPN. In the first prototype, the HPN is defined as a hard-coded parameter and it can be changed only by modifying the source code. An argumentation for implementing this way is that it would be a major addition/modification for the current source code in case of implementing as actual parameter. It is not complicated to change the HPN in the source code and flash the system again if the HPN must be changed. There are some additional functions created to help the main functionality work properly and they are described in the following sections.

2.2 Alarm system setup

To switch an alarm system on and off, the holder calls to SPN. If the system is turned on by the call, AM2560 blinks three times a red led. If the system is turned off, AM2560 blinks three times a green led.

2.3 Immobilization

AT-M is a small relay switch that is controlled by AM2560. When the burglar alarm system is turned on, the state of AT-M is set to deactivate. As a consequence, AT-M disables all possibilities to start the engine of the car.

2.4 PIR-sensor state reporting via wiring

The PIR-sensor is a hardware that is capable of informing the system if motion is detected around the sensor.

When a PIR-sensor is switched on, it sends an INPUT MODE message with the content: LOW via wiring to AM2560.

In a case of theft, the PIR-sensor sends the INPUT MODE message with content: HIGH via wiring to the AM2560 (2; 5). The message content HIGH has the following two meanings:

1. Sensor turns off
2. Sensor state changes

In this case, the HIGH message is sent when the state changes after the sensor are turned on during the setup. The HIGH message means that motion is detected inside the vehicle or power is down for some reason. By default, it is expected that there are no motions in the PIR-sensor range and that actually applies in the whole alarm system.

2.5 Capability of fetching location using text message

GPS-M is a lower priority benefit for the program and it is organized to work only if the user wants to fetch the coordinates of the car in his mobile phone. The location is fetched in case that the user sends a TXT-M from the HPN to the SPN. The message contains only a short command: "Location" and the system reads the message and interprets that the user has requested coordinates. AM2560 uses GPS-M and collects the coordinate information from NMEA data. Finally, the AM2560 commands the GGS to transmit the coordinates back to the HPN.

2.6 Automated low battery level detection

AM2560 follows the state of the car battery. In case that the battery charge level is reduced to the chosen level, the AM2560 requests the GGS to send a message 'Low battery' to the HPN. The battery level threshold is hard-coded in the software. The level can be only changed by modifying the source code.

3 HARDWARE

In this chapter the focus is to introduce the chosen hardware parts for the burglar alarm system.

This project consists of the following hardware parts:

- AM2560: Arduino Mega 2560 microcontroller board
- GPS-M: Global Positioning System Engine Board EM-406A
- GGS: GPRS/GSM Shield
- AT-M: (Self-made) Anti-Theft Module
- ANT: Internal 4G/3G/GPRS/GSM antenna

There is a more detailed description introduced in the following sections. Figure 1 is made to facilitate the understanding of the connections between each hardware part and actor.

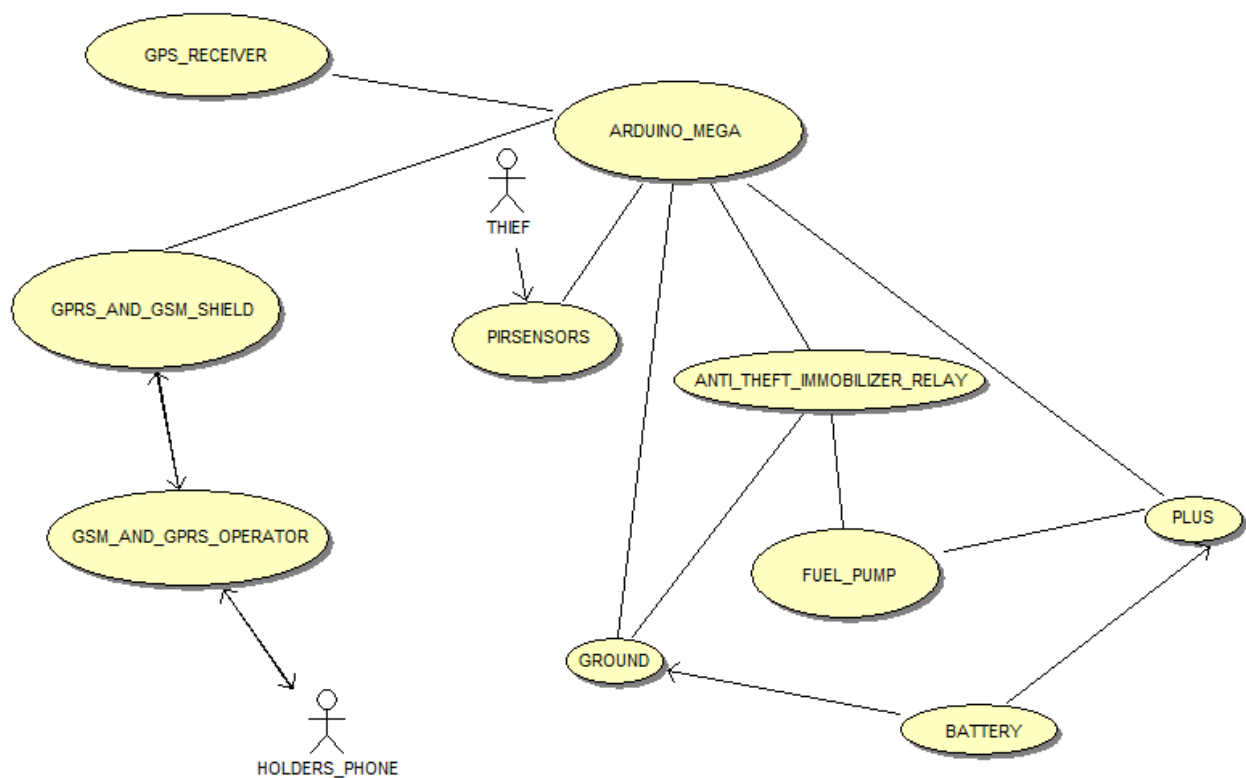


FIGURE 1. Hardware and actors

Because of the varying temperature the circuit boards should be mold into rubber and protected from humidity.

3.1 Arduino Mega 2560

“Arduino is an open-source electronics platform based on easy-to-use hardware and software.”(6)
The microcontroller board is based on the ATmega2560 processor and the board offers 54 input and output pins to manage the system that one wants to create (7).

Pins rx3 and tx3 are used for serial communication from the AM2560 to the GPS-M. The GPS-M uses one +5V pin and one GND pin from the AM2560. The PIR sensor requires +5V pin, GND pin and one digital pin. Those pins must be the specifically programmed pins because there is no more +5V pins and GND pins left. The PIR sensor is using pin 46 as a +5V pin, pin 47 as a GND pin and pin 48 as a digital pin. The GSM shield uses analog input pins from A0 to A5 and Arduino digital I/O pins from 3 to 7. The GSM shield physically covers power pins and makes operating a bit harder.

3.2 GPRS/GSM shield (GGS)

The GGS is used to create a connection between the holder and the burglar alarm system. The GGS is put together with the AM2560 using connector pins.

The GGS uses the SIM900 module for connection creation to network operator. The SIM900 offers a wide AT-command handling and is used with the GGQM ordered from the cooking-hacks. The cooking-hacks is a web page which delivers hardware and provides software examples to be used with the hardware. The SIM900 is connected to the ANT to be able to transmit and receive the GPRS/GSM signal. Operation frequencies of the GPRS/GSM are between 900MHz and 2.1GHz. Both the antenna and the SIM900 module can operate between -40- +85 Celsius degrees.

3.3 GPS module

One functional requirement is to fetch the location of the vehicle and that is the reason why the GPS module is required. Due to a low budget of the work, Navilock GPS Engine Board EM-406A

was chosen as the GPS device and it is physically wired to the AM2560. The GPS-M offers a very high tracking sensitivity with 20 channel all-in-view tracking and supports the NMEA 0183 standard, additionally it was the cheapest AM2560 compatible GPS module available (8). To read the standard and gather all an important data from the GPS-M, the AM2560 needs a function that picks up the specified location data. The NMEA 0183 standard offers more options, but the most important information for the holder is the location of the vehicle in case of a theft. As can be seen in the datasheet (8) of the GPS-M, the cold start of the GPS takes 42sec on the average. If a vehicle thief detects the GPS-M when location fetching is required by the holder, 42 seconds is a sufficient time for a thief to eliminate the GPS system. This is the reason the GPS module should be located in a place that is not directly visible, but the module must have a view to the sky to have a line of sight for satellites. A small layer of plastic is not blocking the signal and that offers a possibility to hide the GPS under a plastic cover of a car.

3.4 Passive Infrared (PIR) sensor

This project uses one PIR sensor that is manufactured by the Sparkfun. The Sparkfun is a webpage delivering various electronics. The PIR sensor is able to detect a motion and trigger actions while detecting. In this project, PIR sensor is monitoring a driver seat surrounding and sending HIGH message to the AM2560 when movement is detected.

The chosen PIR sensor works in the temperature range of -10 – 40 C degrees(9). Winter conditions can be sometimes colder than -10 in Finland and this limitation was acceptable for the project prototype.

3.5 Self-made anti-theft relay

The anti-theft relay part is made on its own circuit board and the following components are soldered on the board:

- RAS-0515 relay
- CDIL 2N2219A transistor
- Diode
- Resistor

The AM2560 is supposed to control the RAS-0515 relay by software. The relay controls a grounding of the fuel pump. In case that the anti-theft is activated in the AM2560, the AM2560 sets a controlling voltage down and the relay cuts the grounding of the fuel pump.

3.6 Hardware installation location

When installing the hardware, the following facts should be taken care of:

- Do not leave direct view to any part of the alarm system
- The GGS, the GPS-M and the AM2560 should be hidden very well
- Make sure that the PIR-sensor is not easy to detect, although it should be installed in a certain area to sense motions inside the whole vehicle
- Place the GGS and the GPS antennas on the surface of the dashboard

3.7 Alarm system battery

It has been thought that the alarm system can either use a vehicle battery or there could be a dedicated battery for the burglar alarm system purposes.

In the first prototype version, the vehicle battery is used as a source of electricity. This works perfectly, since recommended voltage of the AM2560 is between 7 and 12 volts and the AM2560 offers electricity for every other device in use.

The best electricity solution would be the dedicated battery for the alarm system. A drawback of using the vehicle battery for the alarm system is that if the voltage goes low enough, the burglar alarm system can't have power to work and it resets the whole system. Also starting of the vehicle would be impossible due to the lack of the battery voltage. A dedicated battery would maintain the voltage for the burglar alarm system. As a result, the burglar alarm system would be more reliable.

3.8 Wiring

There is a need for an additional wiring to connect all the hardware together. A formerly introduced components need to be connected with the specified wiring figure 2.

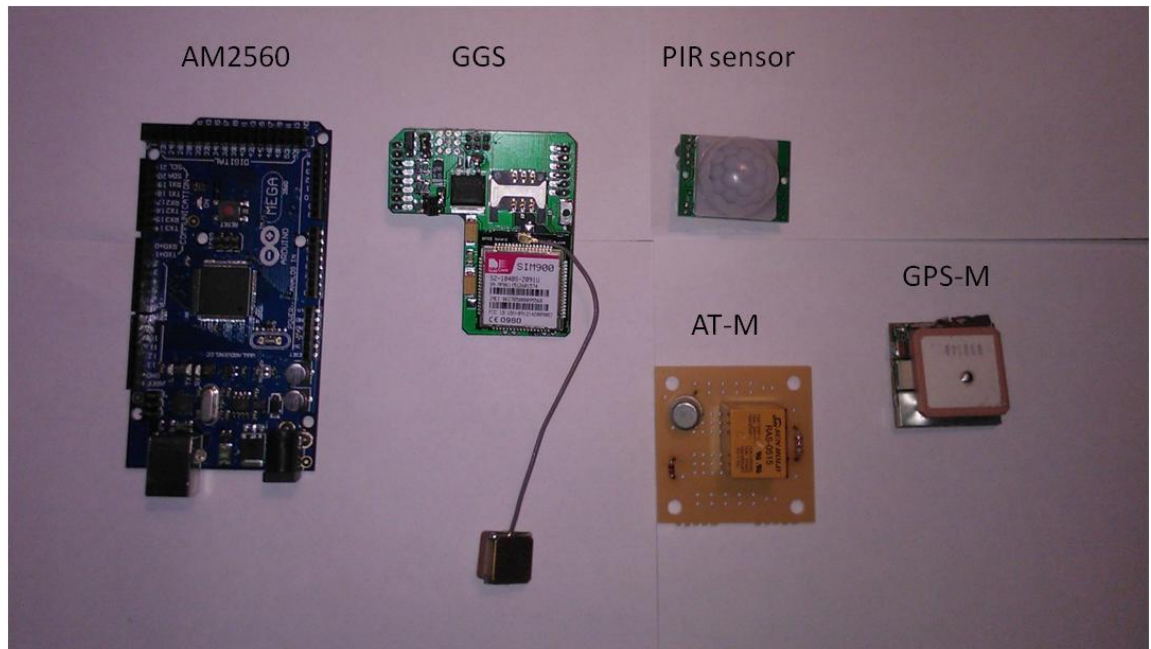


FIGURE 2. Hardware components

The AM2560 needs a consumption voltage wire from the vehicle battery or a dedicated battery and all the modules that need a specified wiring are connected to the AM2560. The GGS and the PIR-sensor are connected in lab conditions into the AM2560 figure 3.



FIGURE 3. The GGS and the PIR-sensor connected to the AM2560

The GPS-M requires five wires to the AM2560

- rx wire
- tx wire
- +5 volts wire and
- two ground wires from the AM2560.

The AT-M requires two control wires and one power wire that goes through the switch of relay.

The PIR sensor requires +5 volts wire, a ground wire and a digital pin wire from the AM2560.

3.9 Hardware testing

The hardware testing plan was created in the beginning of the project. A testing plan consists of the following steps:

1. The AM2560 as a standalone HW
 - Switch power on
 - Switch power off
2. The AM2560 and The GGS and The PIR-sensor are connected all together in the lab conditions
 - Power on/off
 - Software flashing into the AM2560
 - Power off/on
 - Move hand near the PIR sensor and see if motion is detected and an alarm message sent
 - Simulate car starting and see if the alarm message is being sent
3. The GPS-M is connected to the AM2560 in the lab conditions
 - Corresponding test sequence as in 2. above
 - Send SMS "Location" and verify if coordinates are received by the holder

When tests are passed in the lab conditions, the hardware is assembled into the car. The main hardware testing is to assemble the whole hardware in the car and see, if it can tolerate the vibrations of the car.

4. SOFTWARE

In this chapter the structure of the software is described and also how the system is planned to operate is explained. The software is embedded software for Arduino, as described in 3.1.

The Arduino IDE v0.22 is used as a programming environment, because GGS tutorial codes have been developed to work on that particular version of Arduino IDE tools (11). The latest Arduino version 1.0.3 is also tested during this thesis. If the latest version operates as expected, it will be used in the final version of the burglar alarm system.

4.1 Structure

The Arduino IDE is well known for that its source code has only two main functions: a `setup()` and a `loop()`. The `setup()` –function runs only once in the beginning of the execution. As the name `setup` states, the function is made to setup pre-attributes for the `loop()` -function. The `loop()` –function takes care of all the operations of burglar alarm system.

In this project the `setup()` –function sets a serial communication to communicate with the GGS, since the GGS must be able to act immediately in a case of burglary. The other serial gateway must be used for the GPS-M and it is put in use after a SMS command ‘Location’ from the HPN. The Arduino IDE does not allow a simultaneous operating with more than one serial port, hence execution of many serial gateways is done with a Arduino `SoftwareSerial` class. The communication types are prioritized in there, as follows:

1. High priority: a GGS connection to the AM2560 – using primary serial gateway
2. Medium priority: a GPS-M connection to the AM2560 – using secondary serial gateway

The `setup()` -function sets the AM2560 to use the serial gateway for the GGS communication. The serial communication can be changed by using different exception architecture. In the burglar alarm system it means reception of the ‘Location’ SMS. It triggers an action, that the primary serial gateway is set unused and the secondary serial gateway starts communication with the GPS-M. The burglar alarm system has also set up that it is not in the alarming mode and

corresponding actions are defined in the loop() -function. The Arduino hardware indicates that a standby mode is on by blinking the green led three times.

The basic operations of the system are engineered in the loop() –function. As the GGS begins to receive phone calls, the system is going to be ready to switch in the alarming mode. In the loop() –function it is verified if the HPN is the number wherefrom a receiving call comes and the alarming mode is enabled. After activating the alarming, the system gives a 30 seconds waiting period for the PIR sensor to gather a surrounding IR image and then starts to wait for a signal of observed movements from the sensor. The sensor reports to the AM2560 about movements by changing data pins “pull up resistor” to “pull down resistor”. After the “pull down” message, there is an if -clause for the alarming. The if -clause contains a section in which the AM2560 sends AT commands to the GGS to call in the HPN. When the GGS has completed the call, the system goes into the “waiting state” and waits for further commands. As sending an SMS to the SPN, the holder can set the AT-M to affect and make it more difficult for thieves to move the car anywhere. In the final product, the commands are all received as an SMS, but in the prototype it is obvious that setting the burglar alarm system active/inactive state is requested only by calling to the SPN.

Based on the structure of the software, following requirements were put together and listed as scrum items (appendix 1):

- The system sets the Anti-theft on when the alarm is active
- As a user I can receive an alarm call to my mobile phone
- As a user I can receive SMS messages from the system
- The system sends a message if the battery needs to be charged
- The system calls into the owners phone if a thief enters in car
- As a user I can ask a location of my car in case of the thief is moving the car
- The Arduino Mega can control the Anti-theft relay
- As a user I can set the system active and inactive by calling to system
- In active mode Arduino uses PIR sensor to notice burglary
- System reads NMEA from GPS antenna
- If PIR sensor detects burglary the system alarms the owner

4.2 PIR sensor decoding

Home page of the Arduino offers a good example code that can be used for alarming purposes (2). The example code should be engineered to control the GGS if motion is detected. As in a praxis the sensor calibrating should be done if the system is activated by dialing to the SPN from the HPN.

4.3 GPRS and GSM shield controlling

The GSM (Global System for Mobile Communications) is a worldwide communication standard. The GSM is the first digital telecommunication standard and still maintains the widest penetration globally (3). The GSM was chosen to be used in this thesis because the GGS provides an Arduino compatible telecommunication chip. On the other hand, the source code for utilizing the chip is available from the manufacturer. In this project the GSM network is used to connect the burglar alarm system and the holder's mobile phone. The GSM is circuit switched system and the connection establishment using a dedicated communication channel takes few seconds. This is seen decent period for the burglar alarm system.

The GPRS (General Packet Radio Service) is built on top of the GSM and it is supported with the same hardware, but it was not utilized in this project.

The AM2560 should be programmed to switch on the GGS immediately in the beginning of the process. The serial communication should be started in the `setup()` –function. The `loop ()` –function controls the GGS if there is a phone call received from the HPN and it switches the alarm system on and off by changing the activation boolean.

4.4 SMS commands

Three SMS commands are used in this prototype. As the system can read messages and verify the HPN, it can also effect system wide on how to react according to the received messages.

The “Location” command starts the GPS communication with the system and keeps it running until the AM2560 has received location coordinates. The location is set in a char integer and written into the returning message for the HPN.

The “Stop” command sets the AT-M active and disables an engine to start. The “Start” command sets the AT-M inactive and sets the engine to be able to start.

4.5 GPS controlling and NMEA reading

After the “Location” SMS command, the Arduino is assumed to execute the cold start of the GPS-M and to start to read the NMEA –data. The cold start and the NMEA –data reading engineering is found in Arduinos pages (12). The NMEA –data reading process should have its own if –option and it should be executed only when the GPS-M is activated.

The NMEA 0183 Interface Standard defines electrical signal requirements, data transmission protocol and time, and specific sentence formats for a 4800-baud serial data bus. Each bus may have only one talker but many listeners. This standard is intended to support one-way serial data transmission from a single talker to one or more listeners. This data is in printable ASCII form and may include information such as position, speed, depth, frequency allocation, etc. (4)

The GPS-M brings the standard into a readable serial communication bus which the Arduino Mega 2560 is able to read. There is a need for a specified wiring which can be solved by reading a datasheet of the GPS module and the pin labels on the circuit board of the AM2560.

4.6 Testing

Test automation is not covered in this project. All the tests are executed manually as follows:

- Test how the AT –commands work
- Test how the PIR –sensor source code works and how it can be integrated in the alarm system
- Test how to implement the serial communication between the GGS and the AM2560
- Test how to read SMS messages via AT –commands
- Check on how the SMS do work
- Verify if the NMEA reading works properly
- Test PIR –sensor motion detection

5. OUTCOME

The final result of the thesis is described in this chapter. Hardware installations and electrical solutions are explained and the software engineering is presented.

5.1 Hardware installations and electrical solutions

The GPS module was damaged during the transportation and it was not able to be used. The GGS was easy to assemble into the AM2560. The PIR sensor was also connected to the AM2560. The self-made anti-theft relay was smoldered in the circuit board with a transistor, a resistor and a diode.

5.2 Software engineering

The SIM900 does not offer an AT-command to read a phone number of an incoming call, so the incoming call recognizing appeared as a major problem (10). In the setup –function, the serial communication is started by a Serial.begin –function, which takes in the baudrate of the serial communication as an attribute. The GGS factory baudrate is set to 115200 (11). After the baudrate setting, delay of 4000ms is set to allow the function to complete all the required operations. The PIR-signal pin is set up as an input-pin. As the serial communication is started for the GGS on the Arduino, the Arduino should switch the GGS on. It is engineered with the Arduino pinMode –function and set a shieldPowerUpPin as an output pin. After declaration the output is written to “high” with a digitalWrite –function. The high content gives a voltage to power up section of the GGS and switches it on. There is 1500ms delay for enabling voltage in the shieldPowerUpPin and after that period of time, the shieldPowerUpPin is set to “low” state because it is not required to provide the voltage anymore. Delay of 20000 ms is given to start up the whole GGS. A Serial.println(“AT+CMFG=1”) sets the SMS-reading mode on text mode. Setting it up needs also a little delay of 1500ms. The setup() –function source code is shown in the figure 4.

```

void setup() {
  Serial.begin(115200); // GSMConnection.begin(115200);
  delay(4000);
  pinMode(pirOnePin, INPUT);
  pinMode(shieldPowerUpPin, OUTPUT);
  digitalWrite(shieldPowerUpPin, HIGH);
  delay(1500);
  digitalWrite(shieldPowerUpPin, LOW);
  delay(20000);
  Serial.println("AT+CMGF=1");
  delay(1500);
}

```

FIGURE 4. Setup() –function.

Defined booleans, integers and characters are shown in the figure 5. There are SoftwareSerial definitions introduced for two functions. In this thesis, it was only tested how do they operate.

```

#include <SoftwareSerial.h>
boolean motionDetected = false;
boolean setAlarmsystemActive = false;
boolean locationAsked = false;
boolean activated = false;
boolean inactivateSensor = false;
int shieldPowerUpPin = 2;
int GPSRX=50;
int GPSTX=51;
int GSMRX=4;
int GSMTX=5;
int pirOnePin = 50;
char HPN[] = "*****";
char data[] = {255};
SoftwareSerial GPSConnection(GPSRX,GPSTX);
SoftwareSerial GSMConnection(GSMRX,GSMTX);

```

FIGURE 5. Class and variable definitions

The loop –function includes the PIR-sensor handling and the serial communication handling with the GGS and the GPS-M (appendix 2). While working with the SIM900 it was found out that it is not possible to send an AT-command which returns a phone number of the incoming call. It was necessary to find out another way to make the system work smoothly. That is the reason why the

loop() -function includes an SMS reading function, that decodes if the holder wants to activate or deactivate the system. The SMS must be read all the time, so the alarm system controlling can be done in a real-time. Also, the Serial1 -function is created for the GPS-M reading, but the actual functionality is missing. When reading serial port from the GGS and fetching the SMS-message, the console returns the whole message with 'OK' in the end as a sign that the message has ended. The 'OK' is deleted from the string by adding '\0' in the string. After processing the whole string it is observed that the message includes the right content. If the content is 'Activate', the boolean setAlarmSystemActive is set to true. The PIR-sensor content of the program reads the value of the setAlarmSystemActive and decides that it needs to be activated. After the PIR-sensor activation, the system deletes all the messages by an AT-command "AT+CMGDA=DEL ALL". If the content is 'Deactivate', the boolean setAlarmSystemActive is set to false and the alarm system is meant to be shut down. If the boolean motionDetected is true, the AT-command is set to dial in the HPN. An AT-command "ATD+HPN" returns message 'NO CARRIER' which is described in the SIM900 documentation as a message for that the connection cannot be established. The SIM900 documentation does not explain the cause of the problem and in these circumstances problem cannot be solved.

6. SPECULATION

The usage of the project resources changed more planning highlighted because a system this large cannot be created without a good plan. As a literal production the thesis is a good background to create a good burglar alarm system with versatile benefits.

6.1 Achievements

During the project it was found out that the engineering of the burglar alarm system is possible with the Arduino. The software libraries and the required hardware information are available to accomplish comprehensive system.

6.2 Learnings

Planning a large project without the knowledge of the system procedures is very challenging, which caused a lot of new learnings. Software based on the simultaneous serial bus communication appeared as a new solution. The utilization of the SoftwareSerial class could be used wider in a further development of the burglar alarm system. An incoming call phone number examining was a challenging problem. As a result, the burglar alarm system does not verify the number of the caller. It was found out that the SIM900 does not offer all the necessary features to engineer the burglar alarm system as planned. Unfortunately, it was not tested if the SIM900 automatically returns the incoming call phone number in the serial bus due to connection problems. The AT-command "ATD+HPN" returned 'NO CARRIER' message. An actual reason was not found, but it is suspected that there is a communication failure between the GGS and an operator.

One of the weakest sections in the system is the PIR sensor. The reason is that if the connection from sensor is lost, the AM2560 does not reach "state changed" message from the sensor. If the message cannot be reached, the holder never receives information about the burglary. It has been found out that the weakness of the PIR sensor could be taken care of by reading if the voltage of any PIR pin changes (13). If using only one PIR-sensor, it is not possible to sense a motion inside the whole car. An improved version of the alarm system should contain more PIR-sensors to cover whole interior of a car.

The alarm system should also have a dedicated battery, which should be hidden well. The dedicated battery would solve the problem of starting the engine and save the battery of the car.

The aim to get familiar with the software libraries and the information offered by the Arduino environment, the GPS utilizing standard NMEA 0183 and the SIM900 with AT commands was not fully reached. Getting familiar with the NMEA 0183 was not perfectly achieved, but the SIM900 AT commands were comprehensively covered. Arduino libraries were examined and studied. Required functionalities were found and utilized.

The GSM phone call is useful for the burglar alarm system, but there are some drawbacks to be considered as a further work for development. It was not studied on how the burglar alarm system works when e.g. the holder is talking in the phone while alarm call is arriving. In this case, an additional alarming SMS could be one solution. It is also considered that the GPRS connection could be used in case that mobile phone application is developed for the burglar alarm system.

7. CONCLUSION

During the development of the system an anti-theft hardware configuration was planned and the comprehensive features were defined for the system. The hardware parts were easily found for the purposes of the burglar alarm. Two fatal problems appeared while engineering the software. Firstly, lack of the AT-commands for receiving phone calls in SIM900 and secondly 'NO CARRIER' message problem in the GGS. There was also a problem to connect the GGS with the operator, and the testing did not reach the point of examining the received phone calls. The system received and replied to AT-commands in the serial bus.

The functional requirements were defined and the hardware was chosen. The comprehensive software was implemented except the fatal problem related sections. Testing was done for the whole system except the problematic areas mentioned above. Finally, the achievements were explored and the final outcome was discussed.

It was concluded that an anti-theft system is a relatively wide project for a bachelor thesis. More time could have been spent on project planning for hardware query and software selection. The evaluation of the Arduino used in the corresponding systems was done, but there was not much information available. As a result, it was concluded such a burglar alarm system does not exist and it was decided to start engineering it. Important findings are that Arduino is a useful hardware and software for this kind of system and can be extended well enough for an anti-theft requirements nowadays.

The evaluation of what are the current requirements for an anti-theft system in terms of increasing electronics in the cars could be explored as further work. For example, electric and hybrid or gas cars could be targets for a burglar alarm system.

REFERENCES

1. Mika Männistö, Modification of Passenger Cars under the Laws of Finland. Date of retrieval 23 September 2014. Available at:
<http://theseus.fi/bitstream/handle/10024/76637/Opinnaytetyo%20-%20Mika%20Mannisto.pdf?sequence=1>
2. Arduino, PIRsense code. Cited 16.9.2014,
<http://playground.arduino.cc/Code/PIRsense>
3. M. Moulet, M.-B. Pautet, The GSM System for Mobile Communications, Cell & Sys., Palaiseau, France, 1991
4. NMEA,NMEA 0183 Standard. Cited 23.9.2014
http://www.nmea.org/content/nmea_standards/nmea_0183_v_410.asp,
5. Arduino, Digital pins. Cited 16.9.2014,
<http://arduino.cc/en/Tutorial/DigitalPins>
6. What is arduino?, Cited 23.9.2014. <http://arduino.cc/>
7. Arduino, Arduino Mega 2560. Cited 15.9.2014,
<http://arduino.cc/en/Main/arduinoBoardMega2560>
8. GlobalSat, PRODUCT USER MANUAL GPS RECEIVER ENGINE BOARD EM-406A. Cited 20.9.2014. https://www.sparkfun.com/datasheets/GPS/EM-406A_User_Manual.PDF
9. SE-10 PIR Sensor Module Memo. Cited 19.9.2014,
<https://www.sparkfun.com/datasheets/Sensors/Proximity/SE-10.pdf>
10. Cooking-hacks, SIMCom SIM900 AT Command Manual_V1.05. Cited 19.9.2014,
http://www.cooking-hacks.com/skin/frontend/default/cooking/pdf/SIM900_AT_Command_Manual.pdf
11. Cooking-hacks, GPRS/GSM Quadband Module for Arduino Tutorial (SIM900). Cited 19.9.2014, <http://www.cooking-hacks.com/documentation/tutorials/arduino-gprs-gsm-quadband-sim900>
12. Arduino, Connecting a Parallax GPS module to the Arduino. Cited 20.9.2014,
<http://playground.arduino.cc/Tutorials/GPS>
13. T743503 Sulautetut ohjelmistosovellukset, OAMK 2013

SCRUM PLAN OF THE THESIS

APPENDIX 1

| | | | |
|------------------------|---|----------------------|--------------|
| | COMPANY NAME: OAMK | | Waiting |
| | PROJECT NAME: Car Alarm System | | Started |
| | SCHEDULE OF THE PROJECT: 1.1.2013-15.5.2013 | | Done |
| | | | |
| Essential | Feature description/story/functionality/quality requirement | Sprint number | State |
| 1 | Look for hardware | | Done |
| 2 | The system sets Anti-theft on when alarm is active | | Started |
| 3 | As a user I can receive an alarm call to my mobile phone | | Started |
| 4 | As a user I can receive SMS messages from the system | | Started |
| 5 | The system sends a message if battery needs to be charged | | Waiting |
| 6 | The system calls into owners phone if thief enters in the car | | Started |
| 7 | As a user I can ask location of my car in case of thief is moving the car | | Started |
| 8 | The Arduino Mega can control the Anti theft relay | | Waiting |
| 9 | As a user I can set the system active and inactive by calling to system | | Started |
| 10 | In active mode Arduino uses PIR sensor to notice burglary | | Started |
| 11 | System reads NMEA from GPS antenna | | Waiting |
| 12 | If PIR sensor detects bulgalary the system alarms the owner | | Started |
| 13 | Table of contents | | Done |
| 14 | Preface | | Done |
| 15 | Functional requirements | | Done |
| 16 | Chapter about hardware | | Done |
| 17 | Chapter about software | | Done |
| 18 | Chapter about outcome | | Done |
| 19 | Speculation | | Done |
| 20 | Tiivistelmä | | Done |
| 21 | Abstract | | Done |
| 22 | Indroduction | | Done |
| 23 | Final implementation | | Waiting |
| High priority | Feature description/story/functionality/quality requirement | Sprint number | State |
| 24 | | | Waiting |
| 25 | | | Waiting |
| 26 | | | Waiting |
| 27 | | | Waiting |
| 28 | | | Waiting |
| 29 | | | Waiting |
| 30 | | | Waiting |
| 31 | | | Waiting |
| 32 | | | Waiting |
| 33 | | | Waiting |
| Medium priority | Feature description/story/functionality/quality requirement | Sprint number | State |
| 34 | Testing | | Started |
| 35 | As a user I can manage anti thefting | | Waiting |
| 36 | | | Waiting |
| 37 | | | Waiting |
| Low priority | Feature description/story/functionality/quality requirement | Sprint number | State |
| 38 | | | Waiting |
| 39 | | | Waiting |
| 40 | | | Waiting |
| Other | Customer-/System requirement | Sprint number | State |
| | | | |
| | | | |

LOOP FUNCTION

APPENDIX 2

```
void loop() {
  if(Serial.available()){ // GSM
    Serial.println("AT+CMGR=1"); //Reads the first SMS
    Serial.flush();
    int x;
    int data[]= {255};
    for (x=0;x < 255;x++){
      data[x]='\0';
    }
    x=0;
    do{
      while(Serial.available()==0);
      data[x]=Serial.read();
      x++;
      if(data[x-1]==0x0D&&data[x-2]==""){
        x=0;
      }
    }while(!(data[x-1]=='K'&&data[x-2]=='O'));

    data[x-3]='\0'; //finish the string before the OK

    Serial.println(data); //shows the message
    if(data[0]=='A'&&data[1]=='c'&&data[2]=='t'&&data[3]=='i'&&data[4]=='v'&&data[5]=='a'&&data[6]=='t'&&data[7]=='e'){
      setAlarmsystemActive=true;
      Serial.println("AT+CMGDA=DEL ALL");
    }

    if(data[0]=='D'&&data[1]=='e'&&data[2]=='a'&&data[3]=='c'&&data[4]=='t'&&data[5]=='i'&&data[6]=='v'&&data[7]=='a'&&data[8]=='t'
    &&data[9]=='e'){
      setAlarmsystemActive=false;
    }
    delay(5000);
  }
  if(Serial1.available()&&locationAsked!=false){ // GPS
  }
  if(setAlarmsystemActive!=false&&activated!=true){
    delay(5000);
    digitalWrite(pirOnePin,LOW);
    delay(50000);
    setAlarmsystemActive=false;
    activated=true;
  }
  if(setAlarmsystemActive!=true&&activated!=false){
  }
  if(activated=true){
    if(digitalRead(pirOnePin)==HIGH){
      motionDetected=true;
      delay(500);
    }
  }
  if(motionDetected != false) {

    Serial.write("ATD");
    Serial.write(HPN);
    Serial.write(",");
    delay(50000);
    Serial.write("ATH");
    delay(5000);
  }
}
}
```


PUBLICATION PLAN

APPENDIX 3

| | | | |
|--|------------------------|-----------------------|------------------------|
| NAME OF THE PROJECT | | Car alarm system | |
| SCHEDULE OF THE PROJECT | | 01.01.2013-28.5.2013 | |
| MEMBERS IN PROJECT | 1 | | |
| SPRINTS IN PROJECT | 4 | | |
| HOURS FOR A DAY | 6 | | |
| RECOURCES OF THE PROJECT | | | |
| MEMBERS | WEEKS TO USE | DAYS TO USE | HOURS TO USE |
| Sakari Järvelä | 20 | 60 | 360 0 0 |
| OVERALL | 60 | | 360 |
| SPRINT 1 SCHEDULE | | | |
| | | 01.01.2013-4.2.2013 | |
| MEMBERS | WEEKS IN SPRINT | DAYS IN SPRINT | HOURS IN SPRINT |
| Sakari Järvelä | 5 | 15 | 90 0 0 |
| OVERALL | 15 | | 90 |
| SPRINT 1 GOAL: ALL GSM FEATURES READY | | | |
| SPRINT 2 SCHEDULE | | | |
| | | 05.02.2013-11.3.2013 | |
| MEMBERS | WEEKS IN SPRINT | DAYS IN SPRINT | HOURS IN SPRINT |
| Sakari Järvelä | 5 | 15 | 90 0 0 |
| OVERALL | 15 | | 90 |
| SPRINT 2 GOAL: GPS AND PIRSENSOR FEATURES READY | | | |
| SPRINT 3 SCHEDULE | | | |
| | | 12.03.2013-15.4.2013 | |
| MEMBERS | WEEKS IN SPRINT | DAYS IN SPRINT | HOURS IN SPRINT |
| Sakari Järvelä | 5 | 15 | 90 0 0 |
| OVERALL | 15 | | 90 |
| SPRINT 3 GOAL: ANTI THEFT CONTROLLING READY | | | |
| SPRINT 4 SCHEDULE | | | |
| | | 16.04.2013-20.5.2013 | |
| MEMBERS | WEEKS IN SPRINT | DAYS IN SPRINT | HOURS IN SPRINT |
| Sakari Järvelä | 5 | 15 | 90 0 0 |
| OVERALL | 15 | | 90 |
| SPRINT 4 GOAL: WORKING PROTOTYPE OF THE SYSTEM | | | |
| PUBLICATION | | | |
| | | 21.5.2013-28.5.2013 | |
| MEMBERS | WEEKS | DAYS | HOURS |
| Sakari Järvelä | 1 | 7 | 39,9 0 0 |
| OVERALL | 7 | | 39,9 |
| GOAL OF PUBLICATION: Thesis ready | | | |