Remote Control of Devices by Means of GSM Network

MARTIN POSPISILIK
TOMAS SMEKAL
MILAN ADAMEK
PETR NEUMANN
TOMAS DULIK
Faculty of Applied Informatics
Tomas Bata University in Zlin
Nad Stranemi 4511, 760 05 Zlin
CZECH REPUBLIC
pospisilik@fai.utb.cz

Abstract: - This paper describes a possibility of utilization of a GSM network for remote controlling of various appliances as house gates, heating, sunblinds etc. A project of a remote gate controller based on Arduino platform and GSM data transfers is mentioned hereby, but the purpose of utilization of the remote controller can be easily changed to another application. The main idea of the paper is based on the fact that although most remote controllers for entrance gates operate on free frequencies 433 or 868 MHz, their user's comfort is limited. On the other hand, GSM networks enable encrypted and bidirectional data transfers, allowing the user to communicate with the controlled device by means of his personal mobile pohne. Moreover, the user can be informed on any undesirable states that occured at the remote device. For example, he can be informed via his mobile phone when his house's gate is irrupted.

Key-Words: - GSM Network, Data transfers, Arduino, GSM Shield, Remote controlling, Gate controller

1 Introduction

Not only in Europe, the GSM (Global System for Mobile Communication) is widely spread, allowing transferring the speech signals but supporting data transfers as well. The most common devices to be connected to this network are cellular phones. On the other hand, there exist other devices that embody the interface needed to establish the connection. The broadest possibilities are then given by standalone GSM modules that are designed to be embedded into arbitrary devices. In this work, GSM shield based on SIM900 chip has been used.

1.1 GSM Network

The block diagram of the GSM network is depicted in figure 1. The description of the abbreviations is provided in table 1.

The brief description of the operation of the GSM network is as follows: The mobile station is always identified by its IMEI (International Mobile Equipment Number) and its user is identified by his SIM (Subscriber Identification Module). By means of a radio connection it is connected to the system of base stations (BSS). The BSS is connected to the Network Switching Subsystem (NSS). The BTS stations of the BSS are spread around the covered area, creating a cellular system of transceivers. While the BSS is responsible for maintaining the connection between the mobile stations and the network, the NSS is responsible for authorization of the devices connected to the network (using HLR, AUC and VLR) and for switching the data in the communication network. The service of the whole system is then provided by the Operational and Support Subsystem (OSS).

1.1.1 Data Transfers in GSM Network

The development of the GSM network has been launched in 1982. Although it was intended to be used for speech transfers, the packetizing of the digital speech signal allowed to extend its functionality for transferring of general data. In 2000 the standard GPRS (General Packet Radio Service) has been implemented, mainly to allow more comfortable connection of the mobile stations to the Internet.

The GPRS system is designed to use the already set system of radio channels, but it treats the physical layer of the system in a different way. The implementation of the GPRS system in GSM networks is depicted in figure 2. As depicted, the implementation of the GPRS system to the GSM network consisted in adding several functionalities most of which are based on software. The Base

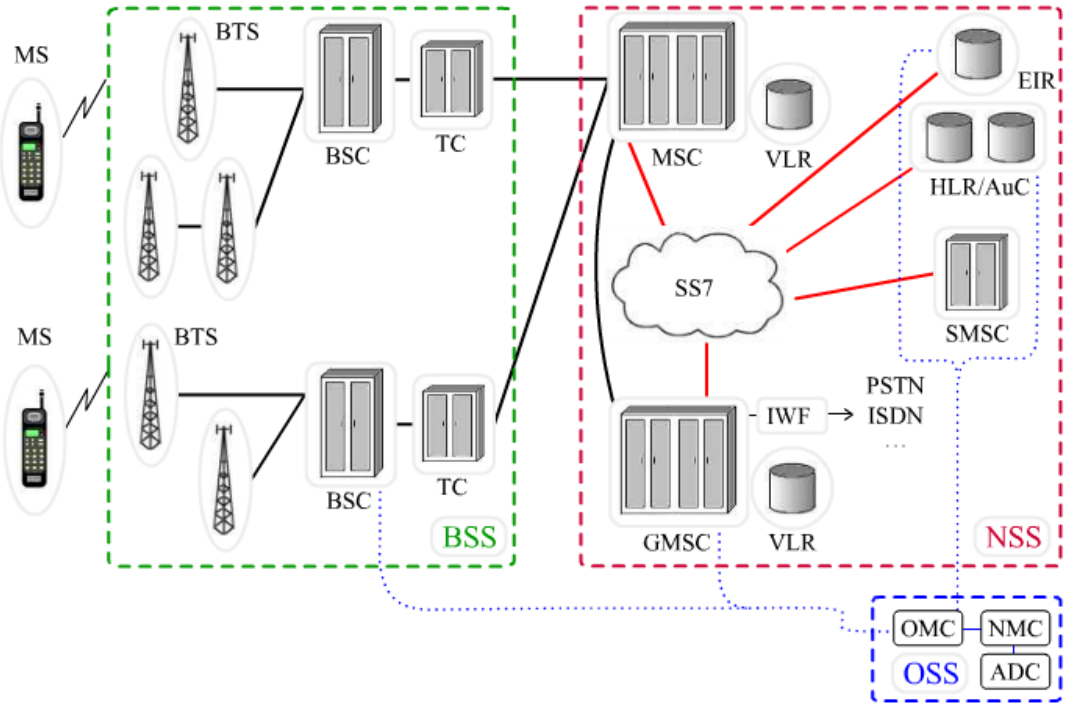

Fig. 1 GSM network structure [4]

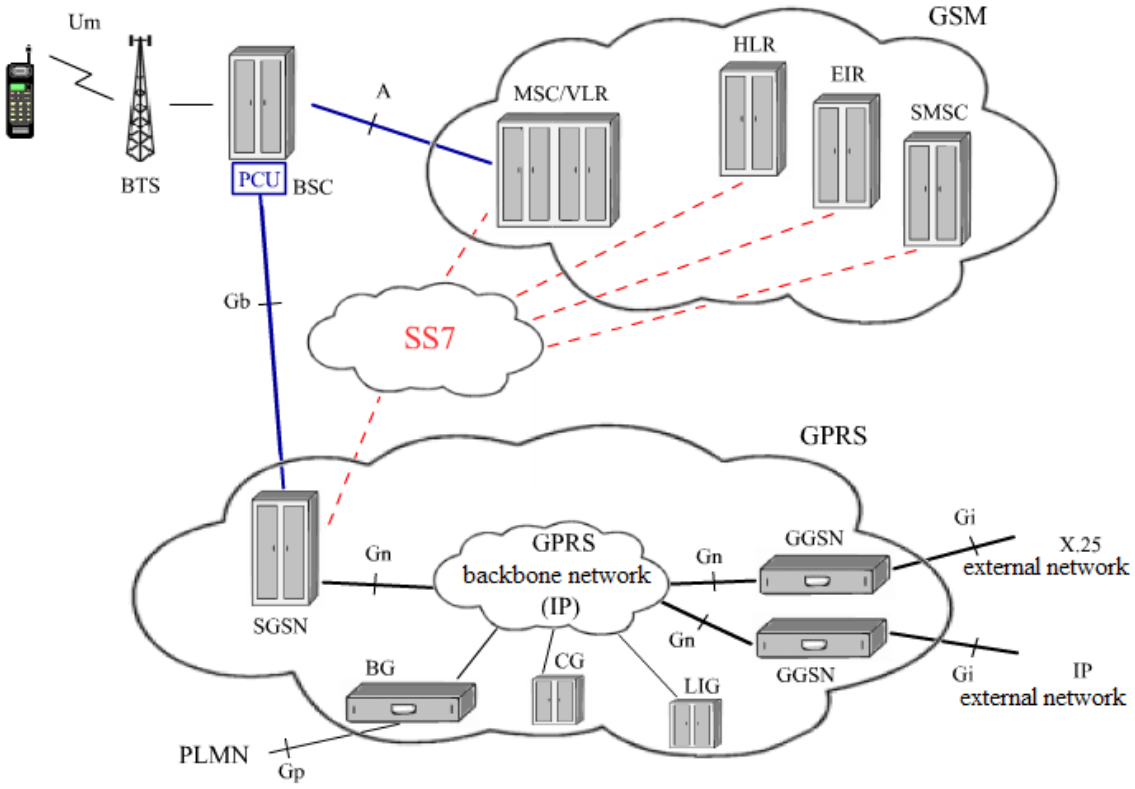

Fig. 2 Implementation of GPRS system to the GSM network [4]

Table 1. Description on abbreviations depicted in Fig. 1

Abbreviation	Desctiption
MS	Mobile station
BTS	Base transceiver station
BSC	Base station controller
TC	Transcoder
BSS	Base station subsystem
NSS	Network switching subsystem
MSC	Mobile switching centre
GMSC	Gateway mobile switching centre
VLR	Visitor location register
SS7	Signalling system 7
IWF	Interworking functionality
EIR	Equipment identity register
HLR	Home location register
AUC	Authentication centre
SMSC	SMS centre
PSTN	Public switched telephone network
ISDN	Integrated services digital network
OSS	Operation subsystem
OMC	Operation and maintenance centre
NMC	Networking management centre
ADC	Administrative centre

1.1.2 Security in GSM Network

There are several security aspects within the GSM network that should be taken into account:

- Calls interception,
- Lost phone abuse,
- User's account abuse,
- Etc.

There are two ways to assure the security of the GSM network:

- User's identification by means of SIM card and the appropriate codes (PIN, PUK),
- Data encryption.

The applied methods that are enlisted above determine the level of the GSM network's security. The authentication of the user is performed by means of his SIM card, PIN (Personal Identification Number) and PUK (PIN Unlocking Key). All devices are also given their unique IMEI number (International Mobile Equipment Identity) that is stored in both the mobile device and the EIR (see Table 1). The IMEI number is then sent to VLR and there it is included into one of the following lists:

- White list (devices that are allowed to connect to the network),
- Black list (devices that are not allowed to connect to the network – stolen phones etc.),
- Grey list (devices that do not support some specifications, malfunctioned devices etc.).

The process of the SIM card identification is depicted in figure 3. When turned on, the mobile device asks for the access to the network. In this step it sends its IMSI (International Mobile Subscriber Identity). To ensure the anonymity of the device, the network generates TMSI (Temporary Mobile Subscriber Identity) on the basis of IMSI and the IMSI is not in use until the next device's attempt to approach the network. The TMSI is then stored in the SIM card (device's side) and VLR (Network switching subsystem). Afterwards, in HLR the random number (RAND in figure 3) is generated. On the basis of this number another number SRES (Signed Response) is calculated, employing the authentification key Ki and encrypting algorithm A3 on the side of the mobile device. The SRES is also computed on the network's side by means of the Authentification Centre (AuC) and the authorization is successful if both values are equal. The real identification process is rather complicated and its detailed description can be found in [4].

The data encryption is based on the key and the bit sequence generated by the appropriate encryption algorithm. The process of data encryption is depicted in figure 4. More information can be found in [4].

The encryption algorithms A3 and A8 are not specified in details, the agreement between the SIM card producer and the network's operator is needed.

As can be seen in figures 3 and 4, the bottleneck of the GSM network's security is the radio interface, because the IMSI can be captured while the device is trying to connect the network and misused by means of a fake mobile device.

1.2 Remotely Controlled Gates

The remotely constructed gates are usually based on one of the following constructions:

- Rail based,
- Self-supporting.

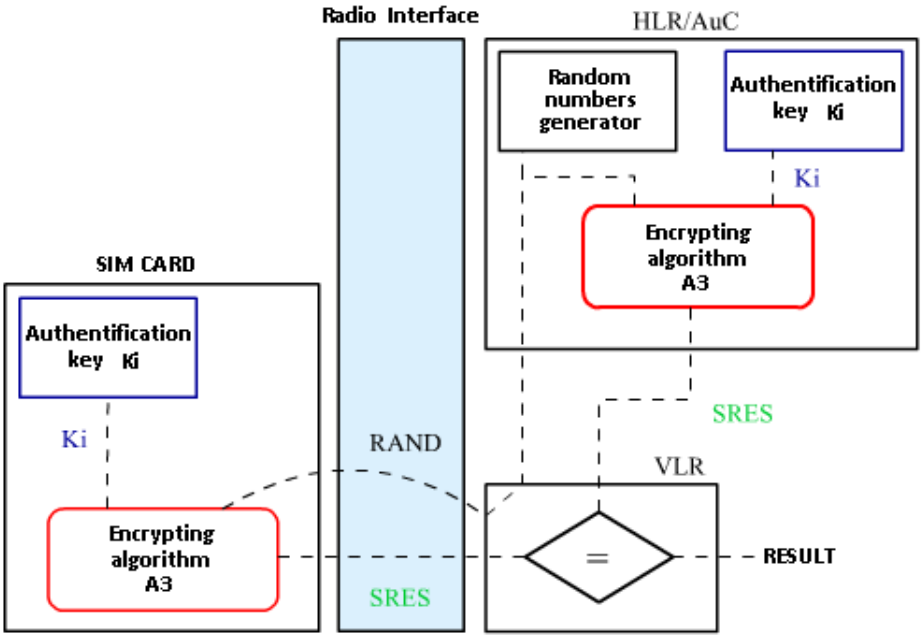

Fig. 3 Principle of SIM card's identification in the GSM network [4]

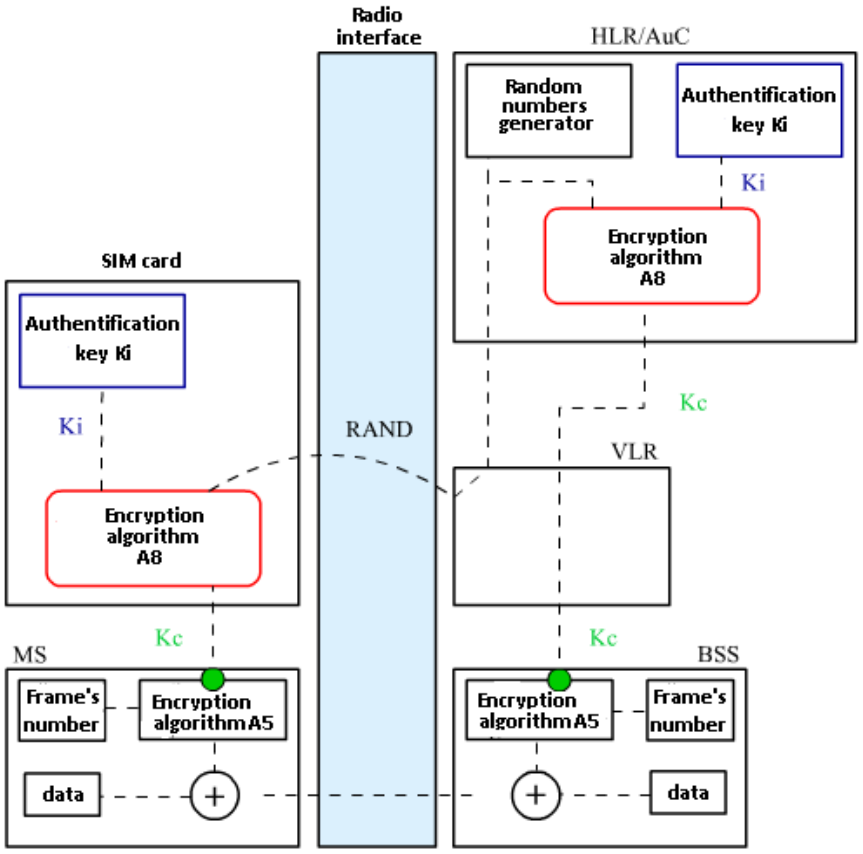

Fig. 4 Principle of data encryption in GSM network [4]

Both of these types are usually equipped with unified actuators that enable opening and closing of the gates together with supplementary functions as giving the information on the position and/or movement of the gate, self-diagnostics, etc. The area of the gate's range is monitored by sensors detecting obstacles that could collide with the gate when opening or closing. Usually the sensors based on infrared rays.

The example of the construction of the remotely controlled gate is depicted in Fig. 5.

The conventional remote controllers operate at the frequency of 433 or 868 MHz respectively. The transceivers are distinguished by the unique code. Three types of authentication can be applied:

- Fixed code; the receiver and the transmitter use always the same code,
- Floating code; every time the transmitter is required to send a command, it uses different code. The receiver never accepts the same code again,
- Floating code with Antiscan; if the aggressor tries to guess the code, the receiver is blocked.

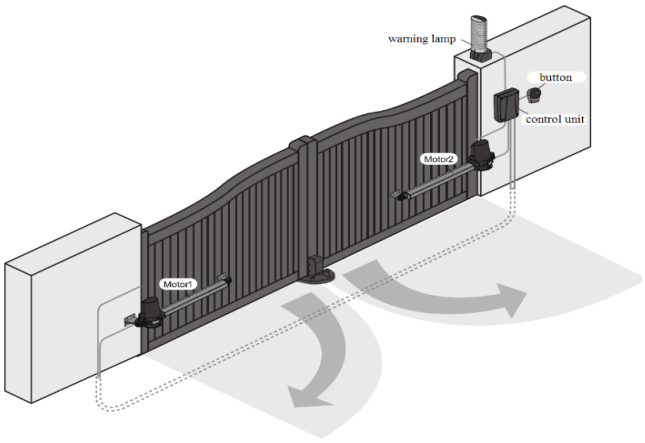

Fig. 5. Example of construction of a remotely controlled gate with two drivers [5].

Application of controllers employing the GSM network can widen the functionality in the following aspects:

- A list of numbers being accepted can be set at the receiver.
- There is a possibility of bidirectional data transfers and sending messages to the user, for example when the gate is violently opened.
- The communication with the gate's controller can be processed easily by means of the user's cellular phone. The example of this solution can be found in [1].

1.3 Arduino

Arduino is an open-source project that started in 2005 in Interaction Design Institute as a low cost development kit for applications with microcontrollers by Atmel. It is a physical platform with a microcontroller and several peripherals that is programmed by means of a simple development environment. Modular construction allows connecting expansion boards, called also as "shields". When creating the program, the language C++ can be applied as well. More information on Arduino project can be found in [6].

Table 2. Parameters of Arduino Uno.

Parameter	Value
Input voltage (operational)	7 - 12 V
Input voltage (limit)	6 – 20 V
Digital I/O pins	14
Analog inputs	6
Maximum current per a pin	40 mA
Flash memory	32 kB
SRAM	2 kB
EEPROM	1 kB
Clock	16 MHz
Dimensions (flat plane)	68.6 x 53.4 mm
Height	25 g

For the purposes of this project, the Arduino model UNO has been selected. It is based on the microcontroller Atmel ATmega328p. It is equipped with 14 I/O pins. Some of these pins support PWM, analog inputs and/or connecting of a 16 MHz crystal. Using the shields, the following peripherals can be embedded in the unit: Ethernet, Bluetooth, LCD display, and, not least, the GSM module.

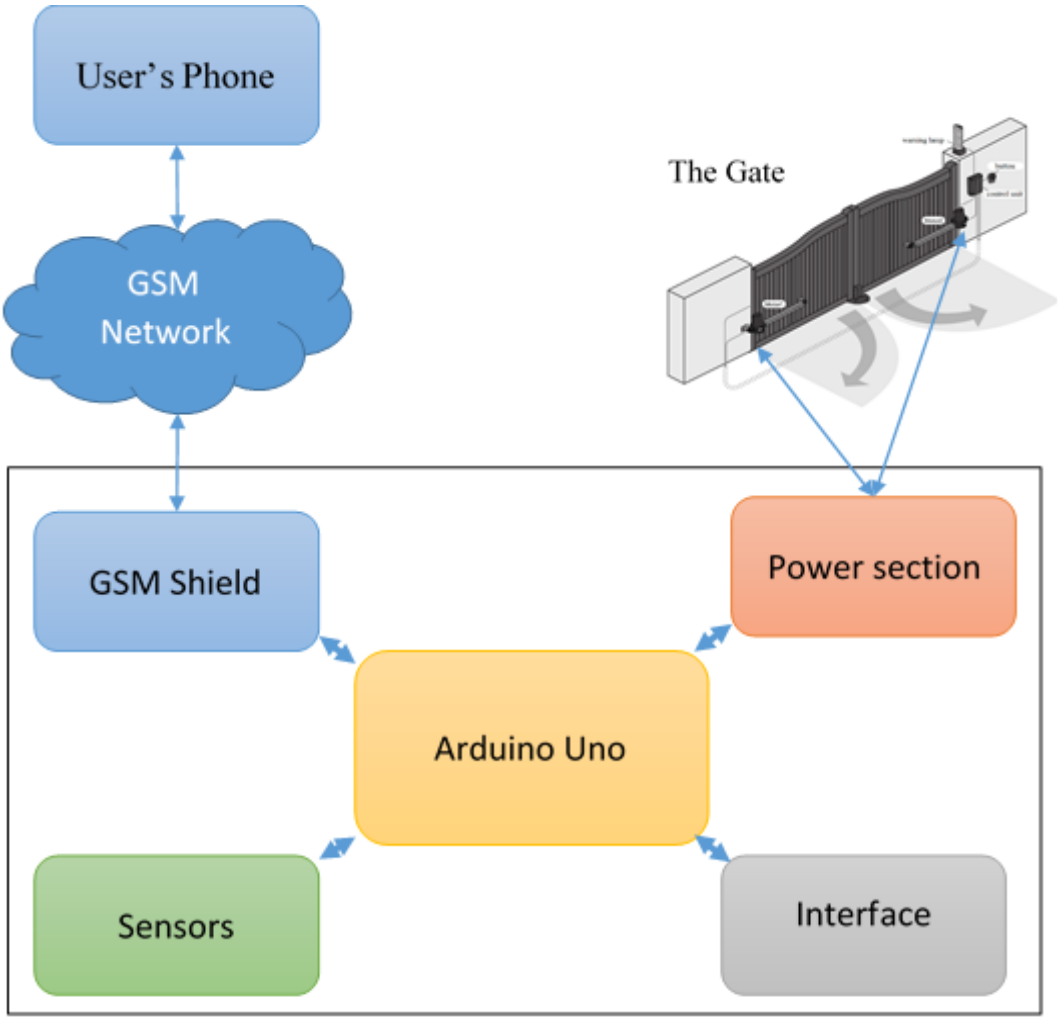

Fig. 6 Concept of the remote controller using GSM network

The parameters of Arduino Uno are provided in the table 2.

1.4 GSM shield

The GSM shield applicable for Arduino Uno that was used in this project, is based on the controller SIM900 by SIMCOM. It supports GSM/GPRS standards and operates at the frequencies of 850, 900, 1,800 and 1,900 MHz. The shield can be connected directly to the board of Adruino Uno by set of pins mounted at the bottom of the printed circuit board.

More information can be found in [7].

2 Project description

The hereby described project consists in development of a low-cost remote controlled driver for opening and closing of the gates. For the development, the above mentioned development kit Arduino Uno equipped with a GSM shield has been used. However, it was necessary to create another shield to provide interface between the controlling unit and the motors of the gate's driver unit.

The block diagram of the project solution is depicted in figure 6. The main development work consisted in making the Power section shield hardware, creating the software for the Arduino Uno board and making a model of the gate being controlled by the developed unit.

As depicted in the figure 6, the main board interacts with four peripherals: GSM Shield,

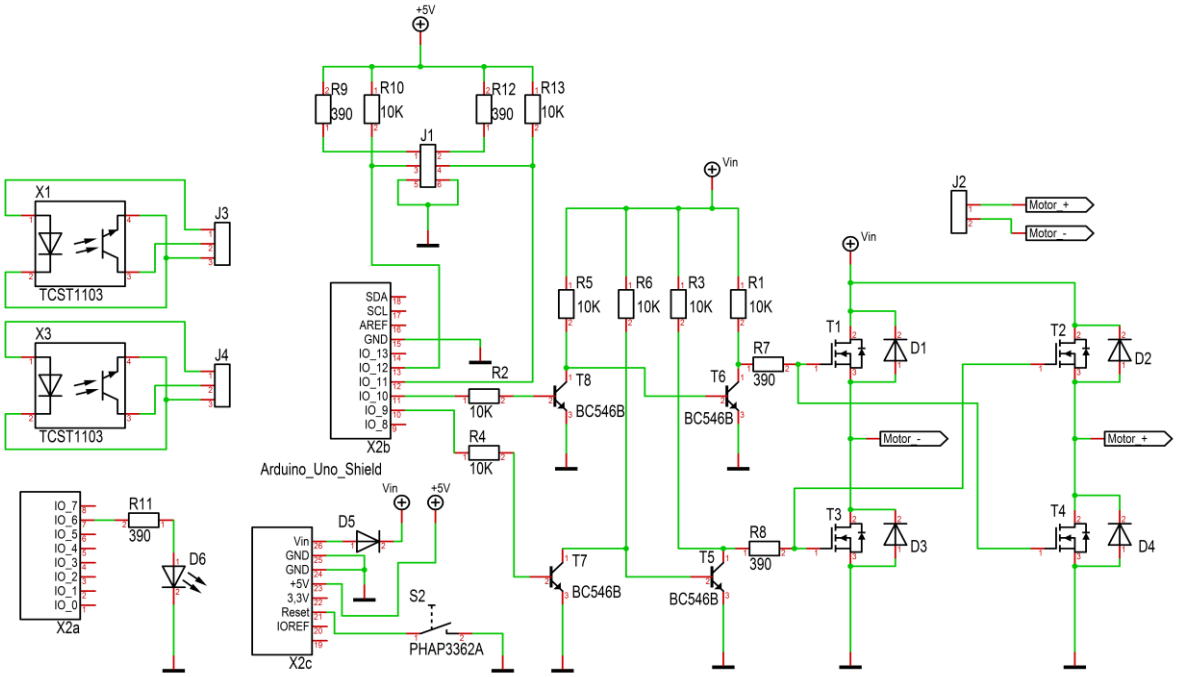

Fig. 7 Connection diagram of the Power section shield [5]

Sensors, Power section and User's interface. While the GSM shield and the main board of Arduino Uno were purchased from the manufacturer, the Power section together with the set of end-switch sensors must have been created within the development project. Details on the project are described in the following chapter.

3 Implementation

Within this chapter, hardware and software implementation is described.

3.1 Hardware

As noted above, as the Arduino Unit and its GSM shield were bought as the submodules, the another shield with the Power section had to be made in that way so it fit the Arduino Unit PCB directly by means of the connectors. Moreover, separated optical sensors were also created to check the obstacles in the area of the gate.

The connection diagram of the Power section shield is depicted in figure 7. The shield can be stacked to the units by means of the connectors X2a, X2b and X2c. The power section itself consists of a H-bridge, that is created by means of transistors T1 to T4 and the drivers of these transistors, that are made with transistors T5 to T8. The bridge is driven directly from the Arduino unit, using a PWM modulation that enables the possibility of controlling the speed of the gate's propulsion. The position of the gate is monitored by a pair of photocells that are connected to the connector J1. In the figure 7 the connection of these photocells is also depicted in the form of standalone circuits. The photocells employ TCST1103 modules that are marked as X1 or X3 in the diagram. Because the power consumption of the model of the gate is not high, the power supply to the Power unit is taken directly from the power supply network of the Arduino unit. The protection diode D5 was added to suppress the voltage ripples generated by the driven motor that could get back to the Arduino unit. If the output current of the driver is insufficient, more powerful H-bridge can be connected to the power output of the shield. The power output is conducted through the connector J2.

The stack of three printed circuit boards can be seen in the figure 8. The PCBs are (from the top):

1. Power section shield,
2. GSM/GPRS shield,
3. Adruino Uno.

3.2 Software

The software implemented in the Arduino unit must distinguish between the following states:

- Opened – the gate is opened, the motor is off, the stop sensor of this position gives logical 1 at its output,
- Closed – the gate is closed, the motor is off, the stop sensor of this position gives logical 1 at its output,
- Opening – both stop sensors give logical 0 at their outputs and the motor is turning in the direction to open the gate,
- Closing – the stop sensors give logical 0 at their outputs and the motor is turning in the direction to close the gate,
- Error – occurs when both stop sensors give the logical 1 at their outputs or both sensors give the logical 0 at their outputs and the motor is off.

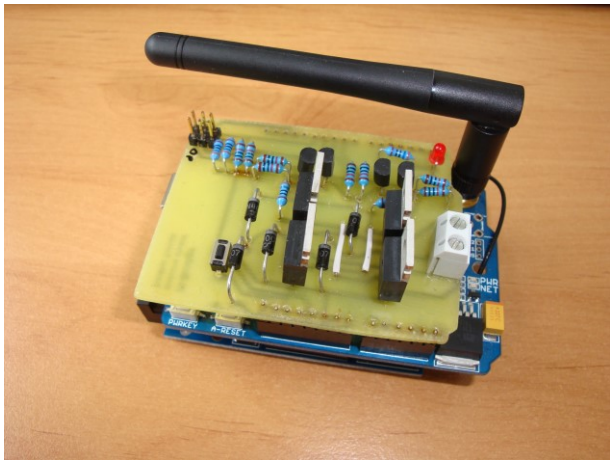

Fig. 8 Completed gate controller [5]

In this project, communication by means of SMS has been chosen. Therefore, the software must support the following commands that are sent via SMS:

- Open – calls the method gate.Open() that process opening of the gate,
- Close – calls the method gate.Close() that process closing of the gate,
- Slower – this command is determined by the set of characters "-" entered consecutively. By this command the speed of the gate's movement is decreased according to the number of the characters sent. The information on required speed of the gate is stored in the memory.
- Faster - this command is determined by the set of characters "+" entered consecutively. By this command the speed of the gate's movement is increased according to the number of the characters sent. The information on required speed of the gate is stored in the memory.
- Status – this command calls the method gate.Status() the result of which is to send the information on the status of the gate to the user by means of the SMS.

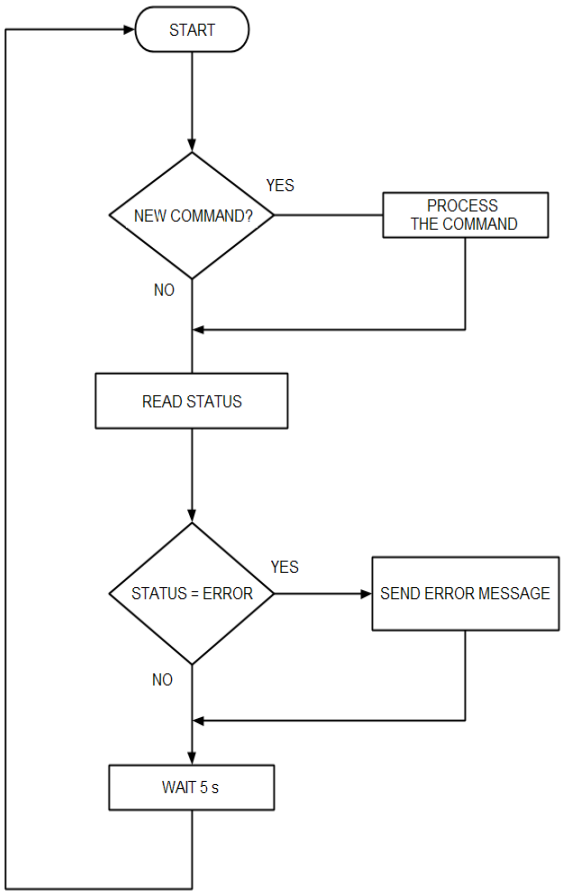

Fig. 9 Flowchart of the software loop [5]

The application is repeated in a loop the diagram of which is depicted in the figure 9.

The Arduino's producer provides an open-source library GSM_GPRS_GPS_Shield that includes a set of classes for treating the GSM modules based on the controllers SIM900 and/or SIM908.

The GSM communication is implemented in two C++ code files the examples of whose can be found below. These files are:

- GSMCommunication.h,
- GSMCommunication.cpp.

GSMCommunication.h:

```
# ifndef GSMCOMMUNICATION_H
# define GSMCOMMUNICATION_H
# include <arduino .h>
```

```
# include " ICommunication .h"
# include <SIM900 .h>
# include <sms.h>
# define GSM_POWER_PIN 7
class GSMCommunication : public
ICommunication
{
private :
SMSGSM sms;
public :
GSMCommunication ();
~ GSMCommunication ();
virtual void Init ();
virtual byte rcvMessage ( char ** message ,
char ** phoneNumber );
virtual void sendMessage ( char * message
);
virtual void sendMessage ( char * message ,
char * phoneNumber );
};
# endif
```

GSM Communication.cpp

```
# include " GSMCommunication .h"
# define SMS_LENGTH 161
# define NUMBER_LENGTH 15
# define AUTH_POS_FIRST 1 //
# define AUTH_POS_LAST 20 //
GSMCommunication :: GSMCommunication ()
{
Init ();
}
GSMCommunication ::~ GSMCommunication ()
{}
void GSMCommunication :: Init ()
{
// gsm. begin (9600) ;
gsm. begin (14400) ;
// gsm. begin (19200) ;
}
byte GSMCommunication :: rcvMessage ( char
** message , char ** phoneNumber )
{
int smsPosition ;
if (( smsPosition = sms. IsSMSPresent (
SMS_UNREAD )) > 0)
{
* phoneNumber = ( char *) malloc (
NUMBER_LENGTH * sizeof ( char ));
* message = ( char *) malloc ( SMS_LENGTH *
sizeof ( char ));
char authStatus = sms. GetAuthorizedSMS (
smsPosition , * phoneNumber , * message ,
,! SMS_LENGTH , AUTH_POS_FIRST ,
AUTH_POS_LAST );
sms. DeleteSMS ( smsPosition );
if( authStatus != GETSMS_AUTH_SMS )
{
return 0;
}
return 1;
}
return 0;
}
void GSMCommunication :: sendMessage ( char
* message )
{
char n [20];
sms. SendSMS (( byte )1, message );
}
void GSMCommunication :: sendMessage ( char
* message , char * phoneNumber )
{
char n [20];
sms. SendSMS ( phoneNumber , message );
}
```

4 Conclusions

This paper provides a description on construction of the controller of the remotely controlled entrance gates that can be commanded via the user's cellular phone, employing data transfers in GSM network. In the framework of this project the functional sample was created on the basis of the development kit Arduino Uno and two embedded shields. The picture of the completed unit can be found in the figure 8.

The created sample is fully operational, providing large possibilities to implement expansion functionalities. For example, as the hardware supports connection of additional units, a logging unit can be added to monitor all attempts to control the driver. Another option is adding of the LAN unit that would allow connecting the driver to the LAN network.

The hereby described system can simply be modified in order to achieve its better performance in the following ways:

- The basic command can be activated just by ringing the remote controller from a known phone number. This decreases the costs of data transfers.
- The states of the controller or of the gate itself can be logged on a storage device.
- When adding Ethernet module, the commands can also be sent directly via Internet.
- Using of more position sensors would allow proportional control of the speed of the gate's closing and opening.

Acknowledgements

This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic within The National Sustainability Programme Project No. LO1303 (MSMT-7778/2014) and also by The European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089.

References:

[1] X1. S. F. Barrett, *Arduino microcontroller: processing for everyone!* (Morgan, 2012, ISBN 978-160-8458-592)
[2] X2. B. Evans, *Beginning Arduino programming: writing code for the most popular microcontroller board in the world* (Apress, 2011, ISBN 978-1-4302-3777-8)
[3] X3. T. Halonen, J. Melero, J. Romero, GSM, GPRS and EDGE Performance: Evolution Towards 3G/UMTS, 2nd Edition (John Wiley, 2003, ISBN 04-708-6694-2)
[4] X4. Richtr, T., *Basic structure of GSM network* (Online: URL http://tomas.richtr.cz/mobil/gsm-strukt.htm)
[5] X5. Smékal, T., *The Remote control of gates using a GSM unit* (Diploma Thesis, 2015, Supervizor: M. Pospisilik)
[6] X6. Arduino - ArduinoBoardUno. (Online: URL http://www.arduino.cc/en/Main/ArduinoBoardUno)
[7] X7. SIM900 Quad-Band GPRS shield with Micro SD card slot - Epalsite Wiki (Online: URL http://wiki.epalsite.com/index.php?title=SIM900_Quad-Band_GPRS_shield_with_Micro_SD_card_slot)