Andrzej Romanowski*, Jakub Betiuk*, Zbigniew Frątczak*

Method for Automatic Driver Notification about Detection of the Emergency Vehicles

1. Introduction

Apart from individual transportation, it cannot be forgotten that cars are used in service of human life and security. Day by day they provide help to the emergency response services so as they can get to the required place as quickly as possible. No matter if it is a fire, a crash or a robbery, emergency groups depend on their specially equipped vehicles. What is more, in order to provide fastest possibility of reaction they are given right of way while responding to the emergency. This action is easily recognizable by flashing lights and sound signals of the privileged vehicle. As a main method of transport of the emergency groups, cars have very important mission to accomplish. One may think that it is an easy task to get from point A to the point B however in real life it appears to be more complicated. Let's think of an ambulance. It is a vehicle under a disposition of a health centre or a hospital. It may be used for transportation of hospital residents, blood and casualties. However its crucial role is to provide help to people injured in various accidents or being seriously ill. In such condition it is critical to get to the accident site within several minutes, as it may decide of life and death of the victim. Imagine a situation when someone calls emergency. Person at emergency dispatch center passes the request to another unit, either hospital or health centre that is closest to the place where accident happened. Group which receives such request begins the action. They jump into the car and leave the garage. Driver turns on both the sound and light signals and does his best to get the medical crew to the place as fast as possible. However, few hundred meters from the hospital they have to face the traffic jam. Such sight is very common in Lodz, a 750 thousand citizens city, especially in the area of city centre, as during the rush hours it may take over an hour to break through traffic jams while driving civil car in casual conditions.

1.1. Practical background

Apart from human mistake there are some other factors highly influencing the perception of the emergency vehicle like audio systems and sound-proofing the car. These days

* Computer Engineering Department, Technical University of Lodz, Poland

major car brands fight for customers on many fields, starting from safety of the cars produced up to additional equipment and comfort. Cars from different classes have different target customers that have different priorities. For example high class vehicles are meant to provide extraordinary comfort of traveling. That is associated with optimal conditions in the passenger compartment. Usually it is achieved by the use of air conditioning installation, comfortable seats, good external sound proofing and extended audio system. The main idea is to isolate the driver from the outside noise and make him feel as if he was in the closed glass bulb. It may be very relaxing, however no one is interested whether it has influence on the ability to notice certain events that occur in the neighborhood of the car. What is more, every year car manufacturers improve the components used to build a car to make them more silent and comfortable. It gets to the point where driver is unable to recognize the sounds in his area, including sirens of the incoming emergency vehicles. As was mentioned before, the emergency vehicles are also equipped with flashing lights that work great together with sound, especially in the evening or night time, however in bright daylight the lights may remain unnoticeable for the person driving the car. They may remain unseen due to low contrast with the surroundings or may be blocked by another vehicle. Therefore it may be assumed that the main method of notification used by the emergency vehicles is sound as it doesn't require the vehicle to be in field of vision of the driver and may be easily noticed, no matter of the position of the vehicle, or time of the day.

Having that in mind and considering the fact that cars are getting more sound-proof, it may be concluded that the chances of the emergency vehicle to be spotted are dependent on the materials used to build the car. To conclude, the more silent the care is, the more difficult is for a driver to notice incoming emergency vehicle and react in time to yield the right of way without causing the privileged vehicle to slow down.

1.2. Motivation

Some people would say that the proposed idea to construct such safety device is pointless, however during the research the following conclusions were derived:

- Car producers do everything to improve comfort of their vehicles, what also involves extreme soundproofing. Most of nowadays premium cars are so soundproofed that driver hardly can hear engine of its own car.
- Sound systems of current automobiles may be extended or improved in various different ways starting from simple subwoofer installation, up to making it 7.1 (or even more) home theater on the wheels. There are no restrictions of the level of the volume that may be produced by the car driven in the traffic.
- There are forums dedicated for people working in Emergency Services (i.e. http://www.ratmed.pl/forum [18]) where people share some important knowledge, but also they complain on drivers, who either can't hear that they are on the way or do not know how they should behave.
- Some drivers do not actually bother to give the way to an emergency vehicle, as they can't understand that incoming ambulance may actually drive to save life of real person who might be his relative.

To sum up, I believe it is high time to introduce an external device which will notify driver that emergency vehicle is in proximity. It won't decrease comfort of traveling by car at any point or won't force people to tone down their sound systems permanently. Installation of such device will give a driver several additional seconds to decide on what to do and those few seconds may save someone's life.

2. Research and Initial Requirements

2.1. Existing Solutions

In order to increase these chances several things can be done. These improvements could be divided into three categories with respect to the type of modifications they require. The groups are as follows:

- Solutions that require modification of the emergency vehicle;
- Solutions that require modification of the civil cars;
- Solutions that require modification of the city infrastructure, traffic lights, etc.

Each group represents solutions that were developed to ease the transit of the privileged vehicle through the city areas. They include solutions starting from automated traffic lights that change to green while ambulance is passing, through combinations of transmitters and receivers communicating via use of radio frequencies, up to signal processing. These groups will be discussed more precisely in the further parts of this document.

Technological advance leads to improvements in many fields, starting from the conquering of the space through the casual day improvements. One of such improvements was the invention of the car. Nowadays cars are available for almost anyone. Cars serve humans not only to ease their lives, but also to save them. Automobiles play great role in the health service. They enable quick transport of injured men to the hospitals or between one medical center and the other. They can drive fast, however sometimes it is impossible because of the traffic jams or other occurrences.

During decades people tried to provide systems capable of detection of emergency vehicles. For the purpose of this thesis, these solutions have been revised in order to find out what algorithms were used for detection and what problems were faced by the inventors.

Regarding the first group of possible improvements, e.g. relying on the modifications of the privileged vehicle itself, a group of solutions were proposed. They utilize the approach based on the radio transmitter-receiver system. In fact, most of them would require the modification of the traffic infrastructure, as well, in order to enable smooth 'green traffic light wave'. So the solutions require either a special radio-frequency emitter to be mounted on the emergency vehicle and broadcasting a fixed signal, or special spectrum analyzers to be mounted on each traffic lights, or both [5, 10, 12, 15, 17].

2.2. Theoretical considerations

The main drawbacks of these solutions are technical imperfections due to chosen frequencies (interference), processing difficulties (e.g. cascaded filters response within limited time), and/or high cost a troublesome implementation at each emergency car and traffic system. Therefore, the presented solutions is based on two main assumptions. First one is to focus on existing sound signals of emergency cars, in order not to modify whatsoever these vehicles and the traffic system in general. Therefore, the eventual implementation should result in a device, which can be applied in normal, civil cars. The second assumption is to use digital signal processing (DSP) to utilize its processing advantages. Discrete Fourier Transform (DFT), and more specifically, Fast Fourier Transform (FFT) are chosen to obtain representation of the time related function (e.g. sound signal) into its frequency domain, for further analysis [19].

DFT has some important properties that while unknown may lead to confusion. As DFT is most often used to process sets of the real numbers. Although, DFT is a powerful way of representing time domain sequence in its frequency domain, it is very ineffective. While considering large amounts of data, number of operations to be performed increases dramatically. Through different implementations, FFT at the base of 2 is most common and was utilized in this project. Radix-2 algorithms are considered simplest FFT algorithms [6, 9]. The decimation-in-time radix-2 algorithm introduced to this application recursively partitions DFT to two DFTs of the half length. One half contains even the other odd-indexed time samples. Results of the short DFTs are reused in order to calculate bigger parts, as a result reducing their time of calculation. In details, it rearranges the discrete Fourier transform into two parts, namely sum over even-numbered indices (n = [0, 2, 4, …, N–2]) and sum over odd-numbered indices (n = [1, 3, 5, …, N–1]) Eq. (1).

$$\begin{aligned}
x(k) &= \sum_{n=0}^{N-1}\left(x(n)e^{-\left(i\frac{2\pi nk}{N}\right)}\right) = \sum_{n=0}^{\frac{N}{2}-1}\left(x(2n)e^{-\left(i\frac{2\pi(2n)k}{N}\right)}\right) + \\
&+ \sum_{n=0}^{\frac{N}{2}-1}\left(x(2n+1)e^{-\left(i\frac{2\pi(2n+1)k}{N}\right)}\right) = \\
&= \sum_{n=0}^{\frac{N}{2}-1}\left(x(2n)e^{-\left(i\frac{2\pi nk}{\frac{N}{2}}\right)}\right) + e^{-\left(i\frac{2\pi k}{N}\right)}\sum_{n=0}^{\frac{N}{2}-1}\left(x(2n+1)e^{-\left(i\frac{2\pi nk}{\frac{N}{2}}\right)}\right) = \\
&= DFT_{\frac{N}{2}}\left[\left[x(0), x(2), ..., x(N-2)\right]\right] + W_N^k DFT_{\frac{N}{2}}\left[\left[x(1), x(3), ..., x(N-1)\right]\right]
\end{aligned} \tag{1}$$

Equation (1) above illustrates that DFT frequency outputs $X(k)$ can be calculated as a sum of the two length- N/2 DFTs – even-indexed and odd-indexed discrete time samples, where odd one is multiplied by twiddle factor:

$$W_N^k = e^{-i\frac{i\pi k}{n}} \tag{2}$$

Separation of time samples into two alternating groups is called decimation in time while radix-2 describes number of subgroups. Image below illustrates scheme of operation of such algorithm.

2.3. Hardware & Software Platform

Having considered all possibilities it was decided to choose a microcontroller produced by either ATMEL Company or Texas instruments. Both companies are well known on the market and their products are famous for reliability and functionality, however each of them has its advantages and disadvantages. It was finally decided to take on the Atmel MCU due to its price and support that enabled to focus on the processing part, omitting difficulties with setting up all the peripherals [1].

AVR32UC3A unit was released as a part of several Evaluation Kit Boards that contained typical devices that could cooperate with this microprocessor [4]. Most universal is EVK1100 that contains Ethernet socket, LED’s, LCD screen, and analog devices such as potentiometer or temperature sensor [7]. Another advantage of use of Evaluation Kit board is fact that it is equipped with ports dedicated for programming and debugging of the microcontroller. Especially Nexus connection that enables use of the AVR ONE! device that offers a wide range of programming modes.

ATMEL’S AVR32UC3A series is considered lowest power 32-bit Flash MCU with Ethernet and USB On-The-Go[11]. Equipped with 512Kbytes Flash memory with embedded 10/100 Ethernet and full-speed USB 2.0 it makes interesting offer not only for this project but also for future improvements or extensions. Power consumption of this unit ranges from 40 mA at 3.3V, while it operates delivering 80 DMIPS at 66MHz, to 40 micro-Amps, while the device is in the standby mode. In addition it consumes 1.65 mW/DMIPS what outperforms other architectures offering comparable set of features and lower processing performance. Other interesting aspect of this MCU is that its core, system bus matrix, memory subsystem and peripherals were designed from the ground up, as a whole, to ensure optimum performance. Flash memory of the AVR32UC3A series uses pipelined, dual- bank architecture that outputs one word every clock cycle while executing sequential code, with or without use of wait state. Implementing the wait state enabled raise of the clock frequency from 40 MHz to 66 MHz. Another advantage of the technology that has high influence on its performance is a six layer bus architecture with dynamic frequency scaling. It enables masters to connect to the slave peripherals at the concurrent basis with speed of 264 Mbytes per second at 66 MHz. In case more masters attempt to connect one slave arbitration process is automatically started.

This MCUs reliability is secured by on-chip system manager that includes several features: internal voltage regulator for 3.3 V single power supply operation, power-on reset, brown-out detector, hardware watchdog timer and a real-time timer. Timing is controlled via on-chip RC oscillator, 2 high frequency external oscillators, one 32 kHz oscillator and two independent on-chip PLLs. Additional security may be provided to flash memory to stop the content from being corrupted or read via unauthorized access. Also important detail on this MCU is fact that is well supported. It was released with a framework enabling fast setting and operation as well as quick setup. It may be programmed with C or C++ language, what makes it available to almost any programmer.

ADC included into AVR32UC3 unit is based on Successive Approximation Register (SAR) 10-bit Analog-to-Digital Converter (ADC). Part of that ADC is also 8-to-1 analog multiplexer that enables conversion from analog to digital signal at 8 separate channels, what provides wide functionality.

The ADC may be used in either 8-bit or 10-bit resolution mode. Results of the conversion are stored in the register common for all available channels or there might be introduced separate register for each of them. Triggers are used to start the conversion and the ADC supports following types of them: software, external or time dependent. All of the above are configurable. Timer/Counter module enables proper synchronization of time dependent operations. TC implemented in the AVR32UC3A family of microcontrollers consists of 3 identical Timer Counter channels that work with 16-bit resolution. All of the channels work independently. Channels are capable of generating processor interrupts if their internal interrupt signals are properly set. That feature was used in the project for several purposes like counting the led-on time, sampling intervals and more.

AVR ONE! device was provided for the purpose of this project. It is a powerful development tool dedicated for on-chip debugging and programming of all Atmel AVR, AVR32 and AVR XMEGA family products. Usually, while debugging microcontroller application, an emulator is used. Such approach only imitates the behavior of the target device, the closer the emulators behavior is to the original device, the greater is the chance that the same application will work on the target controller unit. AVR ONE! enables user to monitor the execution of the program on the target, instead emulating it. It interfaces with internal On-Chip Debug system included in the target AVR device where it is capable of monitoring and controlling of programs flow. It means that when debugging with a use of AVR ONE!, actual application is programmed and executed on the target device. This approach provides possibility to see the program execution in environment while maintaining all the electrical and timing characteristics in the system, what couldn’t be realized with use of traditional emulator. Device on-chip debugging may operate in following modes: run, stopped, hardware breakpoints and software breakpoints.

One of the most important elements of the development, was a flexible programming integrated development environment (IDE), named AVR Studio 32, and a software framework AT32UC3A [2, 3]. IDE is based on popular Eclipse project, enabling to use third party plug-ins. In order to investigate proper memory allocation registers view is added. This IDE supports programming in both C and C++. This environment supports all Atmel microprocessors from the AVR32 family, provides functionality for development of the standalone applications – not requiring any operating system – as well as Linux applications. Other important features are GNU Toolkit that includes GNU C Compiler, and GNU Debugger. This framework is dedicated for the specific microcontroller unit. It covers both assembler and C files with definitions of useful functions [11]. All of the functions are performance optimized, what makes the application based on them operate relatively fast [8]. Framework is divided into modules providing different possibilities. This modules are completely functional sets of functions that contain source code, example of usage, HTML documentation and ready to use template project for IAR EWAVR32 and the GNU GCC compilers. Figure 1 presents all of the modules and its relationships. Each of the modules provides certain functionality: drivers, software services, hardware components, C/C++ utilities, demo apps.

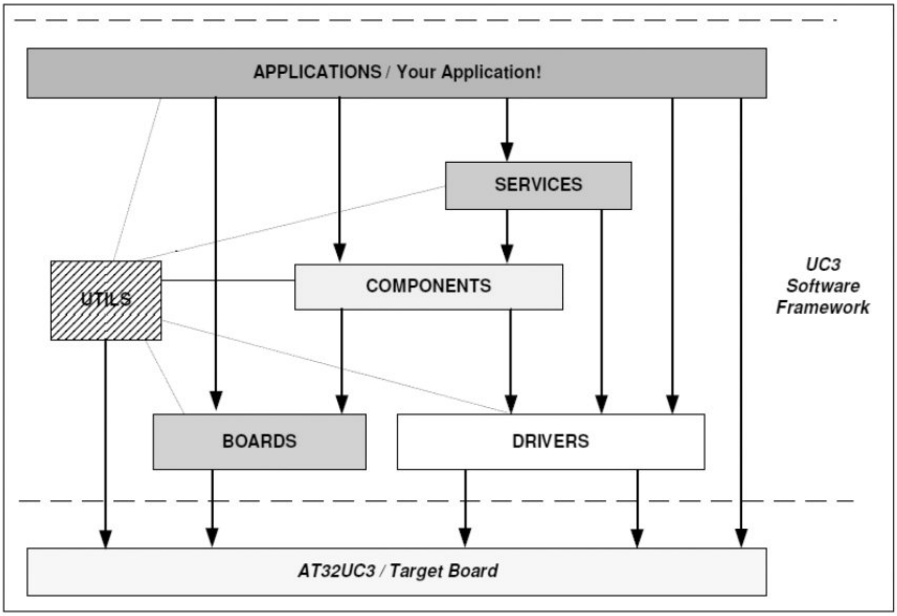

Fig. 1. Block diagram of AVR Software Framework

2.4. Initial requirements

In order to realize the occurrence of the emergency vehicle there was a need to distinguish its appearance in the surrounding. Further step had to involve notification of the drive to help him spot it. Finally it was required to help the driver create environment in which he could easily focus on the situation. This general idea is presented on Figure 2.

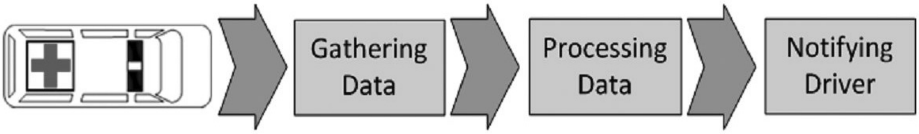

Fig. 2. Block diagram of safety device design

There are two possible ways to approach this problem. One is to detect flashing light signals, the other is to detect siren audio signal. Sound detection is relatively easier, because it does not require visual contact with the target of detection. Audio processing has another advantage, it is less resource consuming method, therefore, should be faster operating, if properly implemented. This idea is presented on Figure 3.

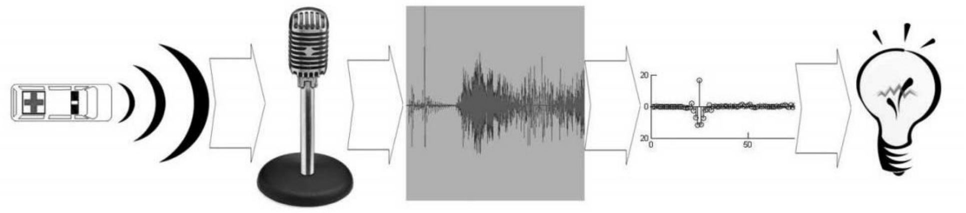

Fig. 3. Safety device prototype operation flow based on audio signal processing

The way of driver notification (in order to draw the attention to incoming emergency car) may be realized in different ways, e.g. producing a sound in the cabin, flashing a light, turning down the cars stereo, etc. For the purpose of the prototype, a blinking diode was chosen to indicate the detection of siren audio pattern. Matlab was chosen to make initial manipulations before final implementation, and then C programming was implemented [9, 16].

3. Implementation

The interesting point of the research was that no standardized rules according to tone of the emergency vehicle are established. This could have fatal consequences for this project, however it is strictly forbidden for the civilian vehicle to be equipped with devices producing either blue or red light signals or acoustic installations producing variable tone sound signals. With the respect to that it may be assumed that there are frequencies and their patterns typical for the emergency vehicles. Therefore, experimental data collection provided necessary signals for further investigation. A number of samples was recorded on the streets in the urban areas, in order to process the data in Matlab environment initially [13]. Figure 4 presents a set of different (both clear, as well, as in heavy traffic conditions) samples collected.

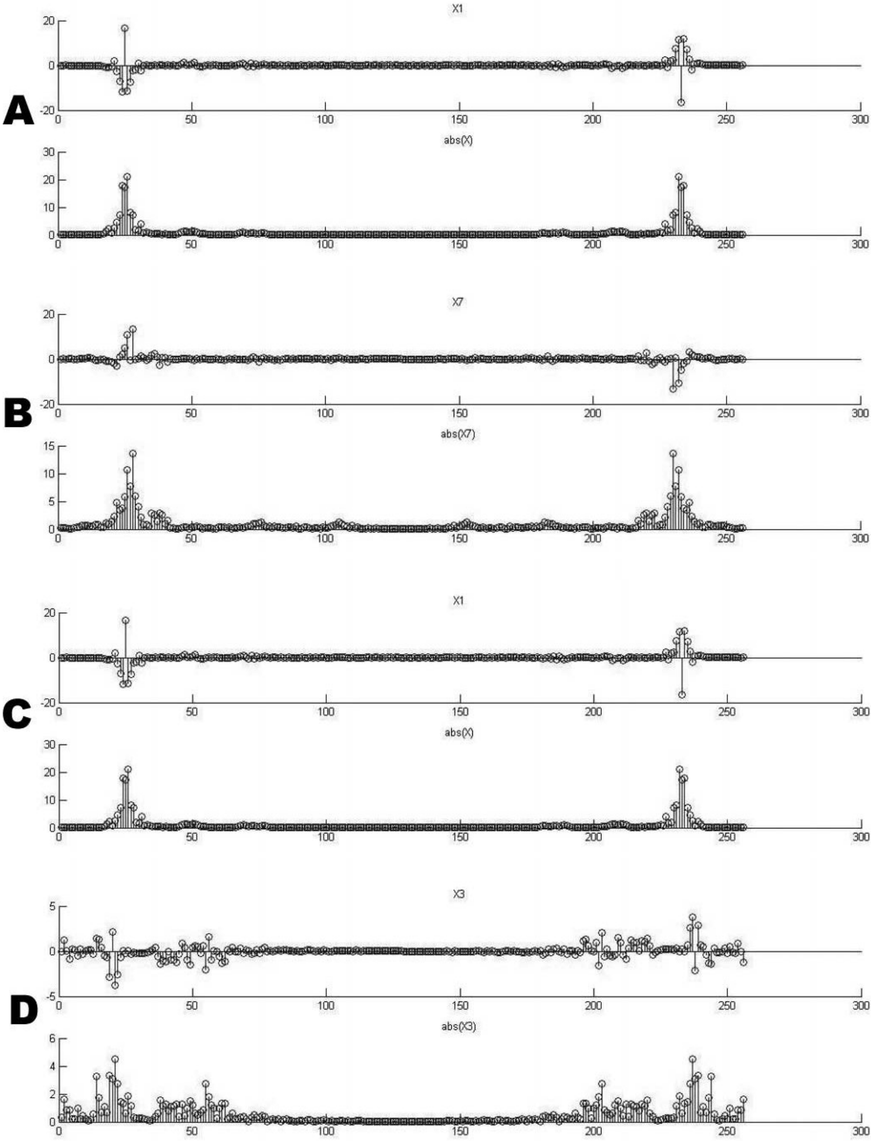

Fig. 4. Four samples of audio signals coming from emergency sirens recorded in the urban area. Each pair (A, B, C, D) presents FFT and its absolute value

First idea involved the operations on the signal within its frequency domain. Approach assumed collection of two frequencies of the ambulance and check the ratio between them. Idea was interesting, however it was later realized that emergency siren produces wide range of sounds at varying frequencies. Moreover this idea would not be applicable while these frequencies are not standardized. Last disadvantage of such approach is that it will become highly complicated while introducing frequency shifting connected with the Doppler Effect.

Another approach was also based on the frequency domain of the given signal, however the detection was meant to be based on the comparison of the recorded data with prepared template of the signal. The drawback is that template would be stored in the flash memory, what would dramatically reduce the system speed. It would require many comparisons, and what is most important such recording should take longer in order to obtain precise sample and in consequence work with high accuracy. Other disadvantage is due to the lack of standardization of such sirens. Last idea involved comparison with the values calculated on the basis of the signal. More precisely algorithm relies on several steps. Simplified block diagram of that solution is presented on the Figure 5. Flow of this program is as follows. At first data is gathered with a microphone. Due to the choice of FFT algorithm based on the radix-2 solution, this algorithm works only with the number of samples which is power of two. And in order to obtain results that are reliable and get them fast it was decided to use 256 samples at sampling frequency of 32 kHz. Sampling frequency was picked that high as the microphone used in the project was able to record frequencies in range from 20 Hz to 16 kHz.

After the samples were collected it was required to get the frequency domain description of the signal recorded. To obtain this form decimation-in-time radix-2 FFT algorithm was used, as it was mentioned before this algorithm takes a number of samples equal to the power of 2. As the result of that calculation we get a frequency domain of the input signal. Frequency domain is described as the 256 values representing the frequencies. Taking into account that range of frequencies is 0 to 32 kHz it appears that 1 step in the frequency domain describes 125 Hz interval. Knowing that DFT is symmetrical, further calculations were applied only to the first half of the range – 0 to 16 kHz. Having designated the range of frequencies characteristic for the emergency vehicle siren detection was done with respect to this interval. For each data set the highest frequency was picked and its index stored in separate element containing only the indexes of the loudest frequency values of the recorded intervals. Algorithm passed to next procedure if all of the amplitudes were over certain volume level and referred to the specific range or else next loudest frequency index was to be stored in the first position of that object. After 10 consecutive highest intervals are recognized next step of processing starts. It has to be recalled that there are two sound patterns mainly used, first with frequency slowly rising and falling and the other in which frequencies are exchanging dynamically. In order to verify what kind of signal appeared following operations are done. At first the peak and the bottom frequencies are looked up from the element containing the loudest frequency values. Next action involves calculation of the average for that object. Having these values iteration over the loudest frequency element is performed. While iterating two comparisons are done simultaneously. One compares the current iterated value to the average and if the values differ only by half of the unit, the counter is increased and assuming that the iteration ends and the counter represents 8 sufficient comparisons out of ten then there exists high possibility of occurrence of the long emergency signal.

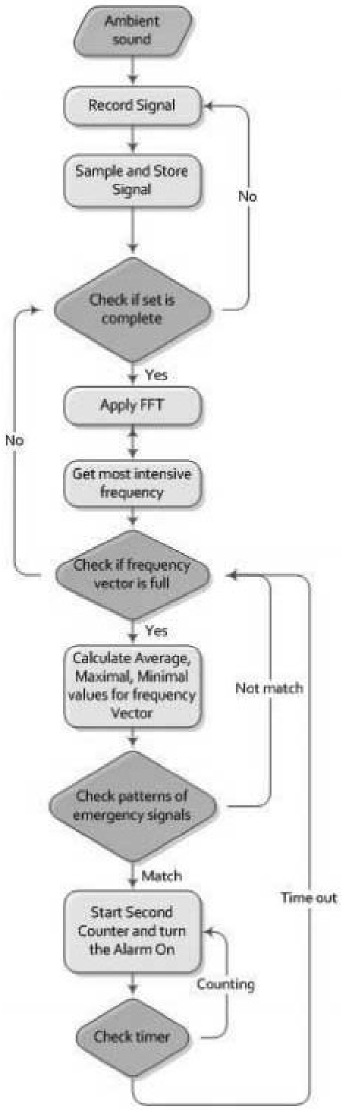
Ambient sound
Record Signal
Sample and Store Signal
No
Check if set is complete
Yes
Apply FFT
No
Get most intensive frequency
Check if frequency vector is full
Yes
Calculate Average, Maximal, Minimal values for frequency Vector
Not match
Check patterns of emergency signals
Match
Time out
Start Second Counter and turn the Alarm On
Counting
Check timer

Fig. 5. Implementation of signal processing algorithm flow diagram

The prototype solution implemented in Atmel's AVR evaluation kit was tested extensively and proved its efficiency even for high level of traffic noise. The indication of successful detection of siren audio pattern was realized by blinking led diode, triggered by the device's output signal. The implementation and programming was realized within the Master's Thesis of Zbigniew Fratczak [14]. Tests proved that the approach is correct and works within urban areas. This revealed to work better in open areas, and in heavy traffic efficiency falls, however it still performs with satisfactory results. Although this device works, testing has shown its weakness. Several changes may improve its performance and broaden the functionality. First of them would be introduction of filters to the device, bounding analysis to the frequencies characteristic for the emergency vehicles, which would result in more efficient use of computing power of the microcontroller unit. Additional amplifier to magnify the microphone signal amplitude would be also helpful. Probably, for end user, not a diode, but employment of the car's audio system would be more efficient in terms of drawing the drivers' attention.

4. Conclusions

Paper covers the results of the successive implementation of prototype device enabled to automatically detect emergency vehicles in traffic conditions with use of audio signal processing. The prototype is fully functional device, which records and recognizes the signals of the right-of-way vehicles in the urban areas. The system utilizes relatively powerful and energy efficient microcontroller AT32UC3A. Processing unit is coupled with a microphone, analog-to-digital converter and signaling diode.

The device was tested in live conditions, however the distance from which the emergency vehicle approaches remains variable, thus device sensitivity depends on the surrounding conditions and actual traffic. Device works more efficient for lower background noise level.

References

[1] AT32UC3A Series Preliminary. (2009), http://www.atmel.com/dyn/resources/prod_documents/doc32058.pdf, retrieved 2010.

[2] AVR32 Studio getting started. (2008), http://www.atmel.com/dyn/resources/prod_documents/doc7732.pdf retrieved 2009.

[3] AVR32 UC3 Software Framework. (2007), http://www.atmel.com/dyn/resources/prod_documents/doc7732.pdf, retrieved 2009.

[4] AVR32UC Technical Reference Manual. (2007): http://www.atmel.com/dyn/resources/prod_documents/doc32002.pdf, retrieved 2010.

[5] Brill W.E. (2002). Patent No. 6,362,749 B1. US.

[6] Decimation-in-time (DIT) Radix-2 FFT. (2006), http://cnx.org/content/m12016/latest/ - retrieved 2010.

[7] EVK1100 Schematics. (2008), http://www.atmel.com/dyn/resources/prod_documents/EVK1100_SCHEMATICS_REVC.pdf - retrieved 2010.

[8] Gadre D.V., *Programming and customizing the AVR microcontroller*. McGraw-Hill, New York, London, 2001.

[9] Grigoryan A.M., *Brief notes in advanced DSP: Fourier analysis with MATLAB*. CRC Press, Boca Raton, 2009.

[10] Gross Jon, G.D. (2003). Patent No. 2,462,103. CA.

[11] Hanly J.R., *C program design for engineers*. Addison-Wesley Longman, Reading, 2001.

[12] Helliker Conrad Dean T.E. (1973). Patent No. 3,735,342. US.

[13] Ingle V.K., *Digital signal processing using MATLAB*. Nelson, Toronto, Ontario, 2007.

[14] Frątczak Z., *System for detection of the emergency vehicles in urban areas.* Praca Magisterska, Katedra Informatyki Stosowanej Politechniki Łódzkiej, Promotor: dr inż. Andrzej Romanowski, Opiekun dodatkowy: mgr inż. Jakub Betiuk. Politechnika Łódzka, 2009, 78s.

[15] King F.N. (1982). Patent No. 4,321,589. US.

[16] Kochan S.G., *Programming in ANSI C*. Hayden Books, Indianapolis, 1988.

[17] McConnell P.R. (1998). Patent No. 5,710,555. US.

[18] Ratmed.pl-Ratownictwo Medyczne. (n.d.). Retrieved January 2009, from Forum: http://www.ratmed.pl/forum.

[19] Tan L., *Digital signal processing : fundamentals and applications*. Academic Press, Amsterdam, Boston, 2008.