Intelligent Traffic Alert System for Smart Cities

Assil Ksiksi, Saeed Al Shehhi, Rashad Ramzan
Department of Electrical Engineering
UAE University
Al Ain, United Arab Emirates

***Abstract*—Traffic accidents are one of the main causes of sudden death worldwide. Inadequate driver response to changing road and traffic conditions leads to a higher probability of road accidents. With the rise of the concept of Smart Cities, safety and security are two important issues that need to be addressed. In this paper, we propose an Intelligent Traffic Alert System (iTAS) that warns drivers of potential dangers on the road using audio and visual alerts. The iTAS consists of transmitter units installed on the side of the road that broadcast vital information such as the speed limit, road conditions, and unexpected traffic situations to drivers in real-time over the commercial FM radio frequency band. On the receiver side, the system uses the radio already present in the vehicle to deliver audio alerts that help keep drivers informed on road and traffic conditions. The installation of an optional receiver unit inside the vehicle makes it possible to display the alerts on a LCD screen in the form of visual warning symbols. A lab prototype of the proposed system was built using off-the-shelf commercial components. The results acquired from testing the prototype were good enough to validate our approach to the design.**

Keywords*—*smart cities; traffic alerts; traffic alert system; safe driving assistance; intelligent road system; road safety

I. Introduction

Road safety is a worldwide issue in general and in the United Arab Emirates (UAE) in particular. The UAE for example has a fatality rate of 12.7 per 100,000 people [1], a number that is much higher than that of developed countries such as the UK, Sweden and Japan. There are many different approaches that can help reduce the number of accidents in any given country. A few popular ones are more aggressive road safety campaigns, education of future drivers in topics related to safety on the road, and implementation of stricter traffic laws.

The use of technology to improve quality of life and safety is one major goal of future smart cities. Quite a few systems have been proposed in the literature that aim to improve safety on the road. In [2] and [3], a system is discussed that involves sending alerts to vehicles by placing passive RFID tags on the side of the road before points of interest, such as sharp turns, construction, pedestrian crossings etc. Physical access is required to update the data on a particular RFID tag. Therefore, this is not a scalable system which can meet the demand of a large city. In [4], the use of Vehicular Ad-Hoc Networks (VANETs) and a centralized Traffic Information Center (TIC) is proposed. In [5], a system is outlined that involves the use of a completely independent VANET where the database is stored separately in each vehicle and all computation is performed locally to remove the dependency on an external TIC.

The major downside to VANETs is that powerful hardware needs to be installed in each vehicle in the network. This results in an expensive system, thus inhibiting widespread deployment. Furthermore, automotive companies need to approve of any new system before adopting it in their vehicles, which means that further standardization and security audits must be performed first.

In this paper, we present the Intelligent Traffic Alert System (iTAS). The system consists of FM transmitter units deployed citywide that broadcast relevant information to nearby vehicles over the commercial FM band. The transmitters broadcast a message consisting of both audio and digital versions of road and traffic information. Drivers simply tune the car's radio to the desired frequency to get the latest alerts and updates in the form of short audio messages. By installing a receiver unit inside the vehicle, drivers can view alerts and further information on a LCD display. The information stored in each road side transmitter can be updated in real-time over GSM by a TIC, making the system scalable to large areas.

The use of commercial FM and the existing FM radio in the car as a receiver makes the system low cost and free of any needed standardization and legal approval. Unlike the systems reported in [4] and [5], there is no signal transmitted from within the car. As a result, the identity, privacy, and security of the vehicle are preserved.

The paper's structure is as follows. In Section II, a high-level outline of the system design is discussed. Section III provides details on the construction of the lab prototype. Section IV discusses the software written for use in the hardware prototype. Section V lists future directions for the research, and Section VI concludes the paper.

II. System Design

The primary goal of the iTAS is to keep drivers informed of changing road and traffic conditions. The system consists of a transmitter unit (TU) and a receiver unit (RU). As seen in Fig. 1, TUs will be installed at fixed intervals along roads in the city, while RUs will be located inside vehicles passing by. Each TU will continuously broadcast information to RUs within range using FM as a simplex communication channel.

The commercial FM frequency (88-108 MHz) band has a few distinct advantages. The FM band can be received by all vehicles with no additional modification, has a relatively good transmission range compared to GHz frequencies, and is not adversely affected by changes in weather and air channel conditions.

A. *Transmitter Unit (TU)*

The TU's task is to broadcast a signal containing both audio and digital versions of pertinent traffic information over commercial FM. It will need a power source, preferably one that is renewable and cheap. Additionally, the information stored on

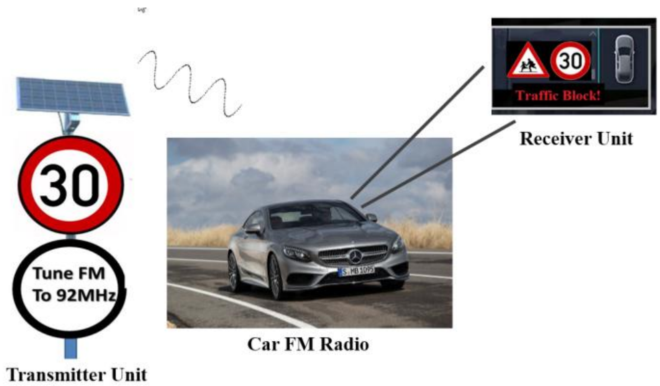

Fig 1. The system overview of iTAS. The TU broadcasts alerts at some frequency (92 MHz in the prototype), and the car radio and optional RU deliver the alerts to the driver.

any single TU must be capable of being updated remotely via GSM. Based on these requirements, our proposed transmitter unit is shown in Fig. 2. The TU consists of several modules that work in concert to broadcast information to vehicles in range.

The heart of the transmitter unit is the microcontroller module. It has two main responsibilities. Firstly, it retrieves stored traffic and road information from the on-board non-volatile flash memory. Secondly, the firmware loaded onto it builds audio and digital representations of the information, merges the two formats, and finally passes them as an audio signal to the FM chip. This process is repeated every *t* seconds to form a continuous broadcast. In addition, any incoming updates to the information (traffic jam, accidents, etc.) will be written to memory for the next iteration of the broadcast.

A GSM module will allow the transmitter to receive updated traffic and road information for alerts over-the-air. The module will then forward the newly received traffic data to the microcontroller, which will first extract relevant information, then write it to the flash memory.

The radio module consists of a FM radio chip connected to a dipole antenna. The chip is interfaced with the microcontroller. When the FM chip receives a signal, it will modulate it using FM and forward it to the antenna for transmission. To keep things simple, the system will transmit using mono FM.

Finally, to power the transmitter, we propose the use of a solar panel and voltage regulator interfaced with a Li-ion battery. The power consumption of the TU depends mainly on the GSM module and FM radio chip. A typical FM transmitter IC, such as the NTE7200, consumes around 4.5 mW [6], compared to at least 30 mW for a RF transceiver IC. As a result, if the design is optimized, the capacity of the battery and the solar panel rating can be quite low.

B. Receiver Unit (RU)

The receiver unit provides two options for the user to consume road and traffic information: audio only, or both audio and digital. The first option is taken care of entirely by the car's built-in FM radio, making it easy for any driver to simply tune in. A block diagram of the unit is shown in Fig. 3.

Receiving further information in digital form requires forwarding the radio's output to custom hardware, which consists of a Arduino microcontroller board interfaced with a LCD display. The audio is sampled by the firmware running on the Arduino and, if any embedded data is detected, the firmware will use Goertzel's algorithm to recover the data embedded in the received audio message. Finally, the retrieved data is displayed on the LCD screen. The hardware should ideally be housed in a clean and portable enclosure. The unit should also be easy to place within the vehicle for convenience. As far as power goes, both the Arduino and LCD display can be easily powered by 2 or 3 AA batteries.

III. Prototype Overview

We built a prototype of the iTAS to test the core idea and operate the system in a realistic setting. As seen in Fig. 4, the prototype consists of a basic transmitter unit attached to the rear of a speed limit sign replica, and a receiver unit. The system was built using off-the-shelf components to reduce implementation time and overall cost.

The power module consists of three 6V solar panels connected in series and two 6V lead-acid batteries. The panels and batteries are connected to a battery charger unit to provide backup in case of poor sunlight.

The prototype lacks some of the features outlined in the previous section. Instead of a microcontroller that generates the audio independently, we used a off-the-shelf car FM transmitter with a built-in SD card slot. Every time we needed to update the information for testing, we wrote a new MP3 file to the SD card. The MP3 file was generated using custom MATLAB code. We did not include a GSM module in the prototype.

The receiver unit is similar in structure to the one proposed in the previous section. It consists of a portable FM radio connected to a Arduino microcontroller that is used for extraction of the embedded data. A LCD display interfaced with the Arduino allowed us to display small icons depending on the information retrieved from the digital data. We wrote the firmware for the microcontroller in standard AVR C.

In the next section, we will provide an overview of how the MP3 was generated using MATLAB, as well as an outline of how the Arduino firmware we wrote works.

IV. Software Implementation

There are two parts to the software side of the prototype:

1. Generation of the MP3 file containing both audio and embedded visual alerts.
2. Retrieval of the embedded data frame from the incoming FM audio broadcast.

The first part was implemented in MATLAB, while the second part was implemented in AVR C. The source code for both parts is available on Github under the MIT License [7].

The transmitted data frame structure is shown in Fig. 5. It is 32 bits long, including the start and stop sequences to simplify recovery at the receiver - 1110 and 0101, respectively. We kept plenty of empty values for future modification.

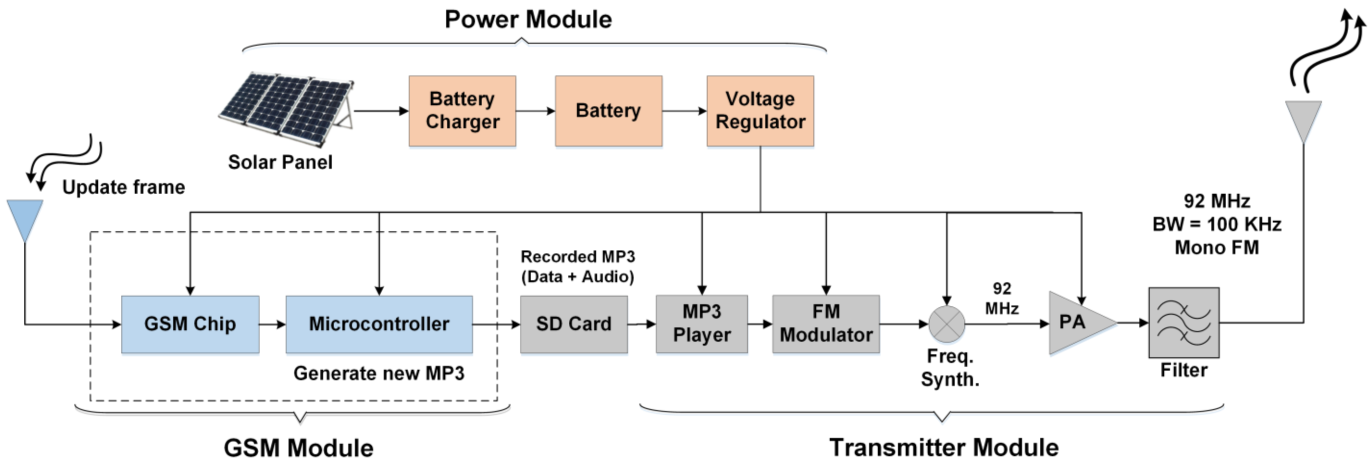

Fig 2. Transmitter unit of iTAS. In the proposed design, the SD card is replaced by flash memory, the MP3 player is not used, and the battery voltage is 5V. The prototype on the other hand lacks a GSM module, and is supplied by a 12V battery, which is why the voltage regulator is included.

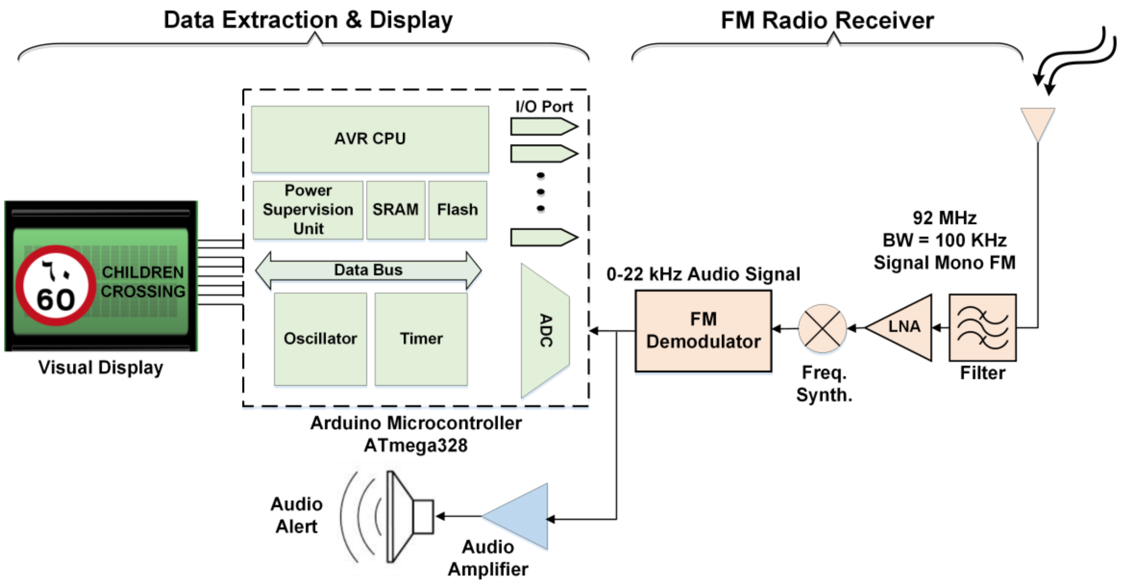

Fig 3. iTAS receiver unit. Audio stream from the radio is forwarded to the microcontroller for further extraction.

A. MP3 File Generation

First, a sample voice recording of a speed limit alert ("The speed limit is 60 km/h") is created with some fixed duration at a specified sampling frequency – we chose a duration of 2 seconds and a sampling frequency of 44 kHz. This was selected to ensure that the embedded data signal is inaudible when played on the car radio.

Next, the data frame is modulated using FSK at just under half the sampling frequency. More specifically, we chose a mark frequency of 21 kHz to represent binary 1, and a space frequency of 20 kHz to represent binary 0, as shown in Fig. 6. The data signal is repeated to fill up the entire duration of audio to ensure faster delivery to the receiver. The audio and data signals are added together to provide the final alert signal, as shown in Fig. 7. Finally, the raw audio is encoded as an MP3 using FFmpeg [8], and then copied onto the SD card for use by the FM transmitter.

B. Data Frame Retrieval

The receiver code is more complex. Since the transmitter is always broadcasting, the receiver does not know when the transmission begins. In addition, it does not know at what time the frame actually starts in the audio signal.

To solve the first issue, the microcontroller's analog-to-digital converter (ADC) is continuously sampled until a voltage greater than some threshold determined by the radio is sensed (in the lab, this was around 0.2V). Once this happens, the firmware enters the main loop. The ADC is further sampled for two entire frame lengths to guarantee at least one frame is captured. This addresses the second issue mentioned above.

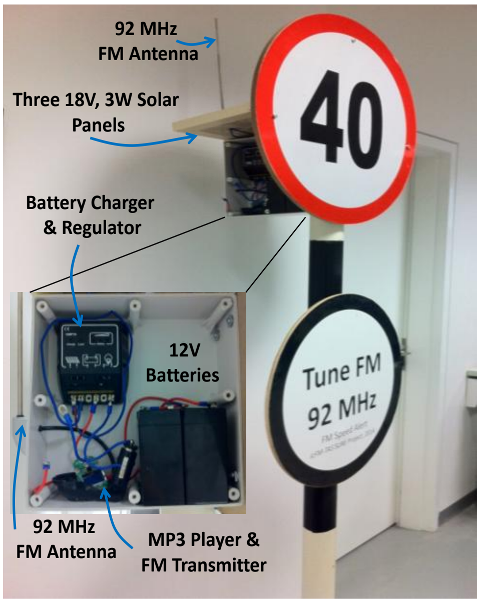

Fig 4. iTAS hardware prototype built in the lab for testing.

Truck Sign [1]	Max Speed [4]	Min Speed [3]	Warning Sign [6]	Traffic [2]	CRC [8]

Fig 5. Frame structure with field sizes in bits. Start and stop sequences are not shown.

Unfortunately, the ADC interrupt timings provided by the ATmega328P proved to be inaccurate for our choice of sampling frequency. We tried modifying the sampling rate of the generated MP3 file to accommodate for this, but it could not be decreased due to limitations in the MP3 specification. Instead, the ADC pin was manually sampled using tight loops. Each loop was optimized based on the clock cycles required for each instruction executed within it (jumps, FP multiplications, addition, etc.). The timing for the instructions was acquired from running the code in Atmel's simulator for the ATmega328P.

The actual recovery of the embedded frame bits is done using Goertzel's algorithm, a modified form of the DFT with extremely low memory and computation requirements. The idea is to compute the power of the mark frequency in a time slice of the signal equivalent to 1 bit. The power computation is performed using the current sample along with the computed values of the last two samples.

At the end of the time slice, if the total power is above a pre-computed threshold, the bit is read as a 1, and a 0 otherwise. The bit is finally stored in a bit array. The storage required for the algorithm depends on the two samples in floating-point format and the array of bits of length twice the frame size (64 bits in our case).

Once the algorithm terminates, the start and stop sequences are used to find the data frame. If a frame is found, the CRC code is checked. In case of a mismatch, the entire process is repeated. In our tests, one run of the algorithm takes approximately *t* ms, which for our case can be found using the following relation:

$$t = 2\,frames \times 32\frac{bits}{frame} \times 5\frac{ms}{bit} = 320\,ms$$

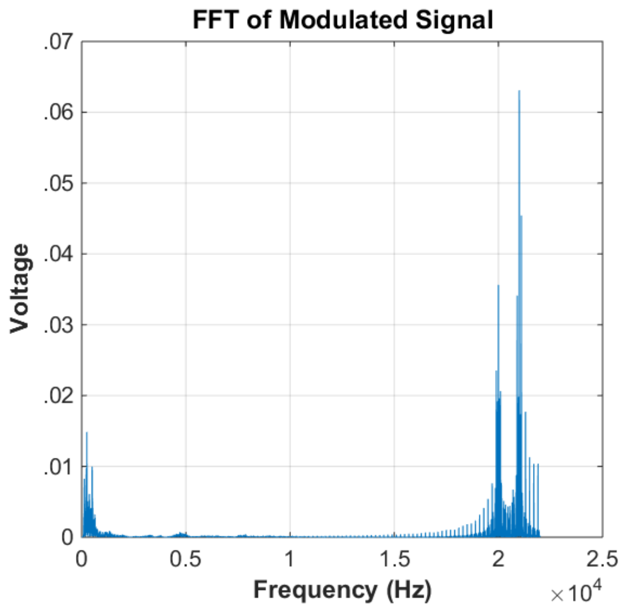

Fig 6. Frequency spectrum of the modulated signal. The audio can be seen at lower frequencies, while the embedded data is in the 20-21 kHz range.

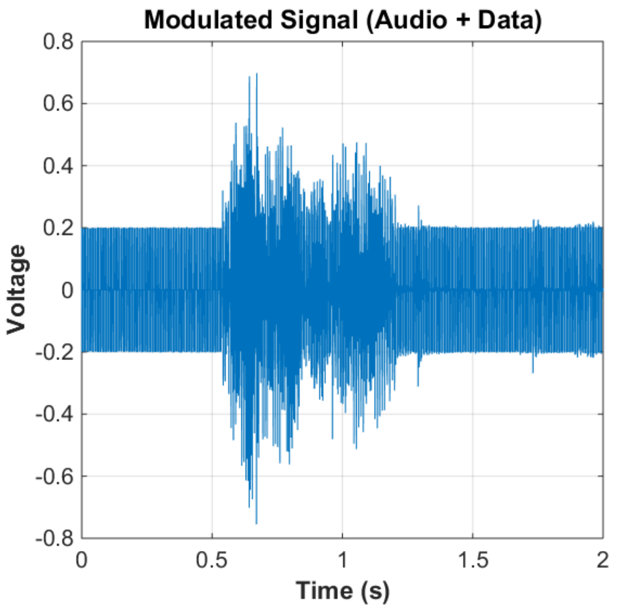

Fig 7. Modulated signal consisting of the linear sum of the audio message and FSK modulated data for embedding.

Given the approximate range of transmission of the prototype's TU (around 500 m), the value of *t* computed above, and the maximum expected vehicle speed, we computed *n*, the maximum number of attempts the receiver has to recover the frame from the broadcast:

$$n = \frac{0.5}{160} \times \frac{3600}{0.32} = 35.156 \approx 35$$

The receiver therefore has approximately 35 attempts to retrieve the frame in the worst case. At lower speeds, the number of attempts available will increase.

An additional piece of code was written to decode the contents of the frame (speed limit, traffic conditions, etc.) and display the information on the LCD screen. A sample of a generated speed limit symbol can be seen in Fig. 8.

V. Results and Future Work

The prototype and the software detailed above allowed us gain further insight into whether or not the system is feasible in real working conditions. We identified two major issues that require improvement before the system can be commercialized.

Firstly, the microcontroller used, the ATmega328P, was not powerful enough to receive transmissions at a high sampling rate. Decreasing the sampling rate proved ineffective, and led to significant audible noise caused by the embedded data frame. The obvious solution is to use a more capable processing unit such as a 32-bit ARM microprocessor. Secondly, installing transmitter units across a large city and maintaining them over a long period of time is a very difficult and costly job. In countries with hot climates, such as the UAE, the transmitter unit must be built using industrial grade components to avoid overheating and malfunctions.

We are working on a new system that addresses these two major issues, as well as other smaller issues we encountered while building and testing the prototype. The new system consists of a centralized server that houses traffic and road data for the entire city. An app installed on the driver's smartphone periodically pings the server over 3G/LTE/WiMAX to receive both visual and audio alerts on traffic conditions in near real-time.

VI. Conclusion

In this paper, we outlined the design of the Intelligent Traffic Alert System (iTAS), a smart traffic alert system that aims to help keep drivers updated on road and traffic conditions. The system consists of two main components: a transmitter unit and a receiver unit. Transmitters are installed on the side of the road and broadcast information relevant to their location over commercial FM. Drivers can tune in to the specified broadcast frequency and listen to audio alerts without installing or purchasing any additional hardware. The receiver unit is optional, but can provide more visual information and alerts extracted from the broadcast if installed. The data stored by each transmitter can be updated remotely over GSM.

We built a functional prototype of the system to test it and determine if it's feasible. In the future, we plan on adding GSM functionality to it, and replacing the microcontroller used to one that is significantly more powerful to avoid the issue encountered at the receiver end. We are also working on a new system that is much easier to deploy at scale. It consists of a centralized traffic information server and an app installed on the driver's smartphone.

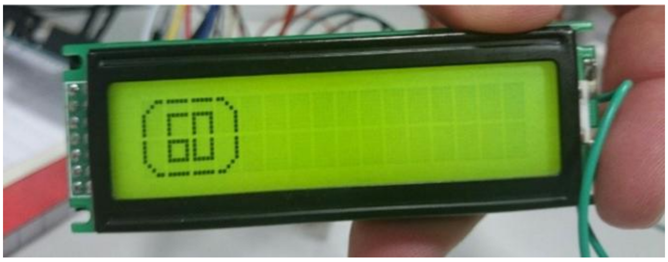

Fig 8. Arduino programmed to show a speed limit of 60 on the LCD display.

References

[1] A. S. Mohammed, "Leading cause of road traffic morbidity and mortality in the United Arab Emirates (UAE) and the main adaptations to reduce it," in *Innovation Arabia 2015*, p. 90.

[2] H. Zoghi, M. Tolouei, K. Siamardi, and P. Araghi, "Usage of ITS in the in-vehicle signing system with RFID tags and vehicle routing and road traffic simulation," in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2008, pp. 408–413.

[3] A. Paul, R. Jagriti, N. Bharadwaj, S. Sameera, and A. S. Bhat, "An RFID based in-vehicle alert system for road oddities," in *2011 IEEE Recent Advances in Intelligent Computational Systems, RAICS 2011*, 2011, pp. 19–24.

[4] G. S. Khekare and A. V. Sakhare, "A smart city framework for intelligent traffic system using VANET," in *2013 International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s)*, 2013, pp. 302–305.

[5] L. Wischhof, A. Ebner, H. Rohling, M. Lott, and R. Halfmann, "Self-organizing Traffic Information System," in *Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual*, 2003, vol. 4, pp. 2442 – 2446.

[6] "NTE7200 IC FM Stereo Transmitter." [Online]. Available: http://www.nteinc.com/specs/7200to7299/pdf/nte7200.pdf. [Accessed: 10-Sep-2015].

[7] "iTAS on Github." [Online]. Available: https://github.com/Cyph0n/iTAS.

[8] "FFmpeg." [Online]. Available: https://www.ffmpeg.org/.