

Software Based AFSK Generation on Arduino

Luis Sánchez Velasco

GranaSAT Aerospace Group
University of Granada
Granada, Spain
hfegetude@gmail.com

Andrés Roldán Aranda

Electronics Department
University of Granada
Granada, Spain
amroldan@ugr.es

Abstract—Master courses need to have a practical element, applying already known concepts to concrete problems. In this paper, the design and manufacturing of a radio link using APRS protocol over AX.25 will be presented as coursework during several lab sessions. The aim is to create a system that is able to poll data from several sensors and send them wirelessly with a software implemented signal generator in a microprocessor.

Keywords—Wireless Communication, APRS, AFSK, AX.25, SDR and Sound Modem

I. INTRODUCTION

The current availability of breakout modules and the power of microprocessors facilitate the creation of products able to read data from several sensors and transmit such data to one or several receivers. This paper tries to explore this concept by using a not so usual transmission method in the current technological scene, which is AX.25 data packages over AFSK modulation. The proposed design is a suitable candidate for laboratory work in a multi-disciplinary course where many students with different backgrounds could be working together in groups of 5-8 people with background in electrical design, signal processing or computer science, for example any Product Design course of a master in technology, for example a master in Wireless communication systems. This product could be used to test wireless satellite communication by having a reliable emitter, also this could be used as the main board in a weather balloon.

This work can be divided in three main parts, as the basic diagram in figure 1 shows. A software design group, a hardware design group and finally an Embedded Software group, each group could work independently from each other if the work is correctly shared among the participants and periodical meetings are done.

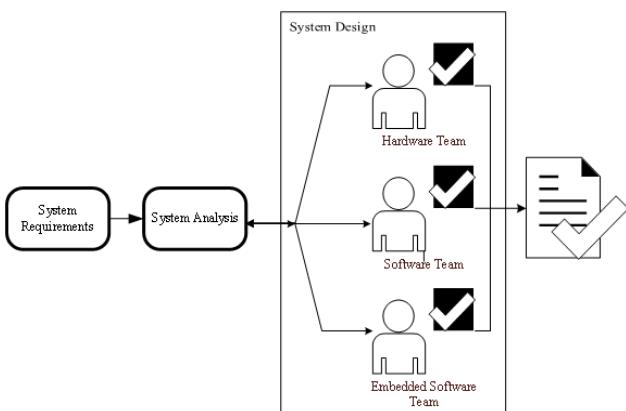


Fig. 1: Product design flow

A. State of the art.

APRS transmitter and receivers, even not so popular, are widely used for high distance transmission systems, as we can see in aprs.fi, a web page that collects data from APRS emitters and put them online, many devices use this system. These emitters are usually built around a PC and a Terminal Node Controller (TNC), a device that, if attached to a radio amplifier and modulator, will transmit the required signal. These devices are usually expensive and not as portable, issues that the design presented here will try to overcome.

II. METHODS

In order to design a successful device the standard product design flow will be executed in the following order: Design Specifications, System Analysis, System Design and testing.

A. Requirements of the Design

The design requirements of a product define the absolute minimum functionality of a device to be considered finished and functional; therefore, these requirements are meant to be met when the work is finalized. These requirements are as follows:

1. The system shall be able to transmit data over AX.25 with AFSK modulation
2. The system shall be able to read from several external sensors, these sensors are: a magnetometer, an accelerometer, a gyroscope, a barometer, a GPS and a thermometer.
3. The data generated by the sensors shall be logged onto a SD card and transmitted over AX.25
4. Images shall be taken with a camera module and placed onto the SD card.
5. A GUI capable of reading data from the USB Serial Port or an SDR and display such data.

These requirements shall be discussed by the students in each group in order to reach a final action plan and how each requirement shall be fulfilled. After such discussion a technical paper could be given to the tutor synthetizing these requirements, simulating a possible client.

B. Proposed Solution, System Analysis

During this production step, the group of students shall agree during several meetings on how to split the group in the different subsections and also, agree on the various details

that the solution may have. In this paper a possible solution will be presented.

The proposed block diagram for the system is depicted on Fig. 2. This solution consists of an Arduino Mega as the main MCU [1] due to its easy implementation and number of peripherals. All of the sensors will be read from an I2C port or a serial port. Due to the current availability at the laboratory, an OV7670 camera module will be used. This module contains a FIFO making it easier to be interfaced with. Finally, the output AFSK will be generated and outputted with PWM signal and then low-pass filtered, this solution will not require any extra digital modules since the analog signal will be software generated, therefore only an external modulator to set the frequency at 144.8 MHz will be required.

A GUI programmed with Python is desirable to be created due to its simplicity and ease of access to external modules such as the Serial Port and the audio input from an SDR.

C. System Design

This part contains the actual in depth solution to the product and its implementation. These tasks shall be distributed among groups of students depending on the background of each of them.

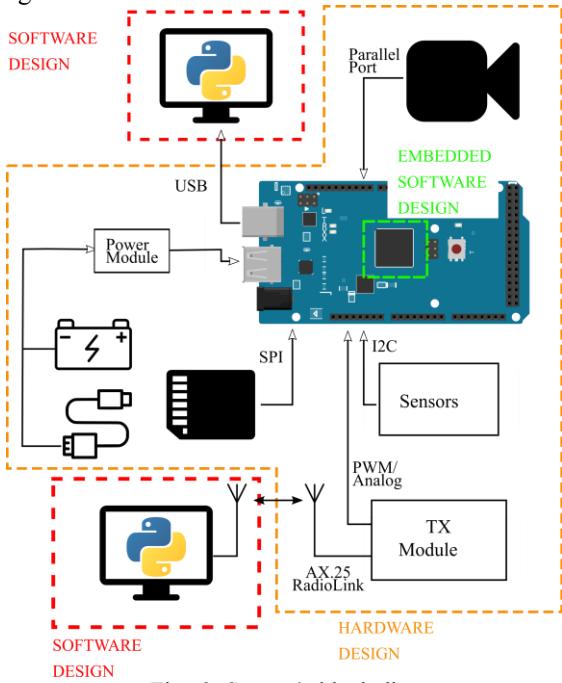


Fig. 2: System's block diagram

1) Hardware design

The hardware design team shall be in charge of selecting the right hardware components and sensors as well as creating a PCB to hold all the components together and correctly interfaced. A 3D model is also desirable to improve the presentation of the product; an example of a 3D modeled board is show below in figure 3.

Small prototypes interconnecting one of more hardware parts are desirable since it will allow the rest of the team in charge of the embedded software and software a faster development as the one depicted in figure 4.

The final selected sensors are:

- BMP180, which integrates both a temperature sensor and a barometer, this module is interfaced through I2C presenting a range of -40 to 65°C for the thermometer and 200-1100 hPa in the barometer. [2]

- LSM303 integrating both an accelerometer and a magnetometer. This system is also interfaced through I2C and the documentation found online is wide. [3]

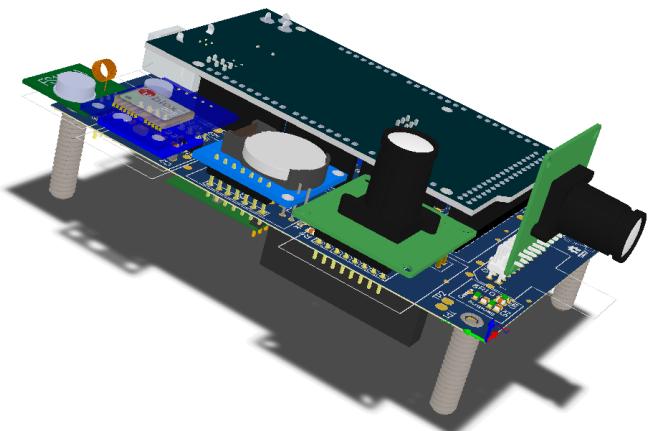


Fig. 2: 3D model of the final PCB

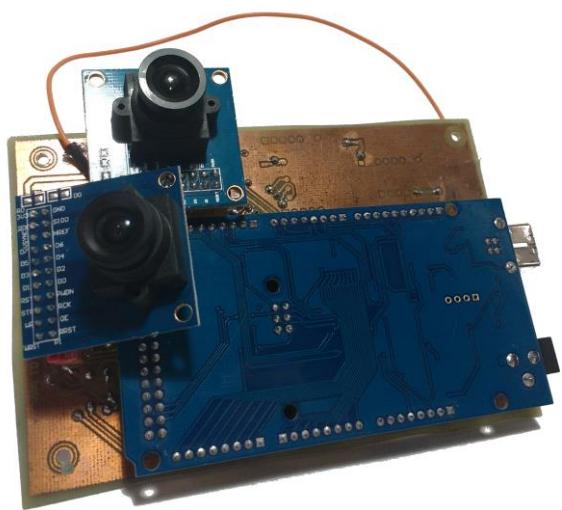


Fig. 3: Final PCB

- The Gyroscope system is integrated by an MPU6050 that also contains an accelerometer, suitable for contrasting the date from the LSM303. The full scale of this device is adjustable from 250 to 2000 °s. [4]
- As a GPS unit, the NEO-6 from u-blox is selected due to its ease of use through the serial port and automatic configuration and calibration is chosen.
- An OV7670 plus a FIFO chip is selected to take the images into the MCU. The OV7670 requires an input signal of 8MHz and will output data at the same rate. Reading from such device, though possible, will make the work way harder with no clear advantage.
- The Arduino will produce an AFSK signal that is not suitable to be transmitted due to its low frequency therefore a radio module such as the HX-11 shall be used to elevate the central frequency to 144.8 MHz, placing it in the Free / Amateur radio band.

After the items, where selected and studied a board as the one depicted in figure number 4. The board was manufactured by an external company for a small amount of money. This board includes:

1. An external battery connector with a DC-DC converter

2. Level shifters in order to interface 3.3 V logic with 5 V logic.
 3. An external connector for an antenna.
 4. Test points in order to facilitate debugging.
- 2) *Embedded Software Design*

One of the main advantages of using the Arduino platform is the wide availability of software libraries to poll data from common hardware modules with ease. All these software libraries will be used to communicate with the I2C devices as well as parse the data coming from the UART port. All of these devices will be polled periodically in order to obtain regular data from the environment. This data will then be transmitted wirelessly by the implemented AFSK transmitter. The analog signal that will drive the modulator will be entirely generated in software at baseband frequency and then elevated by the external radio modules to 144.8 MHz.

a) AX.25 frame generation[5]

The first step will be to generate a suitable AX.25 frame for transmission; this header will contain all the collected parameters with their correspondent identifier separated by a backslash. After that, the header will be generated corresponding to [5].

b) AFSK Generation

The actual implemented modulation is AFSK1200 NRZ that means that the produced tone will oscillate between 1200 and 2200 Hz sending data at 1200 bauds. In order to generate such tones, the 8-bit Timer/ Counter implemented inside the Arduino's ATmega 2560, and then the sampling rate, the samples per baud and the total lookup table size can be obtained using equations 1, 2 and 3:

$$\text{Sampling Rate} = 16 \text{ MHz} \frac{1}{256} = 62500 \frac{\text{samples}}{\text{s}} \quad (1)$$

$$\text{Samples Per Baud} = \frac{\text{Sampling Rate}}{\text{Bauds}} = 52 \quad (2)$$

$$\Delta\delta = \text{TABLE SIZE} \frac{f_m}{\text{Sampling Rate}} \quad (3)$$

Then the phase accumulator algorithm shall be followed AS described in [6]. The program will first load a bit into memory from the AX.25 frame buffer, if that bit results to be 0 then the phase delta will be toggled to the other tone. This is because a NRZ modulation is used, thus the transmitting symbol is only switched once a 0 is found and kept otherwise. Once the phase delta is obtained, the instantaneous phase can be obtained in equation 4 for each iteration:

$$\delta \leftarrow \delta + \Delta\delta \quad (4)$$

The actual sample value is obtained by a lookup table, which values were calculated using the following formula.described in equation 5.

$$\text{TABLE}[i] = 255 \sin\left(2 \pi \frac{i}{\text{TABLE SIZE}}\right) \quad (5)$$

Then the phase accumulator will continue to increment the phase by the phase delta previously selected until the number of samples per baud are met. In such case, a new sample shall be collected to restart the cycle again. This is graphically explained in figure number 5.

Regarding the actual software implementation both the outputting task and frame generation activity where done in parallel to save time and memory. All the generated samples were placed in a software ring buffer and read by an interrupt that will trigger every time a new sample has to be output.

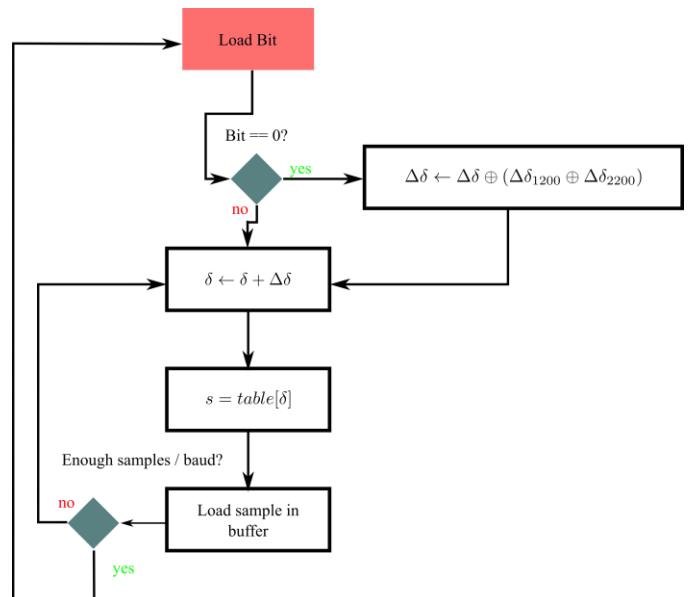


Fig. 5: AFSK Generation algorithm

This interrupt function will collect the sample and place on the comparation register of the ATmega 2560 (OCR) that will set the output pin low on a successful match. Once the counter reaches 255 and resets, the output pin is set again to high. A picture displaying this idea is shown in figure 6.

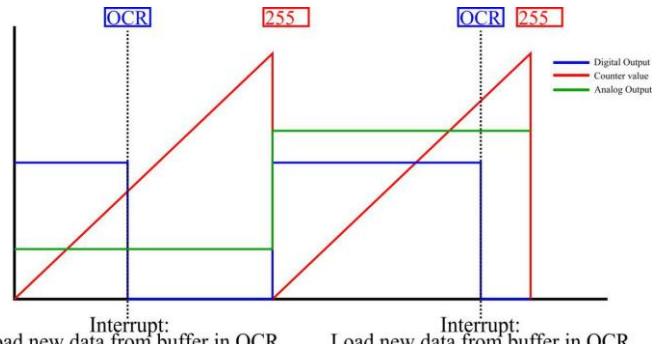


Fig. 6: PWM Signal of the Arduino

c) Interfacing with OV7670 FIFO module [7].

The connection between the MCU and the camera's FIFO is achieved with a parallel port connection. The VSYNC pin from the camera module itself is exposed, displaying a high pulse every time a new frame capture is about to begin.

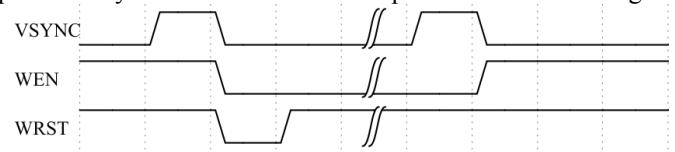


Fig. 7: Capturing a frame into FIFO

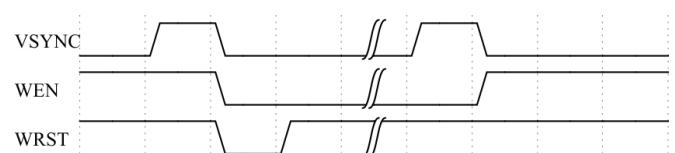
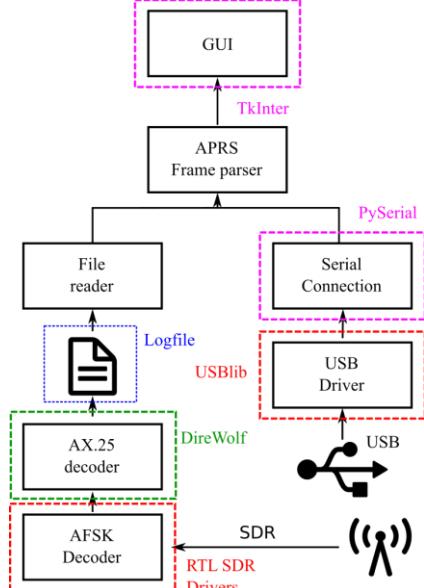


Fig. 8: Reading from the FIFO

Once this pulse occurs the Arduino must reset the write pointer of the FIFO and enable writing on it. This is achieved by setting the WEN pin low during all the transaction and

pulsing the WRST pin. The timing diagram previously described is displayed on figure 8.



To retrieve the image on to the SD card / MCU's main memory, the read pointer of the FIFO must be set to the beginning of the memory. Meanwhile pulses shall be generated into the input read clock, obtaining an image byte per pulse, as displayed in figure 6.

3) Software Design

In order to create a successful that is able to read both from the USB port, logging debug data for development and radio data from an SDR can be achieved easily thanks to the many open source projects that are available online. For this task the selected libraries/ frameworks / programs are:

- Rtl-sdr, open source drivers for the common SDR dongles that allow to stream the received signal as an audio signal.
- Direwolf, complete radio suite allowing to transmit as a TNC and receive data from an SDR, in this project it will be used as an AX.25 decoder
- PySerial, Python package that allows the interaction with a serial port programmatically.
- TkInter, framework used for creating graphical applications in python.

In figure 9, the software architecture of the project is depicted. Having that open source projects already chosen the task transform into a high level programming project where the only parts that need to be implemented by the students are the APRS frame parser, that will take the raw APRS string and transform it into numeric variables that will be displayed by the GUI made with TkInter.

In order to obtain the data from the radio module, an instance of Direwolf will be run which will load all the received APRS data onto a file, that same file will be read by the created python script and then fed into the APRS frame parser.

III. RESULTS

The final PCB was produced a manufactured by an external collaborator, the result is shown in figure 10.



Fig. 10: Final PCB produced

1) AFSK generation

The signal was successfully generated using the software designed, figures 11 and 12 show the output signal captured by a scope after being filtered Low Pass 10Khz to avoid displaying the PWM signal itself. The measured deviation was no more than 200Hz from the aimed frequency, which makes it feasible to be demodulated by a cheap radio module such as an USB SDR for a latter data analysis using dedicated software.

Below an example captured frame is shown:

```
[I88779J]GSA-11>APRS,WIDE2-1:/16-03-95/00:31:43/0000.00N/00000.00E/RoX=0.00/RoY=-0.89/RoZ=-0.71/AcX=1.57/AcY=-0.08/AcZ=-0.04/MaX=-0.09/MaY=-0.34/MaZ=0.34/T=22.63/P=940.44
```

This frame was captured using an USB SDR through SDR# with the AFSK demodulator QTMM, which is displayed in figure 13. The measured range of the measured transmission was of about 10 meters using $\lambda/2$ antenna. This distance could be significantly improved using a properly designed antenna and a RF amplifier.

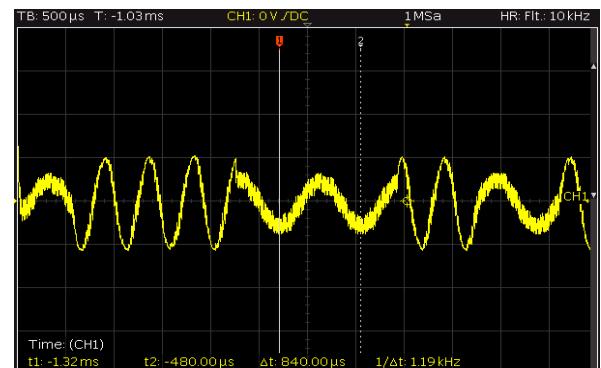


Fig. 11: 1.2 kHz generated signal

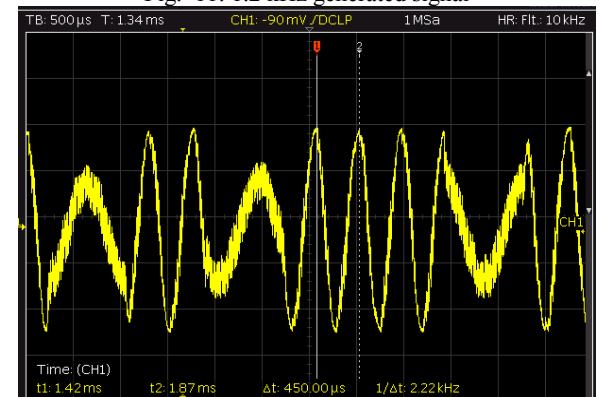


Fig. 12: 2.2 kHz generated signal

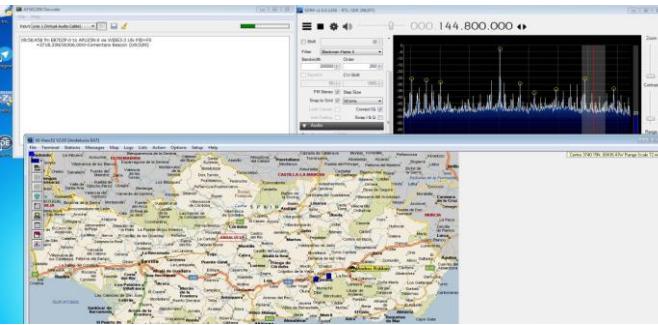


Fig. 13: Signal demodulation with SDR# and QTMM

2) Image capture

The captured image is shown in figure 14, this image shows a test pattern generated by the camera with the purpose of having a known image to test capture and color reception of the frame.

IV. DISCUSSION

During this possible Product Design course project the student could potentially acquire knowledge of how a modern device is designed and fabricated making it a perfect lab work to understand this field. The student will be put in a situation where a product must be designed with other student from different backgrounds, having to auto evaluate their own knowledge in order to create a successful work.

Regarding the electronic design, the student will gain knowledge about RF design and routing along with concepts of digital transmission protocol such as I2C or UART. Also modern PCB design a manufacture will be learn, as well as hardware debugging.



Fig. 14: Captured camera test pattern

In software design, concepts about microcontrollers low level C programming techniques are expected to be used to be able to generate the date at the sufficient speed. In addition, concepts of basic data treatment from sensors and polling are required. If the GUI is finally implemented, the student will learn how to read from external ports in a computer and display data in an understandable way using a high-level language such as Python.

CONCLUSION

The exposed work is a very complete course project suitable for a master degree in a technical field that expects students from very different backgrounds that will challenge them to cooperate while and use their previously acquired technical knowledge to overcome the course. This laboratory/project session is meant for teaching the students the value of team work and how it can achieve the solution of high level technological problems, therefore individual evaluation is discouraged, in order to make the students share the work properly.

REFERENCES

- [1] “Atmega2560 datasheet”, Atmel
- [2] “BMP180 Datasheet”, BOSH
- [3] “LSM303 Datshhet”. ST
- [4] “MPU6050 Datasheet”. InvenSense
- [5] “AX.25 Link Access Protocol for Amateur Packet Radio”, TAPRJ
- [6] Ibrahim, S. H., Ali, S. H. M., & Islam, M. S. (2012). Design a 24-bits pipeline phase accumulator for direct digital frequency synthesizer. In *Proceedings - 2012 International Symposium on Instrumentation and Measurement, Sensor Network and Automation, IMSNA 2012* (Vol. 2, pp. 393–397). <https://doi.org/10.1109/MSNA.2012.6324603>
- [7] Wang, R., Mi, Z., Yu, H., & Yuan, W. (2011). The design of image processing system based on SOPC and OV7670. In *Procedia Engineering* (Vol. 24, pp. 237–241). <https://doi.org/10.1016/j.proeng.2011.11.2633>
- [8] Dilek, S. M., Ayrancı, A., Seker, A., Ceylan, O., & Yagci, H. B. (2012). AX.25 protocol compatible reconfigurable 2/4 FSK modulator design for nano/micro-satellites. In *2012 20th Telecommunications Forum, TELFOR 2012 - Proceedings* (pp. 416–419). <https://doi.org/10.1109/TELFOR.2012.6419235>