Automated Fixed Base Operator for Orlando Apopka Airport

Jordan Ruhl, Carmen Henriquez Rojas, Francisco Galue, Mason Maines

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — When a pilot flies into an airport to land, there are two key pieces of information they need; wind speed and wind direction. Additionally, when a pilot takes off, they must know that their radio equipment is working properly so that they can communicate with other pilots and Air Traffic Controllers. These tasks are usually performed by an airport's Fixed Base Operator. Some small local airports, such as Orlando Apopka Airport, do not have people dedicated to perform these tasks. This paper's primary focus is on our solution to create an automated FBO that, when prompted by a pilot, can broadcast important wind information and perform a radio check without the need for a human operator.**

***Index Terms* — Air Traffic Control, Air Safety, Aircraft Navigation, Broadcasting, Analog Processing Circuits, Codecs, Computer Networks.**

I. Introduction

When a pilot chooses to fly, there are a few things that the pilot must know for certain. Before a pilot takes off they should know that their microphone, radio, and headset are operational. This is so that they can communicate with other pilots in the area to avoid deadly collisions and for communicating with Air Traffic Control at another airport soon after takeoff. Also, as they come into land, a key piece of information to know is the wind direction, wind speed, and gusts at the airport they are landing at. This is because pilots always need to land into a headwind to shorten their landing distance. And if a crosswind exists (and they usually do) the pilot needs to know so they can choose the best runway to land on.

Usually Fixed Base Operators (FBO) are the ones to relay this information to the pilots over the radio, but some airports do not have FBOs. This creates a problem for the pilots who need that very information. One solution to try to mitigate this issue at such airports is a windsock. A windsock is a light and flexible cone of fabric mounted on a mast, usually somewhere along the airstrip of an airport. Windsocks let the pilots know some of the important weather readings, such as wind direction, but they are small and cannot be seen until the aircraft is very close to the airport. On the other hand, there are some automated systems currently on the market that perform task such as broadcasting weather conditions and transmit radio checks, but they are costly and not suited for smaller airports.

The project is a low-cost system that satisfies these two basic needs. This system needs to broadcast important weather information when prompted by pilots in the area. For example, when the system is prompted, the system will broadcast a weather report that includes the latest recorded wind direction and wind speed including gust and variable wind information if it exists. This system also needs to perform a transmit radio check for any pilot that consists of recording the transmission from the pilot and playing it back so the pilot knows exactly how operational their equipment is. Therefore, this can be classified as an "Automated Fixed Base Operator" for small airports. This "Automated Fixed Base Operator" acts as a hub of communication for small airports like Orlando Apopka Airport that do not have a dedicated FBO or weather station. This system would provide a source from which any pilot can obtain crucial weather information or perform any radio communication checks they need prior to taking off and landing their aircrafts. The team calls it the AirBud.

The team's goal is to use their technical experience to connect a weather station and VHF radio through an interface board to a microprocessor that can process all the necessary information. Using these components, the team will build a system that can assist pilots in taking off and landing safely, all while being configurable and cost-effective.

This product idea was brought to the team by Michael Young, an adjunct professor at the University of Central Florida, and is being funded by him as well as Mac Avionics, a local avionics supply and repair shop located in Lakeland, FL, who also donated the VHF Radio used in the product.

II. Previous Operation

There was a previous setup of equipment for the Orlando Apopka Airport to use, but no human operator to run it. The system comprised of a VHF Radio, power supply, a speaker, and a handheld microphone and is shown in Fig. 1.

Figure 1 Previous system used by FBO

Many smaller regional airports use a similar system, but with Orlando Apopka Airport, the lack of human operator causes a need for a better system more suited for similar airports that don't have the manpower to have a dedicated FBO.

III. Problem Formulation

The objective of this project is to research, test, and develop a working product for immediate use at Orlando Apopka Airport. The initial sponsor for the project, Michael Young, met with the team and discussed the requirements for the new system.

- The system shall use a power supply, radio, anemometer, and microcomputer.
- The system shall broadcast the current weather conditions when the pilot keys the mic with the specified parameters.
- The system shall update the pilot and broadcast the current wind conditions if they change outside of the chosen parameters.
- The system shall perform a "Comm Check" when the pilot keys the mic with the specified parameters that does the following
- The system shall allow the user to change any parameter below.
- The system shall have a web IP graphical interface from which the user can read the current winds and make parameter changes.
- The system shall announce crosswind warnings if they are present.
- The system shall announce a favorable runway if conditions fall within chosen parameters.
- The system shall not broadcast if the radio channel is occupied.
- The system shall operate on the airports UNICOM frequency.

IV. Components

The product is comprised of five main components described in detail below.

A. Power Supply

The power supply supplies 14.3 Volts at 5 Amps which will go into the interface board to power the board and delivered to the three other components. The VHF radio will receive 13.8 volts and the microcomputer and anemometer and wind vane unit will receive 5 volts from the interface board.

B. VHF Radio

The VHF Aircraft Radio our system is using is a KX-170B, a solid-state NAV/COMM transceiver manufactured by King in the mid-1970s. It features 720 COMM frequencies and was an industry standard for many years due to its reliability and dependability.

C. Custom Interfacing PCB

The Custom Interfacing PCB will be used to communicate between the VHF Radio, Microcomputer, and Anemometer and Wind Vane Unit. This interface board will also be responsible for power distribution to all circuits and the Microcomputer. As well as be prepared for individual signal testing without the radio at hand.

In the design of the interface PCB one of the main choices was to only use positive power supplies. This led to the design choice of using LM324 and that family of single supply operational amplifiers as opposed to the dual supply operational amplifiers that are more commonly used in laboratory environments. Other key aspects the interface board design include I/O sockets for signals. This involves the choice of a DB9 connector for radio interfacing, an RJ11 port for the anemometer and a 40-pin connector for the microcomputer.

Finally, as a failsafe for all integrated circuits, apart from voltage regulator, all components being used in the system are through hole. This is to prevent the whole PCB becoming useless should one fail. Instead by using sockets they can be replaced without compromising the PCB.

D. Anemometer and Wind Vane Unit

The anemometer our system uses is a Davis Instruments 7911 Anemometer. It is a component of the Weather Monitor II and Weather Wizard III, both of which are complete weather stations also manufactured by Davis Instruments. The 7911 Anemometer features 3 polycarbonate wind cups to measure wind speed and a UV-resistant ABS plastic wind vane to measure wind direction. It comes with a 40-foot long, 26 AWG cable that ends with an RJ-11 connector. It can measure wind speeds up to 173 knots (200 mph) with a 1 knot resolution and a ±5% accuracy. It can also measure wind direction from 0 degrees to 360 degrees with a 1-degree resolution and a ±7% accuracy.

E. Microcomputer

The microcomputer the team is using is a Raspberry Pi 3 Model B and is discussed more in depth in Section VI.

V. Interface Board

The interface board is the main electrical component of our system that binds all components together to form one coherent and functioning product. The main purpose of the interface board is to inject and extract signals into the VHF Radio and the microcomputer it is necessary to modulate them into a compatible form. This is especially important when it comes to setting the radio into transmission mode without requiring a physical method, and modulating the audio signal at reception and prior to transmission.

Key aspects of the interface board that will be required for system operation is the power distribution from the main supply to all circuits and anemometer. It will be required to modulate signals coming to and from the radio. The final key aspect of the interface board is the integrated testing switches and LED's. This will give the user a visual representation of circuit function without the need of the micro computing device. This is to help the user identify any errors that may appear after prolonged use of the system. Allowing them to identify if the problem comes from the circuit or the programming.

A. Power Distribution

The interface board is responsible for powering the wind vein as well as all the circuits inside the board. This is done through three linear voltage supplies throughout the system. The main power supply's voltage will be lowered to 12V for the operational amplifiers used in audio, the push to talk relay, the carrier detect comparator, and the LEDs. The main supply will also be lowered to 5V to power the wind vein, and finally the 3.3V source will be use to supply the logical 1 to the Raspberry Pi 3 in the comparator.

B. Push to Talk Relay (PTT)

The push to talk relay is a circuit design to simulate the action of pressing down on the transmit button on microphone. This is done to be able to simulate that action and automate the sequence required to transmit through the radio. This is done by using a npn transistor with its base connected to the GPIO of the Raspberry Pi 3. This allows the Pi to use the transistor as a relay switch by supplying the base with 3.3mA and setting it into saturation mode. Allowing the collector to send the incoming 13.8V that sit on the PTT line to ground. This in result sets the radio into transmission mode thus allowing the Pi to communicate its message to the pilot.

One key feature of this circuit is a reverse diode on the

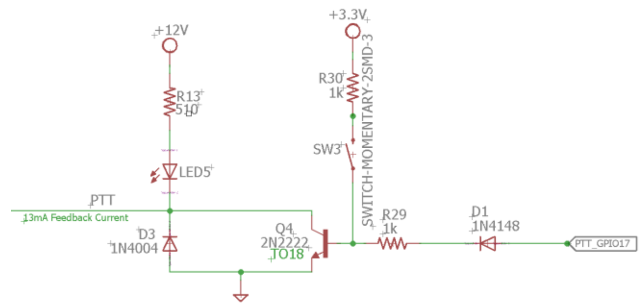

Figure 2 The schematic of the Push-to-Talk circuit that will simulate the action of a microphone keying in.

13.8V line that is rated at 400V. This is because when the inductor inside the radio is grounded the magnetic field around it collapses and sends a large surge down the line. This diode will pull said voltage down to the ground and protect the Pi from any feedback. This is also repeated on the testing switch which simulated the Pi's logical 1. The line to the pi has a smaller 1n4148 diode that will prevent any feedback for when that supply is tapped into for testing.

C. Carrier Detect

The Carrier detect circuit in the interface board is used to notify the Pi of a busy communication channel, as VHF communication is half-duplex, and therefore not interfere with currently undergoing communication. This also serves as the primary trigger in the systems logic. By tapping into the VHF radio's squelch capacitor, it compares the voltage at a high or low with a reference on the other input. The comparator is inverted as the squelch goes low when a carrier (communication) is present on the channel. This has its logical 1 then amplifier to 3.3V by the voltage supply. This signal is then pushed to the PI's GPIO and to an LED for ease of testing

Another function in this circuit is that the radio input is on a switch. This switch will ground the input signal to the comparator and simulate an active channel. This is part of the included test switches to be able to trouble shoot the interface board should there be a malfunction.

D. Wind Direction

Wind direction is one of the two wind related signals that are communicated to the Raspberry Pi through the wind vein. The wind vein is connected to the interface board through an RJ11 connection. Through this the interface board supplies the internals of the 7911 with a 5V supply and a ground. The 5V is then fed to the wind vein, which is a potentiometer, which changes the output from 0-5 depending on the direction of the wind.

This circuit takes that signal and first has an LED in parallel that will shine brighter as the input increases intensity. This serves as a visual representation of the wind direction and helps troubleshoot the circuit. Should that light not be on then there is a problem with the anemometer getting power.

The second aspect of this circuit is a voltage divider that brings down the maximum voltage to 3.3V. This is done to make the output of the wind vein compatible with the MCP3008 analog to digital converter.

E. Wind Speed

Apart from wind direction the 7911 also communicates wind speed with the raspberry pi. This is done through a reed switch that gets closed once every revolution of the cups. No power is fed through the reed switch on the anemometer as when it closes it simply goes to ground. To work around this and provide the Pi with a pulse for every revolution a pnp transistor is used.

By supplying the emitter with a voltage of 3.3V and running a resistor from there to the base it will allow the base to supply a current and activate the transistor every time the reed switch closes. This will in turn send a pulse to the Pi therefore allowing it to analyze the pulses and convert them into a reading of wind speed.

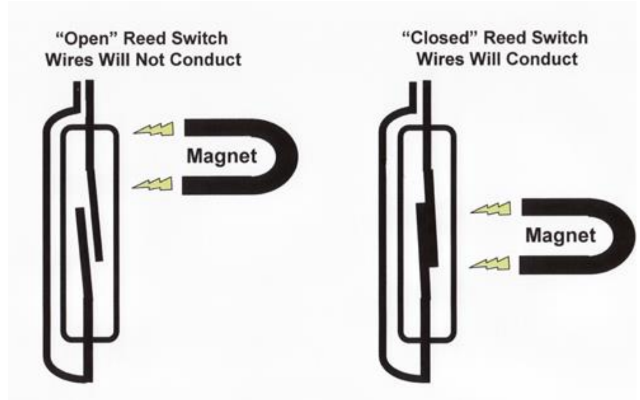

Figure 3 The reed switch when closed ground the base and allows the current to flow through the emitter and into the collector.

F. Automatic Gain Control

The automatic gain control, or AGC, is an internal circuit inside the VHF radio that is intended to amplify the received signal to the desired volume for the user to hear. This is now being adapted to identify the signal strength of the incoming communication for the communication check protocol.

The signal varies between 2.5V for a very weak signal around -102dB to 9.5V at -20dB. Because of where this is located in the radio it comes with some analog noise to it. Thus, the system has an active low pass filter to filter it out. This is then followed by a unit gain buffer to then revert it back to a positive value.

Then because the signal is going to be pushed through to the analog to digital converter cannot exceed 3.3V the filtered now DC signal is reduced to a maximum or around 3.3V through a voltage divider. Lastly the circuit again has an LED to help indicate the power level of the incoming signal as well as help troubleshoot the circuit.

G. Analog to Digital Converter

The MCP3008 is the first of the two integrated circuits that will be used in the interface bard. This converter will provide a 10-bit digital representation of the automatic gain troll and the wind direction. The MCP 3008 will be powered by the Raspberry Pi 3 at 3.3V. This chip takes an input between -0.6V and 3.9V but all inputs to it will be kept within 0 and 3.3V.

The second chip will be the AUDIO CODED Proto. Which will provide the Pi with the RX Audio coming from the VHF Radio. And it will provide the radio with the TX audio used for transmission. It too shall be powered by a

3.3V input and its input signal will be kept within 0 and 3.3V.

Both chips will provide information to the chip through the SPI bus.

Figure 4 AUDIO CODEC Proto by MikroElektronika one of the two integrated circuit systems being used for analog to digital conversion.

H. Received Audio

The RX Audio circuit will take the demodulated received audio from the radio and deliver it to the AUDIO CODEC proto integrated circuit. To do this there are various modifications that need to be done to the original audio.

The first modulation required is to remove any incoming DC bias in the signal by placing the positive end of a polarized capacitor on the input. This is done to allow the offset to be set to a desired value and protect the board from any undesired and unexpected bias changes in the original signal.

The next step is to remove any noise in the signal through a low pass filter. This will filter any noise above 17kHz. And then it will have an applied offset of 2.5V. This is done by supplying the positive input of the operational amplifier with a voltage divide 5V DC input.

Lastly prior to inputting the signal to the codec it will have another voltage divider that will lower the offset a bit more. This is done to maintain more standard values in resistors and keep the variance in the interface board lower. This offset signal will also be sent to the base npn transistor that will switch it on and then vary the brightness of the LED on the emitter. To illustrate that signal is being pushed through it for troubleshooting purposes.

I. Transmit Audio Circuit

The final circuit on the interface board is the transmit audio compressor. This is key in maintaining the longevity of the VHF radio, as replayed audio from the pilot in the communication check may be too strong for the radio and can damage it. This audio is injected from the AUDIO CODEC proto to the microphone input in the radio. Since the VHF radio being used is old it is a risk that is unnecessary. This is chosen to be done through soft compression as hard compression would result in too much fidelity loss. Making the playback harder to hear by the pilot, as well as distorting the signal in such a way that it would essentially defeat the purpose of the play back feature in the communication check protocol.

To do this the codec is first protected by the DC offset applied to the signal using a polarized capacitor with the positive end facing the circuit. The signal is then offset by 6V and passed through a low pass filter. The signal is then fed into a half wave rectifier that removes the DC bias then pushes the positive side of the wave through the gate of a PMOS. This creates a variable resistance in the positive input resistance that will lower the audio and soft compress it should it exceed 2.1V peak-to-peak.

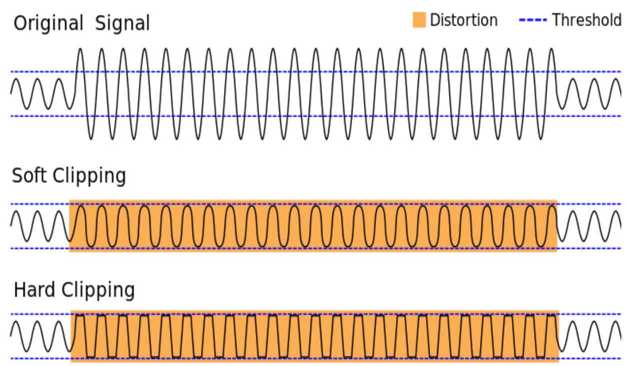

Figure 5 A visualization of the two methods of audio clipping demonstrating the greater loss of fidelity in hard clipping than in soft clipping.

Due to a load inside of the radio, a buffer amplifier is placed prior to delivery of the signal to ensure that the infinite resistance would discharge a key capacitor, in an unintended way, in the rectifier aspect of the circuit.

Lastly the final DC bias on the audio signal is removed prior to delivery to the radio. The output of the signal also has polarized capacitor preventing any feedback voltage coming from the radio to interfere with the final output prior to passing to the radio. This is because when the radio is set to transmit the microphone line has a voltage pushed through it that is unnecessary for this intended purpose as it won’t be powering a microphone any longer.

Finally, like every other circuit in this interface board the audio signal is also tied to an LED. This is done by feeding the base of a transistor the audio signal, thus setting the base into saturation and passing the 12V supply through to an LED and setting on as the audio plays back to the pilot.

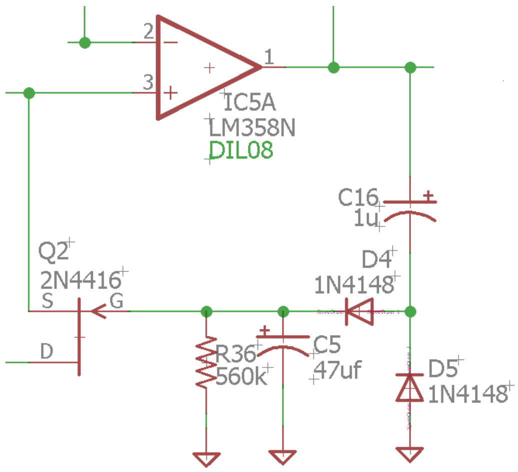

Figure 6 The half-wave rectifier used in the TX audio circuit demonstrating the use of the PMOS as the variable resistor on the positive input.

J. Design Stages

Prior to arriving to the final system design each individual circuit was tested via an individual breadboard circuits. After testing each input and output to their respective receiver the first design for each circuit was finalized. Making sure that all inputs and outputs were within reason for each circuit's intended purpose.

After choosing a design a full system protoboard was assembled with all systems and test switches soldered on. Through this full system test some values were tweaked to make the whole system work together in unison and allow for a fully operational board. This design was then adapted for the final PCB design.

Each of these tested systems went under various tests. All analog signals were viewed under an oscilloscope. All DC voltages were checked with a multimeter for accuracy, and even the transmitted signals were viewed through an RF receiver and transmitter for calibration of the AGC and to make sure TX audio was transmitted with little audio fidelity loss.

K. PCB Design

When determining the layout of the PCB it was very important to lay it out such that any I/O that wasn't the Raspberry Pi 40-pin connector was on the left and the power switch on the right. This is for easier cable management when connecting to the radio. Keeping everything neatly on one side.

The secondary aspect of the PCB design was the choice to set mounting holes for the AUDIO CODEC PROTO. This decision was made to be able to make standoffs for its' stability. Though it caused harder internal wiring it will pay off by improving the device's longevity.

When it came to component choices some polarized capacitors were kept as through hole components due to a much lower cost than surface mounted ones. LED placement was made so that they were all aligned and next to each other so that there was a clear depiction of system function for the user.

The final PCB design was one made for practicality, convenience and longevity. Where each design choice was made for ease of use, and to allow for some failures without it resulting in needing a new PCB to be assembled.

VI. Microcomputer

A necessary part of this solution is a microcomputer that can be the brains of the system and can take in all the information given by the interface board and output the information to operate and broadcast from the radio. The chosen microcomputer, the Raspberry Pi 3 Model B, is a fully-fledged microcomputer with a 1.2GHz 64-bit quad core ARMv8 CPU, 1GB of LPDDR2 RAM running at 900 MHz, 4 USB ports, and naturally runs a distribution of Linux called Raspbian. Other features of this microcomputer include 10/100 Ethernet, 2.4GHz 802.11n wireless, Bluetooth 4.1, 3.5mm audio jack, and a microSD card slot. The Pi's main line of communication are the 40 GPIO pins which also provides support for SPI communication. It runs on 5V and can output 3.3V or 5V through the GPIO pins. The sheer computing power and endless documentation of the Raspberry Pi 3 make this a good option.

VII. Software

To interpret all the signals given to the microcomputer by the interface board, the microcomputer had to constantly run software, developed by the team, that satisfied the requirements. The microcomputer uses a compilation of python scripts that start running on startup of the system, making the product easy to setup and use.

A. Wind Speed

The script that calculates the wind speed uses interrupts to calculate how many times the anemometer rotates in a 2.25 second interval. That number is the wind speed in miles per hour, which then gets converted to knots. Lastly, that value replaces the oldest value on a list and the average wind speed and peak gust speed is then calculated by the 20 values in the list. This allows a more progressive change in wind speed while keeping the accuracy of the peak gust speed for when wind speeds are changing drastically within the 45 second window.

B. Wind Direction

The wind direction script operates much in the same way as the wind speed script. A new wind direction is calculated from the signal given by the wind vane and the ADC and is added to a list. Then the average wind direction, minimum wind direction, and maximum wind direction is calculated from the list. This gives the user the current wind direction as well as the variable wind direction, should they exist.

C. Main Loop

The main loop of the software is where all signals besides wind information are analyzed and where main user functions are determined. Starting with the system turning on, the main loop will be waiting for a carrier detect signal. Once a carrier detect signal is recognized, a function will calculate how many different times carrier detect is recognized within a time frame. This is how many times a user "keys" their microphone. If three carrier detect signals are detected, the script will run a function that broadcasts a wind conditions report. If four carrier detect signals are detected, the script will run a function that performs a transmit radio check.

When broadcasting a wind conditions report, the function will tell the user two main pieces of information; wind direction and wind speed. These are the two most important pieces of information that a user needs to know. Once the values of wind speed and wind direction are calculated, the script will play the necessary mp3 files that correspond with the numbers and words that need to be said. A sample of a standard message is,

"Orlando Apopka Airport. Automated Weather Advisory. Winds. Two. Two. Zero. At. One. Four. Knots."

When a transmit radio check is initiated the script will prompt the user for the check. Then once the script recognizes a carrier detect signal, meaning the pilot has keyed their mic, the script will start recording until the carrier detect signal is no longer recognized. During the recording process, the AGC will be calculated from the data given by the ADC. Once the recording process is finished, the script will play back the recording and play back the signal level. This will allow the user to not only hear the quality of their signal, but also receive a quantitative value of their signal. This will allow users to determine if their equipment is operational to use before they take off.

D. Artificial Intelligence Components

Another useful facet of the system is the use of artificial intelligence to determine what needs to be broadcast during a wind conditions report. At the beginning of the report, the user will be greeted by a good morning, good afternoon, or good evening. This will be determined by what time of day it is and then the script will broadcast the correct greeting.

The main use of artificial intelligence will be to determine what additional wind information is relevant to the user. For example, if a peak gust and variable winds are recorded, the system will announce that they exist and their values, but only if they break a certain threshold determined by the main user. If a peak gust is 1 knot greater than the current wind speed, the system should not report the peak gust because it is not significantly greater and therefore will not impact the user.

The final use of artificial intelligence is to determine if a crosswind exists or if any runway is favorable to land on. If a wind is mostly perpendicular to the runway and the current wind speed is greater than 5 knots, the system will report that a crosswind exists. This will allow pilots to prepare for a crosswind landing or decide to land at a neighboring airport. If the wind is mostly parallel to the runway and the wind speed is greater than 5 knots, the system will determine which runway is favorable to land on, providing the pilot with the best way to land given the current conditions.

D. Settings

An important feature of our product is the use of configurable settings. These are determined by the main user who must log into the website to change them. The configurable settings include

- Carrier Dwell Time (ms): min & max
- Interval Period between Carriers (ms): min & max
- No. of clicks for Weather report
- No. of clicks for Comm radio check
- When to report gusts over steady winds (kts)
- When to report variable winds (degree variation)
- History time for variation (secs)
- When to announce UPDATE: deg change (deg.) & wind change (kts)
- Wait time after last report to announce UPDATE (minutes)
- Runway Headings (degrees)
- When to announce CROSSWIND WARNING: wind speed (kts) and degrees min and max (degrees)
- When to announce WINDS FAVOR runway choice #1: wind speed (kts) and degrees min and max (degrees)

- When to announce WINDS FAVOR runway choice #2: wind speed (kts) and degrees min and max (degrees)

VII. Conclusion

The purpose of the fully functional system is to provide an automated fixed base operator that a hobbyist could use without needing a loan. The system will provide wind updates, wind advisories, and a transmit radio check. All being done using a popular off-the-shelf microcomputer and a specially designed interface board. With cost in mind all choices in this project were made to be on a budget, with the most expensive system being the radio, but with the radio being 40 years old it's easy to find a used one at a decent price. The main challenges of this system included customizing all circuits to fit a very specific parameter to produce signals for both the microcomputer and a radio from two different technological eras. Mixing two very different technologies safely so that neither of them damage the other when communicating. This was achieved through extensive testing to make sure that signals were isolated, then modified to the other technologies parameters before combining them. The secondary challenge was compiling a code that would allow for the synthetization of messages, the recording of messages and have an artificial intelligence level that would provide a high degree of user satisfaction. This was achieved through extensive trial and error with the software to adapt it to changing components, individual tests and eventually a fully integrated system.

The final product provided is one that can satisfy any personal pilot and their unregulated airport.

Acknowledgement

The authors wish to acknowledge the assistance and support of the sponsors for this project for their guidance and help in funding the product. The authors kindly thank Michael Young and MacAvionics of Lakeland, FL.

Jordan Ruhl is a Senior studying Computer Engineering at the University of Central Florida. Software development is his passion as he continues to study and learn new languages and usages like Python and web development. He will pursue a working career in that field with the hopes of pursuing a master degree after a few years in his career.

Carmen Henriquez Rojas a Senior at the University of Central Florida majoring in Electrical Engineering. Currently working in the TI Innovation Lab at UCF. She is experienced with various PCB modeling software and is largely responsible for the final PCB layout of this project. She has worked on various other projects with Limbitless Solutions and work at Kennedy Space Center and helped countless students throughout her employment at the Innovation lab.

Francisco Galue is a Senior at the University of Central Florida pursuing a major in Electrical Engineering. Having interned at ELCA Telecomunicacciones, a telecommunications company in Venezuela, working on cellular phone links his passion for the telecommunications field has grown. This passion and experience have played a crucial role in the design on the project and also giving interest in courses like Satellite Communications and Computer Communication Networks to expand his skills in that field.

Mason Maines is a Senior at the University of Central Florida pursuing a degree in Computer Engineering currently holding an internship position at SunGard Public Sector. He is someone that enjoys all aspects of software as well as hardware, continuing to learn and practice new skills related to these areas, such as Web Software technologies and programming frameworks to other home brewed hardware projects. He looks forward to finding work in the industry that challenges him and allows him to work a variety of different projects.