Vehicle Reverse Parking Assist System

With PWM, Sensor, and Audio

John Akroush, John Kastran, Andrew Lee

Electrical and Computer Engineering Department
School of Engineering and Computer Science
Oakland University, Rochester, MI
e-mails: johnakroush@oakland.edu, jkastran@oakland.edu, andrewlee@oakland.edu

Abstract— **To reduce the number of accidents that occur when parking, this vehicle reverse parking assist system will allow drivers to be aware while reversing by alerting them with a beep if they are close to an object. It will also vary tone as the vehicle moves closer to the object to give the driver an estimate of how close the object is. The final product was accomplished through the combination of software and electrical hardware.**

I. Introduction

"About 20 percent of all vehicle accidents happen in parking lots, according to the Insurance Institute for Highway Safety." [1] Vehicles are a huge part of most people's lives. They provide convenient and quick transportation for people around the world. With this comes the risk of being involved in an accident. Driving can be dangerous at times. Combining this with the busy and stressful lives of many people leading to the "rush" mentality, drivers may not always be fully alert when driving.

The Vehicle Reverse Parking Assist System with PWM, Sensor, and Audio is the perfect tool to help lower the risk of accidents occurring; especially when parking or reversing in any scenario. This will allow drivers to feel confident when reversing in a parking lot. It takes the worry out of accidentally coming in contact with something or someone in your "blind spot" behind you. The project was accomplished by incorporating aspects of software using VHDL and electrical components into one device.

II. Methodology

A. Hardware and Programming

The experiment is using the Nexys A7 FPGA board and Vivado to use VHDL to simulate a distance alert system and implement a physical test. The approach is to determine a good sensor for ranging, ultrasonic HC-SR04. Now to understand how the ultrasonic sensor handles distance. It uses a period with a trigger to enable the sending of a pulse. The time it takes to come back to the ultrasound is the distance. Therefore when the pulse returns, it will send a new distance measurement through the system. Also, there is a switch that acts as a vehicle shifting into reverse or forward motion. The reverse mode will enable the system, and the forward will deactivate the system.

B. Ultrasonic Logic

The system has its input signal from the sensor, but what do with it? The task is to measure the distance as a unit of time. The circuit board in use has a clock of one hundred megahertz. Next, is to count the number of clock cycles after the trigger is sent and before the returning pulse is received.

C. Distance and Alerts

The distance has been quantified as a unit of time. Interpreting the distance as a frequency is going to take more thought. The distance is to warn the user when something is near the rear of a vehicle. If the foreign object is close to the vehicle, then it is to make a tone pitched to alert the user. To separate the different severity of distance, the tone must change pitch. The pitch of each setting must be drastic enough to hear the difference. In this experiment, it was settled on four distinct tones.

D. Organizing Distance

The ultrasonic distance to the tone is another challenge. There needs to be a system to handle different distances. These different distances are not all of the possible distances, but just five. Four tones while a foreign object is there, and fifth for being off while nothing is there. Thresholding the distance count is the chosen method. While this count is above five different threshold values it will be one of the five tone options. It will check the closest distance threshold first, so it has the least delay when warning the user. Then it will check the furthest distance, then the medium, then the close, and then the dangerously close. Its organization is better described in the finite state machine section. Once the threshold is detected, it will send the tone until the next period of the ultrasound. The ultrasound can only send one reading every period so the delay is completely dependent on how fast the distance sensor is. A faster sensor could use light, but for this experiment, we will stay with the ultrasound.

E. *Generating Tone*

Generating a tone is now the next task. It has three components. The first is to create the duty cycle. The second is square wave generation. The duty cycle from the previous controls the volume. The third will take the frequency designation, and turn this information into the pitch of the square wave. These single bits are sent to the speaker causing it to move and create the pressure wave that one can hear (Llamocca). This is the final output that the user will hear and ends the component descriptions.

III. Electrical Design

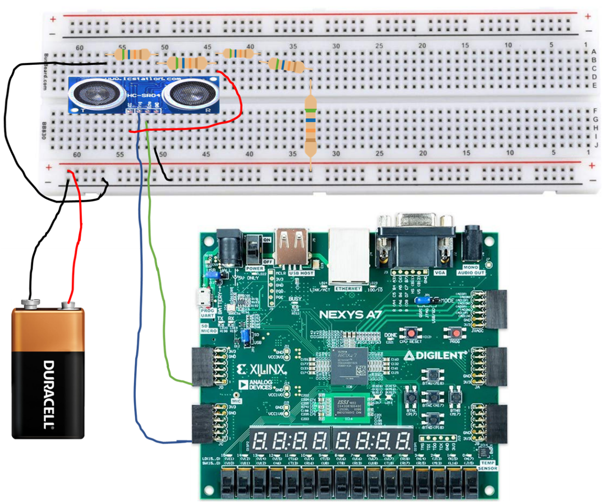

Figure 1: Schematic of Vehicle Reverse Parking Assist with PWM, Sensor, and Audio

Figure 2: HC-SR04 Ultrasonic Distance Sensor

To accomplish the final goal, the following components were used:

- Ultrasonic Distance Sensor (HC-SR04)
- Nexys A7 50T FPGA
- Audio Jack (built-in)
- 10, 100, 220, and 330 Ohm Resistors
- 9V External DC Battery
- Breadboard
- Earphones/speaker
- Wires

All components coincided with the Nexys A7 board which can be seen in the schematic in Figure 1. The trig pin on the sensor shown in Figure 2 is directly connected to the JC1 port on the Nexys board. This is the trigger pin that receives a value back from the program as seen in Figure 3. The echo pin on the sensor is connected to the JD1 pin on the FPGA. The sensor is connected to common ground, but the voltage required was different from that of the source. The Nexys board only provides 3.3V, so we needed an external battery. We chose to use a 9V DC battery. Our sensor requires 5V, so this is where we used the listed resistors to create a voltage divider circuit that would output the correct 5V voltage that we needed. The audio jack was designed to produce a tone that varies in pitch depending on the data from the sensor. All of these components were combined to create the final product.

IV. Experimental Setup

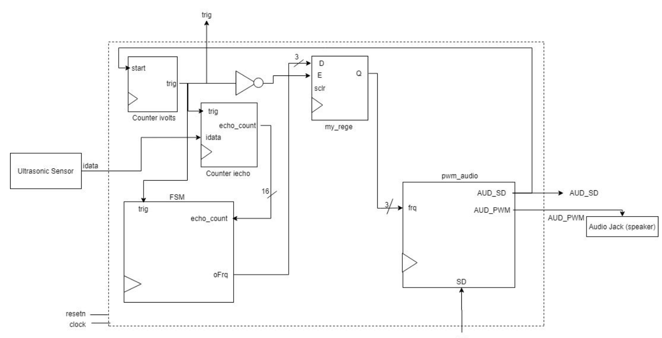

Figure 3: Block Diagram

F. *Counting Blocks*

These two counting blocks shown in Figure 3 work together to keep track of the period for the ultrasonic sensor and the actual distance read from the ultrasonic sensor.

There is a period counter that counts, ten nanoseconds clock, to a full period of seventy microseconds. Every time the counter reaches its total it will reset back down to zero. It will also send a trigger signal to logic high for the duration of ten microseconds. This trigger signal controls the enable on the register, the start and stop signal to the distance counter, and the finite state machine that dictates the tone. This period counter will set the trigger to high for the first ten microseconds, and then low for the next sixty microseconds.

The distance counter will receive the trigger signal height and then wait until that same signal to become low. On the clock tick that it becomes low, it will start counting. The distance counter will count until the ultrasonic sensor returns data logic high. So simply put, the counter starts counting when the ultrasonic sensor has sent its pulse wave and then stops counting as soon as that pulse wave returns. The time it took to send the pulse wave and return it is directly proportional. Finally, every time the trigger is set to high, from the period counter, it will update the actual distance for the finite state machine to interpret this distance as tone.

G. Finite State Machine (FSM)

The purpose of this block is to control the pitch of the tone produced by the speaker based on the distance between the ultrasonic sensor and the foreign object. It will receive the distance from the ultrasonic sensor and make decisions accordingly. The logic is to check the distance at certain thresholds. If the distance is greatest the speaker will play a low pitch unthreatening tone. As that same distance decreases the pitch will become higher. So the distance and the pitch are inversely proportional. There are five thresholds to consider: nothing is there, something is there, close, danger, and most severe. With respect, furthest distance, second furthest, medium, close, and closest distance. These five thresholds of distance, thus lead to the five states, and the five sound pitches. The five sound pitches are, off, low, medium, high, and highest. Now, how does this machine decide what to do?

The state zero will check if the ultrasonic signal is triggered logic high. If the signal is high it will remain in state zero until that trigger is logic low. This makes sure that every cycle of the ultrasound is measured properly by the FSM. It will now check if the distance is greater than the threshold. It will check if greater than 85cm, then the next state check if greater than 50cm, next state check for 18cm, next state check for 10cm. Each state checks for the greater distance, so if the distance is very low it will be sent to give the highest pitch and the severe warning. Now the frequency output from the FSM is dictated by remaining in the appropriate state. This is as desired.

H. Register

This component will hold the correct frequency that the FSM has output. It receives the FSM frequency data all of the time, but it only updates its own register output on the trigger signal. Since the trigger is the mark of a new ultrasonic period it will ensure the latest distance is from the most recent completed count. This is to ensure that the next component, tone control, isn't being sent five different frequencies within fifty nanoseconds. This is from how the finite state machine operates.

I. Tone Control

The purpose of this block is to send signals to the speaker. A speaker is a resistor and a membrane that pushes against the air causing a pressure wave. This movement can be interpreted as logic height and low, and as opposed to analog. A square wave is just that, logical high and logical low. The circuit board clock can be used to generate a constant pulse, and from this can build a duty cycle, to adjust the square wave. Then the frequency input must be handled. This is not trivial. Each tone must be converted into a signal that can be manipulated by a pulse width modulation. The frequency bits are used as the exponent of two and then converted into an integer. This will then become the total number of periods in a given time. The greater the frequency value the more periods per time.

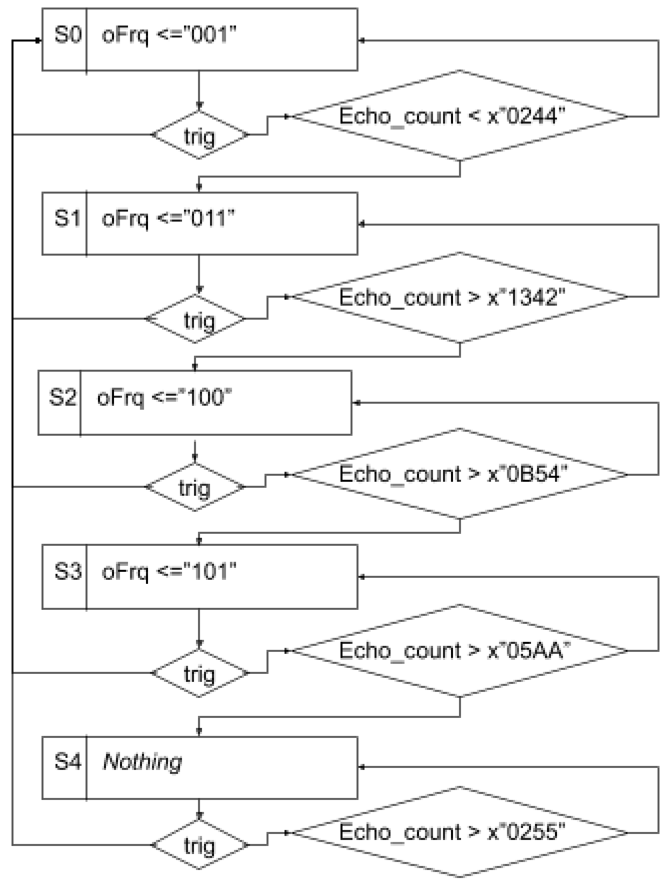

Figure 4: ASM Chart of FSM

J. Top File

The Top File connects all previously described blocks and components together. Here, we designed the input and outputs of the program itself. The top file receives the idata input from the sensor and SD value from the designated switch, as well as, outputs three values; the trig for the sensor, AUD_SD which controls the audio jack shutdown, and AUD_PWM which is the data for the output tone. Inside of this program, the trig signal output from the ivolts counter must be inverted before entering the register. Generic values for components that utilize them are set to the needed values, and every component is ported according to the block diagram shown in Figure 3.

V. Results

The resulting product from this project is a fully functioning vehicle reverse parking assist with PWM, sensor, and audio. The final product was accomplished through the use of VHDL code combined with various electronic hardware devices such as; an ultrasonic sensor, resistors, audio jack, and Nexys A7 FPGA board. The functionality of the vehicle reverse parking assist is as follows, as the driver reverses, the ultrasonic sensor will detect anything in its range. As the vehicle moves closer to the object behind it, the speaker's tone increases in pitch, which depends on the PWM value from the circuit, to alert the driver that they are getting close to coming in contact with the object. Overall, the implementation of the ideas for the project has been successful. A picture of the product is shown in Figure 5, below:

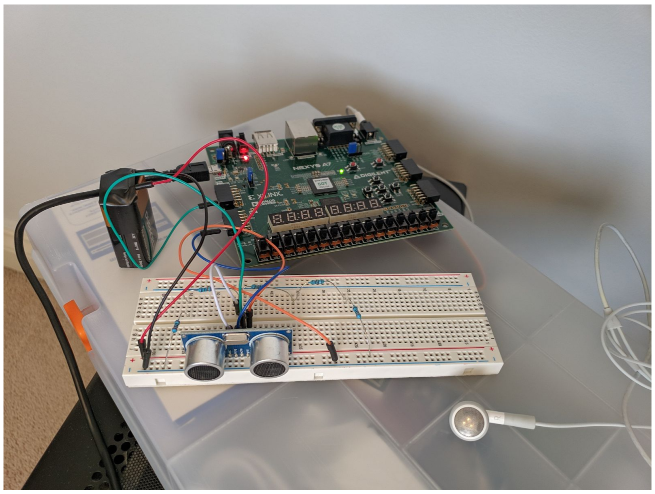

Figure 5: Vehicle Reverse Parking Assist with PWM, Sensor, and Audio

In Figure 6 it depicts the period of trigger toggling on and off. This signal is sent through to many other parts mentioned in the preceding parts of the document.

Conclusions

We have grasped how to use counters and the clock measure signals, and then build more systems on top of these countries to organize the measured signal. This strategy may be applied to many other sensors and could bring about more creative ways to organize data. We saw that bits can be manipulated into interesting patterns to make desired outcomes. The tone controls are a great challenge to understand. There are a lot of moving parts there, and it is very clever. Being creative with just high and low signals is the most important part of learning computer hardware. Issues to be solved are how exactly to make the tone shut off completely and then resume. It is currently not handling the switching from forward to reverse mode. There are future applications that can be built outside of this experiment. The ultrasonic distance can be sent to other subsystems to control lights or mirrors or even breaks.

References

[1] Insights, S., Vehicles, A. and Crashes, P., 2020. Prevent Parking Lot Crashes - State Farm®. [online] State Farm. Available at: <https://www.statefarm.com/simple-insights/auto-and-vehicles/prevent-parking-lot-crashes> [Accessed 20 April 2020].

[2] Llamocca, D., 2020. VHDL Coding For Fpgas. [online] Secs.oakland.edu. Available at: <http://www.secs.oakland.edu/~llamocca/VHDLforFPGAs.html> [Accessed 20 April 2020].

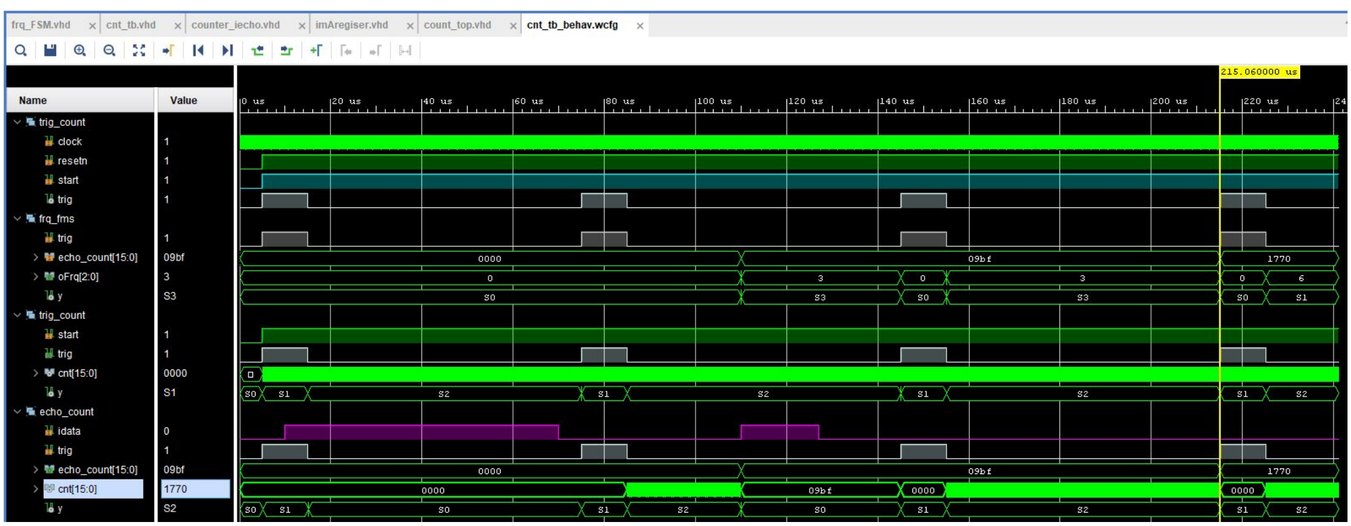

Figure 6: Trigger Pulse in gray, and Incoming Echo in Purple