

Blind Spot Monitor

Vehicle Blind Spot Monitor

List of Authors (Tim Salanta, Tejas Sevak, Brent Stelzer, Shaun Tobiczyk)

Electrical and Computer Engineering Department

School of Engineering and Computer Science

Oakland University, Rochester, MI

e-mails: tsalanta@oakland.edu, tjsevak@oakland.edu,
bwstelze@oakland.edu, smtobicz@oakland.edu

Abstract—Cars have blind spots that cannot easily be observed while driving. This composes a safety hazard; a collision can range from property damage up to injury or even fatality. This project seeks to detect if an obstruction is in this area, and warn a driver before changing lanes or otherwise entering occupied space to avoid collisions. By designing this project we hope that something like this would be able to help with overall traffic safety as well as help with driver ergonomics, which can be comprised when difficult blind spots are checked repeatedly. Our team has developed a blind spot monitor, which works on a smaller scale than a car would need. It could be scaled to a larger size in order to work with any vehicle as well as extended (by adding more sensors) to guide the rear and front of the vehicles. For the purposes of this project, our team has decided on setting up two sensors, to simulate the driver and passenger side of a vehicle. The system that was developed works well for demonstration purposes, the hardware and code work great together, and light up an external LED light that indicate the “blind spot” is occupied. However, in order to use this system in a vehicle more accurate sensors should be used, as the ones that were used in the demonstration were relatively simple. For use in a real life situation, i.e. in a car, the system would require more accurate and more expensive sensors. The coding and hardware would work well, however the sensors appear to pick up a bit of noise due to their error margin. Thus, the recommendation for using this system in a car would require better sensors in order to improve accuracy and reliability.

I. INTRODUCTION

The report will cover the overview of our blind spot monitor project. It will cover the basic design, set up and results achieved. The scope of this project is to use the technology and skills that were practiced in ECE 378 in order to make a blind spot monitor system that will be reliable, cost-effective, and easily implemented (with a larger budget) into many different cars and trucks. The National Highway Traffic Safety Administration reports 300 fatalities and 18,000 injuries occur yearly in the United States due to blind spots.[1] Granted, blind spot monitors are available as options today in many vehicles but it would be a great practice to design a system that would keep vehicle costs down and become standard on just about any vehicle for the general safety of the public. There are common misconceptions that state blind spots can be eliminated by adjusting the vehicle mirrors in a certain way, but this is very difficult or impossibility for some people to do. Thus, we found this project idea to be very useful and applicable for the general safety of society, today. The topics that were learned in class were from completing the laboratory assignments, which practiced the coding as well as learning to use the Nexys4. By the time that our finish product was coded, the methodology to allow the sensors to check for an obstruction in the blind spot changed a few times. There were two major components used in creating the top file for this project. The first is the proximity detector, and the second is the sensor handler. Both were used in the top file, and then ultimately in the test bench file in order to create a working system. An initial idea was to implement a finite state machine into the code for the project but after some discussion it was deemed that it

would not be necessary and in fact better to not use the finite state machine. The way the code operates and was designed will be discussed more in-depth, further on in the report. Our team however is pleased to report successful results from our project and a picture that shows the set up will be shown below.

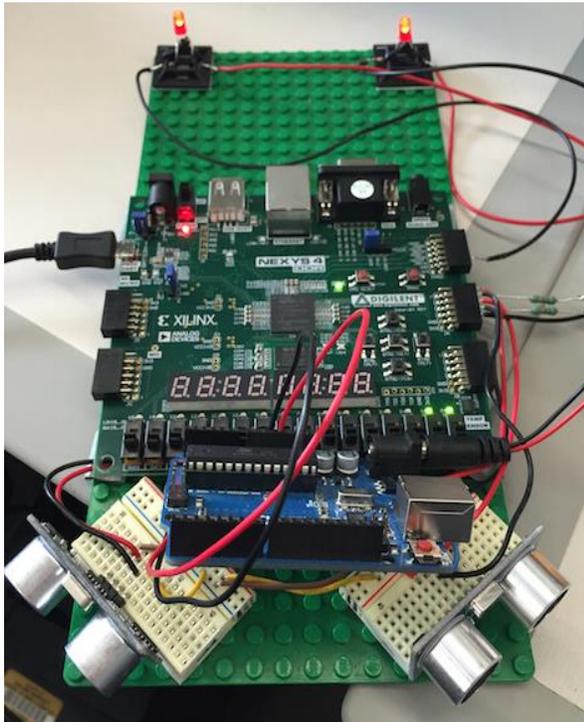


Figure 1: Blind Spot Monitor Overview

In figure 1 above, we can see that both LEDs are activating, indicating that an obstruction was in the blind spot, the area approximately six centimeters away from the sensors. By connecting two sensors to the Nexys4 board, it simulates the two sides that a motor vehicle has and can require blind spot monitoring for.

In order to build a successful, working project, some research was required to be done outside of class as we expected when using different technology. A few things that were learned outside of this class were how each type of sensor works and what types of sensors we should. In addition, we used knowledge of electrical engineering in wiring the LED's from the Nexys4 board. The different types of sensors that are available on the

market today are endless; there is something for every level of precision and budget. We thought about using sensors such as Lidar and infrared. These sensors were ruled out because of its high cost and complexity to implement using FPGA. Each sensor has different capabilities depending on what type of application they are used for. The biggest influence in choosing our sensor was of course budget, however it turned out that our sonar sensor used, was relatively high quality compared to its cost. Another major factor that really influenced our decision for which sensor to use in this application, was the beam pattern of how the sensor picks up critical information (such as a car or person in one's blind spot). The beam pattern performance will be shown below, for reference.

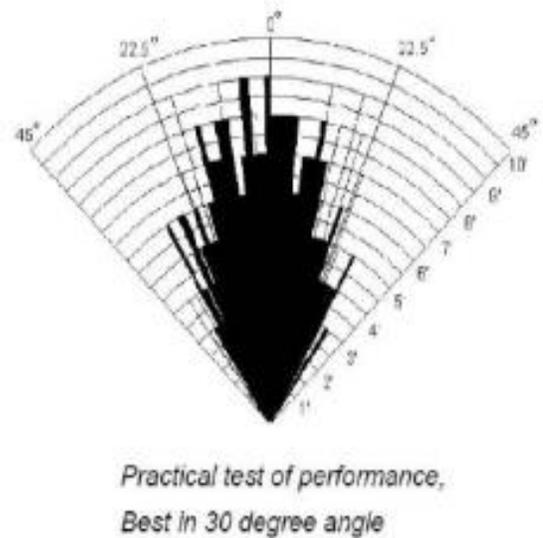


Figure 2: Sonar Sensor Beam Pattern[2]

From the figure above, it can be noted that sensor works best within the 30-degree angle range, however, the sensor was still obtaining information even further than these. This is another reason that this sensor was chosen, because it had a relatively large beam pattern that worked well across the distances necessary. Sensors will be discussed in greater detail in the following parts of the project.

II. METHODOLOGY

A. Design Methodology

A considerable amount of time for this project has been spent discussing how the signal of a vehicle in the oncoming lane, or an obstruction, will be brought into the Nexys4 board. The device chosen for this application was the sonar sensor, with a model number of HC-SR04.



Figure 3: Sonar Sensor Model HC-SR04

It provides a relatively accurate, short-range detection, which can be implemented with Nexys4 board, for an affordable rate. A few key points about this sensor are that it has a 15-foot range, requires 5V, and is most accurate between 15 degrees. A few sample formulas that were used in calculating the distance from the sensor include:

$$\frac{\mu S}{58} = \text{centimeters}$$

$$\frac{\mu S}{148} = \text{inch}$$

The following equation was also used in order to calculate the range of the sonar sensor:

$$\text{Range} = \text{high level time} * \text{velocity} \left(\frac{340m}{s} \right) / 2$$

These equations were both provided by the data sheet of the sensor. [3] The sensor works by shooting the sound wave that can detect different objects. This device can communicate in two different ways. One way is getting the value by using the PW signal. The second way to detect the

object is by using a form of serial communication. This would make the project slightly more complicated, but it would fulfill the same purpose. The following figure that will be shown will describe the timing diagram of the sensor. To begin the sensor measuring, a high signal is sent, which then waits for the reflected signal to come back. Once something is detected, it will send out a voltage of 5V from the output and the delay in this correlates to the distance away from the sensor that an object is.

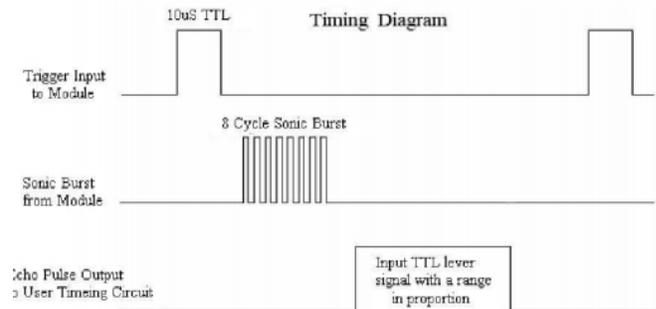


Figure 4: Timing Diagram – Sonar Sensor

The above diagram simply represents the process of how the sensor detects an object within its path or beam pattern. The sensor is supplied with the 10µs pulse to trigger the sensor. Then the sensor will send out 8 cycle burst of ultrasound at 40kHz and raise its echo. [3] The remaining hardware that was used in this project, includes the Nexys4 board that was used regularly in the lab section, along with two LED's that were wired from the Nexys4 output pins to indicate that the sensor is outputting 5V, or that something is in front of the sensor at the distance that was set in the project coding.

B. Design Methodology

The first topic that was discussed was how many sensors we will be using as well as what the output will do. This was done in simple terms in order for the project to be easily planned out. The FPGA module was the center of everything that will be going on in this project. We knew the Nexys4 board would be able to function as this and take in a certain signal, while outputting a

certain signal to turn on the LED. The planning block diagram from the planning phase of this is shown below.

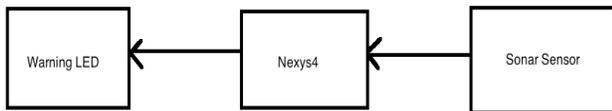


Figure 5: Block Diagram

In the diagram above, it can be noted how the initial planning of the project looked like before the code was designed and the hardware was implemented.

The next part of the project was designing a code and developing it to work accurately, as well as in a continuous cycle. As mentioned in the introduction section we had two modules, a top level, and a test bench for this project. The two modules will be discussed first, which our team has called Proximity Detector, and Sensor Handler. The module called Proximity Detector is in charge of exactly what it sounds like – detecting an object that has come within the respective set distance of the sonar sensor. This code had an output of the LED, an input of the clock, and the in out of the JA signal, used in the body of the code. The overview of this code is to basically take the trigger, echo, clock, reset, and to figure out if an object is within the set distance. It was then duplicated for both sides, in order to have two sensors functioning at the same specifications. Of course, multiple sensors could be running on this same code if necessary. The next module was the sensor handler, which uses if statements to continuously check for an object that has entered the dowel area of the sensor. The way the clock was set up on this module, allows for an essentially instantaneous response to turn the LED on and off, which is a really successful achievement for our team. Next, the top level tied both of these modules together and then the test bench was created. For the test the sensors received bench echo signals in order to figure out if an object is present in front of the sensor. The clock speeds and rates were adjusted accordingly,

in order to have a seamless integration between sensors and LED lights turning on.

Figure 5, gives the most accurate definition of how the code works in the project by showing the necessary conditions to move through and eventually send an output signal to the LED. As mentioned before, the project will send out a trigger signal, and wait for the echo, and continue on a pattern very similar to this in order to continually check if something is indeed in front of the sensor.

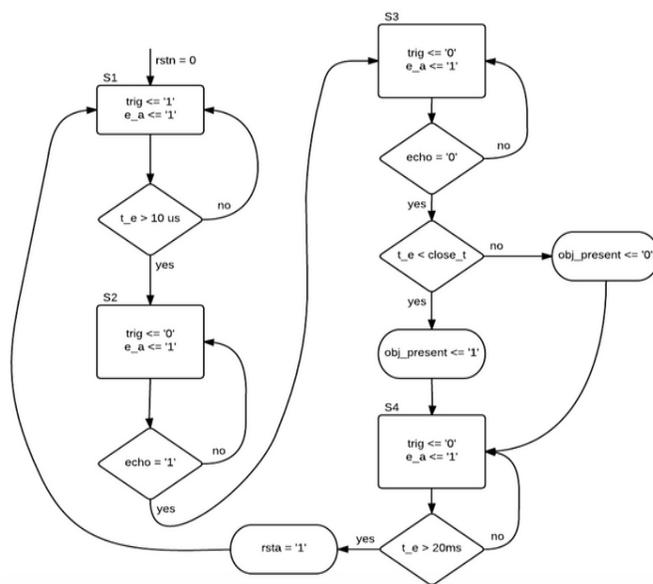


Figure 6: State Diagram

Deciding factor for using the state machine was that it is easy to implement and it was more reliable than just writing a pseudo code like structure. That is hard to loop and would run forever without bond. With state machine you can guarantee that it will stop once the code has been executed. With define states it is easy to debug and understand what is going to on. It also does not create unnecessary flip-flops and latches. In state 1 trigger is set to 1 and to enable the sensor for 10µs. Once the sensor has received the trigger signal it will send an echo back which gives us 1 for the “echo” this will set the trig signal to low and then we will wait for the echo signal to go down again. Once we have detected that the signal

echo signal has gone low from high then we measure the signal “chose_t” for x about of time. Then by the use of formula’s presented earlier in the report we can output 1 or 0 to the signal “obj_present.” If we find that the signal is less then 20ms then we go back to state 4 to recheck if there is an object. However, if the signal is longer then 20ms then we have detected object and we go back to state 1.

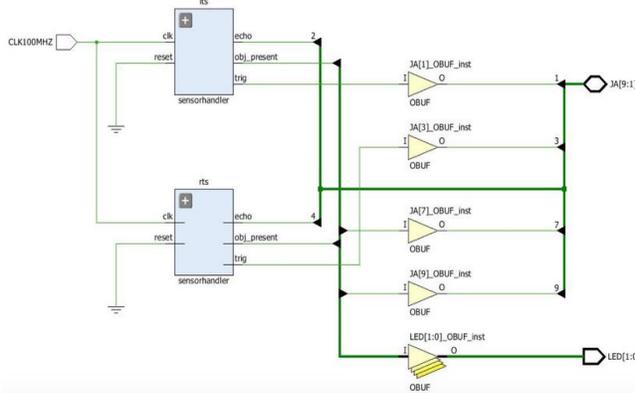


Figure 7: Top Level Design

Figure seven shows the top-level design of the code, and how the software portion of this project has come together.

III. EXPERIMENTAL SETUP

The plans of the general setup have been discussed and the structure that holds the sensors and the use of warning LED’s is completed. As you can see in the figure 1 picture. The anticipated coding will be very similar to our laboratory with Vivado is used to write the code and program the Nexys4 board. In order to further develop the project, additional hardware we may need includes the sensors. If we had more time, we could have built an audio feedback systems or written warning on LCD such a “there is car on left.” However, for the purposes of this project we decided to focus on making sure the task of turning on the LED and turning it off works without glitches or issues, which we have accomplished. The reason we stayed with this plan is that it could potentially be implemented as an aftermarket type of addition to a vehicle, for a relatively affordable price. We have decided to

use two sensors to use two sensors to simulate a real life car. Ideally, two sensors would be obtained and could be programmed exactly the same way – just copied once one sensor is working properly. However, each sensor will be outputting to separate LED’s, and they do not output to the same LED. The reason for this is that there should be an LED in each side mirror on the vehicle for ergonomics of the driver, in order to avoid straining to check various blind spots continuously. By using the LED lights on the Nexys4 board we would be able to set up a different output for each sensor. The final set up is shown in the figure 1 picture, and it displays how the system was presented live, in the class.

In the configuration shown above, the sensor sends data to the FPGA, which would then process it and determine whether a warning to the driver is necessary. In this situation, an LCD display could be mounted and developed in order to warn the driver of distance to the object in question. Also, a warning LED will light up if you enter a danger threshold. Finally, our assumptions were proven that with proper coding learned from class and laboratory assignments, the system above works correctly. That is, it successfully identifies if cars are in the area and that the sensors work - regardless of weather or certain lighting conditions.

IV. RESULTS

Our team was able to archive successful results obtained from the project. To verify our results, we used different signal and waveform detecting tools. This gave us ability to see the actual signals in real time. This way we were able to tell that he 3.3 v supply from the board was not giving us the proper results. With the 5v were able to archive satisfactory results. Two LED’s up front on the side of the “car” worked as they were supposed to. As the object (a solid object) is waved in front of the sensor, the LED lights up to show that there is an object. The physical blind spot monitor is completely built and it has been properly tested thoroughly. However, there seems to be always some noise in the sensors. Our assumption is it’s

because of running two sensors through board. There might be some transmission problems in the line. Or another problem might be the proximity of sensors to each other, and the quality of the sensors that were used. It is also likely that the other sensor can easily pick up sonar waves of one sensor, causing noise or slight error. In order to solve this problem, it is assumed that taking more sonar samples over $10\mu s$ could give us more accurate results. We could have used higher resolutions sensors to also give us more accurate results.



Figure 8: Timing Diagram

In the timing diagram above we can see that the trigger is high for few microseconds and then the signal throws sonar for $10\mu s$ then the result as obtained from the echo.

CONCLUSIONS

In summary, we have learned a lot about how the Nexys4 board could work to take an input from a sensor and output it to an LED, or LCD, or any other type of output that is compatible with the board. We have used the same code that would work for each sensor, and it has provided us with great results thus far. This project has given us a good experience in implementing certain components and coding them to work together for a common purpose, on a larger scale.

We did not run in to as many issues as much as we thought we would. The HC-SR04 worked as it supposed to. These sensors were cost effective and reliable. Although there were some resolution issues, but in general it worked as it supposed to,

efficiently. As of now the sensors can detect object within 2 or 3 feet of the object, which is not very practical in real world. Again, our project was scaled to work on a much more small scale for demonstration purposes. By using further filtering and better sensors we could potentially increase the reliability and range by a factor of two (2). Some of the problems that were faced include the lack of having a few more sensors that were necessary for the project, in order to use different sensors to test the reliability and check the improvements that it would make in the project by modifying and swapping out sensors.

Further improvement in this project would be to display the distance from the object on the LCD screen. We could also add a sensor in front and rear of the vehicle to detect any object when parking or reversing the car.

By eventually taking this project from the cradle to the grave, we have learned what it's like coming up with an idea, designing the solution to this idea in a team environment and making potentially large-scale decisions as a team. Overall, our project has been a great learning experience thus far in working together, communicating, and carrying out our roles in the team.

REFERENCES

- [1] statistics provided by <http://www.forthinjured.com/blind-spot-accident.html>
- [2] datasheet, by Cytron Technologies, https://docs.google.com/document/d/1Y-yZnNhMYy7rwhAgyL_pfa39RsB-x2qR4vP8saG73rE/edit
- [3] datasheet, provide by Micropik, <http://www.micropik.com/PDF/HCSR04.pdf>