– ECE 09.321: Milestone 1 –

Final Project

Thomas Haskell and Rajinder Parhar
Rowan University

December 19, 2018

1 Design Overview

The purpose of this design was to provide a simple way for a user to implement a backup sensor. This sensor would be relatively cheap and simple to set up in a vehicle. Making it the best choice for old vehicles since they usually do not have the capabilities to install back up cameras without a 1000$. It consists of three main components an ultrasonic sensor, a microprocessor board, and a display. This would also have the option to function over wifi/Bluetooth or a direct connection within the vehicle.

1.1 Design Features

These are the different features that were implemented in the design the provide robustness and more value to the product through software means instead of hardware means. This makes as a simpler setup for the average user being able to perform this project on their own.
Design features:

- Wifi module
- MQTT uart RX
- Median Noise Filtering
- Staggered LED log brightness
- Ultrasonic sensor interface

1.2 Featured Applications

- Ardunio capabilities with Wifi
- Easily switching of sensors

- Security protocols preventing false information

1.3 Design Resources

Here you can find the github repsository: Click Here for Repository

Here you can find the data sheet for the HC - SR04 Click Here for DataSheet

Here you can find the data sheet for the ESP8266 Module Click Here for DataSheet

1.4 Block Diagram

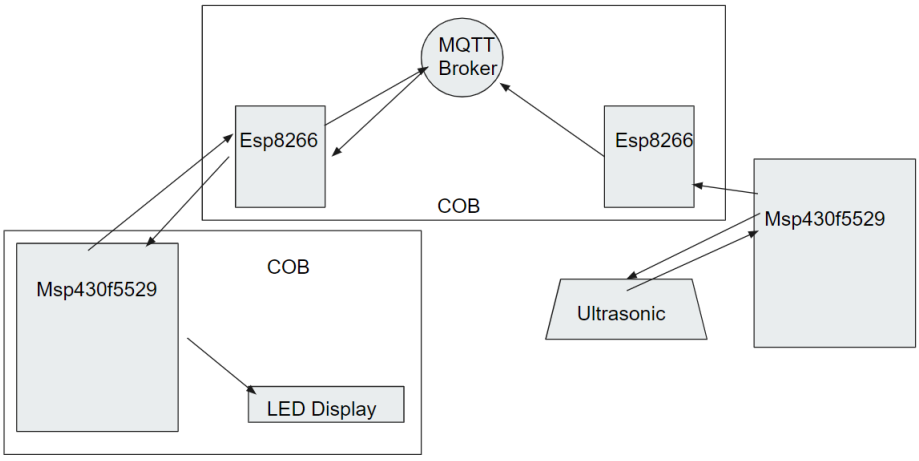

Figure 1: Block Diagram

1.5 Board Image

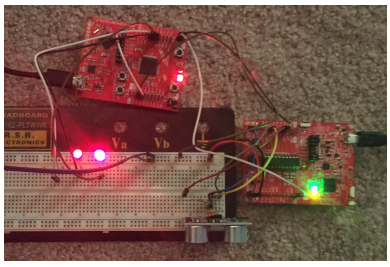

Figure 2: Block Diagram

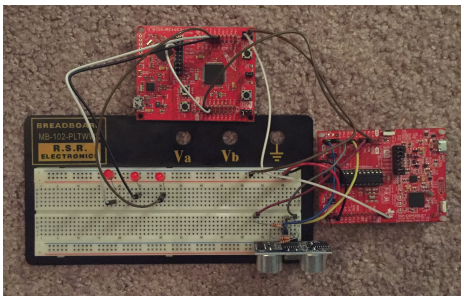

Figure 3: Block Diagram

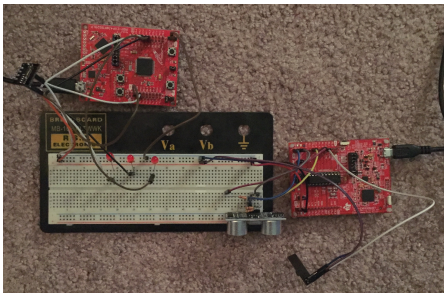

Figure 4: Block Diagram

2 Key System Specifications

OBJECT	SPECIFICATION	DETAILS
Bypass Capacitor	0.1 F	Used in conjunction with the power supply
Voltage Divider	for 3.3V	reduces input of 5v to 3.3v
Wifi Module	ESP8266	3.0-3.6V, 80mA
LED	Basic LED	2v 20mA
Ultra Sonic Sensor	HC-SR04	5v, 15mA

3 System Description

This system is used to solve the issue of older cars not having the luxury of a rear camera. A rear camera is very impactful to the driver as it provides another safety net for the driver to ensure they don't damage other vehicles. The solution we present is more rudimentary but can be a great aid for the driver as it provides them a proper gauge of distance, especially if they lack spatial gauging. This system can be placed on the back of the vehicle near the license plate where it can provide natural cover for the system and still give the proper angle for it to function. The display of LEDs can be placed in view of the driver wherever it would feel most comfortable.

4 SYSTEM DESIGN THEORY

In this system, the Ultrasonic sensor will use a single transducer to send a pulse and to receive the echo. That data is then sent to the MSP430F5529 which will use a lookup table to determine the proper distance. The MSP430 than will send that signal over wifi as an ASCII value through an ESP8266 that then is then received by the other ESP8266. The other microprocessor than receives that value and uses it to tell the three LEDs to light up in a particular order and different stagnation in brightness based on the distance value calculated. “

5 Planned Implementation

1. Increase median filter sample rate, and drop more values
2. Power integration with vehicle
3. Security Patches
4. LCD Display

6 Getting Started/How to use the device

To Understand how to set up the device in detail, look at the picture below. This diagram will offer all the connections needed to set up the device using the MSP430s, ESP8266, and the set of LEDs. The Ultrasonic module will be connected with the 5v pin from the G2553 as that can supplement enough power for the board, as well as the wifi chips. Ensure that all the grounds are tied together as they give a central ground point for the circuit. After matching the pins ensure you are using 1k & 2K resistor for the ultrasonic module so that the microprocessor board can take in 3.3v on the input pin.

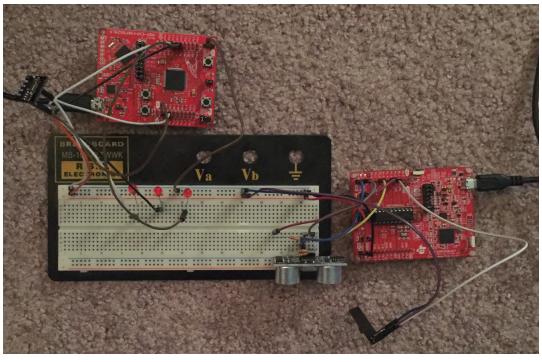

Figure 5: Board layout to setup the wifi setup

7 Getting Started Software/Firmware

The following is need for initial setup:

1. RealTerm
2. Code Composer Studio
3. BreadBoard
4. Variable Power Supply
5. DMM Probes
6. LEDs
7. Voltage Regulator
8. Ultrasonic Sensor
9. Jumpers
10. MSP430 Micro-Processor
11. ESP8266

This list contains the software needed to communicate with the micro-processor. RealTerm is used to communicate using the RX and TX line while Code Composer Studio(CCS) is used for loading the program onto the micro-processor. CCS also has features such as debugging the program and Simulink. The rest of equipment is used for setting up the physical systems and testing the system to ensure it is functioning properly. Using a Oscilloscope to test the voltages would be preferable but a multimeter will also give a similar result.

7.1 Communicating with the Device

8 Test Setup

To test this setup it is possible to use the debugging tool in CCS to see how the code would run. Realterm can also help show the values of what the micro-processor is receiving. To view the the sensor working, it would be beneficial to run real term to see the values of the returned by the sensor to see if it is tuned properly. In order to test the result in realterm, you must program the msp430f5529 board with the main.c in Wifi5529 directory, uncommenting lines 100 and 101. This will send an integer value via the uart connection to realterm. Default there is no transmission via uart to the wifi module however since it does not need to publish any data back to the MQTT broker. In order to test the wifi setup, you must wire the esp8266 boards as shown in figure 6.

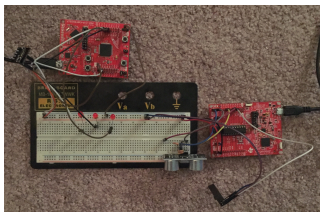

Figure 6: WiFi Configuration for the Backup Sensor

8.1 Bill of Materials

- 2x ESP8266 - 14$
- Ultrasonic Sensor HC SR04 - 8$ for 5
- 1k Resistor - .02$
- 2k Resistor - .02$
- 3x LEDs - .03$
- 2x Msp430 Boards - 20$

8.2 Test Data

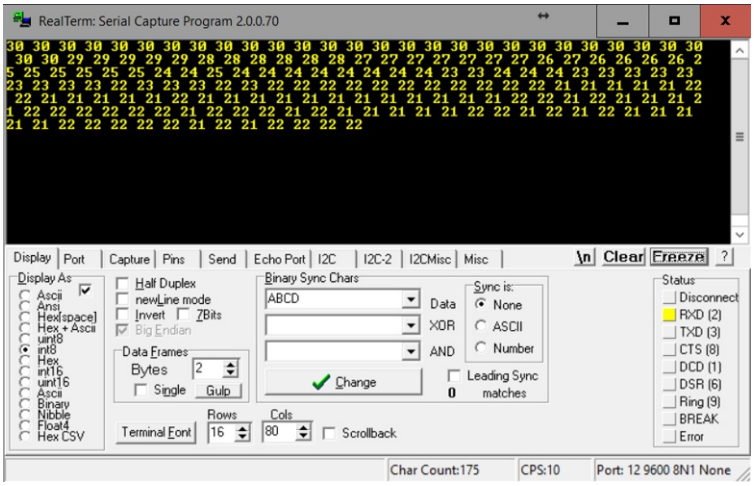

Figure 7: Real term data shown, processing the data