Standalone Instrument Cluster Display

Grant Scott III, Michael Carlson, Erin Lawler
Electrical and Computer Engineering Department
School of Engineering and Computer Science
Oakland University, Rochester, MI
gcscott@oakland.edu, macarlson@oakland.edu, elawler@oakland.edu

***Abstract*— This paper presents an example of a standalone Embedded System representing an Instrument Cluster Display in an automobile. The environment will include a Dragon 12 Plus development board using a Freescale HCS12 microcontroller. This project will use features integrated on the development board including temperature sensor, LCD Display, LED light and Manual Push Buttons along with an additional distance sensor all of which are currently found in an automobile environment. The system developed will use function based Software to ease use with other Microcontrollers and/or Sensors.**

I. Introduction

Henry Ford revolutionized American life in 1908 with the dawn of the Model T automobile. For the next century, the motor vehicle would prove itself to be essential to modern society's demand for mobility and comfort. As the industrialized world continued to progress, our needs for security – while remaining connected to the surrounding environment – elevated as well.

Safety and restraint initiatives collaborating with new avenues of vehicle-to-driver communication sought to meet those needs. Innovative research areas such as User Experience (UE) and Human-Machine Interface (HMI) developed new methods of interacting with drivers and increasing their awareness of surrounding roadway conditions. Intelligent vehicle features fostered heightened driver intelligence. Engineering and political scholars at George Mason University note the essentiality for automotive intelligence, "Driver errors resulting in accidents cost $300 billion annually in the United States. While intelligent vehicles will not be 100 percent accurate…they will likely achieve a level of control and awareness that no human could possess" [3].

Innovations that allow their users to be fully engaged with their gadgets, while also remaining in touch with the surrounding environment, allow for heightened security and foster a sense of interdependence between man and machine. Vehicle-to-Driver communication occurs most productively when the operator's attention is not far-removed from the road. Thus, communicating alert messages within the field of vision – the instrument cluster, or center console – is ideal.

The scope of this project is limited to the DragonLight HCS12 board and its onboard peripherals as well as the Ultrasonic range finding device. This project is intended to be a demonstration of several different vehicle features in a small package. It would be of use to engineers of various vehicle systems including the HVAC/climate control engineers, exterior electronic engineers, and HMI/instrument panel engineers. A wide variety of topics covered in class will be used in the project. Various assembly commands are used, especially in timing-critical instances. The input capture and output compare resources are used to interact with various peripherals including the ultrasonic sensors and the RGB LED. Interrupts are used for timing purposes and to handle button presses, changing the display's units. Analog-to-digital conversion is used to read the temperature sensor value. Finally, general purpose input output ports are used to interface with the LCD display on the Dragon board. These topics are all used in the pursuit of the vehicle systems demonstration project.

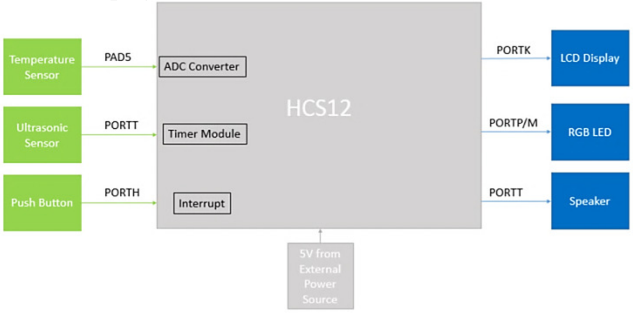

Figure 1. Project Block Diagram

II. Methodology.

A. Reverse Object Detection

Object detection is an integral part of the modern vehicle. As vehicles flow through traffic, calculating the distance from surrounding obstacles in real-time is essential to safe travel. Observing Equation 1, we calculate the test distance from a surrounding obstacle as a factor of High Level Time and sound velocity, 340m/s.

$$Test\ distance\ = \frac{(HLT)(v_s\ )}{2}$$

Equation 1. Ultrasonic Distance Measurement

This portion of the project employed the HC-SR04 ultrasonic sensor. The 5V, 15mA distance finder operates efficiently in the 3cm to 3 meter range, allowing for an adequate model for probable obstacle encounters on the road.

For simplicity, the code used in Lab 4 was applied with very few modifications. This code uses an output compare register to emit the ultrasonic distance finder's trigger pulse, thereby beginning the measurement routine. Figure 2 displays the register's configuration.

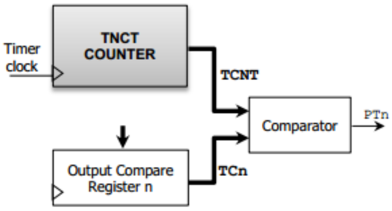

Figure 2. Output Compare Register Configuration

The Output Compare Register is compared with the current value of the TCNT Counter at every bus clock cycle. When the two components are identical, the corresponding channel is set, thus setting the program in motion. The device emits ultrasonic waves for a minimum of 10 µs. The device now listens for the return signal, an 8 cycle sonic burst at 40 kHz from the module.

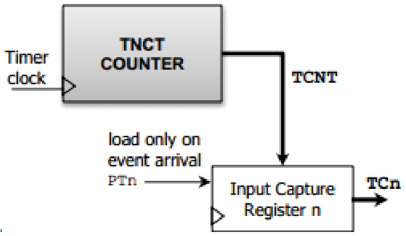

Figure 3. Input Capture Register Configuration

An input capture register, which can be viewed in Figure 3, is utilized to measure this pulse width of this sonic burst. Upon the arrival of the return signal, it loads the corresponding channel and transmits a pulse with a length corresponding to the distance of the object. Figure 4 displays this process in greater detail.

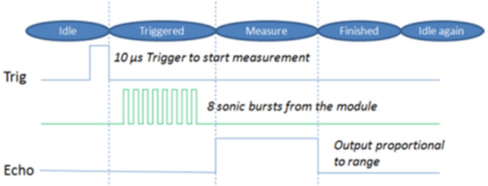
Idle
Triggered
Measure
Finished
Idle again
Trig
10 µs Trigger to start measurement
8 sonic bursts from the module
Echo
Output proportional to range

Figure 4. Ultrasonic Distance Finder Measurement Routine

This pulse width is translated to operate the row of LED's based on distance and also to a PWM frequency which drives the buzzer. Due to the LED component using the pulse width module, a timer is employed to ensure proper operation of the buzzer. As objects approach the ultrasonic sensor, the pitch rises in an attempt to capture the driver's attention and avoid a collision.

B. LCD Display

The on-board LCD Display is used as a visual for the occupant to show the exact temperature (within Worst Case tolerance) in order for the occupants to have the ultimate information to use for decision making. The LCD Display on the Dragon12 can be setup many ways. This project uses 2 line 5x7 font selections along with 4bit data and mode pins. Using 4bit data, results in relaying 2 messages to achieve the correct control and data to the LCD. This creates 7 pins connected to the MCU to control the LCD Display. There is a possibility for 8bit data but this would entail 11 pins connected to the MCU. The tradeoff is the amount of MCU pins versus time to relay messages.

A display with less connections to the MCU would potentially save on MCU cost. An example of this is a display that uses SPI communication especially if the protocol could communicate at ~10MHz which would allow for fast updates. Of course, SPI would force the MCU to be routed on the same PCB as the Display due to the speed. Although, a PCB with both MCU and Display would save on wires in the vehicle. Another feature that could be used is the ability to dim the display at night to not distract the driver.

C. Temperature Sensing

Temperature is used in many areas of the vehicle such as displaying external temperature and climate control. This type of information is useful for the operator and vehicle as vehicles are becoming more self-controlled and knowing the temperature could lead to different operation if the vehicle senses slipping in the cold rather than in a warm climate.

This project uses an on-board TI LM45 which is a small 3 pin device. The output of the LM45 is connected to an ADC port on the HCS12 which is then sampled and used in the SW for displaying on the LCD and warning indicator via the RGB LED. Figure 5[1] displays the block diagram of the HCS12 which incorporates a Sample & Hold type ADC.

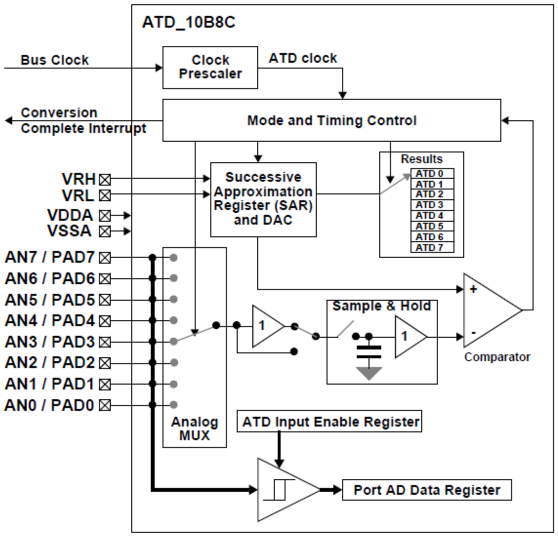

Figure 5. HCS12 ADC Block Diagram

The sample time of the ADC is necessary to verify the accuracy of the sample. If the sample time is long, better accuracy is achieved but the time waiting in the application may cause issues verses a short sample time may offer less accuracy due to capacitance charge sharing. Figure 6[2] displays an equivalent circuit that can be used to calculate the needed sample time using the Source Resistance, External Capacitance, MCU Pin Capacitance, Switch Resistance and Sampling Capacitor.

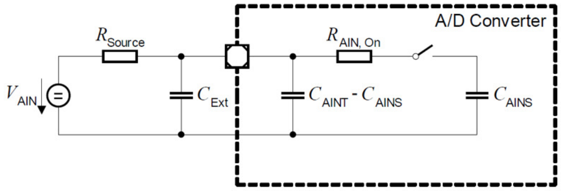

Figure 6. Equivalent Circuit for Sample Time

The LM45 sensor defaults output of 0°C to 100°C at 10mV/°C which the SW will display by default. There was the possibility to add additional components to the LM45 sensor to read negative temperatures but this was not in the scope of this project. Choosing a different sensor that can read the full automotive range is a better option. This project displayed the ability to collect the temperature data and display it multiple ways to the user.

D. RGB LED

The RGB LED serves as a qualitative temperature indication. At zero degrees Celsius, the LED is blue to show the possibility of sleet, freezing rain, and snow. As the temperature increases, the blue color fades to red, ultimately peaking at around sixty-five degrees Celsius, the maximum ambient temperature seen in the automotive industry. The RGB LED is comprised of a common anode and individual cathodes for each color, red, green, and blue. The red, green, and blue cathodes are connected to the Port P ports as shown below:

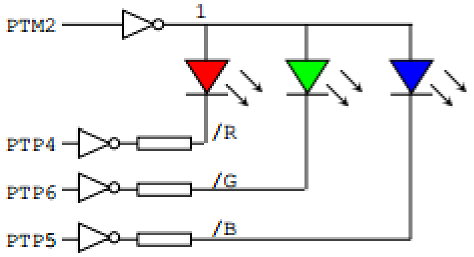

Figure 7: RGB LED diagram showing each PWM port, for RGB respectively this is port 4, 6, and 5.

The PWM channels were configured using the same parameters for simplicity: Clock prescaler SA with a period of 6000Hz, active-high polarity, left aligned, eight-bit, 100 cycle period, or 60Hz PWM frequency. After the settings are configured once, the remaining task is to write a duty cycle value to the PWMDTY*x* register. The PWMDTY*x* register is a zero to 255 value that corresponds to the intensity of the color on that particular PWM port. For example, to get a blue color, the red PWMDTY4 register and green PWMDTY6 registers were set to zero, and the blue PWMDTY5 register to 255. Continuity of the color scale required a bit more consideration.

Since the color scale starts at blue when cold and ends at red when hot, the decision was made to have white represent a mid-range temperature. This required phasing-in the green LED in the middle of the range to avoid mixing red and blue light exclusively to get purple at mid-range temperatures. The concept and equations are illustrated below:

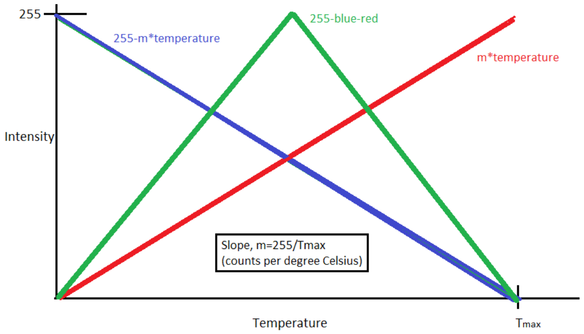

Figure 8: Color scaling vs. temperature

After each temperature read operation, the color values are calculated and the duty cycle of each cathode is updated. The scale is continuous from zero to a calibrated maximum temperature, which we have selected to be eighty-five degrees Celsius.

The method of color scaling described above was the final outcome of several different trial approaches. The first attempt used the maximum temperature and the current temperature to arrive at a multiplier of the maximum intensity. In a floating point system, this would be the most logical method of controlling the LED brightness, however, in our integer only environment on the HCS12 microcontroller, only whole numbers must be used. The next attempt was similar to the first, but numbers were scaled by large numbers first to alleviate the percentage problem. Although this approach could have been successful, it was unwieldy and needlessly complicated. After returning to the project after a few days, the clean, simple solution described previously was derived. This solution avoids any large multiplications or percentage calculation and instead finds the slope of the lines once, reusing that value for subsequent intensity calculations.

It can be noted that the green intensity value at the min-range temperature point is double the values of the red and blue intensities. This was originally a mistake, however, the softer white color that results from the extra green intensity was determined to be acceptable.

E. Temperature scaling and switching

Since vehicles sold in North American markets could be sold in the US, Canada, or Mexico, the ability to switch between English and Metric units was necessary. The temperature sensor device provides a convenient conversion factor to Celsius, so this is how the temperature data is read into the HCS12 code. If Fahrenheit temperature readings are desired, the formula below is used:

$$\mathrm{degF} = \frac{\mathrm{degC}\cdot 9}{5} + 32$$

Equation 2. Celsius to Fahrenheit Equation

A function was written to accept the temperature in degrees Celsius and return degrees Fahrenheit to simplify the main function of the project's code. A bit is toggled to represent either Celsius or Fahrenheit being desired. The default status is to display degrees Celsius. If the bit indicates Fahrenheit to be displayed, the Celsius value is passed to the conversion function, whose return value, degrees Fahrenheit, is displayed on the LCD.

The Port H push buttons are used to toggle this Celsius/Fahrenheit selector bit. Since the push buttons are not used for any other purposes in this project, a specific button was not selected- any push button press will trigger the change in units. To accomplish this, the Port H interrupt was set up to trigger on the falling edge, and the interrupt enabled on all Port H pins. The interrupt service routine simply uses *if* statements to flip the bit. If more time were available, a robust button debounce strategy would benefit the project. However, very rarely does a button bounce event occur, and there are no major effects from this.

III. Experimental Setup

The verification of this project was performed using the interfaces present on the Dragon Light board including the LCD display and RGB LED as well as external equipment to quickly change the temperature locally to the temperature sensor. The external equipment to quickly change the temperature will include Freeze Spray and a heat gun. In addition, the HCS12 built-in serial monitor was used for debugging during the development process. If functioning correctly, the display shows the temperature in the correct units as dictated by a button press, and the RGB LED changes from blue to red as temperature increases. For debugging, a volatile variable was used to change the temperature reading on the fly to test the RGB algorithm. Finally, the frequency output of the speaker rises as objects become closer and closer to the "bumper". Again, a volatile variable was used to simulate the pulse width from the ultrasonic sensor to ensure the frequency output is functioning as desired.

IV. Results

The project is fully functional as proposed. The temperature is read accurately in real-time and displayed on the LCD screen, as well as qualitatively on the RGB LED. The units of the LCD temperature display can be switched by pressing any of the Port H pushbuttons. Distance is sensed using the ultrasonic sensors and an audible tone alerts the user as objects become nearer to the sensor.

The objective of having function based code worked out well for debugging as well as reading the code. This type of coding will allow for any MCU and peripheral equipment or sensors to be used just by modifying what is in the functions. This would allow for enhanced reuse of the written code thereby reducing the amount of potential development time for the next application or to development with multiple external components as most vehicles will have different components for a different variety of options available in the vehicle lines.

Conclusions

Reading the temperature and displaying on the LCD all on-board the Dragon12 function properly with the equipment provided, however, there could be upgrades. The LM45 temperature sensor does not meet all automotive requirements which can reach -40°C to 125°C in some environments such as near the engine. Using a sensor that can reach this full range will give unlimited possibilities as to the use of the sensor and its results. Careful attention to how the temperature sensor is connected to the MCU is needed as an on-board connection means the MCU will be subjected to the same temperatures rather an external sensor may cost more for mounting and wire cost but the MCU cost can decrease as the MCU could be placed in a more stable climate on the vehicle.

The RGB LED is a crude solution for a graphical display, and could be replaced, as in some vehicle, by a snowflake display to indicate cold weather, and a thermometer to indicate hot weather. This would allow the driver to better understand when icy conditions may develop so their driving style can be adjusted accordingly. One benefit to the RGB LED system is that it is language independent. Rather than using Arabic numerals, English text, or any other language, it is a purely graphical indication which could be a benefit for auto companies attempting to reach global markets with minimal incremental cost.

The limitations of the Dragon 12 board are becoming clearer, however, if the microcontroller were to be used independently of the Dragon 12 circuit board, many more opportunities could be found. One of these limitations was the way the speaker was connected. If one wishes to control the RGB LED using PWM, the speaker cannot be controlled using PWM and must have a dedicated timer. Fortunately, the jumper on the board allows for configurability to at least work around this problem, but programming an audible alert would have been far easier, and more flexible if using a PWM resource was possible.

[1] Motorola Inc. "ATD_10B8C Block User Guide V02.12," pp. 10, June 2005.

[2] *Infineon technologies AG, "32-Bit Single-Chip Microcontroller," pp. 160.*

[3] Thierer, A. *"Removing Roadblocks to Intelligent Vehicles and Driverless Cars*, September 2014*,*