Microcontroller Based Advance Traffic Light System Using Voice Recorder for Ambulance Vehicle.

Zikora Agbapu	Sean Owens	Emily Bartlett
40%	30%	30%

Abstract

Traffic lights are very inefficient. They waste time, are quite unsafe, and cause problems for emergency vehicles. When traffic is jam-packed, and an emergency vehicle needs to get through an intersection, all cars in front need to try to squeeze out of the way risking collisions with other objects and vehicles. As long as the flow of traffic is steady, then this problem is not as relevant. When emergency vehicles come to a stop, they continue consuming gas, so the goal is to avoid having the emergency vehicles stop. We plan to fix these problems with our project. Our Project shall use a combination of sound sensors, and infrared sensors to allow for a better traffic flow, and quicker transit times for emergency vehicles.

1. Introduction

Most modern stoplights around today work without any external feedback, as the traffic lights work on timers. Traffic lights that do have feedback use inductive loops as the input. The problem with these loops is that the area that the loops cover is tiny. They usually only detect up to 2 cars. The concept we are presenting has a feedback system using infrared sensors to sense vehicles theoretically as far as it can see. Even with these feedbacks, there can still be some traffic problems. When emergency vehicles need to get through an intersection, they must wait for cars to move out of the way, then stop at the intersection to make sure all traffic has stopped to ensure a safe passage through the intersection. We plan to solve this problem as well by implementing a sound sensor to listen for approaching emergency vehicles [4]. Once one is detected all lights will be stopped except the direction that the emergency vehicle is approaching from. By implementing our system, emergency vehicles can save seconds at every intersection allowing a substantial decrease in total travel time for them while increasing safety. By saving time the emergency vehicles can also save gas [3].

Our project has two primary components; a sound sensor, and an infrared sensor. The sound sensor will be used to detect the sound of approaching emergency vehicles. We will do this by placing sensors on all roads of the intersection. For instance, at a four-way stop, there would be four sensors facing each direction. When an emergency vehicle is sensed by at least one sensor, it will compare the audio to each other sensor. The attached microcontroller is programmed to compare the volume of each sensor allowing it to determine which direction the vehicle is coming from. The microcontroller will continue listening and comparing to check to see if the car is getting closer or farther. If an emergency vehicle is detected as approaching the stoplight (via increased volume on all sensors) and its direction determined, it will halt all traffic except the traffic moving in the same direction as the emergency vehicle. This should help to avoid the emergency vehicle from running into stopped traffic, and for it to pass the intersection quicker. The second primary component of our project is an infrared camera. The infrared camera can detect the heat given off by vehicles, allowing it to see how many cars there are at the stoplight. Like the sound sensor, there will be four infrared sensors one watching each direction of traffic flow. The infrared sensors will show the number of vehicles heading in each direction. The microcontroller will take the number of vehicles and calculate the density of vehicles that are running parallel to each other. From this, it will find the highest density and permit travel along the two parallel roads while keeping the other two streets halted. This project can be cost effective for governments to deploy. Our plan will help emergency vehicles from needing to stop and accelerate. When a vehicle stops, it continues running and burning gas. A vehicle burns even more gas when it accelerates, so by keeping the traffic moving in front of an emergency vehicle, it reduces the need for it to slow down then accelerate it back up to speed. By lowering the need for these to happen at stop lights, it can lower the amount of gas used.

2. Methodology

In this project, we used four traffic lights in Proteus to simulate a four-way traffic intersection. Each of the three lights on the traffic light has an individual pin that is connected to pins in the ATMega32. These LEDs do not require a resistor because each LED has an internal resistor built-in. Instead of using infrared sensors to sense the presence of cars at the traffic light, we used simple switches that we can turn on and off according to the flow of traffic at the light. In real life, the flow of traffic is measured with either induction loops buried within the streets or infrared sensors attached to the traffic lights. These infrared sensors turn on when a car is within the sensor distance and are off when there are no objects within the range of the sensor. The infrared sensor can be replaced with an on/off switch for the sake of simulation.

For the basic operations, the traffic lights were based on a simple timer counter. For this system, we did not include the left-turn lights often seen in real-world busy traffic intersections to simplify the system. This means that the lights on opposite sides are always on at the same time while in normal operation. To achieve this on our simulation, we connected the pins of the parallel traffic lights to the same pins on the MCU so that they always have the same lights turned on at the same time. When one direction of flow (North-South) is on green, the other direction (East-West) is on red. One direction is on green for five seconds, while the other one is red. After five seconds, the moving direction light turns yellow for one second before turning red. While the moving direction is turning from yellow to red, the stationary position stays red, and both directions are red for second together. This single second of both lights on red is to ensure safety, in case there is any driver who tries to beat the red light. After that second, the light for the stationary direction turns green, and the whole process is repeated, although the previously stationary direction of traffic is now moving, and vice versa. The different stages are shown below.

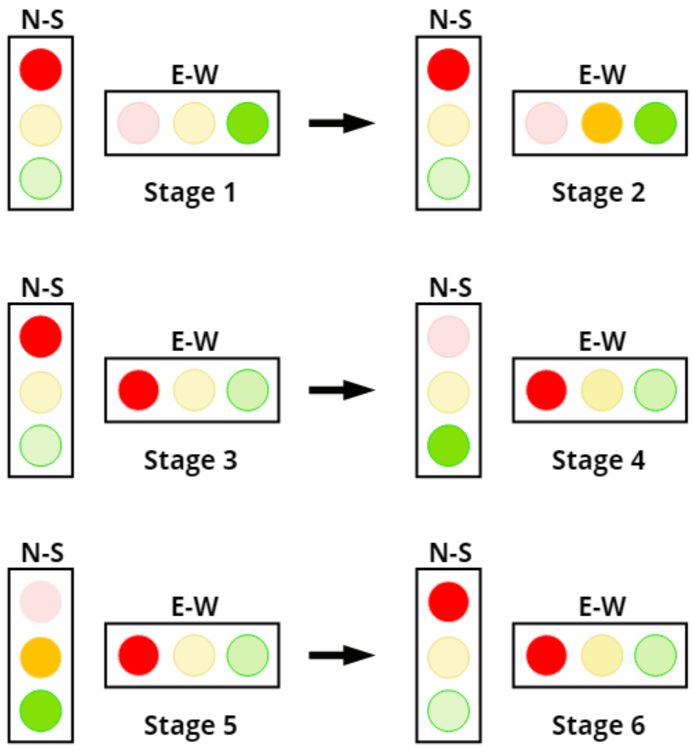

Figure 1: The different stages of traffic light operations

We also had to make our traffic light system adapt to changes in traffic flow and density to allow for the most efficient flow of traffic on the roads. As mentioned earlier, the infrared sensors are modeled with switches that represent the high flow of traffic when on, and a low flow of traffic when off. In the normal operation, the light stays on for five seconds before changing. When one of the buttons is pushed, the system goes into a high flow mode, which extends the length of the light in the high flow direction from five to eight seconds. This is achieved using polling to read the sensor output before changing that light to green. This is achieved using an if-else statement changes the value of the light duration variable from five to eight if the sensor switch is pushed, and vice-versa. When the flow has reduced to normal, the button is turned off, and the light duration returns to five seconds. In the event when there is a high flow of traffic in both directions, the duration of both lights is increased to allow more vehicles to pass through between each light change.

For the emergency vehicles, we used a sonar sensor to monitor for the presence of any ambulances, police cars or fire trucks. This sonar sensor placed in the middle of the intersection and senses the presence of any emergency vehicles in any of the four traffic lanes. In the simulation, this is modeled with a spring push switch that only stays on when it is pushed and is turned off otherwise. When this button is pushed, it triggers an external interrupt which turns all the lights red while any of the emergency vehicles are in the vicinity. These lights all stay red until the emergency vehicles have left the range of the sonic sensor. In the Proteus simulation, this is modeled by releasing the push button which then returns the whole system to the regular operating mode.

3. Circuit Design

The figure below shows the proposed design for the proposed traffic system. For simplicity, the wires have been hidden to make the image clearer and easier to understand. It contains the ATMega32 as the microcontroller, four 3-LED traffic lights at each leg of the intersection, four switches for the infrared sensors and a push button in the center to act as the ultrasonic sensor. In actual application, the ultrasonic sensor would be used and generate a signal when it finds a match to the ambulance sound in its vocal library.

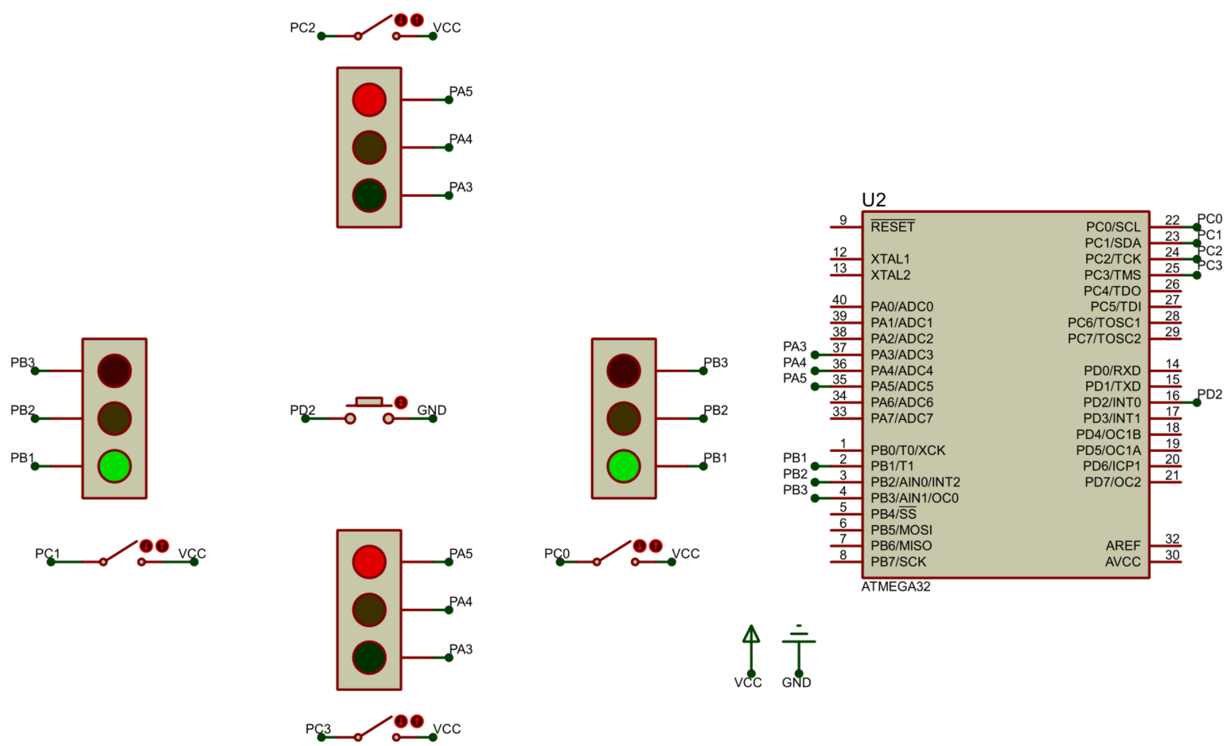

Figure 2: Schematic of the proposed traffic light design

As seen in the figure above, PORTA and PORTB are used as outputs to power the traffic lights. The corresponding LED pins on opposite lights are connected to the same ports in the microcontroller, so both traffic lights in parallel directions show the same color. PA3 to PA5 are connected to the North-South traffic lights, while PB1 to PB3 are connected to the East-West traffic lights. PORTC and PORTD are used for inputs, with PC0 to PC3 used as the inputs for the infrared sensor switches, while PD2 is used for the ultrasonic sensor buttons. This particular pin was chosen because it is also the input for INT0, the external interrupt that controls the emergency subroutine.

Code

```
/*******************************************************
Project :     Smart Traffic Light
Version :     v1.0
Date    :     11/23/2018
Author  :     Zikora Agbapu
Company :     Kennesaw State University
Comments:


Chip type               : ATmega32
Program type            : Application
AVR Core Clock frequency: 10.000000 MHz
Memory model            : Small
External RAM size       : 0
Data Stack size         : 256
*******************************************************/

#define F_CPU 1000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>

//Creating sleep function to control the delay for the lights
void sleep(uint8_t millisec)
{
while(millisec)
{
_delay_ms(1000);
millisec--;
}
}

int main(void)
{
//Connecting the Pins of the traffic lights to the pins of the MCU
DDRA |=1<<PA5;              //enable pin 35 red1
DDRA |=1<<PA4;              //enable pin 36 yellow1
DDRA |=1<<PA3;              //enable pin 37 green1
DDRB |=1<<PB3;              //enable pin 4 red2
DDRB |=1<<PB2;              //enable pin 3 yellow2
DDRB |=1<<PB1;              //enable pin 2 green2

//Setting PORTC as input for infrared sensor buttons
DDRC = 0x00;

//Defining variables
int time_ns;
int time_ew;
int time_yellow = 2;
int time_red = 1.5;
int time_normal = 5;
int time_extended = 7;


//Set up for the ultrasonic sensor button
DDRD = 0<<PD2;             // Set PD2 as input (Using for interrupt INT0)
```

```c
PORTD = 1<<PD2;             // Enable PD2 pull-up resistor

//Setting up the interrupts
GICR = 1<<INT0;                                  // Enable INT0
MCUCR = 1<<ISC01 | 0<<ISC00;       // Trigger INT0 on rising edge

sei();                      //Enable Global Interrupt

//Initializing the traffic lights as all off
 PORTA = 0x00;
 PORTB = 0x00;

while(1)
{
//Stage 1: Keeping Green on EW and Red on NS
       PORTA |= (1<<PA5);                 //on red1
       PORTB |= (1<<PB1);                 //on green2
       PORTB &= ~(1<<PB3);                //off red2
       PORTA &= ~(1<<PA4);                //off yellow1
       PORTB &= ~(1<<PB2);                //off yellow2
       PORTA &= ~(1<<PA3);                //off green1

       //Polling to find the value
       if(PINC & 0b00000011)
              time_ew = time_extended;
       else
              time_ew = time_normal;
       sleep(time_ew);

//Stage2: Turning EW to yellow & green
       PORTB |= (1<<PB2);                 //on yellow2
       sleep(time_yellow);

//Stage 3: Turning EW to red, while NS is red
       PORTB &= ~(1<<PB2);                //off yellow2
       PORTB &= ~(1<<PB1);                //off green2
       PORTB |= (1<<PB3);                 //on red2
       sleep(time_red);

//Stage 4: Turning NS to green
       PORTA &= ~(1<<PA5);                //off red1
       PORTA |= (1<<PA3);                 //on green1

       //Polling to find the value
       if(PINC & 0b00001100)
       time_ns = time_extended;
       else
       time_ns = time_normal;
       sleep(time_ns);

//Stage 5: Turning NS to green & yellow
       PORTA |= (1<<PA4);                 //on yellow1
       sleep(time_yellow);

//Stage 6: Turning NS red while EW is red
       PORTA &= ~(1<<PA3);                //off green1
       PORTA &= ~(1<<PA4);                //off yellow1
       PORTA |= (1<<PA5);                 //on red1
```

```
        sleep(time_red);

//Stage 1: Keeping Green on NS and Red on EW
        PORTA |= (1<<PA5);                  //on red1
        PORTB |= (1<<PB1);                  //on green2
        PORTB &= ~(1<<PB3);                 //off red2
        PORTA &= ~(1<<PA4);                 //off yellow1
        PORTB &= ~(1<<PB2);                 //off yellow2
        PORTA &= ~(1<<PA3);                 //off green1

}
}

ISR(INT0_vect)
{
                int time_emgcy = 3;
                PORTA |= (1<<PA5); //on red1 pin28
                PORTB |= (1<<PB3); //on red2
                PORTB &= ~(1<<PB1); //off green2
                PORTA &= ~(1<<PA4); //off yellow1
                PORTB &= ~(1<<PB2); //off yellow2
                PORTA &= ~(1<<PA3); //off green1
                sleep(time_emgcy);
        }
```

4. Economic Impact of Embedded Systems

Embedded systems are control systems worked into the individual mechanisms. They tend to be designed for specific tasks and allow for programming input to control mechanical and electrical components. This allows for an expanded market for the development of these systems as they become the more popular choice over general systems. The demand for their use has dramatically increased as we advance technologically. This also creates more jobs to design, create, and maintain these systems. Embedded systems are used in everything from small smart devices to ATM's, planes, cars, 3-D printers and more. Their increased use has allowed the overall cost to manufacture these systems to drop and further increase their accessibility. This concept also applies to the market for the microprocessors and other components used in the systems. The demand and market for them have also drastically improved and dropped the overall production prices. According to recent reports from the IDC analyst firm, intelligent systems including embedded systems will likely annually see a compounded growth rate of 7.2% between 2015 and 2020 and overall revenue that exceeds 2.2 Trillion dollars in the year 2020 [1]. The source of this growth is thought to come from the ease of access to the internet and a more significant overall technological presence. The general rise of IoT devices has given the general area of embedded systems renewed interest as they have given many more market participants the reason to develop more and newer systems that go into embedded applications. This general rise in economic interest has meant that many new microdevices and other accompanying parts with which new types of devices are being made. Soon they will be present in every home, car, and piece of wearable tech during this "Smart Tech" boom. The overall price of computing power is decreasing, and so new users are appearing at every price range in product development.

In our chosen case, redesigning the traffic light systems away from being based on microcontrollers and microprocessors helps remove the system's current limitations. Currently, they are unable to modify the program based on real-time input and data collection. Moving away from the predetermined timing of traffic lights allows for a decrease in the wait times and overall gas usage. This would make traffic flow more efficient through techniques called "Intelligent traffic light control." This technique makes use of Sensor Networks and other embedded technology in order to make decisions based on the analysis of total traffic present. [2]. This would allow for the optimization of the flow of traffic and increases the capacity capabilities of the road while still decreasing and preventing traffic congestion. Lowering the congestion also lowers the amount of gas used on a day to day basis which helps to save money. In an economic sense, travel is considered to be a "Cost" in that people must spend time and money to make their journeys, a reduction in those aspects would be considered an economic benefit to the population. As mentioned before, optimization of traffic flow, road capacity, and wait times would altogether reduce the impact of travel costs on the daily commuter. Another benefit would be a reduction in the carbon emissions into the environment as a result of the reduction in time that cars are left sitting at the traffic lights. A future improvement for the system could also include a priority level for vehicles like Buses and Taxis in order to reward and further maximize the reduction of carbon emissions, fuel use, travel time, and overall traffic on the road.

5. Conclusion

This report takes an in-depth look into the importance of embedded systems and its numerous applications in everyday life. In particular, it focuses on traffic lights and considers various solutions on how to make them more efficient and user-friendly. The solution offered in this paper includes the use of microcontrollers to control the timing of the light but also uses an array of infrared and ultrasonic sensors to measure the activity at a given intersection and adjust the timing of the traffic light according to the flow, density, and presence of emergency vehicles at the intersection. To simplify the simulation, the infrared sensors for detecting the presence of cars at the light are replaced with switches, while the ultrasonic sensor is replaced with a push button. This is just one of the many possible applications of embedded systems, which bring the physical and the digital world together in harmony.

References

[1] Redline Group. (2018). *The Technological Advancements in Embedded Systems | Redline Group Recruitment News and Blogs*. [online] Available at: https://www.redlinegroup.com/insights/the-technological-advancements-in-embedded-systems-74615914431 [Accessed 22 Nov. 2018].

[2] Ieeexplore.ieee.org. (2018). *Design of Intelligent Traffic Light Controller Using Embedded System - IEEE Conference Publication*. [online] Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5395443 [Accessed 23 Nov. 2018].

[3] X. Cai, "Design of Traffic Light System Based on Proteus Simulation," *Proceedings of the 3rd International Conference on Mechanical Engineering and Intelligent Systems (ICMEIS 2015)*, 2015.

[4] N. Sahana, K. S. Bhat, M. Aishwarya, L. Kanumilli, and J. M. Bhanu, "Dynamic Traffic Light Control for Intelligent Mobility of Emergency Vehicles," *Bonfring International Journal of Software Engineering and Soft Computing*, vol. 6, no. Special Issue, pp. 167–170, 2016.