IOT BASED ONLINE TRAFFIC CONGESTION MONITORING AND MANAGEMENT SYSTEM

KamatchiSundari.V[1], Kowsika.S[2], Monisha.K[3]

[1] *Professor, Electronics and communication engineering, PSVPEC, TN, India*
[2] *Student, Electronics and communication engineering, PSVPEC, TN, India*
[3] *Student, Electronics and communication engineering, PSVPEC, TN, India*

ABSTRACT

In today's world, traffic jams during rush hours is one of the major concerns. During rush hours, emergency vehicles like Ambulances, Police cars and Fire Brigade trucks get stuck in jams. Due to this, these emergency vehicles are not able to reach their destinations in time, resulting into a loss of human lives. In order to overcome these situations green wave management system is implemented which is used to provide clearance to any emergency vehicle by turning all the red lights to green on the path of the emergency vehicle, and hence providing a complete green wave to the desired vehicle. It is an autonomous system which will help in the identification of emergency vehicle. It is a novel system which can be used to implement the concept of the green wave. The basic idea used for traffic management here is to control congestion by using a decision making algorithm which determines how the traffic light operates based on the information collected from RFID devices. The objective of this project is to propose an effective scheme for road traffic management which is fully automated and foolproof considering the rate of ever growing traffic in urban.

Keyword - *RFID, GSM, IDE, and LCD….*

1. INTRODUCTION

Green wave management system is to pass the emergency vehicles smoothly. India is a second most popular country in the world and is a fast growing economy. It has seen with the terrible congestion problems. Hence wireless networks have been developed. Technologies like Zig bee, Radio frequency Identification (RFID) and global systems (GSM) provide cost effective solutions. RFID is the wireless technology which is used to provide information between RFID tag and RFID reader

1.1 BASICS OF EMBEDDED SYSTEM

An Embedded system is one that has hardware with software embedded in it as one of its components. It is a dedicated computer based system for an application or product it may be either an independent system or large system. As its software usually embeds in Read Only Memory it does not need secondary memory as in a computer system .Embedded is used in many fields such as automobiles, telephones, appliances and peripherals. These are called Embedded system while some embedded system are very sophisticated many have minimum required for memory and program length with no operating system and low software complexity. Typical input and output devices include switches, relay, solenoids, LED small or custom LCD displays, radio frequency devices and sensors for data such as temperature, humidity, light level etc. Embedded system usually have no keyboard, screen, disks, printer or other recognizable input/output devices of a personal computer and may lack human interaction devices of any kind. An embedded system has two main components.

- Hardware
- Main application software.

The application software may perform concurrently the series of task or multiple tasks. It supervises the application network and provides as per scheduling and do the context-switch between the various processes. RTOS defines the way system works. It organizes access to a resource in sequence of the series of tasks of the system. It

schedules their work in and execution by following the plan to control the latencies and to meet the deadlines. It supervises the application software and provides a mechanism to let the processor run a process.

1.2 TYPES OF EMBEDDED SYSTEM

Embedded systems can be classified into three types based on performance of the microcontroller.

- Small scale embedded system
- Medium scale embedded systems
- Sophisticated embedded systems

1.2.1 Small scale embedded systems

These types of embedded systems are designed with a single 8 or 16-bit micro controller that may even be activated by a battery. For developing embedded software for small scale embedded systems, the main programming tools are an editor, assembler, cross assembler and integrated development environment (IDE).

1.2.2 Medium scale embedded systems

These types of embedded systems design with a single or 16 or 32 bit microcontroller, RISCs or DSPs. These types of embedded systems have both hardware and software complexities. For developing embedded software for medium scale embedded systems, the main programming tools are C, C++, and JAVA, Visual C++, and RTOS, debugger, source code engineering tool, simulator and IDE.

1.2.3 Sophisticated embedded systems

These types of embedded systems have enormous hardware and software complexities that may need ASIPs, IPs, PLAs, scalable or configurable processors. They are used for cutting-edge applications that need hardware and software Co-design and components which have to assemble in the final system.

1.3 EXISTING SYSTEM

In existing method, automatic traffic management based upon vehicle type is difficult. There is no wireless technology is available for monitoring.

1.4 CAUSES OF TRAFFIC CONGESTION

These include accidents, harsh weather conditions, and road constructions. An accident can cause a road blockage or slow down traffic flow as drivers try to understand what is happening. Likewise, bad weather conditions can cause drivers to slow down as they worry for their safety. Also, road construction can cause reduction of lanes thereby forcing drivers to crowd the open lanes. Consequently, traffic congestion can be highly increased by these factors, but because of their randomness, not much can be done to prevent such incidents.

2. ARDUINO MEGA CONTROLLER

The Mega 2560 is a microcontroller board based on the ATMEGA2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. The Mega 2560 board can be programmed with the Arduino Software (IDE). For details, see the reference and tutorials. The ATmega2560 on the Mega 2560 comes preprogrammed with a boot loader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

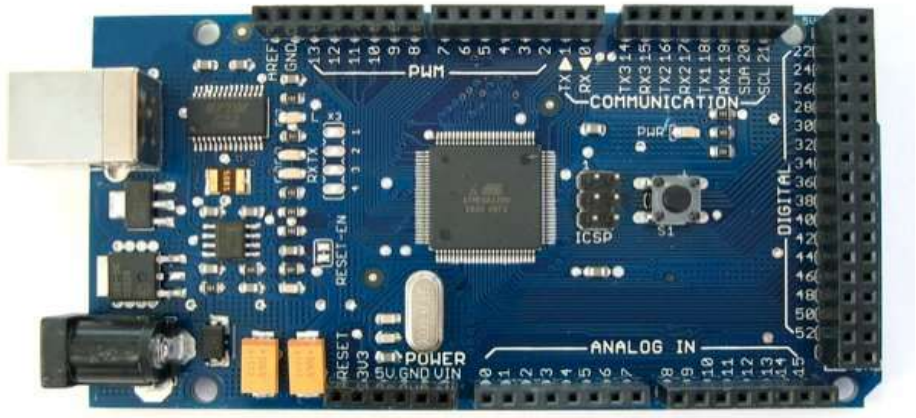

Fig -1: Arduino mega controller 2560

2.1 TECHNICAL SPECIFICATION

You can also bypass the boot loader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP or similar; see these instructions for details. The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available in the Arduino repository. The ATmega16U2/8U2 is loaded with a DFU boot loader, which can be activated by On Rev1 boards connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.On Rev2 or later boards. You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU boot loader). See this user-contributed tutorial for more information.

Table -1: Arduino mega controller requirements

Operating Voltage	5V
Input Voltage	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 Ma
Flash Memory	256 KB of which 8 KB used by boot loader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g

2.2 Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8KB is used for the boot loader), 8 KB of SRAM and 4KB of EEPROM (which can be read and written with the EEPROM library).

a) Input and Output

Serial 0(RX) and 1(TX), serial 1 19(RX) and 18(TX), Serial 2 17(RX) and 16(TX), Serial 3 15(RX) and 14(TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the AT mega 16u2USB-to-TTL serial chip. External interrupts.2 (interrupts 0), 3(interrupt 1), 18 (interrupt 5), 19(interrupt 4), 20(interrupt 3), and 21(interrupt 2). The pins can be configured to trigger an interrupt on a low level, a rising or falling edge, or a change in level. The attach Interrupt () function for details.PWM: 2 to 13 and 44 to 46. Provide 8-bit PWM output with the analog Write () function.SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication using the SPI library. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Arduino /Genuino Uno and the old Duemilanove and Diecimila Arduino boards.LED, 13.There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, its off.TWI: 20 (SDA) and 21 (SCL). Support TWI communication using the Wire library. Note that these pins are not in the same location as the TWI pins on the old Duemilanoveor Diecimila Arduinoboards. See also the mapping Arduino Mega 2560 PIN diagram. The Mega 2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analog reference () function. There are a couple of other pins on the board AREF, Reference voltage for the

analog inputs. Used with analog Reference (). Reset, Bring this line LOW to reset N the microcontroller. Typically used to add a reset button to shield's which block the one on the board.

b) Lcd (Liquid Crystal Display)

LCD screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred overseen and other multi segment LEDs.The reasons being, LCDs are economical, easily programmable, have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on. A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data. The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD. LCDThe most commonly used Character based LCDs are based on Hitachi's HD44780 controller or other which are compatible with HD44580. In this tutorial, we will discuss about character based LCDs, their interfacing with various microcontrollers, various interfaces (8-bit/4-bit), programming, special stuff and tricks you can do with these simple looking LCDs which can give a new look to your application. The liquid crystal library allows you to control LCD displays that are compatible with the Hitachi HD44780 driver. Many of them out there usually tell by 16 pin interface. LCD has 19 different functions in the liquid crystal library available for to use. Functions do things like change the position of the next, move text across the screen, or make the display turn on or off.

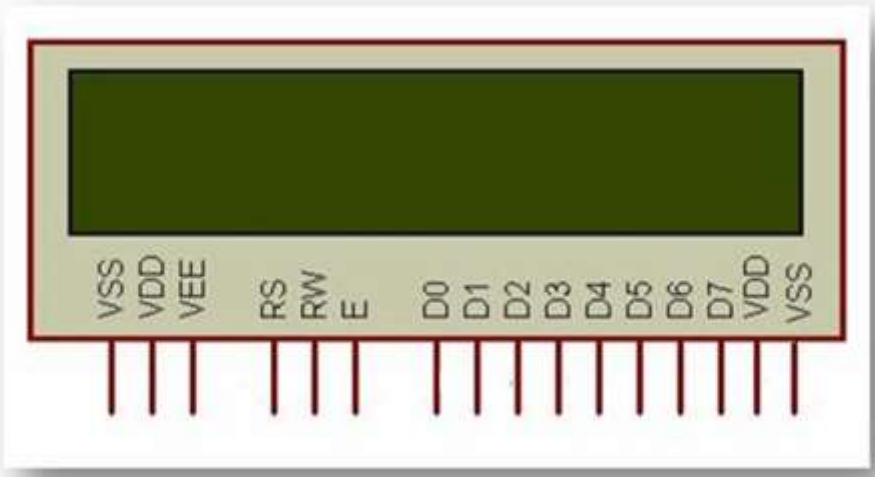

Fig -2: Liquid crystal display pin diagram

3. RFID Reader and Tag

An RFID reader is a device that is used to interrogate an RFID tag. The reader has an antenna that emits radio waves; the tag responds by sending back its data. An RFID tag is a microchip combined with an antenna in a compact package; the packaging is structured to allow the RFID tag to be attached to an object to be tracked. "RFID" stands for Radio Frequency Identification. The tag's antenna picks up signals from an RFID reader or scanner and then returns the signal, usually with some additional data (like a unique serial number or other customized information).A passive tag is an RFID tag that does not contain a battery; the power is supplied by the reader. When radio waves from the reader are encountered by a passive RFID tag, the coiled antenna within the tag forms a magnetic field. The tag draws power from it, energizing the circuits in the tag.

Fig- 3: RFID reader

The tag then sends the information encoded in the tag's memory. The RX and TX pins of RFID reader connected to TX and Rx pins of 8051 Microcontroller respectively. Then the reader senses the data from the Tag and transmits the sensed data to microcontroller via serial port. The EM-18 RFID Reader module operating at 125 kHz is an inexpensive solution for your RFID based application. The Reader module comes with an on-chip antenna and can be powered up with a 5V power supply. Power-up the module and connect the transmit pin of the module to receive pin of your microcontroller. Show your card within the reading distance and the card number is thrown at the output. Optionally the module can be configured for also a Wigand output.

3.1 EXPERIMENTAL SETUP

The project is a simple representation of traffic light controller and hence no other extra components are used. The circuit consists of many LED's (6 as a matter of fact). We need two LED's of Red, Yellow and Green colors at each intersection. All the LEDs are connected to the Arduino UNO's digital I/O pins through respective current limiting resistors of 1KOHMS. All the connections are made as per the circuit diagram. In the practical implementation of the project, we did not use the current limiting resistors as the current from each digital I/O pin of the Arduino UNO is only 20mA. This small current will not burn the LED. But it is advised to use the current limiting resistors of at least 220 OHMS in series with each LED. Also note that Arduino UNO in this project acts as source of current for all the LED i.e. it provides the necessary current to turn ON the LED. Hence, a reliable power supply (like a DC adapter) to power the Arduino UNO must be used.

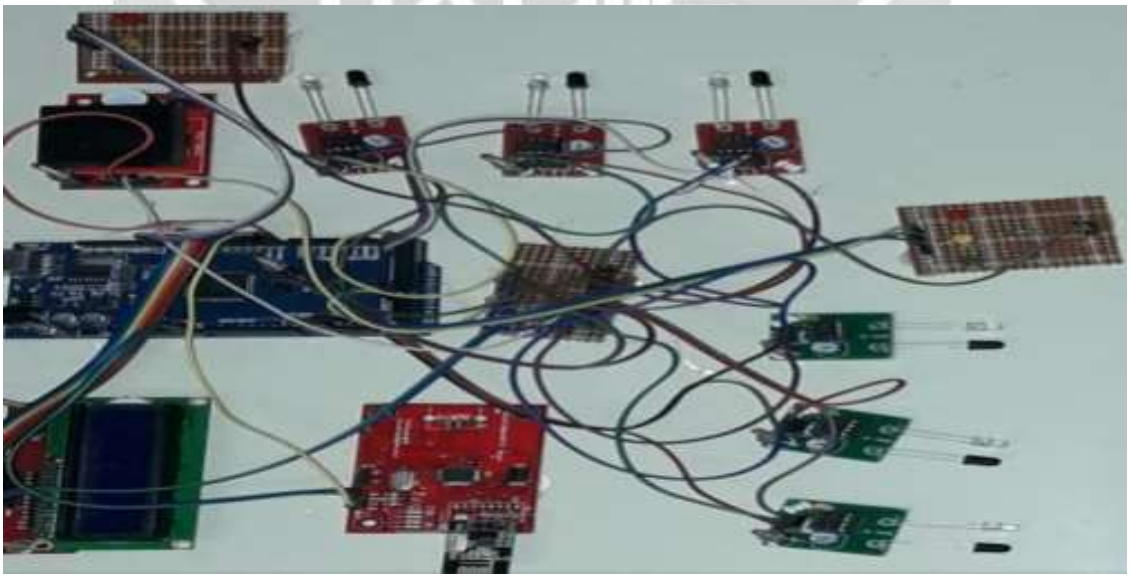

Fig -4: Snap chart of Experimental setup

4. CONCLUSIONS

As the use of vehicles is increasing day so the problem of traffic is arising. The proposed system which has the certain advantages that reduces the human efforts comparing to the existing system. The problem that is faced due to more traffic is called as traffic congestion. It will have specific identification for emergency vehicles so easy to clear the traffic. It requires very less human intervention. It can easily control the traffic congestion and dynamic time schedule is work out. It clears the traffic for emergency vehicles and protects many lives from risk. And finally, it tracks the exact location of the vehicle. Hence common people can know the status of the traffic at any junction. Traffic Monitoring signal timing has been developed by using multiple features of hardware components in IOT platform for efficient utilizing allocating varying time to all traffic signal according to available vehicles count in road path. Work presents a real time traffic information collection and monitoring system to solve the problem of real time monitoring and controlling road vehicles. The proposed system employs key technologies of internet of things, RF transmitter and receiver to supervise traffic information.

5. REFERENCES

[1]. Karen Rose, Scott Eldridge, Lyman Chapin, "The Internet of Things: An Overview Understanding the Issues and Challenges of a More Connected World", The Internet Society (ISOC), 2015.

[2]. Martín Serrano, PayamBarnaghi, Francois Carrez Philippe Cousin, OvidiuVermesan, PeteraFriess, "Internet of Things Semantic Interoperability: Research Challenges, BestPractices, Recommendations and Next Steps", European research cluster on the internet of things, IERC,2015.

[3].Martin Serrano, Insight Centre for Data Analytics, Ireland ,Omar Elloumi, Alcatel Lucent, France, Paul Murdock, Landis+Gyr, Switzerland, "ALLIANCE FOR INTERNET OF THINGS INNOVATION, Semantic Interoperability" , Release 2.0, AIOTI WG03 – loT Standardisation,2015.

[4].Dr. OvidiuVermesan SINTEF, Norway, Dr. Peter FriessEU, Belgium, "Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems", river publishers' series in communications, 2013.