International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 06 Issue 13
December 2019

Implementing Intelligent Traffic Control System for Congestion Control Ambulance Clearances And Stolen Vehicle Detection.

Kambati Sukanya[1], K. Sudha Rani[2], S. Mahaboob Basha[3]
[1]P.G. Scholar, [2]Assistant Professor, [3]Head of the Department
[1,2,3] Branch: Embedded systems(ECE)
[1,2,3] Geethanjali College of Engineering &Technology, Kurnool.
Email: [1]sukanyareddy439@gmail.com , [2]kommireddy.sudha@gmail.com

Abstract
This Work presents an intelligent traffic control system to pass emergency vehicles smoothly. Each individual vehicle is equipped with special RFID tag(placed at a strategic location), which makes it impossible to remove or destroy. We use RFID reader, NSK EDK-125–TTL and PIC16F877A system-on-chip to read the RFID tags attached to the vehicle. It counts number of vehicles that passes on a particular path during a specified duration. It also determines the network congestion, and hence the green light duration for that path. If the RFID-tag-read belongs to the stolen vehicle, then a message is sent using GSM SIM300 to the police control room. Also, when an ambulance is approaching the junction, it will communicate to the traffic controller in the junction to turn on the green light. This module uses ZigBee modules on CC2500 and PIC16F877A system-on-chip for wireless communications between the ambulance and traffic controller. The prototype was tested under different combinations of inputs in our wireless communication laboratory and experimental results were found as expected.

Keywords: - Vehicles, Radiofrequency identification, Zigbee ,GSM, Junctions, Traffic control, Microcontrollers

INTRODUCTION

Everybody is growing rich each day and owning a car, to every available family has turned out to be an obsession. Hardly few people realize that all this and more increases the stress put on traffic. Thickly congested roads are a daily sight in most metropolitan cities. Commuters, including children and employees, spend hours each day stuck in long and never ending traffic jams. The pity is that very often, one can find the green light shining on empty roads, wasting the precious time of many other/ people.

The worst scenario is that of an ambulance, with a person battling death, stuck in a traffic jam and the red light shining on that road. This paper, taking this scenario as our motivation, is thus putting its humble, little effort to help this situation by making traffic lights sufficiently intelligent that it can detect which side of a junction consists of more traffic density and thus deserves a green signal. Eventually, it would help relieve suffocating streets of heavy bottlenecks, saving people their precious time. The paper proposes to implement this intelligent traffic light control system using RFID, GSM and Zigbee technologies.

It also helps detect vehicular movement on that road. Thus, the roads are conceptually divide into two distinct zones, the high density and the low density zones to help

the traffic lights detect which road to let go. In order to create an environment that is friendly to emergency vehicles, they would be requested to have an RFID chip beneath their vehicles for detecting them. The category of emergency vehicles would include ambulances, police vehicles, vehicles of ministers and other nobles. The system would implement an RFID reader below the surface of the road, tuned to detect emergency vehicles, i.e., it will be able to detect emergency vehicles moving towards the traffic lights.

As a result, in case of ambulances, the side with the emergency vehicle will obtain their rightful share of LCD display & turn on buzzer, thus the people may give way with free of traffic. Through this, we can be sure that even if the ambulance doesn't reach the line – of – sight of the RFID reader, the lights will be turned LCD & Buzzer for sure. This is because an RFID reader doesn't require the chip to come to its line-of-sight to understand the presence of an RFID chip. When implemented near hospitals wherein the movement of ambulances is quite heavy, emergency vehicles can be sure that they can reach their destination without much ado.

LITERATURE SURVEY:

Mr. Raja Ghosh & his team speaks of "Auto Density Sensing Traffic Control System using At89s52", it is proposed that dynamic time based coordination scheme were the green signal time of the traffic light is assigned based on the present condition of traffic by using IR sensor.

K. Vidhya, A. Bazila Banu speaks of "Density Based Traffic Signal System", the image captured in traffic signal is proposed & converted into grayscale image then threshold is in order to image.

The Intelligent Traffic Light and Density Control using IR Sensors and Microcontroller by First A Ms Promila Sinhmar proposes the use of IR sensors and microcontrollers, just as the name suggests, in an attempt to reduce the number of vehicles waiting for their chance to move out. The system gives equal preferences to all the vehicles plying, ignoring all the emergency vehicles.

INTRODUCTION TO EMBEDDED SYSTEMS

Application Areas

Nearly 99 per cent of the processors manufactured end up in embedded systems. The embedded system market is one of the highest growth areas as these systems are used in very market segment- consumer electronics, office automation, industrial automation, biomedical engineering, wireless communication, data communication, telecommunications, transportation, military and so on.

Consumer appliances At home we use a number of embedded systems which include digital camera, digital diary, DVD player, electronic toys, microwave oven, remote controls for TV and air-conditioner, VCO player, video game consoles, video recorders etc. Today's high-tech car has about 20 embedded systems for transmission control, engine spark control, air-conditioning, navigation etc. Even wristwatches are now becoming embedded systems. The palmtops are powerful embedded systems using which we can carry out many general-purpose tasks such as playing games and word processing.

Office automation: The office automation products using em embedded systems are copying machine, fax machine, key telephone, modem, printer, scanner etc.

Industrial automation: Today a lot of industries use embedded systems for process control. These include pharmaceutical, cement, sugar, oil exploration, nuclear energy, electricity generation and transmission. The embedded

systems for industrial use are designed to carry out specific tasks such as monitoring the temperature, pressure, humidity, voltage, current etc., and then take appropriate action based on the monitored levels to control other devices or to send information to a centralized monitoring station. In hazardous industrial environment, where human presence has to be avoided, robots are used, which are programmed to do specific jobs. The robots are now becoming very powerful and carry out many interesting and complicated tasks such as hardware assembly.

Medical electronics: Almost every medical equipment in the hospital is an embedded system. These equipments include diagnostic aids such as ECG, EEG, blood pressure measuring devices, X-ray scanners; equipment used in blood analysis, radiation, colonscopy, endoscopy etc. Developments in medical electronics have paved way for more accurate diagnosis of diseases.

Computer networking: Computer networking products such as bridges, routers, Integrated Services Digital Networks (ISDN), Asynchronous Transfer Mode (ATM), X.25 and frame relay switches are embedded systems which implement the necessary data communication protocols. For example, a router interconnects two networks. The two networks may be running different protocol stacks. The router's function is to obtain the data packets from incoming pores, analyze the packets and send them towards the destination after doing necessary protocol conversion. Most networking equipments, other than the end systems (desktop computers) we use to access the networks, are embedded systems

Telecommunications: In the field of telecommunications, the embedded systems can be categorized as subscriber terminals and network equipment. The subscriber terminals such as key telephones, ISDN phones, terminal adapters, web cameras are embedded systems. The network equipment includes multiplexers, multiple access systems, Packet Assemblers Dissemblers (PADs), satel1ite modems etc. IP phone, IP gateway, IP gatekeeper etc. are the latest embedded systems that provide very low-cost voice communication over the Internet.

Wireless technologies: Advances in mobile communications are paving way for many interesting applications using embedded systems. The mobile phone is one of the marvels of the last decade of the 20th century. It is a very powerful embedded system that provides voice communication while we are on the move. The Personal Digital Assistants and the palmtops can now be used to access multimedia services over the Internet. Mobile communication infrastructure such as base station controllers, mobile switching centers are also powerful embedded systems.

Insemination: Testing and measurement are the fundamental requirements in all scientific and engineering activities. The measuring equipment we use in laboratories to measure parameters such as weight, temperature, pressure, humidity, voltage, current etc. are all embedded systems. Test equipment such as oscilloscope, spectrum analyzer, logic analyzer, protocol analyzer, radio communication test set etc. are embedded systems built around powerful processors. Thank to miniaturization, the test and measuring equipment are now becoming portable facilitating easy testing and measurement in the field by field-personnel.

Security: Security of persons and information has always been a major issue. We need to protect our homes and offices; and also the information we transmit and

store. Developing embedded systems for security applications is one of the most lucrative businesses nowadays. Security devices at homes, offices, airports etc. for authentication and verification are embedded systems. Encryption devices are nearly 99 per cent of the processors that are manufactured end up in~ embedded systems. Embedded systems find applications in.every industrial segment-consumer electronics, transportation, avionics, biomedical engineering, manufacturing, process control and industrial automation, data communication, telecommunication, defense, security etc. Used to encrypt the data/voice being transmitted on communication links such as telephone lines. Biometric systems using fingerprint and face recognition are now being extensively used for user authentication in banking applications as well as for access control in high security buildings.

Finance: Financial dealing through cash and cheques are now slowly paving way for transactions using smart cards and ATM (Automatic Teller Machine, also expanded as Any Time Money) machines. Smart card, of the size of a credit card, has a small micro-controller and memory; and it interacts with the smart card reader! ATM machine and acts as an electronic wallet. Smart card technology has the capability of ushering in a cashless society. Well, the list goes on. It is no exaggeration to say that eyes wherever we go, we can see, or at least feel, the work of an embedded system.

HARDWARE IMPLEMENTATION OF THE PROJECT

This chapter briefly explains about the Hardware Implementation of the project. It discusses the design and working of the design with the help of block diagram and circuit diagram and explanation of circuit diagram in detail. It explains the features, timer programming, serial communication, interrupts of atmega328 microcontroller. It also explains the various modules used in this project.

Project Design

The implementation of the project design can be divided in two parts.

- Hardware implementation
- Firmware implementation

Hardware implementation deals in drawing the schematic on the plane paper according to the application, testing the schematic design over the breadboard using the various IC's to find if the design meets the objective, carrying out the PCB layout of the schematic tested on breadboard, finally preparing the board and testing the designed hardware.

The project design and principle are explained in this chapter using the block diagram and circuit diagram. The block diagram discusses about the required components of the design and working condition is explained using circuit diagram and system wiring diagram.

INTRODUCTION TO MICROCONTROLLER

Based on the Processor side Embedded Systems is mainly divided into 3 types

1. **Micro Processor : -** are for general purpose eg: our personal computer
2. **Micro Controller:-** are for specific applications, because of cheaper cost we will go for these
3. **DSP (Digital Signal Processor):-** are for high and sensitive application purpose

MICROCONTROLLER VERSUS MICROPROCESSOR

A system designer using a general-purpose microprocessor such as the Pentium or the 68040 must add RAM, ROM, I/O ports, and

timers externally to make them functional. Although the addition of external RAM, ROM, and I/O ports makes these systems bulkier and much more expensive, they have the advantage of versatility such that the designer can decide on the amount of RAM, ROM and I/O ports needed to fit the task at hand.

A Microcontroller has a CPU (a microprocessor) in addition to a fixed amount of RAM, ROM, I/O ports, and a timer all on a single chip. In other words, the processor, the RAM, ROM, I/O ports and the timer are all embedded together on one chip; therefore, the designer cannot add any external memory, I/O ports, or timer to it. The fixed amount of on-chip ROM, RAM, and number of I/O ports in Microcontrollers makes them ideal for many applications in which cost and space are critical.

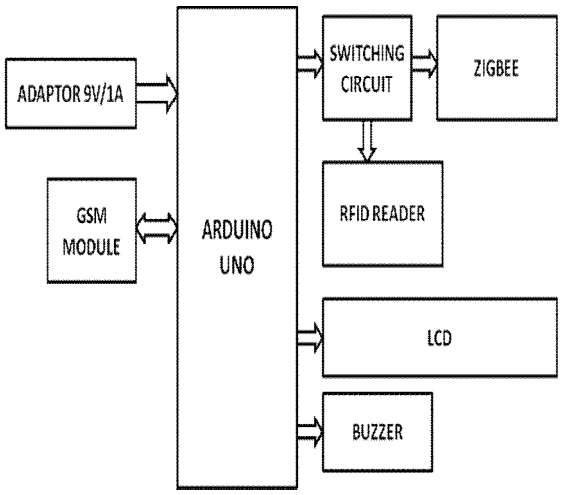
Microprocessor vs. Microcontroller	
Microprocessor	**Microcontroller**
CPU is stand alone RAM, RAM, I/O, timer are separate	CPU,RAM,ROM,I/O and timer are all on a single chip
Designer can decide on the amount of ROM, RAM and I/O ports.	Fix amount of on chip ROM,RAM, I/O Ports.
Expansive, Versatility	For applications in which cost, power and space are critical
General purpose	Single purpose

Table: Microprocessor vs. Microcontroller

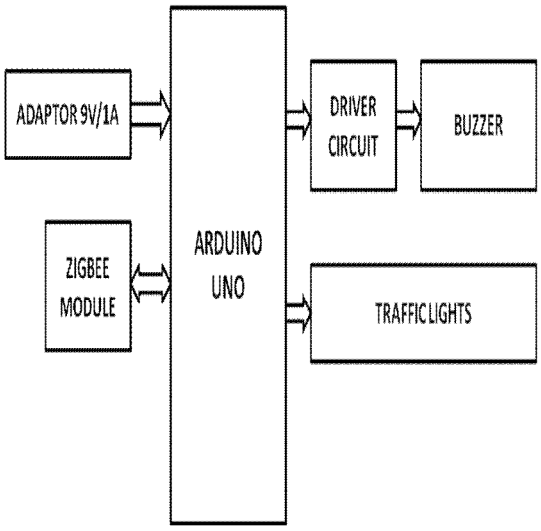
Block Diagram of the Project

The block diagram of the design is as shown in Fig 3.1. It consists of power supply unit, Arduino and other modules.

TXSIDE



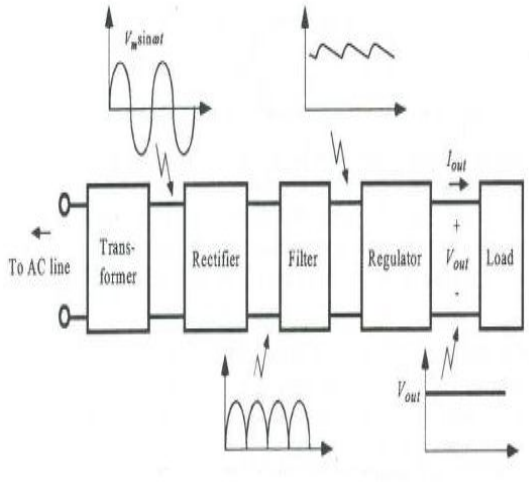
RX SIDE



Power Supply:

The input to the circuit is applied from the regulated power supply. The a.c. input i.e., 230V from the mains supply is step down by the transformer to 12V and is fed to a rectifier. The output obtained from the rectifier is a pulsating d.c voltage. So in order to get a pure d.c voltage, the output

voltage from the rectifier is fed to a filter to remove any a.c components present even after rectification. Now, this voltage is given to a voltage regulator to obtain a pure constant dc voltage.



Components of a regulated power supply

Transformer:

Usually, DC voltages are required to operate various electronic equipment and these voltages are 5V, 9V or 12V. But these voltages cannot be obtained directly. Thus the a.c input available at the mains supply i.e., 230V is to be brought down to the required voltage level. This is done by a transformer. Thus, a step down transformer is employed to decrease the voltage to a required level.

Rectifier:

The output from the transformer is fed to the rectifier. It converts A.C. into pulsating D.C. The rectifier may be a half wave or a full wave rectifier. In this project, a bridge rectifier is used because of its merits like good stability and full wave rectification.

Filter:

Capacitive filter is used in this project. It removes the ripples from the output of rectifier and smoothens the D.C. Output received from this filter is constant until the mains voltage and load is maintained constant. However, if either of the two is varied, D.C. voltage received at this point changes. Therefore a regulator is applied at the output stage.

Voltage regulator:

As the name itself implies, it regulates the input applied to it. A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. In this project, power supply of 5V and 12V are required. In order to obtain these voltage levels, 7805 and 7812 voltage regulators are to be used. The first number 78 represents positive supply and the numbers 05, 12 represent the required output voltage levels.

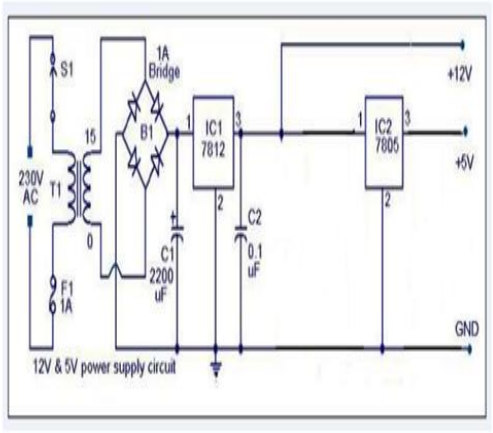


Fig: Power supply circuit diagram

ARDUINO UNO

Arduino is used for building different types of electronic circuits easily using of both a physical programmable circuit board usually microcontroller and piece of code running on computer with USB connection between the computer and Arduino. Programming language used in Arduino is just a simplified version of C++ that can easily replace thousands of wires with words.

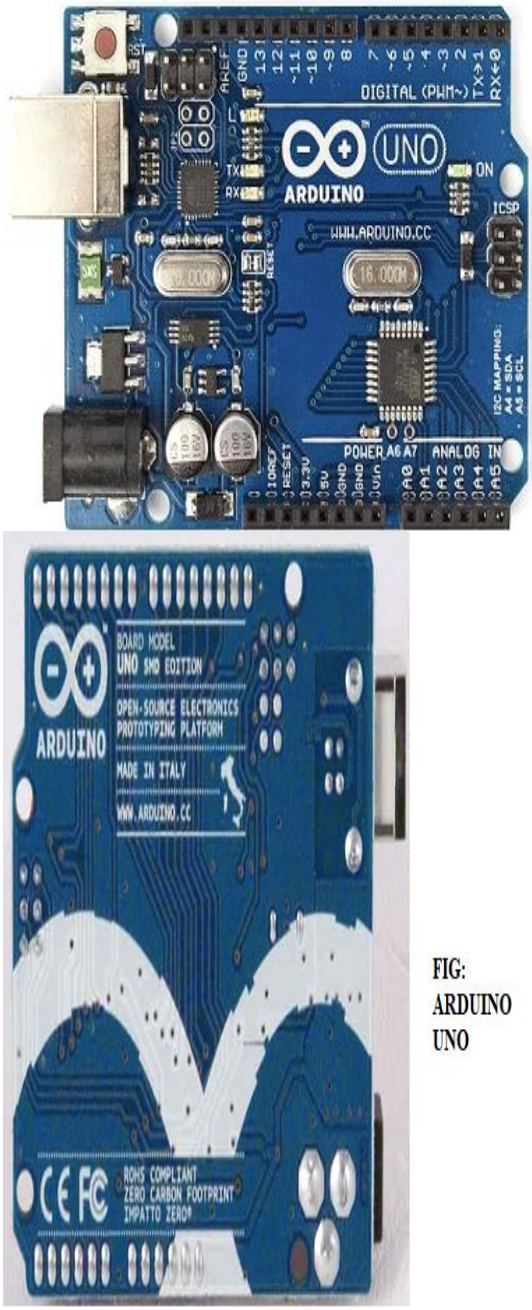


FIG: ARDUINO UNO

ARDUINO UNO-R3 PHYSICAL COMPONENTS

ATMEGA328P-PU microcontroller

The most important element in Arduino Uno R3 is ATMEGA328P-PU is an 8-bit Microcontroller with flash memory reach to 32k bytes.

Features

1. High Performance, Low Power Atmel®AVR® 8-Bit Microcontroller Family
- Advanced RISC Architecture
– 131 Powerful Instructions
– Most Single Clock Cycle Execution
– 32 x 8 General Purpose Working Registers
– Fully Static Operation
– Up to 20 MIPS Throughput at 20MHz
– On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
– 32KBytes of In-System Self-Programmable Flash program
2. Memory
– 1KBytes EEPROM
– 2KBytes Internal SRAM
– Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
– Data Retention: 20 years at 85°C/100 years at 25°C(1)
– Optional Boot Code Section with Independent Lock Bits
- In-System Programming by On-chip Boot Program
- True Read-While-Write Operation
– Programming Lock for Software Security
- Atmel® QTouch® Library Support
– Capacitive Touch Buttons, Sliders and Wheels
– QTouch and QMatrix® Acquisition
– Up to 64 sense channels
3. Atmel-42735B-ATmega328/P_Datasheet_Complete-11/2016
- Peripheral Features
– Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
– One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
– Real Time Counter with Separate Oscillator
– Six PWM Channels
– 8-channel 10-bit ADC in TQFP and QFN/MLF package

- Temperature Measurement
– 6-channel 10-bit ADC in PDIP Package
- Temperature Measurement
– Two Master/Slave SPI Serial Interface
– One Programmable Serial USART
– One Byte-oriented 2-wire Serial Interface (Philips I2C compatible)
– Programmable Watchdog Timer with Separate On-chip Oscillator
– One On-chip Analog Comparator
– Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
– Power-on Reset and Programmable Brown-out Detection
– Internal Calibrated Oscillator
– External and Internal Interrupt Sources
– Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and

4. Extended Standby

- I/O and Packages
– 23 Programmable I/O Lines
– 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
– 1.8 - 5.5V
- Temperature Range:
– -40°C to 105°C
- Speed Grade:
– 0 - 4MHz @ 1.8 - 5.5V
– 0 - 10MHz @ 2.7 - 5.5V
– 0 - 20MHz @ 4.5 - 5.5V
- Power Consumption at 1MHz, 1.8V, 25°C
– Active Mode: 0.2mA
– Power-down Mode: 0.1µA
– Power-save Mode: 0.75µA (Including 32kHz RTC)

OTHER ARDUINO UNO R3 PARTS

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode (), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 k Ohms.

In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analogWrite() function.
- **SPI**: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- **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, it's off. The Uno has 6 analog inputs, labeled A0 through A5, 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 the analogReference() function.

Additionally, some pins have specialized functionality:

- **TWI: A4 or SDA pin and A5 or SCL pin.** Support TWI communication using the Wire library. There are a couple of other pins on the board:
- **AREF:** Reference voltage for the analog inputs. Used with analogReference().
- **Reset:** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of

SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a
Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

Programming

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno from the Tools > Board menu (according to the microcontroller on your board). For details, see the reference and tutorials.

The ATmega328 on the Arduino Uno comes preburned with a bootloader 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).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The
ATmega16 U2/8U2 is loaded with a DFU bootloader, 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: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode. 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 bootloader).

Description

The Atmel AVR® core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in a single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.
Atmel offers the QTouch® library for embedding capacitive touch buttons, sliders and wheels functionality into AVR microcontrollers. The patented charge-transfer signal acquisition offers robust sensing and includes fully debounced reporting of touch keys and includes Adjacent Key Suppression® (AKS™) technology for unambiguous detection of key events. The easy-to-use QTouch Suite

toolchain allows you to explore, develop and debug your own touch applications.

The device is manufactured using Atmel's high density non-volatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed In-System through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core.

The Boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega328/P is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications. The ATmega328/P is supported with a full suite of program and system development tools including: C Compilers, Macro Assemblers, Program Debugger/Simulators, In-Circuit Emulators, and Evaluation kits.

Features	ATmega328/P
Pin Count	28/32
Flash (Bytes)	32K
SRAM (Bytes)	2K
EEPROM (Bytes)	1K
General Purpose I/O Lines	23
SPI	2
TWI (I^2C)	1
USART	1
ADC	10-bit 15kSPS
ADC Channels	8
8-bit Timer/Counters	2
16-bit Timer/Counters	1

Table: Arduino Configurations

Configuration Summary

Block Diagram

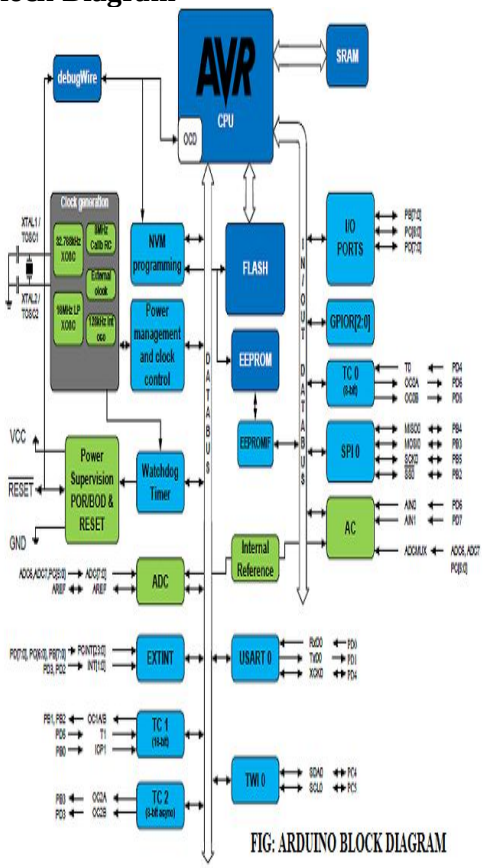


FIG: ARDUINO BLOCK DIAGRAM

FIRMWARE IMPLEMENTATION OF THE PROJECT DESIGN

This chapter briefly explains about the firmware implementation of the project. The required software tools are discussed in the following sections.

Software Tool Required

Arduino 1.0.6 software tools used to program microcontroller. The working of software tool is explained below in detail.

PROGRAMMING MICROCONTROLLER

A compiler for a high level language helps to reduce production time. To program the Arduino UNO microcontroller the Arduino is used. The programming is done strictly in the embedded C language. Arduino is a suite of executable, open source software development tools for the

microcontrollers hosted on the Windows platform.

Arduino is a tool for making computers that can sense and control more of the physical world than your desktop computer. It's an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board.

One of the difficulties of programming microcontrollers is the limited amount of resources the programmer has to deal with. In personal computers resources such as RAM and processing speed are basically limitless when compared to microcontrollers. In contrast, the code on microcontrollers should be as low on resources as possible

ABOUT ARDUINO COMPILER

GET AN ARDUINO BOARD AND USB CABLE

You also need a standard USB cable (A plug to B plug): the kind you would connect to a USB printer, for example. (For the Arduino Nano, you'll need an A to Mini-B cable instead.)

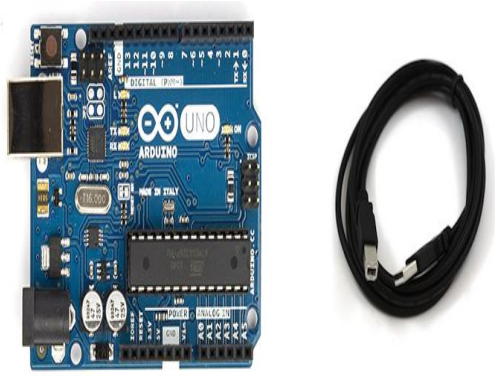


FIG: ARDUINO BOARD AND USB CABLE

CONNECT THE BOARD

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either the USB connection to the computer or an external power supply. If you're using an Arduino Diecimila, you'll need to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it's on the two pins closest to the USB port. Connect the Arduino board to your computer using the USB cable. The green power LED (labelled PWR) should go on.

FIG: OPENING THE ARDUINO WINDOW

Open the blink example

Open the LED blink example sketch: File > Examples > 1.Basics > Blink.

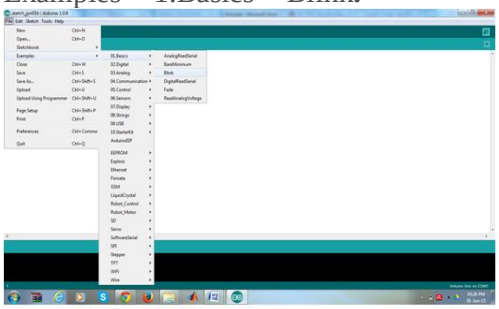


FIG: OPENING BLINK EXAMPLE

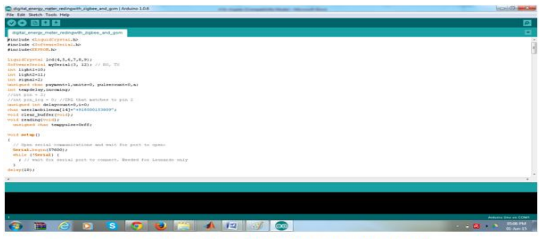


FIG: SOURCE CODE WRITTEN IN ARDUINO COMPILER

Select your board

You'll need to select the entry in the Tools > Board menu that corresponds to your Arduino.

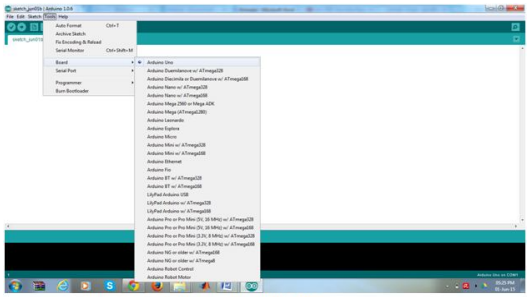


FIG: SELECTING AN ARDUINO UNO

WRITING SKETCHES

Software written using Arduino are called sketches. These sketches are written in the text editor. Sketches are saved with the file extension .ino. It has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino environment including complete error messages and other information. The bottom righthand corner of the window displays the current board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

NB: Versions of the IDE prior to 1.0 saved sketches with the extension .pde. It is possible to open these files with version 1.0, you will be prompted to save the sketch with the .ino extension on save.

Verify	Checks your code for errors.
Upload	Compiles your code and uploads it to the Arduino I/O board. See uploading below for details. Note: If you are using an external programmer, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer"
New	Creates a new sketch.
Open	Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window. Note: due to a bug in Java, this menu doesn't scroll; if you need to open a sketch late in the list, use the File | Sketch book menu instead.
Save	Saves your sketch.
SerialMonitor	Opens the serial monitor.

Additional commands are found within the five menus: File, Edit, Sketch, Tools, Help. The menus are context sensitive which means only those items relevant to the work currently being carried out are available.

SELECT YOUR SERIAL PORT

Select the serial device of the Arduino board from the Tools | Serial Port menu. This is likely to be COM3 or higher (COM1and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu; the entry that disappears should be the Arduino board. Reconnect the board and select that serial port.

UPLOAD THE PROGRAM

Before uploading your sketch, you need to select the correct items from the **Tools > Board** and **Tools > Serial Port** menus. The boards are described below. On the Mac, the serial port is probably something like **/dev/tty.usbmodem241** On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, you look for USB serial device in the ports section of the Windows Device Manager. On Linux, it should be **/dev/ttyUSB0**,**/dev/ttyUSB1** or similar.

Once you've selected the correct serial port and board, press the upload button in the toolbar or select the **Upload** item from the **File** menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino environment will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino **bootloader**, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The

bootloader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The bootloader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

Now, simply click the "Upload" button in the environment. Wait a few seconds - you should see the RX and TX leds on the board flashing. If the upload is successful, the message "Done uploading." will appear in the status bar. (*Note:* If you have an Arduino Mini, NG, or other board, you'll need to physically present the reset button on the board immediately before pressing the upload button.)

FIG: COMPILATION UNDER PROCESS

A few seconds after the upload finishes, you should see the pin 13 (L) LED on the board start to blink (in orange). If it does, congratulations! You've gotten Arduino up-and-running.

Working procedure:

Design of Intelligent traffic control system using Arduino is implemented. The traffic signals are controlled using GSM or RFID technology. The signals can be changed according to the user requirement. This is done mainly to save the human lives who are in ambulances in heavy traffics.

This design consists of GSM and RFID to change the route, buzzer for audio indication and LCD to display the messages and Zigbee module to transmit the data to the receiver side. Four RFID cards will be provided for 4 different routes. If any of the cards is recognized by the RFID reader, this information will be transmitted to the receiver through the Zigbee module. At the receiver side, the route information will be received and green signal will be activated in that particular route.

Similarly GSM is used to activate the green signal in a particular route. Whenever the user sends the message through the modem, this information will be transmitted through the Zigbee module and receiver receives it and activates the green signal according to the message received. GSM and RFID are interfaced with Arduino through the switching circuit. Zigbee is used for data transmission and GSM, RFID are used for the route selection.

Future Scope:

- In this project, there are conditions of Congestion Control, Ambulance Clearance and Stolen vehicle detection.
- It will perform as per presence of either of these conditions.
- It also can be extended to the level that when there is a presence of Ambulance in a particular junction wants to go either of the ways in case of halt condition so here the control system should already know the movement and willing direction of ambulance vehicle in case of emergency purposes.

Reference:-

1. G. Varaprasad, R. S. D. Wahidabanu, "Flexible routing algorithm for vehicular area networks", Proc. IEEE Conf. Intell. Transp. Syst. Telecommun., pp. 30-38, 2010.
2. B. P. Gokulan, D. Srinivasan, "Distributed geometric fuzzy multiagent urban traffic signal control", IEEE

Trans. Intell. Transp. Syst., vol. 11, no. 3, pp. 714-727, Sep. 2010.
3. K. Sridharamurthy, A. P. Govinda, J. D. Gopal, G. Varaprasad, "Violation detection method for vehicular ad hoc networking", Security Commun. Netw., [online] Available: http://onlinelibrary.wiley.com/doi/10.1002/sec.427/abstract.
4. M. Abdoos, N. Mozayani, A. L. C. Bazzan, "Traffic light control in non-stationary environments based on multi agent Q-learning", Proc. 14th Int. IEEE Conf. Intell. Transp. Syst., pp. 580-1585, Oct. 2011.
5. "ZigBee Specifications, ZigBee Alliance IEEE Standard 802.15.4k2013", 2014, [online] Available: http://www.zigbee.org/Specifications.aspx.
6. Traffic Congestion in Bangalore—A Rising Concern., 2013, [online] Available: http://www.commonfloor.com/guide/traffic-congestion-in-bangalore-a-rising-concern-27238.html.
7. A. K. Mittal, D. Bhandari, "A novel approach to implement green wave system and detection of stolen vehicles", Proc. IEEE 3rd Int. Adv. Comput., pp. 1055-1059, Feb. 2013.
8. S. Sharma, A. Pithora, G. Gupta, M. Goel, M. Sinha, "Traffic light priority control for emergency vehicle using RFID", Int. J. Innov. Eng. Technol., vol. 2, no. 2, pp. 363-366, 2013.
9. R. Hegde, R. R. Sali, M. S. Indira, "RFID and GPS based automatic lane clearance system for ambulance", Int. J. Adv. Elect. Electron. Eng., vol. 2, no. 3, pp. 102-107, 2013.
10. P. Sood, Bangalore Traffic Police-Preparing for the Future., 2011, [online] Available: http://www.intranse.in/its1/sites/default/files/D1-S2-.
11. Traffic Management Centre., 2014, [online] Available: http://www.bangaloretrafficpolice.gov.in/index.php?option=com_content&view=article&id=87&btp=87.
12. G. Varaprasad, "High stable power aware multicast algorithm for mobile ad hoc networks", IEEE Sensors J., vol. 13, no. 5, pp. 1442-1446, May 2013.
13. Traffic Solution., 2013, [online] Available: http://phys.org/news/2013–05-physics-green-city-traffic-smoothly.html.