ABSTRACT

Intelligent road sign congestion problem is a phenomena which contributed huge impact to the transportation system in country. This causes many problems especially when there are abnormal weather conditions cases at Intelligent road sign system intersections which are always busy with many vehicles. A Intelligent road sign system controller system is designed in order to solve these problems. This system was designed to be operated when it received signal from abnormal weather conditions vehicles based on radio frequency (RF) transmission and used the Programmable Integrated Circuit Arduino microcontroller to change the sequence back to the normal sequence before the abnormal weather conditions mode was triggered. This system will reduce accidents which often happen at the Intelligent road sign system intersections because of other vehicle had to huddle for given a special route to abnormal weather conditions vehicle. As the result, this project successful analyzing and implementing the wireless communication; the radio frequency (RF) transmission in the Intelligent road sign system control system for abnormal weather conditions vehicles. The prototype of this project is using the frequency of 434 MHz and function with the sequence mode of Intelligent road sign system when abnormal weather conditions vehicles passing by an intersection and changing the sequence back to the normal sequence before the abnormal weather conditions mode was triggered. In future, this prototype system can be improved by controlling the real Intelligent road sign situation, in fact improving present Intelligent road sign system system technology

INTRODUCTION

Many countries in the world are facing the problem at Intelligent road sign system intersection that causes accident between abnormal weather conditions vehicle and other public vehicle. The Intelligent road sign control system in Malaysia specifically has not been equipped with appropriate method when abnormal weather conditions case occurs. This will cause the abnormal weather conditions vehicles such as ambulances difficult to reach the destination on time because of the Intelligent road sign congestion. Moreover, the situation is getting worse when abnormal weather conditions vehicles have to wait for other vehicles to give way at intersections with Intelligent road sign systems. This causes a delay of time and may affect the abnormal weather conditions case. Besides, the collisions with other vehicles from other direction might occur at intersections when abnormal weather conditions vehicles had to override the red Intelligent road sign systems. All these difficulties faced by abnormal weather conditions vehicles can be avoided using this Intelligent road sign system control system based on radio frequency. Due to the problem, literature review for related issue prior to undertaking research project is decisive. The literature review will provide information on the technology available and methodologies used by other research counterparts around the world on this topic.

Wireless sensor network

Wireless sensor networks (WSN), sometimes called wireless sensor and actuator networks (WSAN),[1][2] are spatially distributed autonomous sensors to *monitor* physical or environmental conditions, such as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to a main location. The more modern networks are bi-directional, also enabling *control* of sensor activity. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance; today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, and so on.

The WSN is built of "nodes" – from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. Each such sensor network node has typically several parts: a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting. A sensor node might vary in size from that of a shoebox down to the size of a grain of dust, although functioning "motes" of genuine microscopic dimensions have yet to be created. The cost of sensor nodes is similarly variable, ranging from a few to hundreds of dollars, depending on the complexity of the individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth. The topology of the WSNs can vary from a simple star network to an advanced multi-hop wireless mesh network. The propagation technique between the hops of the network can be routing or flooding.

Various government policies have encouraged the combination of hospital accident and abnormal weather conditions departments into centralized units, responsible for large geographic areas. While this provides an improved quality of medical care, travel time to a crash scene is often compromised for those not near these centralized locations (*4*). According to Brown, there is a positive association between ambulance delay and the ratio of fatal to serious injuries. This study found an increased mortality rate in counties that had a low population density, further suggesting a link between elevated response time and prognosis (*5*). Numerous other studies have also demonstrated the relationship between decreases in response time and corresponding decreases in mortality (*1*).

The ability to accurately detect a vehicle"s location and its status is the main goal of automobile trajectory monitoring systems.& also The high demand of automobiles has also increased the Intelligent road sign hazards and the road accidents. This is because of the lack of best abnormal weather conditions facilities available in our country This design is a system which can detect accidents in significantly less time and sends the basic information to first aid center within a few seconds covering geographical coordinates, the time and angle in which a vehicle accident had occurred. This alert message is sent to the rescue team in a short time, which will help in saving the valuable lives. These systems are implemented using several hybrid techniques that include: wireless communication, geographical positioning and embedded applications. *.1 Automatic Accident Detection and Reporting System using GSM and GPS:* Intelligent road sign has become an important event in the national interest now-a-days. We see that a lot of life spoils in every accident because of typically long response time to access the appropriate care that may be available if informed in-time. Application of our project can significantly shorten the response time of accident. This is

a platform for abnormal weather conditions rescue which will operate optimally in order to reduce the golden time of arrival of rescuers in case of road accidents, when every microsecond counts. Our project aims to present a technology automatically detecting the accident and a hardware tracking device based on GSM/GPS technology informing at the occurrence of accident with sufficient details like exact location and time at which accident happened. This project will establish a communication between the control station and the unit installed in vehicles. Vehicles will have GPS/GSM enabled tracking modules and will be tracked in real time using cellular networks. The software embedded in the microcontroller will control the various operations of the device by monitoring waveform from the vibration sensor. In case of accident the device will send an alert message along with location data from GPS module to control station using GSM network. It is a comprehensive and effective solution to the poor rescue response in case of accident. The accident reporting can automatically find a Intelligent road sign accident, search for the spot and then send the basic information to the rescue agency covering geographical coordinates and the time and circumstances in which a Intelligent road sign accident took place. At the server end, a control function will extract relevant data and store it in a database, to which accident information from prototypes will be polled in real time. Our system combines advanced hardware design and sophisticated control technology into a compact, reliable package.

\

CHAPTER 2

LITERATURE SURVEY

[1] Ganiyu R. A., Arulogun O. T., "Development Of A Microcontroller-Based Intelligent road sign System System For Road Intersection Control", International Journal of Science & Technology Research, Volume 3, ISSUE 5, May 2014.

A Intelligent road sign system control system invented by Carl J. Obeck [5] consists of two-way communication between abnormal weather conditions vehicles approaching a busy intersection with one or more Intelligent road sign systems. The system temporarily pre-empt the sequence of the Intelligent road sign system and provides the most effective method of routing the vehicle through the intersection while redirecting general Intelligent road sign. As part of the invention, the Intelligent road sign system control system will inform the abnormal weather conditions vehicle which it has received the transmitted signal. The stored preset Intelligent road sign patterns may in one representation is responsive to manual intervention from a dispatching centre or to time-of-day conditions. The Intelligent road sign system control apparatus may be operated under control of data or voice transmitted from the abnormal weather conditions vehicle's regular two-way voice communications system to a central control station.

2] Mir Roomi Rahil, Rajesh Mahind, Saurabh Chavan, Tanumay Dhar, "GLCD-Touchpad Based Restaurant Ordering & Automatic Serving System", International Journal of Recent Technology and Engineering (IJRTE), 2013

According to all these papers, a convenient wireless communication between abnormal weather conditions vehicles and the Intelligent road sign system is by using RF. The prototype of this project is using the radio frequency of 434 MHz compared to the range of about 3 kHz to 300 GHz of frequency which have been reserved for the RF theoretically. There are three objectives to be achieved in this project. First is to analyze and implement wireless communication; Radio Frequency (RF) transmission system in Intelligent road sign system control system for abnormal weather conditions vehicles. Second is to design a Intelligent road sign system sequence for abnormal weather conditions mode when receive signal from abnormal weather conditions vehicles. Last objective is to change the sequence back to the normal sequence before the abnormal weather conditions mode was triggered. This project has contributed in implementing the wireless communication by using the radio frequency (RF) transmission of 434 MHZ in the Intelligent road sign system control system for abnormal weather conditions vehicles.

[3] M. A.A. Parkhi, Mr. A.A. Peshattiwar, Mr. K.G. Pande "Intelligent Intelligent road sign System Using Vehicle Density". Yeshwantrao Chavan College of Engg., Nagpur. International Journal of Electrical and Electronic Engoneers, 2016.

The Intelligent road sign system system designed by Levi L. Rose [1] used only for abnormal weather conditions vehicle. Sensor is used to transmit signal that has been installed in every abnormal weather conditions vehicle to the receiver which has been placed at every Intelligent road sign system intersection. When abnormal weather conditions vehicle reach at the Intelligent road sign system intersection, the signal code will be sent information of frequency modulation to the receiver. The receiver demodulates the received code and the red Intelligent road sign system will trigger at all the junctions. Thus, abnormal weather conditions vehicle will have special route from other vehicle to reach the destination. The Intelligent road sign system system designed by M. R. Smith et al [2] provided early warning of the approaching an abnormal weather conditions vehicle to find a way out from Intelligent road sign congestion and lead the abnormal weather conditions vehicle to the destination. The abnormal weather conditions vehicle also may take control of Intelligent road sign system at an intersection. A transmitter placed on an abnormal weather conditions vehicle transmits a signal to the receivers positioned at the Intelligent road sign systems whenever it is on abnormal weather conditions mode. The received signal is then processed by a master controller which in turn pre-empts the sequence of the Intelligent road sign system to control the Intelligent road sign flow at the intersection which taken by the abnormal weather conditions vehicle. The master controller also provides an output which display signs to indicate that there is an abnormal weather conditions vehicle to the other road users from other direction at the Intelligent road sign system intersection. Additionally, the display system indicates whether the abnormal weather conditions vehicle has passed through the intersection or not.

[4] Dinesh Rotake, Prof. Swapnil Karmore "Intelligent Intelligent road sign Signal Control System Using Embedded System". G.H Raisoni College of Engineering, Nagpur. Innovative Systems Design and Engineering, 2012

Ms Promila Sinhmar, 2012 [4] propose multiple Intelligent road sign system control and monitoring system. The system is based on microcontroller. The system contains IR sensors are mounted on the sides of roads respectively. The IR sensors network sense the vehicle passed through it. Microcontroller controls the IR system and counts the number of vehicles passing on the road. The vehicle count is stored in microcontroller memory. Based on a different vehicle count, the microcontroller takes decision and updates the Intelligent road sign system delays as a result. Administrator sitting on the computer can command system 40 (microcontroller) to down-load recorded data, update system delays, erase memory, etc. Thus administrator of a central station computer can access Intelligent road sign conditions on any approachable Intelligent road sign systems and nearby roads to reduce Intelligent road sign congestions to an extent.

[5] Payal Gupta,Dhananjay V.Gadre, Tarun Kumar Rawat, "Real Time Intelligent road sign System Control System(Hardware and Software Implementation)". International Journal of Electronic and Electrical Engineering, 2014.

Shilpa S. Chavan, 2009 [5] introduced Intelligent Intelligent road sign System Controller, which consist of infrared sensor mounted on the road to detect the vehicles, this acts as an input to the ITLC unit. This input signal indicates the length of vehicles on each road. The controller generates output signals for Red, Green and Orange Signal and monitors their timings, taking into consideration the length of vehicles on each road. The same information is transmitted to the mobile user which will request for congestion status. If a vehicle driver at junction sends SMS on GSM mobile phone to ITLC unit, the driver will get a message indicting congestion status of the road. The microcontroller that used is AT89c51.

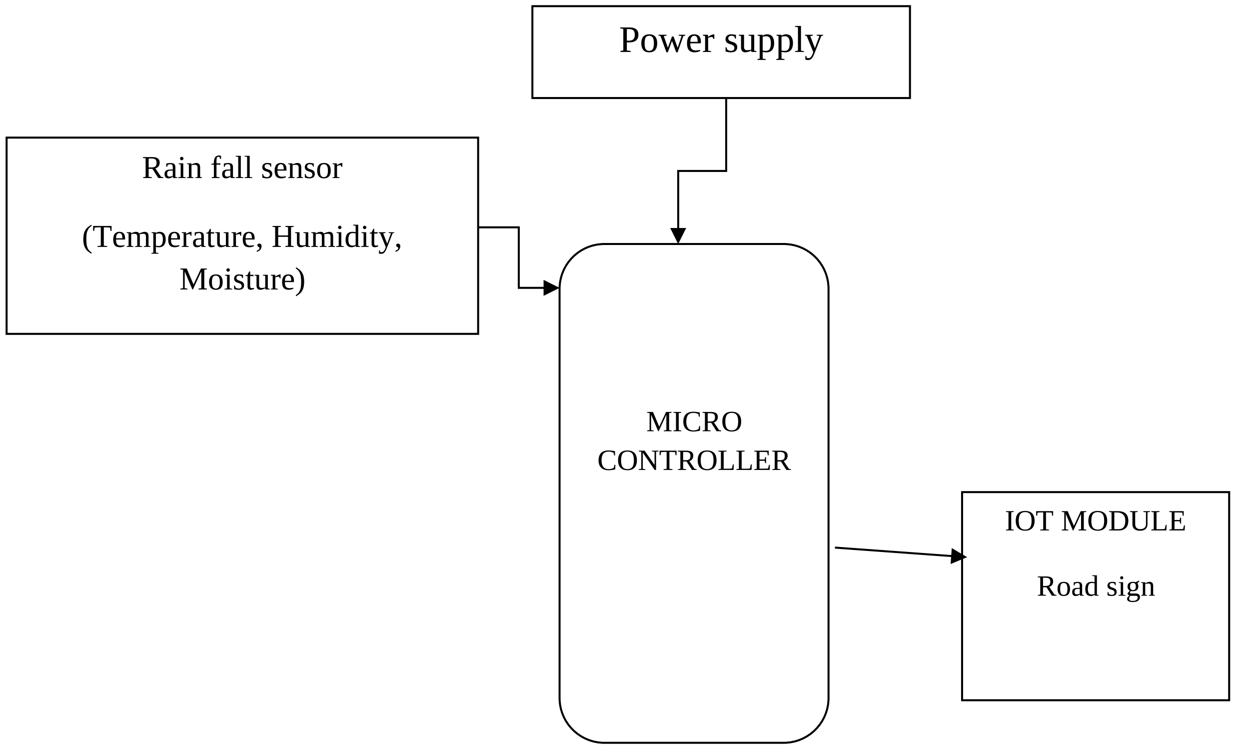
EXISTING SYSTEM

In existing system the Intelligent road sign congestion problem and provided an abnormal weather conditions path for the abnormal weather conditions vehicle where the radio transmitter and antenna placed on the abnormal weather conditions vehicle. The radio will transmit the signal to the other vehicle that nearby. The radio receiver had been placed at four junction Intelligent road sign system will receive the abnormal weather conditions signal from abnormal weather conditions vehicle that passed by the junction. The first signal code contains a frequency for abnormal weather conditions vehicle while the second signal code contains a frequency for other vehicle. The transmitted signals provide miscellaneous Intelligent road sign system pole in normal condition or abnormal weather conditions. When the receiver received the signal from abnormal weather conditions vehicle transmitter, Intelligent road sign system system for abnormal weather conditions vehicle will be activated. As part of the invention, the Intelligent road sign system control system will inform the abnormal weather conditions vehicle which it has received the transmitted signal. The stored preset Intelligent road sign patterns may in one representation is responsive to manual intervention from a dispatching centre or to time-of-day conditions. The Intelligent road sign system control apparatus may be operated under control of data or voice transmitted from the abnormal weather conditions vehicle's regular two-way voice communications system to a central control station.

PROPOSED SYSTEM

This system was designed to be operated when it received signal from abnormal weather conditions vehicles based on radio frequency (RF) transmission and used the Programmable Integrated Circuit Arduino microcontroller to change the sequence back to the normal sequence before the abnormal weather conditions mode was triggered. This system will reduce accidents which often happen at the Intelligent road sign system intersections because of other vehicle had to huddle for given a special route to abnormal weather conditions vehicle. As the result, this project successful analyzing and implementing the wireless communication; the radio frequency (RF) transmission in the Intelligent road sign system control system for abnormal weather conditions vehicles.

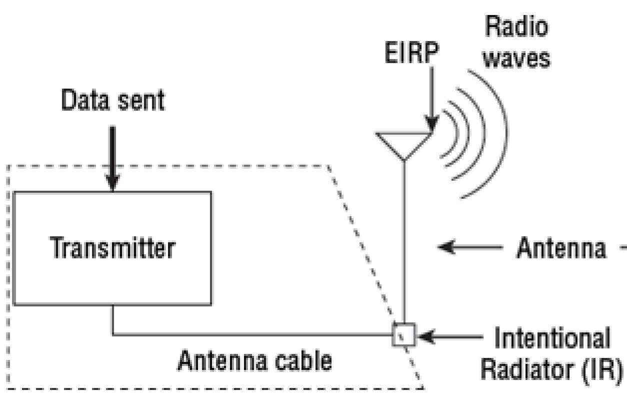
BLOCK DIAGRAM


Power supply
Rain fall sensor
(Temperature, Humidity, Moisture)
MICRO CONTROLLER
IOT MODULE
Road sign

HARDWARE DESCRIPTION

Radio Frequency Transmission

A radio frequency (RF) signal begins as an electrical alternating current (AC) signal that is originally generated by a transmitter [6]. This AC signal is sent via a copper conductor which usually a coaxial cable and radiated out of an antenna element in the form of an electromagnetic wave. Changes of current flow in the antenna produce changes in the electromagnetic fields around the antenna.



RF Based Wireless Remote using TX-RX Modules

This circuit utilizes the RF module, transmitter and receiver (TX-RX) for making a wireless remote, which could be used to drive an output from a distant place [7]. RF module, as the name suggests, uses radio frequency to send signals. These signals are transmitted at a particular frequency and a baud rate. A receiver can receive these signals only if it is configured for that frequency. A four channel encoder/decoder pair has also been used in this system. The input signals, at the transmitter side, are taken through four switches while the outputs are monitored on a set of four system emitting diodes (LED) corresponding to each input switch.

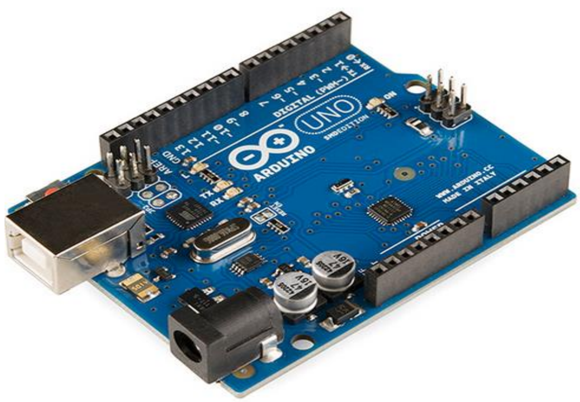
This RF transmission system employs Amplitude Shift Keying (ASK) with transmitter/receiver (Tx/Rx) pair operating at 434 MHz. The transmitter module takes serial input and transmits these signals through RF. The transmitted signals are received by the receiver module placed away from the source of transmission. The system allows one way communication between two nodes, namely, transmission and reception. The RF module has been used in conjunction with a set of four channel encoder/decoder ICs. The encoder converts the parallel inputs (from the remote switches) into serial set of signals. These signals are serially transferred through RF to the reception point. The decoder is used after the RF receiver to decode the serial format and retrieve the original signals as outputs. These outputs can be observed on corresponding LEDs.

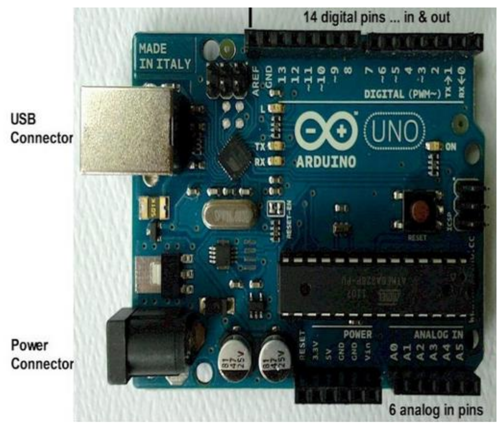
ARDUINO

Arduino is a computer hardware and software company, project, and user community that designs and manufactures microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL),[1] permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form, or as do-it-yourself kits.

The project's board designs use a variety of microprocessors and controllers. These systems provide sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards ("shields") and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, for loading programs from personal computers. The microcontrollers are mainly programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

The Arduino project started in 2005 as a program for students at the Interaction Design Institute Ivrea in Ivrea, Italy, aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats, and motion detectors





Daigram of Arduino ESP SMD R3

Developer	Arduino
Manufacturer	Many
Type	Single-board microcontroller
Operating system	None
CPU	Atmel AVR (8-bit), ARM Cortex-M0+ (32-bit), ARM Cortex-M3 (32-bit), Intel Quark (x86) (32-bit)
Memory	SRAM
Storage	Flash, EEPROM

ARDUINO ESP R3 MICROCONTROLLER

The Arduino ESP R3 is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The ESP differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2(Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

Revision 2 of the ESP board (A000046) has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.

Revision 3 of the board (A000066) has the following new features:

- 1.0 pin out: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible with both the board that uses the AVR, which operates with 5V and with the Arduino Due that operates with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2.

SPECIFICATION

- Microcontroller: ATmega328
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V

- Input Voltage (limits): 6-20V
- Digital I/O Pins:14(of which 6 provide PWM output)
- Analog Input Pins: 6
- DC Current per I/O Pin: 40mA
- DC Current for 3.3V Pin: 50mA
- Flash Memory: 32KB (ATmega328) of which 0.5 KB used by boot loader
- SRAM: 2KB(ATmega328)
- EEPROM: 1KB(ATmega328)
- Clock Speed: 16MHz
- Revision 3 of the board (A000066) has the following new features:
    - ATmega16U2 instead 8U2 as USB-to-Serial converter
    - 1.0 pin out: added SDA and SCL pins for TWI communication placed near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board and the second one is a not connected pin, that is reserved for future purposes
    - Stronger RESET circuit

Microcontroller	ATmega328P
Operating Voltage	5V

Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Power

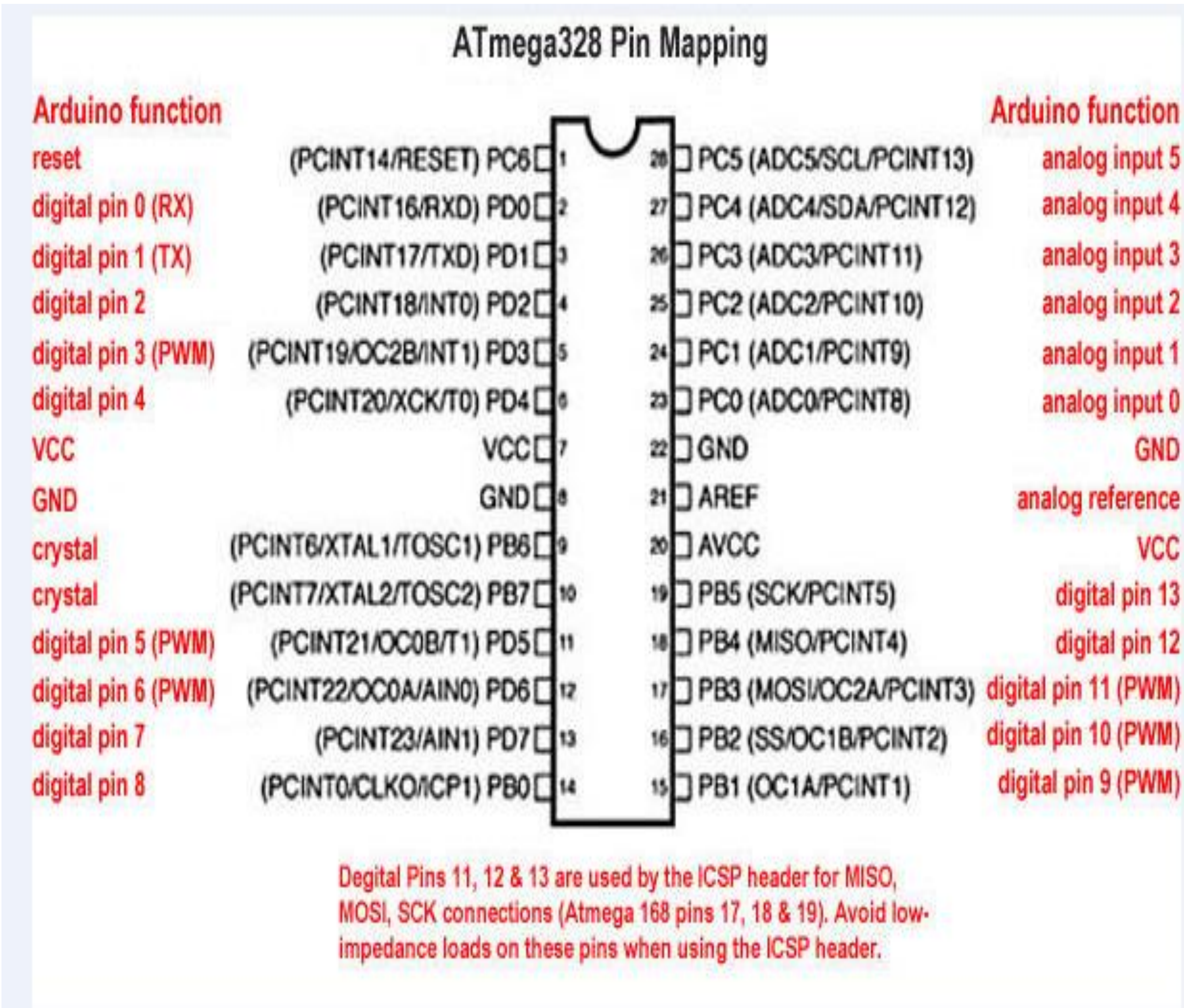
The Arduino ESP can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm centre-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. The power pins are as follows:

VIN.- The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

5V.- The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.

3V3 - A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

GND. Ground pins.


ATmega328 Pin Mapping
Arduino function
Arduino function
reset
(PCINT14/RESET) PC6 1
28 PC5 (ADC5/SCL/PCINT13)
analog input 5
digital pin 0 (RX)
(PCINT16/RXD) PD0 2
27 PC4 (ADC4/SDA/PCINT12)
analog input 4
digital pin 1 (TX)
(PCINT17/TXD) PD1 3
26 PC3 (ADC3/PCINT11)
analog input 3
digital pin 2
(PCINT18/INT0) PD2 4
25 PC2 (ADC2/PCINT10)
analog input 2
digital pin 3 (PWM)
(PCINT19/OC2B/INT1) PD3 5
24 PC1 (ADC1/PCINT9)
analog input 1
digital pin 4
(PCINT20/XCK/T0) PD4 6
23 PC0 (ADC0/PCINT8)
analog input 0
VCC
VCC 7
22 GND
GND
GND
GND 8
21 AREF
analog reference
crystal
(PCINT6/XTAL1/TOSC1) PB6 9
20 AVCC
VCC
crystal
(PCINT7/XTAL2/TOSC2) PB7 10
19 PB5 (SCK/PCINT5)
digital pin 13
digital pin 5 (PWM)
(PCINT21/OC0B/T1) PD5 11
18 PB4 (MISO/PCINT4)
digital pin 12
digital pin 6 (PWM)
(PCINT22/OC0A/AIN0) PD6 12
17 PB3 (MOSI/OC2A/PCINT3)
digital pin 11 (PWM)
digital pin 7
(PCINT23/AIN1) PD7 13
16 PB2 (SS/OC1B/PCINT2)
digital pin 10 (PWM)
digital pin 8
(PCINT0/CLKO/ICP1) PB0 14
15 PB1 (OC1A/PCINT1)
digital pin 9 (PWM)
Degital Pins 11, 12 & 13 are used by the ICSP header for MISO, MOSI, SCK connections (Atmega 168 pins 17, 18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

Input and Output

Each of the 14 digital pins on the ESP can be used as an input or output, using pin Mode(), digital Write(), and digital Read() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 m A 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. See the attach Interrupt() function for details.

PWM: 3, 5, 6, 9, 10, and 11.

Provide 8-bit PWM output with the analog Write() function.

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).

These pins support SPI communication using the SPI library.

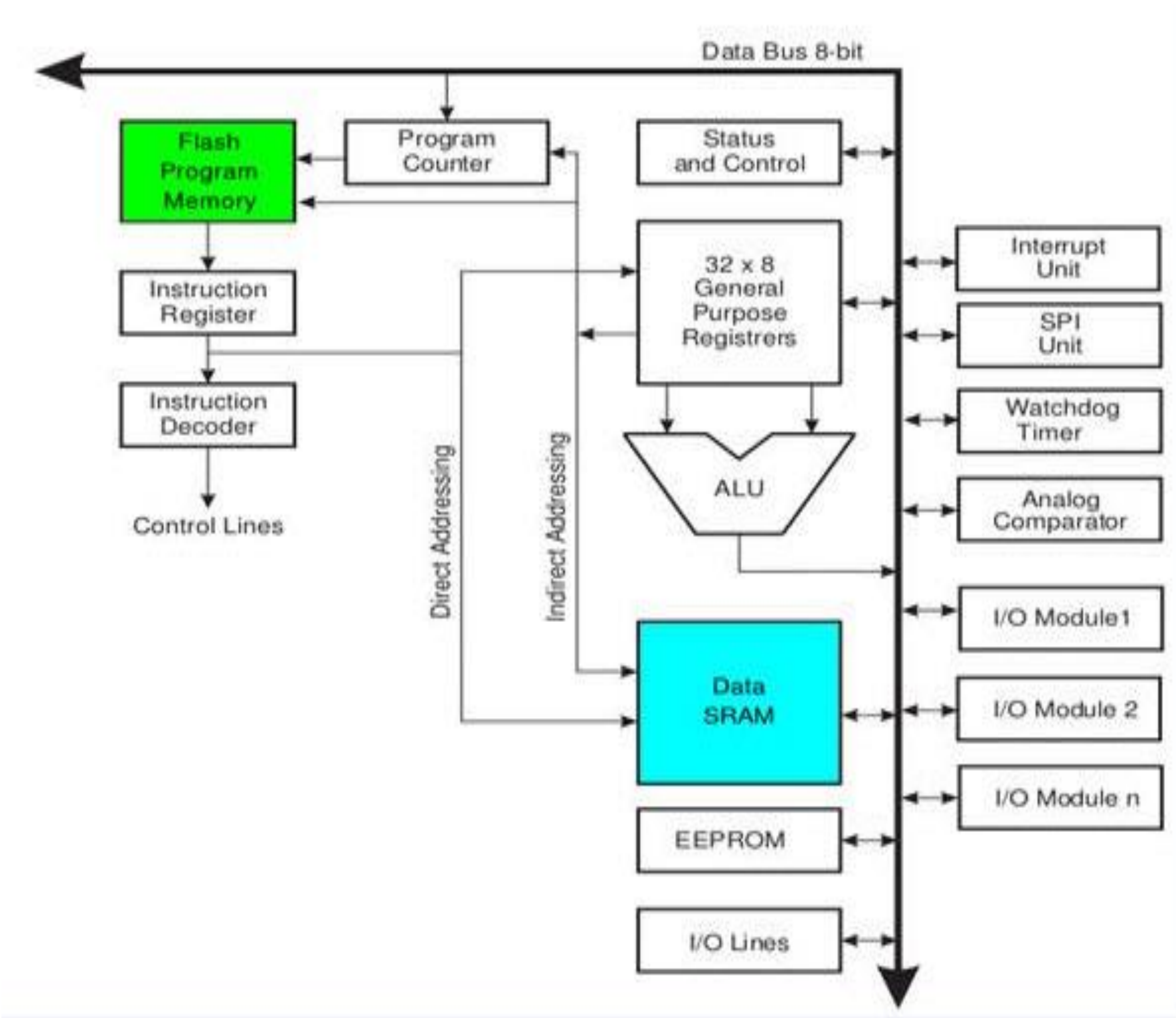
LED: 13.

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. 3 | P a g e 3 Arduino ESP The ESP has 6 analog inputs, labelled 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 analog Reference() function. Additionally, some pins have specialized functionality:

I2C: 4 (SDA) and 5 (SCL).

Support I2C (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 analog Reference(). 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.



Data Bus 8-bit
Flash
Program
Memory
Program
Counter
Status
and Control
Interrupt
Unit
32 x 8
General
Purpose
Registrers
SPI
Unit
Instruction
Register
Instruction
Decoder
Watchdog
Timer
ALU
Analog
Comparator
Control Lines
Direct Addressing
Indirect Addressing
I/O Module1
Data
SRAM
I/O Module 2
I/O Module n
EEPROM
I/O Lines

Communication

The Arduino ESP 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 ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '8U2 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 Software Serial library allows for serial communication on any of the ESP'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 ESP can be programmed with the Arduino software. Select "Arduino ESP from the Tools > Board menu (according to the microcontroller on your board). For details, see the reference and tutorials. The ATmega328 on the Arduino ESP comes preburned 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). You can also bypass the boot loader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details. The ATmega8U2 firmware source code is available . The ATmega8U2 is loaded with a DFU boot loader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. 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. Arduino ESP

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino ESP is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the boot loader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the ESP is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half second or so, the boot loader is running on the ESP. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data. The ESP contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labelled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

USB Over current Protection

The Arduino ESP has a resettable polyfuse that protects your computer's USB ports from shorts and over current. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

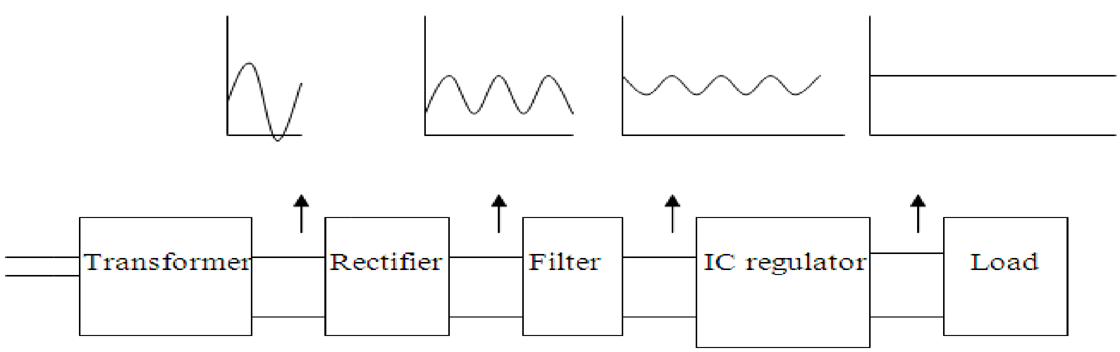
The maximum length and width of the ESP PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

POWER SUPPLIES

INTRODUCTION

The present chapter introduces the operation of power supply circuits built using filters, rectifiers, and then voltage regulators. Starting with an ac voltage, a steady dc voltage is obtained by rectifying the ac voltage, then filtering to a dc level, and finally, regulating to obtain a desired fixed dc voltage. The regulation is usually obtained from an IC voltage regulator unit, which takes a dc voltage and provides a somewhat lower dc voltage, which remains the same even if the input dc voltage varies, or the output load connected to the dc voltage changes.

A block diagram containing the parts of a typical power supply and the voltage at various points in the unit is shown in fig 19.1. The ac voltage, typically 120 V rms, is connected to a transformer, which steps that ac voltage down to the level for the desired dc output. A diode rectifier then provides a full-wave rectified voltage that is initially filtered by a simple capacitor filter to produce a dc voltage. This resulting dc voltage usually has some ripple or ac voltage variation. A regulator circuit can use this dc input to provide a dc voltage that not only has much less ripple voltage but also remains the same dc value even if the input dc voltage varies somewhat, or the load connected to the output dc voltage changes. This voltage regulation is usually obtained using one of a number of popular voltage regulator IC units.



IC VOLTAGE REGULATORS

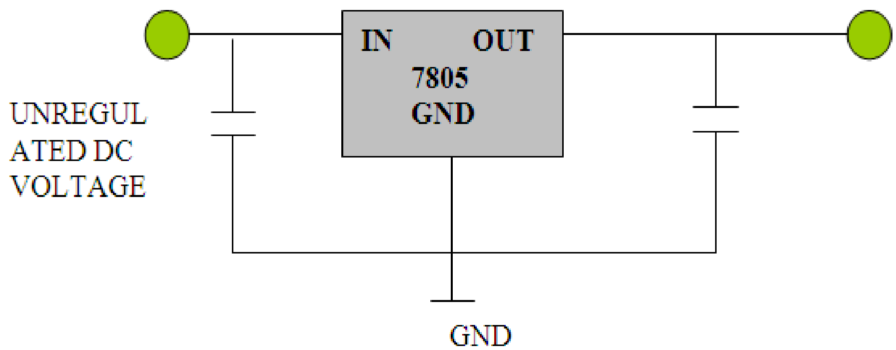
Voltage regulators comprise a class of widely used ICs. Regulator IC units contain the circuitry for reference source, comparator amplifier, control device, and overload protection all in a single IC. Although the internal construction of the IC is somewhat different from that described for discrete voltage regulator circuits, the external operation is much the same. IC units provide regulation of either a fixed positive voltage, a fixed negative voltage, or an adjustably set voltage.

A power supply can be built using a transformer connected to the ac supply line to step the ac voltage to desired amplitude, then rectifying that ac voltage, filtering with a capacitor and RC filter, if desired, and finally regulating the dc voltage using an IC regulator. The regulators can be selected for operation with load currents from hundreds of mille amperes to tens of amperes, corresponding to power ratings from mill watts to tens of watts.

THREE-TERMINAL VOLTAGE REGULATORS

Fig shows the basic connection of a three-terminal voltage regulator IC to a load. The fixed voltage regulator has an unregulated dc input voltage, Vi, applied to one input terminal, a regulated output dc voltage, Vo, from a second terminal, with the third terminal connected to ground. For a selected regulator, IC device specifications list a voltage range over which the input voltage can vary to maintain a regulated output voltage over a range of load current. The specifications also list the amount of output voltage change resulting from a change in load current (load regulation) or in input voltage (line regulation).

Fixed Positive Voltage Regulators:



The series 78 regulators provide fixed regulated voltages from 5 to 24 V. Figure 19.26 shows how one such IC, a 7812, is connected to provide voltage regulation with output from this unit of +12V dc. An unregulated input voltage Vi is filtered by capacitor C1 and connected to the IC's IN terminal. The IC's OUT terminal provides a regulated + 12V which is filtered by capacitor C2 (mostly for any high-frequency noise). The third IC terminal is connected to ground (GND). While the input voltage may vary over some permissible voltage range, and the output load may vary over some acceptable range, the output voltage remains constant within specified voltage variation limits. These limitations are spelled out in the manufacturer's specification sheets. A table of positive voltage regulated ICs is provided in table 19.1.

TABLE 19.1 Positive Voltage Regulators in 7800 series

IC Part	Output Voltage (V)	Minimum Vi (V)
7805	+5	7.3
7806	+6	8.3
7808	+8	10.5
7810	+10	12.5
7812	+12	14.6
7815	+15	17.7
7818	+18	21.0
7824	+24	27.1

MAX 232:

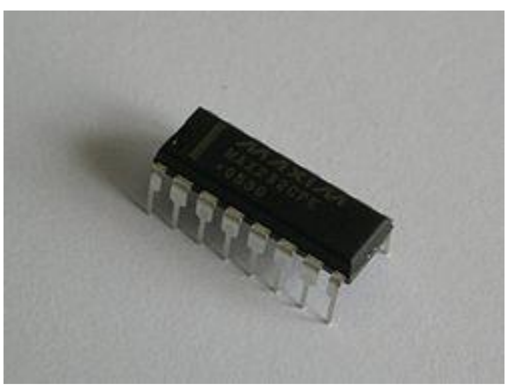
The **MAX232** is an integrated circuit, first created by Maxim Integrated Products, that converts signals from an RS-232 serial port to signals suitable for use in TTL compatible digital logic circuits. The MAX232 is a dual driver/receiver and typically converts the RX, TX, CTS and RTS signals.

The drivers provide RS-232 voltage level outputs (approx. ± 7.5 V) from a single + 5 V supply via on-chip charge pumps and external capacitors. This makes it useful for implementing RS-232 in devices that otherwise do not need any voltages outside the 0 V to + 5 V range, as power supply design does not need to be made more complicated just for driving the RS-232 in this case.

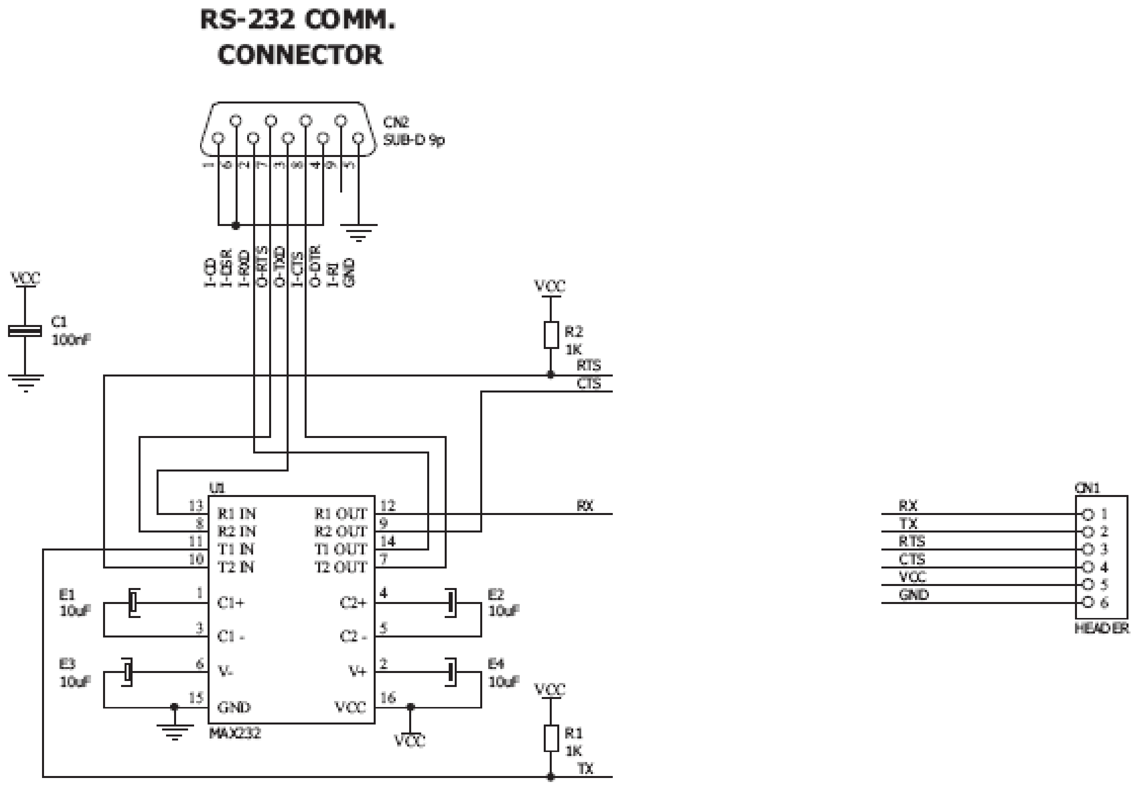
The receivers reduce RS-232 inputs (which may be as high as ± 25 V), to standard 5 V TTL levels. These receivers have a typical threshold of 1.3 V, and a typical hysteresis of 0.5 V.

The later MAX232A is backwards compatible with the original MAX232 but may operate at higher baud rates and can use smaller external capacitors – 0.1 μf in place of the 1.0 μF capacitors used with the original device.

The newer MAX3232 is also backwards compatible, but operates at a broader voltage range, from 3 to 5.5 V



MAX232 CIRCUIT



VOLTAGE LEVEL

It is helpful to understand what occurs to the voltage levels. When a MAX232 IC receives a TTL level to convert, it changes a TTL Logic 0 to between +3 and +15 V, and changes TTL Logic 1 to between -3 to -15 V, and vice versa for converting from RS232 to TTL. This can be confusing when you realize that the RS232 Data Transmission voltages at a certain logic state are opposite from the RS232 Control Line voltages at the same logic state. To clarify the matter, see the table below. For more information see RS-232 Voltage Levels.

RS232 Line Type & Logic Level	RS232 Voltage	TTL Voltage to/from MAX232
Data Transmission (Rx/Tx) Logic 0	+3 V to +15 V	0 V
Data Transmission (Rx/Tx) Logic 1	-3 V to -15 V	5 V
Control Signals (RTS/CTS/DTR/DSR) Logic 0	-3 V to -15 V	5 V
Control Signals (RTS/CTS/DTR/DSR) Logic 1	+3 V to +15 V	0 V

SOFTWARE DESCRIPTION

Embedded C

An embedded system is an application that contains at least one programmable computer (typically in the form of a microcontroller, a microprocessor or digital signal processor chip) and which is used by individuals who are, in the main, unaware that the system is computer-based.

Introduction

Looking around, we find ourselves to be surrounded by various types of embedded systems. Be it a digital camera or a mobile phone or a washing machine, all of them has some kind of processor functioning inside it. Associated with each processor is the embedded software. If hardware forms the body of an embedded system, embedded processor acts as the brain, and embedded software forms its soul. It is the embedded software which primarily governs the functioning of embedded systems.

During infancy years of microprocessor based systems, programs were developed using assemblers and fused into the EPROMs. There used to be no mechanism to find what the program was doing. LEDs, switches, etc. were used to check correct execution of the program. Some 'very fortunate' developers had In-circuit Simulators (ICEs), but they were too costly and were not quite reliable as well.

As time progressed, use of microprocessor-specific assembly-only as the programming language reduced and embedded systems moved onto C as the embedded programming language of choice. C is the most widely used programming language for embedded processors/controllers. Assembly is also used but mainly to implement those portions of the code where very high timing accuracy, code size efficiency, etc. are prime requirements.

Initially C was developed by Kernighan and Ritchie to fit into the space of 8K and to write (portable) operating systems. Originally it was implemented on UNIX operating systems. As it was intended for operating systems development, it can manipulate memory addresses. Also, it allowed programmers to write very compact codes. This has given it the reputation as the language of choice for hackers too.

As assembly language programs are specific to a processor, assembly language didn't offer portability across systems. To overcome this disadvantage, several high level languages, including C, came up. Some other languages like PLM, Modula-2, Pascal, etc. also came but couldn't find wide acceptance. Amongst those, C got wide acceptance for not only embedded systems, but also for desktop applications. Even though C might have lost its sheen as mainstream language for general purpose applications, it still is having a strong-hold in embedded programming. Due to the wide acceptance of C in the embedded systems, various kinds of support tools like compilers & cross-compilers, ICE, etc. came up and all this facilitated development of embedded systems using C. Subsequent sections will discuss what is Embedded C, features of C language, similarities and difference between C and embedded C, and features of embedded C programming.

EMBEDDED SYSTEMS PROGRAMMING

Embedded systems programming is different from developing applications on a desktop computers. Key characteristics of an embedded system, when compared to PCs, are as follows. Embedded devices have resource constraints(limited ROM, limited RAM, limited stack space, less processing power) Components used in embedded system and PCs are different; embedded systems typically uses smaller, less power consuming components. Embedded systems are more tied to the hardware. Two salient features of Embedded Programming are code speed and code size. Code speed is governed by the processing power, timing constraints, whereas code size is governed by available program memory and use of programming language. Goal of embedded system programming is to get maximum features in minimum space and minimum time.

Embedded systems are programmed using different type of language

- Machine Code
- Low level language, i.e., assembly
- High level language like C, C++, Java, Ada, etc.
- Application level language like Visual Basic, scripts, Access, etc.

Assembly language maps mnemonic words with the binary machine codes that the processor uses to code the instructions. Assembly language seems to be an obvious choice for programming embedded devices. However, use of assembly language is restricted to developing efficient codes in terms of size and speed. Also, assembly codes lead to higher software development costs and code portability is not there. Developing small codes are not much of a problem, but large programs/projects become increasingly difficult to manage in assembly language. Finding good assembly programmers has also become difficult nowadays. Hence high level languages are preferred for embedded systems programming.

Use of C in embedded systems is driven by following advantages it is small and reasonably simpler to learn, understand, program and debug. C Compilers are available for almost all embedded devices in use today, and there is a large pool of experienced C programmers.

.

Unlike assembly, C has advantage of processor-independence and is not specific to any particular microprocessor/ microcontroller or any system. This makes it convenient for a user to develop programs that can run on most of the systems. As C combines functionality of assembly language and features of high level languages, C is treated as a ‘middle-level computer language’ or ‘high level assembly language’. It is fairly efficient. It supports access to I/O and provides ease of management of large embedded projects.

Many of these advantages are offered by other languages also, but what sets C apart from others like Pascal, FORTRAN, etc. is the fact that it is a middle level language; it provides direct hardware control without sacrificing benefits of high level languages. Compared to other high level languages, C offers more flexibility because C is relatively small, structured language; it supports low-level bit-wise data manipulation.

Compared to assembly language, C Code written is more reliable and scalable, more portable between different platforms (with some changes). Moreover, programs developed in C are much easier to understand, maintain and debug. Also, as they can be developed more quickly, codes written in C offers better productivity. C is based on the philosophy 'programmers know what they are doing'; only the intentions are to be stated explicitly. It is easier to write good code in C & convert it to an efficient assembly code (using high quality compilers) rather than writing an efficient code in assembly itself. Benefits of assembly language programming over C are negligible when we compare the ease with which C programs are developed by programmers.

Objected oriented language, C++ is not apt for developing efficient programs in resource constrained environments like embedded devices. Virtual functions & exception handling of C++ are some specific features that are not efficient in terms of space and speed in embedded systems. Sometimes C++ is used only with very few features, very much as C.

Ada, also an object-oriented language, is different than C++. Originally designed by the U.S. DOD, it didn't gain popularity despite being accepted as an international standard twice (Ada83 and Ada95). However, Ada language has many features that would simplify embedded software development.

Java is another language used for embedded systems programming. It primarily finds usage in high-end mobile phones as it offers portability across systems and is also useful for browsing applications. Java programs require Java Virtual Machine (JVM), which consume lot of resources. Hence it is not used for smaller embedded devices. Dynamic C and B# are some proprietary languages which are also being used in embedded applications. Efficient embedded C programs must be kept small and efficient; they must be optimized for code speed and code size.

Good understanding of processor architecture embedded C programming and debugging tools facilitate this.

Difference between C and embedded C:

Though C and embedded C appear different and are used in different contexts, they have more similarities than the differences. Most of the constructs are same; the difference lies in their applications.

C is used for desktop computers, while embedded C is for microcontroller based applications. Accordingly, C has the luxury to use resources of a desktop PC like memory, OS, etc. While programming on desktop systems, we need not bother about memory. However, embedded C has to use with the limited resources (RAM, ROM, I/Os) on an embedded processor. Thus, program code must fit into the available program memory. If code exceeds the limit, the system is likely to crash.

Compilers for C (ANSI C) typically generate OS dependant executables. Embedded C requires compilers to create files to be downloaded to the microcontrollers/microprocessors where it needs to run. Embedded compilers give access to all resources which is not provided in compilers for desktop computer applications. Embedded systems often have the real-time constraints, which is usually not there with desktop computer applications.

Embedded systems often do not have a console, which is available in case of desktop applications. So, what basically is different while programming with embedded C is the mindset; for embedded applications, we need to optimally use the resources, make the program code efficient, and satisfy real time constraints, if any. All this is done using the basic constructs, syntaxes, and function libraries of ‘C’.

Keil C51 C Compilers

- Direct C51 to generate a listing file
- Define manifest constants on the command line
- Control the amount of information included in the object file
- Specify the level of optimization to use
- Specify the memory models

Specify the memory space for variables The Keil C51 C Compiler for the 8051 microcontroller is the most popular 8051 C compiler in the world. It provides more features than any other 8051 C compiler available today.

The C51 Compiler allows you to write 8051 microcontroller applications in C that, once compiled, have the efficiency and speed of assembly language. Language extensions in the C51 Compiler give you full access to all resources of the 8051.

The C51 Compiler translates C source files into reloadable object modules which contain full symbolic information for debugging with the µVision Debugger or an in-circuit emulator. In addition to the object file, the compiler generates a listing file which may optionally include symbol table and cross reference information.

SOFTWARE DESCRIPTION

PROTEUS

Proteus PCB design electronic circuits can computer-aided design and circuit boards are designed.

ISIS (Intelligent Schematic Input System)

The ISIS Intelligent Schematic Input System (Intelligent Switching input system), is the environment for the design and simulation of electronic circuits. The component library includes claims more than 10,000 circuit components with 6000 Prospice Simulations models. Own components can be created and added to the library.

ISIS includes a base VSM engine with support for the following functions:

- DC / AC voltmeter and ammeter, oscilloscopes, logic analyzers
- Analog signal generators, digital pattern generator
- Timer functions, protocol analyzers (including RS232, I2C, SPI)

VSM (Virtual System Modeling)

The VSM Virtual System Modeling provides a graphical SPICE circuit simulation and animation directly in the ISIS environment. The SPICE simulator is based on the Berkeley SPICE3F5 model.

It can microprocessor-based systems can be simulated. With the VSM engine can interact during the simulation directly with the circuit. Changes of buttons, switches or potentiometers are queried in real-time and LED indicators, LCD displays, "Hot / Cold" Wires displayed.

Proteus 7.0 is a Virtual System Modeling that combines circuit simulation, animated components and microprocessor models to co-simulate the complete microcontroller based designs. This is the perfect tool for engineers to test their microcontroller designs before constructing a physical prototype in real time. This program allows users to interact with the design using on-screen indicators and/or LED and LCD displays and, if attached to the PC, switches and buttons.

One of the main components of Proteus 7.0 is the Circuit Simulation -- a product that uses a SPICE3f5 analogue simulator kernel combined with an event-driven digital simulator that allow users to utilize any SPICE model by any manufacturer. Proteus VSM comes with extensive debugging features, including breakpoints, single stepping and variable display for a neat design prior to hardware prototyping.

This is the perfect tool for engineers to test their microcontroller designs before constructing a physical prototype in real time. This program allows users to interact with the design using on-screen indicators and/or LED and LCD displays and, if attached to the PC, switches and buttons.

One of the main components of Proteus 7.0 is the Circuit Simulation -- a product that uses a SPICE3f5 analogue simulator kernel combined with an event-driven digital simulator that allow users to utilize any SPICE model by any manufacturer. Proteus VSM comes with extensive debugging features, including breakpoints, single stepping and variable display for a neat design prior to hardware prototyping.

In summary, Proteus 7.0 is the program to use when you want to simulate the interaction between software running on a microcontroller and any analog or digital electronic device connected to it.

CONCLUSION

As a conclusion, this project have achieved the main objective stated earlier which is analyzing and implementing the wireless communication; the radio frequency (RF) transmission in the Intelligent road sign system control system for abnormal weather conditions vehicles. The prototype of this project is using the frequency of 434 MHz compared to the range of about 3 kHz to 300 GHz of frequency which have been reserved for the RF theoretically. Besides, the functionality of this project proved that the other objectives have been successfully attained which are designing an abnormal weather conditions sequence mode of Intelligent road sign system when abnormal weather conditions vehicles passing by an intersection and changing the sequence back to the normal sequence before the abnormal weather conditions mode was triggered. The sequences for this project have been developed using the programming in the microcontroller PIC 16F877A. In future, this prototype system can be improved by controlling the real Intelligent road sign situation and the study can be done by investigating the length, reception and transmission issue for the system to be operated with this Intelligent road sign system system.

REFERENCE

[1] Levi L. Rose, “Abnormal weather conditions Intelligent road sign Control System with Security Transmission Coding”, United States Patent, April 5th, 1997.

[2] Michael R. Smith, Paul J. Davidson and Henry L. Pfister, “Abnormal weather conditions Vehicle Warning and Intelligent road sign Control System”, United States Patent, October 4th, 1998.

[3] Willbur L. Mitchell, “Intelligent road sign System Control For Abnormal weather conditions Vehicles”, United States Patent, April 17th, 1994.

[4] William E. Brill, “Abnormal weather conditions Vehicle Detection System”, United States Patent, March 26th, 2002

. [5] Carl J. Obeck, “Intelligent road sign Signal Control For Abnormal weather conditions Vehicles”, United States Patent, May 7th, 1998. [6] David D. Coleman, David A. Westcott, “CWNA: Certified Wireless Network Administrator Official Study Guide”, Sybex, April 06, 2009.

[7] Mir Roomi Rahil, Rajesh Mahind, Saurabh Chavan, Tanumay Dhar, “GLCD-Touchpad Based Restaurant Ordering & Automatic Serving System”, International Journal of Recent Technology and Engineering (IJRTE), 2013.

[8] “RF Based Wireless Remote using Tx-Rx Modules”, Chawla Radios & Electricals, August 11th, 2011. [9] http://www.blurtit.com/q335716.html, What Is The Difference Between Decoder And Encoder? Lukas Hoffmann, “PIC 16F877A Tutorials for Pitt Robotics Club”, 2010.

[10] http://www.microchip.com, PIC16F87XA Data Sheet, Microchip Technology Inc.