INTELLIGENT TRAFFIC LIGHT CONTROL SYSTEM

PROJECT REPORT

As a partial of fulfilment of the curriculum

By

SRUTHI A V (NSAOEEC084)

LAKSHMI C K (NSAOEEC100)

SAIBUNNEESA S (NSAOEEC101)

S SAJNA (NSAOEEC102)

Under the guidance of

Assistant Prof. HARIKRISHNAN I

Department of Electronics & Communication Engineering

NSS College of Engineering, Palakkad

March 2017

N.S.S.COLLEGE OF ENGINEERING

PALAKKAD, KERALA 678008

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that this is a bonafied report of the project titled "**INTELLIGENT TRAFFIC LIGHT CONTROL SYSTEM"** done by **SRUTHI A V (NSAOEEC084), LAKSHMI C K (NSAOEEC100), SAIBUNEESA S (NSAOEEC101), S SAJNA (NSAOEEC102)** during the academic year 2017 under the guidance of **ASSISTANT PROF. HARIKRISHNAN I** as a part of the partial fulfilment for the award of Bachelor of Technology in **ELECTRONICS AND COMMUNICATION ENGINEERING** from University of Calicut and no part of this work has been submitted earlier for the award of any degree.

PROJECT GUIDE	**STAFF IN CHARGE**	**HEAD OF THE DEPARTMENT**
HARIKRISHNAN I	SREELEJA.N.UNNITHAN	KALA.L
Assistant Professor	Assistant Professor	Associate Professor

ACKNOWLEGDEMENT

First of all, we thank the Almighty God, for granting us strength, courage and knowledge to complete this project successfully.

We acknowledge our sincere thanks to the management of NSS College of Engineering for their help in the successful progression of our project. We would also like to express our gratitude to principal **Dr.T.Sudha**.

We take this opportunity to express our profound gratitude to our project guide **Prof. Harikrishnan I**, our staff in charge **Prof. Sreeleja.N.Unnithan**, and Head of Department **Prof. Kala.L**, for all their guidance and valuable support towards making our project a grand success.

We would also like to thank all the non-teaching staff of our department for their constant encouragement throughout our project. Last, but not the least we take pleasant privilege in expressing our heartfelt thanks to our family and friends who offered precious help in completing our project.

SRUTHI A V

LAKSHMI C K

SAIBUNNEESA S

S SAJNA

DEPARTMENT OF ELECTRONICS AND COMMUNICATION

NSS COLLEGE OF ENGINEERING, PALAKKAD

ABSTRACT

The idea behind this project is to meet the day to day problems faced due to traffic congestion by establishing an effective way to resolve the trouble. The already existing traffic control system has the drawback of equal time delays allotted for all paths which is inconvenient due to the varying measure of traffic. Here we are introducing a density based traffic control system in which time delays are allotted based on vehicle density on each road. The system is equipped with IR sensor modules for density measurement and traffic lights are represented using LEDs.

CONTENTS

CHAPTER 1

INTRODUCTION

Traffic congestion is a severe problem in many major cities across the world and it has become a nightmare for the commuters in these cities.

Conventional traffic light system is based on fixed time concept allotted to each side of the junction. Junction timings allotted are fixed. Sometimes higher traffic density at one side of the junction demands longer green time as compared to standard allotted time. At the same time there may be wastage of time due to lower traffic densities. These are the disadvantages of existing traffic system. In order to overcome the limitations of the present scenario we can vary the time allotted to each road at a junction based on the traffic density. So that longer delay can be assigned to that part of the junction that faces more traffic congestion. This can be provided by proper sensing of a particular area of a road.

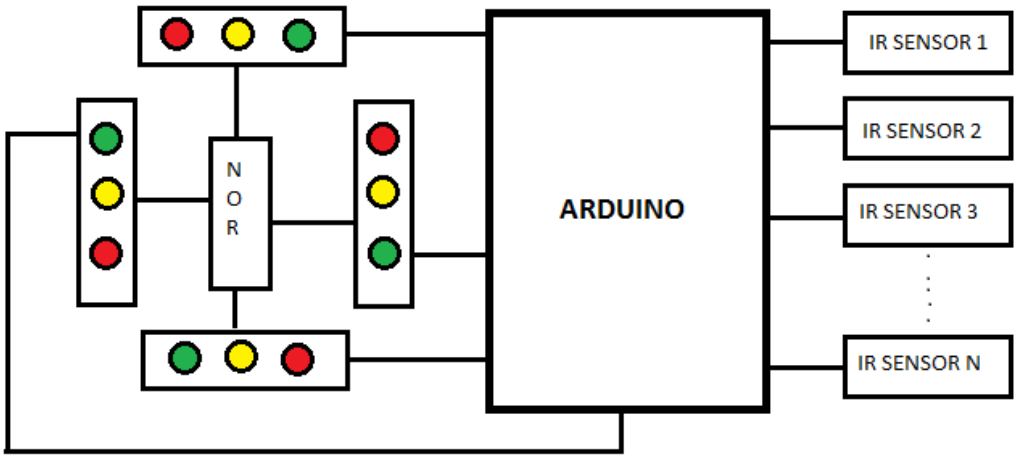
1.1 OBJECTIVES

- Measure the traffic density at a junction using IR sensor modules.
- Allot proper timing based on traffic density measured

CHAPTER 2

HARDWARE IMPLEMENTATION

DENSITY BASED TRAFFIC CONTROL SYSTEM BLOCK DIAGRAM



2.1 IR SENSOR MODULE

- An **infrared sensor** is an electronic device, that emits in order to sense some aspects of the surroundings.
- This consists of a transmitter (light emitting diode) and receiver (photo diode).

Pin No.	Connection	Description
1	Output	Digital Output (High or Low)
2	VCC	Connected to circuit supply
3	Ground	Connected to circuit ground

Table 2.1: Pin description of IR sensor module

Fig 2.1:IR sensor module

2.2 ARDUINO UNO

Arduino Uno is relatively small, reliable and cheap microcontroller that come in package with Arduino board and Atmel ATmega328. The robust characteristics made Arduino Uno able to perform as Intelligent System controller. As an open source platform, Arduino Uno is easily programmable to accommodate System complex instruction set. There are more Arduino Uno specifications which are stated in Table 2.2

Digital Outputs	14 ports(6 PWM output)
Analog Input Pins	6 pins
Flash Memory(ATmega328)	32K Bytes(0.5kB used by boot loader)
SRAM(ATmega328)	2K Bytes
EEPROM(ATmega328)	1K Bytes
Clock Speed	16MHz

Table.2.2:Arduino Uno Specification

The Figure 2.2shows Atmel ATmega328's pin which is used in Arduino Uno development Board.

2.3 IC 7402 QUAD 2 INPUT NOR GATE

7402 IC is a device containing four independent gates each of which performs the logic NOR function. The pin out and truth table of a nor gate is as shown below;

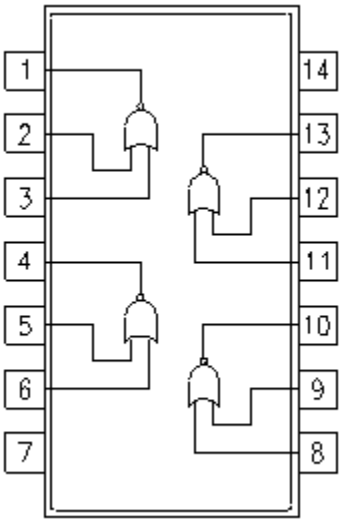


Fig 2.3:7402 NOR gate pinout

$$Y = \overline{A + B}$$

Inputs		Output
A	B	Y
L	L	H
L	H	L
H	L	L
H	H	L

H = HIGH Logic Level
L = LOW Logic Level

Table 2.3:Truth table of NOR gate

2.3 LIGHT EMITTING DIODE

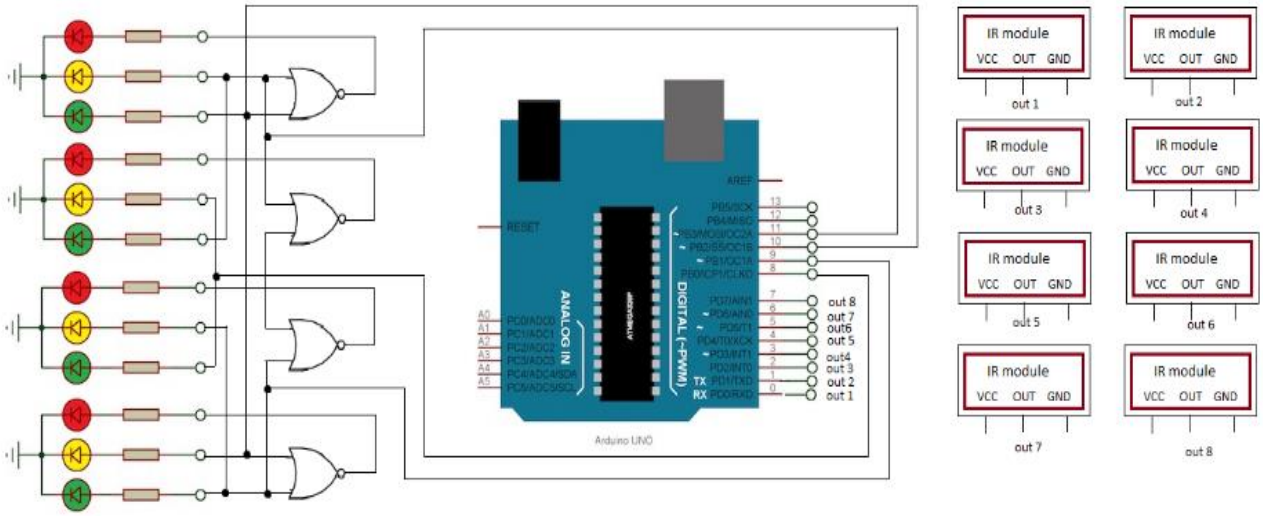
A light emitting diode (LED) is a two-lead semiconductor light source. It is a p-n junction diode that emits light when activated. When a suitable voltage is applied to the

leads, electrons are able to recombine with electron holes within the device, releasing energy in the form of photons. This effect is called electroluminescence, and the colour of the light (corresponding to the energy of the photon) is determined by the energy band gap of the semiconductor. LEDs are typically small (less than 1 mm^2) and integrated optical components may be used to shape the radiation pattern.

CHAPTER 3

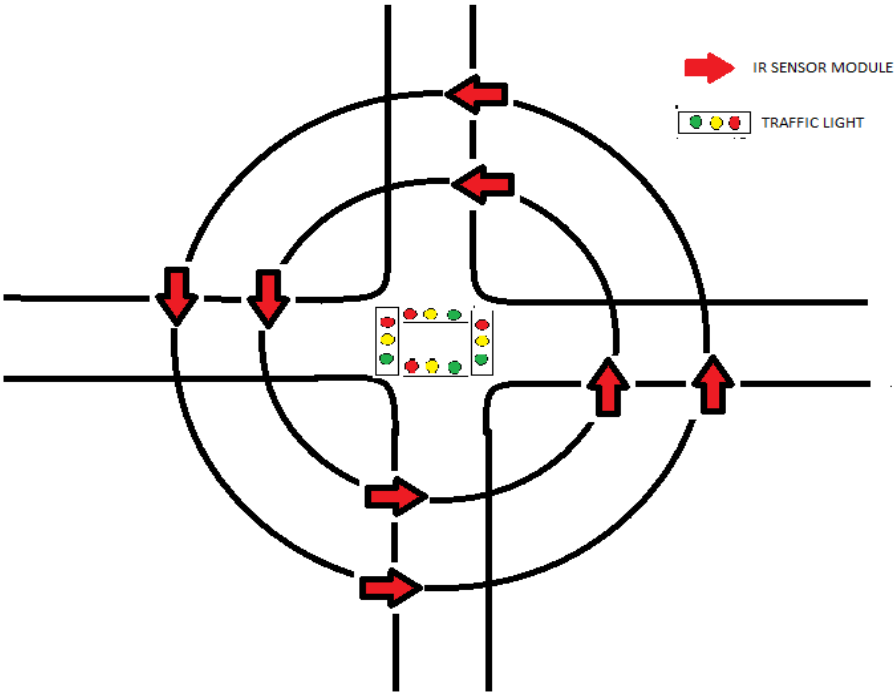
PERFORMANCE ANALYSIS

3.1 CIRCUIT DIAGRAM



3.2 WORKING

In this system IR sensor modules are used to measure the traffic density of a junction. Here junction is divided into three sections; LOW, MEDIUM and HIGH density zones.IR sensors are placed at the boundary between two zones as shown in the figure;



This model consists of two IR sensors on each road of a junction.IR sensors emits radiations which gets reflected back at the presence of an obstacle. In such a manner presence of vehicles on the road can be detected.

The outputs of all the sensors are connected to the arduino board as input. Output from arduino is given to four green LEDs which indicate the green signal of a traffic system. Arduino is flashed with a program which is coded in such a way that different time delay is provided for each combination of IR output. Thus in a road when IR placed at low density zone alone gives an output, a lower time delay is given for green signal and when both the sensors in a road give high output that indicates a high density and higher time delay is given for green signal.

All the other LEDs indicating yellow and red signals are connected to the green through a NOR gate so that they turn on correspondingly.

CHAPTER 4

BILL OF MATERIALS

COMPONENTS	COST	NUMBER OF COMPONENTS
1.Aurdino	550	1
2.IR Sensors	95	8
3 Resistors	1	12
4.Gate(NOR)	5	1
5.Led	1	12

CHAPTER 5

APPLICATION AND FUTURE SCOPE

- A simplest form of density based traffic control system.
- Density calculation can be made accurate using image sensing.
- Emergency vehicle detection can be implemented**.**
- The system can be modified by including the scheme for setting different time delay for different paths. Hence it become user friendly.

CHAPTER 6

RESULT

A working model of density based traffic control system is implemented using arduino board. Traffic lights were represented using LEDs and traffic density was measured using IR sensor modules. The LEDs turned on with varying time delays in correspondence with the traffic density measured by IR sensors on each road.

CHAPTER 7

CONCLUSION

The project "**INTELLIGENT TRAFFIC CONTROL SYSTEM**" has been successfully designed and tested.

Compared with the existing traffic system, the traffic system designed here is more synchronised with time. This system can be implemented for both urban as well as rural areas. It provides a significant reduction in human work and can be considered as a better system for time management.

CHAPTER 8

REFERENCES

www.circuitdigest.com

www.engineersgarage.com

www.yuvamastermind.com

IOT WORKSHOP IDEAS

MAGAZINE: Electronics for you

CHAPTER 9

APPENDIX

A.1 PROGRAM

```
int S1A=0;
int S1B=1;
int S2A=2;
int S2B=3;
int S3A=4;
int S3B=5;
int S4A=6;
int S4B=7;
int G1=8;
int G2=9;
int G3=10;
int G4=11;
void setup()

 {

  pinMode(G1,OUTPUT);
  pinMode(G2,OUTPUT);
  pinMode(G3,OUTPUT);
  pinMode(G4,OUTPUT);
  pinMode(S1A,INPUT);
  pinMode(S1B,INPUT);
  pinMode(S2A,INPUT);
  pinMode(S2B,INPUT);
```

```
  pinMode(S3A,INPUT);
  pinMode(S3B,INPUT);
  pinMode(S4A,INPUT);
  pinMode(S4B,INPUT);

}
void loop()

{

  digitalWrite(G1,HIGH);
  digitalWrite(G2,LOW);
  digitalWrite(G3,LOW);
  digitalWrite(G4,LOW);
  if(S1B==HIGH)
  delay(1200);
  else if(S1A==HIGH)
  delay(700);
  else
  delay(300);


  digitalWrite(G1,LOW);
  digitalWrite(G2,HIGH);
  digitalWrite(G3,LOW);
  digitalWrite(G4,LOW);
  if(S2B==HIGH)
  delay(1200);
  else if(S2A==HIGH)
```

```
 delay(700);
 else
 delay(300);


 digitalWrite(G1,LOW);
 digitalWrite(G2,LOW);
 digitalWrite(G3,HIGH);
 digitalWrite(G4,LOW);
 if(S3B==HIGH)
 delay(1200);
 else if(S3A==HIGH)
 delay(700);
 else
 delay(300);


 digitalWrite(G1,LOW);
 digitalWrite(G2,LOW);
 digitalWrite(G3,LOW);
 digitalWrite(G4,HIGH);
 if(S4B==HIGH)
 delay(1200);
 else if(S4A==HIGH)
 delay(700);
 else
 delay(300);

}
```

A.2 ARDUINO UNO DATASHEET

Arduino Uno

Arduino Uno R3 Front Arduino Uno R3 Back

Arduino Uno R2 Front Arduino Uno SMD Arduino Uno Front Arduino Uno Back

Overview

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, 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 Uno 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 Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode. Revision 3 of the board has the following new features: 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 both with the board that use the AVR which operate with 5V and with the Arduino .Due that operate with 3.3V The second one is not connected pin, that is reserved for future purposes. "Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

Summary

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 40 mA

DC Current for 3.3V Pin 50 mA

Flash Memory 32 KB (ATmega328) of which 0.5 KB used by boot loader.

SRAM 2 KB (ATmega328)
EEPROM 1 KB (ATmega328)
Clock Speed 16 MHz

Schematic & Reference Design

Note: The Arduino reference design can use an Atmega8, 168, or 328, Current models use an ATmega328, but an Atmega8 is shown in the schematic for reference. The pin configuration is identical on all three processors.

Power

The Arduino Uno 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 center-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 : This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator,and can damage your board. We don't advise it.

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

GND : Ground pins.

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).

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 kOhms. 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 attachInterrupt() function for details.

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. See also the mapping between Arduino pins and ATmega328 ports. The mapping for the Atmega8,168, and 328 is identical.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, orother 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 communicati on 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 ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by: der jumper on the back of the board (near the map of Italy) and then resetting the 8U2. 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). See this user-contributed tutorial for more information.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno 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 ATmega8U2/16U2 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 Uno 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 Uno. 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 Uno 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 Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. 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 Uno 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