



**Automatic traffic offences monitoring system at traffic light
controlled intersection**

By

Gehas Gerald Mangwaya

(R145483P)

CHAPTER 1

INTRODUCTION/LITERATURE REVIEW

1.1 Introduction

Traffic monitoring is a very difficult task for traffic control department, especially in large cities. This prototype project is designed to monitor a traffic light controlled intersection by using a camera, whenever a vehicle breaks the traffic rules the camera takes a snap shot.

Cameras are devices that, when linked to the signal controller, automatically photograph vehicles that enters the intersection after the onset of the red light. Typically offending drivers enter the intersection 1 2seconds after the onset of the red and most cameras are set to start photographing the drivers upon the 2second after red. Red light cameras are used to improve safety at traffic-light controlled intersections. Typically, the cameras are installed at high-accident intersections or at locations where drivers are likely to disobey traffic signals. Using sensors, the vehicles are monitored with reference to the traffic light indicators. A proximity induction sensor always on and if a vehicle exceeds the control line on the road, after RED light is on, the camera switches on and the images are stored in Windows Media file. The authority can view the images from a remote computer.

1.2 Literature Review

1.2.1 Brief back ground on Automatic traffic monitoring at robot controlled intersection

Experience with red light camera installations in the Netherlands and in Australia indicate that this technology can reduce incidents of running the red light by 35 to 60 percent[1]. Furthermore, reductions in rightangled accidents of 32 percent have been reported. In order to effectively use red light cameras, it is necessary to have legislation that issuing tickets to vehicle owners. Due to the beneficial safety effects of red

light cameras, the United States Department of Transportation has initiated a red light program for local governments using sensors, the vehicles are monitored with reference to the traffic light indicators [3].

However, in the almost three decades that have passed since the first photo radar was installed, there have been some more substantial extensions and improvements of automatic enforcement systems as well [1]. Automatic monitoring and enforcement has extended to several types of violations other than speeding and new technologies detecting violations as well as for identifying the violating vehicles have appeared. Probably the most notable developments are systems involving the use of digital video with image processing, and systems for electronic recognition and identification of a vehicle. This deliverable aims to review influence developments in the field of automatic, monitoring and detection of traffic violations and provide suggestions for the introduction of new automatic enforcement in African projects.

One main reason for automatic monitoring and enforcement, except of the safety situation, is that the police will not be able to take direct action to each detected violator at normal police enforcement activities in some environments. By using detectors and camera technology the violators can be identified and sanctioned though it's almost three decades of use, automated traffic enforcement has mainly been applied to speed and red light violations. In the recent years, however, there has been an extension to other violations, e.g., following distance, lane keeping, and toll payment violations. The increased use of digital video and image processing technology, as well as the electronic identification of vehicles, has paved the way for extending the applications to a still wider spectrum of violations, as well as making the enforcement considerably more efficient in the future.

Automatic monitoring always consist of at least three different procedures

- Detection of the violation
- Identification of the vehicle involved

In order to combine these procedures into one system the aim is of course to have to create a fully automatic system. Today the detection of the violation is automatic but the identification processes are more or less manual. The strategy for automatic monitoring and enforcement varies a lot between countries depending on detection technology used, police force organisation, traffic legislation and sanction system. It is impossible to recommend a harmonised solution. An important prerequisite for an efficient automated monitoring and enforcement system is the availability of a centralised register of vehicles and their owners at a national level [5]. If the register were not centralised, the process of identifying vehicles and drivers from outside the jurisdiction where the violation takes place would be extremely laborious. That is also a reason why few countries routinely pick up violators from foreign countries. An example of sanctioning across borders is found in Scandinavia, where there are mutual agreements between Norway, Sweden and Denmark regarding the process of fining violators from the neighboring countries.[6]

1.2.2 *Manned vs. unmanned operation*

Some automated monitoring applications are manned; for example in those instances where the automatic system is operated from a police car. As long as the detection of violations is automatic, and the purpose of the manning is only to supervise the equipment, these applications are considered within our definition of automated monitoring system (whereas applications depending on a police officer to initiate the recording when observing a suspect vehicle, is not automated.) [7]. An advantage of manned controls seems to be increased flexibility concerning the choice of sites. Extensive unmanned operation on the other hand requires a certain number of fixed facilities between which the equipment can be rotated. In the long run, however, the costs of such facilities are probably lower than the costs of manned operation for a similar level of monitoring [8].

1.2.3 Automatic monitoring systems

It starts with a detection device that measures the presence of a vehicle, headway, and weight and has a logical unit that determines if the measured value constitutes a violation. For example, activation of a loop detector coincident with a red light as indicated by the traffic light controller. Once a violation is detected, it triggers one of registration device, which is usually some type of a camera. The basic picture will have a view of the vehicle's license plate and superimposed in the picture there will be the date, time, location and other information needed for evidence, processing and quality control. Depending on the application and the legal requirements, multiple pictures may be taken with one or more cameras, and pictures may include a full view of the vehicle, the surroundings, driver's face, back or front of car.

In some jurisdictions video (and particularly digital video) is not yet a "type approved" device for registering violation evidence. It is a forgone conclusion, however, that it will be approved in the near future. The EU VERA project (Video Enforcement for Road Authorities) will formalise and harmonise standards of video use in monitoring and enforcement, but many polic

e forces have already replaced wet film cameras with video [9]. The record of registered violations has to be collected and passed from the automatic monitoring site to a processing center. In the case of a mobile photoradar operating in automatic mode, a police officer simply removes the film cartridge (or video recording media) from the camera and takes it to the laboratory. In fixed installations utilising wetfilm or analogue video, someone has to come periodically, collect the used cartridge and replace them with new ones. These collection modes are manual. Although typical wet film cartridges for enforcement purposes contain 400 frames, the logistics of collecting them and servicing the cameras is a significant costly operation [10]. New systems based on digital video offer the option of transmitting the records of registered violations to a central facility, so that only a small buffer for temporary storage is necessary. This concept has an obvious advantage over the local storage, which delays the processing of the citation and risk a loss of all stored violation information.

The next critical step in the process is to identify the offending vehicle to enable linking the violation to either the owner of the vehicle or to the actual driver (depending on the legal requirement). In all existing systems identification is based on vehicle's registration plate. This is usually done in an office with the aid of optical gadgets or computer based programs to improve the legibility of the digitised characters of the plate. The mode is considered manual if an operator has to look at all the frames and type in the license plate id. It is semiautomatic if a character recognition program is used to identify and type the number while the operator only has to approve the id and resolve unclear cases. Few applications rely on automatic identification. Not only plate recognition programs have to be very sensitive and reliable, also the situation for detecting and registering the violation has to be very restrictive and free of possible confounding factor. Therefore, only in few application identification is completely automatic and even there one suspects that some human eye looks at the evidence pictures attaches to the citation before it is sent.

35mm cameras are the most common cameras used for automated enforcement of red light violation systems. When the traffic signal switches to the red phase, the camera used by the system becomes active (ready to take photographs). Vehicles travelling over the detectors while the camera is active signal the system to photograph the vehicle. A small period of time (us

usually 0,3 seconds) and a preset speed (usually 8 to 30 km/h) necessary to activate the system are incorporated to differentiate between vehicles attempting to stop or turn right on red and vehicles that are clearly running the red light [11].

When a vehicle running a red light activates the system, at least two pictures are taken. The first picture shows that the front of the vehicle is not in the intersection when the signal is red.

This picture must show the pavement marking defining the intersection (usually the stop bar), the traffic signal displaying a red light and the vehicle in question. The second picture then shows the vehicle in the intersection a short time later (0,5 to 1,5 seconds). If driver identification is necessary, a third picture of the driver may be taken. From the pictures taken, the license plate will be magnified to allow for identification of the vehicle. Information pertinent to the violation is superimposed on the photos. The date, time of day, intersection number, photo number, yellow phase time, and time into red phase are usually shown on the first photo.

The second photo may include date, time of day, photo number, yellow phase time, into red phase, and vehicle speed. Current technology allows video cameras to be used as part of an automated monitoring program. In countries that currently have laws forbidding the use of automated enforcement video cameras may be used to record and view the large number of violations on certain intersections [13]. In this way evidence can be presented to officials about the severity of the red light running problem. Digital imaging promises further advancement in the automation of red light citation processing. Digital cameras have the capability to produce higher resolution, more sharply detailed images of vehicles, and are equipped to prevent reflections or headlight from smearing images. Digital cameras are in operation in New South Wales, Australia. Along with producing better vehicle images, the major (expected) benefit of digital cameras is in improving the processing and distribution of notices of violation [14]

Digital cameras have the capability to be linked using dedicated lines or existing phone lines to a computer located in a central facility. Once the images have been transferred from the digital cameras to the central facility, pattern and optimal character algorithms can be used to determine the owner of the vehicle by cross-referencing the license number with records of vehicle registration databases. After license plate numbers are successfully matched with registered vehicle, tickets can be automatically processed and mailed to violators.[15]

Implementation of a digital camera based system is a major step in creating a truly automated monitoring enforcement system. True automation can only be achieved with a digital camera based system. The benefits include reduced labor and potentially more accurate evaluations of the photographs. On the other hand, drawbacks associated with digital camera systems remain [12]. The more salient issues include cost (currently, a digital system costs about eight times that of a 35mm system), legal support (photos are recorded electronically onto a WORM (write once, read many) device to ensure integrity [16]. The effectiveness of the WORM device has not been thoroughly evaluated), and digital camera capabilities such as data management, resolution, contrast latitude, frame rate, etc. In the field of monitoring red light offences, the Swedish firm Sensys has adapted a slightly different approach. Rather than concentrating just on capturing the offender with a photograph, Sensys SICAS (Signalised Intersection Collision Avoidance System) detects every vehicle that approaches the intersection. By using a special algorithm, any vehicle that passes a set distance with a speed higher than a preset speed, gives a signal to the traffic controller to extend the red light phase for the drivers on the crossing road, long enough to allow the offender to pass the intersection. In this way the system both captures the offender and avoids a collision in the intersection [12].

CHAPTER 2

THEORETICAL ASPECTS

2.1 Introduction

The chapter provides a theoretical background of Traffic light, digital camera and sensor systems.

2.2 Traffic light Theory

2.2A Controller for a Simple Traffic Light.

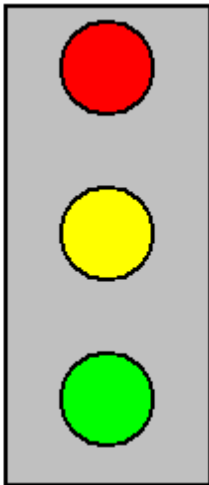


Figure 2.1 Traffic light

Assumption: Two Linked Pairs of Traffic Lights

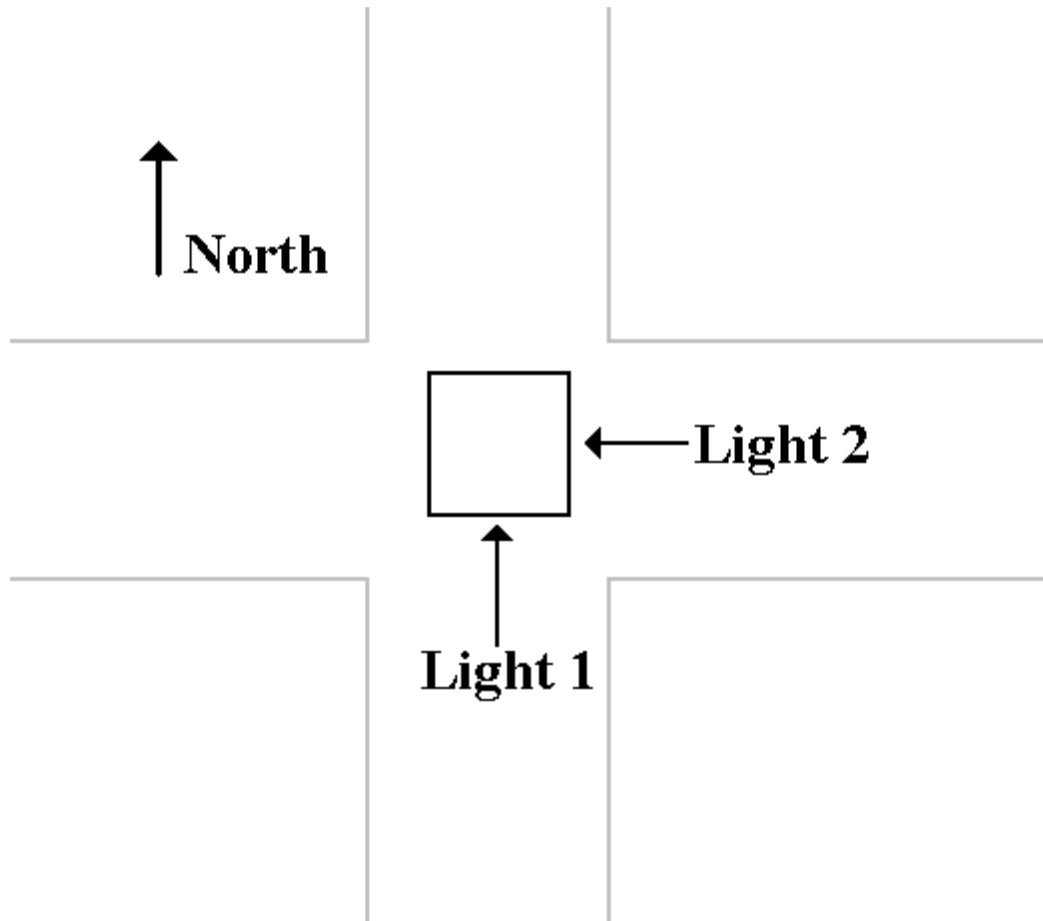


Figure 2.2 Two Linked Pairs of Traffic Lights.

If one light is Green, the “cross light” must be Red.

Assumed Cycling Rules

One Light	Cross Light	Comments
Green	Red	Traffic moving on one street
Yellow	Red	Traffic on cross street must wait for this light to turn red.
Red		Both lights are red for about one second.
Red	Green	Cross traffic now moves.

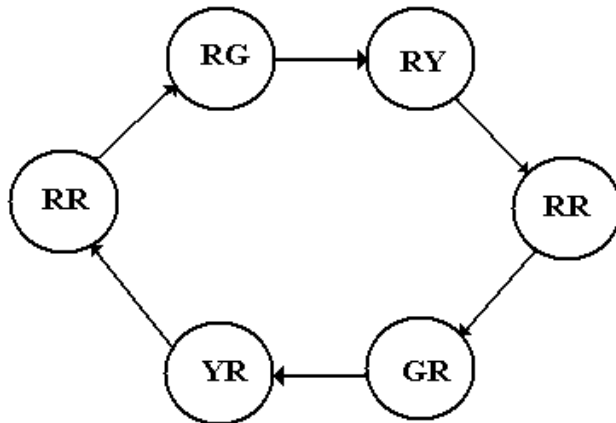
Table 1: The basic sequence for a traffic light without turn signals or Features such as an “advanced green”,

Names of the States

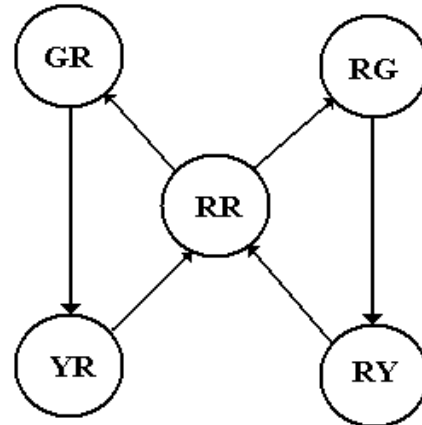
State	Light 1	Light 2	Alias
0	Red	Red	RR
1	Red	Green	RG
2	Red	Yellow	RY
3	Red	Red	RR
4	Green	Red	GR
5	Yellow	Red	YR

Table 2 Sequence of lighting

Step 1a: State Diagram for the System



A Six-State Design



A Five-State Design

Notation: L1L2, so RG □ Light 1 is Red and Light 2 is Green

Figure 2.3 The state design is more easily implemented

Step 2: Count the States and Determine the Flip-Flop Count

There are six states, so we have $N = 6$.

Solve $2^{P-1} < N \cdot 2^P$ for P , the number of flip-flops.

$2^{P-1} < 6 \cdot 2^P$ gives $P = 3$, because $2^2 < 6 \cdot 2^3$.

We denote the states by $Q_2 Q_1 Q_0$ because the symbol “Y” is taken to

Indicate the color Yellow.

Step 3: Assign a 3-bit Binary Number to Each State

This is a modified counter, so the assignments are quite obvious.

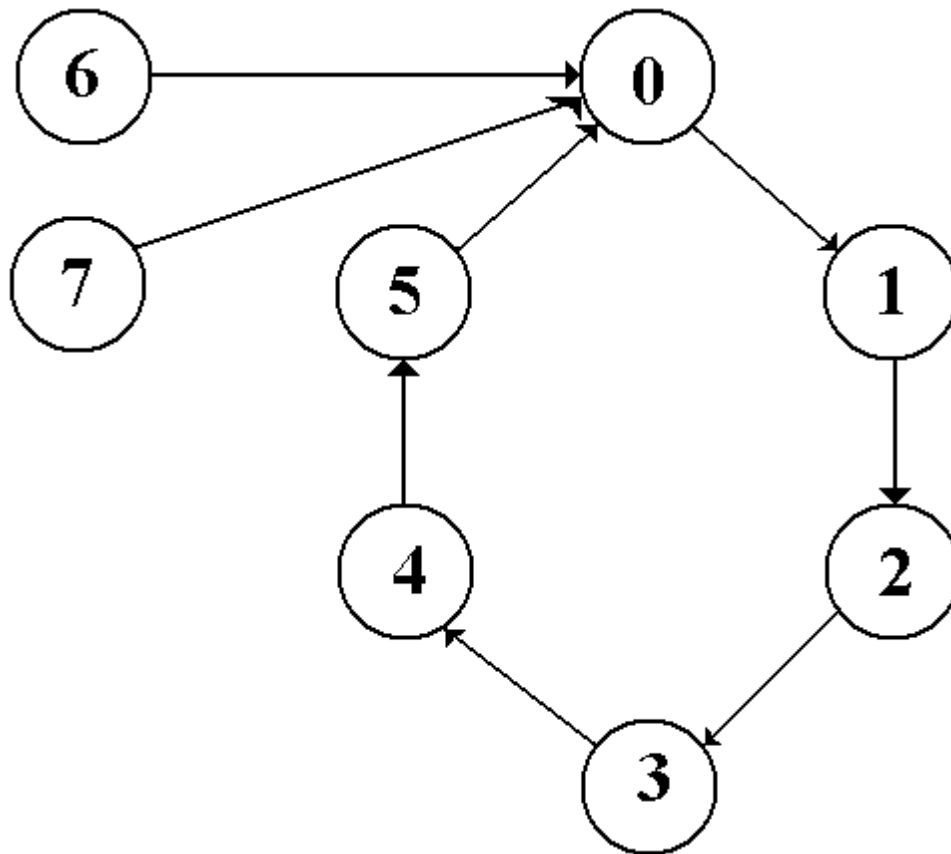
State	Q2	Q1	Q0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1

We have two possible additional states: 6 and 7.

Normally, these are ignored, but we consider them due to safety constraints.

Table 2.3 counter state

Redefined State Diagram to Add Safety



States 6 and 7 should never be entered. Each is “RR” for safety

Figure 2.4 Stage diagram with additional states

Step 4a: Derive the Output Equations.

	Alias	Q2	Q1	Q0	R1	G1	Y1	R2	G2	Y2
0	RR	0	0	0	1	0	0	0	1	0
1	RG	0	0	1	1	0	0	0	0	1
2	RY	0	1	0	1	0	0	0	0	1
3	RR	0	1	1	1	0	0	0	1	0
4	GR	1	0	0	0	1	0	0	1	0
5	YR	1	0	1	0	0	1	1	1	0
6	RR	1	1	0	1	0	0	0	1	0
7	RR	1	1	1	1	0	0	0	1	0

Table 2.3 Derive the Output

Here are the output equations

$$G1 = Q2 \cdot Q1' \cdot Q0' \quad G2 = Q2' \cdot Q1' \cdot Q0$$

$$Y1 = Q2 \cdot Q1' \cdot Q0 \quad Y2 = Q2' \cdot Q1 \cdot Q0'$$

$$R1 = (G1 + Y1)' \quad R2 = (G2 + Y2)'$$

Step 4b: Derive the Output Equations.

Here are the equations again.

$$G1 = Q2 \cdot Q1' \cdot Q0' \quad G2 = Q2' \cdot Q1' \cdot Q0$$

$$Y1 = Q2 \cdot Q1' \cdot Q0 \quad Y2 = Q2' \cdot Q1 \cdot Q0'$$

$$R1 = (G1 + Y1)' \quad R2 = (G2 + Y2)'$$

We derive the Green and Yellow signals, which are easier.

We stipulate that if a light is not Green or yellow, it must be Red.

Now add a safety constraint: If a light is Green or Yellow, the cross light must be Red.

$$R1 = (G1 + Y1)' + G2 + Y2, \text{ and}$$

$$R2 = (G2 + Y2)' + G1 + Y1$$

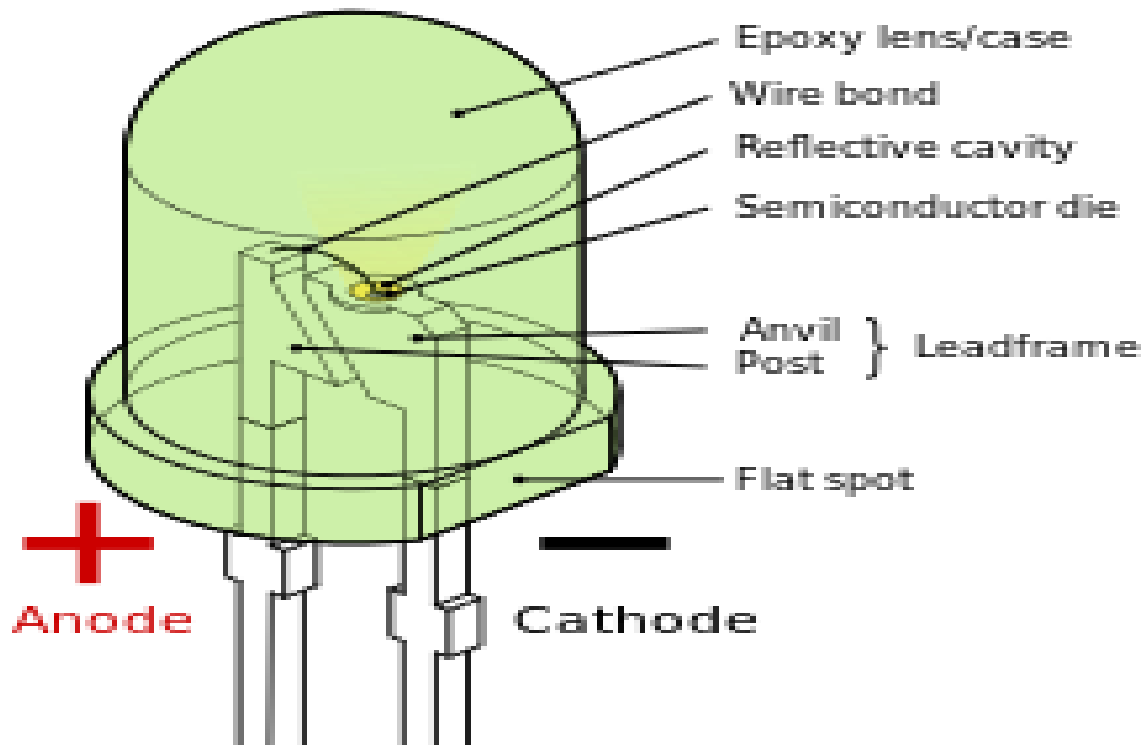


Figure 2.5 LEDs

LEDs require specified currents otherwise they will burn out.



FIG 2.6 Proximity induction sensor

The induction proximity sensors operate whenever a vehicle passes over them while the traffic light is red. 4mm proximity sensors have been used. The range and direction of the sensor was properly chosen such that vehicle in other lanes are not detected. These sensors detect only metals such that a person moving above them will not be detected. A metallic object will have to cut through the magnetic field and this will send a signal to the controlling device.

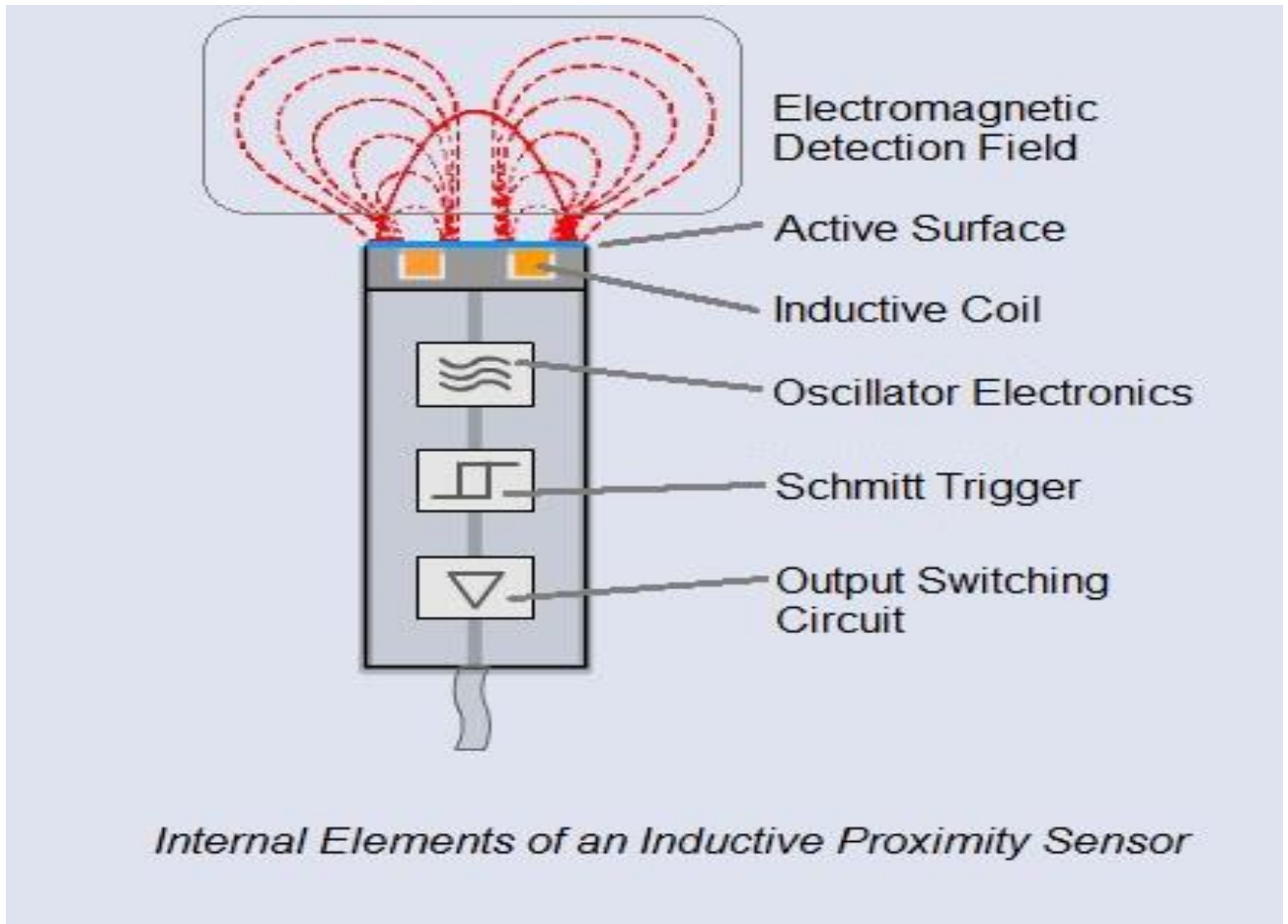


Figure 2.8 Elements of an Induction proximity sensor

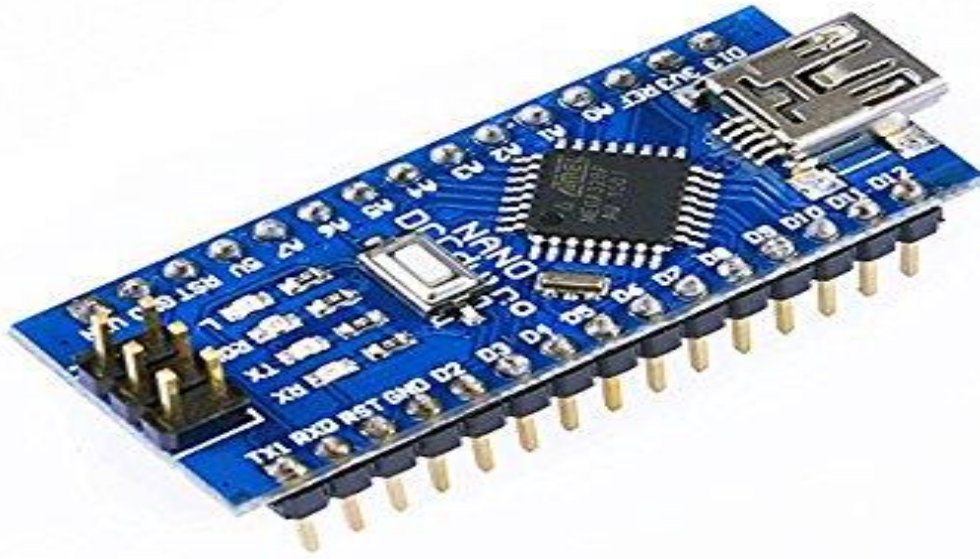


Figure 2.9Arduonano

Technical specs

Microcontroller	ATmega328
Architecture	AVR
Operating Voltage	5 V
Flash Memory	32 KB of which 2 KB used by bootloader
SRAM	2 KB
Clock Speed	16 MHz
Analog I/O Pins	8
EEPROM	1 KB
DC Current per I/O Pins	40 mA (I/O Pins)
Input Voltage	7-12 V
Digital I/O Pins	22
PWM Output	6
Power Consumption	19 mA
PCB Size	18 x 45 mm
Weight	7 g
Product Code	A000005

Power

The Arduino Nano can be powered via the Mini-B USB connection, 6-24V unregulated external power supply (pin 30), or 5V regulated external power supply (pin 27). The power source is automatically selected to the highest voltage source.

Memory

The ATmega328 has 32 KB, (also with 2 KB used for the boot loader. The ATmega328 has 2 KB of SRAM and 1 KB of EEPROM.

Input and Output

Each of the 14 digital pins on the Nano 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 FTDI 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 8bit PWM output with the `analogWrite()` function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.

- 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 Nano has 8 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the `analogReference()` function. Analog pins 6 and 7 cannot be used as digital pins. Additionally, some pins have specialized functionality:

- I2C: A4 (SDA) and A5 (SCL). Support I2C (TWI) communication using the `Wire` library (documentation on the Wiring website).

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.

Communication

The Arduino Nano has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provide UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1(TX). on the board channels this serial communication over USB and the FTDI drivers (included with the Arduino software) provide a virtual com port to software on the computer. 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 FTDI 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 Nano's digital pins.

The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software incl

udes a Wire library to simplify use of the I2C bus. To use the SPI communication, ATmega328 datasheet.

Programming

The Arduino Nano can be programmed with the Arduino software ([download](#)). Select "ArduinoDuemilanove or Nano w/ ATmega328" from the Tools > Board menu (according to the microcontroller-on-your-board).

The ATmega328 on the Arduino Nano comes preburned with a boot loader that allows you to upload new code to it without the use of an external hardware programmer. You can also bypass the boot loader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP or similar.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Nano 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 FT232RL 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 Nano 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 Nano. 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 onetime 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 Arduino Nano is open-source hardware!

VOLTAGE REGULATOR

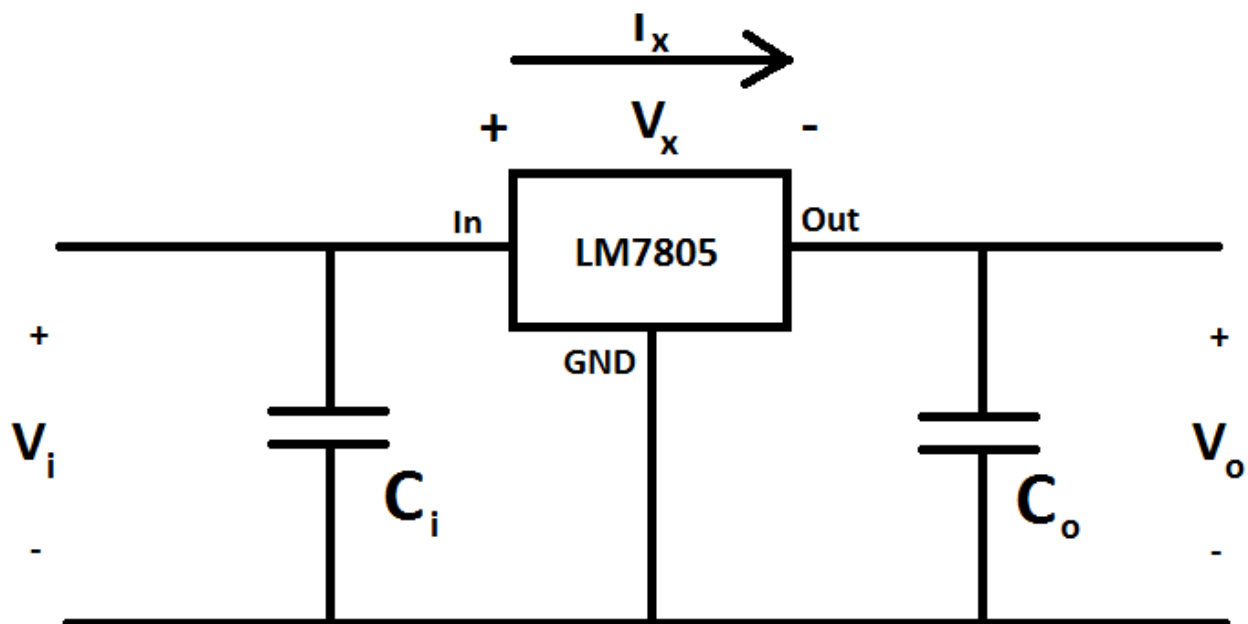


Figure 2.10 Voltage regulator LM7805

7805 is a **voltage regulator** integrated circuit. It is a member of 78xx series of fixed linear voltage regulator ICs. The voltage source in a circuit may have fluctuations and would not give the fixed voltage output. The **voltage regulator IC** maintains the output voltage at a constant value. The xx in 78xx indicates the fixed output voltage it is designed to provide. 78

05 provides +5V regulated power supply. Capacitors of suitable values can be connected at input and output pins depending upon the respective voltage levels.

Pin Diagram:

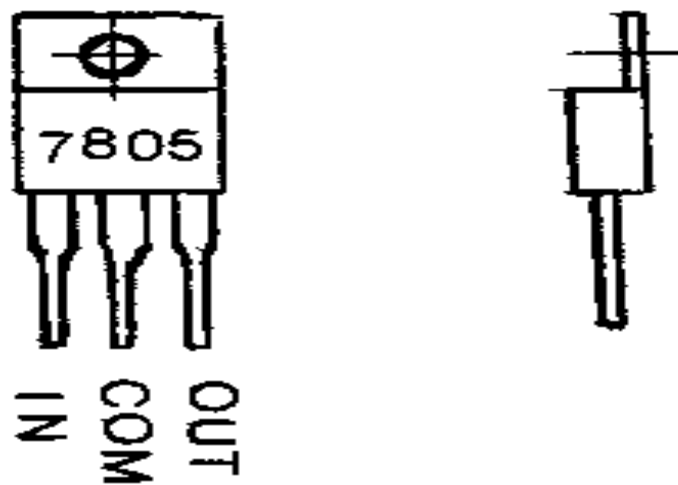


Figure 2.11 Pin Description:

Pin No	Function	Name
1	Input voltage (5V-18V)	Input
2	Ground (0V)	Ground
3	Regulated output; 5V (4.8V-5.2V)	Output

4 Channel Relay Module Programmable w/ RS485 STM8S103F3

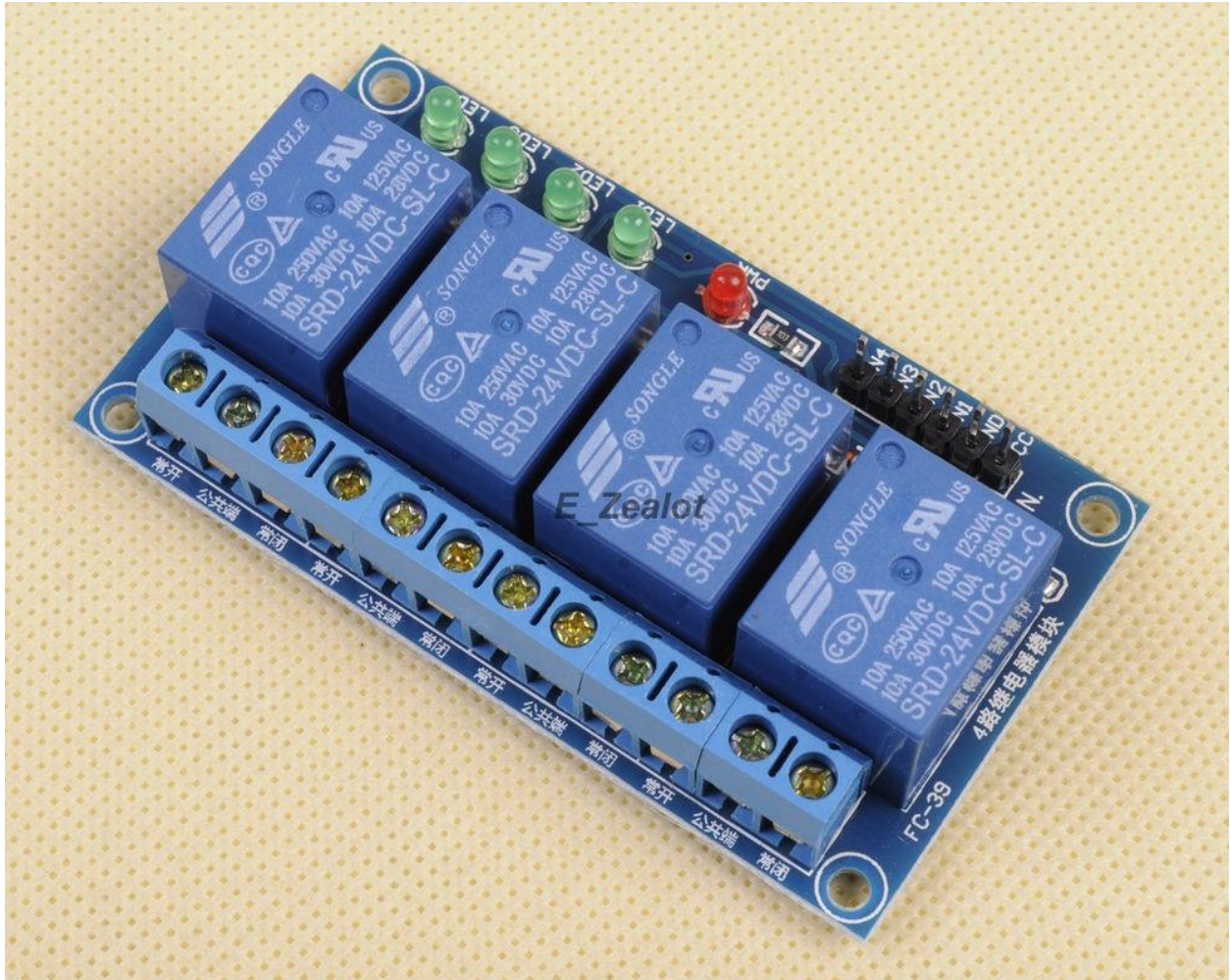


Figure 2.13 Relay Shield

- 1, 4 channel relay output
- 2, STM8S103F3 Microcontroller with sdk, it is programmable
- 3, 4 channel Optocoupler
- 4, 4 status LEDs
- 5, DC input (12V, work with power adapter)
- 6, RS485 port
- 7, Power LED
- 8, User LED
- 9, Reset key

General

Model	N/A
Quantity	1 piece

Form Color Blue
Material PCB

Specification

Max. Load AC:250V 10A, DC:30V 10A
Input Voltage 12V
English Manual / Spec Yes

Dimensions 3.07 in x 3.19 in x 0.79 in (7.8 cm x 8.1 cm x 2 cm)
Weight 2.89 oz (82 g)

Resistors used



Figure 2.14 10kohms resistors

Often it is useful to steer an input pin to a known state if no input is present. This can be done by adding a pullup resistor (to +5V), or a pulldown resistor (resistor to ground) on the input. A 10K resistor is a good value for a pullup or pulldown resistor.

Current limiting resistors



Figure 2.15 220 ohms resistor

Limiting current into an [LED](#) is very important. An LED behaves very differently to a resistor in circuit. Resistors behave linearly according to [Ohm's law](#): $V = IR$. For example, increase the voltage across a resistor, the current will increase proportionally, as long as the resistor's value stays the same. Simple enough. LEDs do not behave in this way. They behave as a diode with a characteristic [I-V curve](#) that is different than a resistor.

For example, there is a specification for diodes called the characteristic (recommended) forward voltage (usually between 1.5 4V for LEDs). You must reach the characteristic forward voltage to turn 'on' the diode or LED, but as you exceed the characteristic forward voltage, the LED's resistance quickly drops off. Therefore, the LED will begin to draw a bunch of current and in some cases, burn out. A resistor is used in series with the LED to keep the current at a specific level called the characteristic (or recommended) forward current.

CHAPTER 3

METHODOLOGY:

3.1SYSTEM OVERVIEW

When the system is powered on the Traffic lights will initialize and then start to follow the order discussed in chapter 2. Meanwhile the arduino nano will be polling all the devices to make sure they are available and ready. Red traffic light readies the cameras, such that any vehicle that passes on or proceed against the red light will have a picture of it taken by the respective camera. Induction proximity sensors will operate if a vehicle passes over them while the traffic light is red, which will trigger the camera to operate and take a picture. If the light is green the camera will not operate. The Python software then gathers the pictures and stores them in a folder of the manning computer. Then the monitoring personnel can view the pictures and use them within their context of jurisdiction.

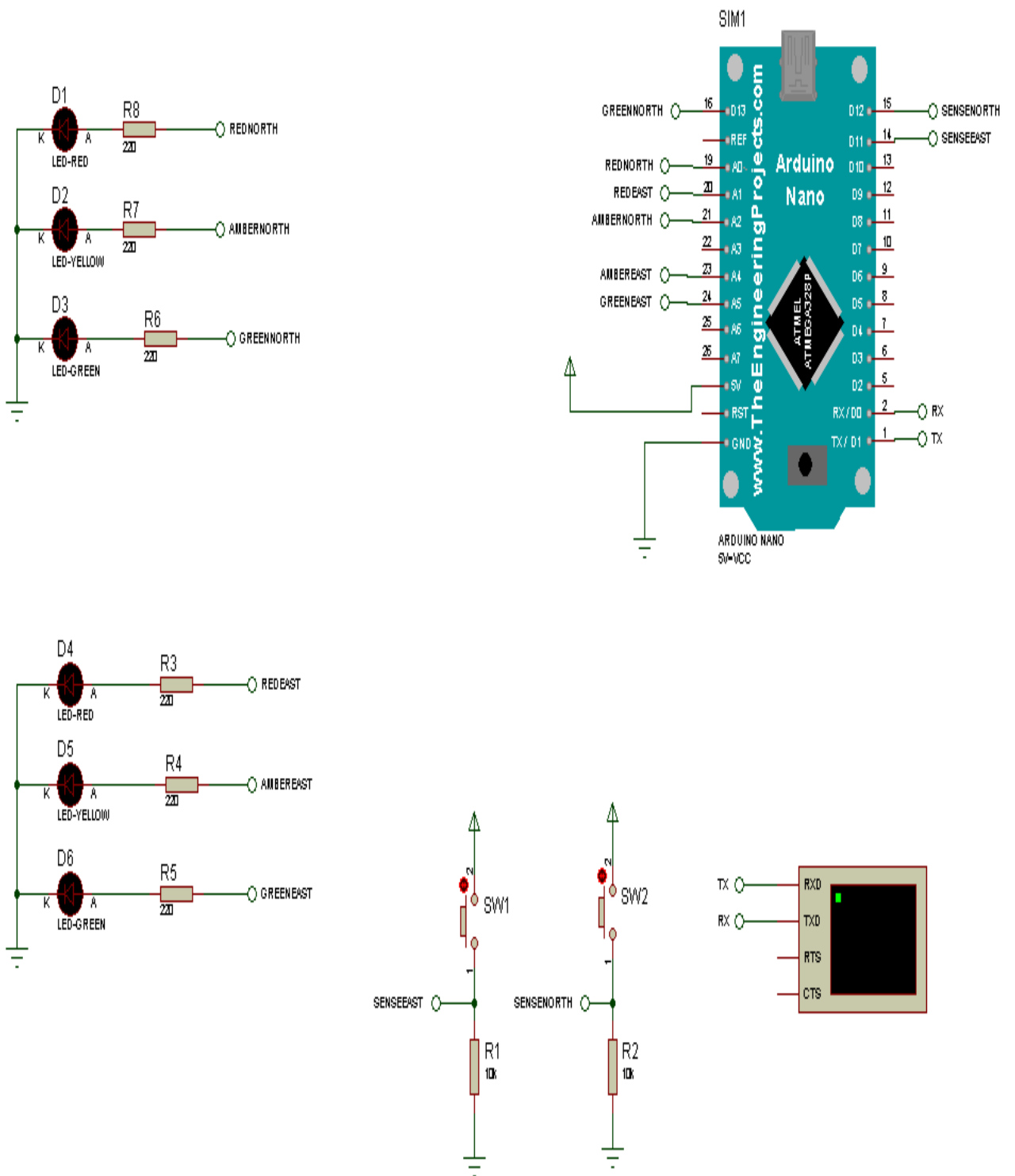


Figure 3.1 system design

Hardware design

A Printed circuit board was designed and etched where the 40 pin IC holder was placed where the arduino nano was mounted. The PCB has 6 x 220 ohms resistors, these resistors are current limiting resistors and are for limiting current that goes to the LEDs. The LEDs are used as traffic lights for the prototype. The resistors are connected between the Arduino nano and LEDs.

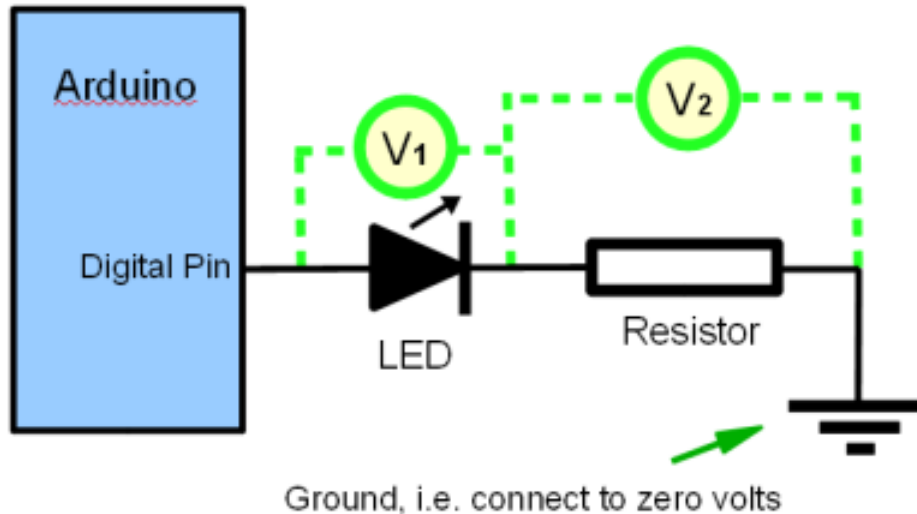


Fig 3.2 connection for current limiting resistor

Also connected to the Arduino nano on the PCB are 2x 10k ohms resistors, these were connected so as to work as pull-down resistors, at the input of the arduino for the two induction proximity sensors. The pull-down resistors are ensuring that the voltage state at input of the arduino is 0v and only rises to 5v if the sensors are interrupted by a vehicle passing through a red traffic light.

A voltage regulator 7805 is also mounted on the PCB with 2x 1000 micro farads capacitor to act as a stiff amplifier. The capacitors will add to power requirements at startup. A 24v power is used to power the circuit via the relay shield

The sensing unit

The induction proximity sensors operate whenever a vehicle passes over them while the traffic light is red. 4mm proximity sensors have been used. The range and direction of the sensors was properly chosen such that vehicle in other lanes are not detected. These sensors detect only metals such that a person moving above them will not be detected.

The figure 3.2 below shows the position in red where the sensors are positioned

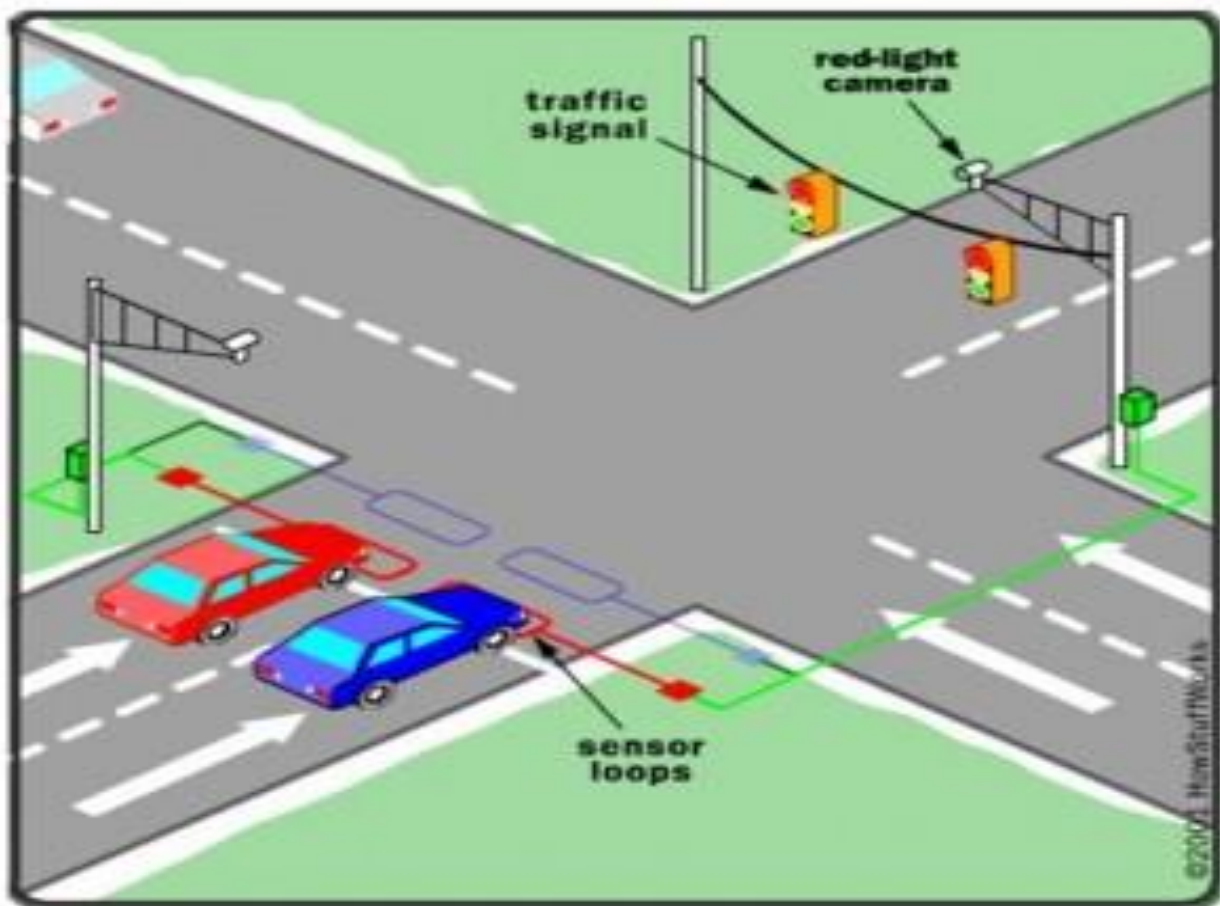


Figure 3.3 The position in red where the sensors are installed

Software design

The arduino was programmed using C language for the traffic lights as well as the sensing parts. The arduino is always polling to check the availability of the sensors, the functioning of traffic lights and cameras.

Python software was installed so as enable camera to communication with the computer. This is software interprets messages from the code to check which direction the vehicle is going. It also arrange where to save the images while taking care of the date and time.

Program

```
/
  GERALD MANGWAYA test programme.
  sends 'good', 'photoE', 'PhotoN' at random, with a skew
  towards sending 'good' more times
/
int i;
longrandNumber;
String state[] = {"good", "photoE", "good", "photoN"};
void setup() {
  randomSeed(analogRead(0));
  Serial.begin(9600);
}
void loop() {
  randNumber = random(4);
  Serial.println(state[randNumber]);
  delay(5000);
```


}

CHAPTER 4

RESULTS

4.1 Introduction

This chapter presents the results and consequent analysis of the automatic system performance tests are presented in Figures 4.1 and 4.2, The results are used to show the different individual tasks that have been combined together to build the automatic offence monitoring system at road intersection.

Test results

The traffic lights after initializing stabilizes and followed the pattern as shown below

Figures 4.1, 4.2, 4.3 and 4.4 below show screen captures of the ping tests conducted as outlined by the procedure in Chapter 3.2.

State	Light 1	Light 2	Alias
0	Red	Red	RR
1	Red	Green	RG
2	Red	Yellow	RY
3	Red	Red	RR
4	Green	Red	GR
5	Yellow	Red	YR

Results of Traffic lights

/ Order of Sequence:

- 0: GreenNorthRedEast,
- 1: AmberNorthRedEast,
- 2: RedNorthRedEast,
- 3: RedNorthGreenEast,
- 4: RedNorthAmberEast,
- 5: RedNorthRedEast

/

Results of violations on the monitor

Python 2.7 (r27:82525, Jul 4 2010, 09:01:59) [MSC v.1500 32 bit (Intel)] on win32

Type "copyright", "credits" or "license()" for more information.

>>> ===== RESTART =====

===

>>>

Gerald Mangwaya Automatic Traffic Light Violation Monitoring system

Enter the Port NUMBER only: 4

MAKE SURE your 'images' folder is present.

Press ENTER to continue...

test photos...

...done

Eastbound Violation!

Northbound Violation!

Eastbound Violation!

Eastbound Violation!

Northbound Violation!

Northbound Violation!

Northbound Violation!

Eastbound Violation!



Figure 4.1 pictures done by monitoring camera



Fig 4.5 pictures from monitoring cameras

4.3 Safety considerations

As observed the Traffic lights delay to allow vehicles to proceed as a safety measure just in case some vehicle is running late. The inductive coils are positioned such that they do not interfere with moving traffic. Power regulation has successfully protected the circuits from burning out

CHAPTER 5

CONCLUSIONS

5.1 Introduction

This chapter discusses the conclusions of this research and proposes recommendations for further research.

5.2 Conclusions

This project has been successfully presented a functional and low cost arduino-based traffic offence monitoring system at traffic controlled road intersection. The traffic light system works together with the sensing system which are all being controlled by arduonano , power section, relay shield and light emitting diode (LED). Then, for effective traffic monitoring, the Arduino software has been loaded and implemented usingan IC programmer using a mikrobasic program written in Basic C language. The developed traffic offence monitoring system at traffic controlled road intersection has been successfully tested by constructing a prototype that resembles the real application. The functionality of the prototype shows that the developed system can be used for a real life traffic control at road intersection. Also, developed system can be employed as a training kit in learning traffic light control and monitoring system.

5.2 Recommendations

Every good engineering design has limitation; the limitation of the developed system could be improved upon by incorporating a wireless network into the developed system [1]. This will add a lot of functionality such as monitoring traffic flow on the highway. Also, owing to the epileptic nature of power supply system, it is imperative to have the server for such a big project to be hosted by a recognized service provider, for which 99.9% of availability is assured.

The system can be further developed as to enforce the law on offenders through the use of image recognition and national vehicle database. Laws that authenticate use of red light cameras as acceptable evidence of need to be put in place

6. Appendix

6.1 Code for Traffic light

/

Author: Gerald Mangwaya

```
intdelaytime;
```

```
intRedNorth = A0;
```

```
intAmberNorth = A2;
```

```
intGreenNorth = 13;
```

```
intSenseNorth = 12;
```

```
intRedEast = A1;
```

```
intAmberEast = A4;
```

```
intGreenEast = A5;
```

```
intSenseEast = 11;
```

```
constintdelayTimes[6] = {10,3,1,10,3,1};
```

```
int i = 0;
```

```
intstartTime, endTime;
```

```
void setup() {
  Serial.begin(9600);
  pinMode(SenseEast, INPUT);
  pinMode(SenseNorth, INPUT);
  pinMode(RedNorth,OUTPUT);
  pinMode(AmberNorth,OUTPUT);
  pinMode(GreenNorth,OUTPUT);
  pinMode(RedEast,OUTPUT);
  pinMode(AmberEast,OUTPUT);
  pinMode(GreenEast,OUTPUT);

  //test the outputs

  for (int j =0; j<5; j++){
    digitalWrite(RedNorth,1);
    digitalWrite(AmberNorth,1);
    digitalWrite(GreenNorth,1);
    digitalWrite(RedEast,1);
    digitalWrite(AmberEast,1);
    digitalWrite(GreenEast,1);
    delay(500);
    digitalWrite(RedNorth,0);
    digitalWrite(AmberNorth,0);
    digitalWrite(GreenNorth,0);
    digitalWrite(RedEast,0);
    digitalWrite(AmberEast,0);
    digitalWrite(GreenEast,0);
```



```

delay(500);
}

loadTime();

// Serial.println("photoE");
// delay(500);
// Serial.println("photoN");
}

void loop() {
idle();

}

void idle(){
//the general method that runs the robots
//use i to increment the sequence
//Serial.print("End: "); Serial.println(endTime);
//Serial.print("i="); Serial.println(i);

sequence(i);
if ((millis()/1000 == endTime)) {
i++;
if (i > 5) {
i = 0;
}
}
}

```

```

loadTime();
}
//don't only check tresspass if a transition to a red phase has occurred
//if ((i==1) || (i==4)){
checkTresspass();
// }
delay(500);
}

voidloadTime(){
startTime = millis()/1000;
endTime = startTime + delayTimes[i];
}

voidcheckTresspass(){
//check if a car has tresspassed, if so, take photo, depending on the ordinate
inttresspassEast, tresspassNorth;
tresspassEast = digitalRead(SenseEast);
tresspassNorth = digitalRead(SenseNorth);
//if (tresspassEast || tresspassNorth) {
if ((i==0)&&(tresspassEast)) Serial.println("photoE");
else if ((i==3)&&(tresspassNorth)) Serial.println("photoN");
//}
else ;//Serial.println("good");
}

void sequence(intseqIndex){
digitalWrite(RedNorth,0);

```

```
digitalWrite(AmberNorth,0);  
digitalWrite(GreenNorth,0);  
digitalWrite(RedEast,0);  
digitalWrite(AmberEast,0);  
digitalWrite(GreenEast,0);
```

```
switch (seqIndex) {  
case 0:  
digitalWrite(RedNorth,0);  
digitalWrite(AmberNorth,0);  
digitalWrite(GreenNorth,1);  
digitalWrite(RedEast,1);  
digitalWrite(AmberEast,0);  
digitalWrite(GreenEast,0);  
break;  
.  
.  
.  
case 5:  
digitalWrite(RedNorth,1);  
digitalWrite(AmberNorth,0);  
digitalWrite(GreenNorth,0);  
digitalWrite(RedEast,1);  
digitalWrite(AmberEast,0);  
digitalWrite(GreenEast,0);  
break;  
}
```

