AUTOMATIC ULTRASONIC HEADWAY CONTROL FOR A ROBOTIC CAR WITH ARDUINO

Project report submitted in the Department of Applied Electronics and Instrumentation Engineering of Dr. B. C. Roy Engineering College, Durgapur in partial fulfilment of the requirement to award the degree of

BACHELOR OF TECHNOLOGY

Submitted by

TATHAGATA ROY

University Roll No. 12000512059

Under the guidance of

ARINDAM CHAKRABORTY

Associate Professor

Dept. of Applied Electronics and

Instrumentation Engineering

Dr. B. C. Roy Engineering College

Durgapur, India

DEPARTMENT OF APPLIED ELECTRONICS AND INSTRUMENTATION ENGINEERING

DR. B. C. ROY ENGINEERING COLLEGE

DURGAPUR, INDIA

2016

DECLARATION BY THE STUDENT

I hereby declare that the work mentioned in the report entitled **"Automatic Ultrasonic Headway Control for a Robotic Car with Arduino"** submitted at **Dr. B.C. Roy Engineering College, Durgapur, India** is an authentic record of my work carried out under the supervision of **PROF. ARINDAM CHAKRABORTY**. I have not submitted this work elsewhere for any other degree or diploma. I am fully responsible for the contents of our report.

Tathagata Roy

Tathagata Roy

(University Roll No.12000512027)

Department of Applied Electronics and Instrumentation Engineering

Dr. B. C. Roy Engineering College, Durgapur, India

Date: _____

DEPARTMENT OF APPLIED ELECTRONICS AND INSTRUMENTATION ENGINEERING

DR. B. C. ROY ENGINEERING COLLEGE

DURGAPUR, INDIA

2016

CERTIFICATE

This is to certify that the report titled "**Automatic Ultrasonic Headway Control for a Robotic Car with Arduino**", being submitted by **Tathagata Roy**for partial fulfilment of the requirement to award theB.Tech. Degree in Applied Electronics and Instrumentation Engineering, Dr. B. C. Roy Engineering College, Durgapur is a record of bonafide project work carried out by him under my guidance. This report is, in my opinion, worthy of consideration for the said purpose in accordance with the regulations of the **MaulanaAbulKalam Azad University of Technology, West Bengal.** This report has not been submitted to any other university or institution for the award of any other Degree or Diploma.

Head, AEIE Department
Dr. B. C. Roy Engineering College
DURGAPUR-713206

Dr. K. M. Hossain
Head of the Department
Dept. of Applied Electronics and
Instrumentation Engineering
Dr. B. C. Roy Engineering College
Durgapur, India

DEPARTMENT OF APPLIED ELECTRONICS AND INSTRUMENTATION ENGINEERING

DR. B. C. ROY ENGINEERING COLLEGE

DURGAPUR, INDIA

2016

SUPERVISOR'S CERTIFICATE

This is to certify that the report titled "**Automatic Ultrasonic Headway Control for a Robotic Car with Arduino**", being submitted by **Tathagata Roy**for partial fulfilment of the requirement to award theB.Tech. Degree in Applied Electronics and Instrumentation Engineering, Dr. B. C. Roy Engineering College, Durgapur is a record of bonafide project work carried out by him under my supervision and guidance. This report is, in my opinion, worthy of consideration for the said purpose in accordance with the regulations of the **MaulanaAbulKalam Azad University of Technology, West Bengal.** This report has not been submitted to any other university or institution for the award of any other Degree or Diploma.

ARINDAM CHAKRABORTY
Associate Professor
Dept. of Applied Electronics and
Instrumentation Engineering
Dr. B. C. Roy Engineering College
Durgapur, India

ACKNOWLEDGEMENT

This thesis owes its existence to the help, support and inspiration of several people. Firstly, I would like to express my sincere appreciation and gratitude to Prof. Arindam Chakborty for his guidance during my groundwork. His support has been precious for the development of this thesis content and the free space he has given us on our research works was invaluable.

I would never forget all the chats I shared with some of my friends and classmates through this project. Their advices were fundamental in shifting my project to a Arduino based one.

I am very grateful to all the people I have met along the way and have contributed to the development of my project. .

Finally, my deepest gratitude goes to my family for their unflagging love and unconditional support throughout my life and my studies and through this project work.

ABSTRACT

Intelligent Transportation Systems (ITS) have been an active area of research for last few years. The modern high speed traffic flow suffers the serious adverse affects of human drivers' that exhibit slower response time and errors in judgment. These types of errors can be reduced or eliminated from the driving experience by introducing computer control systems into the automotive arena.

The basic purpose of this project is to develop a scale model platform for the rapid prototyping and testing of ITS systems and technologies. Specifically, this body of work was concerned with the development of an automatic headway control system that utilized ultrasonic sensors. This control system was intended to automatically maintain headway distance in an effort to create an adaptive cruise control system for this scale model vehicle. Implementation of such systems could conceivably reduce driver fatigue by removing the burden of maintaining safe following distance from the driver.

System dynamics of car-like robots were employed in this research to create a controller for an autonomous path following vehicle. In this project we have built a SERVO CAR with a front castor wheel and two side wheels built to fit in servo motors. The SERVO CAR rear wheels are powered by two servo motors with 360° complete rotation. It is programmed by a ARDUINO-UNO microcontroller interfacing equipped with a ATmega168 microcontroller in it.

The Servo car measures any obstacle in front of it by an Ultrasonic Sensor powered by the Arduino. When it detects no obstacle in its 7cm front radius then it moves forward and when it detects any obstacle in front of its 7cm radius then it moves left or right, as programmed.

Following the development of these, the system was applied to the scale model vehicle to assess the practical use of the system on a mock highway. A satisfactory result is produced after testing was completed, and the application of such systems to scale model platforms is feasible.

Contents

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

When the internal combustion engine and later the automobile, were first introduced to the public, no one could have foreseen the extent to which they would influence daily life. Today, with the information age in full swing, it is still hard to believe the way that computers and other information technology have permeated people's lives. Now it seems only natural to expect information technologies to enhance the way we view automobiles.

People now take for granted automobile systems like emission control and fuel injection. In fact, many people do not realize how many systems inside their automobile are already monitored and controlled by computers [7]. Fuel delivery, ignition, emissions, air-conditioning and automatic transmission systems are examples of the systems used daily by a car that are computer controlled or assisted. Now, in the information age, people have come to rely on other driver assistance technologies, such as mobile phones and in-vehicle navigation systems. The goal of the technologies is to make the experience of driving less burdensome, especially on long trips. In-vehicle navigation systems have become more complex as they become more accepted, and instead of simply providing up to date maps and directions, some systems can now let drivers know about points of interest, restaurants, and fuel stations, for example.

With the advent of new automotive and computer technologies, new models of automobile behaviour, and robust control algorithms and technologies, we have begun to increase driver safety and comfort by adding systems like air bags, anti-lock braking systems and cruise control that are all controlled electronically [3][5][6].

Research groups like the Program on Advanced technology for the Highway (PATH) at the University of California, Berkeley and the European PROMETHEUS project have been studying systems for automobiles that increase a vehicle's overall traction , such as anti-skid braking, alternatives to the internal combustion engine , and methods for making the highway itself part of the driving experience [1][2]. Programs like PATH have been researching methods for centralized control of the highway system, integrating roadside technologies [11][4][12][13]. In this manner, the traffic systems and the automobiles work together to bring passenger safety on the road. The ultimate goal of such technology is to enable cars to perform what has termed as "platooning" [1]. In platooning, automobiles can be grouped together at highway speeds, 65-70 MPH, no more than a few feet apart which makes better use of available roadways[1][2][4][13].

Automatic headway control , also known as adaptive cruise control, is an area tat has been under research for some time[3][4][5][6][10].Adaptive cruise control reduces driver strain by the need to monitor the vehicle's speed. It can also be coupled with other driver convenience systems to provide to provide control for lane changing, obstacle detection and collision avoidance to further reduce driver fatigue when travelling long distances.

Automatic headway control is one of the topics that readers would likely to find in the literature concerning ITS today [18][19][20][22][23]. Cruise control, anti-lock braking systems, collision warning,

air bags and other driver safety systems all fall into the area of research [1][2]. ITS research is also concerned with the development of systems that monitor and help to guide vehicles as well. Traffic signals, traffic cameras, road sensors, central control towers, inter-vehicle communication systems and even the composition of road the road surface and lane marking materials are also in the domain of ITS[1][2][11]. Major ITS goals are to provide safer and more convenient transportation to the general public, and to utilize transportation resources more effectively.

In this project work we are dealing with real time obstacle detection system mounted vehicle which will be detecting obstacles in front of it by a ultrasonic sensor. The obstacle detection in real time is the most versatile and challenging task for road vehicles and passenger safety.

2. JUSTIFICATION

We were working on Robust Control earlier. Due to Complications we changed our project to PC based DSO. This could be used for oscilloscope operations of two inputs simultaneously on a frequency range up to 1.2MHz. It could also save the output screenshots in the drive of the PC or any directed flash drive provided by the user. But due to unavailability of the components the project can't be implemented.

After this project cancellation we learned that before engaging ourselves in any future projects we have to do a very widely calculative ground work. This includes confirmation of the source that the project will run properly, checking the availability of the components in the market, and also checking for the softwares that will be necessary for the project to run.

Then after some researches we got our hands on some microcontroller projects. For implementing microcontroller projects one have to learn the particular batch programming of the specific kind and the input to the microcontrollers by the numbering codes through the PIC Start Microcontroller kit is not easy to implement and prone to errors.

Then we came by this project Automatic Ultrasonic Headway Control by Arduino. Though here Arduino processor, Ultrasonic Sensor, Servo Motor is used this is very easy to implement and the Arduino used is also open sourced so the Arduino Sketch can be done by anybody as one don't have to learn every ins and outs of C language to do this project. Then we decided to work on this project.

3. MOTIVATION

Every work needs a motivation to drive it from the imagination phase to success. No work is self-sufficient without its motivation.

When we were researching about this project we got amazed by what this ultrasonic headway control car can be used for. The ultrasonic car can be used for collision avoidance to keep the number of accidents low.

Here we had to work with a Ultrasonic sensor(HC-SRO4) which is used for non-contact distance measurements ranging from 2cm to 3meter with a required supply voltage of +5v dc[4.3]. This car is powered by two servo motors with a torque of 5.5Kg/cm (4.8v) and 6.5Kg/cm (6v), operating voltage of 4.86v in which there

is a DC motor, a regulatory circuit, a potentiometer and the gear mechanism [4.1]. The processer is a Arduino microcontroller Board.

4. THEORY

Components are given in Appendix [1]

4.1 SERVO MOTOR

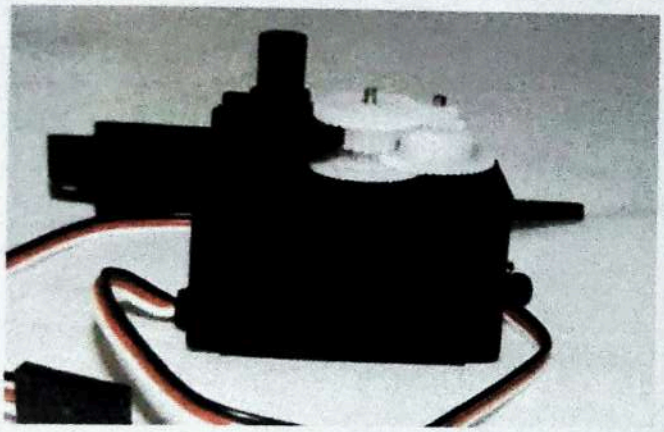


Figure 4.1.1 A Servo Motor with all gears

Servo motors are small controllable motors that lend to implementation in many applications. There are servos with many different speeds, sizes and torques. But all have 3 wires i.e. positive, ground and signal [8]. Control of Servo motors with microcontroller is universal for all models. Servo receive pulse width modulated signals to determine in which manner to move. There are many ways to send this signal to the motor. This application covers how to send the desired PWM signal to the Servo Motor using the Arduino Uno R3 microcontroller for the precision angular movement. The Arduino uno is a popular microcontroller which comes on a development board to accelerate programming, provides simple interfacing with peripheral devices and connection with the computers [8] [9] [17].

The simplicity of a servo is among the features that make them so reliable. The heart of a servo is a small DC motor. This motor runs on electricity from battery connected to the servo and spins at high RPM but put out a very low torque (twisting force used to work). An arrangement of a potentiometer, IC and an arrangement of gears [shown in figure 4.1.1] is inside the servo to take the high speed of motor and slows it down while at the same time increasing the torque.(Basic Law of Physics: Work=Force x Distance).Gears in an inexpensive servo are generally made of plastic to keep it lighter and less costly. On a servo designed to provide more torque for hard work, the gears are made of metal, are much harder to damage [9].

With a small DC motor it takes its power from a battery and the motor spins. Unlike a simple DC motor, however, a servo's DC motor shaft is slowed way down with gears. A positional sensor on the final gear is connected to a small circuit board. The sensor tells the circuit how far the servo output shaft has rotated. The electronic input signal from the computer or the radio also feeds into that circuit board. Electronics on the circuit board decode the signals to determine how far the user wants the servo to rotate. It then compares the desired position with the actual position and decides in which direction to rotate the shaft so it gets its desired position [9].

Specifications

Servo motors are controlled through the control line, usually a yellow or white. The figure below shows how different pulse widths correspond with different position of motor. When the pulse width is less than 1.5ms the motor will move to the 0 positions and hold. When the pulse width is 1.5ms the motor will rotate to the 90 degree position and if the pulse width is

greater than 1.5ms the motor will rotate to the 180 degree position [shown in figure 4.1.2] [8].

When the motor reaches the desired position it will hold until a signal is sent to rotate. This is done in this application using the Arduino 1.0 coding software.

Table 4.1.0 Servo Motor Features

SG-5010 SERVO	Specification
Weight	47g
Dimension	40.6×20.5x38mm
Stall torque	5.5kg/cm(4.8v) 6.5kg/cm (6v)
Operating speed	0.19sec/deg(4.8v) 0.15sec/deg(6.0v)
Operating voltage	4.8-6v
Temperature range	0°C --55°C
Gear Type	Nylon gear

The PWM output pins on the development board can be written to with different pulse widths which are used to control the motor. Here in this project work we were working on **SG-5010** servo motor features of which have been discussed into **Table 4.1.0.** In this servo motor the brown wire is for ground, the red wire is for positive power supply and the orange is used for receiving PWM signals.

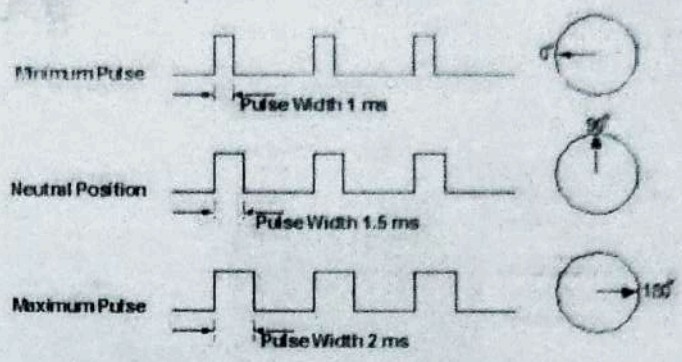


Figure 4.1.2 Servo Motor Operations

4.2 ARDUINO UNO R3

The Arduino microcontroller board is an easy to use yet powerful single board computer that has gained considerable traction in the hobby and professional market. The Arduino is open-sourced which means which means the hardware is reasonably priced and the development software i.e. Arduino IDE is free.

The Arduino project was started in Italy to develop low cost hardware for interaction design. It is not a chip (IC), it is not a board (PCB), it is not a company or a manufacturer, it is not a programming language and it is not computer architecture, although it involves all of these things.

Founded by **Massimo Banziand** and **David Cuartielles,** in 2005, it is based on "Wiring Platform", which dates to 2003, open-source hardware platform, open source development environment, easy to learn language and libraries (based on wiring language) , integrated development environment (based on Processing

programming environment) and it is available for Windows/Mac/Linux [26].

With the Arduino Board, one can write programs and create interface circuits to read switches and other sensors, and to control motors and lights with very little effort. The figure of Arduino Uno R3 board has shown below [figure 4.2.1].

Figure 4.2.1 Arduino Uno R3

TABLE 4.2.1 Hardware Specifications of Arduino Uno R3

Operating Voltage	5v
Input voltage	7v-12v
Input Voltage(limits)	6v-20v
Digital I/O pins	14(of which 6 provide PWM outputs)
Analogue Input pins	6
DC current per I/O pin	40mA

Board Mapping of the Arduino

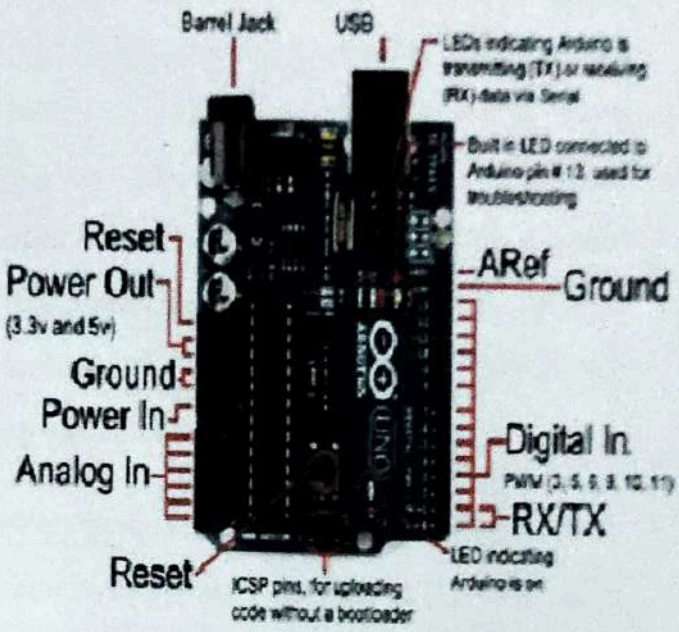


Figure 4.2.2 Board Mapping of Arduino

The Arduino board consists of a Atmel Atmega 328 microprocessor [Appendix-2].

PIN DETAILS

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 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 [Fig.4.2.2].

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[15][16][17][26].

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). One 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 50mA.

GND: Ground pins

MEMORY

The Atmega328 has 32 KB of flash memory for storing code (of which 0.5 KB is used for the boot loader). It has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library). The 0.5 KB of boot loader memory used for Arduino is defined at the architecture.

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 40mA 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.

PWM: 3, 5, 6, 9, 10, and 11: Provide 8-bit PWM output with the analogueWrite () 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.

[**DigitalRead**: Get a HIGH or LOW reading from a pin already declared as an input.

DigitalWrite: Assign a HIGH or LOW value to a pin already declared as an output.

AnalogueRead: Get a value between or including 0 (LOW) and 1023 (HIGH). This allows getting readings from analogue sensors or interfaces that have more than two states.

AnalogueWrite: Assign a value between or including 0 (LOW) and 255 (HIGH). This allows setting output to a PWM value instead of just HIGH or LOW.

PWM: Stands for Pulse-Width Modulation, a method of emulating an analogue signal through a digital pin. A value between or including 0 and 255, used with AnalogWrite.]

The Uno has 6 analogue inputs, each of which provides 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 AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

I2C: 4 (SDA) and 5 (SCL): Support I2C (TWI) communication using the Wire library.

AREF: Reference voltage for the analogue inputs. Used with analogReference().

Reset: Bring this line LOW to reset the microcontroller.

COMMUNICATION

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital **pins 0 (RX) and 1 (TX)**. An 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, an *.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 Uno's digital pins[17][26].

The ATmega328 also support I2C (TWI) and SPI communication. To use the SPI communications see datasheet [App.2].

PROGRAMMING

The Arduino programming language is a simplified version of C/C++. If one knows C, programming the Arduino will be familiar. An important feature of the Arduino is that one can create a control program on the host PC.

The Arduino Uno can be programmed with the Arduino software. Select **"ArduinoUno/ATmega328"** from the **Tools > Board** menu (according to the microcontroller on board).

The ATmega328 on the Arduino Uno comes pre-burned with a boot loader that allows one to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol.

One can also bypass the boot loader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header.

AUTOMATIC 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 is connected to the reset line of the ATmega328 via a 100nanofarad 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, we have to make sure that the software with which it communicates waits a second after opening the connection and before sending this data. The programming will be running automatically each time the Arduino will be connected.

USB OVERCURRENT PROTECTION

The Arduino Uno has a resettable polyfuse that protects computer's USB ports from shorts and overloading. Although most computers provide their own internal protection, the fuse provides

an extra layer of protection. If more than 500mA 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. Three screw holes allow the board to be attached to a surface or case. The distance between digital pins 7 and 8 is 160 mil (0.16") , not an even multiple of the 100mil spacing of the other pins.

When operating on Digital data:

Digital Input/output uses the digital pins, but AnalogIn pins can be used as digital one.To receive a digital signal use digitalRead (pinNumber).and to send a digital signal use digital Write(pinNumber,value) Digital input and output are always either HIGH or LOW.

When operating on Analog data:

Analog input uses the AnalogIn pins, analog output uses the PWM pins.To receive an analog signal use analogRead(pinNumber).To send a PWM signal use analogWrite(pinNumber, value). Analog input values range from 0 to 1023 (1024 values because it uses 10bits).PWM output values range from 0 to 255 (256 values because it uses 8 bits)[15][17][26].

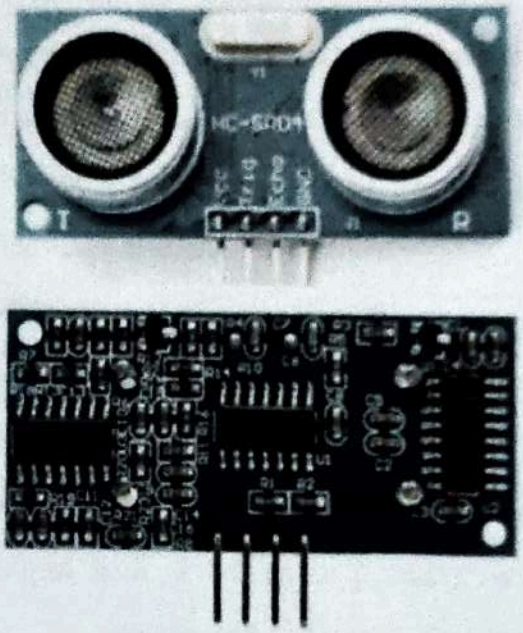
4.3ULTRASONIC SENSOR(HC-SRO4)

Ultrasonic sensor is non-contact distance measurement module. It is perfect for measuring distance between two moving or non-moving bodies. Ultrasonic sensors generate high frequency sound waves and evaluate the echo which is received back by the sensor. Sensors calculate the time interval between sending the signal and receiving the echo to determine the distance to an object. The sensor is essential for measuring distance [App. 3]. However dividing the relative distance between the two sensors by the detection of the time difference, the distance and speed of the object is calculated. The HC-SRO4 has a operating range of 2cm to 400cm[Shown in **Table 4.3.1**]. Its operation is not affected by sunlight or black material like sharp rangefinders.(although acoustically soft materials like cloth can be difficult to detect). It comes complete with ultrasonic transmitter and receiver module [25].

Table 4.3.1 Features of Ultrasonic Sensor

Power Supply	:+5V DC
Quiescent Current	<2mA
Working Current	15mA
Effectual Angle	<15°
Ranging Distance	2cm – 400 cm/1"– 13ft
Resolution	0.3 cm
Measuring Angle	30 degree

Figure 4.3.1 Ultrasonic Sensor



THEORY

Input Trigger Pulse	t_{OUT}	2 µs (min), 5 µs typical
Echo Holdoff	$t_{HOLDOFF}$	750 µs
Burst Frequency	t_{BURST}	200 µs @ 40 kHz
Echo Return Pulse Minimum	t_{MIN}	115 µs
Echo Return Pulse Maximum	t_{MAX}	18.5 ms
Delay before next measurement		200 µs

Table 4.3.1 Output of Ultrasonic Sensor

A short ultrasonic pulse is transmitted at the time 0, reflected by an object. The senor receives this signal and converts it to an electric signal. The next pulse can be transmitted when the echo is faded away. This time period is called cycle period. The recommend cycle period should be no less than 50ms. If a 10µs width trigger pulse is sent to the signal pin, the ultrasonic module will output 40 kHz ultrasonic signal and detects the echo back. The measured distance is proportional to the echo pulse width and can be calculated by the **formula** below. If no obstacle is detected, the output pin will give a 38ms high level signal [See Table].

FORMULA

$$Pulse\ width/58 = distance\ (cm)$$

$$Pulse\ width\ /148 = distance\ (inch)$$

The remaining components will be discussed as we will mount up the project.

5. HARDWARE ENVIRONMENT

5.1 Chassis

To assemble Ultrasonic Servo Car at first we have to take a servo car chassis with slots for two servo motors and a castor wheel. The Arduino board is mounted up and to avoid short circuit with the chassis a piece of cardboard is used of the size of Arduino Board under it. The nuts and bolts are tightened at the four holes of the Arduino.

After attaching the Arduino two servo motors are attached to the chassis and then the wheels are attached to the servo motors. The wheels are specially designed for servo motor.

Two 9V batteries are attached under the chassis of the car by double sided foam tape. A battery is for powering the two servos and another for powering the Arduino Board through which the Ultrasonic Sensor would get the power. The battery for the Arduino is connected with a DC jack.

For connection of **Ultrasonic sensor** a mini bread board has to be stickled at the front portion of the chassis with foam tape. After that the Ultrasonic Sensor has to be attached in the Bread Board facing the front direction.

5.2 Ultrasonic Sensor

The connection of Ultrasonic Sensor is shown on the next page.

5.3 Servo Motors

The Servo Motors' cables are attached with the mini bread board with Bug Strips. Wire from the positive terminal of battery and made available at two terminals of the bug strip and the negative terminal is made available at another two terminals of bug strips through Bread Board.

For **connecting** the Servo Motor -

The rightmost pin is Brown which is Ground and has to be connected to the Negative terminal of the battery. The middle pin is Red which is Positive and has to be connected to the Positive terminal of the battery. The leftmost pin is Orange which is used for receiving the PWM signals from the Arduino. For sending PWM signals to Arduino pin no. 9 and 10 are used for right Servo and Left Servo respectively.

The connections for Ultrasonic Sensor

The connections of the Ultrasonic Sensor are done by male to male cables with the Arduino Uno. The sensor is mounted on a small bread board and then all the connections are done. The trigger pin is attached with the pin 8 of the digital pins for input(Green wire in the Fig) and echo pin is attached with the pin 7 of the digital pins for output to the Arduino(Yellow wire).

The 5v pin of the Arduino is attached with the Vcc pin of the Ultrasonic Sensor

and Gnd pin is attached with the ground pin of the Arduino.

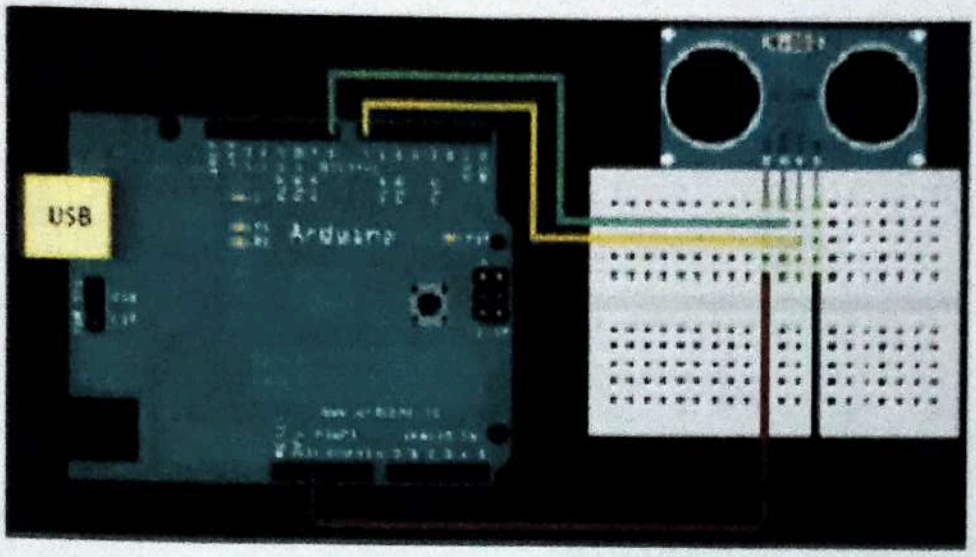


Figure5.2.1Connections- Ultrasonic Sensor

The connections for Servo Motor

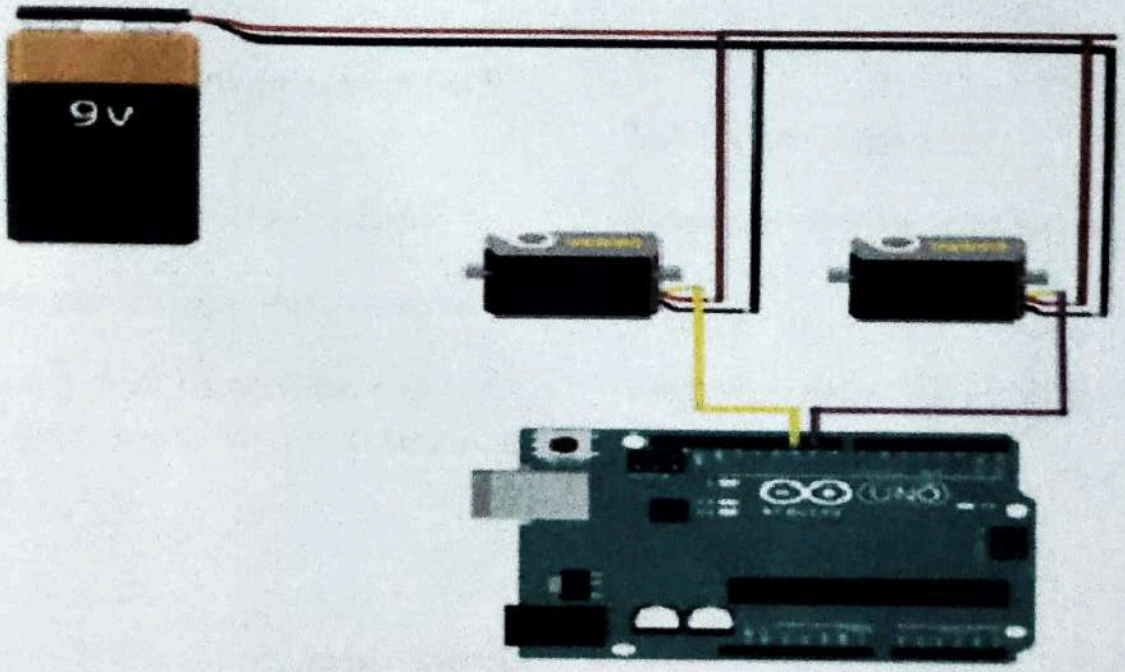


Figure 5.3.1 Connections of Servo

The rightmost pin is Brown which is Ground and has to be connected to the Negative terminal of the battery. The middle pin is Red which is Positive and has to be connected to the Positive terminal of the battery. The leftmost pin is sending PWM signals to Arduino pin no. 9 and 10 are used for right Servo(violet wire) and Left Servo(yellow wire) respectively[Shown in fig. 5.3.1]

Arduino software needs to be installed at PC .After that the sketch needs to be done.

In Arduino void setup() will be executed only when the program begins (or reset button is pressed).void loop() will be executed repeatedly.

6.1 Programming in Arduino.cc

[The discussions after // can't be traced by Arduino ,not included in programme]-

```
// Automatic Headway Control for Ultrasonic Servo based car

#include <Servo.h>  //  servo library is added

#define echoPin 7 //defined as output pin for the ultrasonic sensor

#define trigPin 6 //defined as input pin for sensor

Servo leftservo; //left servo declared

Servo rightservo; //right servo declared

void setup()  //will be executed only when the program begins or reset button is pressed

{

Serial.begin(9600);  // Serial comm.. between Arduino & PC at 9600bits/sec

pinMode(trigPin,OUTPUT); //Pin 6 i.e. trigpin will be triggering Ultrasonic Pulses)

pinMode(echoPin,INPUT);  //Pin 7 i.e. echopin will be receiving Ultrasonic Pulses

leftservo.attach(9); //leftservo isattached to pin 9

attached to pin 10

}

void loop()//

{

int distance,duration;

digitalWrite(trigPin,HIGH);  //set the trigpin high

delay(200);  // delay for 2 seconds

digitalWrite(trigPin,LOW); // set the trig pin low

duration=pulseIn(echoPin,HIGH);

distance=(duration/2)/29.1; // formula to evaluate distance

Serial.println("");

Serial.println(distance);

if(distance >7) // for going forward

{

leftservo.write(360); // left wheel forward

rightservo.write(-360);  // right wheel forward

}

else // obstacle detected turn left

{

{

leftservo.write(-360);  //left wheel backward

rightservo.write(-360);  // right wheel forward

}

}

}
```

6.2 Sketch Details

In the programming part of this Servo Car, as we are programming the Servo Motor here so we had to add the Servo Library here in this programme as #include <Servo.h>.

For the Ultrasonic Sensor we had to define pin number 7 as echo pin and pin number 6 as trigger pin. We were also need to declare our Servo motors.

Coming to void setup (), Serial.begin(9600) i.e. the serial communication between the Arduino and the PC at 9600 bits/sec. The trigpin i.e. pin 6 is declared as output and the echo pin i.e. pin 7 is declared as input pin. So the trigger pin will be triggering Ultrasonic pulses and the Echo pin will be receiving Ultrasonic responses from the obstacle. And we will be attaching the left servo to pin number 9 and the right servo to pin number 10.There are five lines in the void setup.

Coming to the void loop() we were using two variables distance and duration, these are the two components of Ultrasonic Sensor .At first we switched our trigger pin high for 200 millisecond and again low. These pins are emitting Ultrasonic pulses for 2 second high and 2 second low. The Arduino will record the pulses in distance in the form of Centimetres and the distances will be recording in the serial monitor.

In the if loop if the distance is greater than 7 then there will be no movement and the car will be moving forward i.e. the leftservo will move the wheels in the anti-clock direction and the rightservo in the clock direction else if any obstacle is detected within the 7 centimetre radius servos will be moving both in clock-wise direction.

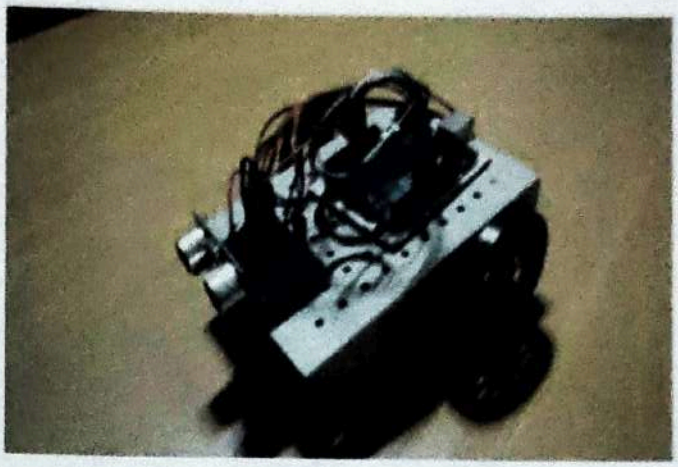


Figure6.1The Project Model

As by the picture above we can see that the Ultrasonic Sensor is in front, will trigger Ultrasonic pulses and will be receiving by the Ultrasonic Sensor. It will send the data to the Arduino. If the distance of any obstacle from the car is greater than 7cm then the car will go forward and if the distance is less than 7cm then the Arduino will instruct the car to turn left or right as programmed .Thus the car will avoid collision.

7.PROCEDURES

7.1 ARDUINO

Before running the Arduino we have to plug-in the Arduino Board using cable, provided with the kit. When the USB driver is not recognised then has to be navigated to and has to select the appropriate driver from the installation directory. Now the Arduino can run[26]. To **select the board** the following has to be after opening the Arduino software-

To select the **serial port** the following needs to be done-

TOOLS > SERIAL PORT > COM9

Elements of Arduino Software-

Text editor which deals with syntax and keyword colouring, automatic indention, programming shortcuts. Compiler, Hardware Interface which uploads programs and communicates with Arduino via USB[26].

7.2 CHASSIS

For attaching the Servo and the Arduino nuts, bolts and screw driver are used. For attaching the mini Bread Board and batteries foam tape is used. For DC jack , to solder the ground and the power **Soldering iron** is used.

8. HARDWARE COMPLICATIONS

The only problem we have faced was during the Arduino sketch compilation. It was showing error when we were uploading any programme to the Arduino. So we started to look for error messages. After searching on the error message we got that we have to upload a driver to get rid of the error messages .After doing that there isn't any error and we have successfully compiled the programmed[26].

9. IMPLIMENTATION

The servo car is moving successfully and whenever it is detecting any obstacle in front of it is turning left or right. One can observe the outputs of the Ultrasonic Sensor by opening the serial monitor in the Arduino software and see that it is detecting its distance from any obstacle in front of it.

10. Expenses

SL. NO.	COMPONENTS	PRICE
1.	SERVO CAR CHASSIS	100
2.	CASTOR WHEEL	30
3.	SERVO MOTORS	1000
4.	WHEELS	20
5	ARDUINO UNO R3	550
6.	ULTRASONIC SENSOR	160
7.	ONE 9V BATTERIES	20
8.	ONE 9V BATTERY CONNECTOR	5
9.	A DC JACK	5
10.	MINI BREADBOARD	20
11.	MALE TO MALE CONNECTOR	10
	TOTAL	**1955**

Table 10.1 Expenses

From the computer simulations and the application of the developed adaptive cruise control laws to the FLASH vehicle we can conclude that automatic ultrasonic adaptive cruise control for a scaled robotic car is realizable.

Practically speaking, this controller was designed around operation of the FLASH scale model vehicle under road conditions that simulated suburban highway systems meant for use at speeds between 20MPH and 30MPH. The test vehicle was able to handle these conditions satisfactorily.

12. SCOPE OF FUTURE WORK

This is just a prototype project. So the Ultrasonic Sensor is not good enough to detect obstacles after a certain speed and if it detects and tries to turn right or left then it can topple off. So a brake has to be added in this prototype to avoid that kind of incidents. If a high quality Ultrasonic Sensor is used then the reaction time will be less and then the Servo Car can determine things more easily and faster.

COMPONENTS –

1) A car chassis (ideal for attaching servo motors)
2) Castor wheel(for front support)
3) Two Servo motors
4) Wheels
5) Arduino Uno R3 and a USB connector
6) Ultrasonic sensor(HC-SRO4-4 pin configuration)
7) Two 9v batteries
8) Two 9v battery connectors
9) A DC jack
10) A mini Bread board
11) Male to male connector pins
12) A piece of cardboard
13) Double sided foam tape
14) Screws
15) Bug strips

Appendix-2

Operating Voltage	5 V
RAM	2Kb
Storing parameters	1 Kb of EEPROM
Storage	32 Kb of flash memory
Clock speed	16 MHz
Architecture	AVR 8-bit RISC
Package	DIP

Table1 ATmega328 features

ATmega Pin details

PIN NO.	DETAILS
1	Reset
2	Digital pin 0
3	Digital pin 1
4	Digital pin 2
5	Digital pin 3(PWM)
6	Digital pin 4
7	VCC
8	GND
9	Crystal Oscillator pin
10	Crystal Oscillator pin
11	Digital pin 5 (PWM)
12	Digital pin 6(PWM)
13	Digital pin 7
14	Digital pin 8
15	Digital pin 9(PWM)
16	Digital pin 10(PWM)
17	Digital pin 11(PWM)
18	Digital pin 12
19	Digital pin 13
20	VCC
21	Analogue reference
22	GND
23	Analog Input 0
24	Analog Input 1
25	Analog Input 2
26	Analog Input 3
27	Analog Input 4
28	Analog Input 5

Table 2 Atmega pin Details

ATmega 328 PIN MAPPING

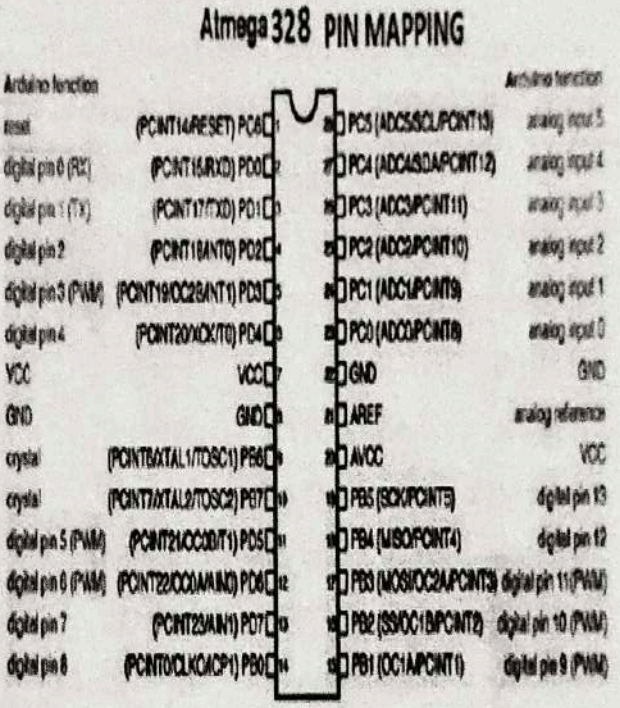


Figure1 Pin Mapping of Atmega 328

Appendix 3

Ultrasonic Sensor Pins

1. **VCC**: +5VDC
2. **Trig**: Trigger (INPUT)
3. **Echo**: Echo (OUTPUT)
4. **GND**: GND

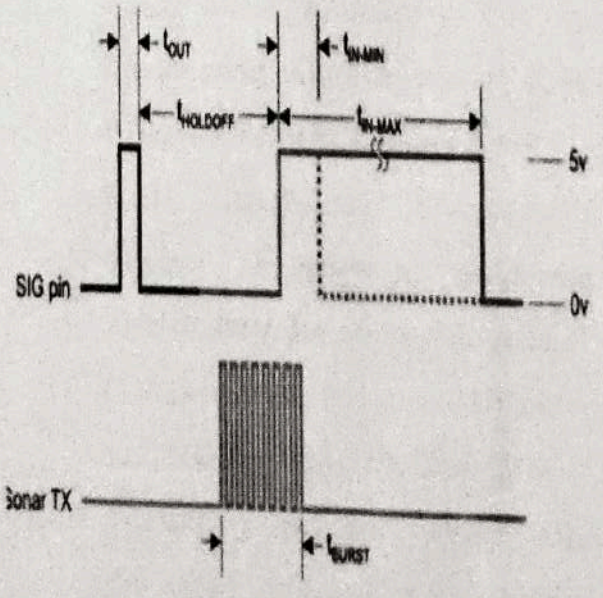


Figure2 Operations of Ultrasonic Sensor

1. Jesus Mena, *Finding the PATH to Automated Highways.* Berkeley Engineering- Forefront , US Berkley, Berkeley, CA 1990
2. Steven E. Shaladover et al., *Automatic Vehicle Control Developments in the PATH Program.* IEEE Transactions on Vehicular Technology, Vol.40, p.114-130, IEEE, Piscataway, NJ, 1991.
3. D. H. McMahon et al., *Vehicle Modelling and Control for Automated Highway Systems. Vehicle Dynamics Laboratory.* University of California, Berkeley. Berkeley, CA.
4. Jingsheng Yan. *Platoon Modal Operations Under Vehicle Autonomous Adaptive Cruise Control Model.* Thesis , Virginia Tech. 1994
5. Mingdong Yao. *The Development of Automatic vehicle Headway Control Low and a Simulation Tool.* Thesis, Virginia Tech. 1996
6. Ming Lu. System Dynamics Model for Testing and Evaluating Automatic Headway Control Models for Trucks Operating on Rural Highways. Thesis, Virginia Tech. 1996
7. Bosh. *Automatic Handbook.* Third Ed., Robert Bosh BmgH, 1993 by Simon Monk
9. Control of a 180 Servo Motor with Arduino UNO Development Board by Eric Mitchell
10. Caudil, R. J., Gerrard, W.L., *Vehicle Follower Longitudinal Control for automated Transit Vehicles.* Transactions ASME Journal of Dynamic Systems, Measurement, and Control, Vol. 99 , December 1977, pp. 241-248
11. S. E. Shaladover. *Dynamic Entrainment of Automated Guideway Transit Vehicles. High Speed Ground Transport., vol.12. No. 3, pp 87-113, Fall 1978.*
12. S. E. Shladover and R. E. Parsons . *Safety Issues for Intelligent Vehicle/Roadway Systems. Presented at ASME* Winter Ann. Meet., San Francisco, CA in Engineering Applications of Risk Analysis, ASME, and Dec.1989.
13. S. E. Shaladover. *Longitudinal Control of Animated Guideway Transit Vehicles within Platoons. J.* Dynamic Sys., Meas., and Contr., vol. 100, pp. 302-310, Dec. 1978
14. Pushkin Kachroo. *Setup* for *Advanced Vehicle Control Systems Experiments in the Flexible Low-cost Automated Scaled Highway (FLASH) Laboratory. SPIE's Photonics East Symposium, Mobile*

15. W.Durfee,*Arduino Microcontroller Guide*, University of Minnesota

16. Hans-PetterHalverson, *Introduction to Arduino -An Open-Source Prototyping Platform*

17. Introduction to the Arduino microcontroller -Hands-on Research in Complex Systems - Shanghai Jiao Tong University

18. Scott Mememer, *Cruise Control with a Brain, Edmunds Motor New. http://www.howstuffworks.com, Sep. 2, 2001.*

19. Automotive Wire. *TRW To Show Adaptive Cruise Control System That Automatically Adjusts Vehicle speed, Enhance Driving Comfort and Convenience.* Accident Reconstruction Source, Mar.2, 2000.

20. Peter Clarke. *Adaptive Cruise Control takes to The Highway.* EE Times, Oct. 20,1998

21. R. D. Henry, *Automatic Ultrasonic Headway control for a Scaled Robotic car* ,Virginia Polytechnic Institute & State University, Blacksburg , Virginia,Dec. 18, 2001

22. Cro, J.W. Parker, R. H. *Automatic Headway control—An Automobile Vehicle Spacing System. SAW, no.* 700086, Jan. 1970

Tran. Veh. Techol. Vol. VT-18, Nov. 1970

24. Design of Electromechanical Robotic Systems, Fall 2009-MIT

25. Ikuo Ihara, *Ultrasonic Sensing: Fundamentals and Its Applications to Non-destructive Evaluation* , Nagaoka University of Technology

26. Massimo Banzi co- founder of Arduino , *Getting Started with Arduino- The Open Source Electronics Prototyping Platform,2nd edition*