Report

Adaptive Cruise Control

Submitted by

Group- 7

Anushri Shah [110040458]

Shyam Savsani [110047108]

Yash Patel [110046695]

Submitted to

Prof. Rashid Rashidzadeh

Table of Content

CHAPTER: 1

INTRODUCTION

1.1 PROJECT SUMMARY

In this contemporary era, population has become major concern which augments their needs too. In recent decade, cars became main dominant vehicle for transportation purpose which relatively shows growth in traffic at road. Later traffic can be becoming cause of accidents and driving in congested area looks boring. Therefore, moving with advanced technology, adaptive cruise system make driving easier and safer [1].

Adaptive cruise control is prototype designed for vehicle which measure the distance of vehicle ahead and acceleration gets adjusts automatically for safety purpose [2]. Data gathered by speedometer is later uploaded on cloud which creates the network thereby everyone connected to same cloud have access to those data. This adaptive cruise control system comprises the Arduino uno(microcontroller), distance sensor (to measure the distance of vehicle ahead), LCD display (to display the on screen) and 5 buttons (to get access of various functions). Programming of this prototype would be done in MATLAB software. This system incorporates five basic functions which are increase speed, decrease speed, cruise set, adaptive cruise set and cancel. Firstly, increase speed button implements the increment of vehicle speed. Secondly, decrease speed button implements the decrement of vehicle speed. Moreover, cruise set button will set speed constant but, we can increase and decrease our set speed of vehicle. Furthermore, adaptive cruise set button will automatically increment and decrement of vehicle speed according to the output of ultrasonic sensor hence, we can also increase and decrease speed if we want. When system enters in adaptive cruise mode display keeps blinking which indicates the adaptive cruise mode. Lastly, cancel button assists to quits both cruise mode. To make this system more advanced data will be uploaded on one IOT (internet of things) based application called ‘thingspeak’, which helps to get access globally and monitor it.

1.2 OBJECTIVE

- Increase the safety level to prevent some major accidents.
- Give the flexibility to the driver to make drive easy.
- Makes easy to adopt and implement the signal and speed limit rules and regulations.
- Maintains the good average level which assists to save fuels.
- Makes global network on cloud, which makes easy to monitor the speed of every vehicle globally.

CHAPTER: 2

SYSTEM REQUIREMENT STUDY

2.1 USER CHARACTERISTICS

2.1 How User Works on System

Firstly, when vehicle starts the system is ready to use. User will need to press the adaptive cruise button to enter cruise mode. Later, user needs to choose either increase or decrease or make constant speed. If users in adaptive mode lcd display keep blinking to differentiate the mode. If user wants reset the speed or had any emergency then pressing cancel button can assist to quit the system. For getting into cruise mode same iterative steps to follow.

2.2 TOOLS & TECHNOLOGY

2.2.1 Hardware Requirements:

- Ultrasonic Sensor.
- LCD Display.
- Arduino Uno.
- Breadboard.
- 5 buttons.
- Connecting wires.
- Potentiometer.

2.2.2 Software Requirements:

- MATLAB (2021a).
- ThingSpeak.

2.3 COMPONENTS USED IN DESIGNING

We will be providing the detailed description of every component used in designing this Adaptive cruise control system.

- **ULTRASONIC SENSOR:**

Fig 2.1: An Ultrasonic Sensor [3]

An **ultrasonic sensor** is a "contactless" device which can be read surrounded objects without any physical contacts [3]. This device is works on electromagnetic field when there is any fluctuation in field replies to detection of object. There are many types of proximity cards like capacitive proximity sensor or photoelectric sensor used for plastic target; an inductive proximity sensor used for detecting metal target; in this project inductive proximity sensor has been used. Compare to wired devices used for detection, proximity sensor has high reliability and long-life functionality as there is no presence of any mechanical parts and physical contacts. Distance is measured by this formula:

Distance = (Time*Speed of Sound)/2

- **ARDINO UNO:**

Arduino is an open-source hardware and software project, created with a simple aim in mind, to be as simple as possible [4]. It is used by artists, hackers, hobbyists, and professionals to easily design, prototype and experiment with electronics. An Arduino contains a microchip, which is a very small computer that you can program. A switch and a sensor could be a digital and an analog input respectively into the Arduino. Any object we want to turn on and off and control could be an output. These systems provide sets of digital and analog input/output (I/O) pins that can interface to various expansion boards and other circuits. The boards feature serial communication interfaces, including Universal Serial Bus (USB) on some models, for loading programs from personal computers. For programming the microcontrollers, the Arduino project provides an integrated development environment (IDE) which also supports the languages C and C++. The Arduino language is very similar to C. It's almost the same language but Arduino provides us with several libraries to make things a bit easier.

Fig 2.2: Arduino UNO [4]

- **LCD DISPLAY:**

Liquid Crystal Display: A 16*2 LCD Display is used to display the speed of the host car which shows the varying speed in different modes. In our project we used Liquid Crystal Display (LCD Display) for better output. This display is used for displaying numerical data captured from various sensors. This is programmed using the library (ExampleLCD/LCDAddon). It has the following features: Easy interface to Arduino, ultra-thin design, cheap and economical.

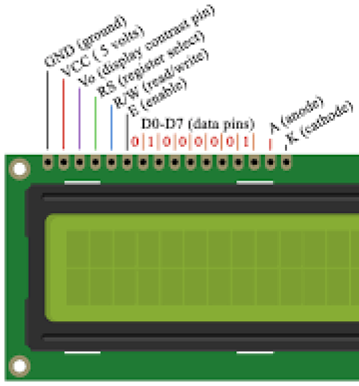


Fig 2.3: LCD Display

- **POTENTIOMETER:**

Basically, **potentiometer** is used to set contrast of LCD Display. It is easy to use and connect to LCD Display for better output in LCD Display.

Fig 2.4: Potentiometer

- **BUTTONS:**

In our project 5 buttons provides user interface component. All five buttons have different functions. All buttons are pre-programmed to their functions. Functions are: increase speed, decrease speed, cruise set (Speed Set), adaptive cruise set, and cancel function.

Fig 2.5: Button

2.4 CLOUD CONNECTIVITY

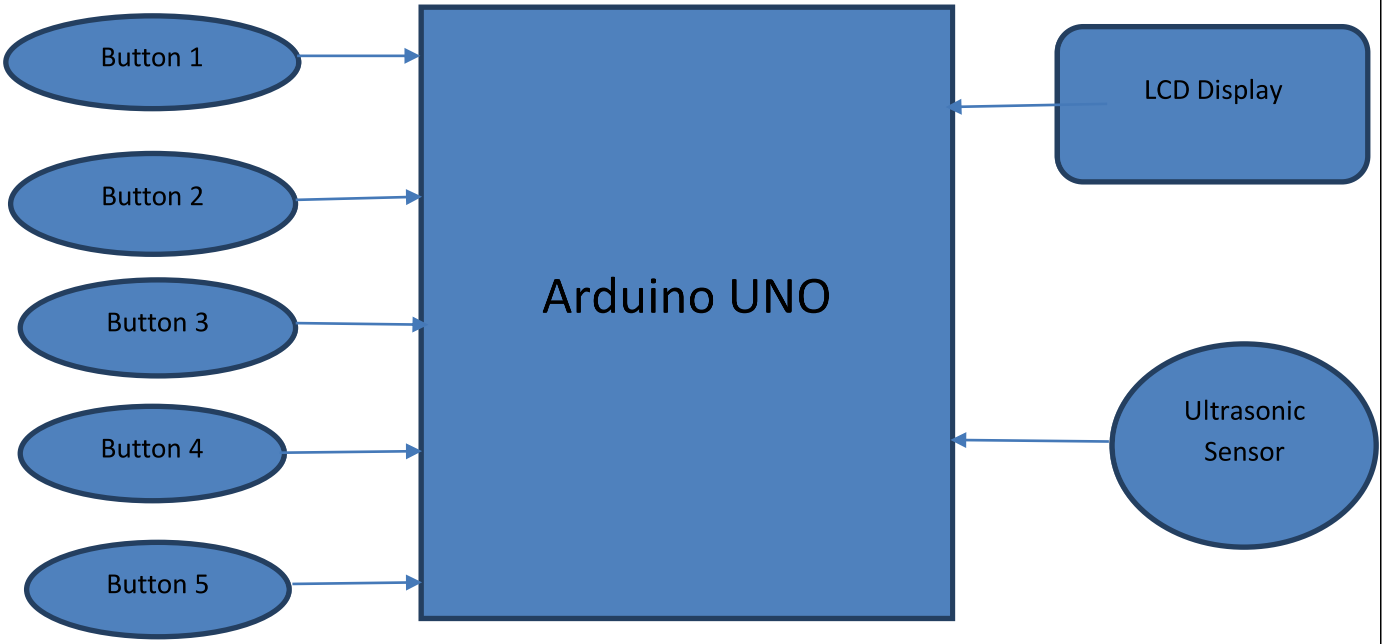
ThingSpeak is used to store speed in cloud via writeAPI key. Our write API key and Channel ID is *21WP3JEZRN1GFAD2* and *1457868*. Syntax used for sending data in cloud is:

thingSpeakWrite(1457868,cur_speed,'WriteKey','21WP3JEZRN1GFAD2');

CHAPTER: 3

SYSTEM DESIGN

3.1 BLOCK DIAGRAM



3.2 WORKING:

The basic working of the system starts with the initialization of the system with the display of speed zero on the LCD screen. Once the increase button is pressed there should be an increase of speed from the initialization stage displayed on the display screen. Further, the pressing decrease button should decrease speed. Since the set speed button is off the speed will not remain constant and will decrease if the increase button is released. Pressing the set speed button will fix the speed at a constant leading the vehicle to enter the normal cruise control mode. To change the speed in cruise control mode the increase and decrease button are operative. If the cancel button is active the system exits the cruise control mode and the speed starts decreasing slowly. When the Adaptive button is pressed, the speed is set to a constant and is held at the same constant till there is no object or vehicle in front. Once the object is detected the speed starts decreasing and as the distance between the object and the host car increases the speed increases. The display blinks when the system changes the mode from cruise control to adaptive cruise control. To exit the adaptive cruise control mode, the cancel button is pressed which stops the blinking and the systems speed starts. All the data is uploaded to cloud via API to ThingSpeak.

CONNECT ULTRASONIC SENSOR TO ARDUINO:

The ultrasonic sensor works by sending sound waves from the transmitter, which then bounce off an object and then return to the receiver. We can determine how far away the object is by the time it takes for the sound waves to get back to the sensor.

PIN FUNCTION CONNECTS TO:

1. VSS 5V.
2. TRIG Arduino Digital Pin 8.
3. ECHO Arduino Digital Pin 9.
4. GND.

CONNECT INPUT BUTTONS TO ARDUINO:

Hardware requirements for the process are mentioned below:

Push buttons – 5

Resistor (220 Ohm) – 5

Jumper Wires

Connect one of the Arduino ground pins to the breadboards ground pin and then connect the 5V of the Arduino to the 5V of the breadboard to complete the circuit. Connect a jumper wire from the 5-volt pin to one side of the pushbutton and follow the same for all the five buttons. Further, connect a 220-ohm resistor from the ground pin to the other side of the button and follow the same for all the other buttons.

BUTTON FUNCTION CONNECT TO:

1. Increase Button Arduino Digital Pin 13.

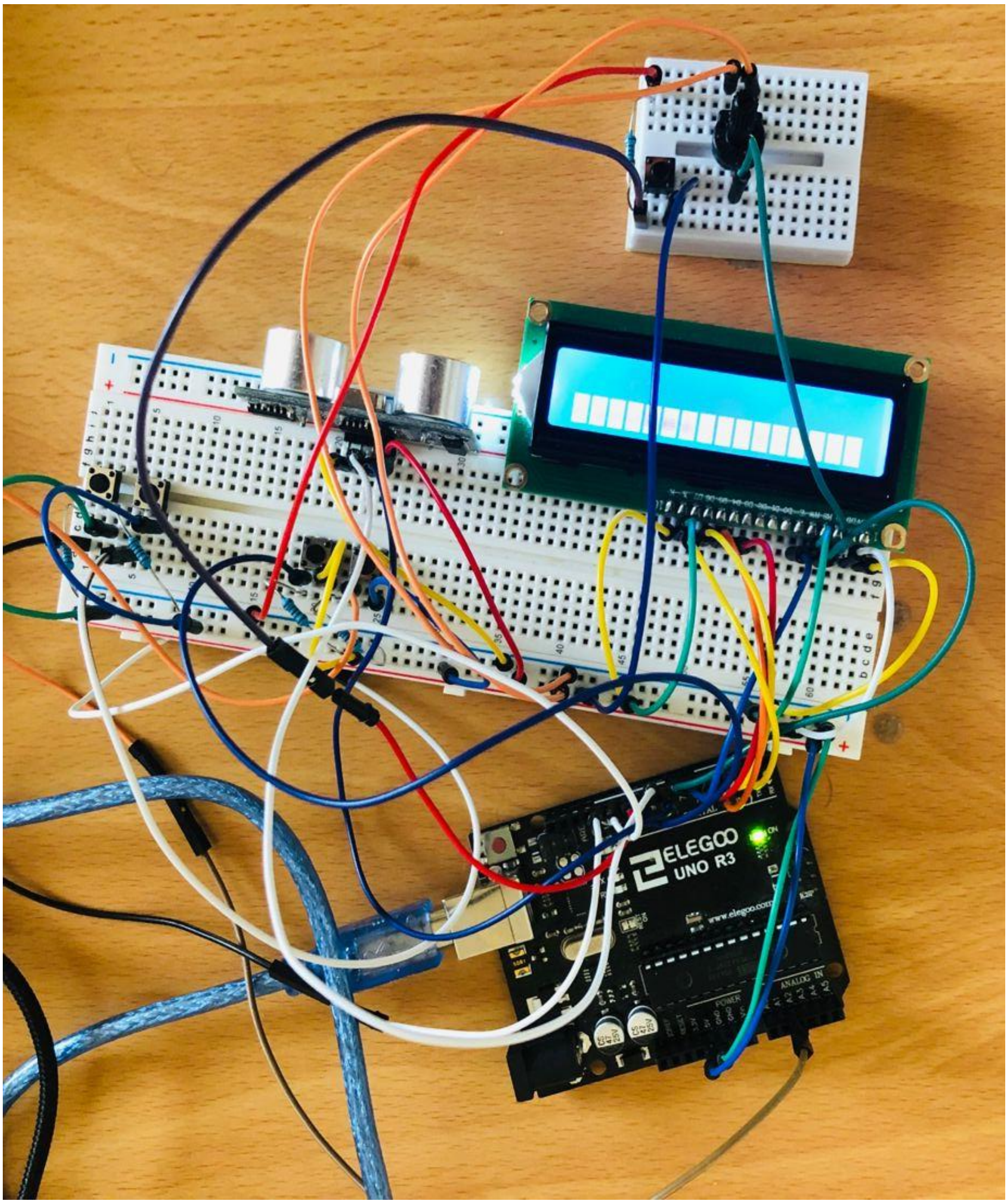
2. Decrease Button Arduino Digital Pin 12.

3. Cruise Set Button Arduino Digital Pin 11.

4. Adaptive Cruise Set Button Arduino Digital Pin 10.

5. Cancel Button Arduino Analog Pin 2.

3.3 Hardware Model



3.2 MATLAB Code:

```
clear;

clc;


a = arduino('COM5','Uno','Libraries',{'Ultrasonic' 'ExampleLCD/LCDAddon'});

speed = 0;

cur_speed = 0;

max_speed = 200;

set_speed = 0;

cursetspeed = 0;

increase = 'D13';

decrease = 'D12';

cruise = 'D11';

adp = 'D10';

cancel = 'A2';

cru = 0;

adc = 0;

tempStepCru = 0;

tempStepThreshCru = 7;

tempStep = 0;

tempStepThresh = 4;

set_adp_cruise = 0;

stored_speed = 0;


ultrasonicObj = ultrasonic(a,'D8','D9');


lcd =
addon(a,'ExampleLCD/LCDAddon','RegisterSelectPin','D7','EnablePin','D6','DataPins',{'D5','D4','D3',
'D2'});

initializeLCD(lcd);
```

```
while 1

    printLCD(lcd,'Speed');
    printLCD(lcd,num2str(cur_speed));

    inc_state = readDigitalPin(a,increase);
    dec_state = readDigitalPin(a,decrease);
    cru_state = readDigitalPin(a,cruise);
    adp_state = readDigitalPin(a,adp);
    can_state = readVoltage(a,cancel);
    d = readDistance(ultrasonicObj);
    distance = (d*343)/2;

    if inc_state == 1 && cur_speed <= max_speed
        cur_speed = cur_speed + 1;
    end

    if dec_state == 1 && cur_speed > 0
        cur_speed = cur_speed - 1;
        printLCD(lcd,'Speed');
        printLCD(lcd,num2str(cur_speed));
    end

    if cru_state == 1
        set_speed = 1;
    end

    if set_speed == 0
        tempStepCru = tempStepCru + 1;
        if mod(tempStepCru , tempStepThreshCru) == 0 && cur_speed > 0
            cur_speed = cur_speed - 1;
```

```
            tempStepCru = 0;
        end
    end
    if adp_state == 1
        set_adp_cruise = 1;
    end
    if set_adp_cruise == 1
        tempStep = tempStep + 1;
        if distance < 20 && cur_speed > 0
            if stored_speed < cur_speed
                stored_speed = cur_speed;
            end
            if mod(tempStep , tempStepThresh ) == 0
                cur_speed = cur_speed - 1;
                tempStep = 0;
            end
        else
            if stored_speed > cur_speed
                if mod(tempStep , tempStepThresh ) == 0
                    cur_speed = cur_speed + 1;
                    tempStep = 0;
                end
            end
        end
    end
    if can_state > 4
        set_adp_cruise = 0;
        set_speed = 0;
    end
thingSpeakWrite(1457868,cur_speed,'WriteKey','21WP3JEZRN1GFAD2');
end
```

LIST OF FIGURES

REFERENCES

[1] Pratyush Bhatia, "Vehicle Technologies to Improve Performance and Safety" ,WaybackMachine.[Online] [Accessed: 27 July, 2021]

[2] What is adaptive cruise control, and how does it work? - ExtremeTech". www.extremetech.com. [Online] [Accessed: 27 July, 2021].

[3] O'Brien, Daniel J, "Proximity sensor on Android smartphones", TheCodeArtist. [Online] [Accessed: 27 July, 2021].

[4] Hernando Barragán (2016-01-01), "The Untold History of Arduino". arduinohistory.github.io. [Online] [Accessed: 27 July, 2021].