SIMPLE PARKING SENSOR

Almira GÜRKAN

*Abstract*— An embedded system is a microprocessor-based computer hardware system with software that is designed to perform a dedicated function, either as an independent system or as a part of a large system. Embedded systems are used in many devices such as mobile phones, televisions, calculators, cars, video cameras, burglar alarms, refrigerators that are included in our daily lives through technology and they are a part of our lives.

Keywords—embedded systems, hardware, software, sensor

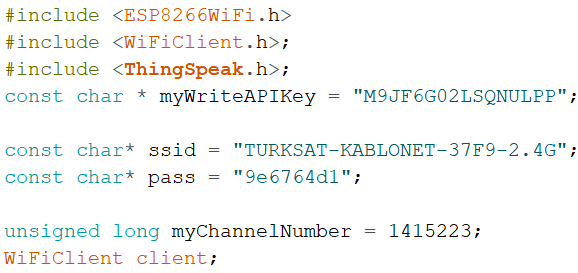
# Introduction

This project is designed to provide information to the user by showing the distance between the vehicle and another object on the android operating system phone by making a parking sensor. The user will also be warned by the sound and light in the system. One of the biggest problems in cities is parking vehicles in crowded traffic environment. In addition, incorrect parking and misuse of parking spaces make this situation even worse. This system, which was developed to solve these problems, provides great support to the driver.

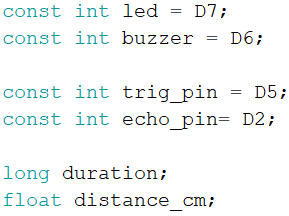
# Project Details

In this system, your distance between two vehicles or your proximity to the object to be approached is measured with the help of sensors as the main function. With the sound and led, you can understand the distance. This will be achieved by increasing the density of the LED's blinking and buzzer's sounds as the distance between the vehicle and the object decreases. In addition, the distance between the vehicle and the object will display with the application running on the smart phone.

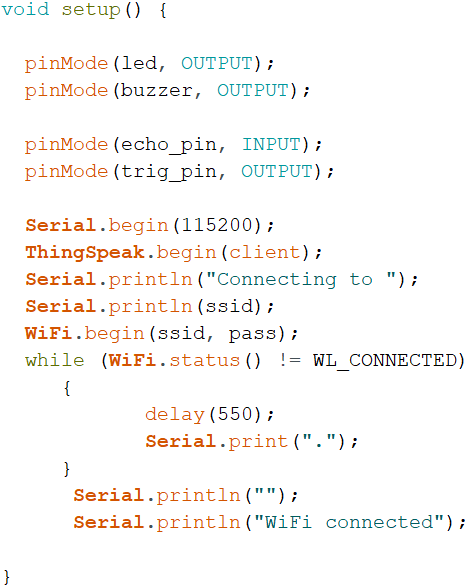
# Implementation



I started by coded ThingSpeak API key. I defined the wifi name and password to connect to the same wifi as the laptop.

 I introduced which pin I use for what purpose to arduino. I set ledPin to D7 and buzzer to D6, trig\_pin to D5, echo\_pin to D2, I also defined the variables duration and distance\_cm to calculate the distance.

In the setup part I set led, buzzer and trig\_ping as an output, echo\_pin as an input. I start the serial communication for showing the results on the serial monitor.



In loop, I sent power to trig\_pin with 'digitalWrite(trig\_pin, HIGH)'. I have powered the trig\_pin at 10 microsecond intervals to send power at regular intervals. So, I wrote 'delayMicroseconds(10)', then I stopped power with 'digitalWrite(trig\_pin, LOW)'. I added ThingSpeak.begin(client) and WiFi.begin(ssid, pass) to initialize them.



Then, I used the pulseIn command. This command reads a pulse on a echo\_pin. Here, value is HIGH , pulseIn() waits for the pin to go from LOW to HIGH , starts timing, then waits for the pin to go LOW and stops timing. I used by duration variable I calculated distance.

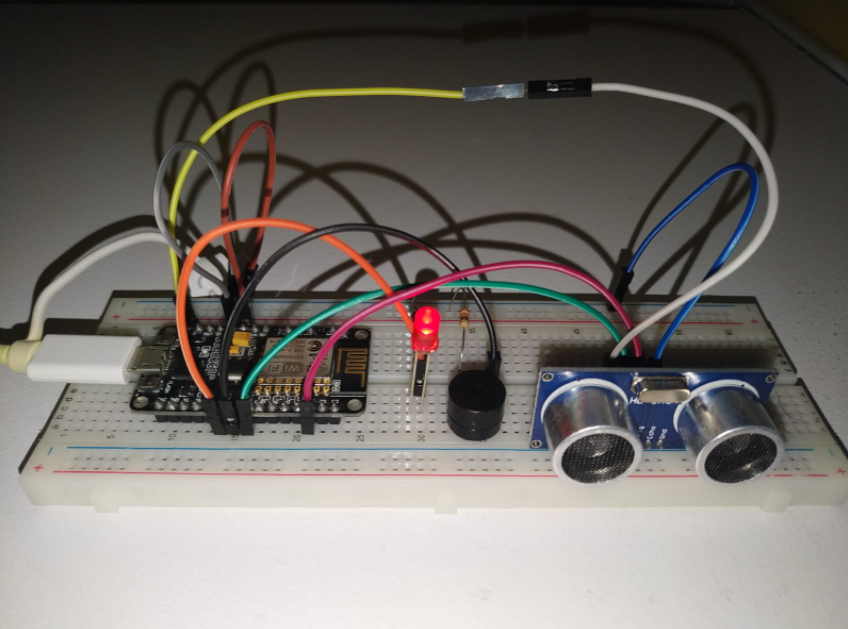
Duration return the length of the pulse in ms. Speed of sound 340m/sn and distance is product of time and speed. So, distance\_cm is the product of duration and 0.034cm/ms.

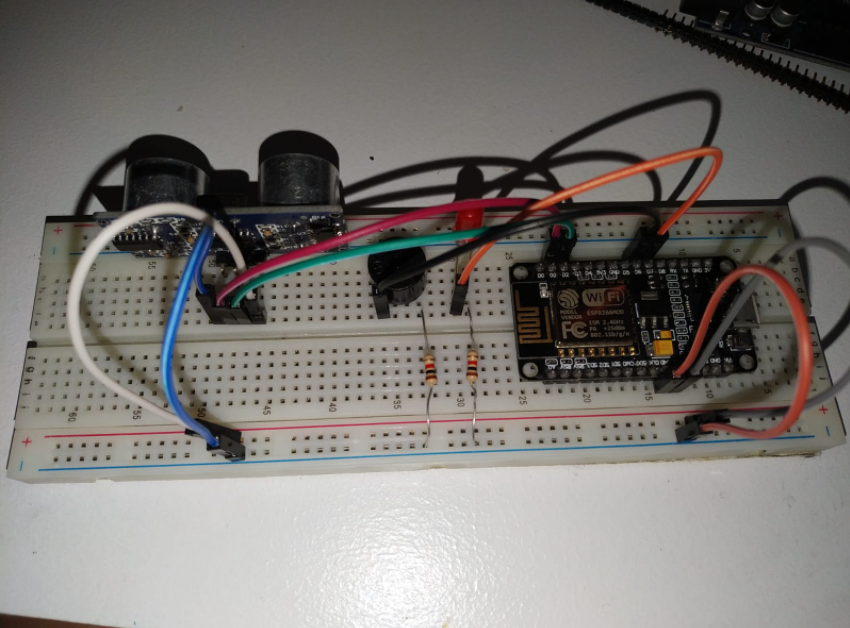
Also, I coded ThingSpeak. writeField (myChannelNumber, 1,distance\_cm, myWriteAPIKey) to send the values received from the circuit to thingspeak.

Finally, I used the if else commands to set the flashing of the LED and giving sounds according to the distance. If distance less than 10, the led blinks and sounds frequently. Else if distance less than 30, the led blinks and sounds normaly and else if distance more than 30, the led blinks and sounds rarely.

# CIRCUIT

In Circuit, I used NodeMCU, ultrasonic sensor (HC-SR 04), breadboard, led, buzzer, resistors and jumper cable. NodeMCU is an open-source Lua based firmware and development board specially targeted for IoT based Applications. I aimed to transmit the distance values to my phone using the wifi module. Firstly, I connected the trigger pin on the ultrasonic sensor to the NodeMCU pin D5. Similarly, connect echo pin to D2. Buzzer is connected to D6. LED connected to D7. I used resistor to connect led and buzzer. Resistor is simply the circuit element that reduces the electric current. As an example I used my circle, if I had connected the led directly to the power supply, too much current would flow through the led and the led would probably break down. To prevent this, I connected a resistor to led. I also connected a resistor to buzzer. Firstly I used 1KΩΩ resistor for both. But buzzer did not work. Then, I used 330ΩΩ resistor for buzzer. I connected the nodeMCU to the computer with a USB cable to give current to the circuit. Finally, I tested again the circuit, in the end the circuit is working correctly.





# IOT

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu A mobile application is built for Android as IoT. The application simply shows a notification which is “The distance is less than 10cm. STOP!” if the distance is less than 10cm between the car and the parked place. Otherwise it says “The distance is more than 30cm.” The example screenshot of application is given below.

Thingspeak and Mit App Inventor were used, while making this app. It works like this: Data which is transferred to ThingSpeak, is taken from Mit App Inventor.

**Examination the code of app:**

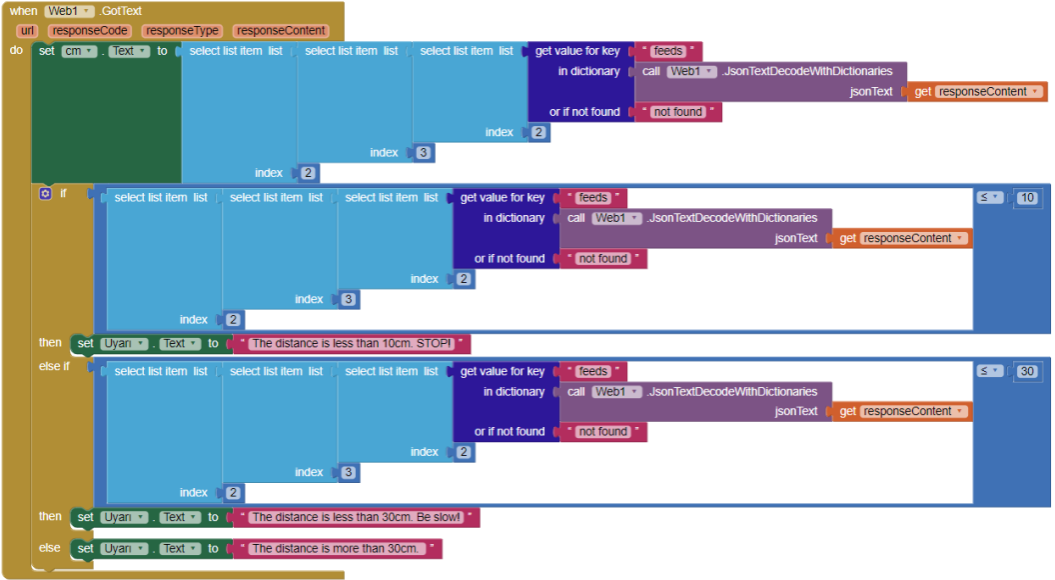
The code has 3 parts.

Firstly, we can think of Clock variable as a for loop that tells the app how many seconds to update the data which comes from thingspeak. After that, it provides a connection between application and website. The code example is below.

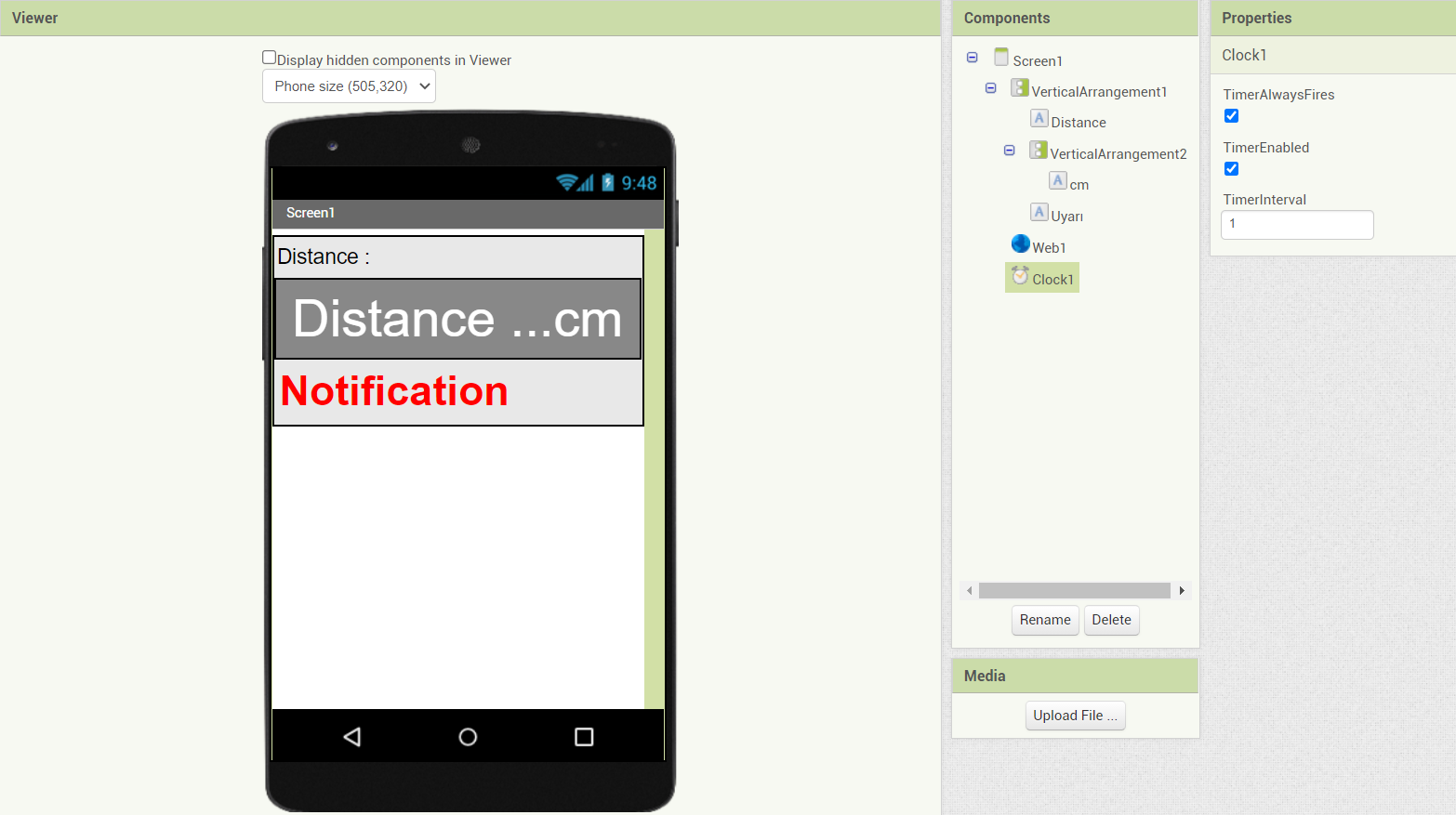
metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Second part is about recieved value from thingSpeak. There is if-else condition for values and notifications as we see below.



Finally, there is the part where the application is designed. How it is designed is shown below.



##### References

* https://www.arduino.cc/en/software
* https://www.omnisci.com/technical-glossary/embedded-systems
* EMBEDDED SYSTEM BASICS AND APPLICATION, Prof. Dr. Ali Ziya Alkar,2009
* <http://www.ee.hacettepe.edu.tr/~alkar/ELE417/week1_hacettepe.pdf>
* <https://www.electronicshub.org/embedded-systems-projects-ideas/>
* https://iotdesignpro.com/projects/smart-inventory-management-system-using-iot