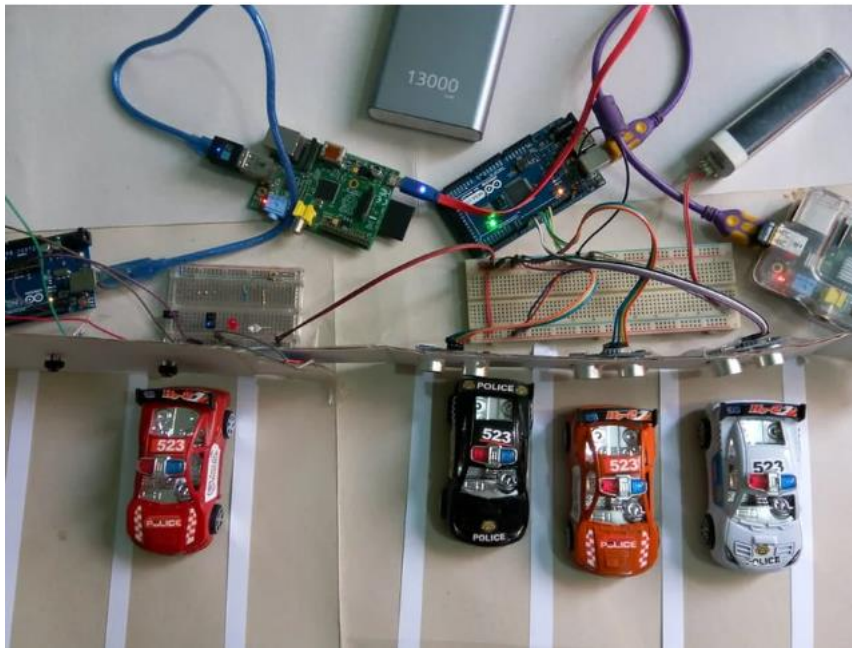


Smart Parking System



Steps involved in this project:

1. Making three devices in Artik Cloud Platform
2. Making one application in Artik Cloud
3. Making one rule in Artik Cloud
4. Preparing & Programming Arduino
5. Connecting the Sensors to Arduino
6. Preparing Raspberry Pi
7. Preparing Intel Edison
8. Developing Web Application
9. Developing Android Application
10. Completing the Project.

So, Let's start one by one.

Step 1: Things Used in This Project



Hardware

1. Raspberry Pi 3 Model B (2pcs)
2. Arduino UNO (2pcs)
3. Intel Edison with Arduino Breakout
4. Grove starter kit plus for Intel Edison
5. Ultrasonic Ranger HC-SR04 (4pcs)
6. IR receiver (generic) (4pcs)
7. IR transmitter (generic) (4pcs)
8. Resistor 221 ohm (4pcs)
9. Resistor 10k ohm (4pcs)

Software & Online Services

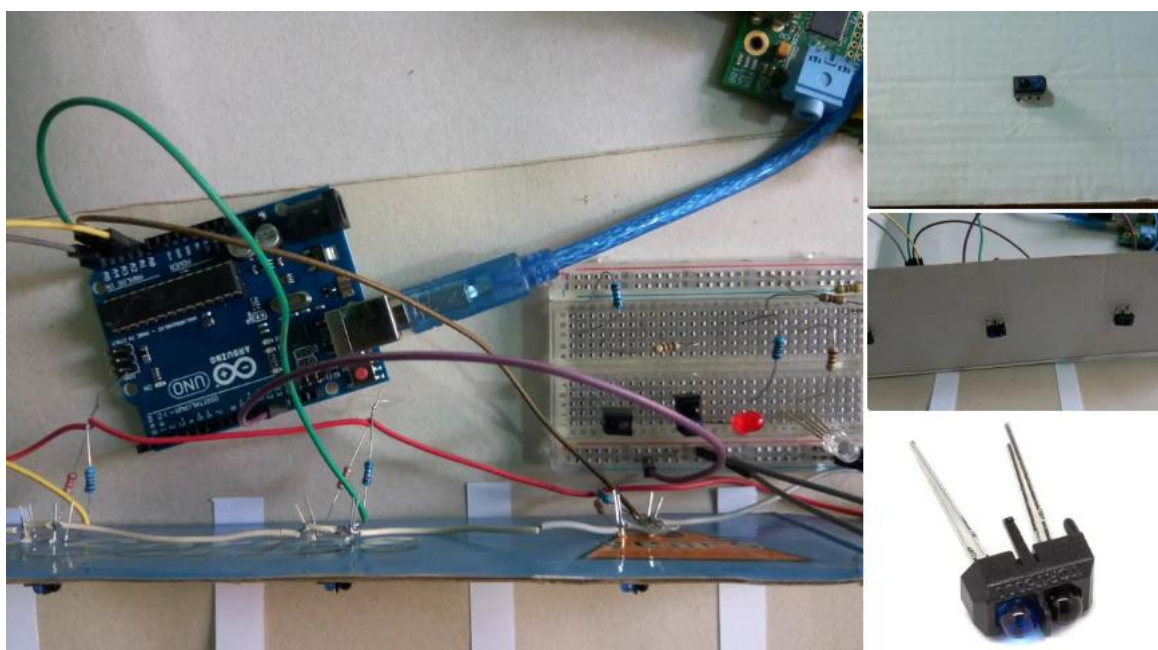
1. Samsung IoT ARTIK Cloud for IoT
2. Google Developers Google Maps
3. Arduino IDE
4. Intel IoT XDK

Programming Language

1. HTML
2. Java Script
3. Node.js
4. Python

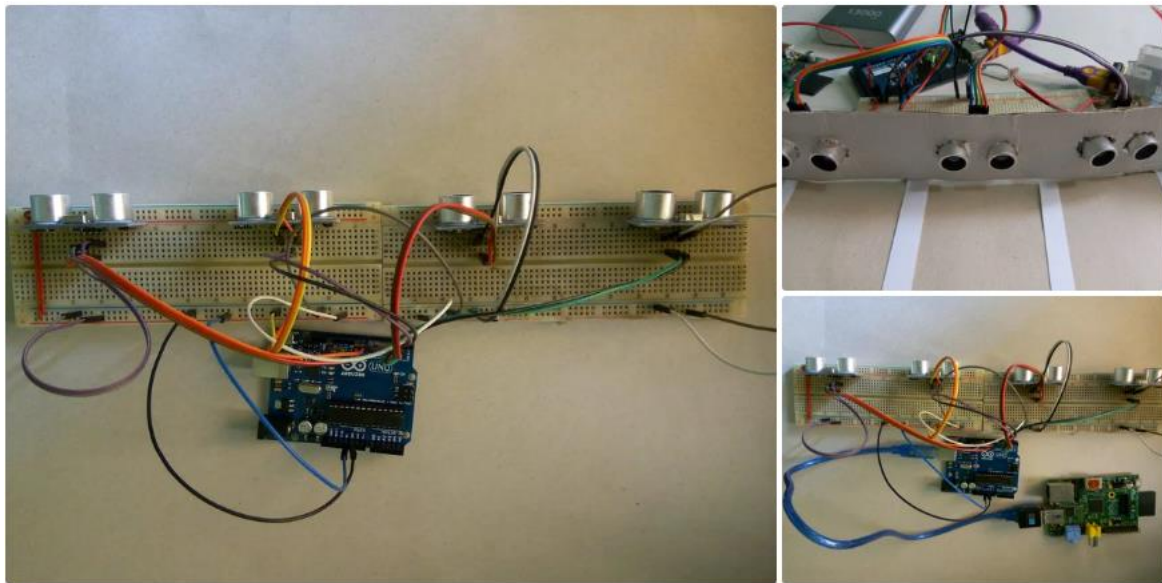
The main work of the Arduino in my project is to collect data from sensors connected to parking stations and to send the data to Raspberry pi using serial port. Two Arduino board is required as I implemented the demo project for two parking stations. For the two parking stations I used two different types of sensors. For one I used IR sensors can be used in the place where direct sun light is not available and for another I used ultrasonic sensors and can be used in any location.

For IR sensor interfacing Arduino analog pins were used and for ultrasonic sensors Arduino digital pins were used. Arduino sketch for both types of sensors



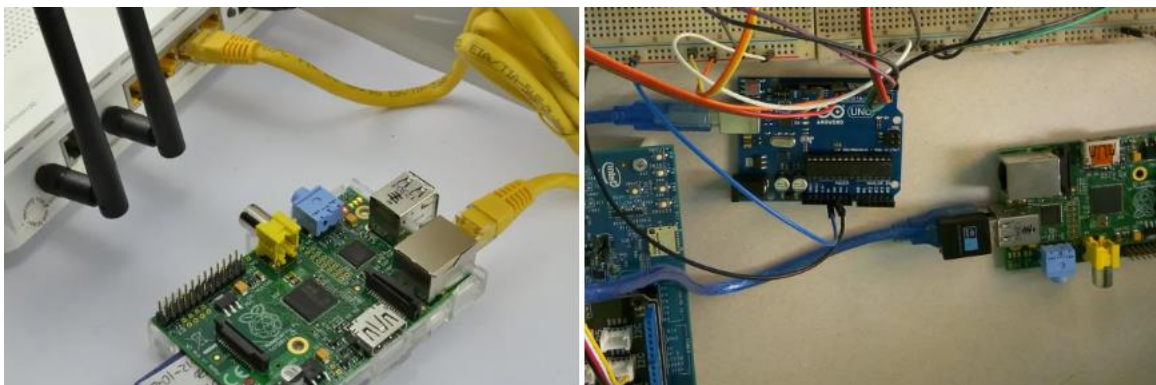
are attached. Upload appropriate sketch to your Arduino board. I developed the sketch for only four sensors. If you require more modify the sketch accordingly.

For connecting all the sensors to Arduino please see the schematic section. As you already know, ultrasonic sensor communicate Arduino or other microcontrollers using digital pin. For the reason I used digital pins to connect ultrasonic sensors with Arduino.



I am assuming that, you have some previous experience in working with Raspberry pi. Lots of beginners tutorials are available in the Internet.

Let's start our work with raspberry pi. Raspberry pi is working as a WebSocket client and communicates with Artik cloud using WebSocket protocol. I used Node.js for programming. Unlike Python Node is not preinstalled on Raspberry Pi.



A). Install Node and npm on Raspberry pi using following commands in terminal.

```
sudo apt-get update  
sudo apt-get install nodejs npm
```

B). Install WebSocket client ws to raspberry pi

```
npm install --save ws
```

C). Install serialport to raspberry pi

```
npm install serialport
```

If you successfully completed all three steps then your pi is ready to communicate with Arduino using serial port and Artik cloud using websocket. As I said earlier two raspberry pis are required for two parking station. So, download the source code for the pis from the Github and upload rainbow-parking.js for the pi connected to Rainbow parking station and indigo-parking.js on the pi connected to Indigo parking station. You can use ftp client like [FileZilla](#) to upload the code on raspberry pi. Don't forget to change DEVICE ID & DEVICE TOKEN in the source code.

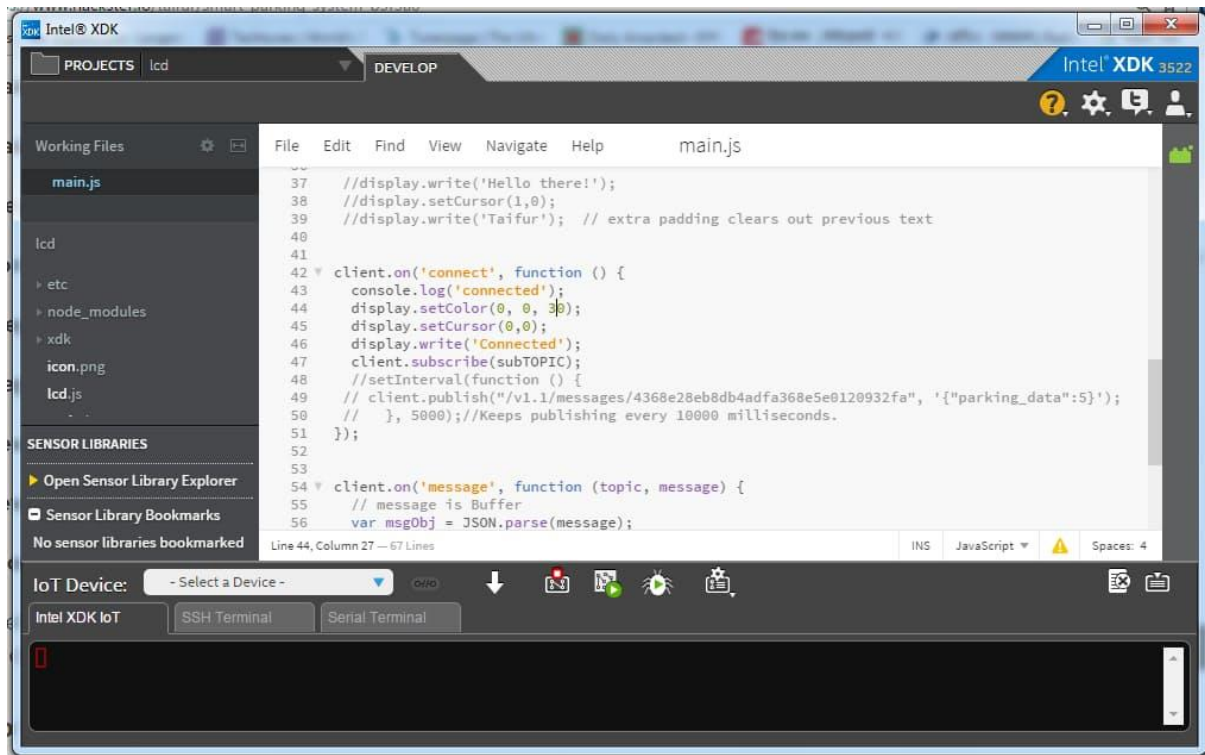
D) After uploading the code on raspberry pi run the code by typing the following command into terminal window.

```
sudo node rainbow-parking.js
```



A). Download Intel XDK from the [link](#) and install it to your computer.

B). Find edison-lcd folder from attached zip file and open with Intel XDK.

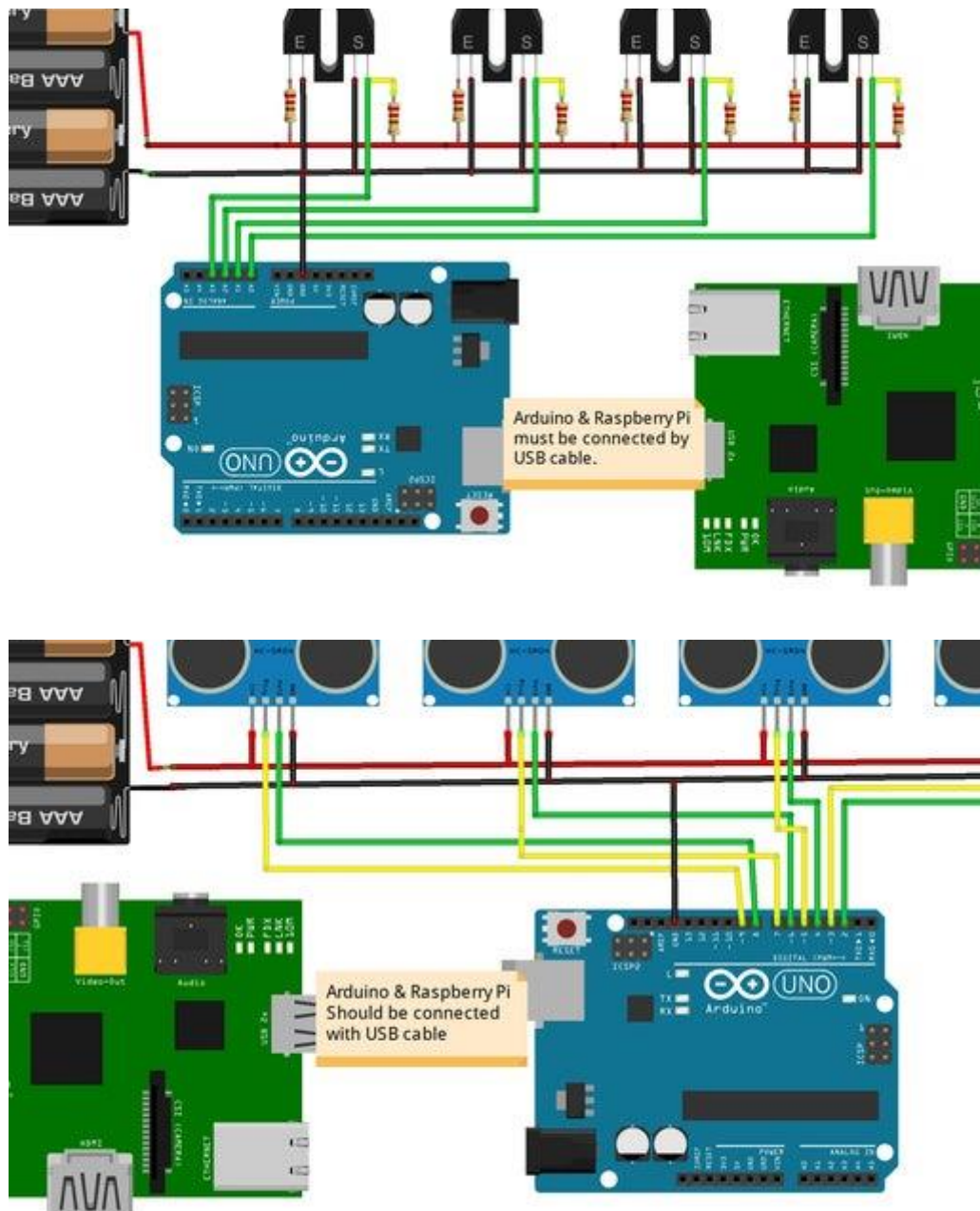


C) Connect i2c LCD display with Edison board. You may use base shield for easy connection.

D) Connect Edison board with computer using serial cable and upload the code and then run it. Uploading and running code in Edison board is very easy using Intel XDK.

If it is the first time you are using Edison board, [this tutorial](#) from Intel may help you.

Block Diagram, Schematic & Source Code



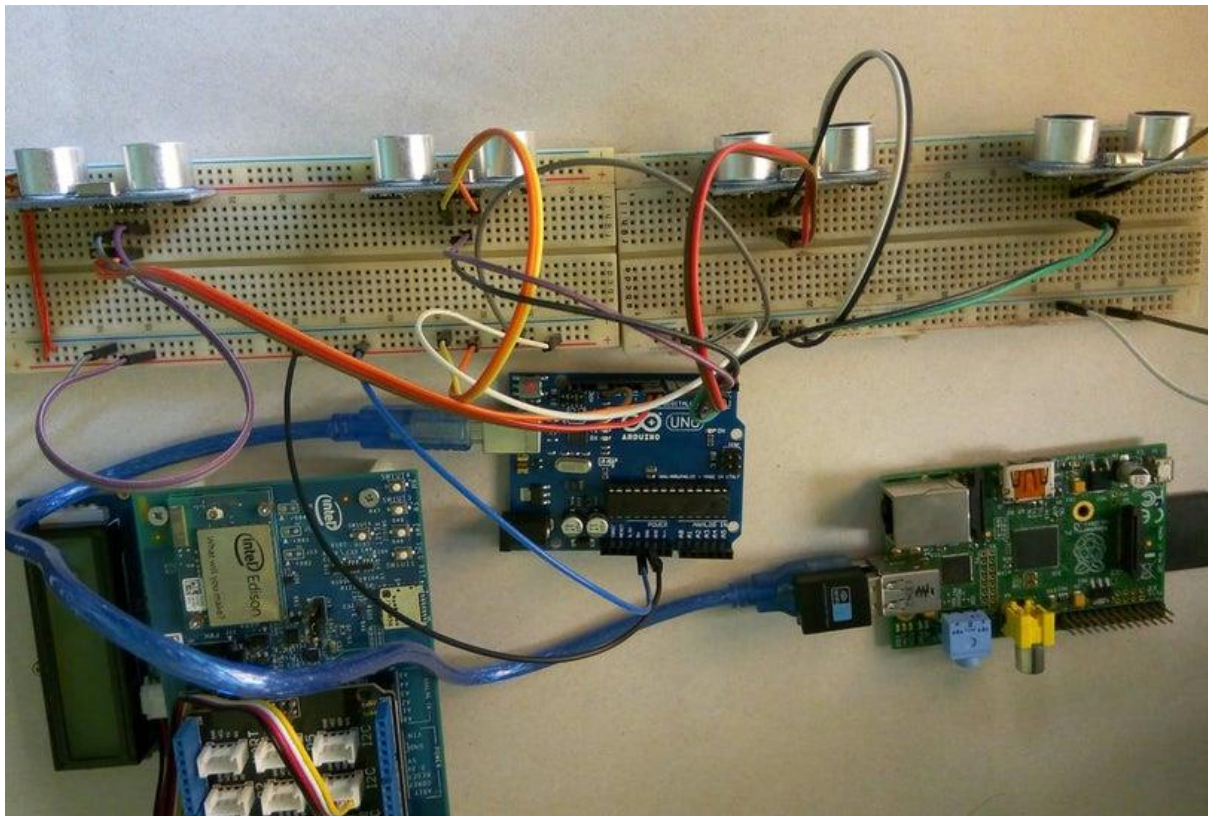
Block diagram shows how the total system works. Parking data is read by Sonar sensors, IR sensors & Arduino. IR sensor is effective for indoor parking lot & sonar sensor can be used for indoor and outdoor application. After reading and processing the data Arduino sends number of engaged and free slot of the parking lot to Raspberry Pi using serial port. Raspberry Pi sends the data to Artik Cloud using WebSocket protocol. Other protocol such as MQTT or HTTP may be used. Now, the parking information can be read from web application or mobile application. I have shown both way in my project. For my demo project I implemented two parking lot one by sonar sensor and another by IR sensor. I would be nice if there are some big display board in the important location of the city which shows free parking lot so that a driver can find it easily without

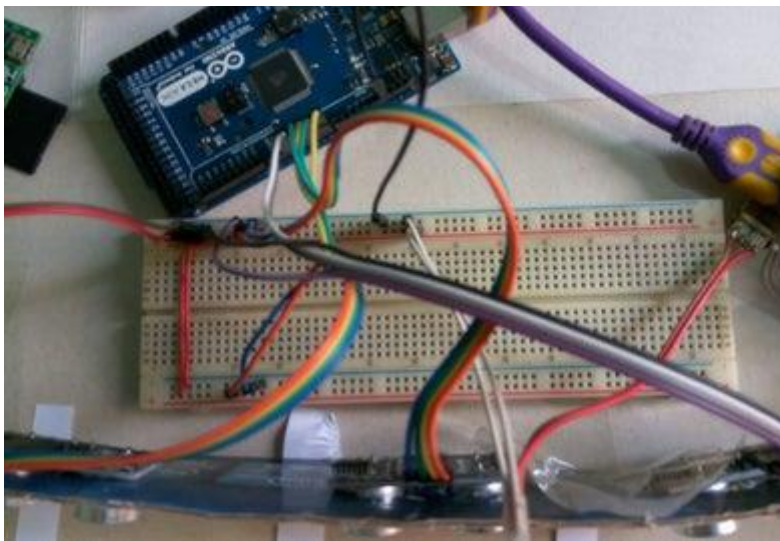
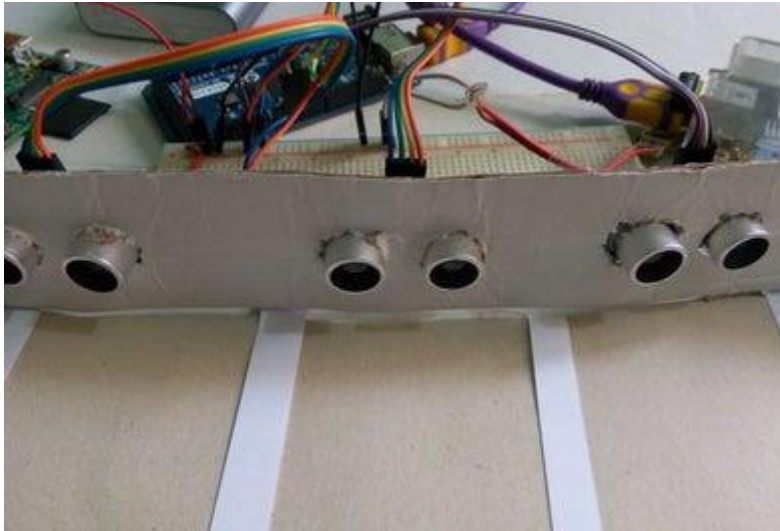
accessing personal device. From the thinking I shows how Intel Edison can be used for the job.

The schematic diagrams are very easy to understand. I am reading IR sensors value using analog pins of Arduino and sonar sensors value using digital pins of Arduino.

I think you will feel comfortable to modify the source code as your requirement because I tried to add enough comments in the code so that, it becomes easy & understandable.

Setup All the Hardware





I attached some photo of two parking lot and reading data from one parking lot using web application.