Smart Cities Using IOT

By: Shubham Kumar, skshubhamkumar907@gmail.com

Soumya Sharma, thesoumya06@gmail.com

* Smart Car Parking
* Smart Street Light System

**SMART CAR PARKING**

In this fast-paced world, a very common happening that is observed daily is congestion caused due to traffic. With the increase in the number of vehicles, car parking remains one of the major issues in developed cities.

The search for an empty parking slot exists as a very tiring activity for people, and also contributes to the wastage of engine oil every day. These problems cause a rise for an efficient and smart car parking system.

Our IOT-based smart car parking system creates a garage like platform for a specified number of cars, and allows cars to enter only if a vacant space exists.

On a bigger level, it works on the principle that such modules of intelligent parking systems distributed throughout parking lots will create an easier outlook for users to find empty parking spaces.

In this project, we used six IR sensors which act as the parking slot detectors using sensing methodologies. The sensors will give logical output into the arduino UNO microcontroller, which will be used to decide whether further users can enter the slot.

Servo motors are used to virtually denote entry as well as exit gates. The gates are also logically programmed to allow entry of vehicles only if there are empty slots within the parking lot.

Jumper wires are used to connect various components on the breadboard.

The LED bulbs are used as indicators to indicate whether the slots in the parking lot are vacant or not.

This makes it easier for users to rapidly search for empty parking slots efficiently.

INITIAL HARDWARE SETUP:

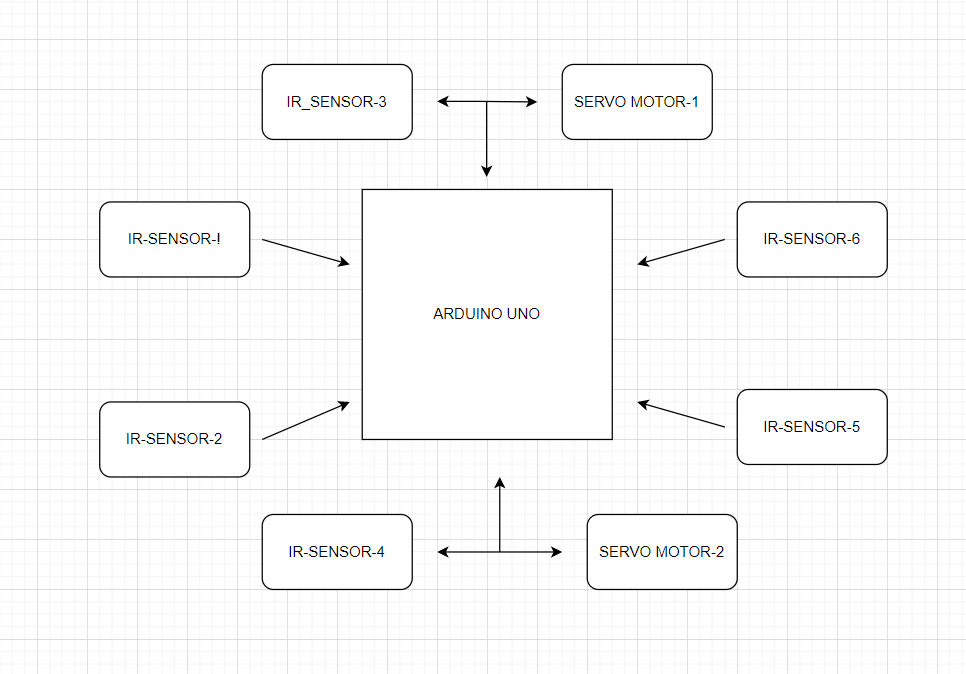
After testing the individual components with separate codes, we first connected six I sensors to breadboard and arduino using jumper wires.

The 4 LEDs were connected to the breadboard which would act as indicators.

Two Servo motors were connected and programmed, acting as gates for the parking lot.

They were given special conditions to work only when vacancy is available.

Programmes were effectively compiled and executed.



Flow Diagram

CODE

#include <Servoh>

Servo servol;

Servo servo2;

const int analogInPin1 = A0;

const int analogInPin2 = Al;

const int analogInPin3 = A2;

const int analogInPin4 = A3;

const int servo\_sensor] = A4;

const int servo\_ sensor2 = A5;

int led = 2;

int led = 3;

int led2 = 4;

int led3 = 5;

const int servo\_1=8; const int servo\_2 =9;

int value 1=0; int value 2 =0;

void setup

{

pinMode (analogInPin1 ,INPUT); pinMode (analogInPin2,INPUT); pinMode (analogInPin3,INPUT); pinMode (analogInPin4,INPUT);

pinMode(servo\_sensor1,INPUT);

pinMode(servo \_sensor2,INPUT);

servol.attach(servo\_1); servo2.attach(servo 2);

pinMode (led,OUTPUT); pinMode (led1,OUTPUT); pinMode (led2,OUTPUT); pinMode (led3,OUTPUT);

void loop(

{

if((analogRead(A3)>500)&&(analogRead(A2)>500)&&(analogRead(A1)>

500)&&(analogRead(A0)>500))

{

digitalWrite(led, LOW);

digital Write(led 1, LOW);

digitalWrite(led2, LOW);

digital Write(led3, LOW);

else

value\_1 = analogRead(servo sensorl);

value\_2 = analogRead(servo \_sensor2);

// servo 1 code //

{

if (analogRead(A4)<500)

{

servol.write (90);

delay(3000);

}

if (analogRead(A4)>500)

{

servol.write(0);

}

delay(60);

// servo 2 code //

if (analog Read(A5) <500)

{

servo2. write(90);

delay(3000);

}

if (analogRead(A5)>500)

{

servo2. write(0);

delay(60);

}

I Parking Slot 1 IR & Led

if (analogRead(A0)<500) // if the IR sensor value is < 500 then the motor will start rotating

{

digitalWrite(led, HIGH);

}

else if (analogRead(A0)>500) // if the IR sensor value is < 500 then the motor will start rotating

{

digital Write(led, LOW);

}

/I Parking Slot 2 IR & Led

if (analogRead(A1)<500) // if the IR sensor value is < 500 then the motor will start rotating

digital Write(led1, HIGH);

}

else if (analogRead(A1>500) // if the IR sensor value is < 500 then the motor will start rotating

{

digital Write(led1, LOW);

}

// Parking Slot 3 IR & Led

if (analogRead(A2)<500) // if the IR sensor value is < 500 then the motor will start rotating

{

digitalWrite(led2, HIGH);

}

else if (analogRead(A2>500) // if the IR sensor value is < 500 then the motor will start rotating

{

digital Write(led2, LOW);

}

// Parking Slot 4 IR & Led

if (analogRead(A3)<500) // if the IR sensor value is < 500 then the motor will start rotating

{

digital Write(led3, HIGH);

}

else if (analogRead(A3)>500) // if the IR sensor value is < 500 then the motor will start rotating

{

digital Write(led3, LOW);

}

}

RESULTS AND DISCUSSION

The project we made as stated earlier in our report is a portion of such a large aspect smart car parking. There are two main objectives of our project-

1) To implement an intelligent car parking system for minimizing time wasted in searching for parking slots

2) To prevent wastage of oil and energy in the process of searching empty parking slots.

Both of the above mentioned objectives have been met.

CONCLUSION AND FUTURE SCOPE

An effective execution of this task would bring about less traffic and disarray in jam-packed parking spots like shopping centers and business structures where numerous individuals share a stopping zone.

It gives drivers Also, as it would diminish the holding up time, long lines, pressure, stress and increment the productivity of the stopping framework.

As the Smart Car Parking System Requires insignificant labor, there are least possibilities for human mistakes, expanded security notwithstanding a quick and well disposed vehicle leaving experience for drivers.

SMART STREET LIGHT SYSTEM

It is found in various urban communities that the streetlamp is one of the gigantic costs in a city. The expense spent is gigantic that all the sodium fume lights consume more power. The cost spent on the streetlamp can be utilized for other improvement of the nation.Currently a manual framework is utilized where the light will be made to turned ON/OFF i.e., the light will be done to turn ON at night and turned OFF in the first part of the day. Subsequently there is a ton of wastage of energy between the ON/OFF. This is one of the significant reasons for moving to the programmed framework, since there is less wastage of force and hence saving a ton of financial costs. Aside from this, different inconveniences of the current framework are portrayed beneath.

EXISTING FRAMEWORK PROBLEMS:

* Manual turning off/on of streetlamps
* More energy utilization
* High cost
* More labor

Presently moving to the proposed framework mechanized with the utilization of light sensors, in addition to the saving of energy and guaranteeing security, we can likewise see a couple of additional benefits following.

PROPOSED FRAMEWORK ADVANTAGES:

* Programmed exchanging of streetlamps
* Upkeep cost decrease
* Decrease in CO₂ emanation
* Decrease of light contamination
* Remote correspondence
* Energy saving

CODE

int smooth;

int LDR;

int threshold = 40;//sun's intensity

int brightness = 0;

int ledState = 0;

int sensor1 = 11;

int sensor2 = 8;

int sensor3 = 9;

int led1=5;

int led = 6;

int led2=2;

int carPresent = 0;

int carPresent1 = 0;

float beta = 0.65;

void setup() {

// put your setup code here, to run once:

Serial.begin(115200);

pinMode(sensor1, INPUT);

pinMode(sensor2, INPUT);

pinMode(sensor3, INPUT);

pinMode(led,OUTPUT);

pinMode(led1,OUTPUT);

pinMode(led2,OUTPUT);

}

void loop() {

smooth = smooth - (beta \* (smooth - analogRead(A0)));

delay(1);

LDR = round(((float)smooth / 1023) \* 100);

if (LDR <= 40)

brightness=0;

else

{

brightness = map(LDR, 40, 100, 0, 255);

}

checkSensors();

if (carPresent == 1)

{

ledState = 1;

digitalWrite(led,HIGH);

digitalWrite(led1,HIGH);

analogWrite(led,brightness);

analogWrite(led1,brightness);

}

else if (carPresent == 0)

{

ledState = 0;

digitalWrite(led,HIGH);

//digitalWrite(led1,HIGH);

analogWrite(led,ledState);

//analogWrite(led1,ledState);

if(carPresent1 == 1)

{

ledState = 1;

if(ledState == 1)

{

analogWrite(led1,brightness);

analogWrite(led2,brightness);

}

}

else if (carPresent1 == 0)

{

ledState = 0;

digitalWrite(led1,HIGH);

digitalWrite(led2,HIGH);

analogWrite(led1,ledState);

analogWrite(led2,ledState);

}

}

String data = (String)ledState+","+(String)brightness+";";

Serial.print(data);

// Serial.print(digitalRead(sensor1));

// Serial.print("\t");

// Serial.print(digitalRead(sensor2));

// Serial.print("\t");

// Serial.print(ledState);

// Serial.print("\t");

// Serial.println(brightness);

delay(100);

}

void checkSensors()

{

if (digitalRead(sensor1) == 0)//Car captured in 1st sensor

{

if (digitalRead(sensor2) == 1)//Car still didnt reach the 2nd sensor

carPresent = 1;

}

else if (digitalRead(sensor2) == 0)//Car reached the 2nd sensor

{ //No cars detected behind the first car

if (digitalRead(sensor1) == 1)

{

carPresent = 0;

carPresent1 = 1;

}

else if (digitalRead(sensor1) == 0 )

{

analogWrite(led,brightness);

analogWrite(led1,brightness);

analogWrite(led2,brightness);

digitalWrite(led,HIGH);

digitalWrite(led1,HIGH);

digitalWrite(led2,HIGH);

}

}

else if(digitalRead(sensor3) == 0)//car reached the 3rd sensor

{

//No cars detected behind the first car

if (digitalRead(sensor2) == 1)

{

carPresent = 0;

carPresent1 = 0;

}

else if (digitalRead(sensor2) == 0 )

{

carPresent = 0;

carPresent1 = 1;

}

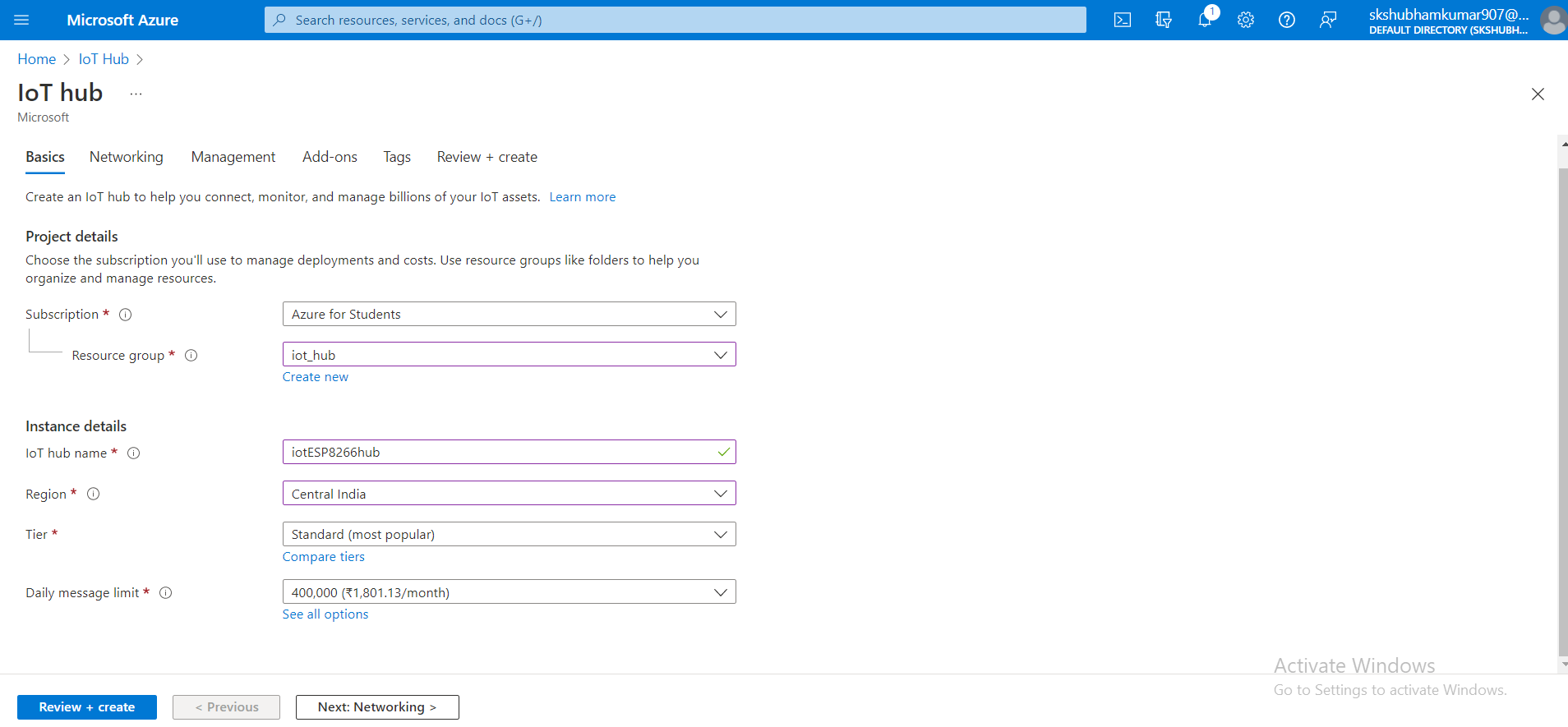
}

}

Once the code gets uploaded to the Microcontroller the IOT device will start performing the required operation.

Now, we would like to integrate the data collected from the IOT device to Azure Cloud.

For that we will create an IOT hub from the Azure portal



After creation of IoT hub, we need to add IoT device inside it.

We will get the connection string and access keys from the device.

We need to add a stream analytics job i.e. -Azure Stream Analytics is a fully managed stream processing engine that is designed to analyze and process large volumes of streaming data with sub-millisecond latencies.

The database query will be stored in JSON format. We will select SQL Query from the tab.

We need to update the code inside stream analytics job.

SELECT DeviceId

, Meter

, EventProcessedUtcTime

INTO

[database]

FROM

[device]

We need to save the Query. For testing we need to go to the Overview and click on Start.

We need to insert this data SELECT \* FROM Meter;

We can get the output on the screen.

CONCLUSION AND FUTURE SCOPE

An effective execution of this task would bring a change in the energy saving patterns of streetlights and the cost associated with it.

It decreases the upkeep cost and enables remote correspondence. Along with that it brings scope of tracking and analyzing the streetlight use patterns. It encourages more secure environment as any problem with the lights are immediately bought to attention and dealt with.

Integrating it with cloud technology also helps in creating a central repository and maintaining the whole infrastructure in a well organized and secure way.