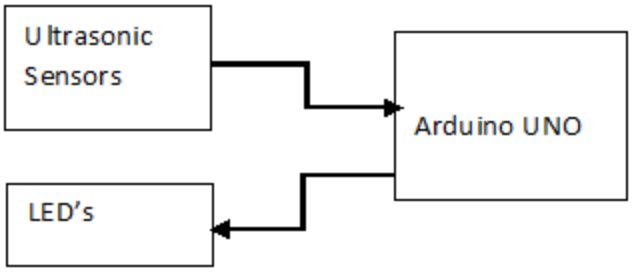
SMART PARKING SYSTEM 1

Project Description:

Features:

- Finding a parking place is not easier in shopping malls and public places.
- A smart parking system allows the public to find information about the parking slot.
- By Integrating Ultrasonic Sensors with the Parking Slots we can help the user in finding the empty slots.
- At the entrance, we can keep two LEDs for every slot and if the parking slot is filled it will indicate with Red Led and if it is empty it will indicate with green Led.
- whenever the user want to visit any malls or other places he will see the info regarding Parking Slots at the entrance itself.

Technical Arcitecture:



MATERIALS:
2 LEDs (1 green and 1 red)
2 Resistances 220
1 Ultrasonic sensor (distance sensor)
1 Build&Code UNO board
1 Breadboard

Presented by: Shivashankar, student of MITE, Moodabidri

SMART PARKING SYSTEM 2

Ultra sonic sensor

A distance sensor functions exactly as bat when it flies in the night without hitting any object. The bat sends ultrasonic waves while flying, and if this waves bounce back it means that there is an object near.

The ultrasonic sensor does the same thing. It sends ultrasonic waves and if they bounce back then it recognizes that there is an object near. The distance to the object is measured by the time it takes for the ultrasonic wave to bounce back.

The ultrasonic sensor is composed of two different modules: emitter and receiver. The emitter sends ultrasonic waves, which are sound waves that humans cannot hear (same waves that the bats send). These waves go forward until they hit an object. Then, they bounce back to the sensor, when the ultrasound is detected by the receiver.

CONNECTIONS:

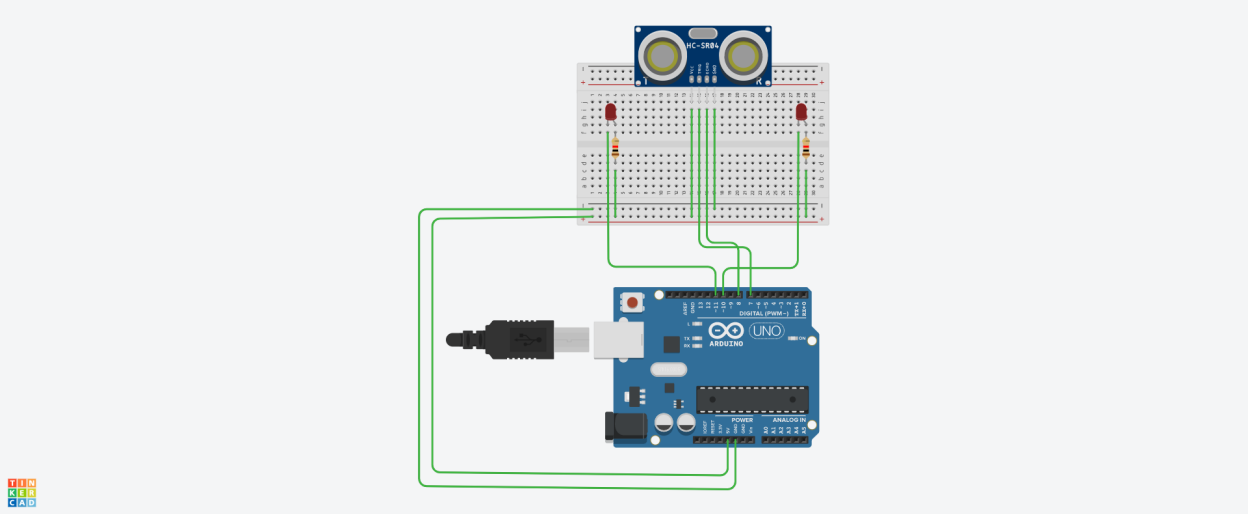
1.The protoboard receives the electricity from the Build&Code UNO 5V Pin. And goes back to the Build&Code UNO through the GND Pin. All grounds of the circuit must be connected to each other so they have the same GND value. In the image, the GND is represented with a black cable, where all the components are connected to each other and to the GND board. The red cable represents the 5V, that supply electricity the ultrasonic distance sensor.

2.To activate the green and red LED you have to connect them to a digital pin. These pins will be the ones that send electricity to the LEDs. In the image you will see that the red LED is connected to the digital pin 11 with a green cable and the green LED is connected to the digital pin 10 with a purple.

3.The ultrasonic sensor has 4 pins. You have already connected 2 of them (5V and GND). The 2 other pins are ECHO and TRIG, which 2 digital pins, these pins are the ones that will send the information to the Build&Code UNO board. In the image you will see that the TRIG pin is connected to the digital pin 7 with a yellow cable, and the ECHO pin is connected to the digital pin 8 with an orange cable.

Presented by: Shivashankar, student of MITE, Moodabidri

SMART PARKING SYSTEM 3




```
int trigPin = 7;  //Define the pins that you will work with
int echoPin = 8;
int LEDR = 10;
int LEDV = 11;
float Speed = 0.0343;  // Sound speed at cm/us
long duration, distance;
void setup()
{
  pinMode(trigPin, OUTPUT);  //Define digital pin 7 as an output
  pinMode(echoPin, INPUT);   //Define digital pin 8 as an input
  pinMode(LEDR, OUTPUT);   //Define digital pin 10 as an output
  pinMode(LEDV, OUTPUT);   //Define digital pin 11 as an output
  digitalWrite (LEDR , LOW);  // Define digital pin 10 in a low
 status
  digitalWrite (LEDV , LOW);  //Define digital pin 11 in a low status
}
oid loop()
{
  digitalWrite(trigPin, LOW);        // Make sure that the TRIG is
```

Presented by: Shivashankar, student of MITE, Moodabidri

SMART PARKING SYSTEM 4

```
  deactivated
19  delayMicroseconds(2);                // Make sure that the TRIG is in
  LOW
20  digitalWrite(trigPin, HIGH);         // Activate the output pulse
21  delayMicroseconds(10);               // Wait for 10µs, the puls
  remains active during this time
22  digitalWrite(trigPin, LOW);          //Stop the pulse and wait for
  ECHO
23  duration = pulseIn(echoPin, HIGH) ; // pulseIn measures the time
  since the defined pin (echoPin) changes its status from low to high
  (from 0 to 1)
24  distance = Speed* duration / 2;    //Divide by 2 because we want to
  have only the “go” time, not the “go and back” time     // and divide
  by 29,1 because 1 is divided by the sound speed (1/SpeedSound) at
  cm/us
25  if ( distance < 20)
26  {
27    digitalWrite (LEDR , HIGH);      //If the sensor detects a
  distances less than 20 cm the red LED turns on
28    digitalWrite (LEDV , LOW);       //and turns off the green LED
29  }
30  else
31  {       // otherwise
32    digitalWrite (LEDR , LOW);     // turn off the red LED
33    digitalWrite (LEDV , HIGH);    //turn on the green LED
34  }
35 }
```