Automatic Car Parking Management System

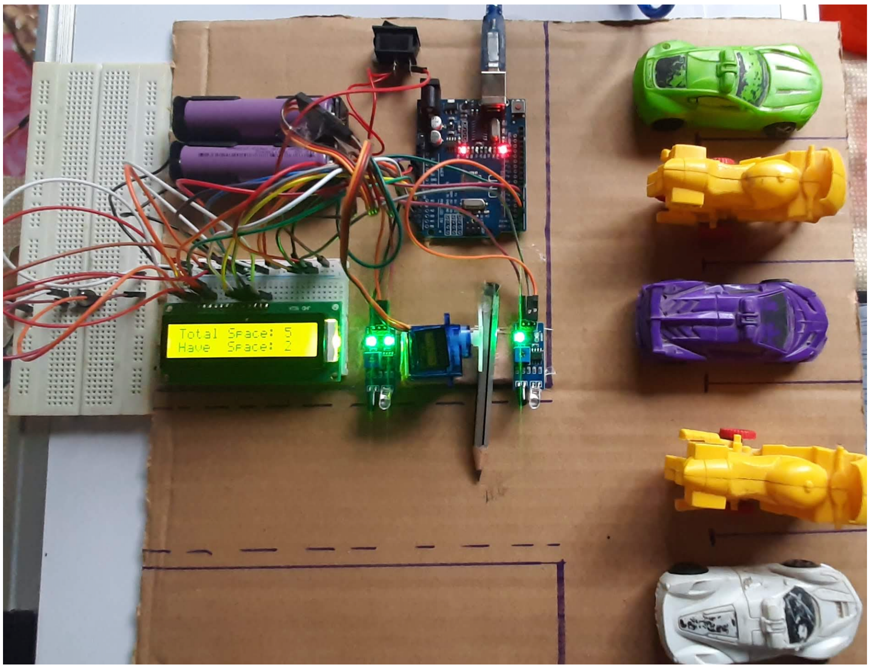
Component Details –

1. Solderless Breadboard
2. Arduino Uno
3. IR Sensor x 2
4. Servo Motor SG-90
5. 16×2 LCD Display
6. 4.7k Resistor
7. 1k Resistor
8. Male to Male Jumper Wires
9. power on off switch
10. 100R Resistor
11. 18650 Battery Holder – 2 Cell
12. 18650 Battery Cell 3.7V x 2

Description –

In smart cities, there is a greater need for new and effective technology to tackle many of the problems that are visible on the surface, as well as to make cities less crowded. Finding a parking spot is one of the most aggravating issues for drivers. Particularly in public venues such as shopping malls, 5-star hotels, and multiplex cinema halls. Even within the park, drivers waste time and fuel hunting for a spot to park their cars. This will damage the driver's emotions as well as pollute the environment while searching for a parking spot. In this study, we create and design a smart parking system that effectively addresses these issues. Much research have been conducted in recent years with the goal of reducing car parking issues and making it more convenient and humane. It has recommended a smart parking system survey. They concentrate on practical smart parking technologies developed to address existing issues using a wireless sensor network and real-time data processing from the sensors. The system appears to be unfixable and employs a complicated access technology; additionally, there is no guidance mechanism for parking places. The Arduino Uno is used to create a smart car parking system. The device uses IR sensors mounted in the parking slots to detect empty slots and assists the driver in finding parking in a new city. The system lacks a payment mechanism as well as guide technology that can automatically find available parking spaces. The goal of the smart auto parking initiative is to make parking simple and straightforward. This project assists car drivers in parking their vehicles with the least amount of wasted time by providing reliable information on the availability of parking spaces. The servo motors, LCD display, and IR sensor are all connected to an Arduino Uno microcontroller unit. The LCD shows how much space is available, and the IR sensors keep track of how many automobiles enter and exit the parking place. The IR sensors identify whether a parking place is available.

Circuit Diagram –



Explanation –

Importing the “LiquidCrystal.h” library at the beginning of the code is necessary in order to communicate with the 16×2 LCD screen. After initializing the LCD pins as A0, A1, A2, A3, A4, and A5, the “Servo.h” library is included to allow servo motor control. The IR sensors are connected to pins 2 and 4 of the Arduino Uno, and the servo motor is initialized with the name “myservo1.”
The first thing the software does is define the total number of parking spots, which in this example is five. If more places are required, they may be added later. Together with two flags, “flag1” and “flag2,” both initially set to 0, the variable “space” is initialized. The servo motor (‘myservo1') is connected to pin 3 and the IR sensors are set up in INPUT mode in the “setup()” method. When the barrier is closed, the servo motor’s starting setting of 100 degrees is reached. The “Car Parking System” message is then displayed on the LCD display for the first two seconds before being cleared. The entire value of parking spaces is assigned to the’space’ variable. The “loop()” function is the central component of the code. The first IR sensor is read first. It opens the barrier if the sensor detects LOW, which indicates the presence of an automobile, and ‘flag1' is 0 (which indicates that the barrier is closed). When’space’ is larger than 0 and there is a parking spot available, the barrier opens, ‘flag1’ is set to 1, and ‘flag2' is checked. The servo motor advances to 0 degrees (barrier open) and’space’ reduces by 1 if ‘flag2’ is also 0. The software shows “Sorry not Space Available” on the LCD if there are no parking spaces available. Next, the second IR sensor is read by the code. It sets ‘flag2’ to 1 if the sensor detects LOW and ‘flag2' is 0. The barrier opens and’space’ rises by one if ‘flag1’ is 0. The automobile entry and egress is efficiently managed by this procedure. Before closing the barrier by turning the servo motor to 100 degrees, the code adds a 1-second delay when an automobile passes both IR sensors and “flag1” and “flag2” become 1. Following that, both flags are reset to 0, reactivating both IR sensors. The LCD gives consumers access to real-time information by showing the overall number of parking spaces as well as the available spaces. After the code has been clarified, the Arduino IDE may be used to upload it to the Arduino Uno.