به نام خدا

پروژه نهایی سیستم های نهفته و بی‌درنگ

سرکار خانم دکتر عبدی

علی الهی

۹۷۲۴۷۴۳

برای پیاده‌سازی این پروژه از سایت Tinkercad استفاده شده است.

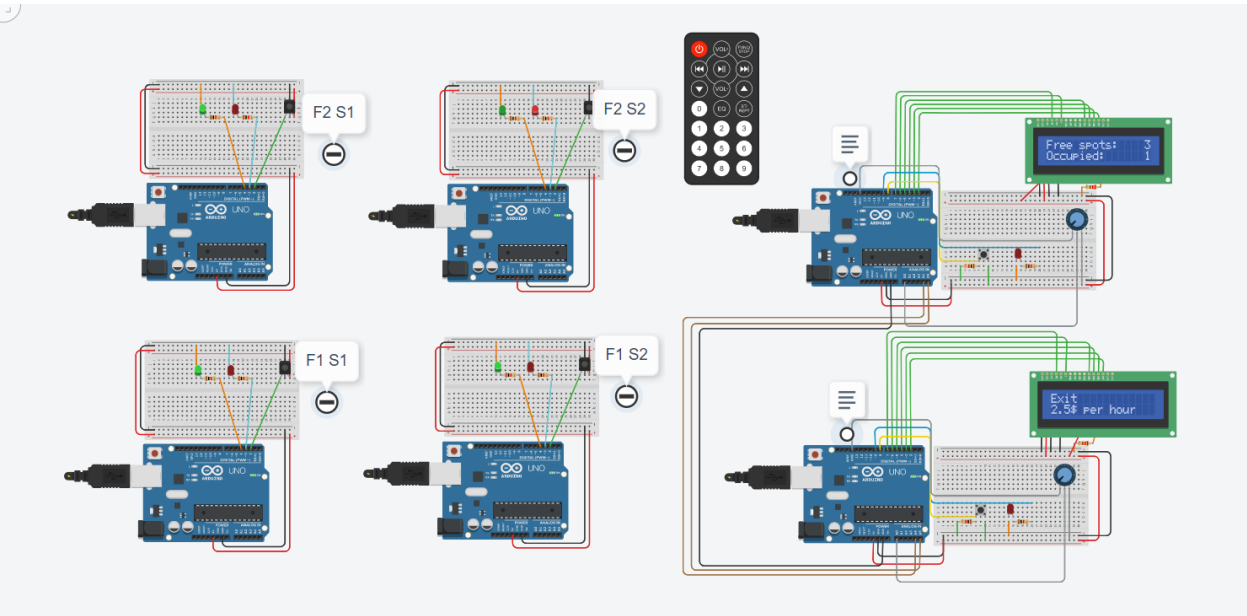
لینک تینکرکد:

https://www.tinkercad.com/things/iOOz43CXVcQ-es-project-public-parking/editel

لینک توضیحات و کارکرد مدار را از لینک زیر میتوانید مشاهده کنید:

https://www.aparat.com/v/MGOEc/%D8%B4%D8%A8%DB%8C%D9%87_%D8%B3%D8%A7%D8%B2%DB%8C_%D8%B3%DB%8C%D8%B3%D8%AA%D9%85_%DA%A9%D9%86%D8%AA%D8%B1%D9%84%D8%B1_%DB%8C%DA%A9_%D9%BE%D8%A7%D8%B1%DA%A9%DB%8C%D9%86%DA%AF_%D9%87%D9%88%D8%B4%D9%85%D9%86%D8%AF_%D8%A8%D8%A7_Arduino

قسمت اول-مدار طراحی شده



در این مدار از ۶ Arduino استفاده شده است.

بخش ورودی: از یک Arduino یک LCD یک LED و یک دکمه استفاده شده‌است. در اینجا LED نشان دهنده اهرم بالا رونده است که ۱۵ ثانیه در عمل باز میشود و به خودرو ها اجازه عبور میدهد.

بخش خروجی: کاملا مشابه بخش ورودی با این تفاوت که کد های متفاوتی دارند.

سنسور های IR: استفاده از چهار سنسور IR که با استفاده از یک ریموت، کنترل میشوند.

قسمت دوم-کد های مربوط به هر Arduino

بخش ورودی: از کتابخانه‌های Wire و LiquidCrystal استفاده شده است.

```
15  void setup()
16  {
17      pinMode(btn, INPUT);
18      pinMode(led, OUTPUT);
19      Wire.begin(1);
20      Wire.onRequest(exit_parking);
21      lcd.begin(16, 2);
22      lcd.setCursor(0, 0);
23      lcd.print("Welcome !!!      ");
24  }
```

در این بخش ابتدا led را در مود خروجی و دکمه را در مود ورودی قرار میدهیم. سپس تنظیمات مربوط به وایر را انجام میدهیم تا هنگام فشرده شدن دکمه خروج در مدار بعدی متد exit_parking فراخوانی شود. در آخر هم پیام خوش آمدید را هم روی نمایشگر نشان میدهیم.

```
26  void loop()
27  {
28      btnState = digitalRead(btn);
29      //delay(50);
30      showParkingStatus();
31      if (btnState == HIGH)
32      {
33          if (!isFull())
34          {
35              enQueue();
36              numberOfCars++;
37              printMySpot();
38              digitalWrite(led, HIGH);
39              delay(100);
40              digitalWrite(led, LOW);
41          }
42      }
43  }
44
```

در قسمت تکرار شونده برنامه هم منتظر فشرده شدن دکمه ورودی میمانیم، در همین هنگام متد showParkingStatus فراخوانی میشود تا وضعیت پارکینگ را نشان دهد. هنگامی که دکمه فشرده شد چک میکنیم که آیا پارکینگ پر است یا خیر، در صورت پر نبودن پارکینگ یک ماشین به پارکینگ اضافه میکنیم و تعداد ماشین های موجود در پارکینگ را یک عدد افزایش میدهیم. سپس متد printMySopt فراخوانی میشود تا روی نمایشگر ورودی مکانی را که ماشین مورد نظر باید پارک شود نمایش دهد. در آخر هم led موردنظر به اندازه‌ای روشن میماند که نشان دهنده‌ی اهرم ورودی است. (در ادامه بدنه این متد ها نیز بررسی خواهد شد)

```
45  void showParkingStatus()
46  {
47      if (isFull())
48      {
49          lcd.setCursor(0, 0);
50          lcd.print("Parking is Full ");
51          lcd.setCursor(0, 1);
52          lcd.print("                ");
53      }
54      else
55      {
56          lcd.setCursor(0, 0);
57          lcd.print("Free spots:     ");
58          lcd.print(4 - numberOfCars);
59          lcd.setCursor(0, 1);
60          lcd.print("Occupied:       ");
61          lcd.print(numberOfCars);
62      }
63  }
```

در متد showParkingStatus ابتدا چک میکنیم که آیا پارکینگ پر است یا خیر، در صورت پر بودن نمایشگر ورودی پیامی چاپ میکند مبتنی بر اینکه پارکینگ پر است و ماشین ها اجازه ورود نخواهند داشت. در غیر اینصورت نمایشگر تعداد جاهای خالی موجود در پارکینگ و همینطور تعداد جاهای اشغال شده را نشان میدهد.

```
65  void exit_parking()
66  {
67      if (!isEmpty())
68      {
69          numberOfCars--;
70          Wire.write(deQueue());
71      }
72  }
```

این متد هنگامی فراخوانی میشود که در Arduino خروجی دکمه خروج فشرده شود، در اینصورت اگر ماشینی در پارکینگ موجود باشد، تایم ورودی آن خودرو به قسمت خروجی ارسال میشود تا هزینه نهایی محاسبه شود.

```
74  void printMySpot()
75  {
76      lcd.setCursor(0, 0);
77      lcd.print("Your Spot:      ");
78      lcd.setCursor(0, 1);
79      if (4 - rear > 2)
80      {
81          lcd.print("Floor:2   Spot:");
82          lcd.print(2 - rear);
83      }
84      else
85      {
86          lcd.print("Floor:1   Spot:");
87          lcd.print(4 - rear);
88      }
89  }
```

در این متد مکانی که ماشین ورودی باید در آن پارک کند را نشان میدهد، از طبقه دوم و مکان دوم آن شروع میکند تا به طبقه اول و مکان اول برسد.

دو متد isFull و isEmpty نیز پر یا خالی بودن پارکینگ را بررسی میکنند.

متد enqueue یک ماشین به پارکینگ اضافه میکند، یعنی زمان ورود خودرو را در یک آرایه ذخیره میکند.

متد dequeue نیز یک ماشین را از پارکینگ خارج میکند و زمان ورود خودرو را برمیگرداند.

بخش خروجی: در این فایل هم از همان دو کتابخانه استفاده شده است.

```
10  void setup()
11  {
12      pinMode(btn, INPUT);
13      pinMode(led, OUTPUT);
14      Wire.begin();       You, 6 days ago • Feat: Entrance and Exit gat
15      lcd.begin(16, 2);
16  }
```

در این بخش ابتدا led را در مود خروجی و دکمه را در مود ورودی قرار میدهیم. تنظیمات وایر را انجام میدهیم و lcd را ستاپ میکنیم.

```
18  void loop()
19  {
20      lcd.setCursor(0, 0);
21      lcd.print("Exit            ");
22      lcd.setCursor(0, 1);
23      lcd.print("2.5$ per hour     ");
24      btnState = digitalRead(btn);
25      //delay(100);
26      if (btnState == HIGH)
27      {
28          Wire.requestFrom(1, 1);
29          if (Wire.available())
30          {
31              int exit_time = millis() / 100;
32              int enter_time = Wire.read();
33              if (enter_time != 0)
34              {
35                  lcd.setCursor(0, 0);
36                  lcd.print("Your Payment:   ");
37                  lcd.setCursor(0, 1);
38                  double cost = (exit_time - enter_time) * fee;
39                  lcd.print(cost);
40                  lcd.print("$              ");
41                  digitalWrite(led, HIGH);
42                  delay(100);
43                  digitalWrite(led, LOW);
44              }
45          }
46      }
47  }
```

در این قسمت ابتدا اطلاعاتی از قبیل خروج و مقدار هزینه را نشان میدهد. در ادامه منتظر فشرده شدن دکمه خروج میمانیم.

بعد از فشرده شدن دکمه خروج ریکوئستی به سمت ورودی میزنیم تا تایم ورود خودرو را دریافت کنیم، سپس هنگامی که تایم دریافت شد، تایم خروج را هم محاسبه کرده و در مقدار fee ضرب میکنیم تا هزینه نهایی بدست بیاید.

هزینه نهایی را روی نمایشگر نشان میدهیم(در این مرحله کاربر هزینه را پرداخت میکند) و دوباره اهرم برای مدت زمانی بالا میرود تا خودرو خارج شود.

بخش سنسور IR: در این بخش از کتابخانه IRremote استفاده شده است.

```
12  void setup()
13  {
14      irrecv.enableIRIn(); // Start the receiver
15      pinMode(red_led, OUTPUT);
16      pinMode(green_led, OUTPUT);
17      digitalWrite(green_led, HIGH);
18  }
```

در ابتدا ریموت را استارت میکنیم در ادامه دو led سبز و قرمز را در حالت خروجی قرار میدهیم و led قرمز را روشن میکنیم به این معنی که در ابتدا جای پارک خالیست.

```
20  void loop()
21  {
22      if (irrecv.decode(&results))
23      {
24          if (results.value == 16582903)
25          {
26              ledOn = !ledOn;
27              if (ledOn)
28              {
29                  digitalWrite(red_led, HIGH);
30                  digitalWrite(green_led, LOW);
31              }
32              else
33              {
34                  digitalWrite(red_led, LOW);
35                  digitalWrite(green_led, HIGH);
36              }
37          }
38          irrecv.resume();
39      }
40  }
41
```

در این مرحله که بخش اصلی برنامه است، منتظر فشرده شدن دکمه روی ریموت می‌ماند، بعد از فشرده شدن دکمه در شرط چک میشود که کدام دکمه فشرده شده است. در مثال بالا value ریموت برای دکمه یک آورده شده است. اگر دکمه ۱ روی ریموت فشرده شود چراغ قرمز روشن و سبز خاموش میشود. دفعه بعدی چراغ سبز روشن و چراغ قرمز خاموش میشود.