



دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

پروژه درس ساختار و زبان کامپیوتر

سیستم قفل در هوشمند

اعضاء تیم

اشکان مجیدی

ایلیا محروقی

پریا حاجی پور

استاد درس

دکتر اسدی

بهمن ۱۴۰۱

چکیده

قفل هوشمند یکی از انواع قفل‌هایی است که بسیار پرکاربر شده‌است. این نوع قفلها با رمز عددی، تشخیص چهره، اثر انگشت و ... باز میشوند.

فهرست مطالب

۱	وسایل مورد نیاز و راه اندازی	۱
۱	۱-۱ برد Arduino Uno	۱
۳	۲-۱ LED	۳
۴	۳-۱ 2×16 LCD	۴
۴	۴-۱ s3003 servo motor	۴
۶	۵-۱ زنگ اخبار	۶
۷	۶-۱ 4×4 flat keypad	۷
۸	۲ اتصالات	۸
۸	۱-۲ اتصال keypad به برد Arduino	۸
۱۰	۲-۲ وصل کردن led و buzzer	۱۰
۱۱	۳-۲ وصل کردن LCD و servo	۱۱
۱۴	۳ نتیجه گیری	۱۴
۱۴	۱-۳ کد نهایی	۱۴
۱۵	۲-۳ نتیجه نهایی	۱۵

فهرست تصاویر

۱-۱	بخش‌های اصلی برد آردوینو	۲
۲-۱	قطعات برد آردوینو	۲
۳-۱	نحوه اتصال LED به برد آردوینو	۳
۴-۱	پایه‌های LCD	۴
۵-۱	مثالی از نحوه اتصال LCD به Arduino	۴
۶-۱	نحوه اتصال موتور servo به برد آردوینو	۵
۷-۱	نحوه اتصال Buzzer به برد آردوینو	۶
۸-۱	پایه‌های دو نوع keypad	۷
۱-۲	نمایش پسورد روی lcd	۱۱
۲-۲	موتور سروو	۱۲
۳-۲	نمایش تمام اجزا	۱۳

فصل ۱

وسایل مورد نیاز و راه اندازی

این پروژه از قطعات الکترونیکی گوناگونی تشکیل شده که نام آن ها در زیر آمده است و در ادامه به بررسی هریک و نحوه راه اندازی آن ها می پردازیم.

- برد Arduino Uno

- LED ساده

- 2×16 LCD

- s3003 servo motor

- زنگ اخبار (buzzer)

- 4×4 flat keypad

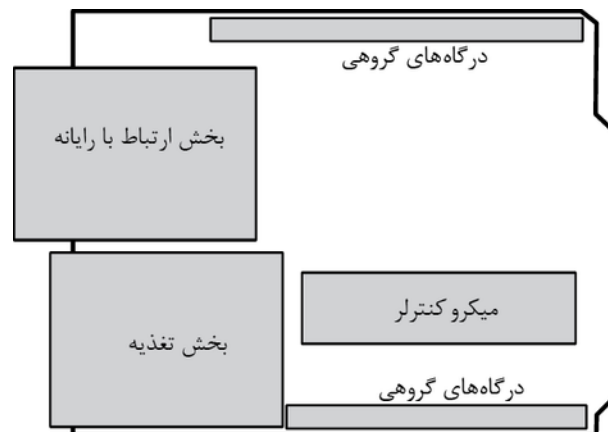
۱-۱ برد Arduino Uno

آردوینو یک پلتفرم سخت افزاری و نرم افزاری open source است. پلتفرم آردوینو شامل یک میکروکنترلر تک بردی open source است که قسمت سخت افزار آردوینو را تشکیل می دهد. علاوه بر این، پلتفرم آردوینو یک نرم افزار آردوینو IDE که به منظور برنامه نویسی برای بردهای آردوینو طراحی شده است و یک بوت لودر نرم افزاری که بر روی میکروکنترلر بارگذاری می شود را در بر می گیرد. پلتفرم آردوینو به منظور تولید سریع و ساده پروژه های سخت افزاری تعاملی و ساخت وسایلی که با محیط تعامل

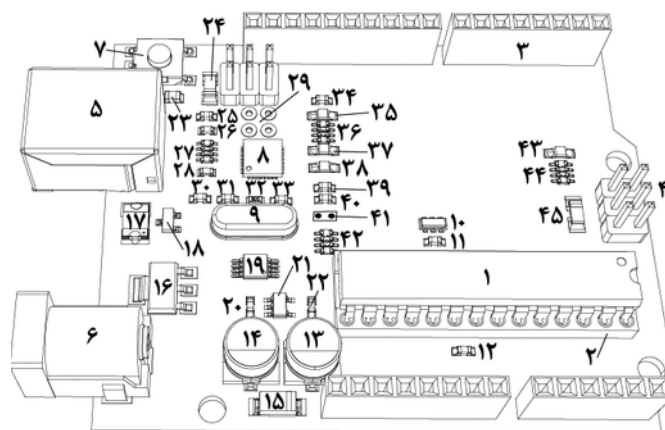
داشته باشند طراحی شده است. شیوه نگارش دستورها در نرم افزار آردوینو تا حدود زیادی مشابه برنامه نویسی به زبان C و C++ است.

مدل های مختلفی از بردهای سخت افزاری آردوینو وجود دارد که یکی از آنها، مدل اونو است. آردوینو اونو (Arduino UNO) یکی از پرکاربردترین مدل های سخت افزار آردوینو است. سخت افزار آردوینو اونو از چهار بخش کلی زیر تشکیل می شود:

- میکروکنترلر
- بخش تغذیه
- بخش ارتباط با رایانه
- درگاه های گروهی



شکل ۱-۱: بخش های اصلی برد آردوینو



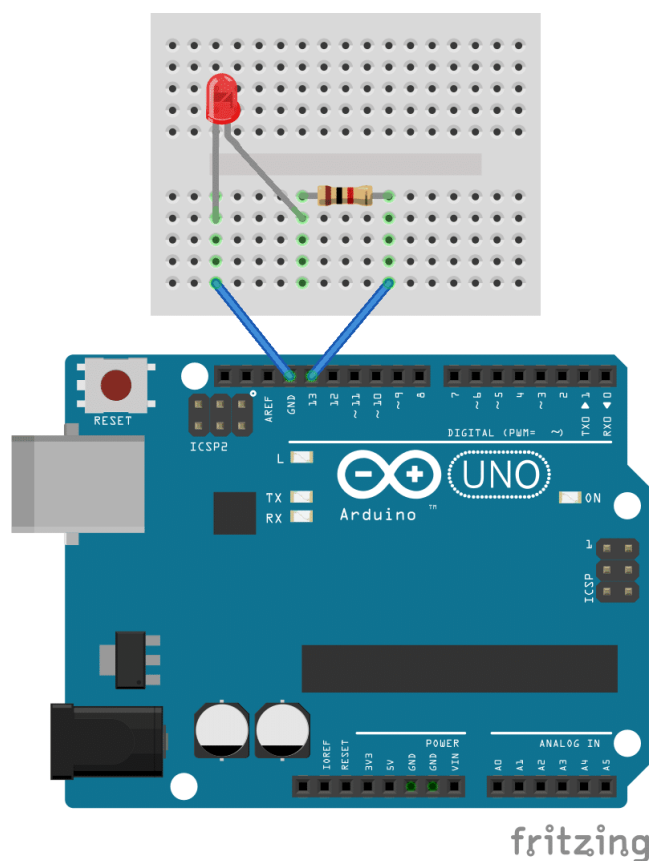
شکل ۱-۲: قطعات برد آردوینو

برای اتصال برد به کامپیوتر از قطعه‌ی شماره ۵ استفاده می شود که یک USB prot Type B است.

همچنین قطعات الکترونیکی دیگر استفاده شده در پروژه را به بخش ۳ که درگاه ارتباطی برد از نوع پین هدر مادگی است، وصل می‌کنیم. این بخش ارتباط را با پایه‌های میکروکنترلر برقرار می‌کند.

۲-۱ LED

می‌دانیم برای روشن شدن ال‌ای‌دی باید یک سر آن را به Ground و سر دیگر را ولتاژ متصل کنیم. اما به دلیل امکان سوختن LED در ولتاژهای بالا، یک عدد مقاومت در مدار تعبیه می‌کنیم. قابل توجه است که سر مثبت آن باید به ولتاژ وصل باشد. حال چون می‌خواهیم روشن یا خاموش بودن LED را با کمک برد آردوینو کنترل کنیم، پس بجای اینکه سر مثبت را به ولتاژ مستقیم وصل کنیم، آن را به یکی از درگاه‌های ارتباطی برد متصل کرده و هنگامی که بخواهیم LED را روشن کنیم، آن درگاه را برابر ۱ می‌کنیم. در غیر این صورت برابر ۰ قرارش می‌دهیم.



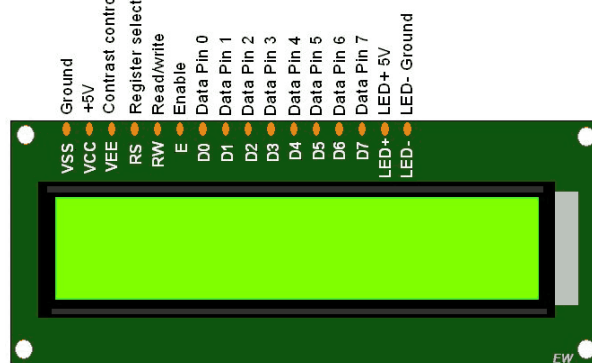
شکل ۱-۳: نحوه اتصال LED به برد آردوینو

۳-۱ 2×16 LCD

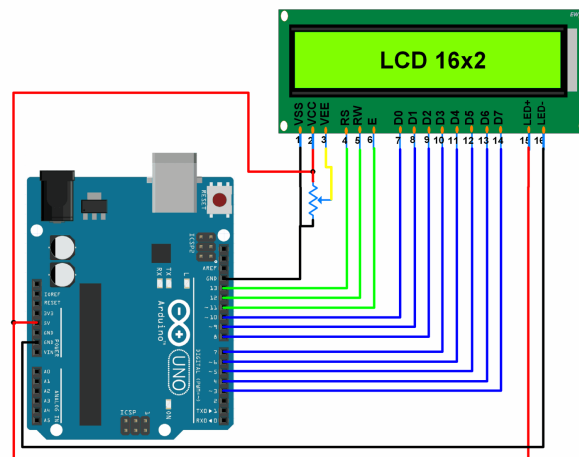
ال سی دی ها (Liquid Crystal Displays) در سیستم های embedded برای نمایش دادن پارامتر های مختلف و وضعیت سیستم استفاده می شوند.

2×16 LCD یک دستگاه با ۱۶ پایه است که دو سطر دارد و هریک ۱۶ کاراکتر دارند. از این LCD می توان به دو صورت ۸- بیت و ۴- بیت استفاده کرد. همچنین میتوان با استفاده از آن کاراکتر های شخصی سازی شده ساخت.

این LCD دارای ۸ پایه برای دیتا (D0-D7) و ۳ پایه کنترلی می باشد (RS, RW, En).



شکل ۱-۴: پایه های LCD

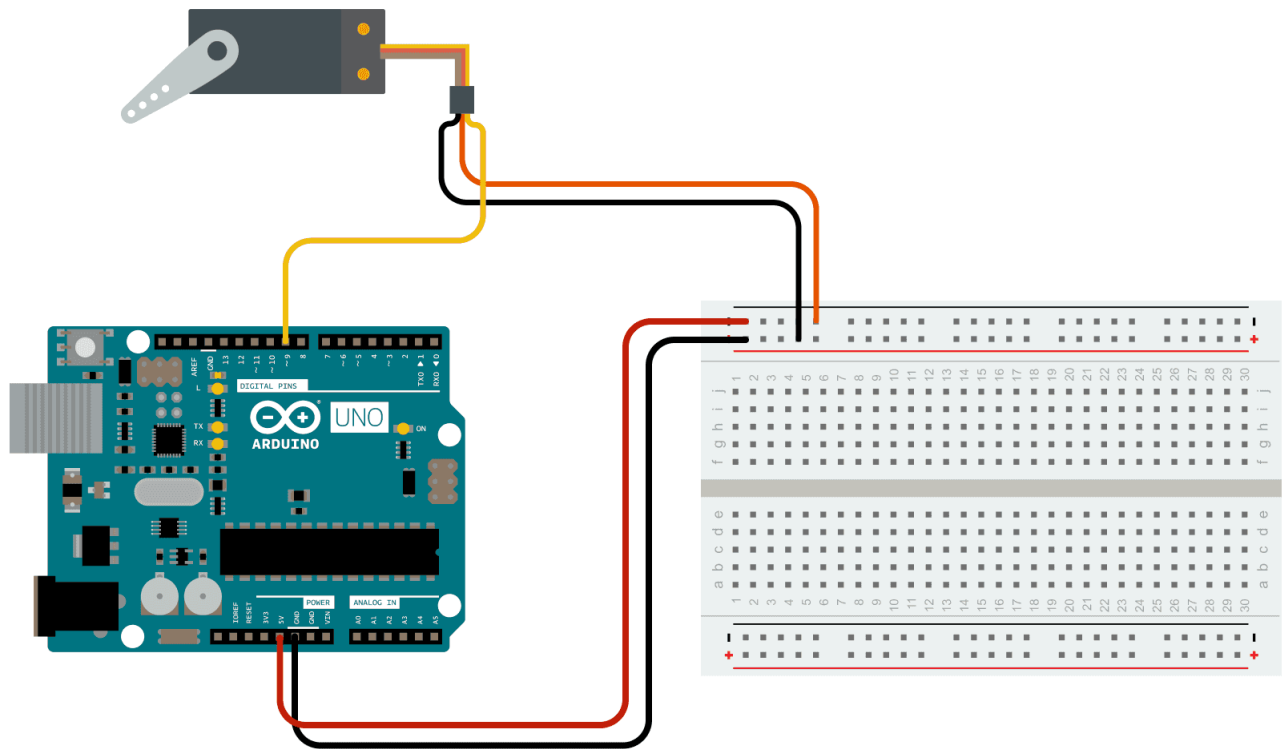


شکل ۱-۵: مثالی از نحوه اتصال LCD به Arduino

۴-۱ s3003 servo motor

برای اتصال موتور servo هم مانند LED عمل می کنیم با این تفاوت که دو ورودی آن را به Ground و ۵v (VCC) وصل کرده و یک ورودی دیگر را به منظور کنترل به یکی از درگاه های ارتباطی برد وصل

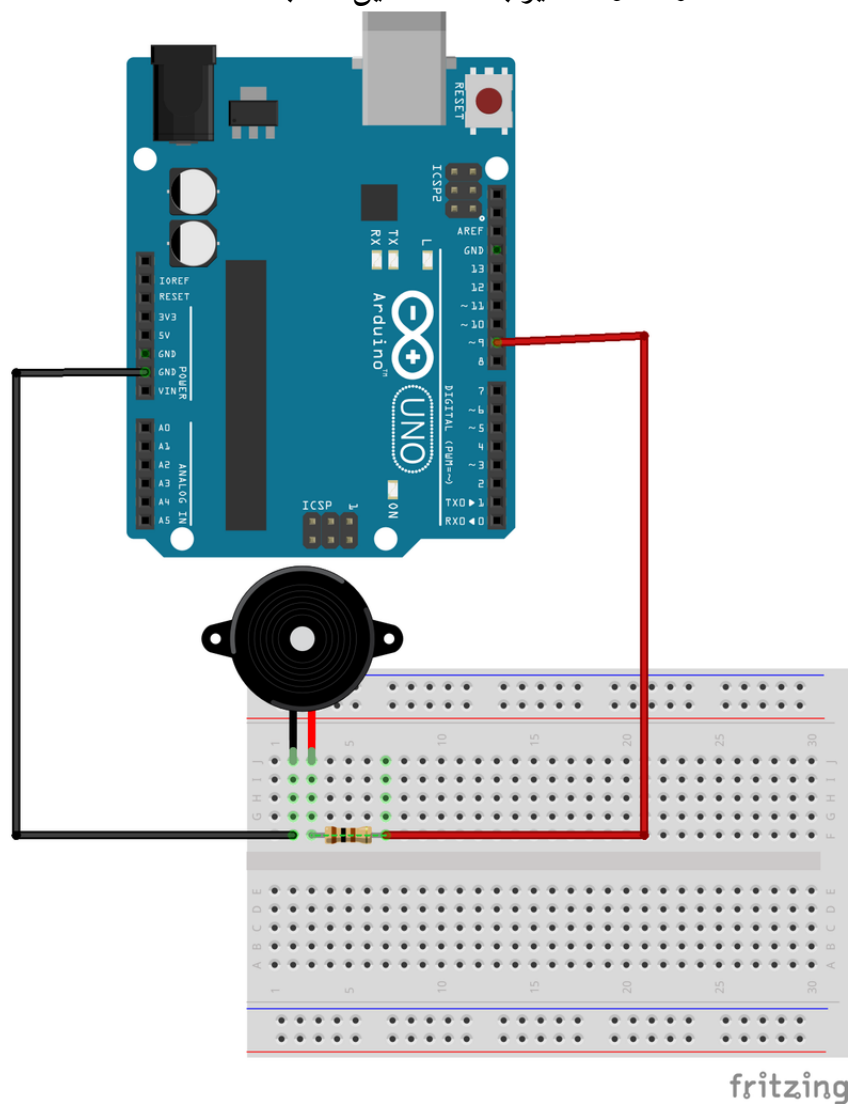
می‌کنیم. حال زمان، نحوه‌ی چرخش و ... را توسط آن درگاه و با کد ورودی برد کنترل می‌کنیم.



شکل ۱-۶: نحوه اتصال موتور servo به برد آردوینو

۵-۱ زنگ اخبار

در این بخش هم مانند LED عمل می‌کنیم به گونه‌ای که یک ورودی بازر را به Ground و ورودی دیگر را به یکی از درگاه‌های برد متصل می‌کنیم تا با استفاده از آن روشن یا خاموش بودن بارز را کنترل کنیم. علت استفاده از مقاومت نیز به همان دلیل مشابه LED است.

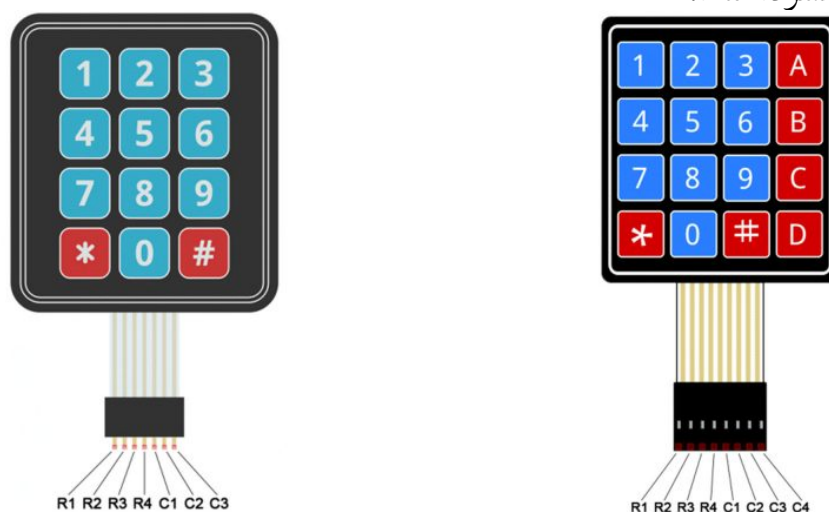


شکل ۱-۷: نحوه اتصال Buzzer به برد آردوینو

۶-۱ 4 × 4 flat keypad

از دیرباز تا کنون راه‌های گوناگونی برای ورودی دادن داده به سیستم‌ها وجود داشته مانند دکمه، عکس و .. کپد یکی از موثرترین روش‌ها برای ورودی دادن است. مزیت آن ارزان بودن و راحتی کار با آن است و به همین دلیل است که کپد‌ها هم‌جا اطراف ما هستند، مانند کیبورد کامپیوتر.

کپد انواع مختلفی دارد که در این پروژه از نوع 4×4 آن استفاده شده است. این وسیله ۸ پایه دارد که ۴ تا از آن‌ها مربوط به سطر هستند و ۴ تا مربوط به ستون و به این روش مشخص می‌کنند که چه دکمه‌ای فشرده شده.



شکل ۱-۸: پایه‌های دو نوع keypad

فصل ۲

اتصالات

در این قسمت به نحوه اتصال قطعات و بارگزاری کد زده شده روی آنها می پردازیم.

- اتصال keypad به برد Arduino

- وصل کردن led و buzzer

- وصل کردن lcd و servo

۱-۲ اتصال keypad به برد Arduino

همانطور که در قسمت قبل گفته شده است. برای ارتباط بین این دو دیوایس باید ۸ خروجی keypad را به ۸ ورودی Arduino وصل کرد. برای این کار ما یک آرایه ۳*۴ در کدمان در نظر می گیریم که هر خانه آرایه کرکتری است که روی keypad قرار دارد و با توجه به دکمه‌ای که روی آن است بتوانیم کار مورد نظر را انجام دهیم. دقت کنید که درست است که keypad ما ۴*۴ است ولی از ستون سمت راست استفاده نمی کنیم. پس از کد زیر استفاده می کنیم:

```
1 const byte ROWS = 4;
2 const byte COLS = 3;
3 char keys[ROWS][COLS] = {
4   { '1', '2', '3' },
5   { '4', '5', '6' },
6   { '7', '8', '9' },
```

```

7   { '*', '0', '#' }
8   };
9   byte rowPins[ROWS] = { 9, 8, 7, 6 };
10  byte colPins[COLS] = { 2, 4, 3 }; // cause 5 were busy we used 6
11  Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

```

با استفاده از کد زیر کرکتر ورودی را می گیریم:

```

1  char key = keypad.getKey();

```

برای اینکه متوجه بشیم که کد کاربر # را نگه داشته است نیز از کد زیر استفاده می کنیم.

```

1  void holdListener() {
2      switch (keypad.getState()) {
3          case PRESSED:
4              startFlag = false;
5              break;
6          case RELEASED:
7              startFlag = false;
8              break;
9
10         case HOLD:
11             if (!startFlag && inputKey == '#') {
12                 holdStart = millis();
13                 startFlag = true;
14                 delay(100);
15             } else if (startFlag && (millis() - holdStart >= 5000)) {
16                 reset();
17             }
18             break;
19     }
20 }

```

۲-۲ وصل کردن led و buzzer

همانطور که گفته شده است برای وصل کردن led صرفاً به یک خروجی از برد نیاز داریم که یک سر led را به gnd و سر دیگر آن را به خروجی Arduino وصل کنیم. برای buzzer نیز به همین شکل است. پس در کل ۲ خروجی برای دو led سبز و قرمز و یک خروجی برای خروجی buzzer می‌خواهیم. پس برای initialize کردن آن‌ها از کد زیر استفاده می‌کنیم:

```
1 #define greenLed 5
2 #define redLed 11
3 #define buzzer 10
```

و در تابع setup از کد زیر استفاده می‌کنیم:

```
1 pinMode(greenLed , OUTPUT);
2 pinMode(redLed , OUTPUT);
3 pinMode(buzzer , OUTPUT);
```

پس برای پسورد اشتباه این تیکه از کد را خواهیم داشت:

```
1 if (wrongPass) {
2     lcd.clear();
3     lcd.setCursor(0, 0);
4     lcd.print("Wrong Pass");
5     digitalWrite(redLed , HIGH);
6     if (wrongPassCount < 3) {
7         delay(10000);
8     } else {
9         for (double i = 0; i <= 5000; i += 5) {
10             tone(buzzer , i);
11             delay(30);
12         }
13         delay(30000);
14         noTone(buzzer);
15         wrongPassCount = 0;
16         // delay(20000);
17     }
18     lcd.clear();
19     lcd.print("Enter your pass:");
```

```

20  lcd.setCursor(0, 1);
21  digitalWrite(redLed, LOW);
22  wrongPass = false;
23  input = "";
24  readFlag = true;
25  }

```

۳-۲ وصل کردن LCD و servo

همانطور که گفته شده است باید درگاه‌های LCD را به برد Arduino وصل کنیم. برای اینکار ما برد Arduino را به یک board bread وصل می‌کنیم. حال در کدمان به صورت زیر عمل می‌کنیم:

```

1 LiquidCrystal lcd(13, A0, A1, A2, A3, A4, A5);

```

و در تابع setup به صورت زیر عمل می‌کنیم:

```

1 lcd.begin(16, 2); // set up the LCD's number of columns and rows

```

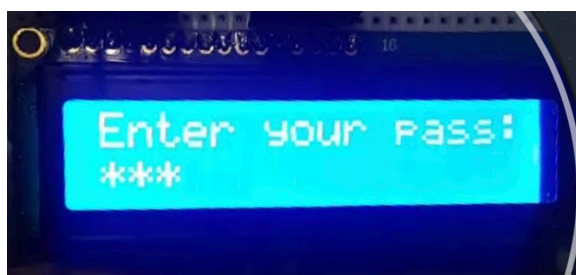
برای نوشتن روی آن کافی است کد زیر اجرا شود:

```

1 lcd.clear();
2 lcd.setCursor(0, 0);
3 lcd.print("Password accepted");

```

استفاده کامل از lcd در منطق کد در قسمت بعدی آمده است. از lcd برای نشان دادن پسور و ورودی استفاده می‌کنیم.



شکل ۲-۱: نمایش پسورد روی lcd

برای servo کافی است کد زیر را در قسمت setup قرار دهیم:

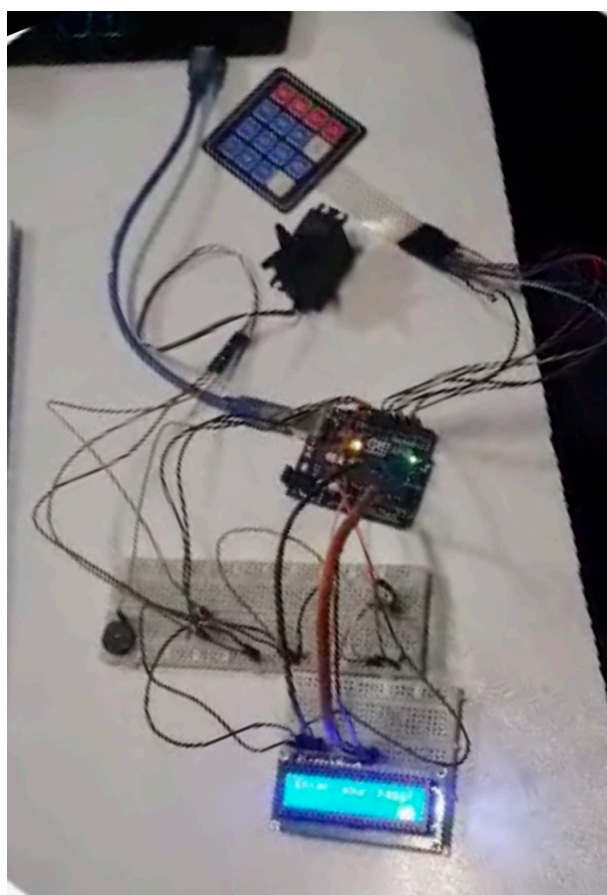
```
1 myservo.attach(12);
```



شکل ۲-۲: موتور سرو

برای حرکت servo نیز کافی است کد زیر اجرا شود:

```
1  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180
    degrees
2      // in steps of 1 degree
3      myservo.write(pos); // tell servo to go to position in variable 'pos'
4      delay(30);           // waits 15ms for the servo to reach the
    position
5  }
6  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0
    degrees
7      myservo.write(pos);           // tell servo to go to position
    in variable 'pos'
8      delay(30);                     // waits 15ms for the servo to
    reach the position
9  }
```

شکل ۲-۳: نمایش تمام اجزا

فصل ۳

نتیجه گیری

در این فصل، ضمن جمع بندی نتایج جدید ارائه شده در پایان نامه یا رساله، مسائل باز باقی مانده و همچنین پیشنهادهایی برای ادامه ی کار ارائه می شوند.

۱-۳ کد نهایی

همانطور که مشاهده می کنید در قبل از تابع `setup` تمامی سخت افزارها را `initialize` می کنیم. سپس در تابع `setup` ادامه `initialize` کردن را انجام می دهیم. منطق کد به صورت زیر است:

- در تابع `loop` که پی در پی انجام اجرا می شود ما کرکتر ورودی را می گیریم.
- ما یک `password string` یک `input string` داریم.
- تابع `validkey` برای این است که متوجه شویم آیا کاربر عددی بین ۱ تا ۹ زده است یا نه.
- حال که کرکتر را گرفتیم و درستی آنرا بررسی کردیم می آییم و پسورد ورودی گرفته شده را ورودی می گیریم. اگر کاراکتری بین ۱ تا ۹ وارد نشده بود. با توجه به این که کرکتر برای ریست کردن یا تایید کردن رمز وارد شده است کار مورد نظر را انجام می دهیم.
- دقت کنید که در کل فرایند ما `lcd` را آپدیت می کنیم.

۲-۳ نتیجه نهایی

همانطور که دیدیم برای انجام این پروژه باید برد اصلی و مابقی سخت افزارها را به همدیگر به صورت درست وصل می کردیم و برای انجام تغییرات روی آنها از واسطه کدی که زبان ++C نوشته شده بود استفاده کردیم. شاید ما در این پروژه وسیله بسیار پر کاربرد و بزرگی درست نکردیم ولی با برد Arduino می توان پروژه بسیار سنگین تر و سخت تری پیاده سازی کرد مانند ربات هایی که کارهای متفاوتی انجام می دهند.