

به نام خدا

گزارش طراحی و پیاده سازی سخت افزار

نام پروژه : سیستم سخت افزار وندینگ ماشین

نام درس : سیستم های نهفته و بی درنگ

نام استاد : دکتر محمدزاده

اعضای گروه:

- امیررضا سلاطین : طراحی و پیاده سازی سخت افزار و اتصال سخت افزار به نرم افزار رابط کاربری
- صدرا غفران : طراحی و پیاده سازی نرم افزار رابط کاربری
- حسین صمدی : طراحی های سه بعدی و چاپ سه بعدی

تاریخ تحویل : 1404/4/7

- چکیده پروژه 2
- مقدمه 2
- ابزار و تجهیزات مورد استفاده 2
- طراحی سخت افزار 3
- برنامه نویسی و پیاده سازی نرم افزار 4
- چالش ها 6
- آینده پروژه 7
- نتیجه گیری 7

چکیده پروژه

هدف از انجام پروژه سخت افزاری گفته شده این است که بتوانیم کارهایی که دستگاه وندینگ ماشین انجام می‌دهد را پیاده‌سازی کنیم. برای این کار از قطعاتی مانند سروو موتور و پمپ‌های آب استفاده کردیم تا کارهایی که دستگاه‌های وندینگ ماشین انجام می‌دهند را با هزینه خیلی کم و با استفاده از روش جدید پیاده‌سازی کنیم و در نتیجه همان عملکرد را با هزینه پایین‌تر پیاده‌سازی کردیم که علاوه بر آماده کردن نوشیدنی‌های عادی با استفاده از اطلاعات گرفته شده از مشتری نوشیدنی مخصوص هر فرد را آماده می‌کند.

مقدمه

پروژه سخت افزار، یک سیستم وندینگ ماشین است که وظیفه آماده سازی و تحویل نوشیدنی بر اساس انتخاب مشتری و همچنین قابلیت شخصی سازی نوشیدنی برای مشتری بر اساس سن و قد و معیارهای دیگر است. این پروژه برای آسان سازی سرو و آماده سازی نوشیدنی بدون دخالت دست انسان ساخته می‌شود. هدف ما از ساخت این پروژه، ایجاد دستگاه با هزینه کم و بهینه است.

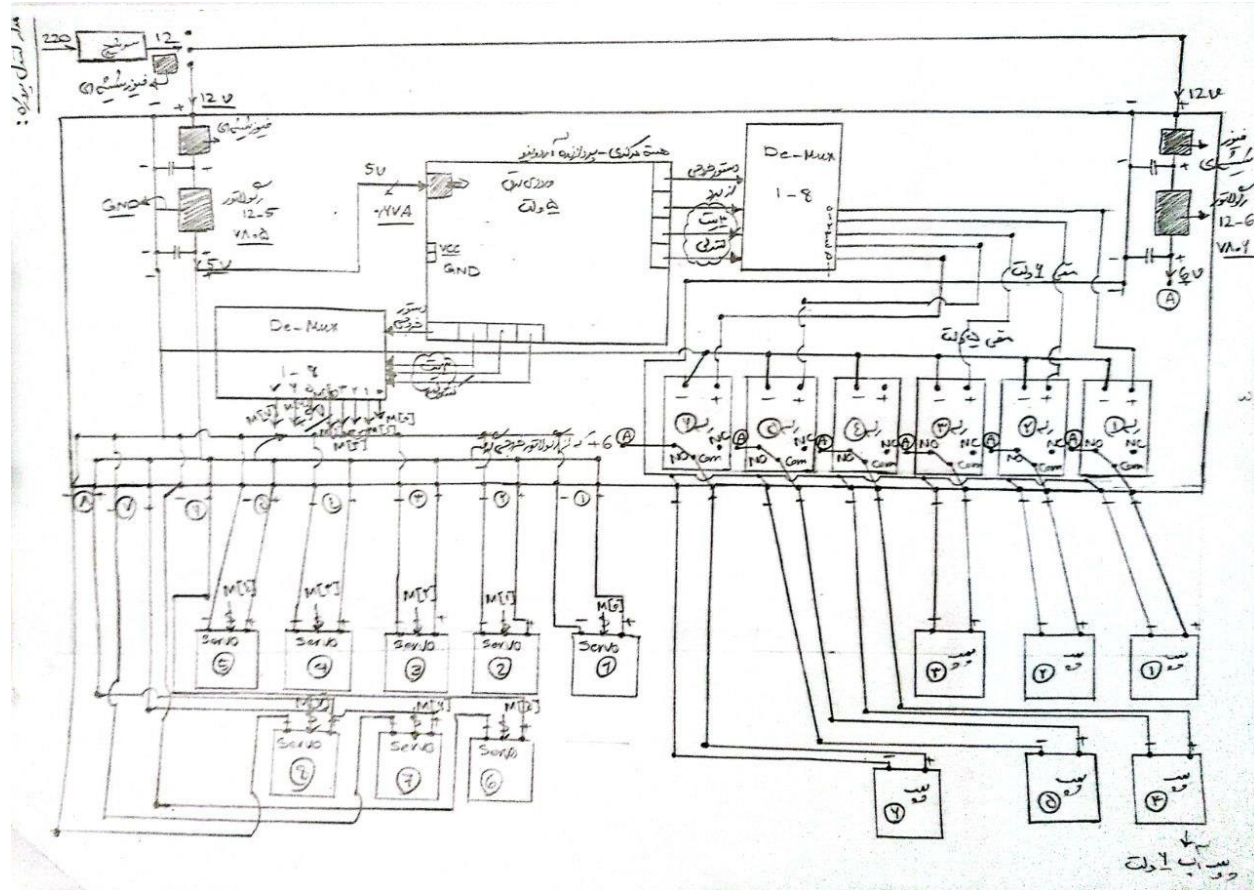
ابزار و تجهیزات

ابزار های استفاده شده در این پروژه:

- برد آردوینو UNO R3 برای هسته مرکزی و کنترل اجزای مختلف سخت افزار
- دی مالتی پلکسر برای کاهش تعداد استفاده از پین های خود برد
- ماژول رله 5 ولتی
- سروو موتور MG995
- پمپ آب 5 ولتی
- سوییچینگ 12 ولت 5 آمپر
- رگولاتور 7805
- فیوز شیشه ای با جریان های مختلف

طراحی سخت افزار

شماتیک سخت افزار دستگاه و سیستم مورد نظر:



در مدار نشان داده شده اتصالات و عملکرد هر قسمت را در ادامه توضیح می دهیم:

- هسته مرکزی برد آردوینو است که عملکرد کل سیستم و مدیریت دستگاه های مختلف را بر عهده دارد و خروجی برد بر اساس پین های خود برد است.
- دی مالتی پلکسر ها برای کاهش استفاده از پین های برد است (چون تعداد پین های برد کم بود و در نتیجه با هزینه پایین طراحی و ساخت انجام می شود). نحوه عملکرد به گونه ای است که در هر کدام 3 پین برای انتخاب اینکه کدام دستگاه کار کند و یک پین برای ارسال دستور استفاده می شود که برای دستگاه های سروو و پمپ ها استفاده می شود که اگر از دی مالتی پلکسر استفاده نمی کردیم باید به ازای هر پمپ و سروو یک پین اختصاص میدادیم.
- سروو موتور ها برای باز و بسته کردن دهانه مخزن های پمپ استفاده می شوند و دستور آنها از طریق برد ارسال می شود که دارای کتابخانه ی مخصوص برای استفاده هستند.

- مازول های رله 5 ولتی برای مدیریت استفاده از پمپ ها قرار داده می شوند یعنی برای روشن و خاموش کردن پمپ های آب استفاده می شوند (چون پمپ ها با اتصال برق و قطع اتصال خاموش و روشن می شوند و برای مدیریت این خاموش و روشن شدن از رله استفاده می کنیم که خود مازول رله دارای پین هایی برای اتصال و مدیریت از سمت برد را دارند که به اتصال ولتاژ 5 ولتی تحریک شده و اتصال را برقرار می کنند).
- پمپ های آب برای آوردن آب از مخزن به لیوان های مورد نظر هستند که اسم آنها پمپ های 6 ولتی است اما با ولتاژ 5 ولتی هم روشن می شوند.
- رگولاتور 7805 برای تبدیل ولتاژ 12 ولتی به 5 ولتی به منظور روشن کردن سروو موتور ها است.
- رگولاتور 7806 برای تبدیل ولتاژ 12 ولتی به 6 ولتی به منظور روشن کردن پمپ ها است.
- سوئیچینگ برای تبدیل برق شهری (220 ولت) به ولتاژ مورد استفاده یعنی 12 ولت است.

برنامه نویسی و پیاده سازی

```
#include <Servo.h>
```

```
Servo my_servo;
```

```
String command = "";
```

```
void setup() {
```

```
pinMode(6, OUTPUT); //Enable of Servo motors
```

```
pinMode(7, OUTPUT); //Enable of water pumps
```

- قسمت <Servo.h> include: برای اضافه کردن کتابخانه برای استفاده از سروو موتور ها
- Servo my_servo: در اینجا یک Object از نوع سروو ایجاد میکنیم که بعدا آنرا به یک پین که متصل به سروو است وصل کنیم.
- Void setup(): این بلاک در کل دستگاه فقط یکبار اجرا میشه و وظیفه آن مشخص کردن اینکه هر پین چه ویژگی دارد یعنی خروجی است یا ورودی یا از نوع سروو و....
- pinMode(pin number, OUTPUT/INPUT): از این دستور برای مشخص کردن نوع پین استفاده می کنیم که خروجی است یا ورودی و به عنوان ورودی اول شماره پین را قرار میدهم بعد نوع پین را با حروف بزرگ وارد می کنیم.
- String command: برای ورودی دستور از UI یک متغیر ایجاد کردیم.

```
my_servo.attach(11);
```

این دستور برای مشخص کردن یک پین به عنوان پین مخصوص سروو موتور است که به یک Object ساخته شده از جنس سروو این پین را اختصاص میدیم.

```
Serial.begin(9600);
Serial.println("Enter number_button (1-6):");
```

- `Serial.begin(9600)`: این دستور برای ایجاد راه ارتباطی بین سخت افزار آردوینو و UI است و 9600 نشان دهنده سرعت ارتباط داده است که BaudRate آن 9600 است.
- `Serial.println()`: برای ارسال دستور به UI استفاده میشه که دستور داخل () را به UI میفرسته.

```
int number_button = 0;
```

```
void loop() {
  while (Serial.available()) {
    char c = Serial.read();
    if (c == '\n') {
      processCommand(command);
      command = ""; //reset command after process finished
    } else {
      command += c;
    }
  }
}
```

- `int number_button`: این متغیر برای تعیین دکمه زده شده استفاده می شود که در ادامه نحوه مقدار دهی آنرا توضیح میدهم.
- `void loop()`: دستور های داخل این بلاک تا آخر در برنامه پشت سر هم اجرا میشوند و یک حلقه بی نهایت است.
- `Serial.available()`: این برای بررسی این است که آیا ارتباط بین UI و سخت افزار ایجاد شده است یا نه.
- `Serial.read()`: این دستور برای خواندن دستور ارسال شده از طرف UI است که حرف به حرف در C ذخیره میشود و وقتی به `\n` رسید یعنی پایان دستور و به بخش پردازش دستور میرود.
- `processCommand()`: این تابع که در ادامه تعریف آنرا میبینیم با گرفتن دستور ارسال شده از UI سخت افزار را شروع به کار میکند تا دستورات انجام شوند.

```
void processCommand(String cmd)
{
  cmd.trim();
  if(cmd[0] == 'B')
    number_button = cmd[1] - '0';
  //give number of button from UI and put in number_button//
  if (number_button >= 1 && number_button <= 5) {
    //Turn on enables
    digitalWrite(6, 0);
  }
}
```

`digitalWrite(7, 0);`

- `cmd.trim()`: این دستور خطوط اضافه در پشت یا جلوی متن را حذف میکند تا دستور دقیق بدست آید.
- `number_button = cmd[1] - '0'`: این یک دستور خیلی خوب در برنامه نویسی است تا از طریق جدول ASCII عدد به صورت string را به int در اعداد تک رقمی تبدیل کنیم.
- شرط بررسی شده در if دوم برای بررسی این است که دکمه 6 زده شده یا نه چون دکمه 6 با بقیه متفاوت است (در تعداد سروو های استفاده شده)
- `digitalWrite(pin number, (0,1)/(LOW,HIGH))`: این دستور برای روشن/خاموش کردن یک پین که مشخص میکنیم استفاده میشه (منظور از روشن کردن منظور رساندن برق به پین تعیین شده). برای روشن کردن میتوانیم هم از 1 HIGH و برای خاموش کردن از 0 LOW استفاده میکنیم.
- 2 دستور آخر `digitalWrite()` برای روشن کردن Enable های دی مالتی پلکسر ها هستند.

```
digitalWrite(3, bitRead(number_button - 1, 0)); //S0 = x
digitalWrite(4, bitRead(number_button - 1, 1)); //S1 = x
digitalWrite(5, bitRead(number_button - 1, 2)); //S2 = x
//write output 1//
my_servo.write(90);
delay(2700);
my_servo.write(0);
////////////////////////
delay(2000);
```

- `bitRead(number, index)`: این تابع عدد number را به باینری تبدیل میکند و index انتخاب شده از این عدد باینری را به عنوان خروجی میدهد.
- `my_servo.write()`: این دستور برای چرخاندن سروو موتور استفاده می شود که پره سروو را به زاویه ورودی که به صورت زاویه است تنظیم می کند یعنی اگر ورودی 180 باشد به اندازه 180 درجه از حالت اصلی و تنظیم شده خود میچرخد.
- `delay(time)`: این تابع برای ایجاد تاخیر استفاده میشود و به اندازه عدد ورودی که میلی ثانیه است، تاخیر ایجاد میکند (در آنجایی که این دستور آمده است).

چالش ها

یک چالش مهم در طراحی این دستگاه ناپایدار شدن مدار در هنگام استفاده از پمپ ها بود که با استفاده از یک خازن این ناپایداری رفع می شود.

هنگام اتصال برق انرژی داخل خازن ذخیره می‌شود و در صورت نوسان تامین انرژی می‌کند.

آینده این پروژه

- در آینده قصد داریم با افزودن سنسور ها خالی یا پر بودن مخازن را بررسی کنیم و در صورت خالی بودن به پشتیبانی اطلاع دهیم.
- افزودن قابلیت صوتی به دستگاه برای گرفتن سفارش از طریق صوت به کمک پردازش زبان طبیعی
- افزودن نوشیدنی های گرم در کنار آماده کردن نوشیدنی های سرد

نتیجه گیری

در این پروژه توانستیم با طراحی و پیاده سازی یک سیستم سخت افزاری ساده و مقرون به صرفه، عملکرد اولیه یک دستگاه وندینگ ماشین را شبیه سازی کنیم. استفاده از قطعاتی مانند آردوینو، سرووموتور، و پمپ های آب، به ما این امکان را داد که با هزینه ای کم، فرآیند آماده سازی نوشیدنی را به صورت خودکار پیاده سازی کنیم. همچنین با اتصال سخت افزار به رابط کاربری، توانستیم ارتباطی مؤثر بین کاربر و دستگاه برقرار کنیم. این پروژه پایه ای برای توسعه سیستم های پیچیده تر در آینده است که می توانند هوشمندتر و کاربردی تر باشند.

پایان