

به نام خدا



دانشکده مهندسی برق

گزارش کار پروژه ابزار دقیق

مزرعه هوشمند

استاد : ایمان شریفی

اسامی اعضا :

محمد برابادی - مصطفی حمیدی فرد - سجاد قدیری - محمد مهدی مالوردی

مقدمه

یقیناً کشاورزی یکی از مهم ترین مشاغل دنیا به حساب می آید. برای بهره وری نیاز به صبر و دقت کافی در طول زمان است. به مرور زمان، دستگاه هایی اختراع شدند که انجام خیلی از کارها را برای کشاورز آسان می کرد. این شروع شکل گیری چیزی به نام کشاورزی مدرن است. به عنوان مثال دستگاهی مانند شیردوش، تراکتورهای بدون سرنشین و ... از جمله مواردی هستند که کشاورزی را به میزان زیادی بهبود بخشیدند. اما این پایان ماجرا نیست، چرا که هم چنان مواردی هست که کشاورز برای در نظر گرفتن آن ها باید هزینه مالی و زمانی زیادی صرف کند. پس زمان آن رسیده است که روش های بهتر و تازه برای حل مسئله پیدا کنیم.

امروزه با پیشرفت تکنولوژی، مفاهیمی چون اینترنت اشیا بسیار فراگیر شده و به حوزه های متنوعی راه یافته اند و باعث نگرشی جدید در این حوزه ها شده اند. یکی از این حوزه ها کشاورزی هوشمند و مزرعه هوشمند (Smart Farming) است. در کشاورزی هوشمند، هدف استفاده از دستگاه هایی است که از سنسورهای مختلف تشکیل شده اند. این دستگاه ها با استفاده از این سنسورها قادر به جمع آوری یک سری داده و ارسال آن ها به سرور جهت تجزیه و تحلیل هستند. بعد از تجزیه و تحلیل، دستور لازم برای عملکرد مناسب ارسال خواهد شد. بنابراین با استفاده از این فناوری هوشمند سازی، تنها نیاز است که کشاورز نحوه استفاده از این دستگاه ها را بداند.

مزرعه هوشمند با معرفی تکنولوژی های موثر بر کشاورزی با هدف کاهش هزینه، کارایی بهتر در کشاورزی و محصولات باکیفیت، آینده ای روشن تر را نوید می دهد.

در کشاورزی هوشمند، جهت داشتن بهترین بازده، از سنسورهای مختلفی استفاده می شود که می توان به عنوان مثال به موارد زیر اشاره کرد:

- سنسور هوشمند خاک: با استفاده از این سنسورها می توان عواملی مانند رطوبت، درجه حرارت خاک و ... را اندازه گرفت.

- سنسور هوشمند هوا: این سنسورها جهت اندازه گیری دما، پیش بینی شرایط آب و هوایی و... استفاده می شوند.

- سنسور هوشمند آب: همانطور که می دانید PH آب در سلامت گیاهان تاثیر مستقیمی دارد. یکی از موارد استفاده از این سنسورها همین موضوع است.

از مزیت های مزرعه هوشمند می توان به موارد زیر اشاره کرد :

1 - کاهش استفاده از سموم دفع آفات، کودها و آب

2 - بهره وری بالای محصول در کشاورزی هوشمند

3- سازگاری بیشتر با محیط زیست

.....

سنسور ها :

به دلیل نبود همه سنسور ها در پروتئوس و ایجاد امکان پردازش به صورت بلادرنگ، داده های متناسب با هر سنسور در متلب تولید می کنیم و نیازی به راه اندازی الکترونیکی این سنسور ها نداریم، ولی برای کامل تر بودن گزارش و پروژه، در هر قسمت مدار راه اندازی و کد راه اندازی آردوینو قرار داده شده است.

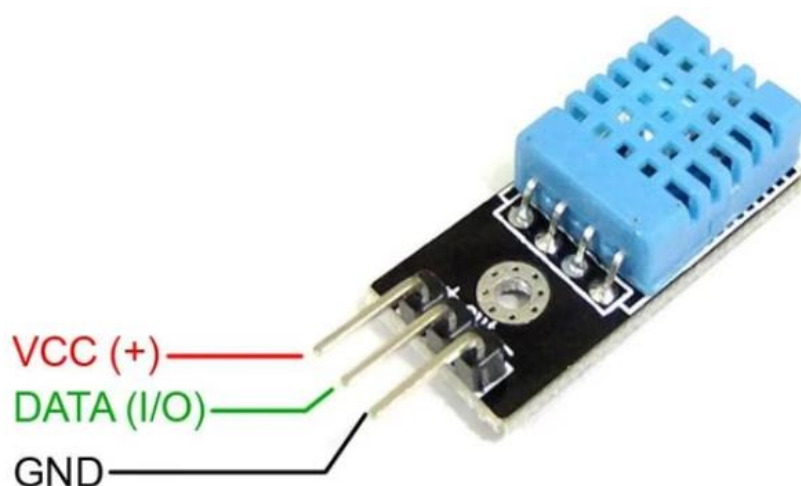
LM35DZ :

یک سنسور دما است که سیگنال آنالوگ متناسبی را با توجه به دمای لحظه ای تولید می کند . پوششی مناسب در برابر گرم شدن دارد و برخلاف ترمیستور نیازی به کالیبراسیون خارجی ندارد . این سنسور می تواند دما را در بازه $[150^{\circ}\text{C}, -55^{\circ}\text{C}]$ اندازه گیری کند . اگر سطح دما و رطوبت محیط بهینه و مناسب باشد (مثلا دمای محیط) ، دقت بالایی دارد. قیمت این سنسور نیز بسیار معقول و به صرفه است .

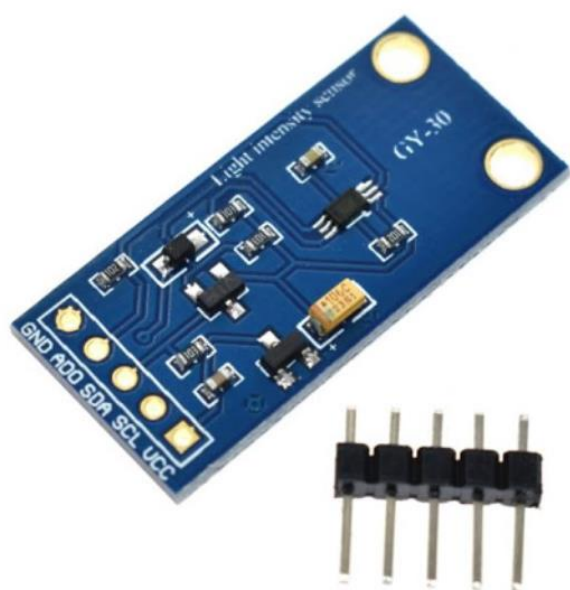
3. اندازه کوچک و قیمت پایین و ارزان

....

```
110 float dht_temp() {  
111  
112  
113 int chk = DHT.read11(DHT11_PIN);  
114 float temp = DHT.temperature;  
115 delay(1000);  
116 return temp;  
117  
118 }  
119  
120  
121 float dht_hum() {  
122  
123 int chk = DHT.read11(DHT11_PIN);  
124 float hum = DHT.humidity;  
125 delay(1000);  
126 return hum;  
127  
128 }
```

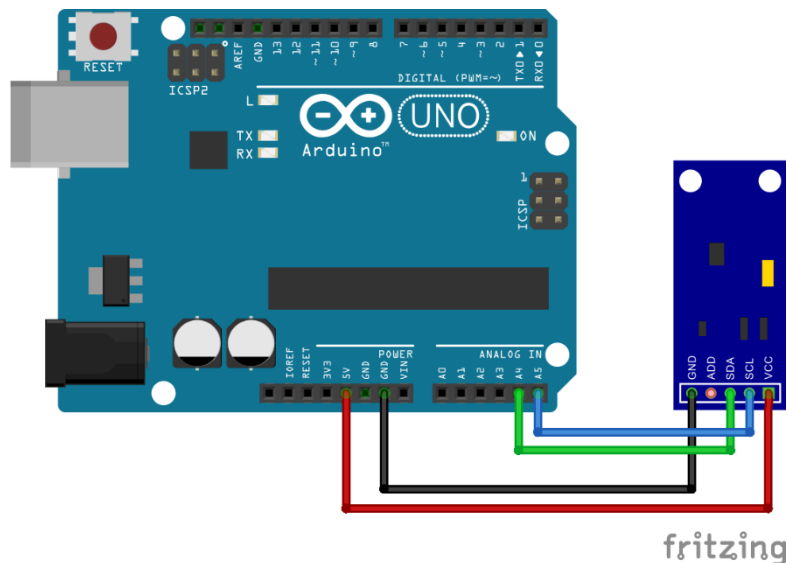


: BH1750FVI



ماژول BH1750 یک برد مجهز به سنسور حساس به شدت نور است که دارای یک مبدل A2D شانزده بیتی می باشد و مستقیماً سیگنال دیجیتال در خروجی ارسال می کند . این ماژول برای تشخیص میزان نور محیط با دقت و رزولوشن بالا مناسب بوده و به راحتی با آردوینو قابل راه

اندازی می باشد. محدوده ولتاژی آن متغیر است و در بازه $[3.3^v \text{ or } 4.5^v, 6^v]$ عمل میکند .



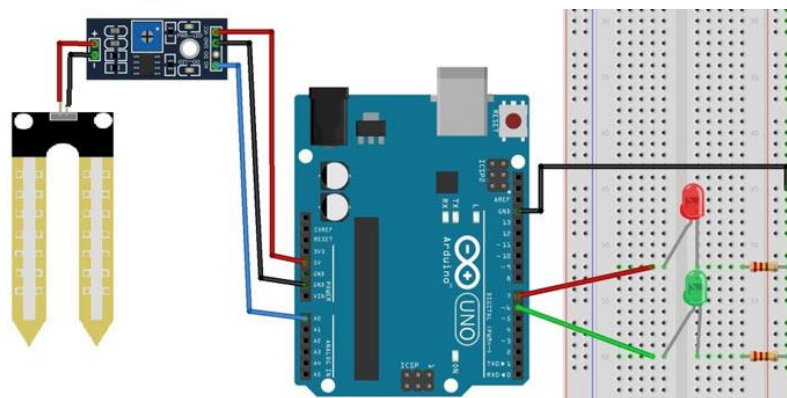
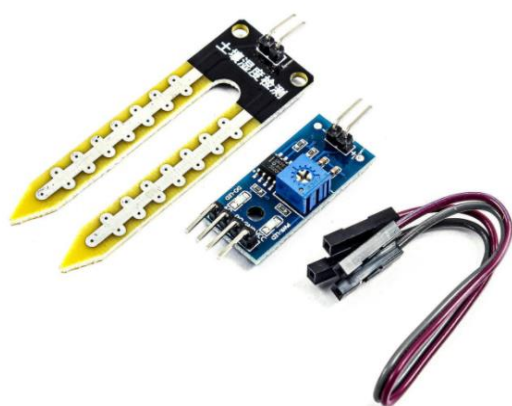
```
142 float light_meter() {  
143  
144     // begin returns a boolean that can be used to detect setup problems.  
145     if (lightMeter.begin(BH1750::CONTINUOUS_HIGH_RES_MODE)) {  
146         Serial.println(F("BH1750 Advanced begin"));  
147     }  
148     else {  
149         Serial.println(F("Error initialising BH1750"));  
150     }  
151 }  
152  
153  
154 float lux = lightMeter.readLightLevel();  
155 //Serial.print("Light: ");  
156 //Serial.print(lux);  
157 //Serial.println(" lx");  
158 delay(1000);  
159  
160 return lux;  
161 }  
162 }
```

YL-69:

این سنسور برای اندازه گیری رطوبت و یا آب موجود در خاک استفاده می گردد . این سنسور در اصل میزان تشنگی گیاه شما را مشخص می نماید .

خروجی این سنسور به هر دو صورت آنالوگ و دیجیتال قابل استفاده بوده و محدوده ولتاژ عملیاتی آن در بازه $[3.3^V, 5^V]$ می باشد . از مزیت های آن می توان به اندازه کوچک ، قیمت پایین ، نصب آسان ، قابلیت تنظیم

حساسیت از طریق پتانسیومتر موجود روی برد و ... اشاره کرد . همچنین برای افزایش طول عمر آن باید از اتصال دائم به تغذیه خودداری و تنها در زمان خواندن رطوبت آن را روشن کنیم .



```
130 float yl_hum() {  
131  
132   float hum = analogRead(A2);  
133   float hum2;  
134   hum = map(hum, 0, 1023, 100, 0);  
135   hum2 = map(hum, 0, 100, 0, 255);  
136   analogWrite(greenLED, hum2);  
137   delay(1000);  
138  
139   return hum;  
140 }
```

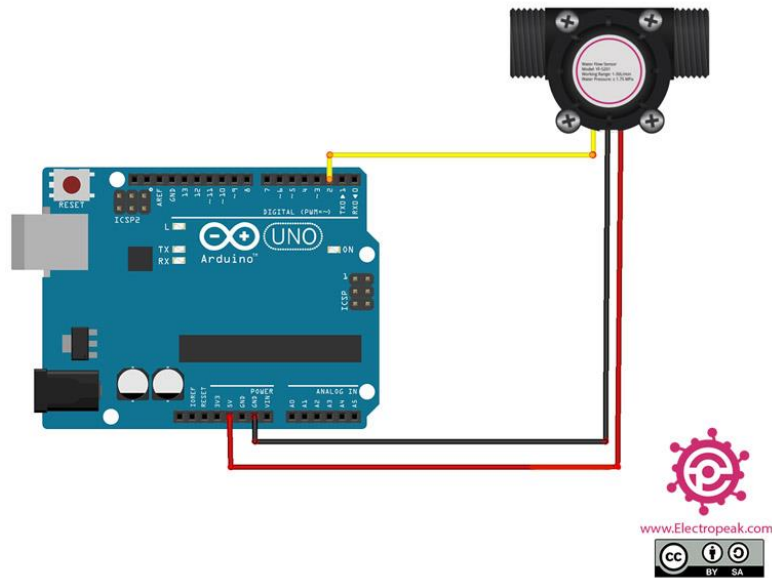
YF-S201:

YF-S201 سنسوری برای محاسبه میزان جریان عبوری آب است. اجزای اصلی تشکیل دهنده آن شامل یک بدنه پلاستیکی برای عبور جریان آب، یک روتور در تماس با آب و یک سنسور اثر هال می باشد. عملکرد این ماژول بر اساس اثر مغناطیسی هال است به طوری که این سنسور با هر بار دور کامل روتور توسط جریان آب یک پالس الکتریکی تولید می کند. با شمارش این پالس های تولیدی می توان میزان حجم آب عبوری از سنسور را محاسبه کرد.

رنج اندازه گیری جریان آب توسط این سنسور 1-30 L/Min و فشار مجاز آن تا 1.75 MPa می باشد. هر پالس الکتریکی تقریباً معادل عبور 2.25 میلی لیتر آب است.

POWER
GND
OUT





```

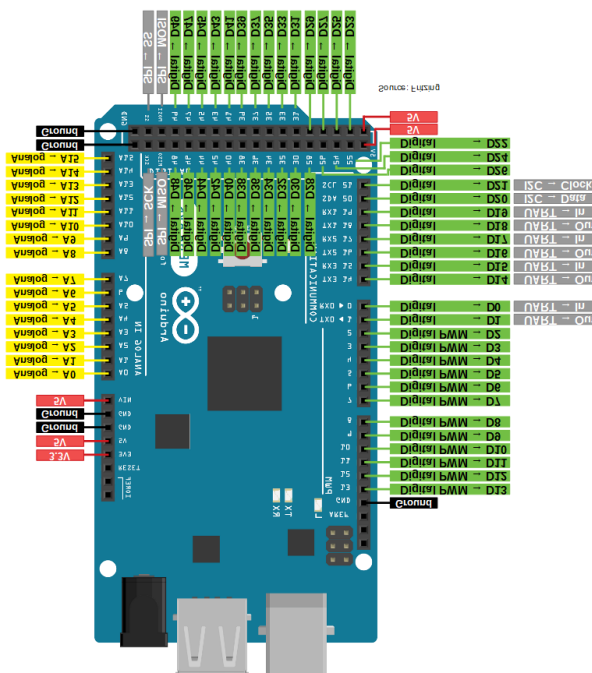
75 | attachInterrupt(0, pulse, RISING); // Setup Interrupt
    |
165| float flow_meter() {
166 |     double flow; //Liters of passing water volume
167 |     flow = .00225 * pulse_freq;
168 |     Serial.print(flow, DEC);
169 |     Serial.println("L");
170 |     delay(500);
171 |
172 |     return flow;
173 | }
174 |
175 | void pulse () // Interrupt function
176 |
177 | {
178 |     pulse_freq++;
179 | }

```

پرسش 1: پین های تعبیه شده در برد آردوینو و برد رسیبری پای به منظور برقراری ارتباط سریال به صورت واقعی کدام ها هستند؟

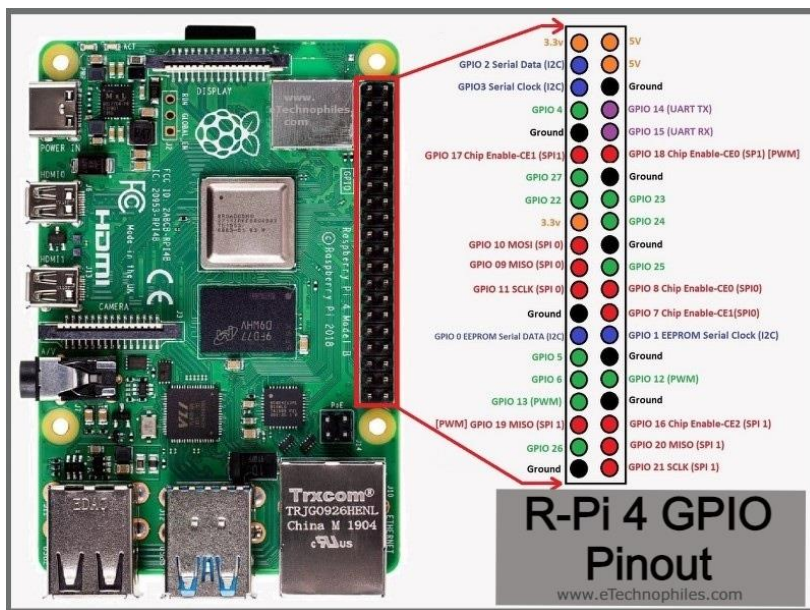
چون داده های سنسور های ما به صورت سریال از متلب می آیند و ما بیشتر از یک جفت پایه سریال نیاز داشتیم، همچنین چون روش های سریال مجازی و ارتباطات سریالی از قبیل i2c, SPI جوابگو نیاز ما نبودند، از آردوینو مگا برای پیاده سازی مجازی پروژه و از آردوینو اونو برای پیاده سازی واقعی پروژه استفاده می کنیم. در آردوینو اونو پایه های 0 و 1 به ترتیب rx و tx آردوینو هستند. در آردوینو مگا چهار جفت پایه ارتباط سریال واقعی داریم. پایه های

آردوینو مگا هستند.



سایر پروتکل های ارتباط سریال نظیر SPI,i2c و ... نیز در این عکس مشخص شده اند.

در رزبری پای نیز پایه های 15 و 14 به ترتیب rx و tx هستند. سایر پایه های ارتباط سریال این برد نیز در عکس زیر نشان داده شده است.



پرسش 2: در مورد رسیبری پای واقعی برای دریافت اطلاعات از چه آدرس پورته در کد پایتون فوق باید استفاده نماییم ؟

در رسیبری پای واقعی برای دریافت اطلاعات از آدرس `ttyAMA0` استفاده می کنیم .

(در حالت مجازی از آدرس `ttyS2` استفاده شده است)

پرسش 3: دلیل تخصیص آدرس `0.0.0.0` به `host` را بیان کنید.

It means “ Listen to any external address “

یعنی هرکسی می تواند به `host` ما دسترسی داشته باشد .

مراحل و جزئیات پروژه :

- 1 - تولید داده های سنسور ها در محیط سیمولینک متلب به صورت Real Time
- 2- پردازش داده های دریافتی از متلب و وب سرور بر روی آردوینو در محیط پروتئوس و استفاده از actuators
- 3 - دریافت داده های متلب در آردوینو و ارسال آنها برای رسیبری پای در ویرچوال باکس
- 4 - دریافت داده ها در رسیبری پای و انتقال آنها بر روی Local Host با استفاده از json
- 5- طراحی PCB برای ایجاد امکان انجام و اجرای پروژه به صورت واقعی و تکمیل پروژه

1 - تولید داده های سنسور ها در محیط سیمولینک متلب به صورت Real Time :

در ابتدا با توجه به محدوده کاری سنسور های انتخاب شده، کار را در پیش گرفتیم .
با تعریف شرایط مطلوب اولیه برای هر سنسور و با قرار دادن ورودی های متغیر با زمان همچون sine wave ، شروع به طراحی سیستم کنترل کننده ای کردیم تا با تغییر تصادفی شرایط بیرونی و اثر گذاشتن عوامل مرتبط بر هر سنسور ، بتوانیم پارامتر های مزرعه هوشمند خود را بر روی اعداد مطلوب اولیه، ثابت نگه داریم . (چون قادر به دریافت دیتا از طریق serial receive به صورت real time نبودیم، متأسفانه نتوانستیم قسمت کنترل را در متلب انجام دهیم).

***در انتهای کار ، کاربر پارامتر مرجع (reference) مطلوب خود را از طریق سایت تعیین می کند .**

برای مثال برای پارامتر دما ، یک ورودی سینوسی با دامنه 5 که حول دمای 25°C نوسان می کند قرار دادیم . سپس کنترل کننده دما را طوری طراحی کردیم که پس از زمان معینی شروع به اصلاح ورودی کرده و آن را بر روی دما مطلوب اولیه با نوسانات بسیار کم نگه دارد .

در انتها با استفاده از virtual serial port driver خروجی را به نرم افزار پروتئوس انتقال دادیم و صحت انتقال داده ها را با استفاده از virtual monitor در پروتئوس بررسی کردیم.

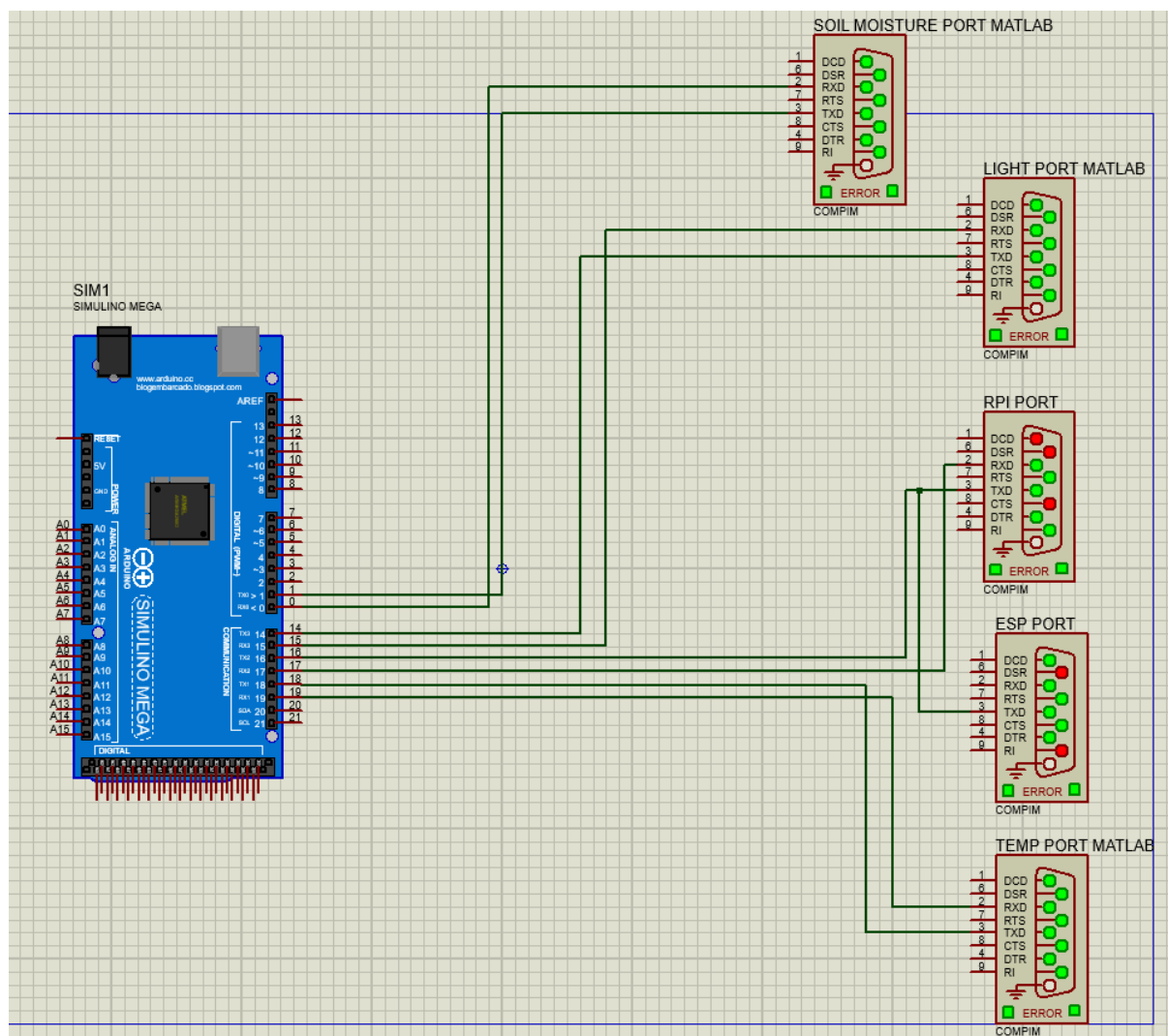
به دلیل کمبود جفت پایه های سریال و عدم توانایی قابلیت هایی نظیر سریال مجازی و ارتباط سریال i2c, SPI در پروتئوس کرک شده، از آردوینو مگا استفاده کردیم که چهار جفت پایه سریال دارد و در نتیجه داده های سه سنسور را می توانیم از متلب دریافت کرده و در پروتئوس روی آن ها پردازش انجام بدهیم و به وسیله سریال شماره دو برای رزپیری پای بفرستیم.

نکات مهم :

* برای گسسته سازی از fixed step با مقدار 0.05 استفاده کردیم . قابل توجه است که با افزایش این مقدار در واقع در فرآیند گسسته سازی، تاخیر افزایش پیدا می کند و ممکن است اصلا منجر به همگرایی نشود.

2- پردازش داده های دریافتی از متلب و وب سرور بر روی آردوینو در محیط پروتئوس و استفاده از actuators

3 - دریافت داده های متلب در آردوینو و ارسال آنها برای رسپیری پای در ویرچوال باکس



توضیحات:

- از آردوینو مگا و پایه های سریال آن استفاده کردیم.
- یک COMPIM برای ارتباط با رزبری و یک COMPIM برای ارسال داده ها به سمت ESP قرار داده شده است.
- COMPIM های دیگر صرفا برای ارتباط با سیمولینک متلب استفاده شده است.
- به دلیل کرک بودن پروتئوس ها در این پروژه، اگر از قطعات دیگر مانند LED یا موتور DC در پروتئوس استفاده می کردیم، برنامه پروتئوس پس از ده الی بیست ثانیه به صورت خودکار بسته می شد؛ به همین جهت صرفا کد قسمت LED و موتور را در کد آردوینو نوشتیم و از گذاشتن قطعات مربوطه در پروتئوس صرف نظر کردیم. برای

مشاهده و تایید تست کد ها به تغییرات رنگ پایه آردوینو هنگام ران بودن پروتئوس
توجه شود.

4 – دریافت داده ها در رسپیری پای و انتقال آنها با استفاده از json و نمایش آنها بر روی Local Host با استفاده از HTML و CSS :

در ابتدا با استفاده از Flask و route که در کلاس آموزش داده شد، یک سرور ایجاد کردیم.

سپس با استفاده از HTML یک [صفحه ورودی](#) به شکل زیر طراحی کردیم.

Smart Farming

Mohammad Barabadi

Sajad Ghadiri

MohammadMahdi Malverdi

Mostafa Hamidifar

در این طراحی برای background صفحه از linear-gradient استفاده شده و بدون تکرار می باشد و با استفاده از div یک بلوک ساختم و با بزرگترین هدینگ یعنی h1 کلمه smart farming را قرار دادیم و همه چیز در مرکز (Center) قرار گرفته است .

همچنین نمایش اسامی را با انیمیشن (animation) انجام دادیم و با دستور forward برای animation-fill-mode اسامی را در انتهای کار ثابت در جای خود نگه می داریم و برای زیبایی بیشتر از فونت بهره بردیم .

در مرحله بعد یک صفحه برای Monitoring داده های سنسور ها طراحی کردیم. برای background از یک عکس مزرعه استفاده شده است که آن را در زیر مشاهده می کنید :



با استفاده از تگ `<table>` یک جدول ساختیم و با استفاده از attribute ها در CSS رنگ آن و شفافیت رنگ آن را تعیین کردیم . سپس پارامتر های مورد نظر برای سنسور هایمان را با استفاده از تگ `<th>` و `<td>` نوشتیم . حال برای نمایش داده ها به صورت پیوسته ، در تگ های `span` که برای هر پارامتر نوشته شده یک id قرار دادیم . همچنین یک کلید با استفاده از تگ `<button>` ایجاد کردیم تا کاربر بتواند مقدار دلخواه هر پارامتر را تعیین و به سیستم کنترلی ما در متلب یا پروتئوس ارسال کند .

```
<th class="satr">
  Humidity
</th>
<td class="satr"
  style="display: flex; justify-content: space-around; align-items: center; height: 100%;
  <div>
    <span id="humidity">-</span>
    <span>%</span>
  </div>
  <div>
    <div class="input">
      <input type="number" placeholder="Humid Ref" value="input" id="inputH">
      <button type="button" id="bhumidity">Send</button>
    </div>
  </div>
</td>
```

با دستور `display: flex` از چپ به راست می‌نویسیم و دستور `align-item : center` از بالا و پایین فاصله را یکسان می‌کند . همچنین با دستور `Justify-content` با تغییر اندازه مرورگر ، فاصله بین عناصر جدول به صورت `dynamic` تغییر می‌کند .
برای زیباسازی آن از دستور `border-collapse: collapse` استفاده کردیم .

برای ارسال داده از سمت سرور به رسیبری پای ، کد زیر برای هر 4 پارامتر نوشته شده است.

```
SCRIPT_ROOT = {{ request.script_root | tojson | safe }};
```

```
$(function fl() {  
    $('#btemperature').bind('click', function () {  
        if (document.getElementById('inputT').value == 0) {  
            alert("You Don't Put a Value")  
        }  
        else {  
            alert("You Change Reference Of Temp");  
            $.getJSON(SCRIPT_ROOT + '/_transfer', {  
                inputT: $('#input[id="inputT"]').val()  
            }, function (data) {  
                $('#inputT').text(data.myresult);  
            });  
            return false;  
        }  
    });  
});
```

در این کد ابتدا چک می‌شود که بعد از فشرده شدن دکمه `send` آیا مقدار ارسال شده صفر(خالی) است یا خیر . اگر صفر باشد پیام `you don't put a value` نمایش داده می‌شود و اگر مقدار ارسالی غیر صفر باشد ، پیامی برای حصول اطمینان از تغییر به کاربر نمایش داده می‌شود و سپس این داده را به رسیبری پای می‌فرستد .

5- طراحی PCB برای ایجاد امکان انجام و اجرای پروژه به صورت واقعی و تکمیل پروژه:

برای تکمیل پروژه، یک طرح اولیه از برد مدار چاپی که برای پیاده سازی واقعی این پروژه مورد نیاز است طراحی کردیم. بخش های شماتیک و پی سی بی آن به شکل زیر می باشد:

