



دانشگاه صنعتی شریف
دانشکده‌ی مهندسی کامپیوتر

عنوان:

پروژه گلدان هوشمند

نگارش:

امیر محمد ابوئی، بنیامین قاسمی نیا

استاد :

دکتر اجلالی

نیم سال اول ۱۴۰۱ - ۱۴۰۰

سلام

چکیده

با گسترش شبکه اینترنت در سراسر جهان و افزایش ظرفیت و عملکرد آن زندگی بشر در ابعاد مختلف دچار تغییر و بهبود شده است. هوشمندسازی سیستم‌های پیرامون از یک طرف و از طرفی دیگر وصل کردن سامانه‌ها به یک شبکه کلی از حوزه‌های جذابی و پرکاربری است که در اخیر بسیار مورد توجه قرار گرفته است. در واقع مسئله مورد نظر برگرفته از یک مشکل واقعی که نحوه آبیاری گیاهان با استفاده بهینه از منابع است. در این پروژه ما سعی کردیم که هوشمندسازی آبیاری گیاهان را انجام دهیم و سامانه‌ای طراحی کنیم که بر اساس شرایط محیط تصمیم لازم جهت آبیاری را بگیرد. علاوه بر هوشمندسازی سیستم آبیاری، کاربران می‌توانند به صورت از راه دور و اینترنتی نیز از وضعیت محیط باخبر شوند و در صورت نیاز کنترل سامانه را بر عهده بگیرند.

کلیدواژه‌ها: آبیاری گیاهان، اینترنت اشیاء، خانه هوشمند

فهرست مطالب

| | |
|----|---------------------------------------|
| ۱۱ | ۱ مقدمه |
| ۱۱ | ۱-۱ تعریف مسئله |
| ۱۲ | ۱-۱-۱ هوشمندسازی آبیاری گیاهان |
| ۱۲ | ۱-۱-۲ نظارت بر فرآیند آبیاری |
| ۱۲ | ۲-۱ اهمیت مسئله |
| ۱۳ | ۳-۱ راهکار پیشنهادی ما |
| ۱۴ | ۴-۱ ساختار گزارش |
| ۱۵ | ۲ نحوه پیکربندی سخت‌افزاری |
| ۱۵ | ۱-۲ بلوک دیاگرام سیستم |
| ۱۶ | ۲-۲ عملکرد و ارتباط کلی بخش‌های مختلف |
| ۱۶ | ۲-۲-۱ سنسور رطوبت سنج |
| ۱۶ | ۲-۲-۲ ماورای صوت |
| ۱۶ | ۳-۲ ماژول شیر برقی |
| ۱۷ | ۴-۲ ماژول رله |
| ۱۷ | ۲-۴-۱ برد Arduino D1-Rev1 |
| ۱۷ | ۲-۴-۲ برد Arduino Uno |

| | |
|----|-----------------------------------|
| ۱۸ | ۳ سخت افزار مورد نیاز |
| ۱۸ | ۱-۳ آردوینو UNO |
| ۱۸ | ۱-۱-۳ امکانات و ویژگی ها |
| ۲۱ | ۲-۳ ماژول وای فای ESP8266EX |
| ۲۱ | ۱-۲-۳ امکانات کلیدی |
| ۲۱ | ۲-۲-۳ ویژگی های جزئی |
| ۲۳ | ۳-۳ آردوینو D1 نسخه Rev1 |
| ۲۴ | ۴-۳ سنسور رطوبت سنج YL-69 |
| ۲۴ | ۱-۴-۳ ویژگی ها و امکانات کلیدی |
| ۲۵ | ۲-۴-۳ عملکرد کلی |
| ۲۶ | ۵-۳ شیر برقی |
| ۲۶ | ۱-۵-۳ ویژگی های کلی |
| ۲۶ | ۲-۵-۳ نحوه عملکرد |
| ۲۷ | ۶-۳ رله |
| ۲۷ | ۱-۶-۳ ویژگی ها |
| ۲۸ | ۷-۳ ماژول ماورای صوت |
| ۲۸ | ۱-۷-۳ ویژگی های کلی و نحوه عملکرد |
| ۲۹ | ۲-۷-۳ نحوه استفاده |
| ۳۰ | ۸-۳ آداپتور |
| ۳۰ | ۱-۸-۳ ویژگی های کلی |
| ۳۱ | ۴ رابط کاربری |
| ۳۱ | ۱-۴ نیازمندی ها |

| | |
|----|--|
| ۳۲ | ۲-۴ تنظیمات اولیه Blynk |
| ۳۵ | ۳-۴ نحوه کارکرد |
| ۳۵ | ۱-۳-۴ اعلان روشن شدن سامانه |
| ۳۶ | ۲-۳-۴ بازکردن شیربرقی در حالت دستی |
| ۳۶ | ۳-۳-۴ اختار عدم تنظیم شیربرقی در حالت خودکار |
| ۳۷ | ۴-۳-۴ حالت‌های دیگر |

| | |
|----|----------------------------------|
| ۳۸ | ۵ نحوه پیاده‌سازی فیزیکی |
| ۳۸ | ۱-۵ اتصال ماژول رطوبت سنج |
| ۳۹ | ۲-۵ اتصال ماژول ماورای صوت |
| ۴۰ | ۳-۵ اتصال ماژول شیربرقی |
| ۴۲ | ۴-۵ اتصال رله |
| ۴۳ | ۵-۵ اتصال دو برد اصلی به یک‌دیگر |
| ۴۴ | ۶-۵ جمع‌بندی |

| | |
|----|-------------------------------|
| ۴۵ | ۶ نحوه پیاده‌سازی نرم‌افزاری |
| ۴۵ | ۱-۶ کتابخانه‌های مورد نیاز |
| ۴۶ | ۲-۶ تنظیمات اولیه و متغیرها |
| ۴۸ | ۳-۶ تعیین فاصله‌ی سطح آب |
| ۴۹ | ۴-۶ نحوه کار سیستم در هر گردش |
| ۵۰ | ۵-۶ ارتباط با مشتری |
| ۵۰ | ۱-۵-۶ فعال‌سازی سیستم |
| ۵۰ | ۲-۵-۶ دریافت حالت سیستم |
| ۵۱ | ۳-۵-۶ دریافت حالت شیر |

| | |
|----|---------------------------------|
| ۵۲ | ۴-۵-۶ دریافت مقدار آستان رطوبت |
| ۵۲ | ۵-۵-۶ بروزرسانی وضعیت مشتری |
| ۵۳ | ۶-۶ چاپ متغیرها در ترمینال |
| ۵۴ | ۷-۶ نحوه کار برد آردوینو UNO |
| ۵۴ | ۶-۷-۱ ارسال کد از طرف مدار اصلی |
| ۵۴ | ۶-۷-۲ نحوه کار در هر گردش |

۷ جمع‌بندی ۵۶

| | |
|----|--------------------------|
| ۵۶ | ۷-۱ مرور کلی |
| ۵۷ | ۷-۲ هزینه‌های صورت گرفته |
| ۵۸ | ۷-۳ کارهای آینده |
| ۶۰ | ۱- مطالب تکمیلی |

فهرست شکل‌ها

| | | |
|----|------|--|
| ۱۵ | ۱-۲ | بلوک دیاگرام طراحی شده برای گلدان هوشمند |
| ۱۸ | ۱-۳ | برد آردوینو UNO |
| ۲۰ | ۲-۳ | نمای بالا از برد Arduino Uno R3 |
| ۲۳ | ۳-۳ | برد آردوینو D1 نسخه Rev1 با تراشه ESP8266 |
| ۲۴ | ۴-۳ | سنسور رطوبت سنج YL-69 با خروجی دیجیتال و آنالوگ |
| ۲۵ | ۵-۳ | پین‌اوت ماژول YL-69 |
| ۲۶ | ۶-۳ | شیربرقی سلونوئید ۱۲ ولتی |
| ۲۷ | ۷-۳ | رله مورد نیاز برای اتصال به شیر برقی |
| ۲۸ | ۸-۳ | ماژول ماورای صوت HY-SRF05 |
| ۲۹ | ۹-۳ | نحوه قرارگیری پایه‌های ماژول ماورای صوت HY-SRF05 |
| ۳۰ | ۱۰-۳ | آداپتور مورد استفاده در مدار |
| ۳۳ | ۱-۴ | نمونه‌ای از ابزارهای داخل Blynk |
| ۳۴ | ۲-۴ | تنظیم نمایشگر رطوبت در نرم‌افزار Blynk |
| ۳۵ | ۳-۴ | اعلان فعالسازی در نرم‌افزار Blynk |
| ۳۶ | ۴-۴ | اعلان برقراری جریان در نرم‌افزار Blynk |
| ۳۷ | ۵-۴ | اعلان اخطار عدم تغییر وضعیت در نرم‌افزار Blynk |

| | | |
|----|-----|--|
| ۳۷ | ۴-۶ | رابط کاربری برنامه گلدان هوشمند در نرم‌افزار Blynk |
| ۳۹ | ۵-۱ | تصویر مازول رطوبت‌سنج در خاک |
| ۴۰ | ۵-۲ | جاسازی مازول ماورای صوت در درب مخزن |
| ۴۰ | ۵-۳ | نمای بالای مازول ماورای صوت به همراه قیف ورودی آب |
| ۴۱ | ۵-۴ | اتصال مازول شیربرقی به مخزن آب |
| ۴۱ | ۵-۵ | آداپتور ۱۲ ولتی مورد نیاز برای شیربرقی |
| ۴۲ | ۵-۶ | رله تک کانال بکاربرده شده در مدار |
| ۴۳ | ۵-۷ | اتصال دو برد برای برقراری ارتباط سریال |
| ۴۴ | ۵-۸ | شمای کلی طراحی و مدار پیاده‌سازی شده |
| ۶۰ | ۱ | برنامه Blynk در Google Play |
| ۶۰ | ۲ | برنامه Blynk در App store |
| ۶۱ | ۳ | صفحه شروع نرم‌افزار Blynk |
| ۶۱ | ۴ | وارد شدن به حساب در نرم‌افزار Blynk |
| ۶۲ | ۵ | پروژه‌های داخل نرم‌افزار Blynk |
| ۶۲ | ۶ | محیط پروژه مدنظر در نرم‌افزار Blynk |
| ۶۳ | ۷ | گام اول در نصب کتابخانه |
| ۶۴ | ۸ | گام دوم در نصب کتابخانه |

فهرست جدول‌ها

| | |
|----|------------------------------|
| ۲۲ | ۱-۳ ویژگی‌های مازول وای فای |
| ۵۸ | ۱-۷ جدول قطعات و مخارج پروژه |

فصل ۱

مقدمه

در این فصل به طور خلاصه در مورد انگیزه اصلی پروژه و اهمیتش در زندگی روزمره صحبت خواهد شد.

۱-۱ تعریف مسئله

هوشمندسازی و اتوماسیون صنعتی در دهه‌های اخیر مورد توجه بسیاری قرار گرفته است. هوشمندسازی می‌تواند فواید گوناگونی داشته باشد. اول اینکه با هوشمندسازی می‌توانیم مخارج مربوط به نیروی انسانی را کاهش دهیم و سپس با استفاده از اینکار کیفیت کار مدنظر را بالاتر ببریم. در واقع خطای انسانی مسئله‌ای اجتناب ناپذیر است. بنابراین با حذف این خطا، خطای کار پایین می‌آید. هوشمندسازی به تنهایی نمی‌تواند کافی باشد، زیرا بازهم نیاز به مدیریت و نظارت بی‌وقفه دارد. پدیده‌ای که باعث تسهیل این فرآیند در سالیان اخیر شده و امکان استفاده از آنرا راحت‌تر کرده است، رشد و گسترش پروتکل‌های ارتباطی اینترنتی است. در واقع با توجه به افزایش ظرفیت‌های اینترنت، گسترش دستگاه‌هایی که از اینترنت پشتیبانی می‌کنند و دیگر موارد، امکان نظارت بر فرآیند اتوماسیون از راه دور فراهم شده است. بدین ترتیب مثلاً فردی می‌تواند در خارج خانه از شرایط ماشین ظرفشویی خانه‌اش باخبر شود و یا اینکه اگر چراغی روشن مانده باشد، آنها را خاموش کند. اینها تنها بخشی از خدماتی است که اینترنت اشیاء^۱ در اختیار ما قرار می‌دهد.

¹Internet of Things

به طور می‌توان پروژه انجام شده را به دو بخش کلی تقسیم کرد که به شرح زیر است.

۱-۱-۱ هوشمندسازی آبیاری گیاهان

در واقع گیاهان برای بقا نیاز به مراقبت زیادی دارند و انسان‌ها باید شرایط آنها را مدنظر قرار دهند. یکی از این شرایط میزان رطوبتی است که در خاک گلدان قرار دارد. بدین ترتیب باید سامانه‌ای هوشمند طراحی کنیم که با سنجیدن محیط، تصمیم لازم برای برقرار کردن جریان آب به سمت گلدان را بگیرد.

۲-۱-۱ نظارت بر فرآیند آبیاری

به هر حال هر مقدار هم که اعتماد نسبت به فرآیند داشته باشیم، نیاز است که بر درستی انجام کار نظارت شود. علاوه بر آن، در مواقعی نیاز است که فرآیند از حالت خودکار خارج و به دست خود فرد واگذار شود. اگر بتوان نظارت را بدون وابستگی مکانی انجام داد، بسیار برای کاربران به صرفه خواهد بود. بنابراین هدف این بخش این است که ارتباطی بین فرآیند و کاربر به صورتی اینترنتی برقرار شود که عمل نظارت به خوبی صورت گیرد.

۲-۱ اهمیت مسئله

این نکته شناخته شده است که نگهداری از گیاه خانگی می‌تواند علاوه بر سلامت فیزیکی به دلیل بهبود کیفیت هوا، اثرات مثبت ذهنی و روانی نیز بر جو خانه بگذارد. بدین جهت انسان از مدت‌ها قبل اقدام به پرورش و نگهداری گیاه‌های تزئینی به‌طور خانگی گرفته است؛ در دنیای امروزی ممکن است به دلیل مشغله‌های ذهنی زیاد و درگیری‌های کاری، رسیدگی منظم به گیاه‌ها را برای صاحبش دشوار سازد.

بنابراین هر روشی که بتواند برای نگهداری گیاهان خانگی به صاحب آن کمک بکند، اهمیت بالایی دارد. به‌طور خاص این پروژه می‌تواند به فرآیند آبیاری منظم گیاه‌های خانگی کمک کند. آبیاری فراتر از حد درست می‌تواند موجب پوسیدگی ریشه گیاهان شود و آبیاری دیر به دیر آن‌ها موجب خشک شدن گیاه شود.

از سوی دیگر اگر به بعدهای کشاورزی صنعتی بپردازیم، متوجه خواهیم بود که مسئله کمبود آب کشاورزی مسئله‌ای فوق‌جدی در جهان است. در اینباره چند نکته قابل بیان است. اول اینکه در بیشتر کشورها از جمله کشور خودمان دچار خشکسالی هستیم و لذا منبع آب به طور قطعی کم است. دوم اینکه نحوه آبیاری‌های حال حاضر در بیش‌تر کشورها اصلاً بهینه نیست. بنابراین گام اول برای این مسئله بهبود شرایط کنونی است. با استفاده از یک هوشمندسازی مناسب، اصلاح روش آبیاری انجام شدنی است. بعد از اصطلاح شرایط می‌تواند به بهینه‌سازی بیشتر روشی که داریم بپردازیم. پس یک اهمیت مهم این پروژه به مسئله کمبود آب کشاورزی مرتبط است.

۱-۳ راهکار پیشنهادی ما

در این پروژه هدف داریم راهکاری برای خودکار سازی و هوشمندسازی فرآیند آبیاری گیاه ارائه دهیم، همچنین این امکان را برای پرورش‌دهنده گیاه فراهم سازیم تا با راحتی بیشتری از وضعیت گیاه(های) خود مطلع شود. در این قسمت راهحل پیشنهادی را به طور کلی مطرح می‌کنیم. توجه داشته باشید به طور کلی هرچقدر ما شناخت دقیق‌تری از وضعیت محیطی گیاه مدنظر داشته باشیم، هوشمندسازی به نحوه مطلوب‌تری صورت می‌گیرد. در این پروژه ما به دنبال یک مثال اسباب‌بازی^۲ بودیم و لذا به یک سنسور برای شناخت وضعیت گیاه بسنده کردیم.

راهحل پیشنهادی ما بدین‌صورت است که با قراردادن یک سنسور در گلدانی که گیاه در آن نگهداری می‌شود، می‌توان میزان رطوبت خاک آن را درنظر گرفت. در ادامه بُرد آردوینو^۳ که به این سنسور متصل است، از طریق اینترنت به یک برنامه برای گوشی‌های موبایل متصل می‌شود و اطلاعات مربوط به رطوبت موجود در خاک گیاه را برای کاربر به نمایش می‌گذارد. تا همینجا این کار می‌تواند به پرورش‌دهنده گیاه برای رصد وضعیت گیاهش کمک کند. قابلیت دیگری که این سیستم دارد آبیاری خودکار است. برای این کار یک شیر هوشمند در مخزنی کنار گلدان قرار داده‌ایم که به برد متصل است. سپس با توجه به دستور از مرکز سامانه، این شیر برقی با قطع یا وصل شدن، جریان آب را به سمت گلدان برقرار می‌کند. کاربر نیز خود امکان آبیاری در زمان دلخواه را با استفاده از برنامه دارد.

آب مورد استفاده برای آبیاری گیاه از یک مخزن ثابت تهیه می‌شود؛ در این مخزن یک سنسور ماورای صوت وجود دارد که سطح آب مخزن را می‌سنجد تا در صورت اتمام، کاربر مطلع شود. در

^۲Toy Example

^۳Arduino

واقع میزان آب درون این مخزن به طور مداوم به کاربر گزارش داده می‌شود. در پروژه‌های بزرگتر اما می‌توان مخزن را نیز غیرثابت در نظر گرفت و با هوشمندسازی می‌تواند مخزن را با منبع آب شهری شارژ کرد.

۴-۱ ساختار گزارش

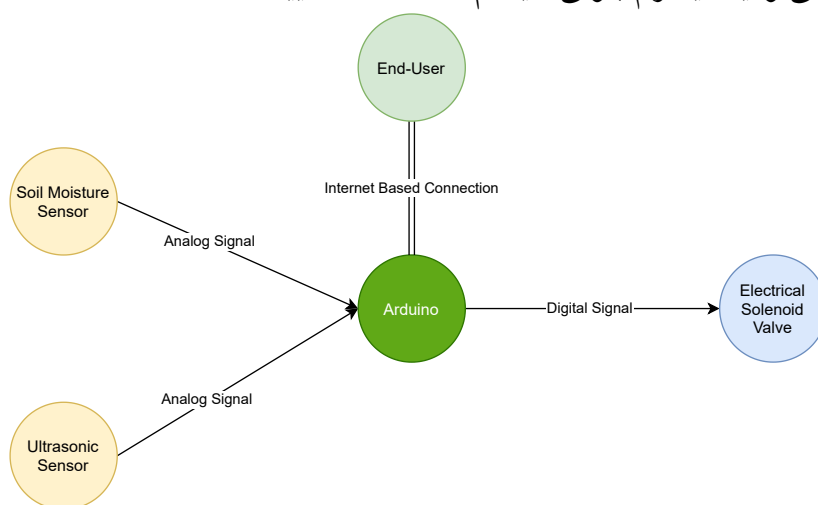
در این پروژه سعی شده است که ابتدا نیازمندی‌های سخت‌افزاری بررسی شود. بنابراین در فصل بعدی ابتدا به بررسی ماژول‌های سخت‌افزاری و انتظاری که از آنها داریم پرداخته می‌شود. سپس در فصل سوم به بررسی مشخصات ماژول‌های سخت‌افزاری انتخابی می‌پردازیم. در ادامه وارد نکات پیاده‌سازی پروژه می‌شویم. در فصل چهارم ابتدا در مورد رابط کاربری بحث می‌کنیم. سپس در فصل پنجم در مورد پیاده‌سازی فیزیکی صورت گرفته پروژه توضیح داده خواهد شد. در فصل ششم هم به آنالیز کد نرم‌افزاری پیاده‌سازی شده می‌پردازیم. در فصل آخر هم یک جمع‌بندی کلی پروژه داریم و مسائلی که در ادامه باید به آنها فکر شود، می‌پردازیم. در قسمت مطالب تکمیلی نیز می‌توانید نحوه راه‌اندازی برنامه Blynk و استفاده از رابط کاربری پروژه را مشاهده نمایید. همچنین در انتهای این قسمت نحوه نصب کتابخانه Blynk روی آردینو توضیح داده شده است.

فصل ۲

نحوه پیکربندی سخت‌افزاری

۱-۲ بلوک دیاگرام سیستم

در شکل ۱-۲ می‌توانید دیاگرام بلوکی سیستم را مشاهده نمایید



شکل ۱-۲: بلوک دیاگرام طراحی شده برای گلدان هوشمند

۲-۲ عملکرد و ارتباط کلی بخش‌های مختلف

۱-۲-۲ سنسور رطوبت سنج

همان‌طور که از نامش پیداست، این سنسور در خاکی که گیاه در آن قرار گرفته، کار گذاشته می‌شود. در بخش ۳-۴ مفصل‌تر در مورد سخت‌افزار سنسور توضیح داده‌ایم. کاری که این سنسور انجام می‌دهد این است که مقدار خشک‌بودن خاک را توسط دو شاخه‌اش اندازه گرفته و آن را به شکل سیگنال آنالوگ، برای برد ارسال می‌کند. این مقادیر دریافتی، با محدوده‌ای که برای خشکی تعیین می‌کنیم سنجیده می‌شود و در صورتی که خشکی از سطحی که انتظار داشتیم بیشتر بود و سیستم در حالت اتوماتیک بود، آبیاری گیاه صورت می‌گیرد.

۲-۲-۲ ماورای صوت

این ماژول، با قرارگرفتن در بالای مخزن آب، سیگنال‌های ماورای صوت ارسال کرده و بازتاب آن‌ها را دریافت می‌کند. با داشتن اختلاف زمانی بین سیگنال ارسالی و دریافتی می‌توان عمق آب را تشخیص داد. در عمل، سیگنال توسط کد برنامه فرستاده می‌شود و سیگنال بازگشتی به شکل آنالوگ به برد داده می‌شود؛ سپس اختلاف زمانی این دو سنجیده می‌شود تا از روی آن عمق آب محاسبه گردد.

۳-۲ ماژول شیر برقی

این ماژول کنترل شلنگ خروجی از مخزن آب را برعهده دارد. عملکرد این ماژول با ولتاژ ۱۲ ولت است و در دو حالت باز یا بسته می‌تواند قرار بگیرد. با توجه به میزان رطوبت یا دستور ارسالی توسط کاربر، با کمک رله‌ای که میان برد و این ماژول قرار دارد، باز و بسته‌شدن آن کنترل می‌شود و آبیاری صورت می‌گیرد.

۴-۲ ماژول رله

عملکرد این ماژول این است که سیگنال‌های ۰ یا ۵ ولتی که از برد آردوینو ارسال شده‌اند را به منظور قطع یا وصل کردن ماژول شیر برقی به‌کار بگیرد. در صورتی که سیگنال ارسالی ۵ ولت باشد، رله باعث می‌شود شیر برقی شروع به کار کند و وقتی سیگنال ارسالی ۰ ولت باشد، آن را از اتصال خارج می‌کند.

۱-۴-۲ برد Arduino D1-Rev1

این برد اصلی سیستم است و همگی سیگنال‌های دریافتی و ارسالی سیستم در این برد کنترل می‌شوند، به استثنای سیگنال‌های ارسالی به رله مربوط به شیر برقی. در عوض سیگنال‌های ارسالی به رله، توسط یک برد دوم صورت می‌گیرد. خود آن برد، تنها یک سیگنال ۰ و ۱ توسط این برد دریافت می‌کند که مشخص می‌کند سیگنال ارسالی به رله چه باشد. به جز آن، همگی سیگنال‌های ورودی توسط ماژول رطوبت‌سنج، ماژول ماورای صوت، و سیگنال‌های ارسالی توسط اپ‌کاربر در همین برد پردازش شده و سیگنال‌های خروجی مربوطه به آن‌ها ارسال می‌شود.

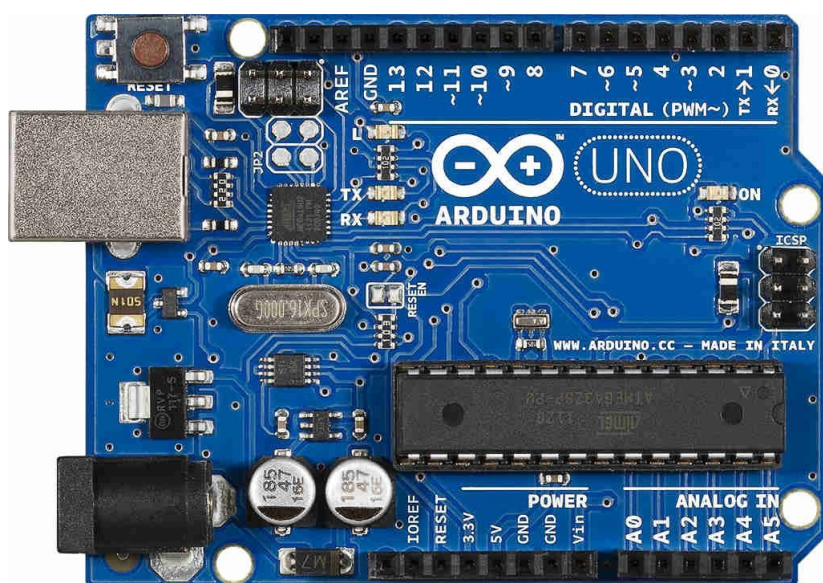
۲-۴-۲ برد Arduino Uno

این برد ثانویه، صرفاً برای ارسال سیگنال‌های باز یا بسته‌شدن برای رله متصل به شیر برقی است. خود این برد از برد اصلی یک سیگنال ۰ یا ۱ دریافت می‌کند که اگر ۰ بود مشخص می‌کند که سیگنال ارسالی به رله، ولتاژ ۰ باشد و اگر ۱ بود، سیگنال ارسالی ولتاژ ۵ باشد.

فصل ۳

سخت افزار مورد نیاز

۱-۳ آردوینو UNO



شکل ۱-۳: برد آردوینو UNO

۱-۱-۳ امکانات و ویژگی ها

• ابعاد

— پهنا: ۶۸/۶mm

– درازا: ۵۳/۴mm

• پردازنده: ATmega328P

– حافظه

* CPU AVR تا ۱۶ مگاهرتز

* ۳۲ کیلوبایت کش

* ۲ کیلوبایت SRAM

* ۱ کیلوبایت EEPROM

– امنیت

* Power On Reset (POR)

* Brown Out Detection (BOD)

– امکانات جانبی

* ۸-bit $2 \times$ تایمر/شمارنده با ثبات تناوب^۱ اختصاصی و کانال های مقایسه ای^۲

* ۱۶-bit $1 \times$ تایمر/شمارنده با ثبات تناوب اختصاصی، گرفتن ورودی و کانال های

مقایسه ای

* ۱ \times USART با تولیدکننده تناوبی نرخ-باد^۳ و تشخیص دهنده ابتدای قاب^۴

* ۱ \times controller/peripheral رابط موازی کاربری پرفرمانس^۵

* ۱ \times Dual mode controller/peripheral I2C

* ۱ \times Analog Comparator (AC) با ورودی معیار مقیاس پذیر^۶

* Watchdog Timer با اسیلاتور مجزای بر-تراشه

* شش کانال PWM^۷

* وقفه^۸ و wake-up هنگام تغییر پین

¹Period Register

²Compare Channel

³Baud-Rate

⁴start-of-frame

⁵Serial Peripheral Interface (SPI)

⁶Scalable reference input

⁷Pulse-Width Modulation

⁸Interrupt

- پردازنده ATmega16U2

* پردازنده ۸ بیتی AVR تحت RISC

* حافظه

۱۶ کیلوبایت ISP Flash

۵۱۲ بایت EEPROM

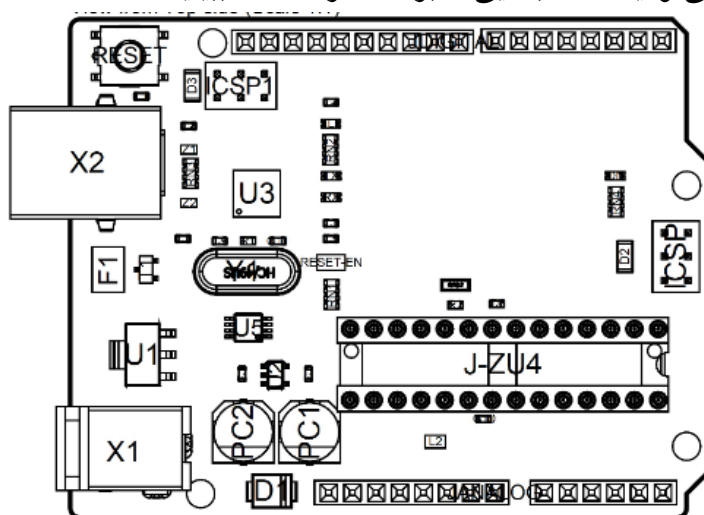
۵۱۲ بایت SRAM

• رابط کاربری debugWIRE برای دیباگ کردن و برنامه ریزی بر-تراشه

- انرژی

* ۵/۵-۲/۷ ولت

در شکل ۲-۳ می توانید نمای بالایی از برد و اجزای آن را ببینید.



Board topology

| Ref. | Description | Ref. | Description |
|-------|--------------------------------|-------|---------------------------------------|
| X1 | Power jack 2.1x5.5mm | U1 | SPX1117M3-L-5 Regulator |
| X2 | USB B Connector | U3 | ATMEGA16U2 Module |
| PC1 | EEE-1EA470WP 25V SMD Capacitor | U5 | LMV358LIST-A.9 IC |
| PC2 | EEE-1EA470WP 25V SMD Capacitor | F1 | Chip Capacitor, High Density |
| D1 | CGRA4007-G Rectifier | ICSP | Pin header connector (through hole 6) |
| J-ZU4 | ATMEGA328P Module | ICSP1 | Pin header connector (through hole 6) |
| Y1 | ECS-160-20-4X-DU Oscillator | | |

شکل ۲-۳: نمای بالا از برد Arduino Uno R3

۲-۳ مازول وای فای ESP8266EX

۱-۲-۳ امکانات کلیدی

- پشتیبانی از 802.11 b/g/n
- پشتیبانی از 802.11 n تا سقف ۷۲/۲Mbps
- یکپارچه سازی^۹
- ۲ رابط کاربری مجازی وای فای
- مونیتور خودکار beacon (TSF)^{۱۰}
- پشتیبانی از زیرساختار BSS Station mode، SoftAP mode و Promiscuous mode

۲-۲-۳ ویژگی های جزئی

این ویژگی ها مطابق جدول صفحه بعد هست.

⁹Defragmentation

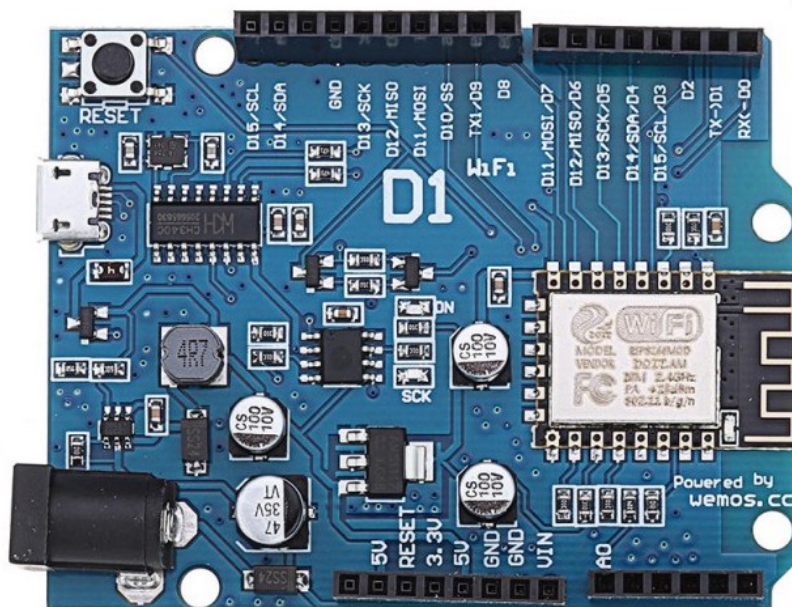
¹⁰Timing synchronization function

جدول ۳-۱: ویژگی های ماژول وای فای

| دسته بندی | آیتم | پارامترها |
|-----------|-----------------------|---|
| Wi-Fi | Certification | Wi-Fi Alliance |
| | Protocols | 802.11 b/g/n (HT20) |
| | Frequency Range | 2.4 GHz ~ 2.5 GHz (2400 MHz ~ 2483.5 MHz) |
| | TX Power | 802.11 b: +20 dBm |
| | | 802.11 g: +17 dBm |
| | | 802.11 n: +14 dBm |
| | Rx Sensitivity | 802.11 b: ~ 91dbm (11 Mbps) |
| | | 802.11 g: ~ 75dbm (54 Mbps) |
| | | 802.11 n: ~ 72 dbm (MCS7) |
| | Antenna | PCB Trace, External, IPEX Connector, Ceramic Chip |
| Hardware | CPU | Tensilica L106 32-bit processor |
| | Peripheral Interface | UART/SDIO/SPI/I2C/I2S/IR Remote Control |
| | | GPIO/ADC/PWM/LED Light & Button |
| | Operating Voltage | 2.5V ~ 3.6V |
| | Operating Current | Average value: 80 mA |
| | Operating Temperature | Range ~ 40°C ~ 125°C |
| | Package Size | QFN32-pin (5mm × 5mm) |
| Software | External Interface | - |
| | Wi-Fi Mode | Station/SoftAP/SoftAP+Station |
| | Security | WPA/WPA2 |
| | Encryption | WEP/TKIP/AES |
| | Firmware Upgrade | UART Download/OTA (via network) |
| | Software Development | Supports Cloud Server Development/Firmware and SDK for fast on-chip programming |
| | Network Protocols | IPv4, TCP/UDP/HTTP |
| | User Configuration | AT Instruction Set, Cloud Server, Android/iOS App |

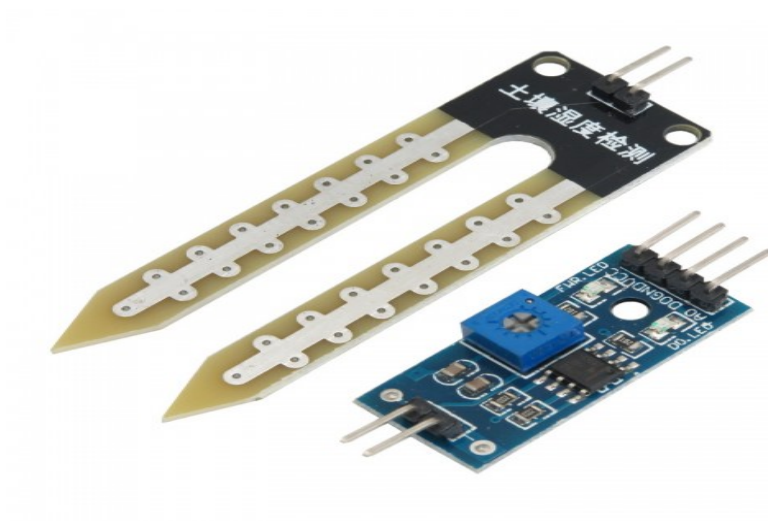
۳-۳ آردوینو D1 نسخه Rev1

این برد، همان برد Arduino Uno است که ماژول وای‌فای به آن متصل است.



شکل ۳-۳: برد آردوینو D1 نسخه Rev1 با تراشه ESP8266

۴-۳ سنسور رطوبت سنج YL-69



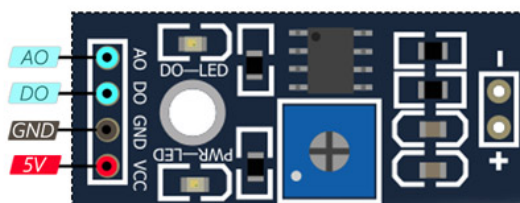
شکل ۴-۳: سنسور رطوبت سنج YL-69 با خروجی دیجیتال و آنالوگ

۱-۴-۳ ویژگی ها و امکانات کلیدی

- منبع تغذیه VCC: ۳/۳ الی ۵ ولت
- دارای خروجی ولتاژ دیجیتال^{۱۱}
- دارای خروجی ولتاژ آنالوگ^{۱۲}

¹¹DO: Digital Output

¹²AO: Analogue Output



شکل ۳-۵: پین اوت ماژول YL-69

- امکان تنظیم میزان حساسیت
- دارای نمایانگر انرژی (قرمز) و نمایانگر خروجی دیجیتال (سبز)
- دارای ۴ پین
- ابعاد ماژول $32\text{mm} \times 14\text{mm}$
- ابعاد میله‌ها $60\text{mm} \times 20\text{mm}$
- دارای روکش محافظ پوشیده با طلا^{۱۳} برای جلوگیری از اکسید شدن میله‌ها

۳-۴-۲ عملکرد کلی

میله‌های این ماژول در خاک قرار می‌گیرند و ماژول ولتاژی را بین این دو میله از طریق خاک عبور داده و مقاومت خاک را می‌سنجد. هرچه خاک رطوبت بیشتری داشته باشد، به علت رسانایی آب، مقاومت کم‌تری توسط ماژول خوانده می‌شود. پس درواقع این ماژول نه میزان رطوبت، بلکه به نوعی میزان «خشکی» خاک را می‌سنجد و عدد خوانده شده هرچه کم‌تر باشد یعنی خاک مرطوب‌تر است.

¹³Immersion Gold

۳-۵ شیر برقی



شکل ۳-۶: شیر برقی سلونوئید ۱۲ ولتی

۳-۵-۱ ویژگی‌های کلی

- ولتاژ: ۱۲ ولت DC
- قدرت: ۵ وات
- نوع عملیاتی: معمولاً بسته (N/C)
- دمای مایع عبوری: ۰ الی ۱۰۰ درجه سانتی‌گراد
- نوع مایع: آب یا مایعات کم‌غلظت
- قطر مجاری: ۲۰ میلی‌متر

۳-۵-۲ نحوه عملکرد

یک شیر سلونوئید، شیری است که به شکل الکتریکی کنترل می‌شود. این شیرها دارای سلونوئید هستند که خود متشکل از یک سیم‌پیچ به همراه یک پیستون فرومغناطیسی است. در حالت غیرفعال، پیستون روزنه ریزی را می‌بندد. هنگامی که جریان الکتریکی از سیم‌پیچ رد شود، در آن میدان مغناطیسی ایجاد

می‌کند و این میدان یک نیروی رو به بالا به پیستون وارد می‌کند که باعث می‌شود آن روزنه باز شود. ماژول مذکور، با ولتاژ ۱۲ ولت کار می‌کند و برای اینکه بتوان توسط خروجی ۵ ولت برد آردوینو آن را کنترل کرد، از یک رله استفاده کرده‌ایم.

۳-۶ رله



شکل ۳-۷: رله مورد نیاز برای اتصال به شیر برقی

۳-۶-۱ ویژگی‌ها

- کمپانی سازنده: Tongling
- ابعاد: $43\text{mm} \times 17\text{mm}$
- دارای فیبر فایبرگلاس دو-رو متالیزه
- کانکتور ترمینال پیچی
- دارای یک خروجی رله میلون
- ولتاژ مورد نیاز: ۵ ولت DC
- ولتاژ خروجی: ۲۲۰ تا ۲۴۰ ولت

- توان: ۲۵۰۰ وات
- جریان خروجی: حداکثر ۱۰ آمپر
- دمای نگهداری: ۴۰- الی ۸۵+ درجه سانتی گراد

۷-۳ ماژول ماورای صوت



شکل ۳-۸: ماژول ماورای صوت HY-SRF05

۱-۷-۳ ویژگی های کلی و نحوه عملکرد

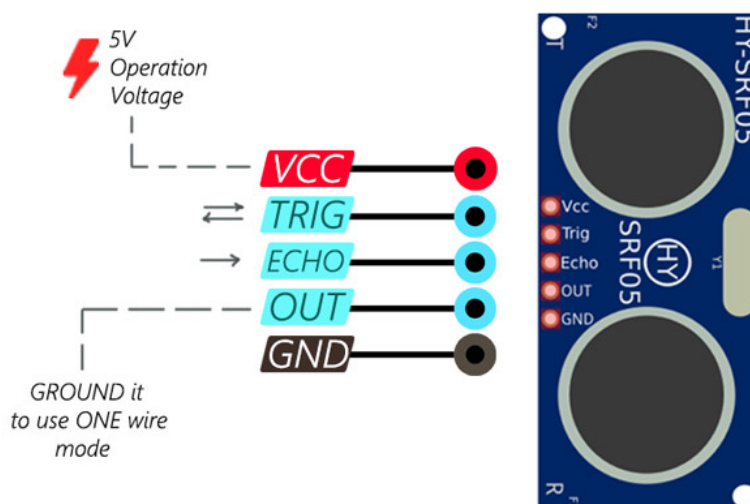
سنسور ماورای صوت، یکی سنسور مجاورتی است. نحوه عملکرد این سنسور این است که سیگنالی را که از محدوده شنوایی انسان فرکانس بالاتری دارد را به جسم مورد نظر ارسال کرده و انعکاس آن را دریافت می کند. زمان های ارسال و دریافت سیگنال ها در خروجی های ماژول موجود هستند و بدین صورت می توان با دقت بسیار خوبی فاصله سنسور تا سطح جسم (در مورد پروژه ما، سطح آب) را سنجید.

- محدوده پشتیبانی: ۲ الی ۳۰۰ سانتی متر
- فرکانس های ارسالی: ۴۰ کیلوهرتز
- ولتاژ تغذیه: ۵ ولت

• ابعاد $43\text{mm} \times 20\text{mm} \times 17\text{mm}$

• دارای ۵ پایه

۳-۷-۲ نحوه استفاده



شکل ۳-۹: نحوه قرارگیری پایه‌های ماژول ماورای صوت HY-SRF05

برای استفاده از ماژول، باید سیگنالی به مدت حداقل ۱۰ میکروثانیه به پایه TRIG ارسال کنیم، سپس ماژول به طور خودکار ۸ عدد پالس با فرکانس مذکور در بالا، ارسال می‌کند و تا زمانی که پالس‌های انعکاس یافته را دریافت کند، خروجی پایه ECHO همواره ۱ می‌ماند. بدین صورت وقتی که این خروجی ۰ شود، می‌توان زمان بازتاب و در نتیجه عمق را محاسبه کرد.

۳-۸ آداپتور

مشخصات آداپتوری که به برد اصلی آردوینو وصل می شود در ادامه آورده شده است.



شکل ۳-۱۰: آداپتور مورد استفاده در مدار

۳-۸-۱ ویژگی های کلی

- فرکانس کاری: ۴۷ الی ۶۳ هرتز
- ولتاژ ورودی: ۱۰۰ تا ۲۴۰ ولت AC
- ولتاژ خروجی: ۱۲ ولت DC
- جریان ورودی: ۰/۷ آمپر
- جریان خروجی: ۱ آمپر
- دمای نگهداری: ۴۰- الی ۸۵+ درجه سانتی گراد

فصل ۴

رابط کاربری

در این فصل به بحث در مورد رابطه کاربری می‌پردازیم. ابتدا به نیازمندی‌های سامانه مدنظرمان و سپس به نحوه پیاده‌سازی رابط کاربری می‌پردازیم.

۴-۱ نیازمندی‌ها

توجه کنید که هدف اصلی پروژه ساخت یک نمونه کوچک از یک پروژه صنعتی است. بنابراین نیاز به ویژگی‌های مختلفی در سامانه نداریم. اما به هر حال سعی شده است که پروژه حد کافی از ویژگی‌ها و ملزومات را داشته باشد. نیازمندی‌هایی که برای سمت کلاینت نیاز داریم مطابق موارد زیر است.

- قابلیت اطمینان و زمان بروزرسانی مناسب: باتوجه به اینکه نیاز داریم که از راه دور سامانه را کنترل کنیم پروتکل ارتباطی باید زمان بروزرسانی مناسبی داشته باشد. البته باید توجه داشت که این زمان به عوامل دیگری نظیر سرعت اینترنت و قطع یا وصل شدن آن نیز مربوط است.
- مانیتور کردن مقادیر سنسورها: همانطور که مشخص است دو سنسور اساسی در سامانه داریم که کاربر نیاز دارد که به صورت (تقریباً) پیوسته مقادیرش را بداند. بنابراین نیاز است که صفحه نمایش مناسبی برایشان داشته باشیم.
- اعلانات کاربر: نیاز داریم که در برخی حالت‌ها مثلاً بازشدن شیر، روشن شدن سیستم و ...

اعلاناتی به کاربر اعلام می‌شود. این اعلانات بهتر است که حالت جمع‌شونده^۱ داشته باشند. یعنی اینکه کاربر پیام‌های قبلی را از دست ندهد.

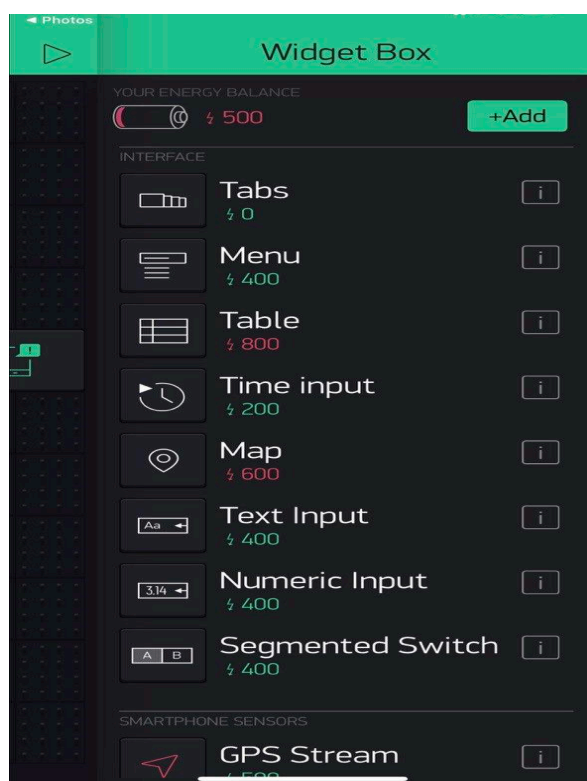
- کنترل حالت سامانه: سامانه مدنظر دارای دو حالت دستی و خودکار است. بدین منظور کاربر نیاز دارد که این حالت را تنظیم کند. همچنین اگر در حالت خودکار باشیم باید مقداری برای آستان رطوبت دریافت شود که سامانه به صورت خودکار شیر آب را باز یا بسته کند. یک دکمه نیز برای قطع یا وصل کردن شیر نیاز داریم.

با توجه به نیازمندی‌ها گزینه‌های مختلفی در پیش‌رو داشتیم. از بین آنها پلتفرم Blynk به نظرمان بسیار مناسب می‌آمد. علت این‌بود که با استفاده از این پلتفرم دیگر نیاز به پیاده‌سازی نرم‌افزاری از ابتدا نداشتیم و خیلی درگیر جزئیات که هدف پروژه نبود، نمی‌شدیم. علاوه بر این نکته، این پلتفرم ابزارهای متنوعی را در اختیار کاربر می‌گذارد که برای پروژه مدنظر مناسب می‌آمدند. در قسمت نرم‌افزاری نیز کتابخانه‌هایی برای آردینو دارد که پیاده‌سازی تابع‌های مربوط به سمت سرور را آسان می‌کند. با توجه به این مطالب Blynk انتخاب شد.

۴-۲ تنظیمات اولیه Blynk

ابتدا برای کار کردن با این پلتفرم نیاز است که یک اکانت ساخته شود. ساختن اکانت هزینه‌ای ندارد و به راحتی صورت می‌گیرد. سپس در داخل نرم‌افزار باید یک پروژه جدید ساخته شود. در پروژه امکانات و ابزارهای مختلفی در اختیار کاربر گذاشته می‌شود. برای مثال در تصویر زیر می‌توانید بخشی از ابزارهایی که نرم‌افزار در اختیار می‌گذارد را مشاهده نمایید.

¹Accumulative



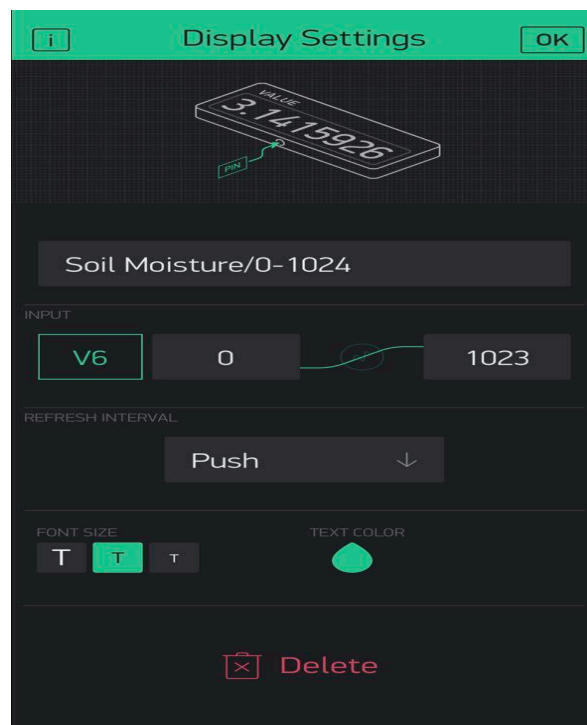
شکل ۴-۱: نمونه‌ای از ابزارهای داخل Blynk

تصویر بالا تنها بخشی از ابزارها هستند و کاربران می‌توانند هرطور که نیاز دارند، ابزارها را اضافه کنند. البته باید توجه داشت که برای استفاده از تمام ابزارها نیاز به پرداخت هزینه است. در هر حال ما سعی کردیم که ابزارهای زیر را به رابط کاربری اضافه کنیم.

۱. نمایشگر: دو نمایشگر برای نمایش میزان رطوبت و سطح آب مخزن استفاده کردیم.
۲. چراغ: یک چراغ برای نمایش وضعیت شیربرقی یا همان برقرار بودن یا نبودن جریان آب مخزن استفاده کردیم.
۳. اسلایدر: برای تنظیم آستان مقدار رطوبت نیاز به یک دریافت‌کننده داشتیم که به نظرمان اسلایدر سهولت کافی برای تنظیم را دارد.
۴. کلید: دو کلید برای پروژه نیاز است. یکی برای وضعیت سامانه که می‌تواند خودکار یا دستی باشد. یک کلید نیز برای وضعیت شیربرقی یا جریان آب نیاز است.
۵. اعلانات: برای دریافت اعلانات از سمت سرور نیاز است که یک ابزاری را به رابط کاربری اضافه

کنیم. در واقع صرفاً تصویر تلفن همراه در رابطه کاربری ظاهر می‌شود که نشان‌دهنده این است که سمت کلاینت از دریافت اعلانات پشتیبانی می‌کند.

حال بعد از اضافه کردن ابزار نیاز است که آنها را به پورت مربوط در آردوینو متصل کنیم. یک حسن دیگر پلتفرم این است که پورت‌هایی را به صورت مجازی در اختیار می‌گذارد که در واقع کانالی برای دریافت و ارسال اطلاعات بین برد اصلی و کلاینت می‌شود. در طول پروژه ما نیز از این پورت‌های مجازی استفاده کردیم و دیگر به صورت مستقیم در کلاینت به پورت‌های حقیقی برد اتصالی نداشتیم. مثلاً در تصویر زیر می‌توانید نحوه تنظیم کردن نمایشگر میزان رطوبت را مشاهده نمایید.



شکل ۴-۲: تنظیم نمایشگر رطوبت در نرم‌افزار Blynk

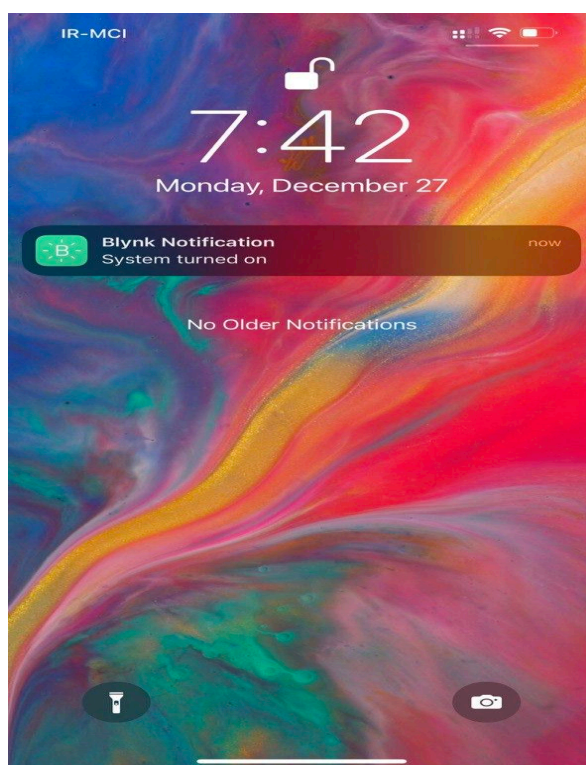
همانطور که در تصویر بالا مشاهده می‌کنید، پورت مجازی شماره ششم که مقادیر ۰ تا ۱۰۲۴ را به خود می‌گیرد به این نمایشگر نسبت داده شده است. از سمت کلاینت یا به اصطلاح Front نیاز به کار دیگری نیست و صرفاً یک مقدار Authentication مربوط به پروژه را ذخیره می‌کنیم تا در کد سرور از آن استفاده کنیم.

۴-۳ نحوه کارکرد

در این قسمت به بررسی حالت‌های مختلفی که در سمت کاربر ممکن است پیش‌بیاید را بررسی می‌کنیم.

۴-۳-۱ اعلان روشن شدن سامانه

در ابتدا نیاز است که یک پیام به کاربر ارسال شود که سامانه روشن است. در تصویر زیر این پیام در قسمت اعلانات تلفن همراه دریافت شده‌است.

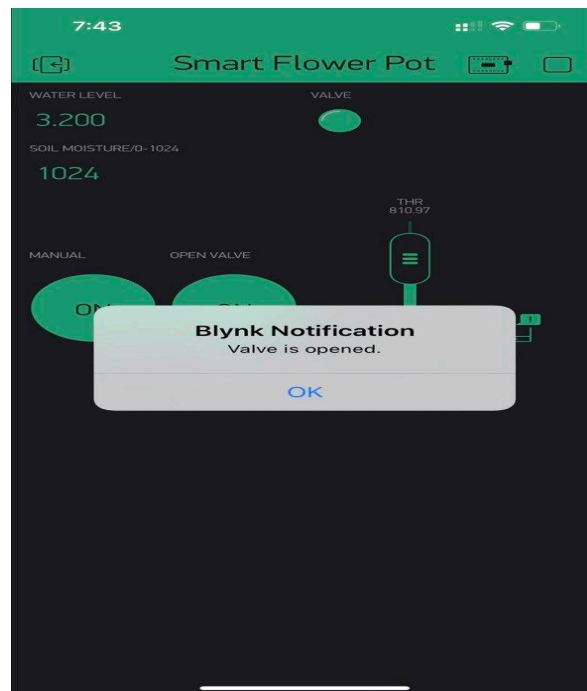


شکل ۴-۳: اعلان فعالسازی در نرم‌افزار Blynk

توجه کنید که حسن این اعلان این است که دیگر نیازی نیست که کاربر داخل برنامه باشد. همچنین اگر در تمام دستگاه‌هایی که نرم‌افزار بر روی‌شان نصب باشند، این اعلان نمایش داده می‌شود.

۴-۳-۲ بازکردن شیربرقی در حالت دستی

در حالتی که سامانه در حالت دستی باشد، می‌توان شیر را قطع یا وصل کرد. پس از وصل کردن، اعلامی از سوی سرور ارسال می‌شود. همچنین چراغ داخل نرم‌افزار روشن می‌شود. تصویر زیر این حالت را نشان می‌دهد.

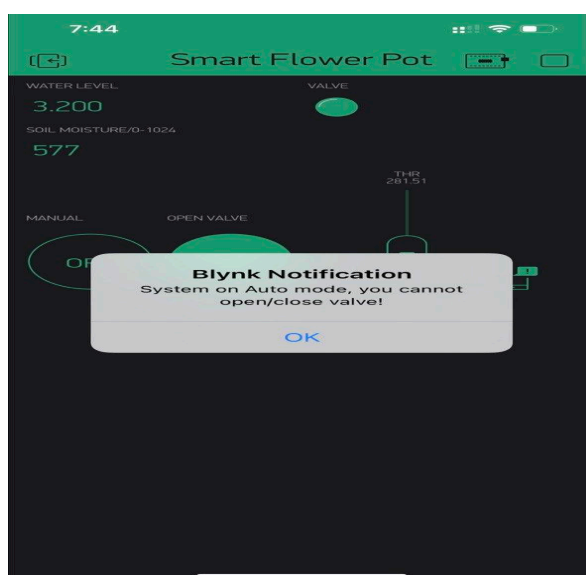


شکل ۴-۴: اعلان برقراری جریان در نرم‌افزار Blynk

بدیهی است که در تصویر بالا روشن بودن کلیدها به این معنی است که هم وضعیت در حالت خودکار است و هم اینکه کاربر کلید شیر را زده‌است.

۴-۳-۳ اخطار عدم تنظیم شیربرقی در حالت خودکار

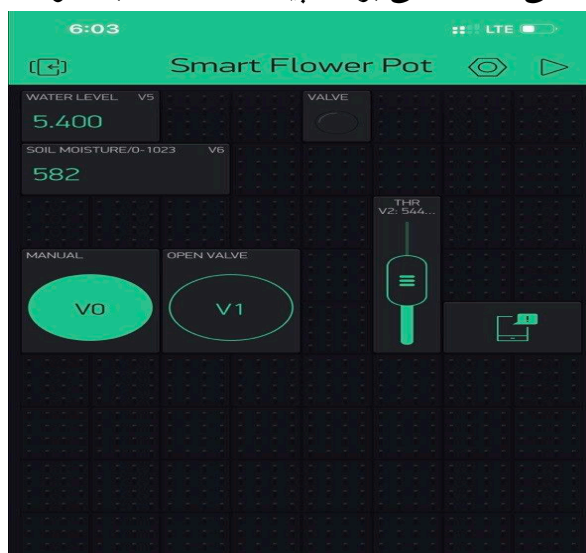
توجه کنید در حالت خودکار کاربر نمی‌تواند شیر را قطع یا وصل کند. به این ترتیب یک اخطار از سمت سرور در صورتی که کاربر اقدام به تغییر وضعیت کند، ارسال می‌شود.



شکل ۴-۵: اعلان اختار عدم تغییر وضعیت در نرم افزار Blynk

۴-۳-۴ حالت های دیگر

سامانه می تواند حالت های دیگری نیز داشته باشد که اعلانی برای آنها نیاز نیست. مثلاً وقتی که روی حالت خودکار باشیم و شیر برقی روشن شود. بدیهی است که کاربر با نگاه به چراغ داخل برنامه به این موضوع پی می برد. به طور کلی شمای کلی برنامه پیاده سازی شده به صورت زیر است.



شکل ۴-۶: رابط کاربری برنامه گلدان هوشمند در نرم افزار Blynk

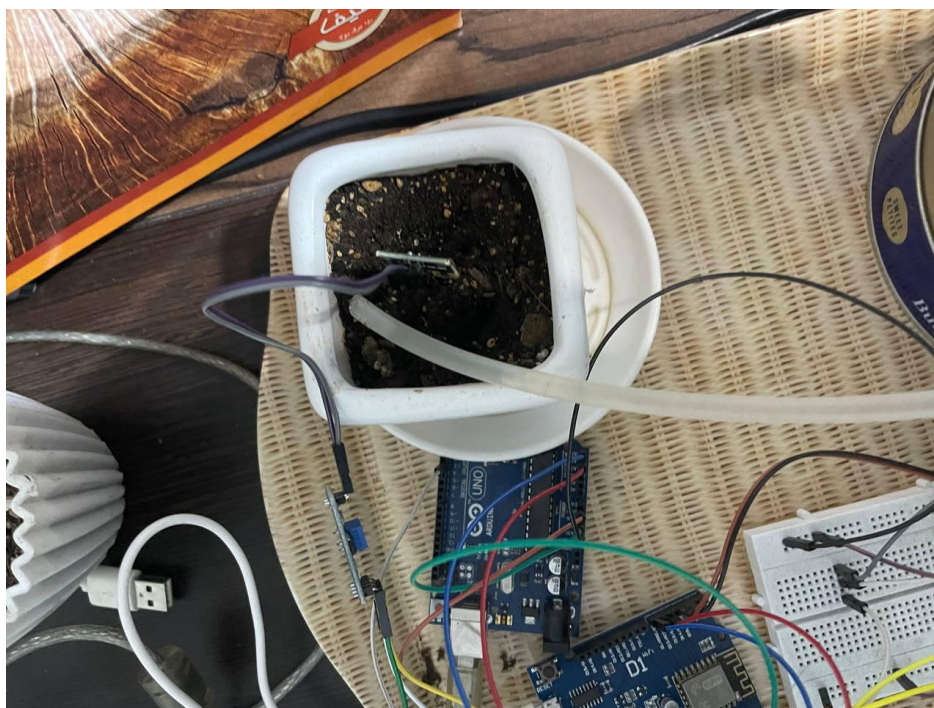
فصل ۵

نحوه پیاده‌سازی فیزیکی

در این فصل به مراحل پیاده‌سازی فیزیکی سامانه مدنظرمان می‌پردازیم.

۵-۱ اتصال ماژول رطوبت سنج

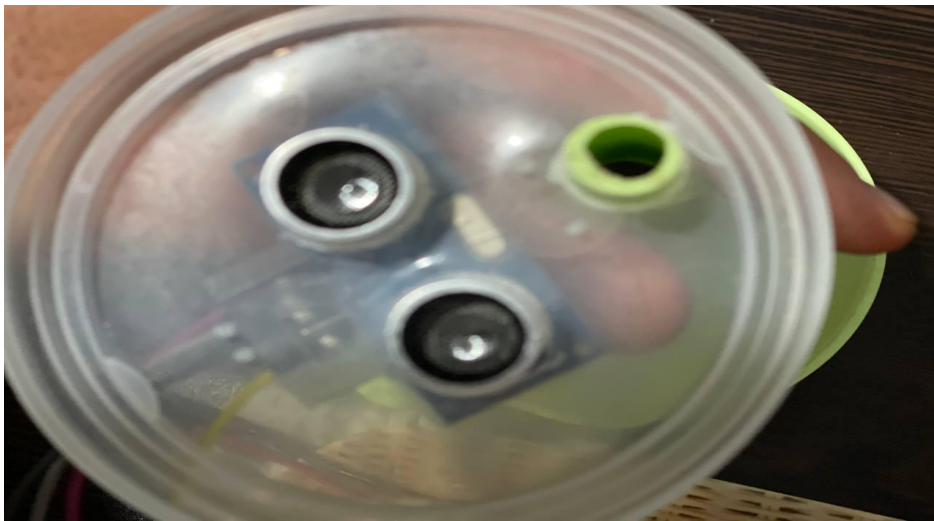
به طور کلی ماژول رطوبت‌سنج مدنظر که YL_69 است. یک سنسور و برد کوچک دارد. سنسور با دو سیم به برد وصل می‌شود. حال برای استفاده از برد رطوبت‌سنج نیاز است که زمین و VCC از برد آردوینو را به آن وصل کنیم. علاوه بر اینها روی ماژول رطوبت‌سنج یک پتانسیومتر قرار دارد که برای خروجی دیجیتال ماژول مورد استفاده قرار می‌گیرد. ما در این پروژه صرفاً به خروجی آنالوگ ماژول نیاز داشتیم و لذا آن را به برد اصلی وصل کردیم. در تصویر زیر نحوه قرارگیری ماژول رطوبت‌سنج را می‌توانید مشاهده کنید.



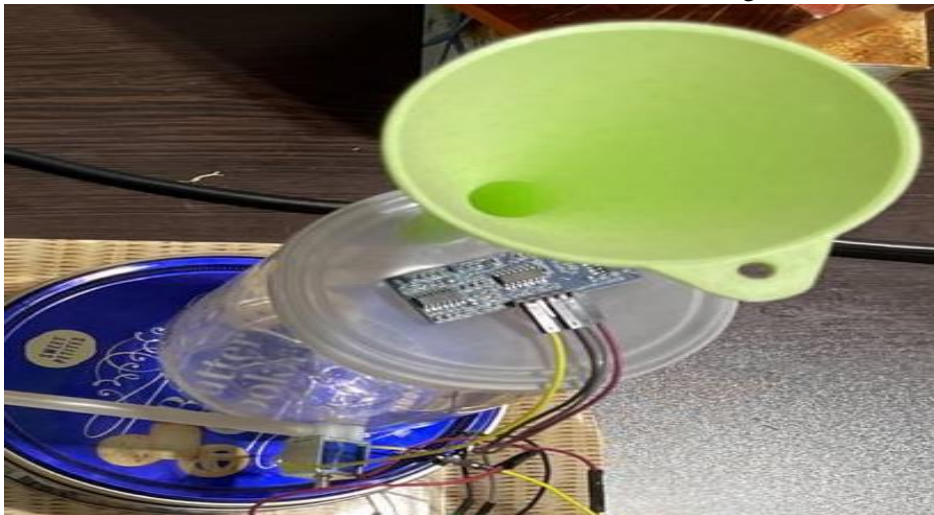
شکل ۵-۱: تصویر ماژول رطوبت‌سنج در خاک

۵-۲ اتصال ماژول ماورای صوت

همانطور که پیش‌تر گفته شد، این ماژول برای بررسی میزان سطح آب داخل مخزن است. برای استفاده از این ماژول نیاز به برقراری ارتباط با ۴ سیم به برد اصلی بود. دو سیم برای زمین و VCC و دو سیم مربوط به Trigger و Pulse بود. توجه کنید که Trigger ورودی ماژول و Pulse خروجی آن است. برای جاسازی ماژول ماورای صوت دو حفره در درب مخزن ایجاد کردیم و ماژول را جاسازی کردیم. در ادامه سعی کردیم که مخزنی مناسب تهیه کنیم که مناسب درب باشد. در دو تصویر زیر می‌توانید نحوه قرارگیری این ماژول را مشاهده نمایید.



شکل ۵-۲: جاسازی ماژول ماورای صوت در درب مخزن



شکل ۵-۳: نمای بالای ماژول ماورای صوت به همراه قیف ورودی آب

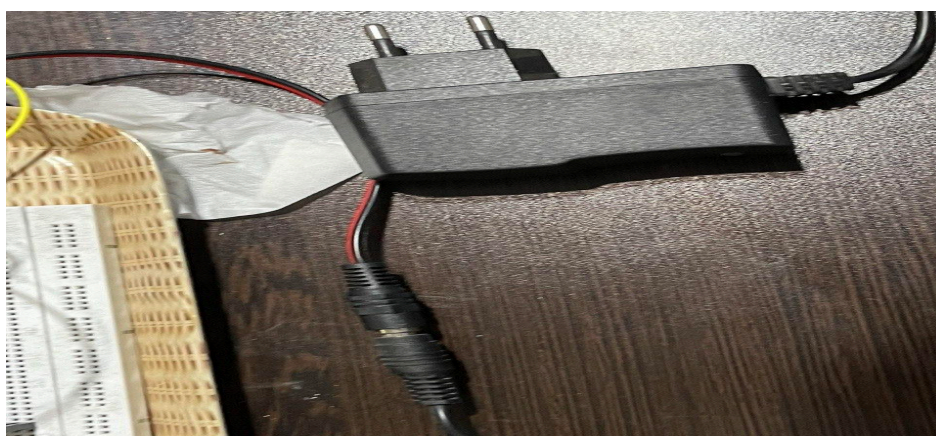
۵-۳ اتصال ماژول شیربرقی

این ماژول برای قطع و وصل کردن جریان آب به کار می‌رود. این شیر برای کار کردن نیاز به منبع ۱۲ ولتی دارد. توجه کنید که بردهای آردوینو خروجی‌هایی با ولتاژ حداکثر ۵ ولت را دارند. بنابراین نمی‌توان شیر برقی را مستقیم به برد وصل کرد. از این رو از یک آداپتور استفاده کردیم که مستقیماً به برق شهری وصل می‌شود. حالا یک خروجی آداپتور به رله وصل می‌شود و خروجی دیگر مستقیماً به شیر

برقی می‌شود و لذا سیم‌دیگر شیربرقی متصل به رله می‌شود. تصاویر زیر شیربرقی و آداپتور بکارگرفته شده در دو شکل زیر قابل مشاهده است.



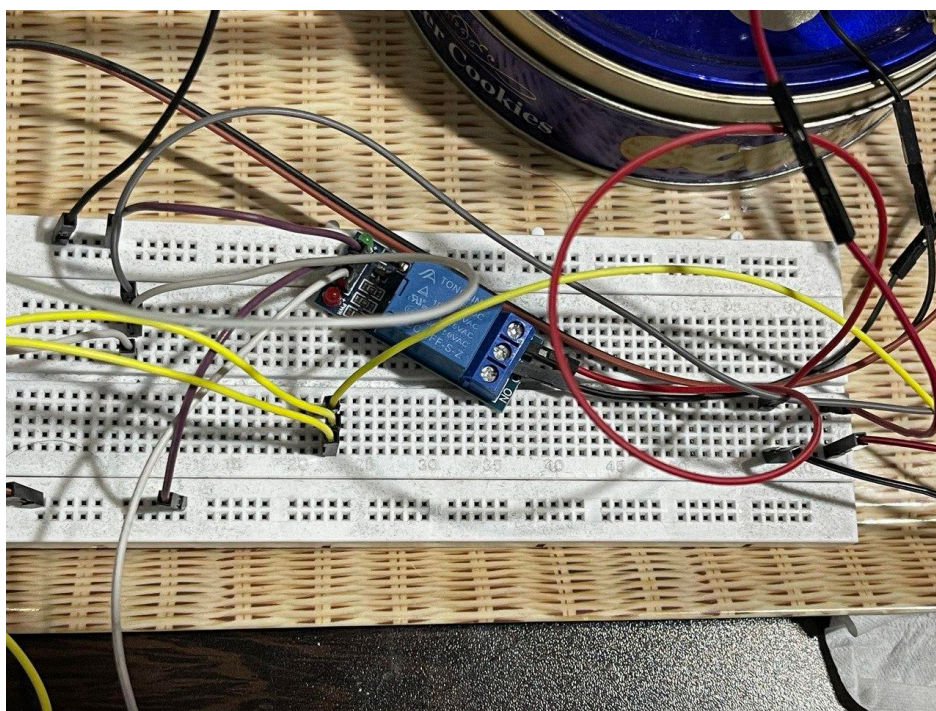
شکل ۵-۴: اتصال ماژول شیربرقی به مخزن آب



شکل ۵-۵: آداپتور ۱۲ ولتی مورد نیاز برای شیربرقی

۴-۵ اتصال رله

عملکرد رله متناظر با یک کلید است. یک طرف مربوط به مدار شیر برقی است. در واقع اتصال‌دهنده خروجی آداپتور و شیر برقی است. اما در طرف دیگر برای کار کردن رله نیاز به یک ورودی قطع یا وصل از طرف برد اصلی است. همچنین نیاز است که GND و VCC رله به برد اصلی متصل باشد. نکته قابل توجه این است که در طول پیاده‌سازی با توجه به مشکلی که پیش‌آمد، رله را به یک برد جایگزین وصل کردیم. لذا منظور از برد اصلی در این قسمت همان برد آردوینو UNO است. در تصویر زیر رله و خروجی‌ها و ورودی‌های قابل مشاهده است.

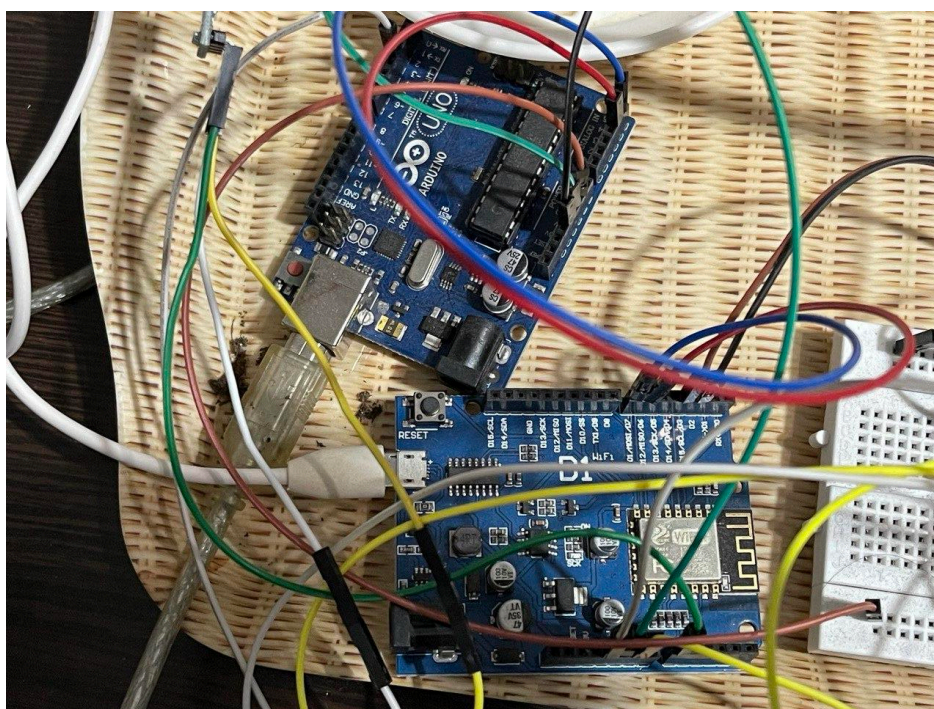


شکل ۵-۶: رله تک کانال بکاربرده شده در مدار

همچنین توجه داشته باشید، با توجه به نیازمان رله بکار برده شده، تک کانالی است و به صورت Normally closed است.

۵-۵ اتصال دو برد اصلی به یک‌دیگر

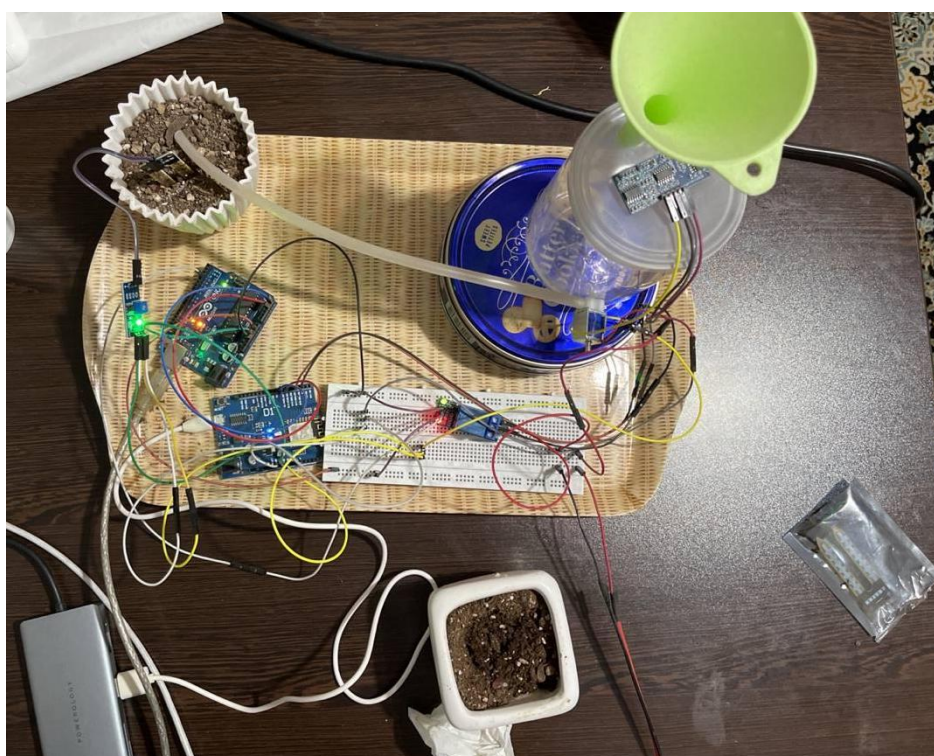
در طول پروژه به یک مشکلی برخوردیم و آن این بود که رله نمی‌توانست با برد اصلیمان کار کند. علت این موضوع به طور دقیق روشن نشد اما احتمالاً مشکلی که وجود داشت این بود که خروجی دیجیتال برد اصلی ضعیف بود و از ولتاژ کافی برخوردار نبود. به هرروی تصمیم بر آن شد که یک برد آردینو UNO را به مدار اضافه کنیم. وظیفه این برد جدید صرفاً برای قطع و وصل کردن رله بود. توجه کنید که برای صحت عملکرد کدی که برای برد اصلی و تعاملش با رله پیاده‌کرده بودیم، دقیقاً همان کد را روی آردینو UNO اجرا کردیم و مشاهده شد که رله کار می‌کند. بدین ترتیب فهمیدیم که کد درست است و خروجی برد اصلی احتمالاً ایراداتی دارد. حال نکته بعدی که ما به آن نیازمند بودیم، این بود که ارتباطی بین برد اصلی و برد UNO ایجاد کنیم. برای اینکار از ارتباط سریالی استفاده کردیم. در این ارتباط دو سیم وظیفه برقراری ارتباط را به عهده می‌گیرند. همچنین نیاز است که GND های دو برد را به یک‌دیگر وصل کنیم تا ارتباط برقرار شود.



شکل ۵-۷: اتصال دو برد برای برقراری ارتباط سریال

۵-۶ جمع‌بندی

به علت زیاد بودن سیم‌ها از یک برد ^۱ هم استفاده کردیم که بعضی از خروجی‌ها از جمله VCC و GND را روی آن وصل کردیم. همچنین از یک قیف برای ریختن آب درون مخزن استفاده کردیم که تماس احتمالی آب ورودی با ماژول ماورای صوت را کاهش دهد. در ادامه با توجه به اینکه شیربرقی خیلی قدرت چشم‌گیری نداشت، آن روی ارتفاع گذاشتیم تا سرعت خروج آب بیش‌تر شود. تصویر کلی سامانه در شکل زیر قابل مشاهده است.



شکل ۵-۸: شمای کلی طراحی و مدار پیاده‌سازی شده

¹Breadboard

فصل ۶

نحوه پیاده سازی نرم افزار

۶-۱ کتابخانه های مورد نیاز

همانطور که پیش تر گفته شد، برای قسمت رابط کاربری و اینترنت از پلتفرم Blynk استفاده شده است. این پلتفرم یک کتابخانه ای به نام BlynkSimpleEsp8266 در اختیارمان قرار می دهد. با این کتابخانه فعالیت های متنوعی می توان انجام داد که اصلی ترین آنها به شرح زیر است.

۱. اتصال به کلاینت با استفاده از ماژول WIFI قرار گرفته روی برد که Esp8266 نام دارد. در واقع نکته این است که با استفاده از این کتابخانه ابتدا به مودم خانه و سپس به کلاینت وصل می شویم.
۲. ارسال مقادیر مورد نیاز به کلاینت با استفاده از پورت های مجازی یا واقعی.
۳. دریافت آنی اطلاعات از سمت کلاینت به محض تغییر.

توضیحات بیشتر هر کدام از فعالیت ها در ادامه آمده است. کتابخانه بعدی که مورد استفاده قرار گرفته است همان SoftwareSerial است. اگر توجه کنید که خود بردهای آردوینو پین هایی به صورت Built-in دارند که ارتباط سریالی را به صورت UART برقرار می کنند. ما در پیاده سازیمان از این UART استفاده نکردیم. به جای آن از این کتابخانه استفاده کردیم که امکان استفاده از پین های دیگر را نیز می دهد. یکی از دلایل این کار این بود که اگر از UART استفاده می کردیم امکان چاپ و دیباگ کردن مقادیر را در ترمینال آردوینو نداشتیم. این موضوع به صورت دقیق تحقیق نشده است ولی در هنگام

پیاده سازی این مشکل وجود داشت و تصمیم به استفاده از روش دیگری شد. نحوه استفاده از کتابخانه آسان است و همانطور که در کد زیر قابل مشاهده است نیاز است که یک شی از کلاس SoftwareSerial ساخته شود و دو پین به آن نسبت داده شود.

```
1 #include <BlynkSimpleEsp8266.h> // Blynk for internet-based connection
2 #include <SoftwareSerial.h>      // Serial Communication package
3
4
5 // Serial Communication port setting
6 SoftwareSerial softSerial(13, 12);
```

۶-۲ تنظیمات اولیه و متغیرها

اولا یادآور می شویم که سامانه طراحی شده دو حالت دستی و خودکار دارد. بنابراین برای راحتی کار در کد دو فلگ به این دو نام تعریف کردیم. همانطور که در بخش رابط کاربری توضیح داده شد، نرم افزار Blynk پورتهایی را به صورت مجازی در اختیارمان قرار می دهد. ما از ۷ پورت مجازی استفاده کردیم. نام گذاری هرکدام در تکه کد زیر قابل مشاهده است.

```
1 // States of systems
2 #define AUTO 0
3 #define MANUAL 1
4
5 // Virtual Ports for Blynk
6 ///////////////////////////////////////////////////
7 // V0 -> Manual/ Auto button          //
8 // V1 -> Open/Close of valve button  //
9 // V2 -> Threshold of moisture       //
10 // V3 -> Valve state LED              //
11 // V4 -> Timer Display                //
12 // V5 -> Water level Display          //
13 // V6 -> Soil Moisture Display        //
```

```
14 //////////////////////////////////////////////////
```

توجه کنید که ما یک مازول WIFI در برد داریم که این مازول نیاز دارد تا به اینترنت متصل شود. بنابراین باید نام مودم که همان ssid و کلمه عبور را در کد تنظیم کنیم که مازول بتواند به پروژه متصل شود. نکته مهم بعدی این است که در واقع باید برد به کلاینت متصل شود. ارتباط برد با استفاده از سرورهای Blynk صورت می گیرد. نحوه کار به این صورت است که در هنگام ساخت پروژه در نرم افزار، یک توکن یا Authentication کد در اختیار قرار می گیرد. با استفاده از این کد می توانیم به کلاینت متصل شویم و ارتباط برقرار کنیم. بنابراین در تکه کد زیر موارد بالا را تنظیم شده است.

```
1 // You should get Auth Token in the Blynk App.
2 // Go to the Project Settings (nut icon).
3 char auth[] = "G3k5NqlDZoTHM0f8cFcn0VQzkQOMmQ-U";
4 // Your WiFi credentials.
5 char ssid[] = "Mobinnet";
6 char pass[] = "";
```

همانطور که پیش تر گفته شد، وضعیت رابط کاربری و مقادیر نمایشی باید هرچند لحظه ای بروزرسانی شوند. برای اینکار خود کتابخانه Blynk یک تایمر در اختیارمان قرار داده است که می توانیم از آن استفاده کنیم.

برای راحتی کار در ادامه ی پیاده سازی سعی شده که متغیرهای global تعریف کنیم که در تابع های مختلف از آنها استفاده کنیم. متغیر فاصله همان فاصله تا سطح آب است که در مازول ماورای صوت استفاده می کنیم. متغیر state همان وضعیت کل سیستم است. متغیر vavle_state همان وضعیت شیر را نشان می دهد. متغیر moist_thr مقدار آستان فعال سازی شیر برقی در حالت خودکار است. soil_moisture نیز متغیری است که از مازول رطوبت سنج می خوانیم.

```
1 // defines variables
2 BlynkTimer timer;
3 float distance;
4 int state = 0;
5 int valve_state = 0;
6 float moist_thr = 500;
7 int soil_moisture = 0;
```

۳-۶ تعیین فاصله ی سطح آب

تابع زیر تابعی است که در هر دوره صدا می شود و باید مقدار فاصله ماورای صوت تا سطح آب را محاسبه کند. نحوه کار کلی این است که امواج فراصوت از ماژول ماورای صوت خارج می شوند و سپس فاصله زمانی برگشت این امواج تا زمان ارسال آنها حساب می شود. با توجه به سرعت این امواج که ۳۴۰ متر بر ثانیه است، می توانیم فاصله مدنظر را حساب کنیم.

به طور دقیق تر ما از مد ۱ ماژول ماورای صوت که در اختیارمان بود استفاده کردیم. در واقع در این مد به خروجی out ماژول نیاز نداریم. نحوه پیاده سازی کد این است که در ابتدا مقدار trigPin برای لحظاتی کوتاه صفر می شود تا ماژول ریست شود. سپس مقدار High روی این پین ارسال می شود. بعد از آن حدود ۱۰ میلی ثانیه صبر می کنیم تا trigPin را خاموش کنیم. بعد از اینکار منتظر ورودی echoPin می شویم تا موج بازگشت را گزارش دهد. تابع pulseIn که در اختیارمان است این کار را انجام می دهد. توجه کنید خود این تابع اگر از یک زمانی صبر می کند تا موج بازگشتی دریافت شود. اگر این اتفاق نیفتاد خودش مقداری را خروجی می دهد. در آخر هم با استفاده از زمان رفت و برگشت می توانیم فاصله را حساب کنیم. توجه کنید که ما باید زمان بدست آمده از pulesIn را نصف کنیم زیرا صرفا فاصله تا سطح آب را می خواهیم.

```

1 float cal_distance() {
2     long duration;
3     //Clear the trigPin
4     digitalWrite(trigPin, LOW);
5     delayMicroseconds(2);
6     // Set the trigPin on HIGH state for 10 micro seconds
7     digitalWrite(trigPin, HIGH);
8     delayMicroseconds(10);
9     digitalWrite(trigPin, LOW);
10    // Read the echoPin, returns the sound wave travel time in
    microseconds
11    duration = pulseIn(echoPin, HIGH);
12    // Calculating the distance
13    distance = duration * 0.034/2;
14    return distance;

```


15 }

۴-۶ نحوه کار سیستم در هر گردش

به طور کلی در هر گردش کد که هر ۵۰۰ میلی ثانیه یکبار انجام می شود؛ نیاز است که فعالیت هایی صورت گیرد. دریافت مقدار مازول رطوبت سنج نکته خاصی ندارد. صرفا کافی است که مقدار با استفاده از analogRead خوانده شود. برای محاسبه فاصله نیاز است که از تابع cal_distance استفاده کنیم. در واقع اگر وضعیت در حالت خودکار بود و مقدار soil_moisture از آستان بیشتر بود یعنی باید شیر روشن شود. توجه کنید که هرچه مقدار soil_moisture بیشتر باشد یعنی اینکه خاک خشک تر است. در آخر هم یک delay ۵۰۰ میلی ثانیه ای می دهیم. تکه کد زیر قسمت محوری برد اصلی است.

```

1 void loop() {
2     Blynk.run();
3     timer.run();
4     distance = (float) ((int) (cal_distance() * 10)) / 10; // Calculate
                    distance to water
5     soil_moisture = analogRead(A0); // read from
                    analog pin A3
6     /// Checking whether we are in auto or manual state.
7     if (state == AUTO) {
8         if (soil_moisture > moist_thr) // Turning valve on, if
                    soil_moisture is above than threshold
9             valve_state = 1;
10        else
11            valve_state = 0;
12    }
13    /// Monitoring values in terminal
14    print_stats();
15    delay(500);
16 }
```

۵-۶ ارتباط با مشتری

در این قسمت به تابع هایی که مربوط به Blynk است، می پردازیم.

۱-۵-۶ فعال سازی سیستم

بدیهی است که در ابتدا نیاز داریم که به سرورهای Blynk وصل شویم. همچنین نیاز است که برای تایمیری که تعریف کردیم، تابع و زمانی نسبت دهیم. با اینکار در هر دوره زمانی تابع نسبت داده شده، صدا زده می شود. تابع نسبت داده شده همان update_client است. همچنین توجه کنید که هریک ثانیه یکبار، این تابع صدا زده می شود.

```

1 void blynk_start_connection () {
2     Blynk.begin(auth, ssid, pass); // Connect to wifi with ssid, pass.
3     After that, connect to client with auth.
4     Blynk.connect();
5     timer.setInterval(1000L, update_client); // Here you set interval (1
6     sec) and which function to call, We frequently update the state of
7     client.
8 }

```

در ادامه یک تابع داریم که وقتی مازول به کلاینت وصل شد، یک اعلان می دهد که سامانه فعال شد. در خط سوم کد نیز تمام مقادیر دریافت و سامانه sync می شود.

```

1 BLYNK_CONNECTED() {
2     Blynk.notify("System turned on");
3     Blynk.syncAll();
4 }

```

۲-۵-۶ دریافت حالت سیستم

همانطور که پیش تر گفته شد، پورت مجازی V0 برای دریافت وضعیت سامانه است که توسط کاربر تنظیم می شود. همچنین پس از اینکه مقدار دریافت شد، به کاربر یک اعلان ارسال می شود که وضعیت

سامانه تغییر کرده و به وضعیت موردنظر تغییر یافته است. توجه کنید که در تکه کدهای مربوط به چاپ مقادیر هم وجود دارد. این چاپ ها صرفاً برای مانیتور کردن وضعیت سیستم در ترمینال آردوینو است و به کلاینت مربوط نیست.

```

1 BLYNK_WRITE(V0)
2 {
3     state = param.asInt(); // assigning incoming value from pin V0 to a
    variable
4     Serial.print("Change state to: ");
5     Serial.println(state);
6     if (state == AUTO ) {
7         Blynk.notify("System now on Auto mode!");
8     } else {
9         Blynk.notify("System now on Manual mode!");
10    }
11 }

```

۳-۵-۶ دریافت حالت شیر

پورت مجازی V1 برای دریافت حالت شیر است. در واقع سامانه دو حالت دارد. اگر در حالت خودکار باشیم، کاربر اجازه تغییر را ندارد. در این صورت اگر کاربر تلاش برای تغییر وضعیت شیر کند، با اعلان خطر مواجه خواهد شد. اما در حالتی سامانه در حالت دستی باشد، می توانیم شیر را تغییر وضعیت بدهیم. در این صورت اعلانی برای کاربر نیز ارسال می شود.

```

1 BLYNK_WRITE(V1)
2 {
3     int pinValue = param.asInt(); // assigning incoming value from pin V1
    to a variable
4     Serial.print("Click on close/open button: ");
5     if (state == AUTO) {
6         Blynk.notify("System on Auto mode, you cannot open/close valve!");
7     } else {

```

```

8     valve_state = pinValue;
9     if (valve_state == 0) {
10         Blynk.notify("Valve is closed.");
11     } else {
12         Blynk.notify("Valve is opened.");
13     }
14 }
15 }

```

۴-۵-۶ دریافت مقدار آستان رطوبت

پورت مجازی بعدی برای دریافت مقدار آستان برای رطوبت است.

```

1 BLYNK_WRITE(V2)
2 {
3     moist_thr = param.asDouble(); // assigning incoming value from pin V2
4     Serial.print("Moisture threshold has been changed to: ");
5     Serial.println(moist_thr);
6 }

```

نکته ای که باید توجه داشت این است که طبق برداشتمان از توضیحات نرم افزار Blynk هر زمان که کاربر تغییری در رابط کاربری دهد، تابع مربوط به پورت مرتبط اجرا می شود.

۵-۵-۶ بروزرسانی وضعیت مشتری

همانطور که گفته شد، نیاز است که در هر یک دوره زمانی مناسب مقادیر سنسورها و وضعیت شیربرقی به کلاینت ارسال شود. برای اینکار هر یک ثانیه یکبار، تابع `update_client` صدا زده می شود. در دو خط اول روی پورت های مجازی V5 و V6 مقدار فاصله و رطوبت خاک نوشته می شود. سپس با توجه به وضعیت شیرمقدارش را ارسال می کنیم که نتیجه آن روشن شدن یا نشدن چراغ روی نرم افزار است. نکته بعدی این است که ارسال وضعیت شیربرقی به سمت برد دوم نیز همین جا انجام می گیرد. علت

اینکار این بود که تغییر وضعیت شیر و رابط کاربری تقریباً هم زمان باشد. در تکه کد زیر توضیحات داده شده قابل مشاهده است.

```

1 void update_client() {
2     Blynk.virtualWrite(V5, distance);           // Write Water level on blynk
3     Blynk.virtualWrite(V6, soil_moisture);      // Write Soil Moisture value
4     on blynk
5     if(valve_state == 0) {
6         Blynk.virtualWrite(V3, 0);              // Turning off LCD in Blynk
7         softSerial.write('0');                  // Send a code to Arduino UNO
8         board
9     } else {
10        Blynk.virtualWrite(V3, 255);             // Turning on LCD in Blynk
11        softSerial.write('1');                  // Send a code to Arduino UNO
12        board
13    }
14 }

```

۶-۶ چاپ متغیرها در ترمینال

برای مانیتور کردن وضعیت سیستم و همچنین دیباگ کردن یک تابع نوشتیم که مقادیر مورد نیازمان را در ترمینال چاپ کند.

```

1 void print_stats () {
2     Serial.print("soil: ");
3     Serial.println(soil_moisture);
4     Serial.print("Distance: ");
5     Serial.print(distance);
6     Serial.print(" Valve_state: ");
7     Serial.println(valve_state);
8 }

```

۷-۶ نحوه کار برد آردوینو UNO

همانطور که پیش تر گفته شد، نیاز است که برد آردوینو وضعیت شیربرقی را دریافت کند و سپس وضعیت رله را تغییر دهد.

۱-۷-۶ ارسال کد از طرف مدار اصلی

با استفاده از کتابخانه software serial این کار صورت گرفت. توجه کنید که برای مقدار ۰ کاراکتر '0' و برای مقدار ۱ کاراکتر '1' ارسال می شود.

```

1  if(valve_state == 0) {
2      Blynk.virtualWrite(V3, 0);           // Turning off LCD in Blynk
3      softSerial.write('0');               // Send a code to Arduino UNO
4      board
5  } else {
6      Blynk.virtualWrite(V3, 255);         // Turning on LCD in Blynk
7      softSerial.write('1');               // Send a code to Arduino UNO
8      board

```

۲-۷-۶ نحوه کار در هر گردش

در هر گردش که هر ۲۰۰ میلی ثانیه یکبار است؛ کد روی تابع `softSerial.available()` متوقف می شود. این توقف تا زمانی صورت می گیرد که سیگنالی از طرف فرستنده دریافت شود. البته به طور کلی اگر ارتباط قطع شود نیز از این تابع خارج می شویم. بعد از اینکه داده ای دریافت شده باشد، بر حسب اینکه کاراکتر دریافتی '0' (همان نماد OFF) در کد زیر باشد یا کاراکتر '1' (همان نماد ON) باشد، تصمیم لازم برای خروجی رله را می گیریم.

```

1 void loop()
2 {
3     if (softSerial.available()) { // If signal from main board available
4         val = softSerial.read();
5         if(val == OFF) {

```

```
6         digitalWrite(relay_port, LOW);    // Turn off relay
7     } else if (val == ON) {
8         digitalWrite(relay_port, HIGH);    // Turn on relay
9     }
10 }
11 delay(200);
12 }
```

فصل ۷

جمع‌بندی

در این بخش به مرور پروژه و برخی نکات باقی‌مانده می‌پردازیم.

۷-۱ مرور کلی

همانطور که پیش‌تر گفته شد، هدف اصلی بررسی ماژول‌های سخت‌افزاری مختلف و انتخاب آن‌ها در وهله اول بود. این کار سعی شد، با انجام جستجوهای صورت بگیرد. در کل ماژول اصلی که مرتبط با کارمان بود، ماژول رطوبت‌سنج بود که نوع YL_69 مورد استفاده قرار گرفت. برای بهتر شدن پروژه تصمیم گرفتیم که یک ویژگی جدید اضافه کنیم که سطح آب را درون مخزن را بسنجیم. برای اینکار نیز ماژول‌های مختلفی بود، اما برآن شدیم که از ماژول ماورای‌صوت استفاده کنیم که هم دقت بهتری دارد و هم اینکه در راستای کارهای دیگر می‌توان از آن استفاده نمود. در ادامه به برخی از ماژول‌های دیگر نظیر WIFI و شیربرقی نیاز داشتیم که سعی شد انتخاب‌های معقولی صورت گیرد.

پیاده‌سازی فیزیکی خیلی چالش برانگیز نبود. در مرحله خرید ماژول‌ها سعی کردیم که ۲ تا ۳ ماژول تهیه کنیم اما خوشبختانه ماژول‌ها درست کار می‌کردند و دقت و عملکرد مناسبی داشتند. برای مخزن خیلی هزینه خاصی متحمل نشدیم و سعی کردیم با منابع داخل خانه (!) کار را به انجام برسونیم. یک چالشی که در ادامه پروژه پیش آمد، مربوط به کار نکردن ماژول رله بود. در واقع نکته این بود که خروجی دیجیتال برد اصلی، نمی‌توانست رله را قطع یا وصل کند. برای حل مشکل شرایط یکسان کد را روی آردینو UNO آزمایش کردیم و دیدیم که رله با اتصال به آردینو UNO کار می‌کند. بدین ترتیب تصمیم

گرفتیم که این برد را به مدار اضافه کنیم و از آن برای قطع یا وصل کردن رله استفاده کنیم. در قسمت نرم‌افزاری یکی از انتخاب‌های مناسبان استفاده از پلتفرم Blynk بود. در واقع با استفاده از این موضوع دیگر نیاز نبود که خودمان از پایه سایت یا ربات تهیه کنیم. خوشبختانه نرم‌افزارهایش روی سیستم‌های مختلف از جمله IOS و Android به راحتی و مجانی قابل نصب بود. بنابراین بعد از نصب آسان نرم‌افزار، لازم بود که پروژه‌ای داخل نرم‌افزار بسازیم و به نوعی با استفاده از ابزارهای داخل نرم‌افزار، ارتباط لازم را با سرور برقرار کنیم و یا اینکه مقادیر مورد نیاز رو نمایش دهیم. برای قسمت کد سرور یا همان کدی که داخل آردینو بود لازم بود ۳ بخش اساسی را پیاده‌سازی کنیم.

۱. بخش ارتباط سنسورها: نیاز بود که کدی طراحی کنیم که مقادیر سنسورهای رطوبت‌سنج و ماورای‌صوت در هر مرحله بخواند و ذخیره کند.

۲. بخش ارتباط با client: شاید اصلی‌ترین بخش پروژه همین قسمت بود که باید هم تابع‌های مینوشتیم که از کلاینت پیام‌ها را دریافت کند و هم اینکه مقادیر حال حاضر سنسورها را به کلاینت بفرستد. خوشبختانه مستندات Blynk کامل بود و با خواندن‌ها کار خیلی چالش برانگیزی نبود.

۳. بخش ارتباط با آردوینو UNO: در این قسمت نیاز بود، کد مربوط به آردوینو UNO را می‌نوشتیم. اول نیاز داشتیم که مقدار مربوط به رله را از ماژول اصلی دریافت کنیم که اینکار با ارتباط سریالی که برقرار کردیم انجام شد و سپس هم باید خروجی مربوط به رله را با توجه به مقدار دریافتی، تنظیم می‌کردیم.

با انجام قسمت‌های مختلف و آزمایششان در هر مرحله، نیاز به فعالیت خاصی برای تست کلی نبود. عملکرد کلی سامانه بدون نقص بود. فقط در مواردی چون فشار آب خروجی از شیربرقی زیاد نبود، آزمایش‌ها طوری انجام می‌شد که این مسئله مشکل‌ساز نباشد.

۷-۲ هزینه‌های صورت گرفته

همانطور که مشخص است پروژه نیاز به ماژول‌ها و هزینه‌هایی داشت. اولاً در قسمت نرم‌افزاری استفاده‌مان از پلتفرم Blynk در حدی بود که نیاز به نسخه پولی نباشد. در مورد کابل‌ها و بردبرد و برد آردینو UNO نیاز به تهیه نداشتیم و خوشبختانه در سالیان گذشته خریداری شده بود. یک دسته‌ای ماژول‌ها نیز در اوایل سال ۱۴۰۰ خریداری شده و برخی دیگر نیز در یک ماه اخیر خریداری شدند. به

طور کلی هزینه‌هایمان که غیر از موارد جزئی و برد آردینو UNO هست در جدول ۷-۱ آمده‌اند. ممکن است برخی قیمت‌ها هم‌اکنون گران‌تر باشند که مسئله اجتناب‌ناپذیر است.

جدول ۷-۱: جدول قطعات و مخارج پروژه

| ردیف | نام قطعه | فروشنده | قیمت قطعه (تومان) |
|----------------|------------------------------|------------|-------------------|
| ۱ | Arduino D1 | دانشجوکیست | ۱۵۴۰۰۰ |
| ۲ | شیر برقی ۱۲ ولتی | | ۷۴۴۰۰ |
| ۳ | ماژول رطوبت خاک و زمین YL-69 | | ۱۶۹۰۰ |
| ۴ | ماژول رله تک کانال | | ۲۶۶۰۰ |
| ۵ | ماژول ماورای صوت | | ۳۵۹۰۰ |
| ۶ | آداپتور سوئیچینگ ۱۲ ولت | | ۶۳۳۰۰ |
| مجموع هزینه‌ها | | | ۳۷۱۱۰۰ |

۷-۳ کارهای آینده

قطعا این پروژه یک مثال کوچکی از یک پروژه صنعتی است که طبعا با آن ابعاد قابل قیاس نیست. اما به نظرمان همین پروژه کوچک هم می‌تواند گسترش‌های مناسبی داشته باشد. در ادامه سعی کردیم که برخی از ویژگی‌ها که می‌توان به پروژه اضافه کرد را بررسی کنیم.

- اضافه کردن سنسورهای سخت‌افزاری دیگر: در این راستا می‌توانیم سنسورهای نظیر دماسنج، سنسور شدت نور را اضافه کرد که کمک می‌کند تا ارزیابی دقیق‌تری از محیط داشته باشیم.
- تغییرات کد پیاده‌سازی شده: مطلب قابل تأمل که در آزمایش‌هایمان مشاهده کردیم این بود که واکنش سنسورها در برخی موارد لحظه‌ای به سرعت تغییر می‌کرد. مثلا در ماژول ماورای صوت وقتی آب وارد مخزن می‌شد، طبعا دقت ماژول دچار مشکل شد. زیرا لرزش در سطح آب برای لحظاتی قابل توجه است. بدین منظور می‌توانیم کد را حال حاضر را کمی تغییر بدیم تا از این دسته از پدیده‌ها کاربر را دچار مشکل نکند. مثلا یک راهکار این است که چند مقدار اخیر ماژول را ذخیره کنیم و میانگین آنها را به عنوان آماره خروجی دهیم.
- اضافه کردن امکانات به سامانه: ویژگی‌های مختلفی را می‌توان به سامانه اضافه کرد. یک ویژگی

که مدنظرمان بود این است که تایمری را اضافه کنیم که مثلا اگر زمانی خاک خشک بود، به مدت یک دقیقه شیر برق باز بماند. توجه‌کنید که مقدار مازول رطوبت ممکن است در این یک دقیقه از آستان تنظیم شده پایین‌تر شود ولی بازهم جریان آب قطع نمی‌شود. این نکته کمک می‌کند که آبیاری به صورت یکنواخت انجام شود.

- تغییرات رابط کاربری: برای داشتن رابط کاربری خود پلتفرم ویژگی‌های مختلفی را ارائه می‌دهد که برخی از آنها نیاز به هزینه داشت. مثلا یک ویژگی جالب برای رابطه کاربری این بود که از کاربران با استفاده از یک لینک به مقادیر دسترسی داشته باشند. یعنی در واقع آنها صرفا بتوانند پروژه را مشاهده و دسترسی تغییر داشته باشند. علاوه بر این‌ها می‌توانیم، یک تایمر برای نکته‌ای که در قسمت قبل گفته شده اضافه کنیم.

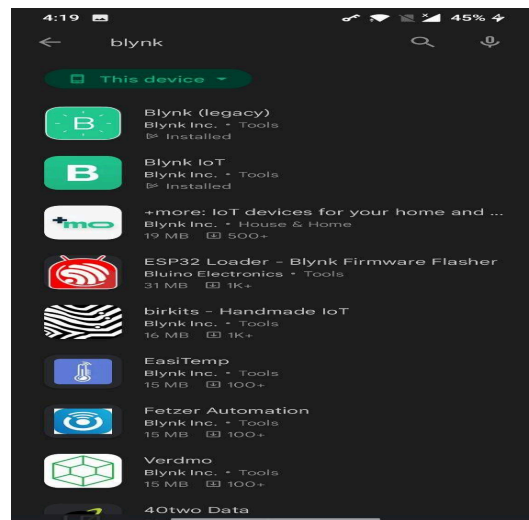
- مسئله مقیاس‌پذیری^۱: اگر توجه کنید در یک زمین بزرگ برای آبیاری گیاهان ممکن است که قسمت‌های مختلف زودتر یا دیرتر خشک شوند یا مثلا بعضی از گیاهان آستان مختلفی برای آبیاری نیاز داشته باشند. بدین ترتیب اولاً یک سنسور واحد نمی‌تواند کل محیط را ارزیابی کند و ما نیاز سنسورهای بیشتری داریم. اما به هر حال باید در یک سامانه واحد هماهنگی‌ها و تصمیمات صورت گیرد. لذا این مسئله می‌تواند در ابعاد مختلف چالش برانگیز باشد.

¹Scalability

۱- مطالب تکمیلی

نصب نرم افزار بر روی تلفن همراه

توجه کنید که برای نصب نرم‌افزار روی تلفن همراه نیاز به پرداخت هزینه نیست. بدین ترتیب برحسب سیستم عامل تلفن همراه می‌توانید در App Store یا Google Play عبارت Blynk را جستجو نمایید.

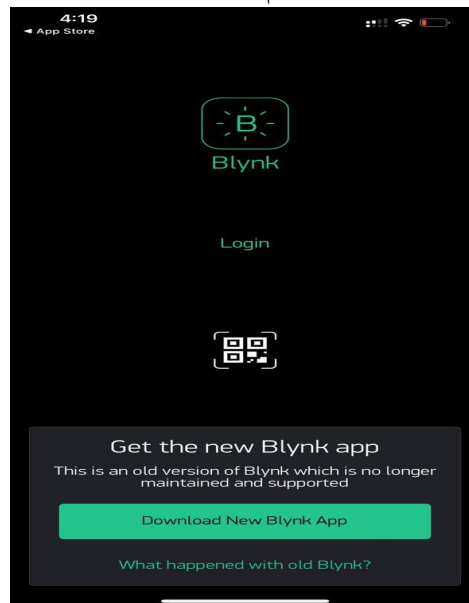


شکل ۱: برنامه Blynk در Google Play



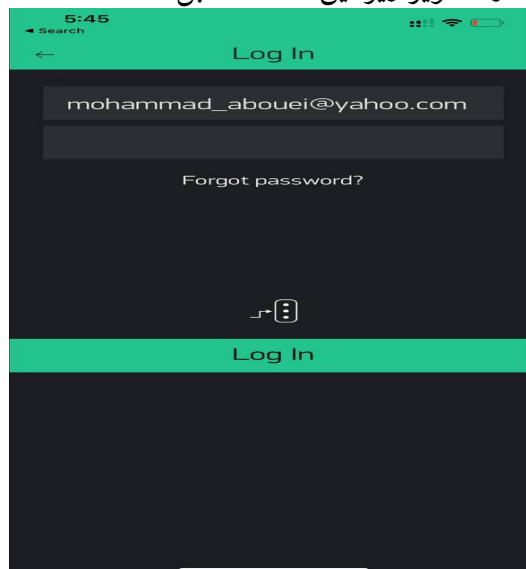
شکل ۲: برنامه Blynk در App store

حال اگر توجه کنید دو نسخه از این نرم‌افزار وجود دارد. ما از نسخه قدیمی‌تر آن استفاده کردیم که Blynk 0.1 (legacy) نام دارد. پس از نصب نرم‌افزار وارد آن شوید و با صفحه زیرمواجهه خواهید شد.



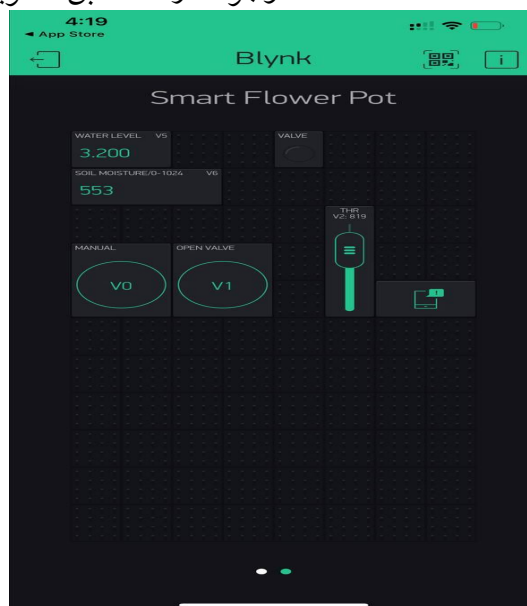
شکل ۳: صفحه شروع نرم‌افزار Blynk

حال روی قسمت login کلیک کنید و با وارد یک صفحه جدید خواهید شد. در این صفحه در قسمت ایمیل mohammad_abouei@yahoo.com را وارد کنید. همچنین برای کلمه عبور نیز عبارت ۱۲۳۴۵۶۷۸ را وارد نمایید. در تصویر زیر این صفحه قابل مشاهده است.



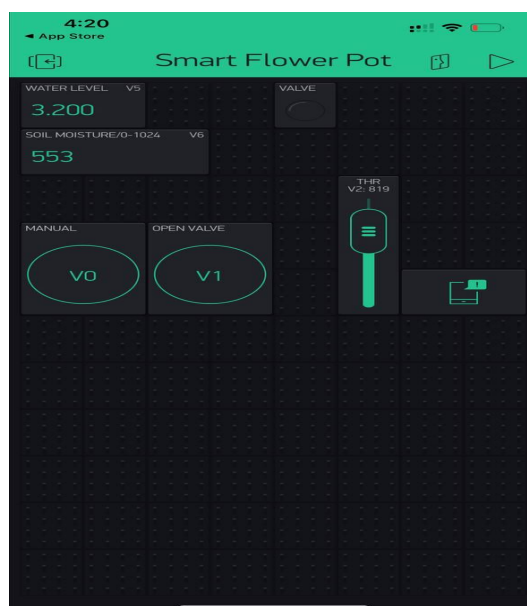
شکل ۴: وارد شدن به حساب در نرم‌افزار Blynk

پس از وارد شدن پروژه‌های مختلفی که با این اکانت ساخته شده نشان داده می‌شود. که در اکانت معرفی شده فقط یک پروژه Smart Flower Pot وجود دارد. مطابق تصویر زیر روی آن کلیک نمایید.



شکل ۵: پروژه‌های داخل نرم‌افزار Blynk

حال که وارد پروژه مدنظر شدیم، روی گزینه بالا سمت راست یعنی علامت شروع کلیک می‌کنیم. در این حالت هر موقع برد اصلی به اینترنت وصل شد، برنامه اطلاعات را دریافت می‌کند و کار تمام است.

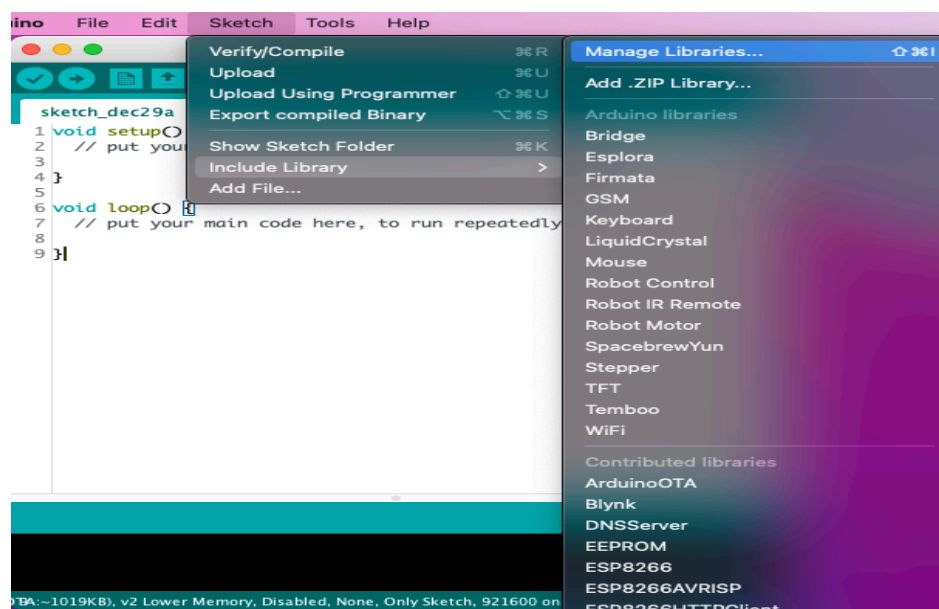


شکل ۶: محیط پروژه مدنظر در نرم‌افزار Blynk

نصب کتابخانه Blynk در آردینو

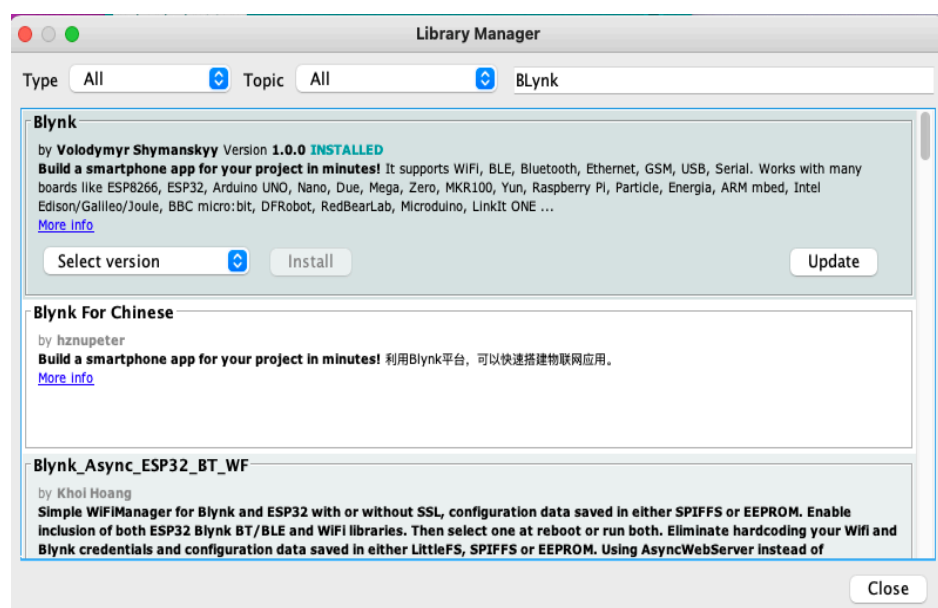
با توجه به راهنمای نصبی که خود سایت Blynk آموزش داده، ۳ راه برای نصب کتابخانه Blynk وجود دارد. در این **لینک** راه‌های مختلف قابل مشاهده هستند. در این قسمت ما روشی که خودمان پیش‌گرفتیم را توضیح می‌دهیم. این روش مطابق با روش اول در لینک مذکور است.

ابتدا وارد نرم‌افزار آردینو شوید. قسمت sketch را انتخاب کنید و سپس وارد قسمت Include library شوید. از این قسمت گزینه manage libraries را انتخاب کنید. مطابق با تصویر زیر که این کار صورت گرفته است.



شکل ۷: گام اول در نصب کتابخانه

حال پس از انتخاب بالا، صفحه‌ای ظاهر می‌شود. در قسمت جستجوی آن عبارت Blynk را جستجو نمایید (ممکن است کمی جستجو طول بکشد). حال گزینه اول را مطابق با تصویر زیر انتخاب کنید و با انتخاب آخرین ورژن گزینه نصب را انتخاب کنید تا نصب انجام شود.



شکل ۸: گام دوم در نصب کتابخانه

توجه کنید که چون ما از قبل این کتابخانه را نصب کردیم، گزینه بروزرسانی در تصویر بالا آمده است که طبیعی است. اگر این کتابخانه را نصب نداشته باشید، به جای گزینه بروزرسانی، نصب ظاهر می‌شود. پس از اینکه نصب انجام شد، کار تمام است و امکان استفاده از کتابخانه را در کد خواهید داشت.

نکته بعدی که قابل توجه این است که چون از ماژول ESP8266 که همان ماژول WIFI است، استفاده کردیم. نیاز است که در برخی ورژن‌های آردوینو درایور آن را نصب کنید. در این **لینک** می‌توانید نحوه نصب آن را مشاهده کنید.

مراجع

1. Arduino Uno Product Reference Manual
<https://docs.arduino.cc/static/28c18e21bd6f0d793c323ed839a6575d/A000066-datasheet.pdf>
2. User Manual Documentation for the Blynk IoT Application
<https://docs.blynk.cc/>
3. ESP8266EX Wi-Fi module datasheet
https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf
4. ElectroPeak introduction on the YL-69 Moisture Module
<https://electropeak.com/soil-moisture-sensor-1>
5. TheCafeRobot tutorial on using the YL-69 Moisture Module <https://thecaferobot.com/learn/interfacing-soil-moisture-sensor-with-arduino-2/>
6. Tameson product description for Solenoid Valves
<https://tameson.com/solenoid-valve-types.html>
7. Daneshjookit tutorial for the 12V Relay Module
<https://daneshjookit.com/module/useful/%D8%B1%D9%84%D9%87-relay/2054-relay-module.html>
8. Robo-Electronics specification page for the SRF04 Module
<https://www.robot-electronics.co.uk/srf04.html>
9. ElectroPeak introduction on the SRF04-SRF05 Ultrasonic Modules
<https://electropeak.com/learn/getting-started-with-ultrasonic-module-and-arduino>