به‌نام خداوند بخشنده مهربان

Dashboard Documentation

آکادمی اینترنت اشیاء

پاییز ۹۷

فهرست عناوین :

---

۱. معرفی کلی :

داشبوردی که پیش روی شما قرار داد، با تکنولوژی توسعه‌ی فرانت‌اند یعنی HTML5,CSS و JavaScript ایجاد شده است.

که به وسیله‌ی آن می‌توانید داده‌هایی که در هر لحظه توسط سنسورها به سرور ارسال می‌شوند را در چهار ویوی مختلف به شرح زیر مشاهده و پیگیری نمایید؛

ویو ی اول : که به آن Grafical view می‌گوییم متشکل از ۱۲ تایل (tile) است که در هر تایل مقدار میانگین داده‌های جمع‌آوری شده توسط سنسورهای مربوطه به نمایش در می‌آید. و تصویر پس‌زمینه نیز مطابق با میانگین به دست آمده و Thresholdی که تعریف شده تغییر می‌کند.

علاوه بر این‌ها چنانچه هر کدام از مقادیر دریافتی از سرور خارج از Threshold تعیین شده باشد، گوشه‌ی سمت راست هدر تایل مربوط به آن علامت هشدار ظاهر خواهد شد.

ویوی دوم : یا chart view، این امکان را به کاربر می‌دهد تا هریک از داده‌های جمع‌آوری شده توسط سنسورها را به صورت یک چارت مولتی چنل مشاهده نماید.یعنی به عنوان مثال اگر ۸ سنسور رطوبت‌خاک داریم، بتوانیم نمودار مربوط به همه‌ی آن‌ها را در یک چارت مشاهده کنیم.

در ویوی سوم : یا combined view بستری فراهم شده تا کاربر بتواند تلفیقی از دو ویو قبل را داشته باشد و با کلیک بر تایل گرافیکی هریک از سنسورها چارت مربوطه برایش نمایش داده شود.

و در ویوی چهارم : یا error view برای هریک از سنسورها یک button (که المانی از یک چراغ led است) در نظر گرفته شده که بسته به Threshold تعیین شده، چنانچه مقدار دریافتی در محدوده‌ی نرمال باشد به رنگ سبز، چنانچه از Threshold بسیار تجاوز کرده باشد قرمز و چنانچه به عبور از حد نرمال نزدیک باشد به رنگ زرد تغییر میکند.

۲. آشنایی با فایل‌ها :

تمامی فایل‌های مربوط به داشبورد در دایرکتوری final dashboard، و در پوشه‌های view1، view2، view4، قابل دسترس هستند.

- **پوشه‌ی view1** :

از آنجایی که برای ایجاد اسکلت خام و خالی تایل‌ها از کدهای از پیش تعریف شده‌ی داشبورد tipboard بهره بردیم؛ سیو کردن این اسکلت خام با فایل‌های مخصوص آن همراه بود لذا به منظور مرتب سازی، فایل‌های دخیره شده را به صورت زیر دسته‌بندی کردیم :

در کنار فایل html (value.html) دو پوشه با نام‌های js css ایجاد کردیم که درون هریک از آن‌ها پوشه‌ای به نام lib قرار دارد و فایل های css و js آماده‌ای که از ذخیره‌ی اسکلت خام حاصل شده بود را درون آن‌ها قراردادیم.

و فایل های css و jsی را هم که خودمان نوشته بودیم درون پوشه‌های مذکور و خارج از پوشه‌ی lib قرار دادیم.

علاوه بر این ها در پوشه‌ی ویو ۱ فایل html ویو ۳ با نام view3.html هم قرار دارد و از آن جا که این ویو تلفیقی از ویوی۱ و۲ می‌باشد، لذا از همان کدهای جاوااسکریپتی که ویو ۱ از آن‌ها استفاده می‌کند بهره برده و فایل css آن به نام view3.css نیز داخل پوشه‌ی CSS قرار داده شده و برای پیاده‌سازی چارت در آن، به گونه‌ای که با کلیک بر هدر هر تایل چارت مربوط به آن نمایش داده شود، فایلی به نام chart.js نوشته و داخل پوشه ی js قرار داده شده است.

پوشه‌ی img نیز محل قرارگیری تمام تصاویر و المان‌هایی است که در داشبورد استفاده شده.

غیر از موارد بالا پوشه‌ی دیگری نیز به نام Highcharts در این پوشه قرار گرفته که کتابخانه‌ای از فایل‌های آماده برای استفاده از گیج watt meter در آن قرار دارند.

اما به غیر از این گیج، ما از دو گیج دیگر نیز برای نمایش temperature و Water Level استفاده کردیم و به همین منظور فایل‌هایی با نام‌های d3.v3.min.js و gauge.min.js را که به صورت آماده دانلود کردیم درون پوشه ی lib که در پوشه ی js است، قرار داده و آن‌ها را در فایل html خود include کردیم.

آنچه در مورد فایل‌های js این ویو نیاز به توضیح دارد، آن است که اولا، همانطور که قبلا گفته شد ویو ۳ هم از همین فایل‌ها (به علاوه ی فایل chart.js) استفاده می‌کند و ثانیا نیمی از تایل‌های این ویو شامل : humidity، SoilMoisture of flowers، SoilMoisture of vegetables، Water Level، Ph of vegetables، و Ph of Flowers،

از فایل‌های جاوااسکریپت مجزایی که همنام با خودشان هستند استفاده می‌کنند و نیم دیگر آن‌ها شامل Temperature، Motion Detector، Photoresistor ، watt meter، Floor Humidity، و Gas، از یک فایل جاوااسکریپت یکپارچه به نام valueN.js استفاده می‌کنند.

اما به غیر از سنسورها، باید برای نمایش وضعیت Actuator هایی نظیر مه پاش، پمپ آبیاری گل‌ها، پمپ آبیاری سبزیجات، شیر برقی پر کننده‌ی منبع آب، و لامپ روشنایی، نیز محلی در نظر می‌گرفتیم لذا برای هرکدام از آن‌ها المانی ایجاد کرده و آن‌ها را در گوشه‌ی سمت راست و پایین تایل‌های مرتبطشان که به ترتیب تایل‌های humidity، SoilMoisture of flowers، SoilMoisture of vegetables، Water Level، و Photoresistor، هستند قرار دادیم.

که از میان این Actuatorها ۴ مورد اول از طریق فایل status.js و مورد آخر از طریق فایل valueN.js وضعیت خود را نشان می‌دهند.

- **پوشه‌ی view2** :

تمامی فایل‌های مربوط به این ویو مستقیما درون همین پوشه قرار دارند. و فایل‌های html و css آن به نام‌های mychart.html و mychart.css در دسترس هستند.

و برای هر تایل کد جاوااسکریپت به صورت مجزا نوشته شده، که شامل موارد زیر است :

- ✓ Fh1.js و fh2.js که مخفف Floor Humidity بوده و مقادیر مربوط به این سنسور را دریافت کرده و نمایش می‌دهند. (از سنسور باران به منظور سنسور رطوبت کف استفاده شده.)
- ✓ g.js که مخفف Gas بوده و مقادیر مربوط به این سنسور را دریافت کرده و نمایش می‌دهد.
- ✓ h.js که مخفف Humidity بوده و مقادیر مربوط به این سنسور را دریافت کرده و نمایش می‌دهد.

- ✓ Lbs.js که مخفف Light Bulb Status بوده و مقادیر مربوط به این سنسور را دریافت کرده و نمایش می‌دهد.
- ✓ md.js که مخفف motion detector بوده و مقادیر مربوط به این سنسور را دریافت کرده و نمایش می‌دهد.
- ✓ Phf.js که مخفف ph of flower بوده و مقادیر مربوط به این سنسور را دریافت کرده و نمایش می‌دهد.
- ✓ Phv.js که مخفف ph of vegetable بوده و مقادیر مربوط به این سنسور را دریافت کرده و نمایش می‌دهد.
- ✓ Phr.js که مخفف Photoresistor بوده و مقادیر مربوط به این سنسور را دریافت کرده و نمایش می‌دهد.
- ✓ Ps.js که مخفف Pump Status بوده و وضعیت پمپ را دریافت کرده و به صورت صفر و یک نمایش می‌دهد. (وضعیت ۳ پمپ مختلف مربوط به آبیاری گل‌ها ، سبزیجات و نازل مه پاش را به صورت multi channel نمایش می‌دهد.)
- ✓ Smf.js که مخفف Soil Moisture of Flower بوده و مقادیر مربوط به سنسور رطوبت خاکی که در گلدان‌های گل قرار دارند را دریافت کرده و نمایش می‌دهد.
- ✓ Smv.js که مخفف Soil Moisture of vegetable بوده و مقادیر مربوط به سنسور رطوبت خاکی که در باکس‌های سبزی قرار دارند را دریافت کرده و نمایش می‌دهد.
- ✓ t.js که مخفف Temperature بوده و مقادیر مربوط به این سنسور را دریافت کرده و نمایش می‌دهد.
- ✓ ts.js که مخفف Tap Status بوده و وضعیت شیر برقی را دریافت کرده و به صورت صفر و یک نمایش می‌دهد.(وضعیت شیر برقی که منبع آب را پر می‌کند.)
- ✓ wl.js که مخفف Water Level بوده و مقادیر مربوط به سنسور ultra sonic را دریافت کرده و نمایش می‌دهد.
- ✓ wm.js که مخفف Watt meter بوده و مقادیر مربوط به این سنسور را دریافت کرده و نمایش می‌دهد.

- **پوشه‌ی view4** :

تمامی فایل‌های مربوط به این ویو مستقیما درون همین پوشه قرار دارد، و فایل‌های html و css آن به نام‌های view4.html و view4.css در دسترس هستند.

و برای هر تایل کد جاوااسکریپت به صورت مجزا نوشته شده که عناوین هریک گویای ماهیت کار آن بوده و نیازی به معرفی ندارند.

سازوکار کدهای ویو ۱:

فایل value.html در پوشه‌ی view1، و فایل vlue.css در پوشه‌ی css این ویو، ساختار static ویوی ما را تشکیل می‌دهند.

و آنچه ویژگی‌های پویای این ویو را ایجاد کرده، همانطور که قبلا هم توضیح داده شد، شامل ۸ فایل جاوااسکریپت زیر است :

۱. Humidity.js
۲. averageOfSM-veg.js
۳. averageOfSM-flwr.js
۴. waterLevel.js
۵. averageOfPH-veg.js
۶. averageOfPH-flwr.js
۷. valueN.js
۸. status.js

سازوکار ۶ مورد اول کاملا مشابه است و با توضیح یک مورد سایر موارد نیز به همین توضیح قابل تعمیم خواهند بود. برای توضیح این سازوکار، فایل averageOfSM-veg.js انتخاب شده است؛و پس از آن به شرح سازوکار valueN.js و status.js خواهیم پرداخت.

- **averageOfSM-veg.js:**

کلیت کار به این صورت است که در هر ۳ ثانیه یک بار تابعی فراخوانی میشود که آرایه ای را درون objectی از کانال سنسور مورد نظر (که در اینجا Soil moisture است.) به ترتیب طی کرده و یک Read API ایجاد می‌کند که آخرین داده‌ی کانال مورد نظر را دریافت کند. و این API را به همراه اندیس عنصر مورد نظر به تابعی ارسال می‌کند که این تابع با یک اکس ام ال رکوئست (xml request) به این url (API) آخرین داده‌ی کانال را دریافت کرده و در آرایه soilmoistureArray میریزد که طول آن برابر با تعداد کانال‌های موجود برای آن سنسور خاص است.(مثلا در اینجا طول آن ۸ است.)

بعد از آن مقدار دریافتی از سرور با ترشولد تعیین شده در const smVegErrorValue مقایسه شده و اگر مقدار آن از کمترین میزان مجاز کمتر و یا از بیشترین مقدار مجاز بیشتر باشد اندیس مرتبط با آن در آرایه‌ی soilmoistureError (که در ابتدای برنامه و قبل اجرای هر تابعی تعریف و تمام اندیس های آن با false پر شده است) True می‌شود.

در نهایت هنگامیکه آرایه soilmoistureArray پر می‌شود تابع smVegCalculateAverage را فراخوانی کرده و این آرایه را برای محاسبه‌ی میانگین مقادیر دریافت شده به آن پاس می‌دهد.

در تابع smVegCalculateAverage میانگین مقادیر آرایه محاسبه و مقدار میانگین تحت عنوان average به تابع smVegUpdateSoilmoistureTile برای بروز رسانی تایل ارسال می‌گردد.

در تابع smVegUpdateSoilmoistureTile ابتدا مقدار میانگین در محلی که در فایل html برایش مشخص شده قرار می‌گیرد و سپس المان meter متناسب با آن مقدار پر می‌شود.

بعد از آن نوبت به تغییر تصویر پس‌زمینه می‌رسد. (تنها در این مورد با این روش عکس تغییر کرده و در سایر موارد روش ساده‌تری به کار رفته است.)

در نهایت تمامی عناصر آرایه‌ی soilmoistureError بررسی می‌شوند و چنانچه هر کدام از آن‌ها true باشند مقدار متغیر thereIsAtLeastOneDanger نیز true شده و در نتیجه‌ی آن علامت هشدار در محلی که برایش در نظر گرفته شده ظاهر خواهد شد.

- **valueN.js:**

فایل valueN.js در حقیقت وظیفه دارد که tileهای فتورزیستور، موشن (motion)، باران (رطوبت کف)، گاز، دما و وات متر از view اول را آپدیت کند. در آبجکت channel تمامی اطلاعات مربوط به چنل ها (id و apikey) آورده شده است تا در طول برنامه به راحتی در دسترس باشد. آبجکت errorValue در بردارنده ی مقادیر مجاز برای هر سنسور می‌باشد که اگر از این محدوده خارج بشود چراغ

خطر بالا سمت راست آن tile روشن می گردد. متغیر updateInterval نیز دربردارنده ی interval بین هر دو آپدیت متوالی صفحه به میلی ثانیه می‌باشد که در حالت دیفالت بر روی ۳۰۰۰ قرار دارد.

روند کلی آپدیت شدن به این صورت است که هر ۳ ثانیه در ابتدا تابع updateThePage فراخوانی می‌شود که در آن اطلاعات ذکر شده ی سنسور ها و وضعیت actuator لامپ از سرور گرفته می شود. در حقیقت آدرس مربوطه برای گرفتن اطلاعات به httpGetAsync پاس داده می‌شود و به ازای هر tile یک تابع که نام آن به Callback ختم می‌شود صدا زده می شود. به عنوان مثال اگر قرار است دما آپدیت بشود از داخل httpGetAsync تابع temperatureCallback با responseی که از سمت سرور گرفته شده فراخوانی می شود. در این تابع مقدار گرفته شده از سرور ذخیره می‌شود و اگر out of range اتفاق افتاده بود یک flag به عنوان نشانه true می‌شود تا بعداً برای روشن کردن چراغ خطر مورد استفاده قرار بگیرد.

سپس تابع calculateAverage فراخوانی می‌شود که در آن مقدار میانگین سنسور ها (متشکل از همه ی نود ها) محاسبه می‌شود و برای هر tile تابعی صدا زده می‌شود که نام آن با update شروع شده و به Tile ختم می شود. مثلاً برای دما تابع updateTemperatureTile صدا زده می شود. این تابع رابط فایل با html می باشد. Tileی که قرار است تغییر بکند انتخاب می‌شود و با توجه به رنجی که میانگین در آن وجود دارد تغییراتی حاصل می شود. همچنین لازم به ذکر است که جدای از مواردی که در بالا آورده شد، وات متر به دلیل نوع خاص گیجی که داشت به صورت مجزا از طریق تابع مربوط به خود آپدیت می شود.

- **status.js** :

همانطور که قبلا گفته شد این برنامه آخرین وضعیت Actuator ها را دریافت و با توجه به آن، المان (عکس) مربوط به آن را تغییر می‌دهد. (این تغییر بین دو وضعیت روشن و خاموش اتفاق می‌افتد و مقداری که از سرور دریافت میشود صفر یا یک است که صفر به معنای خاموش و ۱ به معنای روشن بودن آن Actuator خاص است.)

روند این برنامه نیز دقیقا مثل روند averageOfSM-veg.js است که بالاتر توضیح دادیم؛

در این برنامه نیز هر ۳ ثانیه یک بار دو فانکشن pumpStatusUpdateThePage و tapStatusUpdateThePage فراخوانی می‌شوند که در هریک از آن‌ها آرایه‌ی مربوطه طی شده و Read API برای دریافت آخرین داده از کانال مورد نظر ساخته شده و به توابع tapStatusHttpGetAsync و pumpStatusHttpGetAsync ارسال می‌شوند.

در هردوی آن توابع (tapStatusHttpGetAsync و pumpStatusHttpGetAsync) آخرین مقدار کانال مورد نظر دریافت و پس از آن که به تایپ Int تبدیل شد در متغیر num ذخیره شده و چنانچه این مقدار صفر باشد بسته به کانالی که این مقدار از آن دریافت شده، تصویر متناظر این Actuator تغییر می‌کند.

سازوکار کدهای view2 :

در این ویو برای ایجاد چارت از کتابخانه‌های آماده chart.js استفاده شده که برای اضافه کردن آن به پروژه آدرس CDN آن را در تگ script در هدر فایل html مربوط به آن اضافه کردیم.

سپس همان‌طور که قبلا هم توضیح داده شد، برای هر تایل کد جاوااسکریپت به صورت جداگانه نوشته شده که روند کار همه‌ی آن‌ها مشابه هم است و برای توضیح Smv.js را انتخاب می‌کنم؛

- Smv.js :

روند کلی این برنامه نیز مشابه سایر کدها است، به این صورت که در هر ۳ ثانیه یک بار، تابع smvUpdateThePage فراخوانی می‌شود که عناصر آرایه‌ی تعریف شده در smvChannel را طی کرده و متناسب با همان id و API Key یک Read API ساخته و آن را به تابع بعدی یعنی smvHttpGetAsync ارسال می‌کند.

اما همزمان با فراخوانی smvUpdateThePage تابع دیگری نیز به نام yAxisValues (که قرار است مقادیر محور افقی یعنی زمان را مشخص سازد) صدا زده می‌شود که یک Read API مربوط به یک کانال واحد (که در اینجا کانال ۷۰۶ می‌باشد) را هر ۳ ثانیه یکبار دریافت می‌کند و وظیفه‌ی آن این است که با توجه به APIای که به عنوان url دریافت می‌کند ۲۰ داده‌ی آخری که به کانال وارد شده را دریافت و از میان اطلاعت مربوط به این ۲۰ داده، زمان درج آن‌ها را استخراج و در یک آرایه به نام time بریزد.

اما برای دریافت مقادیر محور عمودی -که همان مقدار ارسالی به پلت‌فرم توسط سنسورها است -smvHttpGetAsync همزمان مشغول به کار است و اطلاعات ۲۰ مقدار آخر هر کانال را دریافت کرده و از میان این اطلاعات، فیلد۱ که مربوط به مقادیر است را استخراج و در آرایه vegSMArray می‌ریزد.

سپس با استفاده از یک حلقه ی تو در تو بر روی تمامی عناصر آرایه‌ی vegSMArray گردش کرده و چنانچه حتی یکی از مقادیر در هر کدام از کانال‌ها خارج از ترشولد تعیین شده باشد رنگ مربوط به چارت نماینده‌ی مقادیر آن کانال به نشانه‌ی هشدار به رنگ قرمز درمی‌آید.

vegSMArray یک آرایه دو بعدی با طولی برابر با آرایه‌ی مربوط به کانال‌هاست (طول آن برابر با تعداد کانال‌های تعریف شده برای سنسور رطوبت خاک است.) که هر کدام از عناصرش به تنهایی آرایه‌ای ۲۰ تایی هستند.

در کنار vegSMArray آرایه‌ی دیگری دقیقا با همان مشخصات به نام temp تعریف شده تا هر عنصر vegSMArray با عنصر متناظرش در temp مقایسه شود و اگر مقدار جدیدی دریافت شده بود، آنگاه تابع setValueForChart به منظور آماده و یکپارچه کردن مقادیر برای ساختن چارت فراخوانی می‌شود.

این کار به منظور جلوگیری از پرش و ریلود مکرر چارت‌ها انجام می‌شود به همین خاطر تا زمانیکه مقدار جدیدی از سرور دریافت نشود این تابع فراخوانی نمی‌شود.

در تابع setValueForChart آبجکتی به نام onechart ساخته می‌شود که این object شامل پراپرتی هایی است که برای کشیدن نمودار به آن‌ها نیاز داریم؛ یعنی labels و datasets ، که پراپرتی اول مقادیر محور افقی و پراپرتی دوم شامل آرایه‌ای از مقادیر محور عمودی و مشخصات ظاهری نمودارها همچون، رنگ و پهنای آن‌ها است.

سپس در مرحله آخر تابع smvUpdateTile را فراخوانی کرده و آبجکت onechart را به آن پاس می‌دهد. در این تابع کد های آماده‌ی chart.js دست به کار شده و مقادیری را که در قالب onechart گرفته شده، در محلی که در فایل html برایش تعیین شده به صورت یک نمودار نمایش می‌دهد.

چنانچه تنظیمات بیشتری در مورد نمودارها مد نظر باشد می‌توان در قسمت option آن‌ها را اضافه کرد.

ساز و کار کد های view3 :

همانطور که قبلا که گفته شد این ویو تلفیقی از دو ویوی قبلی یعنی چارت و گرافیک است به گونه‌ای که تایل‌های گرافیکی دور صفحه قرار گرفته و با کلیک بر روی هدر هریک از آن‌ها چارت مربوط به آن در تایل وسط به نمایش در می‌آید.

در این ویو برای پیاده‌سازی و ساختن اسکلت تایل‌ها از CSS Grid استفاده شده و سپس برای اجرای تایل‌های گرافیکی دقیقا از همان فایل‌های جاوااسکریپت ویو ۱ استفاده شده و برای پیاده‌سازی چارت نیز فایل chart.js نوشته شده که در آن فانکشن event listener منتظر کلیک بر روی هدر تایل گرافیکی است و با رخ دادن این اتفاق (کلیک کردن) همان برنامه‌ای که در فایل های جاوااسکریپت ویو ۲ بود شروع به کار می‌کند و نمودار مورد نظر را در item6 که برای قرارگیری چارت تعریف شده (با همان سازوکاری که بالاتر توضیح داده شد) نمایش می‌دهد.

سازوکار کدهای view4 :

این ویو نمایش‌گر ارور هریک از سنسورهای موجود می‌باشد که با Html, CSS و JavaScript کدنویسی شده و در آن از هیچ کتابخانه خاصی استفاده نشده است، به این شکل که هریک از انواع سنسورها داده های مربوط را گرفته، هر ۳ثانیه یک‌بار API Read به‌سمت سرور زده شده و داده‌ها را گرفته پاسخ دریافتی از سمت سرور درون یک آرایه ریخته می‌شود این مقایسه صورت می‌گیرد و بر اساس رنج تعریف شده رنگ دایره‌ها با استفاده از Id و APIkey سبز و زرد و قرمز را نشان می‌دهند. در این ویو داده-ها را با فرمت Json از سمت تینگ تاک گرفتیم و با فرمت Xml ریکوئست زدیم.

به طورکلی ۱۳ نوع سنسور مختلف داریم که برای هریک رنج موردنظر تعریف شده است:

۱. Soil Moisture

شامل ۱۸ سنسور می‌باشد که در تمامی گلدان‌ها وجود دارد. از صفر تا ۱۰ و از ۴۰ تا ۱۰۰ رنگ قرمز، از ۱۰ تا ۲۰ و از ۳۰ تا ۴۰ رنگ زرد و از ۲۰ تا ۳۰ را سبز نشان می‌دهد.

۲. Temperature

سنسور دما در هر تراس یک عدد قرار داده شده است. رنج آن برای ۱۰ تا ۲۰ و ۳۰ تا ۴۰ زرد، از صفر تا ۱۰ و از ۴۰ تا ۱۰۰ قرمز و از ۲۰ تا ۳۰ را سبز نشان می‌دهد.

۳. Humidity

سنسور رطوبت نیز همانند دما در هر تراس یک عدد قرار دارد و رنج آن همانند سنسور دما می‌باشد.

۴. Floor humidity

همانند سنسور رطوبت می‌باشد.

۵. Ph

شامل ۱۸ سنسور می‌باشد که در هر گلدون قرار می‌گیرد. از صفر تا ۳ و از ۱۱ تا ۱۴ را قرمز، از ۳تا ۶ و از ۸ تا ۱۱ زرد و ۶ تا ۸ را سبز نشان می‌دهد.

۶. Water level

شامل یک سنسور می‌باشد که اگر سطح آب از ۱۰ تا ۲۰ و ۸۰ تا ۹۹ بود قرمز، اگر صفر تا ۱۰ و ۹۹ تا ۱۰۰ بود زرد و ۲۰ تا ۸۰ را سبز نشان می‌دهد.

۷. Photo resistor

شامل دو سنسور در هر تراس می‌باشد که رنج آن همانند سنسور رطوبت است.

۸. Pump status

برای هریک از پمپ‌های گل و سبزی و مه‌پاش یک سنسور وضعیت پمپ داریم که در صورت روشن بودن سبز و در حالت خاموشی قرمز می‌شود.

۹. Light bulb status

در هرتراس یک عدد موجود است و مقدار آن صفر و یکی است.

۱۰. Gas

در هر تراس یک سنسور گاز موجود است. از ۵ تا ۹ و از ۲۶ تا ۵۰ زرد، از صفر تا ۵ و از ۵۰ تا ۱۰۰ قرمز و از ۹ تا ۲۶ سبز می‌باشد.

۱۱. Tap status

شامل یک سنسور است که مقادیر آن به صورت صفر و یک نمایش داده می‌شود.

۱۲. Motion detector

در هر تراس یک سنسور تشخیص حرکت به کار گذاشته شده است که مقادیر آن در صورت دیتکت شدن فرد ۱ و در صورت نبود آن صفر خواهد بود.

۱۳. Watt meter

برای این سنسور مقداری تعیین نشده است.