

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشکده مهندسی

گروه برق

پایان نامه کارشناسی

گرایش : الکترونیک

عنوان :

طراحی و ساخت نمونه آزمایشی آفتاب نگار

استاد راهنما : جناب آقای دکتر مهرداد بابازاده

نگارش : محمدرضا مصطفوی

تابستان ۱۴۰۰

تقدیر و تشکر

شکر شایان نثار ایزد منان که توفیق را رفیق راهم ساخت تا این پایان نامه را به پایان برسانم. از استاد فاضل و اندیشمند جناب آقای دکتر بابازاده به عنوان استاد راهنما که همواره اینجانب را مورد لطف و محبت خود قرار داده اند، کمال تشکر را دارم.

خدای را بسی شاکرم که از روی کرم، پدر و مادری فداکار نصیبم ساخته تا در سایه درخت پر بار وجودشان بیاسایم و از ریشه آنها شاخ و برگ گیرم و از سایه وجودشان در راه کسب علم و دانش تلاش نمایم. والدینی که بودنشان تاج افتخاری است بر سرم و نامشان دلیلی است بر بودنم، چرا که این دو وجود، پس از پروردگار، مایه هستی ام بوده اند دستم را گرفتند و راه رفتن را در این وادی زندگی پر از فراز و نشیب به من آموختند، آموزگاران که برایم زندگی، بودن و انسان بودن را معنا کردند. تقدیم به وجود با ارزششان.

فهرست مطالب

چکیده.....	۱
مقدمه.....	۲
فصل اول : کلیات تحقیق.....	۴
تاریخچه آفتاب نگار [۱].....	۴
مزایا و معایب آفتاب نگار کمپبل استوکس.....	۶
آفتاب نگار الکتریکی [۲].....	۷
طرح پیشنهادی.....	۸
فصل دوم : تجهیزات مورد استفاده.....	۱۰
بدنه طرح پیشنهادی.....	۱۰
تجهیزات الکترونیکی.....	۱۵
فوتورزیستور.....	۱۵
ماژول مالتی پلکسر ۱۶ کاناله.....	۱۶
ماژول کارت حافظه.....	۱۷
ماژول ساعت DS۳۲۳۱.....	۱۹
آردوینو پرو مینی.....	۲۰
فصل سوم : روش تحقیق.....	۲۴
اتصالات الکتریکی.....	۲۴
عملکرد کلی پروژه.....	۲۷
برنامه نویسی.....	۲۸

حلقه های اصلی.....	۲۸
انتخاب برد و پورت.....	۲۸
سریال مانیتور.....	۳۰
کتابخانه در آردوینو.....	۳۱
برنامه نوشته شده برای آفتاب نگار طراحی شده.....	۳۲
فصل چهارم : یافته های تحقیق.....	۴۳
نتیجه گیری.....	۴۶
پیشنهادهات.....	۵۱
مراجع.....	۵۳

فهرست تصاویر

- تصویر ۱- آفتاب نگار کمپبل استوکس [۱]..... ۵
- تصویر ۲- نوارهای استفاده شده در آفتاب نگار کمپبل استوکس..... ۶
- تصویر ۳- پیرانومتر [۲]..... ۸
- تصویر ۴- نمای کلی آفتاب نگار طراحی شده..... ۹
- تصویر ۵- اجزای بدنه آفتاب نگار طراحی شده..... ۱۲
- تصویر ۶- نمای داخلی قطعه شماره ۱..... ۱۳
- تصویر ۷- نمای بالای قطعه شماره ۲..... ۱۴
- تصویر ۸- نمایی از قطعه شماره ۳..... ۱۴
- تصویر ۹- فوتورزیستور..... ۱۵
- تصویر ۱۰- مازول مالتی پلکسر ۱۶ کاناله..... ۱۷
- تصویر ۱۱- مازول کارت حافظه SD Card..... ۱۸
- تصویر ۱۲- مازول ساعت DS۳۲۳۱..... ۱۹
- تصویر ۱۳- آردوینو پرو مینی..... ۲۲
- تصویر ۱۴- شماتیک آردوینو پرو مینی [۴]..... ۲۳
- تصویر ۱۵- شماتیک اتصالات مدار طراحی شده..... ۲۵
- تصویر ۱۶- نمایی دیگر از اتصالات پردازنده و مازولها..... ۲۵
- تصویر ۱۷- نحوه اتصال سنسورها و مقاومتها..... ۲۶
- تصویر ۱۸- اتصال مالتی پلکسر به خروجی سنسورها..... ۲۶
- تصویر ۱۹- تکمیل و جداسازی بخش مربوط به سنسورها..... ۲۷
- تصویر ۲۰- انتخاب برد آردوینو..... ۲۹
- تصویر ۲۱- انتخاب پورت آردوینو..... ۲۹
- تصویر ۲۲- سریال مانیتور آردوینو..... ۳۰
- تصویر ۲۳- بخش اول برنامه نوشته شده..... ۳۲
- تصویر ۲۴- بخش دوم برنامه نوشته شده..... ۳۳
- تصویر ۲۵- بخش سوم برنامه نوشته شده..... ۳۴
- تصویر ۲۶- بخش چهارم برنامه نوشته شده..... ۳۵

- تصویر ۲۷- بخش پنجم برنامه نوشته شده..... ۳۶
- تصویر ۲۸- بخش ششم برنامه نوشته شده..... ۳۷
- تصویر ۲۹- بخش هفتم برنامه نوشته شده..... ۳۸
- تصویر ۳۰- بخش هشتم برنامه نوشته شده..... ۳۹
- تصویر ۳۱- بخش نهم برنامه نوشته شده..... ۴۰
- تصویر ۳۲- بخش دهم برنامه نوشته شده..... ۴۱
- تصویر ۳۳- بخش یازدهم برنامه نوشته شده..... ۴۲
- تصویر ۳۴- نتایج نهایی ۱۶ سنسور..... ۴۳
- تصویر ۳۵- نمودار مربوط به سنسورهای ۷ و ۸ و ۱۵ و ۱۶..... ۴۴
- تصویر ۳۶-نمونه‌های از اطلاعات نمایش داده شده در سریال مانیتور..... ۴۵
- تصویر ۳۷- نمودار مربوط به سنسورهای ۷ و ۸..... ۴۷

چکیده

با توجه به رو به اتمام بودن منابع انرژی تجدید ناپذیر مانند نفت و گاز و مشکلاتی که استفاده از این انرژی‌ها به وجود آورده است مانند تغییرات اقلیم، آلودگی هوا و تخریب محیط زیست، استفاده از انواع انرژی تجدیدپذیر ضروری می‌باشد. انرژی خورشیدی می‌تواند به طور مستقیم از خورشید و حتی در آب و هوای ابری به دست آید. انرژی خورشیدی، در سطح جهان و به صورت گسترده برای تولید برق، گرما و شیرین سازی آب مورد استفاده قرار می‌گیرد.

برای اندازه‌گیری میزان تابش خورشید از دستگاه‌هایی به نام آفتاب نگار استفاده می‌شود. این دستگاه‌ها از گذشته تا کنون شامل تغییرات زیادی شده‌اند بگونه‌ای که از شکل یک کره‌ی شیشه‌ای به یک دستگاه الکترونیکی شامل سنسور رسیده است.

آفتاب نگار طراحی شده شامل ۱۶ سنسور فوتورزیستور است که دور یک حجم گنبدی شکل قرار داده شده‌اند. اطلاعات دریافتی از سنسورها با زمان وقوع آنها، از طریق ماژول‌ها و قطعات مورد نیاز درون یک کارت حافظه ذخیره می‌شوند و همچنین به صورت لحظه‌ای نمایش داده می‌شوند.

با بررسی اطلاعات بدست آمده متوجه دقت مناسب سیستم طراحی شده می‌شویم، به‌طوری که می‌توان تحلیل‌های اولیه راجع به وضعیت آب و هوا و میزان روشنایی هوا در روز آزمایش و همچنین مسیر حرکت خورشید را ارائه نمود.

مقدمه

گسترده‌گی نیاز انسان به منابع انرژی همواره از مسائل اساسی مهم در زندگی بشر بوده و تلاش برای دستیابی به یک منبع تمام نشدنی انرژی از آرزوهای دیرینه انسان بوده است. به دلیل افزایش روزافزون نیاز به انرژی و محدودیت منابع فسیلی از یک سو و افزایش آلودگی محیط زیست ناشی از سوزاندن این منابع از سوی دیگر، استفاده از انرژی‌های تجدیدپذیر را روز به روز با اهمیت‌تر و گسترده‌تر نموده است.

انرژی خورشیدی، نور و گرمای تولید شده توسط خورشید است. انرژی ستاره خورشید یکی از منابع عمده انرژی در منظومه شمسی می‌باشد. طبق آخرین برآوردهای رسمی اعلام شده عمر این انرژی بیش از ۱۴ میلیارد سال می‌باشد. در هر ثانیه ۲.۴ میلیون تن از جرم خورشید به انرژی تبدیل می‌شود. با توجه به وزن خورشید که حدود ۳۳۳ هزار برابر وزن زمین است، این کره نورانی را می‌توان به عنوان منبع عظیم انرژی تا ۵ میلیارد سال آینده به حساب آورد.

سال‌های زیادی است که اکثر صنعت جهان روی سوخت‌های فسیلی (زغال سنگ، نفت و گاز طبیعی) پایدار است. سوخت‌های فسیلی منابع تجدید ناپذیر هستند، یعنی به وجود آمدن آنها نیازمند سال‌های زیادی است. بنابراین این سوخت‌ها مدت زیادی نمی‌توانند جوابگوی تمام نیازهای انسان باشند و یک روز تمام خواهند شد. پس باید جایگزین مناسبی برای این سوخت‌ها پیدا شود، که انرژی خورشیدی می‌تواند جایگزین بسیار مناسبی به خصوص برای نسل‌های آینده باشد. بعضی از انرژی‌ها تأثیر بد و نامطلوبی بر محیط خواهند داشت،

مانند سوخت‌های فسیلی. با بهره‌گیری از انرژی خورشیدی، دیگر جای نگرانی نخواهد بود زیرا اگر منابع انرژی موجود به اتمام برسند، انرژی تجدیدپذیر و دائمی مانند انرژی خورشیدی را تا ابد خواهیم داشت و تأثیرات مخرب زیست محیطی و ایمنی از نظر سلامتی، بشر را تهدید نخواهد کرد. همچنین خورشید برای زمینه‌های مختلف به عنوان یک منبع گرمایی و انرژی استفاده می‌شود. پرورش محصولات کشاورزی، گرم کردن، خشک کردن مواد غذایی، ذخیره کردن انرژی در صفحات خورشیدی، نمونه‌ای از این کاربردها هستند.

اولین قدم در راستای استفاده از انرژی‌های تجدیدپذیر پس از شناخت آنها، توانایی اندازه‌گیری و همچنین انتخاب محل مناسب جهت بهره‌برداری حداکثری از آنها است. امروزه ابزار اندازه‌گیری مختلفی برای اندازه‌گیری این انرژی‌ها موجود است که یکی از آنها آفتاب نگار^۱ است، که میزان تابش آفتاب را در هر زمان و در یک مکان یا منطقه خاص اندازه‌گیری و ثبت می‌کند.

^۱ Sunshine Recorder

فصل اول : کلیات تحقیق

تاریخچه آفتاب نگار^[۱]

آفتاب نگار کمپبل استوکس^۱، یکی از انواع آفتاب نگار است. این وسیله نخستین بار توسط جان فرانسیس کمپبل^۲ در سال ۱۸۵۳ اختراع و توسط جرج گابریل استوکس^۳ در سال ۱۸۷۹ اصلاح شد. طرح اولیه که توسط کمپبل ارائه شده بود؛ متشکل از یک کره شیشه‌ای بود که درون یک کاسه چوبی قرار گرفته و خورشید با سوزاندن روی کاسه اثر می‌گذاشت. اصلاح استوکس این بود که نگهدارنده‌ی کره را از فلز ساخت و یک نگهدارنده نوار را در پشت کره قرار داد تا با سوزانده شدن نوارها بتوان به مدت زمانی که خورشید در آسمان تابیده است، پی برد. این دستگاه با ایجاد رد سوختگی روی نوار، مدت تابش آفتاب را ثبت می‌کند.

این وسیله تشکیل شده از یک گوی شیشه‌ای که عملکردی مانند ذره‌بین دارد و پرتو آفتاب را روی صفحه‌ای که در پشت آن قرار دارد متمرکز می‌کند. محفظه شیشه‌ای کروی محتوی آب است که خود محفظه در داخل یک کاسه چوبی قرار دارد، محفظه پرتوهای نورانی خورشید را شبیه به یک عدسی جمع کرده و روی جدار داخلی کاسه چوبی می‌اندازد. چوب (یا نوار آفتاب نگار) در اثر پرتو متمرکز شده و گرمای به دست آمده از خورشید سوخته و به تدریج خطی روی آن کشیده می‌شود.

^۱ Campbell–Stokes recorder

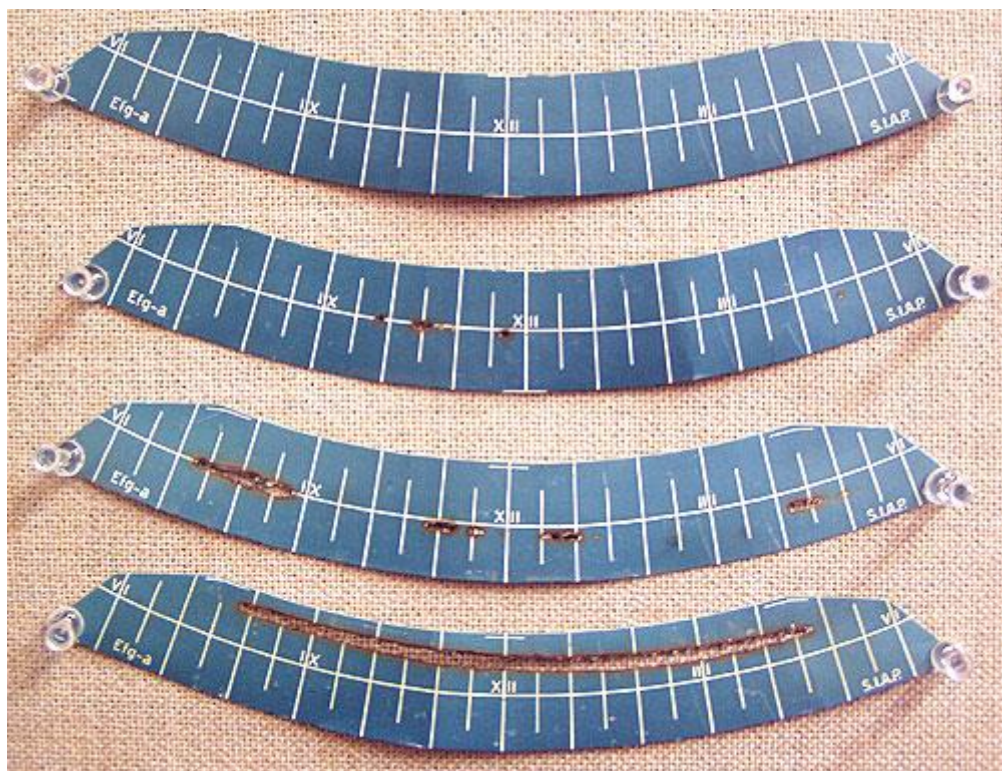
^۲ John Francis Campbell

^۳ George Gabriel Stokes

این وسیله رو به جنوب و به صورت کاملاً تراز افقی و در ارتفاع ۱.۵ متری نصب شده و سپس متناسب با عرض جغرافیایی محل نصب تنظیم می‌گردد. با توجه به این که در طول سال زاویه تابش خورشید و طول روز تغییر می‌کند، از گراف‌های متفاوتی استفاده می‌شود.



تصویر ۱- آفتاب نگار کمپیل استوکس [۱]



تصویر ۲- نوارهای استفاده شده در آفتاب نگار کمپبل استوکس

مزایا و معایب آفتاب نگار کمپبل استوکس

عمده‌ترین مزیت این نوع آفتاب نگار، سادگی و سهولت استفاده از آن است. هیچ قسمت متحرکی ندارد، بنابراین به تعمیر و نگهداری بسیار کمی نیاز دارد. این آفتاب نگار را می‌توان در هر کجای دنیا با کمترین تغییر مورد استفاده قرار داد. همچنین این نوع آفتاب نگار فقط از انرژی خورشیدی استفاده می‌کند.

هنگامی که تابش خورشید ضعیف باشد، ممکن است قدرت کافی برای سوزاندن مناسب نوار آفتاب نگار فراهم نشود. این اتفاق اغلب هنگام سحر و غروب رخ می‌دهد. ممکن است باران موجب خیس شدن نوار شود و هنگام برداشتن پاره شود و در نتیجه خواندن آن با

مشکل مواجه شود. این نوع آفتاب نگار همچنین به عوامل خارجی دیگری همچون گرد و غبار، آلودگی یا فضولات خشک شده پرندگان روی کره شیشه‌ای و ... حساس است که نیاز به بازرسی و تمیز کردن مکرر دارد.

آفتاب نگار الکتریکی [۲]

همانطور که اشاره شد هنگام استفاده از آفتاب نگار کمپبل استوکس، نیاز است که یک فرد هر روز نوار استفاده شده را برداشته و تعویض نماید. ممکن است خواندن این نوارها بنا به دلایل گفته شده در قسمت قبل، با مشکل مواجه شود. همچنین ذخیره کردن تعداد بالای این نوارها و دسترسی دوباره به آنها کار مشکلی خواهد بود که نیازمند صرف زمان و هزینه اضافی است. برای حل این مشکلات می‌توان از آفتاب نگارهای الکتریکی استفاده نمود. این نوع آفتاب نگارها برخلاف آفتاب نگارهای معمولی که فقط با انرژی خورشیدی کار می‌کردند، به یک منبع تغذیه خارجی نیاز دارند.

از جمله این نوع آفتاب نگارها که امروزه مورد استفاده قرار می‌گیرد می‌توان به پیرانومتر^۱ اشاره نمود. بطور کلی از پیرانومترها بیشتر برای تحقیقات اقلیمی و اهداف نظارت بر آب و هوا استفاده می‌شد، اما علاقه اخیر جهانیان به انرژی خورشیدی، باعث افزایش علاقه به پیرانومترها شده است.

^۱ pyranometer



تصویر ۳- پیرانومتر [۲]

طرح پیشنهادی

همانطور که در بخش‌های قبل ذکر شد، آفتاب نگارهای الکتریکی مزیت‌های بیشتری نسبت به آفتاب نگارهای معمولی دارند و همچنین قابلیت‌های بیشتری نیز می‌توان به آنها اضافه نمود. طرحی که در این مطلب به آن پرداخته خواهد شد هم در مجموعه آفتاب نگارهای الکتریکی قرار می‌گیرد. مهم‌ترین مزیتی که این طرح را از آفتاب نگارهای الکتریکی موجود متمایز می‌کند، پایین بودن هزینه تمام شده این طرح نسبت به مدل‌های دیگر است. در این طرح سعی بر این بوده است که از سنسورها و قطعات الکترونیکی‌ای استفاده شود که علاوه بر

داشتن قیمت پایین، به آسانی قابل دسترس بوده و همچنین از دقت مناسبی نیز برخوردار باشند. امکان به کارگیری این طرح در تمامی نقاط کره زمین، از جمله اهداف دیگر این طرح است.



تصویر ۴- نمای کلی آفتاب نگار طراحی شده

فصل دوم : تجهیزات مورد استفاده

بدنه طرح پیشنهادی

آفتاب نگار طراحی شده از دو بخش مجزا تشکیل شده است، بخش مربوط به قرارگیری و اتصالات سنسورها و بخش مربوط به تجهیزات الکترونیکی که کار پردازش و ذخیره اطلاعات را برعهده دارند. از دلایل جداسازی این دو بخش می‌توان به کم کردن اتصالات نهایی به پردازنده اصلی و همچنین ایجاد سهولت در تعویض یا تعمیر هر یک از بخش‌ها بدون نیاز به تغییر در بخش دیگری اشاره نمود.

ابتدا بدنه در نرم افزار سالی‌دورک^۱ طراحی و سپس با استفاده از پرینترهای سه‌بعدی، ساخته شد. برای ساخت بدنه از فیلامنت^۲های PLA^۳ که رایج ترین نوع فیلامنت مصرفی در پرینترهای سه‌بعدی است، استفاده شده است. PLA، نوعی ترموپلاست است که از مواد ارگانیک نظیر ساقه ذرت و نیشکر تولید می‌شود. این بنیان طبیعی PLA باعث شده تا موقع پرینت هیچگونه بوی بدی تولید نکرده، بخار آن سمی نبوده و در پایان با راحتی بیشتری قابل بازیافت باشد. پرینت PLA به مراتب از دیگر متریال‌های فیلامنتی راحت‌تر است و این نکته مهم‌ترین دلیل حضور پررنگ این متریال در بازار مواد مصرفی است. از جمله دلایل دیگر که باعث رایج شدن این نوع فیلامنت شده است، قیمت کمتر و همچنین چاپ آسان و بی دردسر آن نسبت به انواع دیگر فیلامنت‌ها است.

^۱ Solidwork

^۲ Filament

^۳ Poly Lactic Acid

فیلامنت ABS^۱ یکی دیگر از مواد مصرفی رایج در بازار پرینترهای سه‌بعدی است. ABS در عین ارزان‌تر بودن نسبت به بقیه متریال‌ها، دارای استحکام خوبی بوده و نسبت به فیلامنت PLA منعطف‌تر است. همین امر باعث شده تا در استفاده نهایی و قرارگیری در کالاهای مصرفی نقش بیشتری داشته باشد. دو موضوع مهم وجود دارد که بسیاری از کاربران را از مصرف این نوع فیلامنت باز می‌دارد. یکی سمی بودن بخار ناشی از ABS در حین چاپ و دیگری فرایند چاپ آن، که معمولاً در مقایسه با PLA کمی چالشی‌تر بوده و نیازمند تجربه است. شایان ذکر است که حلال ABS، استون^۲ می‌باشد و می‌توان از این خاصیت حلالی جهت پرداخت و پولیش سازه‌های چاپ شده استفاده کرد. در حالی که حلال PLA ماده سمی کلروفرم^۳ است. این که می‌بینیم در بازار مواد مصرفی، معمولاً PLA پرتعدادتر از ABS است، به دلیل استفاده راحت‌تر و متنوع‌تر بودن PLA است. تقریباً همه پرینترهای سه‌بعدی ساخته شده در دنیا می‌توانند PLA را پرینت کنند اما این موضوع برای ABS صادق نیست. [۳]

قطعات پرینت شده این طرح از جنس فیلامنت PLA سفید رنگ، در تصویر ۴ به نمایش درآمده است.

^۱ Acrylonitrile Butadiene Styrene

^۲ Acetone

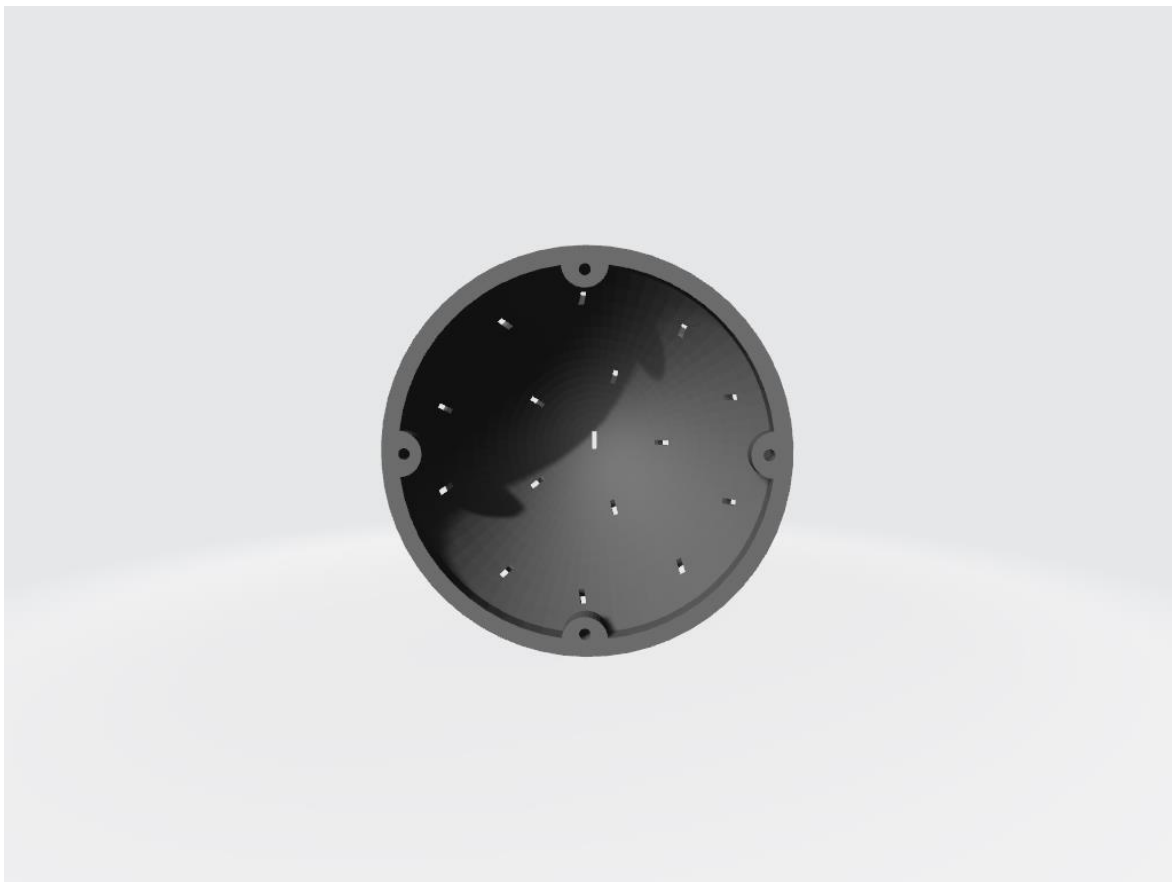
^۳ Chloroform



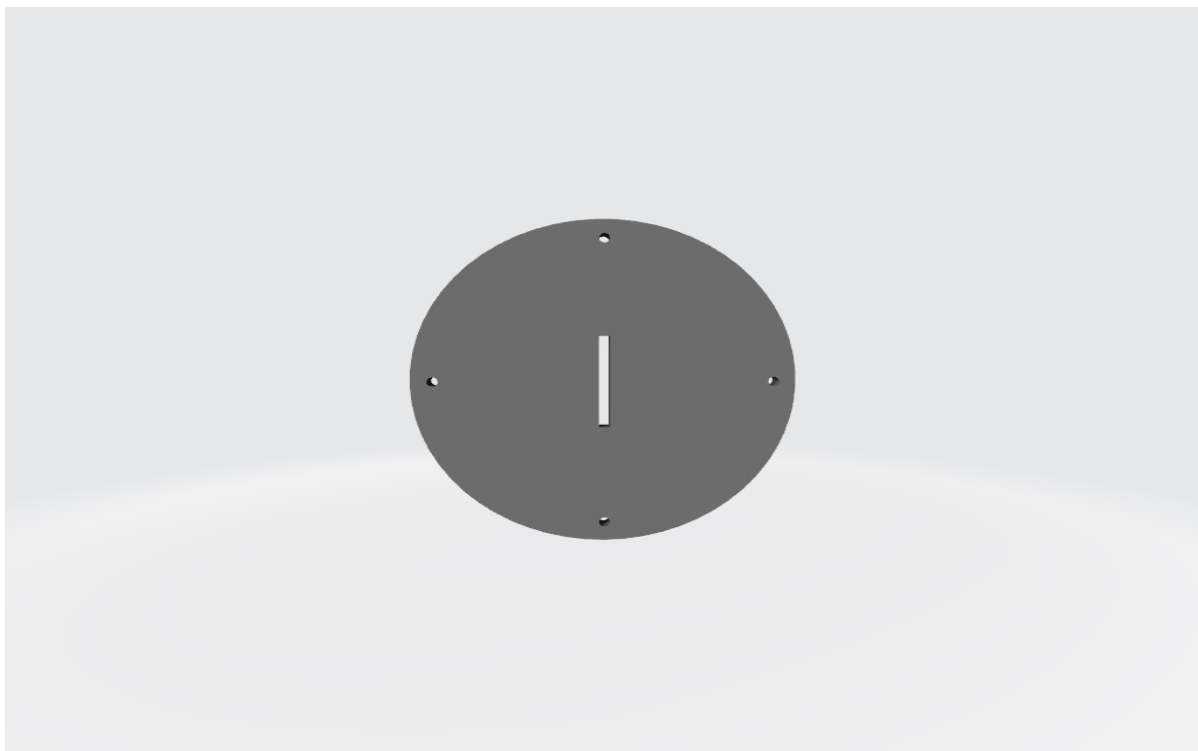
تصویر ۵- اجزای بدنه آفتاب نگار طراحی شده

با توجه به تصویر ۵، قطعه‌ای که با شماره ۱ مشخص شده است مربوط به بخش قرارگیری و اتصالات سنسورها است. روی این قطعه ۱۶ حفره برای قرارگیری سنسورها در داخل آنها وجود دارد که یکی از این حفره‌ها در رأس این گنبد و مابقی دور تا دور آن ایجاد شده‌اند که با یکدیگر ۲۴ درجه فاصله دارند. همانطور که مشاهده می‌شود این قطعه به صورت سهمی‌گون است، درحالی که اگر بصورت نیم کره طراحی می‌گردید ارتفاع قطعه پایین آمده و سنسورها به هم نزدیک‌تر می‌شدند و در نتیجه تفکیک اطلاعات آنها دشوار می‌شد و برای زیاد کردن فاصله بین آنها باید قطر این نیم کره افزایش پیدا میکرد که نتیجه‌ای جز افزایش حجم کلی دستگاه نداشت. قطعه‌ای که با شماره ۲ مشخص شده است، به زیر قطعه شماره ۱

متصل می‌شود تا آن را از قطعه شماره ۳ جدا کند. در این قطعه یک حفره‌ای وجود دارد تا از آنجا ۸ پایه خارج شود که بتوان اطلاعات سنسورها را توسط آنها دریافت کرد. همچنین این قطعه وظیفه اتصال قطعه شماره ۱ با قطعه شماره ۳ را به عهده دارد. قطعه‌ای که با شماره ۳ مشخص شده است، محل قرارگیری تجهیزات الکترونیکی و ذخیره کننده اطلاعات است.



تصویر ۶- نمای داخلی قطعه شماره ۱



تصویر ۷- نمای بالای قطعه شماره ۲

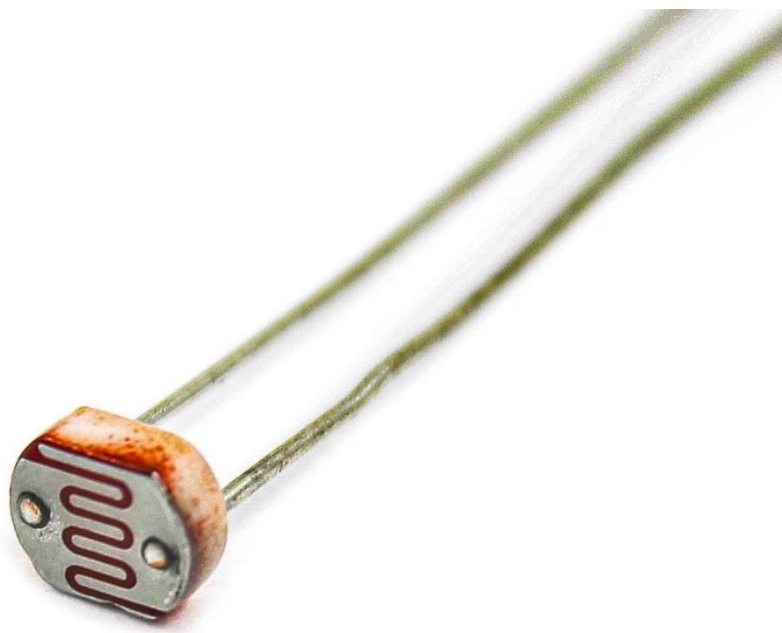


تصویر ۸- نمایی از قطعه شماره ۳

تجهیزات الکترونیکی

فوتورزیستور^۱

سنسورهای فوتورزیستور نوعی از مقاومت‌ها هستند. همانگونه که از اسم آن می‌توان فهمید، به نور حساس است و با تغییر میزان نور مقدار مقاومت نیز تغییر می‌کند. با روشن‌تر شدن محیط اطراف فوتورزیستور، مقدار مقاومت کاهش یافته و با کاهش نور مقدار مقاومت افزایش می‌یابد. مشخصه‌ی خروجی این سنسور به صورت غیرخطی است.



تصویر ۹- فوتورزیستور

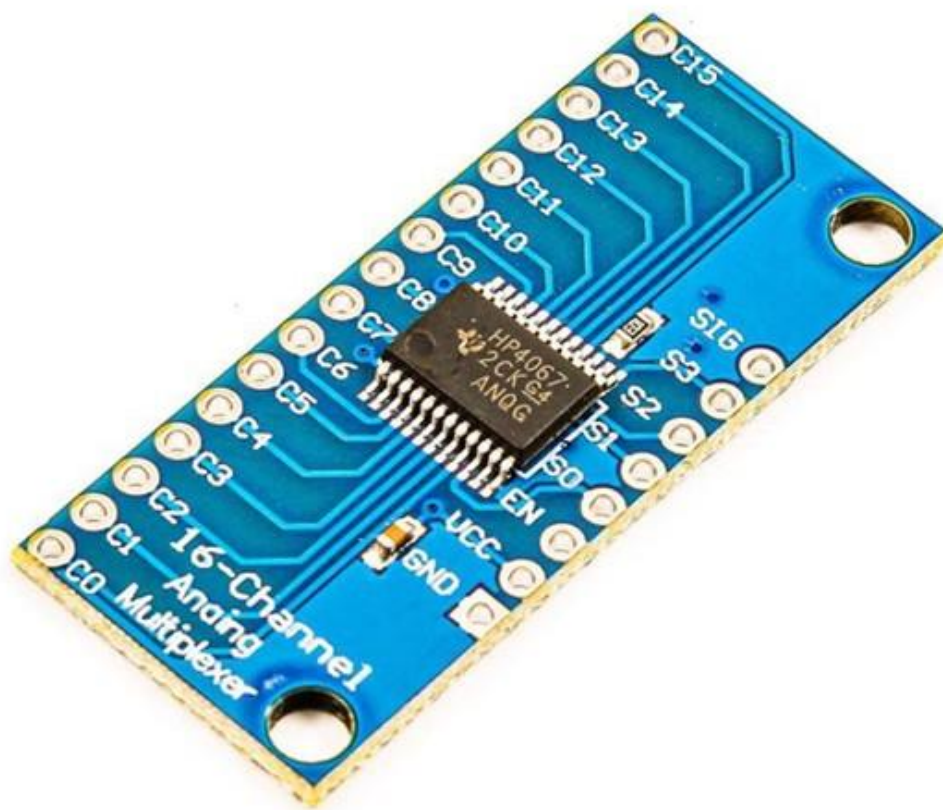
^۱ Photoresistor

ماژول مالتی پلکسر ۱۶ کاناله

از ماژول مالتی پلکسر ۱۶ کاناله CD74HC4067 می توان برای زمانی که تعداد زیادی ورودی آنالوگ در یک مدار وجود دارد، استفاده کرد. در این حالت نیاز است در هر زمان یکی از آن ورودی ها انتخاب شده و پردازش شوند. از این مالتی پلکسر می توان برای انتخاب از بین ۱۶ ورودی آنالوگ استفاده کرد. برای این منظور ۴ پایه S0-3 تعبیه شده اند که با دادن مقادیر مناسب به آنها یکی از ورودی های آنالوگ انتخاب شده و در پایه SIG به عنوان خروجی قرار می گیرد.

ماژول مالتی پلکسر ۱۶ کاناله CD74HC4067 دارای ۲۴ پایه به شرح زیر است:

- VCC : تغذیه ماژول (۵ ولت)
- GND : زمین
- EN : پایه فعال ساز (این پایه Active Low است).
- S0-3 : انتخاب یکی از ۱۶ پایه ورودی آنالوگ به عنوان سیگنال خروجی نهایی
- C0-15 : ورودی های آنالوگ
- SIG : سیگنال خروجی



تصویر ۱۰- ماژول مالتی پلکسر ۱۶ کاناله

ماژول کارت حافظه

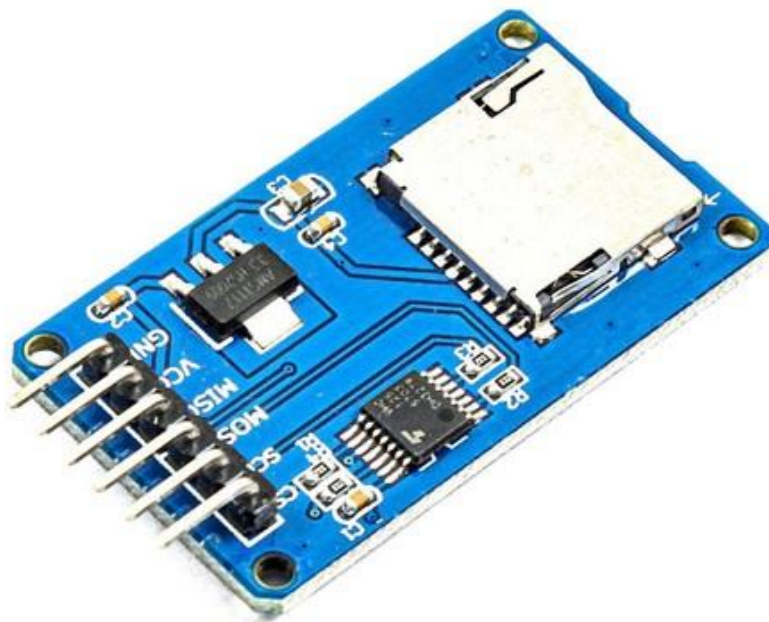
با استفاده از ماژول کارت حافظه می‌توان با کارت حافظه ارتباط برقرار کرد و اطلاعات مورد نظر را از آن خواند و یا بر روی آن نوشت. ارتباط این ماژول از طریق پروتکل ^۱SPI می‌باشد. حداکثر ظرفیت قابل شناسایی این ماژول برای کارت حافظه مدل SD، دو گیگابایت و برای مدل Micro-SD، شانزده گیگابایت است. همچنین ماژول کارت حافظه قابلیت استفاده

^۱ Serial Peripheral Interface

برای کارتهای Micro-SD و Micro-SDHC را دارد و دارای یک رگولاتور ۳.۳ ولت است که ولتاژ پایه‌ها برای اتصال به کارت حافظه را به ۳.۳ ولت کاهش می‌دهد.

این ماژول دارای ۶ پایه به شرح زیر است:

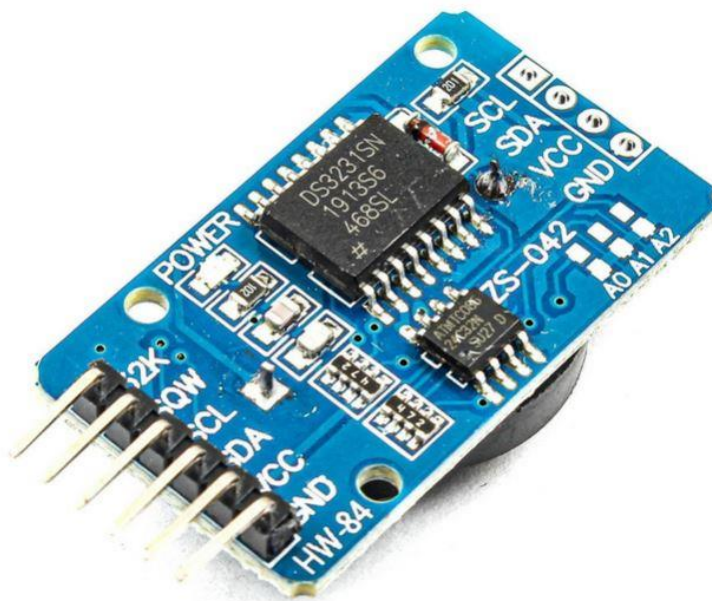
- VCC : تغذیه ماژول (۵ ولت)
- GND : زمین
- CS : پایه ChipSelect
- MISO : پایه فرستنده – معادل پایه Tx پورت سریال
- MOSI : پایه گیرنده – معادل پایه Rx پورت سریال
- SCK : پایه کلاک ماژول



تصویر ۱۱- ماژول کارت حافظه SD Card

ماژول ساعت DS3231

ولتاژ کاری این ماژول ۳.۳ الی ۵ ولت بوده و از باتری پشتیبان و جریان مصرفی بسیار پایین استفاده می‌کند. باتری پشتیبان قرار گرفته بر روی این ماژول زمانی که برق در دسترس نیست، از عقب افتادن زمان جلوگیری خواهد کرد. منبع کلاک این ماژول از یک کریستال ساعت به فرکانس ۳۲.۷۶۸ کیلوهرتز استفاده می‌کند. این ماژول قابل کارکرد در دو حالت ۲۴ ساعته و ۱۲ ساعته را دارد. همچنین علاوه بر ساعت بصورت ثانیه، دقیقه، ساعت امکان ذخیره تاریخ بصورت روز، هفته، ماه، سال را نیز دارد.



تصویر ۱۲- ماژول ساعت DS3231

آردوینو پرو مینی^۱

آردوینو، یک پلتفرم متن باز است که در ساخت پروژه‌های الکترونیکی مورد استفاده قرار می‌گیرد. با توجه به طراحی متن باز این پلتفرم، نرم‌افزار و طرح سخت‌افزار آن به صورت آزاد در اختیار تمام افراد قرار گرفته و افراد می‌توانند به وسیله آن پروژه‌های الکترونیکی خود را به سادگی انجام دهند. این ویژگی پلتفرم آردوینو باعث شده تا افراد زیادی در سراسر جهان از این پلتفرم استفاده کنند و روز به روز بیشتر آن را توسعه دهند. مزیتی که به واسطه رایگان بودن پلتفرم آردوینو به وجود آمده این است که معمولاً هر کسی در هر کجای جهان یک مدار مکمل به نام شیلد، برای انجام یک کار خاص در آردوینو طراحی می‌کند و معمولاً کتابخانه‌ای برای این شیلد می‌نویسد و اغلب اوقات این کتابخانه‌ها به صورت رایگان با سایر افراد به اشتراک گذاشته می‌شود. این ویژگی منحصر به فرد باعث شده تا هر مازول، سنسور یا قطعه‌ای قابلیت اتصال به آردوینو را داشته و بتوان صدها کتابخانه برای آن پیدا کرد.

آردوینو شامل یک مدار قابل برنامه‌ریزی فیزیکی (اغلب به عنوان یک میکروکنترلر) و یک قطعه نرم‌افزاری یا محیط یکپارچه توسعه (IDE) است که بر روی کامپیوتر اجرا می‌شود. IDE^۲ محیطی عمدتاً گرافیکی است که تمام یا شماری از ابزارهای لازم برای توسعه نرم‌افزار را در خود دارد. در IDE، دسترسی به ابزارها و اعمال آنها در پروژه جاری تسهیل شده است. برد های آردوینو در بین افرادی که در حوزه الکترونیک فعالیت می‌کنند بسیار محبوب است.

^۱ Arduino Pro Mini

^۲ Integrated Development Environment

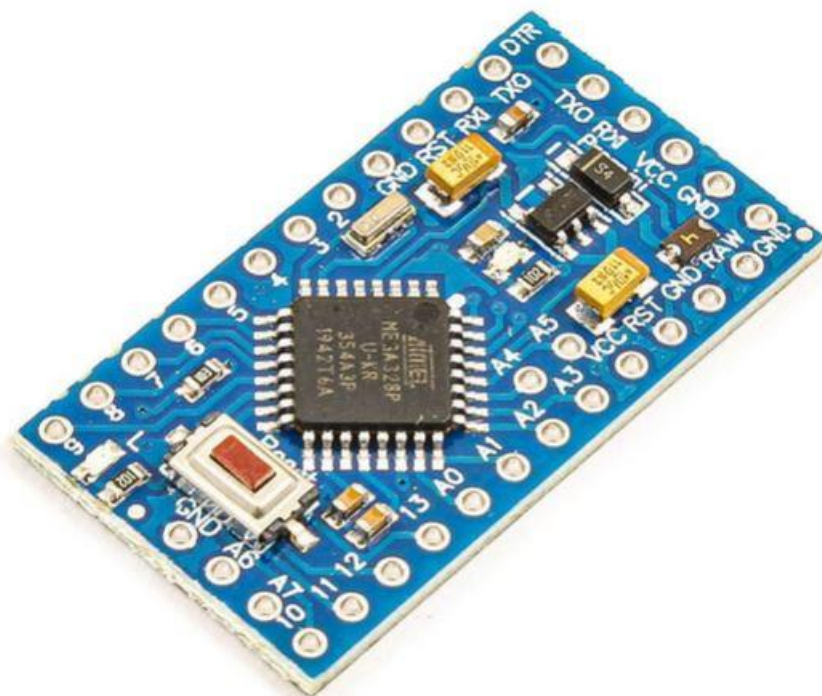
به این دلیل که برخلاف بسیاری از بردهای قابل برنامه‌ریزی، آردوینو برای بارگذاری کدهای جدید بر روی برد، به سخت‌افزار جداگانه نیاز ندارد و می‌توان به راحتی با استفاده از یک کابل USB این کار را انجام داد.

آردوینو پرو مینی یک برد آردوینو بر پایه میکروکنترلر ATmega328 می‌باشد و به لحاظ کارایی و عملکرد کاملاً مشابه با سایر بردهایی که از این نوع میکروکنترلر استفاده می‌کنند بوده و تنها تفاوت این برد در شکل فیزیکی آن می‌باشد. در این برد به منظور کاهش سایز ورودی آداپتور و پورت USB (که در اکثر پروژه‌ها کاربرد دائمی نداشته) حذف شده است. در نتیجه برای انجام پروژه‌هایی با سایز کوچک و همین‌طور در ربات‌های پرنده و یا هواپیماهای مدل که وزن و اندازه اجزای آن بسیار حائز اهمیت است، توصیه می‌گردد. از طرفی حذف پورت USB در این مدل برد موجب شده تا نتوان به سادگی سایر بردهای آردوینو، کدها را بر روی برد آپلود نمود و برای این کار لازم است کد را از طریق کابل‌ها و مبدل‌های USB به سریال و یا از طریق یک برد آردوینو دیگر که دارای رابط USB می‌باشد به عنوان پروگرامر استفاده نموده و کدها را درون میکروی برد آپلود کرد.

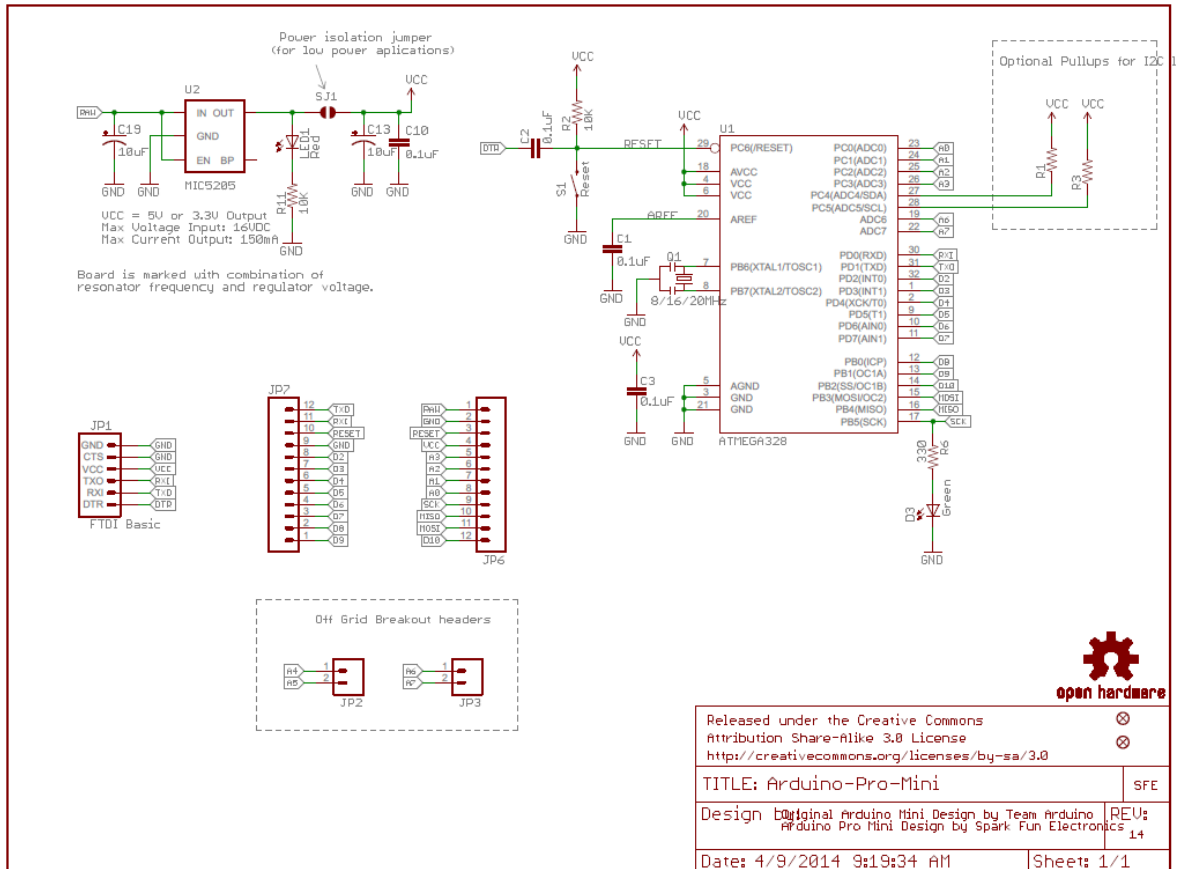
از دلایل انتخاب این برد برای آفتاب نگار طراحی شده، می‌توان به قیمت پایین آن و همچنین ابعاد کوچک آن اشاره نمود که تمامی ویژگی‌های لازم برای این پروژه را شامل می‌شد و همچنین از قطعات اضافی که روی بردهای دیگر وجود دارد و در این پروژه ضروری نبود صرف نظر شده است.

مشخصات فنی آردوینو پرو مینی :

- ولتاژ کاری برد : ۳.۳ ولت
- پین های ورودی/خروجی دیجیتال : ۱۴
- پین های ورودی/خروجی آنالوگ : ۸
- جریان DC هر پین ورودی/خروجی : ۴۰ میلی آمپر
- فرکانس کلاک : ۸ مگاهرتز



تصویر ۱۳- آردوینو پرو مینی

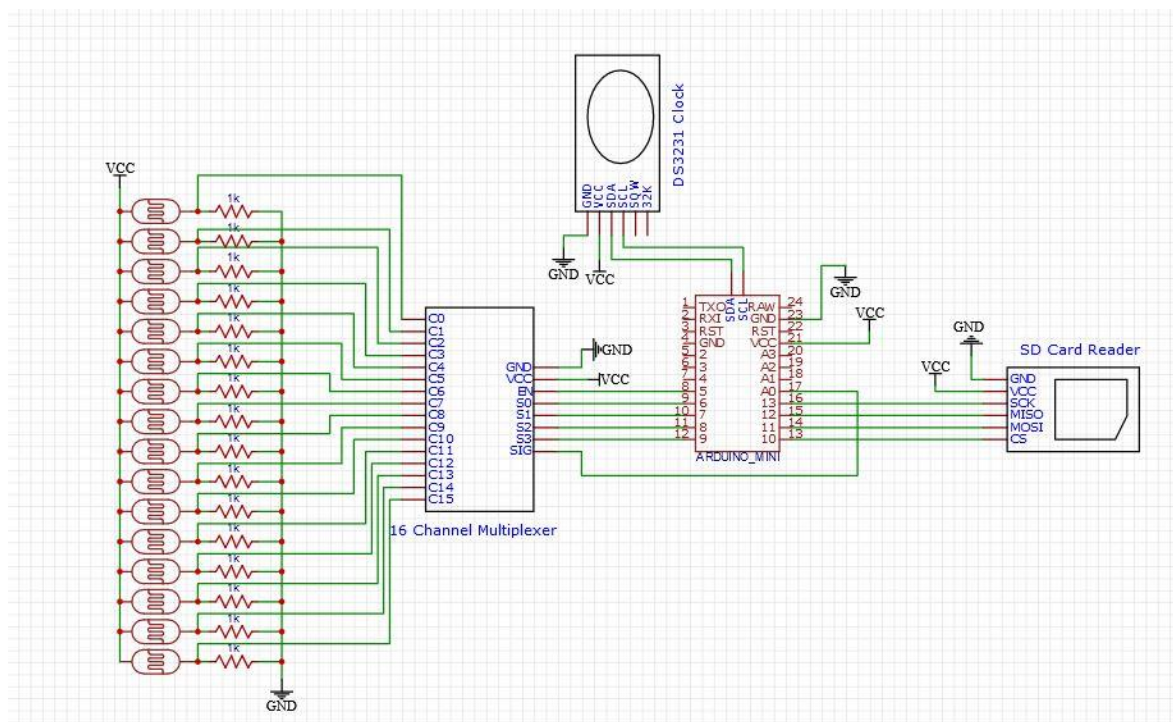


تصویر ۱۴- شماتیک آردوینو پرو مینی [۴]

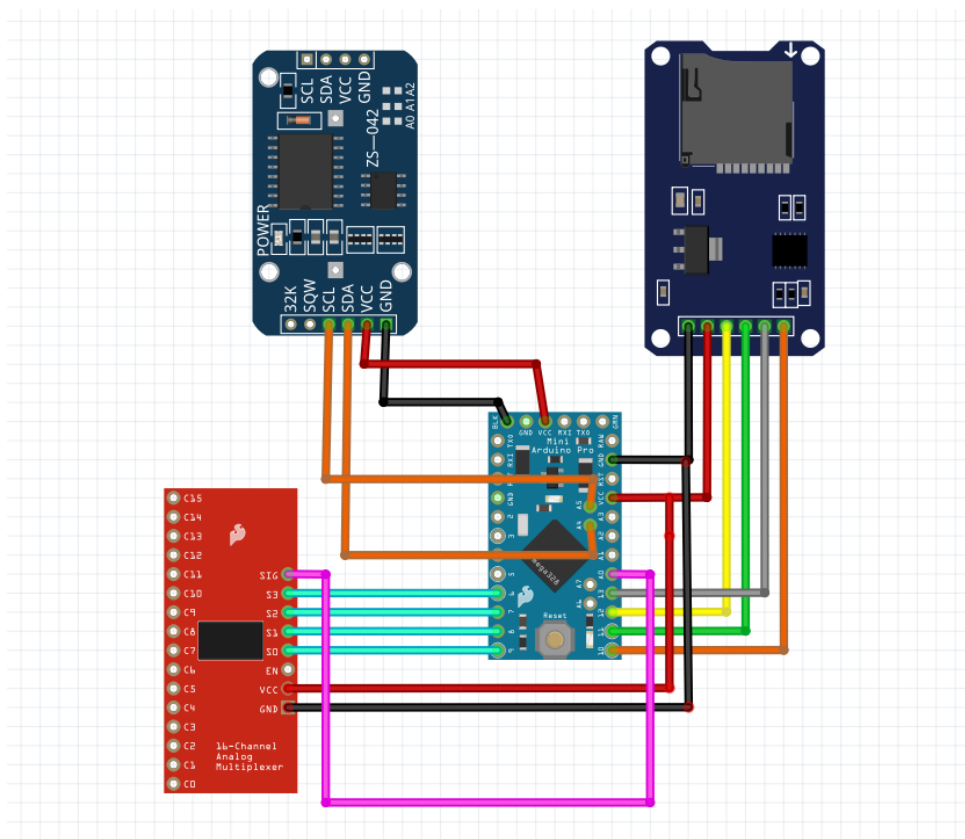
فصل سوم : روش تحقیق

اتصالات الکتریکی

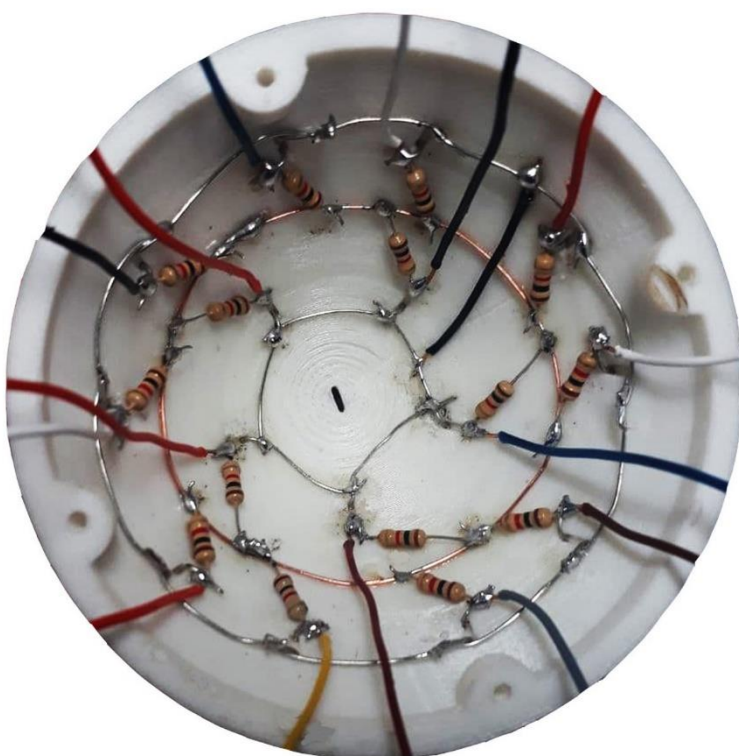
برای خواندن مقدار فوتورزیستورها، از تقسیم ولتاژ استفاده می‌کنیم. به این گونه که یک پایه فوتورزیستور را به VCC و پایه دیگر آن را توسط یک مقاومت ۱ کیلو اهم به زمین متصل می‌کنیم. از محل اتصال فوتورزیستور و مقاومت مقدار ولتاژ را توسط پایه آنالوگ پردازنده می‌خوانیم. با توجه به این که در آفتاب نگار طراحی شده از ۱۶ فوتورزیستور استفاده شده، اگر بخواهیم همه ولتاژهای تولید شده از این تعداد سنسورها را مستقیماً به پردازنده متصل کنیم، اولاً باید پردازنده‌ای انتخاب شود که این تعداد پایه آنالوگ داشته باشد که مستلزم صرف هزینه بیشتری خواهد بود و ثانیاً با توجه به اینکه محل قرارگیری سنسورها از پردازنده فاصله دارند، نیازمند فضای بیشتری بود و همچنین کنترل آنها به دلیل درهم پیچیده شدن اتصالات، دشوار می‌شد. برای جلوگیری از معضلات ذکر شده، از مالتی پلکسر ۱۶ کاناله استفاده شده است که با مقداردهی به پایه‌های کنترلی آن می‌توان تمام خروجی سنسورها را در نهایت توسط یک پایه به پردازنده انتقال داد. استفاده از مالتی پلکسر به کاهش ابعاد طرح و همچنین کنترل بهینه آن کمک شایانی کرده است.



تصویر ۱۵- شماتیک اتصالات مدار طراحی شده



تصویر ۱۶- نمایی دیگر از اتصالات پردازنده و ماژول‌ها



تصویر ۱۷- نحوه اتصال سنسورها و مقاومت‌ها



تصویر ۱۸- اتصال مالتی پلکسر به خروجی سنسورها



تصویر ۱۹- تکمیل و جداسازی بخش مربوط به سنسورها

عملکرد کلی پروژه

هدف از این سیستم، نمونه برداری از مقدار روشنایی محیط توسط خورشید در زمان‌های مختلف از یک شبانه روز است. هرچه تعداد این نمونه‌ها بیشتر باشد، در نتیجه فاصله زمانی بین نمونه‌ها کمتر بوده و داده‌های نهایی از دقت بیشتری برخوردار خواهند بود. کارکرد این طرح به این گونه است که هرچند دقیقه یکبار (با توجه به دقت مورد نیاز) اطلاعات همه‌ی ۱۶ سنسور به نوبت توسط مالتی پلکسر به پردازنده انتقال داده شده و همراه با ساعت و دقیقه آن لحظه که توسط ماژول ساعت دریافت می‌شود، توسط ماژول کارت حافظه در کارت حافظه-ای که درون آن قرار گرفته ذخیره می‌شود و در سریال مانیتور نمایش داده می‌شوند.

برنامه نویسی

با توجه به اینکه از آردوینو به عنوان سیستم پردازنده استفاده شده است، بهترین انتخاب برای برنامه نویسی زبان C و همچنین در محیط Arduino IDE خواهد بود.

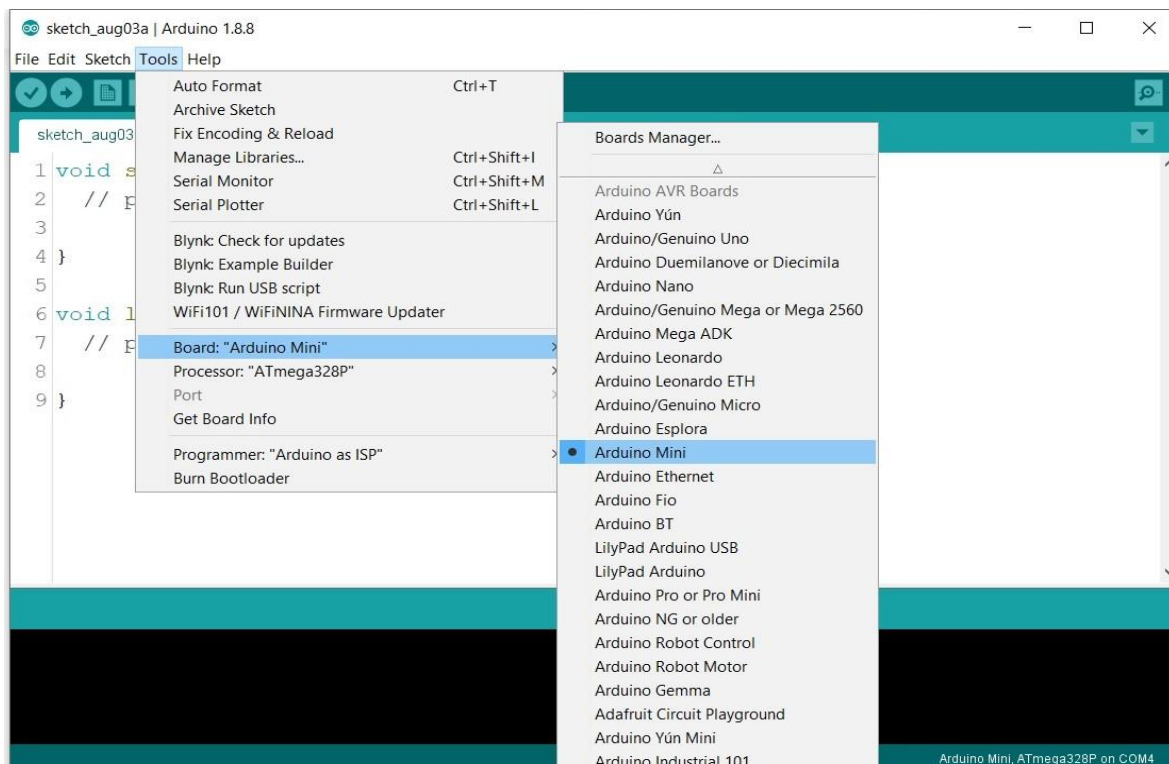
حلقه های اصلی

در آردوینو به هر برنامه، یک طرح^۱ گفته می شود. مبنای کدنویسی آردوینو زبان C و C++ است که برای راحتی کار با آن، تا حد زیادی ساده سازی شده است. یک برنامه آردوینو حداقل شامل دو بخش کلی `Void Setup` و `Void Loop` است. در ادامه ی هر کدام از این بخش ها دو آکولاد وجود دارد. محتوای هر کدام از این بخش ها بین این دو آکولاد تعریف می شود. `Setup` مجموعه ای از کدها را دربر می گیرد که هر وقت آردوینو به برق متصل شود یا کلید `Reset` فشرده شود، تنها یک بار اجرا می شود. `Loop` مجموعه ای از کدها را در خود دارد که باید بدون وقفه و به صورت تکراری اجرا شوند.

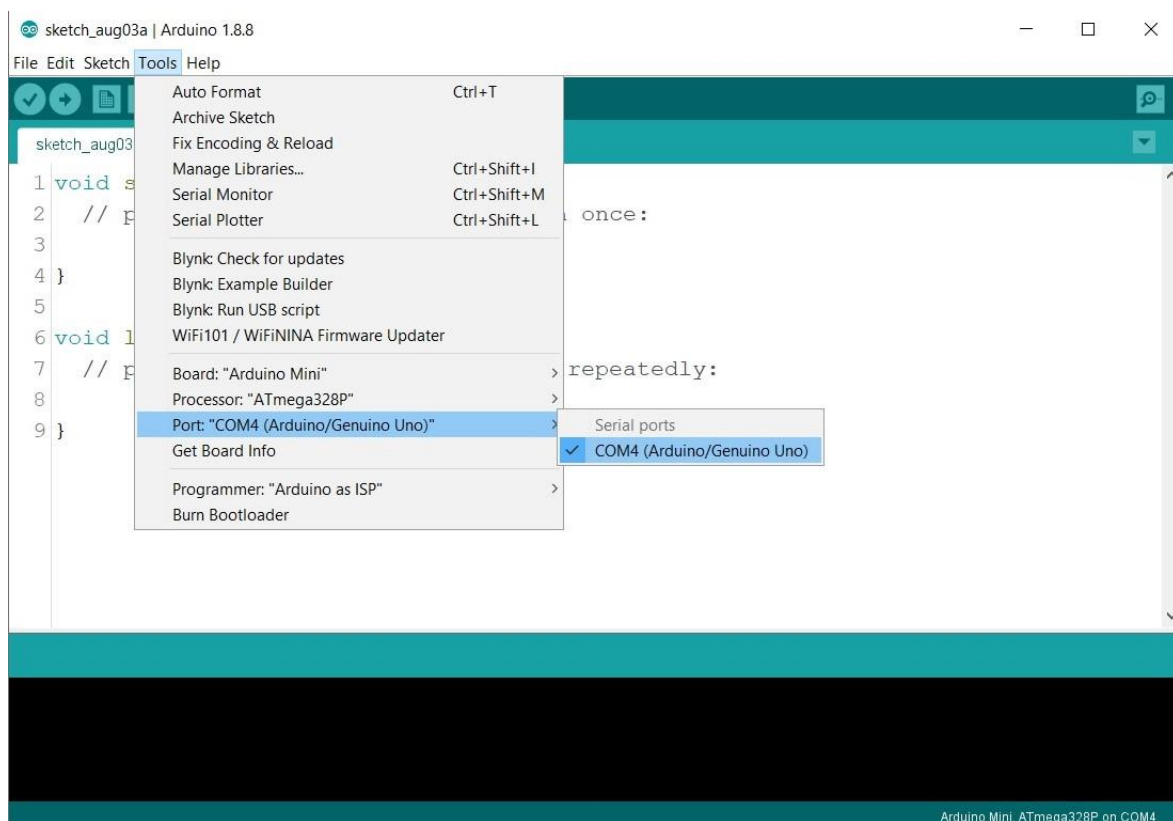
انتخاب برد و پورت

هر زمان که یک برد آردوینو به رایانه متصل شود، قبل از هر چیز باید برد و پورت مورد استفاده را به برنامه معرفی کرد. برای این منظور از مسیر `Tools → Board` مدل و از مسیر `Tools → Port` پورتهای که آردوینو به آن متصل شده است، انتخاب می شود.

^۱ Sketch



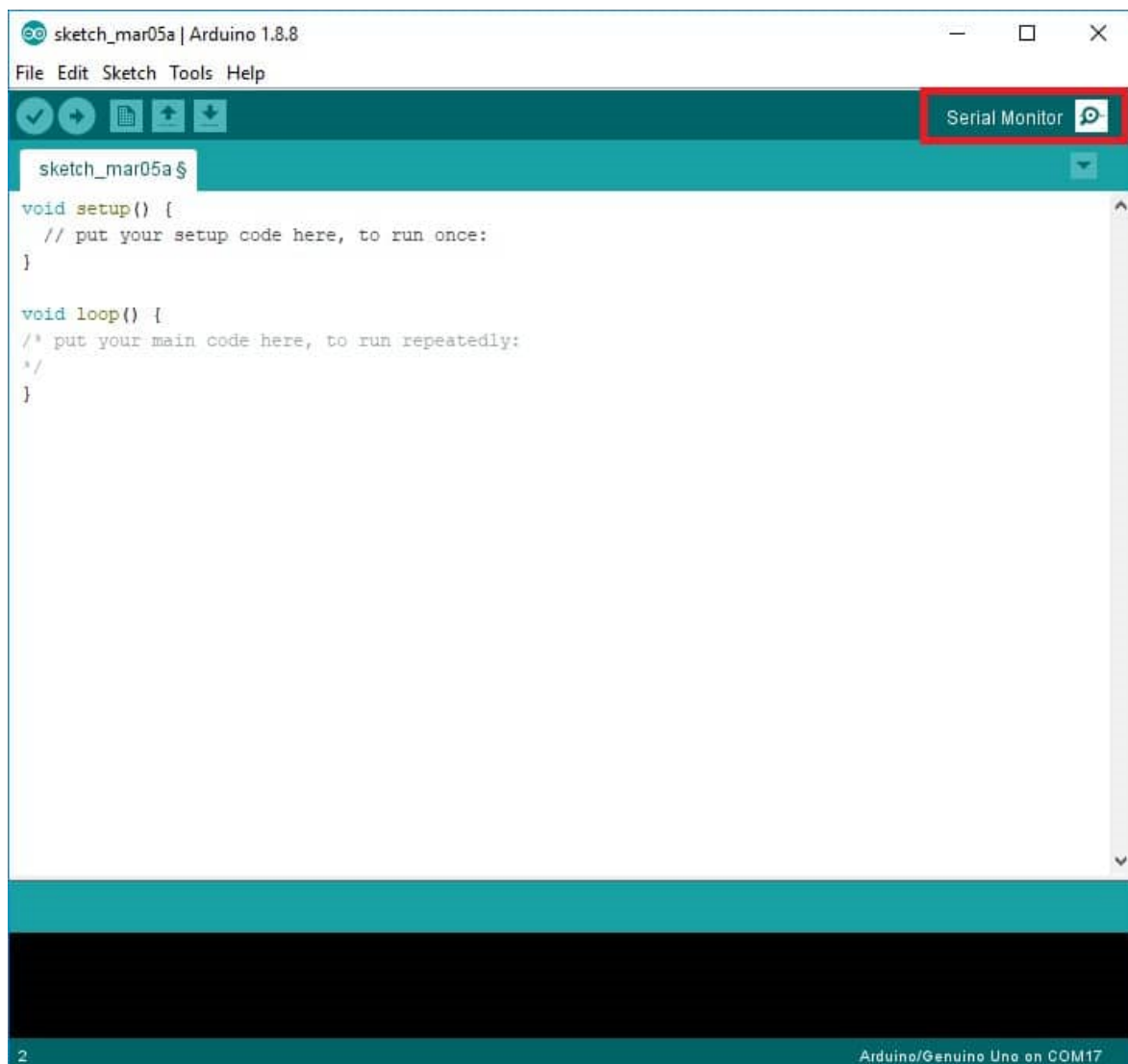
تصویر ۲۰- انتخاب برد آردوینو



تصویر ۲۱- انتخاب پورت آردوینو

سریال مانیتور

در آردوینو محیطی به نام سریال مانیتور^۱ وجود دارد که در آن می‌توان متن مورد نظر را نمایش داد. کلیدی که در انتهای سمت راست نوار بالا وجود دارد پنجره سریال مانیتور را باز می‌کند.



تصویر ۲۲- سریال مانیتور آردوینو

^۱ Serial Monitor

در سریال مانیتور می‌توان خروجی یک تابع، اطلاعات یک ماژول یا هر اطلاعات دیگری نوشت. این کار کمک می‌کند تا از روند اجرای برنامه مطلع شد و بتوان آن را عیب‌یابی کرد. علاوه بر آن در نوار بالای سریال مانیتور می‌توان یک دستور یا عبارت را به برد ارسال کرد. برای استفاده از سریال مانیتور کافیست در حلقه `setup` دستور `"Serial.begin(9600)"` را نوشت. عدد درون پرانتز، سرعت داده‌برداری سریال را نشان می‌دهد و می‌توان مقدار آن را عوض کرد.

کتابخانه در آردوینو

کتابخانه‌ها مجموعه‌ای از تعاریف و توابع هستند که یک بار برای همیشه نوشته می‌شوند و از آن پس، فقط مورد استفاده قرار می‌گیرند. استفاده از کتابخانه، حجم کد نوشته شده را کم کرده، برنامه خواناتر و قابل فهم‌تر شده و منابع سخت‌افزاری کمتری استفاده می‌کند.

ساده‌ترین راه برای نصب یک کتابخانه، دانلود آن با خود نرم‌افزار است. برای این کار کافیست به مسیر `Sketch → Include Library → Manage Libraries` رفته و پس از مشاهده لیست کتابخانه‌ها، عنوان کتابخانه را جستجو کرده و با کلیک بر روی گزینه `install` کتابخانه موردنظر را نصب کرد.

برنامه نوشته شده برای آفتاب نگار طراحی شده
کل برنامه نوشته شده و توضیح خطوط آن در ادامه ذکر شده است.

```

1 #include <SPI.h>
2 #include <SD.h>
3 const int chipSelect = 10;
4
5 #include <Wire.h>
6 #include <Sodaq_DS3231.h>
7 DateTime now;
8
9 long now_data[16];
10 long before_data[16];
11 long sum_data[16];
12 int maximum;
13 long max_push;
14 int now_ave, before_ave;
15 float now_light_percent;
16 long now_sum;
17 int cloud, c, cloudy_time;
18
19 //Mux control pins
20 int s0 = 2;
21 int s1 = 3;
22 int s2 = 4;
23 int s3 = 5;

```

تصویر ۲۳- بخش اول برنامه نوشته شده

۱- کتابخانه‌های SPI و SD جهت استفاده از پروتکل SPI و همچنین استفاده از ماژول

کارت حافظه فراخوانی می‌شود. و سپس عنوان chipSelect به پایه ۱۰ نسبت داده می‌شود.

۲- در این بخش کتابخانه‌های مورد نیاز جهت راه‌اندازه ماژول ساعت فراخوانی می‌شود.

۳- در سه خط اول این بخش، سه آرایه با طول ۱۶ تعریف شده است. در ادامه نیز متغیرهایی تعریف شده‌اند که به آنها برای ذخیره سازی اطلاعات نیاز خواهیم داشت.

```
final2 $
19 //Mux control pins
20 int s0 = 2;
21 int s1 = 3;
22 int s2 = 4;
23 int s3 = 5;
24
25 int value ;
26 void setup() {
27     pinMode(s0, OUTPUT);
28     pinMode(s1, OUTPUT);
29     pinMode(s2, OUTPUT);
30     pinMode(s3, OUTPUT);
31
32     Serial.begin(9600);
33     Wire.begin();
34     rtc.begin();
35     DateTime dt(2021, 6, 7, 0, 0, 0, 0);
36     rtc.setDateTime(dt);
37     while (!Serial) {
38         ;
39     }
40
41     Serial.print("Initializing SD card...");
```

تصویر ۲۴- بخش دوم برنامه نوشته شده

۴- ابتدا عناوین S0, S1, S2 و S3 به ترتیب به پایه‌های ۲، ۳، ۴ و ۵ نسبت داده شده و

در ادامه این پایه‌ها به عنوان پایه‌های خروجی تعریف شده‌اند. از این پایه‌ها برای کنترل

این که کدام ورودی به خروجی انتقال داده شود، استفاده می‌شود.

۵- در این بخش سرعت بادریت^۱ برای نمایش همزمان اطلاعات دریافت شده در قسمت

سریال مانیتور برنامه در صورت متصل بودن کامپیوتر به آردوینو، تعیین می‌شود.

۶- تنظیمات اولیه مربوط به ماژول ساعت (سال، ماه، روز، ساعت، دقیقه، ثانیه) در این

بخش انجام می‌گیرد.

^۱ Baud rate


```
final2 $
41 Serial.print("Initializing SD card...");
42
43 if (!SD.begin(chipSelect)) {
44     Serial.println("Card failed, or not present");
45     while (1);
46 }
47 Serial.println("card initialized.");
48
49 if (SD.exists("datalog.txt"))
50 {
51     SD.remove("datalog.txt");
52 }
53 }
54
55
56 void loop() {
57
58     now = rtc.now(); } ۹
59     File dataFile = SD.open("datalog.txt", FILE_WRITE); } ۱۰
60     if (dataFile) {
61         dataFile.print(now.hour());
62         dataFile.print(':');
63         dataFile.print(now.minute());
```

تصویر ۲۵- بخش سوم برنامه نوشته شده

۷- موفقیت یا عدم موفقیت در راه اندازی و تشخیص کارت حافظه توسط این خطوط در

مانیتور سریال نمایش داده خواهد شد.

۸- در این بخش فایلی به نام datalog جهت ذخیره اطلاعات دریافتی ایجاد و در کارت

حافظه ذخیره می شود.

۹- در این خط زمان از ماژول ساعت دریافت و در متغیر now ذخیره می شود.

۱۰- با استفاده از این دستور فایلی که در کارت حافظه ساخته شده بود، باز شده و آماده

نوشته شدن خواهد بود.

```
final2 $
60  if (dataFile) {
61      dataFile.print(now.hour());
62      dataFile.print(':');
63      dataFile.print(now.minute());
64      dataFile.print(':');
65      dataFile.print(now.second());
66      Serial.print(now.hour());
67      Serial.print(':');
68      Serial.print(now.minute());
69      Serial.print(':');
70      Serial.print(now.second());
71  }
72
73  for(int i = 0; i < 16; i++){
74      value = readMux(i);
75      now_data[i]= value;
76
77      dataFile.print(",");
78      dataFile.print(value);
79      // print to the serial port too:
80      Serial.print(",");
81      Serial.print(value);
82      delay(500);
  }
```

تصویر ۲۶- بخش چهارم برنامه نوشته شده

۱۱- در این بخش ساعت، دقیقه و ثانیه به ترتیب ابتدا در کارت حافظه ذخیره و سپس در

مانیتور سریال نمایش داده می‌شوند.

۱۲- در این بخش یک حلقه ایجاد شده است که قرار است ۱۶ بار (به تعداد سنسورها)

تکرار شود، به طوری که هر بار شماره سنسور را به تابع readMux فرستاده و مقدار

آن سنسور را دریافت می‌کند. سپس ابتدا آن را در آرایه‌ای به نام now_data ذخیره

کرده و در ادامه در کارت حافظه ذخیره و پس از آن در مانیتور سریال نمایش می‌دهد.

```
final2 $
85 maximum = find_max(now_data,0);
86 find_min(now_data,0);
87 max_push = max_push + maximum;
88 for(int j=0; j < 16; j++){
89     sum_data[j] += now_data[j];
90 }
91 find_max(sum_data,1);
92 find_min(sum_data,1);
93 Serial.println("");
94 Serial.print("max_push: ");
95 Serial.println(max_push);
96 now_sum=0;
97 for(int k=0;k<16;k++){
98     now_sum += now_data[k];
99 }
100 now_ave = now_sum/16;
101 now_light_percent = (now_ave/890)*100;
102 Serial.print("now_light_percent : ");
103 Serial.print(now_light_percent);
104 Serial.println("%");
105 if(c==1){
106     if(now_ave - before_ave > 40){
107         c=0;
```

تصویر ۲۷- بخش پنجم برنامه نوشته شده

۱۳- آرایه now_data به دو تابع find_max و find_min فرستاده شده و سپس

مقدار maximum برگشت داده شده توسط تابع find_max با مقدار قبلی متغیر

max_push جمع شده و دوباره در متغیر max_push قرار می گیرد.

۱۴- با استفاده از این حلقه بعد از هر نوبت اندازه گیری اطلاعات، مقدار هر سنسور با مقادیر

گذشته خود جمع شده و در متغیر sum_data ذخیر می شود.

۱۵- متغیر sum_data به دو تابع find_max و find_min فرستاده می شود.

۱۶- مقدار جدید متغیر max_push که در بخش ۱۳ به دست آمد، در این بخش در

سریال مانیتور به نمایش در می آید.

۱۷- در این بخش ابتدا تمام مقادیر موجود در آرایه `now_data` با یکدیگر جمع شده و سپس میانگین آنها محاسبه می‌شود. چون بیشترین مقداری که می‌توان دریافت کرد ۸۹۰ است، با تقسیم مقدار میانگین به دست آمده بر این عدد و سپس ضرب آن در ۱۰۰، درصد آرایه `now_data` به دست می‌آید. در آخر درصد به دست آمده از طریق سریال مانیتور به نمایش در می‌آید.

```

105 if(c==1){
106     if(now_ave - before_ave > 40){
107         c=0;
108     }
109     else {cloudy_time++;}
110 }
111 if(c==0){
112     if(before_ave - now_ave > 40){
113         cloud++;
114         c=1;
115         cloudy_time++;
116     }
117 }
118 Serial.print("number_of_clouds : ");
119 Serial.println(cloud);
120 Serial.print("cloudy_time : ");
121 Serial.print(cloudy_time*2);
122 Serial.println(" minutes");
123 weather_condition(c,now_light_percent);
124 Serial.println("-----");
125
126 for(int m=0;m<16;m++){
127     before_data[m] = now_data[m];
128 }

```

Diagram annotations in the image:

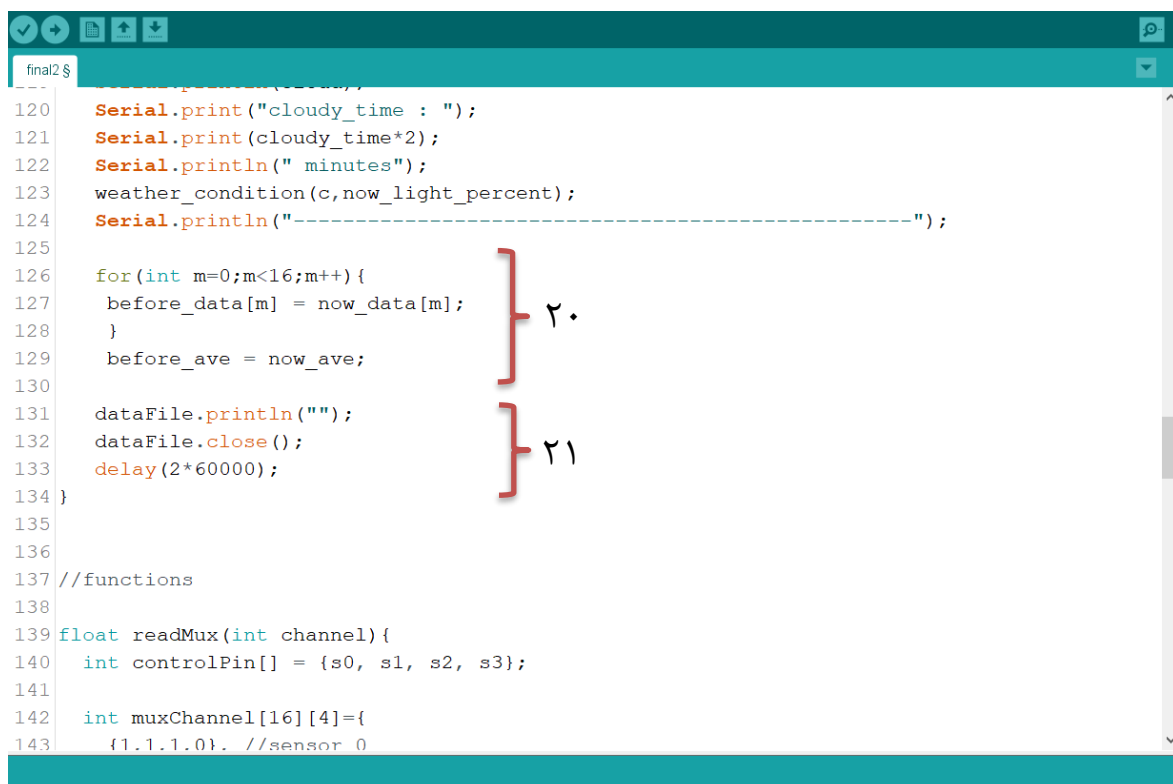
- A red bracket labeled **۱۸** spans lines 105 through 117, covering the conditional logic for updating cloud counts.
- A red bracket labeled **۱۹** spans lines 118 through 124, covering the serial printing statements.

تصویر ۲۸- بخش ششم برنامه نوشته شده

۱۸- در این بخش با توجه به مقادیر `before_ave` و `now_ave` می‌توان به وجود سایه توسط ابر پی برد. اگر مقدار `C` برابر ۱ باشد به این معنی است که هم اکنون خورشید پشت ابر قرار دارد و اگر مقدار `C` برابر صفر باشد، به این معنی است که هیچ ابری

جلوی خورشید نیست. متغیر `cloudy_time` مدت زمان وجود ابر جلوی خورشید را اندازه می‌گیرد.

۱۹- ابتدا مقادیر محاسبه شده در بخش قبل، در سریال مانیتور نمایش داده می‌شوند و سپس مقادیر `c` و `now_light_percent` که در بخش ۱۷ محاسبه شده بود، به تابع `weather_condition` ارجاع داده می‌شوند. و در آخر خطی برای جداسازی اطلاعات این نوبت اندازه‌گیری با نوبت دیگر رسم می‌شود.

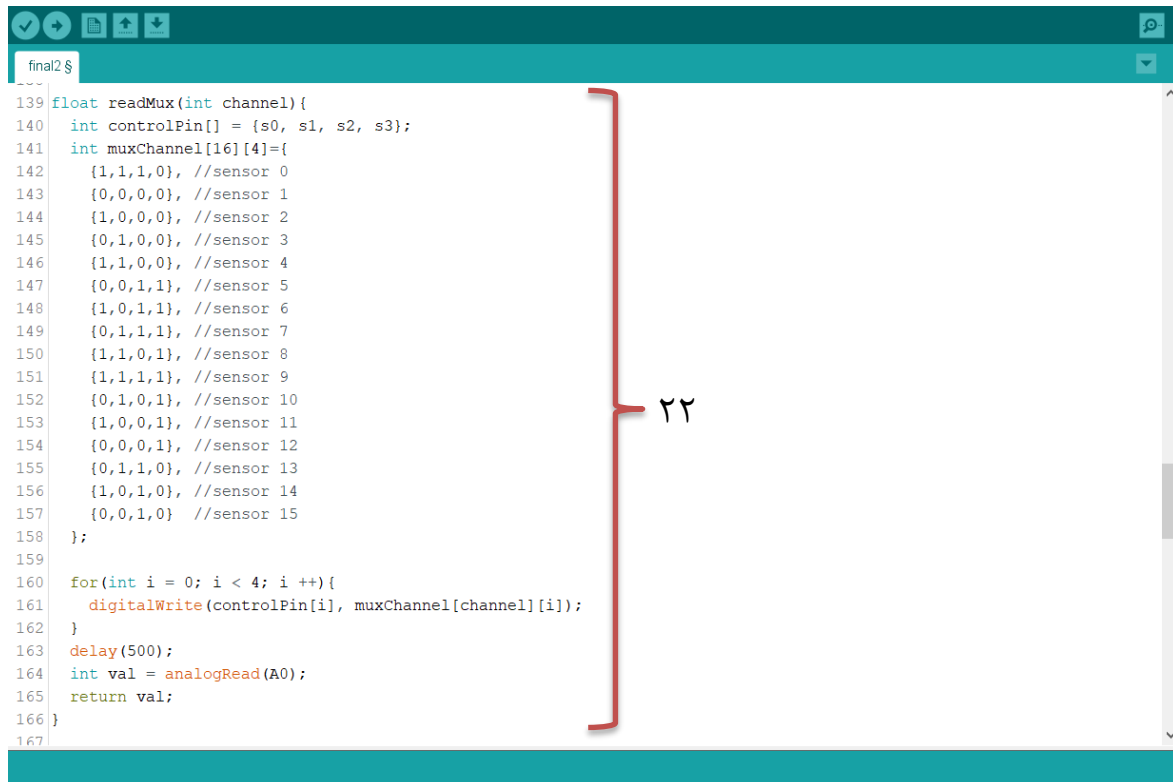


```
120 Serial.print("cloudy_time : ");
121 Serial.print(cloudy_time*2);
122 Serial.println(" minutes");
123 weather_condition(c,now_light_percent);
124 Serial.println("-----");
125
126 for(int m=0;m<16;m++){
127     before_data[m] = now_data[m];
128 }
129 before_ave = now_ave;
130
131 dataFile.println("");
132 dataFile.close();
133 delay(2*60000);
134 }
135
136
137 //functions
138
139 float readMux(int channel){
140     int controlPin[] = {s0, s1, s2, s3};
141
142     int muxChannel[16][4]={
143         {1,1,1,0}. //sensor 0
```

تصویر ۲۹- بخش هفتم برنامه نوشته شده

۲۰- در این بخش ابتدا مقادیر آرایه `now_data` به آرایه `before_data` انتقال داده شده و سپس مقدار `now_ave` به `before_ave` منتقل می‌شود.

۲۱- فایلی که در بخش ۸ باز شده بود، بسته می‌شود و در آخر به مدت ۲ دقیقه وقفه صورت می‌گیرد.

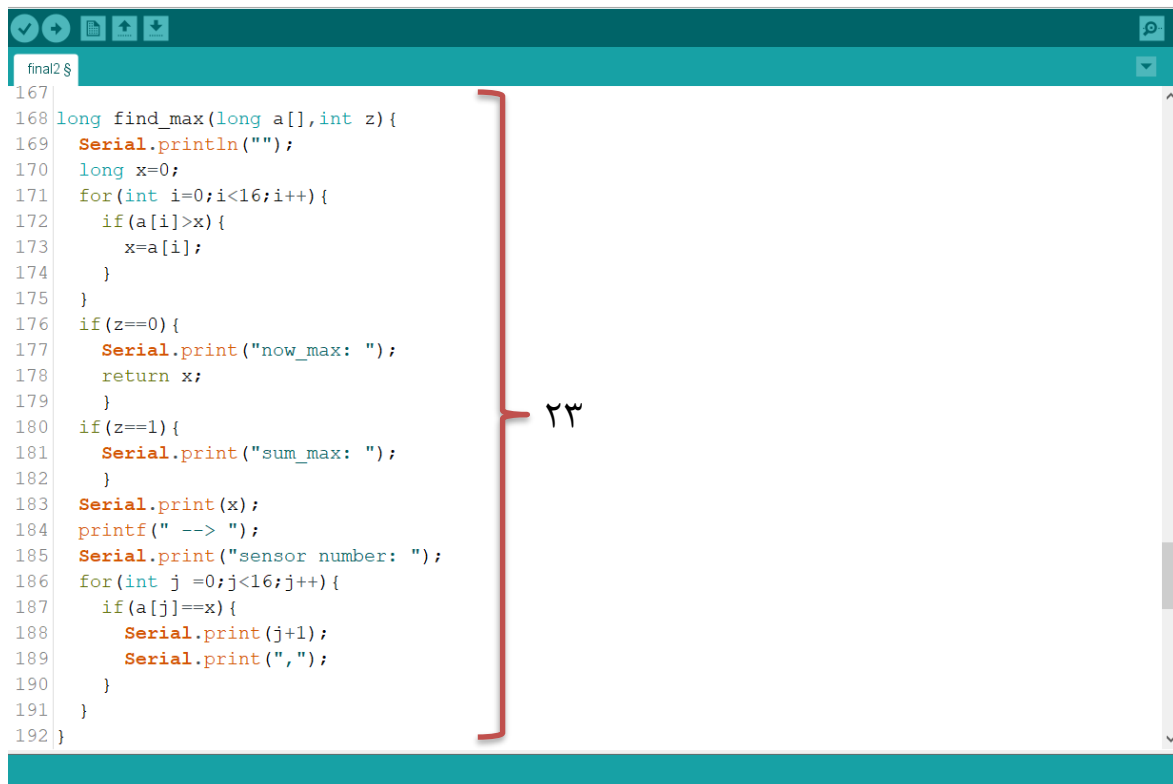


```
139 float readMux(int channel){
140     int controlPin[] = {s0, s1, s2, s3};
141     int muxChannel[16][4]={
142         {1,1,1,0}, //sensor 0
143         {0,0,0,0}, //sensor 1
144         {1,0,0,0}, //sensor 2
145         {0,1,0,0}, //sensor 3
146         {1,1,0,0}, //sensor 4
147         {0,0,1,1}, //sensor 5
148         {1,0,1,1}, //sensor 6
149         {0,1,1,1}, //sensor 7
150         {1,1,0,1}, //sensor 8
151         {1,1,1,1}, //sensor 9
152         {0,1,0,1}, //sensor 10
153         {1,0,0,1}, //sensor 11
154         {0,0,0,1}, //sensor 12
155         {0,1,1,0}, //sensor 13
156         {1,0,1,0}, //sensor 14
157         {0,0,1,0} //sensor 15
158     };
159
160     for(int i = 0; i < 4; i++){
161         digitalWrite(controlPin[i], muxChannel[channel][i]);
162     }
163     delay(500);
164     int val = analogRead(A0);
165     return val;
166 }
167
```

تصویر ۳۰- بخش هشتم برنامه نوشته شده

۲۲- در این بخش تابعی به نام readMux تعریف شده است. همانطور که از نام تابع مشخص است، وظیفه کنترل مالتی پلکسر را برعهده دارد. کارکرد این تابع به این گونه است که ابتدا شماره سنسوری که مقدار آن مدنظر است را دریافت کرده، سپس با توجه به شماره سنسور مقادیر S0-3 را طوری مقدار دهی می‌کند که مالتی پلکسر مقدار آن سنسور مشخص شده را به خروجی هدایت کند. در نهایت این مقدار خروجی

در متغیری به نام val ذخیره شده و به محلی که این تابع فراخوانی شده است، برگشت داده می‌شود.




```
167
168 long find_max(long a[],int z){
169     Serial.println("");
170     long x=0;
171     for(int i=0;i<16;i++){
172         if(a[i]>x){
173             x=a[i];
174         }
175     }
176     if(z==0){
177         Serial.print("now_max: ");
178         return x;
179     }
180     if(z==1){
181         Serial.print("sum_max: ");
182     }
183     Serial.print(x);
184     printf(" --> ");
185     Serial.print("sensor number: ");
186     for(int j =0;j<16;j++){
187         if(a[j]==x){
188             Serial.print(j+1);
189             Serial.print(",");
190         }
191     }
192 }
```

تصویر ۳۱- بخش نهم برنامه نوشته شده

۲۳- این بخش مربوط به تعریف تابع find_max است. این تابع دو ورودی دارد که یکی از آنها آرایه بوده و دیگری یک متغیر است که مقدار آن صفر یا ۱ خواهد بود. از اسم تابع مشخص است که وظیفه آن پیدا کردن بیشترین مقدار موجود در آرایه ارسالی به آن است. دیگر ورودی این تابع صرفاً این وظیفه را دارد مشخص کند که آرایه ارسال شده به تابع، now_data است یا sum_data تا در نمایش اطلاعات عناوین به درستی در سریال مانیتور نمایش داده شوند.

```
final2 $
193
194 long find_min(long a[],int z){
195     Serial.println("");
196     long x=100000;
197     for(int i=0;i<16;i++){
198         if(a[i]<x){
199             x=a[i];
200         }
201     }
202     if(z==0){
203         Serial.print("now_min: ");
204     }
205     if(z==1){
206         Serial.print("sum_min: ");
207     }
208     Serial.print(x);
209     printf(" --> ");
210     Serial.print("sensor number: ");
211     for(int j =0;j<16;j++){
212         if(a[j]==x){
213             Serial.print(j+1);
214             Serial.print(",");
215         }
216     }
217 }
218
```



تصویر ۳۲- بخش دهم برنامه نوشته شده

۲۴- در این بخش تابع find_min تعریف شده است. اطلاعات و عملکرد این بخش مانند بخش قبل است با این تفاوت که در این تابع کمترین مقدار آرایه ارسال شده به آن تعیین شده و در سریال مانیتور نمایش داده می شود.


```
final2 $
213     Serial.print(j);
214     Serial.print(",");
215 }
216 }
217 }
218
219 void weather_condition(int x, float y){
220     if(x==0){
221         Serial.print("The sky is clear, ");
222     }
223     if(x==1){
224         Serial.print("The sky is cloudy, ");
225     }
226     if(y>85){
227         Serial.println("and the sun is shining brutally.");
228     }
229     if(y>65 && y<85){
230         Serial.println("and the sun is shining softly.");
231     }
232     if(y>35 && y<65){
233         Serial.println("and the sun is shining weakly.");
234     }
235     if(y<35){
236         Serial.println("and the sun is off.");
237     }
238 }
```

تصویر ۳۳- بخش یازدهم برنامه نوشته شده

۲۵- این بخش مربوط به تعریف تابع `weather_condition` است. این تابع دو ورودی مربوط به وجود یا عدم وجود ابر و مقدار درصد روشنایی حال حاضر را دریافت می‌کند و با توجه به مقادیر این دو ورودی، یک جمله توصیفی از وضعیت آسمان و خورشید بیان می‌کند.

فصل چهارم : یافته های تحقیق

نحوه تست عملکرد این طرح این گونه بود که چند ساعت قبل از طلوع خورشید و در تاریکی مطلق سیستم روشن شده و شروع به ثبت اطلاعات می کند، و در پایان روز و چند ساعت بعد از غروب خورشید در وضعیت تاریکی مشابه شروع تست، سیستم خاموش می شود. اطلاعات به صورت لحظه ای توسط سریال مانیتور قابل مشاهده می باشد و همچنین می توان اطلاعات ذخیره شده در کارت حافظه را برداشته و جداگانه بررسی کرد.

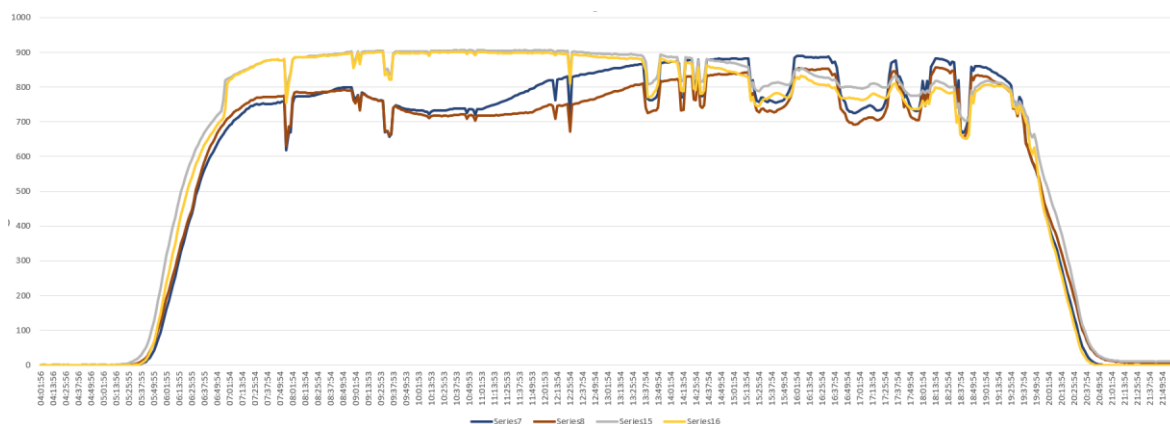
اطلاعات اندازه گیری شده یک شبانه روز تمام سنسورها (از ساعت ۴ بامداد تا ۱۰ شب) در تصویر ۳۴ که به صورت نمودار و در برنامه اکسل رسم شده، آورده شده است.



تصویر ۳۴- نتایج نهایی ۱۶ سنسور

با توجه به اینکه در تصویر ۳۴ اطلاعات به وضوح قابل تشخیص نیستند، در ادامه بصورت انتخابی از بین سنسورها بصورت دو منحنی در کنار هم قرار داده می شود تا قابل مقایسه با یکدیگر باشند.

ذکر این نکته ضروری است که این تست در شرق زنجان و در ارتفاع ۱۰ متری از سطح زمین و در محلی که هیچ سایه‌ای توسط اجسام محیطی در آنجا رخ نمی‌دهد، انجام شده است. همچنین قرارگیری سیستم به گونه‌ای است که سنسورهای ۱۵ و ۱۶ تقریباً به سمت طلوع آفتاب و سنسورهای ۷ و ۸ به سمت غروب آفتاب هستند. در ادامه به مقایسه نمودار حاصله از اطلاعات این سنسورها با یکدیگر می‌پردازیم.



تصویر ۳۵- نمودار مربوط به سنسورهای ۷ و ۸ و ۱۵ و ۱۶

با توجه به تصویر ۳۵، مشاهده می‌کنیم که نمودار مربوط به سنسورهای ۱۵ و ۱۶ که رو به سمت طلوع قرار داشتند (منحنی‌های خاکستری و زرد رنگ)، در ابتدای روز مقدار روشنایی بیشتری نسبت به دیگر سنسورها ثبت کرده‌اند که با گذشت زمان و تغییر محل خورشید در آسمان، مقدار روشنایی دریافتی توسط آنها رو به کاهش است. در طرف مقابل سنسورهای ۷ و ۸ (منحنی‌های آبی و قهوه‌ای) که به سمت غروب خورشید قرار دارند، در ابتدای روز مقدار روشنایی کمتری را ثبت کرده‌اند اما با گذشت زمان می‌بینیم که روشنایی دریافتی توسط آنها رو به افزایش است.

همانطور که در تصاویر ۳۴ و ۳۵ مشاهده می‌کنیم در زمان‌هایی مقدار منحنی‌ها به یکباره کاهش و پس از مدت کوتاهی به مقدار قبلی خود باز می‌گردند، این بخش‌ها نمایانگر زمان‌هایی است که خورشید پشت ابر قرار می‌گیرد و سایه کلی بر روی ناحیه‌ای از زمین ایجاد می‌شود. همچنین همانطور که مشخص است، روشنایی هوا قبل از طلوع شروع شده و همچنین در زمان غروب مقداری روشنایی وجود دارد که پس از مدتی به تاریکی کامل می‌رسیم. نمونه‌ای از یک خروجی در سریال مانیتور که مربوط به یک نوبت اندازه‌گیری است، در تصویر ۳۶ به نمایش درآمده است که در ادامه به تعریف هر خط آن می‌پردازیم.

```

14:59:54,859,880,902,885,895,874,882,839,875,825,762,785,748,820,869,844 ۱
now_max: 902 --> sensor_number: 3 ۲
now_min: 748 --> sensor_number: 13 ۳
sum_max: 232367 --> sensor_number: 15 ۴
sum_min: 197536 --> sensor_number: 8 ۵
max_push: 234327 ۶
now_light_percent: 95.11% ۷
number_of_clouds: 7 ۸
cloudy_time: 40 minutes ۹
The sky is clear, and the sun is shining brutally. ۱۰

```

تصویر ۳۶-نمونه‌ای از اطلاعات نمایش داده شده در سریال مانیتور

- ۱- زمان دریافت اطلاعات و همچنین دریافتی در این خط نشان داده شده است.
- ۲- شماره سنسوری که بیشترین مقدار را در حال حاضر ثبت کرده است همراه با مقدار آن در این خط نشان داده می‌شود.
- ۳- شماره سنسوری که کمترین مقدار را در حال حاضر ثبت کرده است همراه با مقدار آن در این خط نشان داده می‌شود.
- ۴- شماره سنسوری که بیشترین مقدار را از ابتدای فرآیند اندازه‌گیری ثبت کرده است همراه با مقدار آن در این خط نشان داده می‌شود.

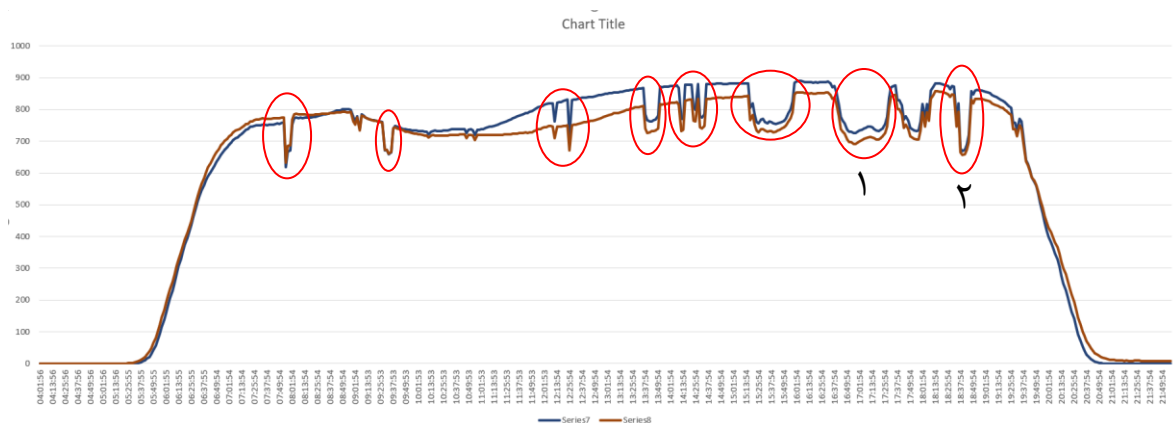
- ۵- شماره سنسوری که کمترین مقدار را از ابتدای فرآیند اندازه‌گیری ثبت کرده است همراه با مقدار آن در این خط نشان داده می‌شود.
- ۶- جمع بیشترین مقادیر در هر نوبت از ابتدای فرآیند اندازه‌گیری در این خط نشان داده می‌شود.
- ۷- مقدار درصد روشنایی حال حاضر در این خط نشان داده می‌شود.
- ۸- تعداد دفعاتی که خورشید پشت ابر قرار گرفته است نشان داده می‌شود.
- ۹- مدت زمان پشت ابر بودن خورشید به طور تقریبی محاسبه و نشان داده شده است.
- ۱۰- با توجه به اطلاعات بدست آمده، وضعیت ابرها و تابش خورشید شرح داده می‌شود.

نتیجه‌گیری

اطلاعاتی که به صورت *real time* از طریق سریال مانیتور نمایش داده می‌شوند، اطلاعات جامع و کاملی از وضعیت فعلی و همچنین ساعات گذشته ارائه می‌کند. با توجه به مقادیر *now_max* و *now_min* می‌توان متوجه شد که در حال حاضر کدام سنسورها بیشترین و کمترین مقدار از تابش خورشید را دریافت می‌کنند که می‌تواند با توجه به شماره سنسورها و محل قرارگیری آنها در مورد تشخیص محل قرارگیری فعلی خورشید راهنمایی کند. مقادیر *sum_max* و *sum_min* اشاره به سنسورهایی دارند که به ترتیب بیشترین و کمترین مقادیر روشنایی را از ابتدای شروع روند اندازه‌گیری دریافت و ثبت کرده‌اند. محل و جهت قرارگیری این سنسورها می‌تواند راهنمایی بسیار خوبی برای شناسایی محل مناسب برای استقرار و جهت قرار گرفتن پنل‌های خورشیدی باشد. هرچه مقدار *max_push* بیشتر

باشد، نشان‌دهنده‌ی این است که روز پر نور و آفتابی داشتیم. این مقدار با جمع مقادیر ماکزیمم سری‌های اندازه‌گیری بدست می‌آید. اگر نقاط ماکزیمم نمودار سنسورها را به هم متصل کنیم، یک منحنی جدید تشکیل می‌شود که max_push بیانگر مساحت زیر این منحنی است. با مقایسه مقادیر max_push حاصل شده از روزهای مختلف می‌توان به وضعیت جوی نسبی آنها پی برد و با یکدیگر مقایسه کرد، واضح است که هرچه این مقدار بیشتر باشد بیانگر هوای آفتابی و هرچه کمتر باشد بیانگر آسمانی ابری است. تعداد ابرهایی که در مقابل خورشید قرار گرفته‌اند و همچنین زمان محاسبه شده تقریبی این رویدادها نیز می‌تواند اطلاعات مناسبی درباره‌ی وضعیت جوی ارائه دهد.

تصویر شماره ۳۶ دو نمودار مربوط به سنسورهای ۷ و ۸ را که رو به غروب خورشید بودند را نمایش می‌دهد.



تصویر ۳۷- نمودار مربوط به سنسورهای ۷ و ۸

در نگاه اول متوجه فرورفتگی‌های یکسان روی هر دو نمودار می‌شویم. در تصویر شماره ۳۶ این فرورفتگی‌ها با علامت قرمز مشخص شده‌اند. همانطور که در بخش قبل اشاره شد، این فرورفتگی‌ها برای زمانی است که خورشید پشت ابر قرار می‌گیرد. با توجه به نوع و شکل این

فرورفتگی‌ها می‌توان به شدت و مدت زمان ابری بودن هوا پی برد. هر چقدر طول این فرورفتگی‌ها بیشتر باشد، نشان از طولانی بودن سایه و بزرگی ابرها خواهد بود. همچنین عمق فرورفتگی‌ها رابطه مستقیمی با ضخامت ابرها و میزان تاریکی ایجاد شده توسط سایه آنها دارد. با مقایسه دو فرورفتگی‌ای که با شماره ۱ و ۲ مشخص شده‌اند متوجه می‌شویم که تقریباً ضخامت و تاریکی ایجاد شده توسط هر دو ابر یکسان بوده است اما ابری که باعث فرورفتگی شماره ۱ است، بزرگتر بوده و خورشید مدت زمان بیشتری پشت آن قرار داشت. در مناطقی که ممکن است کل روز هوا ابری باشد، می‌توان با مقایسه نمودارهای بدست آمده با روز یا روزهای قبل آن منطقه که هوا آفتابی بود، به میزان ابری بودن هوا پی برد.

برای یافتن مسیر حرکت خورشید کافی است مشخص کنیم که در هر سیکل جمع-آوری اطلاعات، بیشترین مقدار دریافت شده مربوط به کدام سنسور است. برای این منظور می‌توان فایل ذخیره شده در کارت حافظه را در برنامه اکسل وارد کرده و سپس از طریق تابع^۱های موجود بیشترین مقدار هر سطر را که مربوط به اطلاعات ۱۶ سنسور در هر سری اندازه‌گیری است را به دست آورد تا مشخص شود که مربوط به کدام سنسور است. با متصل کردن این مقدار ماکزیمم‌ها می‌توان به مسیر تقریبی خورشید پی برد که مسیر خورشید از سمت چه سنسوری به سمت کدام سنسور حرکت کرده است. واضح است که در مناطق مختلف مسیر بدست آمده متفاوت خواهد بود. همچنین مسیر بدست آمده در فصول مختلف از سال از حرکت خورشید در یک منطقه، به دلیل تغییر موقعیت زمین نسبت به خورشید

^۱ =IF(MOD(ROW();5)=0;CHAR(MATCH(MAX(B1;Q1);B1;Q1;0)+65;"")

متفاوت خواهد بود. برای به دست آوردن مسیر دقیق لازم است که تمام سنسورها و مقاومت‌های استفاده شده، دقیق و یکسان باشند.

برای بررسی تغییرات اقلیمی، نیاز است که این اطلاعات به طور سالیانه جمع‌آوری و ذخیره شده و سپس با یکدیگر مقایسه شوند. این اطلاعات برای نتیجه‌گیری کافی نیستند بلکه بهتر است در کنار اطلاعات دیگری همچون تصاویر ماهواره‌ای و اطلاعات ثبت شده توسط سازمان هواشناسی بررسی شوند.

برای بدنه، از متداول‌ترین نوع فیلامنت استفاده شده است که علاوه بر کیفیت و استحکام مناسب، قیمت مناسبی نیز دارد و به راحتی نیز با تمام پرینترهای سه‌بعدی قابل پرینت است. نکته قابل توجه در مورد این نوع فیلامنت، عدم توانایی ایجاد تغییرات یا سوراخ‌کاری پس از پرینت قطعه توسط مته یا دریل است. زیرا در اثر گرمای ایجاد شده توسط آنها، قطعه در آن ناحیه ذوب شده و تغییر حالت می‌دهد. با تست طرح در هوای بارانی، آب به داخل قطعات نفوذی نداشت اما بهتر است برای اطمینان از اسپری‌های پوششی و یا رنگ مناسب روی طرح استفاده شود.

از آنجایی که بنا بود آفتاب نگار طراحی شده از هر جهت بهینه باشد، اندازه و حجم طرح نیز از این قاعده مستثنی نبودند. برای سیستم پردازنده، از کوچک‌ترین عضو خانواده آردوینو استفاده شد. آردوینو پرو مینی در عین حال که اندازه کوچک و وزن کمی داشت، توانایی‌های لازم جهت رفع نیازهای طرح را نیز دارا بود که قیمت پایین آن، این برد را در صدر گزینه‌ها برای سیستم پردازنده قرار داد. این نوع آردوینو در دو مدل ۳.۳ ولت و ۵ ولت

وجود دارد. اگر منبع تغذیه غالب مدار ۵ ولت باشد و قصد استفاده از نوع ۳.۳ ولتی این نوع برد وجود دارد، باید از رگولاتور برای کاهش ولتاژ در ورودی منبع تغذیه این برد استفاده شود. برای اندازه‌گیری نور محیط سنسورهای مختلفی وجود دارد که ساده‌ترین و ارزان‌ترین آنها، فوتورزیستورها هستند. با توجه به اینکه در این طرح از تعداد سنسورهای زیادی استفاده شده است، پس مسئله قیمت سنسورها از اهمیت بیشتری برخوردار است. از طرفی اندازه‌ی سنسورهای انتخابی نیز مهم است که فوتورزیستورها از این نظر نیز بسیار مناسب و سازگار با اهداف طرح هستند. چالشی که هنگام کار و استفاده از این نوع سنسورها وجود داشت، متفاوت بودن حساسیت و تغییرات مقدار دو فوتورزیستور مشابه نسبت به تغییر وضعیت روشنایی یکسان بود. مشخصه خروجی این نوع سنسور باعث عدم توانایی در کالیبره کردن آنها بصورت نرم‌افزاری بود. البته با زیاد کردن تعداد سنسورها و همچنین کوتاه کردن زمان بین دوره‌های اندازه‌گیری، توانستیم تا حدی این مشکلات را پوشش دهیم.

چالش دیگری که در این پروژه وجود داشت، محدوده حساسیت سنسورها به روشنایی بود. همانطور که در بخش‌های قبل عنوان شده است، فوتورزیستورها در اصل مقاومت هستند که با تغییر روشنایی پیرامون آنها، مقدار مقاومت فوتورزیستورها نیز تغییر می‌کند. همچنین گفته شد که خواندن مقدار سنسورها از طریق تقسیم ولتاژ بین فوتورزیستور و مقاومت ۱ کیلو اهمی صورت می‌پذیرد. زمانی که روشنایی محیط زیاد شود، مقدار مقاومت فوتورزیستور کاهش می‌یابد. محدوده اندازه‌گیری روشنایی تا جایی است که مقاومت فوتورزیستور به صفر میل می‌کند و تمام ولتاژ اعمال شده به پایه آن، در پایه دیگر آن که محل خروجی این سنسور است، قرار بگیرد. در این صورت دیگر هر چه مقدار روشنایی محیط بیشتر شود، در

خروجی تغییری مشاهده نخواهد شد. برای رفع این مورد می‌توان از یک فیلتر رنگی بر روی سنسورها استفاده نمود تا مقدار کلی خروجی سنسورها کمتر شود. با توجه به مشخصه خروجی غیرخطی از ایجاد فیلترهای برنامه‌نویسی برای این مورد نمی‌توان استفاده نمود.

با توجه به اهداف اولیه تعیین شده برای طرح و همچنین با بررسی اطلاعات خروجی آفتاب نگار طراحی و ساخته شده، می‌توان به این نتیجه رسید که اهداف مورد نظر تا حدود مناسبی به وقوع پیوسته‌اند.

پیشنهادهای

۱- با توجه به اینکه فوتورزیستورها نوعی مقاومت با مشخصه خروجی غیرخطی هستند، عموماً برای اهدافی با خروجی صفر و یک مورد استفاده قرار می‌گیرند که صرفاً مشخص کننده روشنایی یا تاریکی محیط خواهد بود. به منظور اندازه‌گیری مقدار روشنایی محیط، برای رسیدن به نتایج عالی، بهتر است از سنسورهای مخصوص اندازه‌گیری روشنایی استفاده شود.

۲- برای ثبت و انتقال اطلاعات سنسورها، بهتر است از پروتکل‌های بی‌سیم مانند لورا و یا زیگبی استفاده نمود تا علاوه بر صرفه جویی در زمان، برای آفتاب نگارهایی که در مناطق غیرشهری قرار دارند کار انتقال اطلاعات راحت‌تر صورت بپذیرد.

۳- همانطور که در فصل دوم عنوان شد، بدنه سیستم طراحی شده از جنس فیلامنت PLA است. این نوع فیلامنت در برابر تابش شدید خورشید که دمای سطحی آن را به بالای ۶۰ درجه سلسیوس برسد مقاوم نبوده و تغییر شکل می‌دهد. پیشنهاد می‌شود

برای استفاده از این دستگاه در مناطق گرمسیر که با تابش شدید خورشید همراه است، برای بدنه از جنس رزین که دارای مقاومت بالایی در برابر گرما و ضربه است، استفاده شود.

۴- برای اینکه یک پردازنده مطمئن از هر نظر داشته باشیم، بهتر است از میکروهای خانواده ARM استفاده کنیم.

۵- با توجه به تعداد بالای سنسورها و اهمیت اتصال درست و بی نقص این سنسورها به مقاومت ها و منابع تغذیه، بهتر است یک برد PCB طراحی شده و پایه سنسورها و مقاومت ها روی برد به یکدیگر متصل شوند تا علاوه بر اتصال درست، هرگونه تعمیر و تعویض در صورت نیاز نیز به خوبی صورت بگیرد.

۶- می توان سنسورهای دیگری مانند سنسور سنجش نور فرابنفش، سنسور اندازه گیری دما و ... را نیز به این طرح در جهت تایید و تکمیل اطلاعات دریافتی سنسورهای اصلی اضافه نمود.

۷- برای مواقعی که ممکن است برق اصلی دستگاه قطع شود، بهتر است یک منبع تغذیه پشتیبان جهت ادامه ثبت اطلاعات و همچنین اطلاع رسانی تعبیه و در نظر گرفته شود.

البته ذکر این نکته ضروری است که با اعمال هر کدام از پیشنهادات ذکر شده، هزینه نهایی سیستم افزایش چشمگیری خواهد داشت.

مراجع

- 1- https://en.wikipedia.org/wiki/Campbell–Stokes_recorder
- 2- <https://en.wikipedia.org/wiki/Pyranometer>
- 3- <https://www.mrfilament.com/guide/%D9%81%DB%8C%D9%84%D8%A7%D9%85%D9%86%D8%AA-abs/>
- 4- <https://store.arduino.cc/usa/arduino-pro-mini>