

به نام خدا



دانشگاه تهران

پردیس دانشکده‌گان فنی

دانشکده مهندسی برق و کامپیوتر

گزارش پروژه کارشناسی

پایان‌نامه برای دریافت درجه کارشناسی

در رشته مهندسی برق گرایش بایوالکتریک

طراحی و پیاده‌سازی سامانه پوشیدنی IoT برای مانیتورینگ بلادرنک علائم

حیاتی و تحلیل هوشمند داده‌ها

توحید بهشتی__810100100

استاد راهنما : دکتر سیدکمال الدین ستاره دان

نیمسال اول 1405_1404

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

تعهدنامه اصالت اثر

باسمه تعالی

اینجانب توحید بهشتی تأیید می‌کنم که مطالب مندرج در این پایان‌نامه حاصل تلاش اینجانب است و به دستاوردهای پژوهشی دیگران که در این نوشته از آنها استفاده شده است مطابق مقررات ارجاع گردیده است. این پایان‌نامه قبلاً برای احراز هیچ مدرک هم سطح یا بالاتر ارائه نشده است. کلیه حقوق مادی و معنوی این اثر متعلق به دانشکده فنی دانشگاه تهران می‌باشد.

نام و نام خانوادگی دانشجو : توحید بهشتی



امضای دانشجو :

با توجه به افزایش نیاز به پایش مداوم وضعیت سلامت افراد، به ویژه در بیماران قلبی و تنفسی، توسعه سامانه‌های هوشمند و کم‌هزینه برای مانیتورینگ علائم حیاتی اهمیت زیادی پیدا کرده است. در این پروژه، یک سامانه پایش سلامت (Health monitoring system) مبتنی بر اینترنت اشیاء (IoT) طراحی و پیاده‌سازی شد که قادر است برخی از علائم حیاتی مهم شامل ضربان قلب (BPM)، سطح اکسیژن خون (SpO_2)، دمای بدن و همچنین دمای و رطوبت محیط بیمار را به صورت بلادرنگ اندازه‌گیری و نمایش دهد.

در این سامانه از میکروکنترلر ESP32 به عنوان هسته اصلی پردازش و ارتباط بی‌سیم استفاده شده است. سنسور MAX30100 برای اندازه‌گیری ضربان قلب و میزان اکسیژن خون، سنسور DS18B20 برای دمای بدن و سنسور DHT11 برای اندازه‌گیری شرایط محیطی (در واقع دما و رطوبت اتاق) به کار گرفته شدند. داده‌های دریافتی پس از پردازش اولیه در میکروکنترلر، از طریق شبکه Wi-Fi در قالب یک وب سرور تعبیه شده (Embedded Web Server) در اختیار کاربر قرار می‌گیرند و به صورت لحظه‌ای در یک داشبورد تحت وب قابل مشاهده هست.

در فرآیند پیاده‌سازی، چالش‌های بسیاری از جمله ناپایداری ارتباط I2C، خطای خواندن دمای سنسور DS18B20 و وابستگی عملکرد سنسور MAX30100 به ساختار حلقه اصلی برنامه و مشکلات اختلاف نرخ ارسال داده توسط هر کدام از سنسورها، شناسایی و برطرف شد. در نهایت، سامانه توسعه یافته توانست داده‌های حیاتی را با پایداری مناسب دریافت و به صورت برخط نمایش دهد. این پروژه نمونه‌ای از یک سیستم پایش سلامت کم‌هزینه و قابل توسعه است که می‌تواند مبنایی برای گسترش به سامانه‌های ابری، ذخیره‌سازی بلندمدت داده‌ها و تحلیل هوشمند علائم حیاتی در آینده باشد.

کلمات کلیدی: اینترنت اشیاء (IoT)، پایش سلامت، ESP32، MAX30100، سیستم غیرمستودکننده، وب سرور داخلی، تشخیص ناهنجاری

فهرست

چکیده.....	4
فصل اول : مقدمه و بیان مساله.....	7
1-1- مقدمه.....	7
1-2- تاریخچه‌ای از موضوع تحقیق.....	8
1-3- شرح مسئله تحقیق.....	9
1-4- تعریف موضوع تحقیق.....	9
1-5- اهداف و آرمان‌های کلی تحقیق.....	10
1-6- روش انجام تحقیق.....	11
1-7- ساختار پایان‌نامه.....	12
فصل دوم: مفاهیم اولیه و پیش زمینه.....	14
1-1- اینترنت اشیاء و سامانه‌های پایش سلامت.....	14
2-2- مبانی سنجش علائم حیاتی و سنسورهای مورد استفاده.....	15
2-3- معماری نرم‌افزارهای بلادرنگ و برنامه‌نویسی غیرمسدودکننده.....	18
فصل سوم: طراحی و پیاده سازی سیستم.....	19
3-1- مقدمه.....	19
3-2- معماری کلی سیستم و مدل لایه‌ای.....	19
3-3- طراحی سخت‌افزار و پیاده‌سازی فیزیکی.....	20
3-4- طراحی نرم‌افزار و معماری غیرمسدودکننده.....	22
3-5- طراحی الگوریتم‌های پردازش سیگنال و سیستم هشدار.....	22
3-6- ابزارهای توسعه و محیط پیاده‌سازی.....	24
3-7- معیارهای ارزیابی طراحی.....	24
3-8- تحلیل طراحی و جمع‌بندی.....	25
فصل چهارم: پیاده سازی سامانه.....	27
4-1- مقدمه.....	27
4-2- پیاده‌سازی سخت‌افزار و یکپارچه‌سازی ماژول‌ها.....	27
4-3- پیاده‌سازی نرم‌افزار و ساختار اجرایی نهایی.....	28
4-4- پیاده‌سازی سیستم هشدار و منطق تصمیم‌گیری.....	29
4-5- پیاده‌سازی رابط کاربری تحت وب.....	30
4-6- اجرای سناریوهای آزمون و بررسی عملکرد سیستم.....	32

33	4-7- ارزیابی پایداری در کار مداوم.....
33	4-8- تحلیل رفتار سیستم و محدودیت‌های نسخه فعلی.....
34	4-9- جمع‌بندی فصل پیاده‌سازی.....
35	فصل پنجم: نتیجه‌گیری و پیشنهادهای توسعه آینده.....
35	5-1- نتیجه‌گیری.....
36	5-2- پیشنهادهای توسعه آینده.....
37	5-3- جمع‌بندی نهایی.....
38	مراجع.....
39	پیوست‌ها.....
39	پیوست 1- جدول آستانه‌های سیستم هشدار.....
40	پیوست 2- جدول اتصالات ماژول‌ها به ESP32 (pin Assignment).....
41	پیوست 3- دسترسی به کد پروژه.....

فصل اول : مقدمه و بیان مساله

1-1- مقدمه

در دهه‌های اخیر، پیشرفت هم‌زمان فناوری‌های دیجیتال، سیستم‌های نهفته و ارتباطات بیسیم، زمینه را برای به وجود آمدن نسل جدید از سامانه‌های هوشمند فراهم کرده است که قادرند داده‌های محیطی و زیستی را به‌صورت پیوسته جمع‌آوری، پردازش و منتقل کنند. در این میان، حوزه پایش سلامت مبتنی بر اینترنت اشیا (IoT-Based Health Monitoring) به دلیل ارتباط مستقیم با کیفیت زندگی انسان، از اهمیت ویژه‌ای برخوردار شده است. افزایش امید به زندگی، شیوع بیماری‌های مزمن قلبی و تنفسی، و نیاز به نظارت مستمر بر وضعیت بیماران، ضرورت توسعه سامانه‌های پایش بلادرنگ و کم‌هزینه را بیش از پیش نمایان ساخته است.

در طول دوران تحصیل در گرایش بایوالکتریک، همواره علاقه مند بودم مفاهیم تئوری و اکادمیک مانند پردازش سیگنال‌های زیستی، طراحی مدارهای الکتریکی و سیستم‌های نهفته را در قالب یک سیستم واقعی و کاربردی پیاده‌سازی کنم. به همین دلیل در این پروژه تلاش کردم یک سیستم پایش سلامت بر پایه اینترنت اشیا طراحی کنم که هم از نظر سخت‌افزاری و هم از نظر نرم‌افزاری به‌صورت کامل توسط خودم توسعه داده شود و در عین حال مقرون به صرفه و قابل توسعه بیشتر باشد.

پایش مستمر علائم حیاتی نظیر ضربان قلب، سطح اکسیژن خون و دمای بدن، نه‌تنها در محیط‌های بیمارستانی بلکه در کاربردهای خانگی و مراقبت از راه دور نیز اهمیت دارد. با این حال، بسیاری از سامانه‌های موجود یا دارای هزینه بالا هستند یا به زیرساخت‌های پیچیده ابری وابستگی کامل دارند. از سوی دیگر، پیاده‌سازی عملی چنین سیستم‌هایی با چالش‌های متعددی در سطح سخت‌افزار و نرم‌افزار همراه است؛ چالش‌هایی که در بسیاری از مطالعات صرفاً به‌صورت تئوریک مطرح شده و کمتر در قالب یک تجربه مهندسی کامل مورد بررسی قرار گرفته‌اند.

پروژه حاضر با هدف طراحی و پیاده‌سازی یک سامانه پوشیدنی مبتنی بر ESP32 برای پایش بلادرنگ علائم حیاتی شکل گرفته است؛ سامانه‌ای که علاوه بر اندازه‌گیری پارامترهای زیستی، بر پایداری معماری نرم‌افزاری، مدیریت زمان‌بندی سنسورها، حذف نویز و ارائه بازخورد هوشمند در صورت بروز ناهنجاری تمرکز دارد. در این تحقیق تلاش شده است که مفاهیم تئوریک سیستم‌های نهفته، پردازش سیگنال‌های زیستی و اینترنت اشیا در قالب یک سامانه عملی، کم‌هزینه و قابل توسعه تجمیع شوند.

2-1- تاریخچه‌ای از موضوع تحقیق

پایش علائم حیاتی سابقه‌ای طولانی در پزشکی دارد. از دستگاه‌های الکتروکاردیوگراف (ECG) در دهه‌های ابتدایی قرن بیستم گرفته تا مانیتورهای چندپارامتری بیمارستانی، همواره هدف، ثبت دقیق سیگنال‌های زیستی برای تشخیص و پایش وضعیت بیمار بوده است. با ورود میکروکنترلرها و سیستم‌های دیجیتال در دهه‌های اخیر، امکان کوچک‌سازی تجهیزات و انتقال داده‌ها به صورت الکترونیکی فراهم شد. ظهور فناوری اینترنت اشیاء نقطه عطفی در این مسیر محسوب می‌شود. IoT امکان اتصال اشیاء فیزیکی به شبکه را فراهم می‌سازد و این قابلیت باعث شد مفهوم “پایش از راه دور” از یک ایده تحقیقاتی به یک راهکار عملی تبدیل شود. در بسیاری از مطالعات دانشگاهی، ساختار کلی سیستم شامل یک سنسور زیستی، یک میکروکنترلر مجهز به ماژول ارتباطی و یک پلتفرم ابری برای ذخیره‌سازی و تحلیل داده‌ها معرفی شده است.

با وجود پیشرفت‌های چشمگیر، بررسی دقیق‌تر این حوزه نشان می‌دهد که تمرکز بسیاری از پروژه‌ها بر ارسال داده به فضای ابری بوده و کمتر به چالش‌های عملی نظیر ناپایداری ارتباط I2C، تأثیر ساختار حلقه برنامه بر کیفیت داده‌های سنسور PPG، یا تداخل زمانی میان Wi-Fi و نمونه‌برداری سنسورها پرداخته شده است. همچنین بسیاری از سامانه‌های تجاری موجود اگرچه از نظر عملکرد کامل هستند، اما ساختار آن‌ها بسته بوده و امکان توسعه یا سفارشی‌سازی محدود دارند.

در این میان، سنسورهای مبتنی بر فوتوپلتیسموگرافی (PPG) مانند MAX30100 و MAX30102 امکان اندازه‌گیری ضربان قلب و SpO_2 را در قالب ماژول‌های کوچک و کم‌هزینه فراهم کرده‌اند. با این حال، عملکرد این سنسورها به شدت به شرایط تماس، حذف نویز حرکتی و نحوه مدیریت زمان‌بندی نرم‌افزار وابسته است. این وابستگی در عمل نشان می‌دهد که طراحی یک سامانه پایدار تنها با اتصال صحیح قطعات حاصل نمی‌شود، بلکه نیازمند درک عمیق تعامل میان سخت‌افزار و نرم‌افزار است.

3-1- شرح مسئله تحقیق

مسئله اصلی این تحقیق، طراحی یک سامانه پایش سلامت کم‌هزینه و مستقل است که بتواند علائم حیاتی را به صورت بلادرنگ اندازه‌گیری و نمایش دهد و در عین حال از نظر معماری نرم‌افزاری و سخت‌افزاری پایدار باشد.

در مراحل اولیه توسعه، چالش‌های متعددی مشاهده شد. در بخش سخت‌افزار، سنسور DS18B20 در صورت عدم استفاده از مقاومت Pull-up مقدارهای غیرواقعی مانند 85°C یا 127°C تولید می‌کرد. همچنین در صورت عدم اجرای تابع مقداردهی اولیه در بخش setup، میکروکنترلر قادر به شناسایی سنسور روی باس One-Wire نبود. این موضوع نشان داد که حتی جزئیات کوچک در طراحی می‌توانند منجر به خطاهای سیستمی شوند.

در بخش سنسور MAX30100، مشکل پیچیده‌تری مشاهده شد. این سنسور برای عملکرد صحیح نیازمند فراخوانی پیوسته تابع update در حلقه اصلی برنامه است. استفاده از توابع مسدودکننده مانند delay پردازش سنگین وب‌سرور باعث توقف اجرای الگوریتم داخلی تشخیص ضربان و در نتیجه تولید خروجی صفر می‌شد. علاوه بر آن، سیگنال‌های PPG به حرکت دست و نحوه قرارگیری انگشت بسیار حساس بودند و حذف نویز و جهش‌های ناگهانی نیازمند طراحی فیلتر مناسب بود.

از سوی دیگر، اختلاف نرخ نمونه‌برداری سنسورها مسئله مهمی ایجاد می‌کرد DHT11 و DS18B20 و MAX30100 هر یک دارای زمان پاسخ و نرخ به‌روزرسانی متفاوتی هستند. در صورت مدیریت نادرست زمان‌بندی، خواندن یک سنسور می‌توانست باعث اختلال در عملکرد دیگری شود. بنابراین مسئله اصلی به این صورت تعریف شد:

چگونه می‌توان یک سامانه پایش سلامت مبتنی بر ESP32 طراحی کرد که ضمن اندازه‌گیری پایدار چند پارامتر زیستی، از معماری نرم‌افزاری غیرممسدودکننده استفاده کند، نسبت به خطاهای سخت‌افزاری مقاوم باشد و بتواند در صورت بروز ناهنجاری، بازخورد هوشمند ارائه دهد؟

4-1- تعریف موضوع تحقیق

موضوع این تحقیق، طراحی و پیاده‌سازی یک سامانه پوشیدنی IoT برای مانیتورینگ بلادرنگ علائم حیاتی و تحلیل اولیه داده‌ها است. این سامانه شامل چهار لایه اصلی است: لایه اندازه‌گیری (سنسورها)،

لایه پردازش (میکروکنترلر ESP32)، لایه ارتباطی (Web server و Wi-Fi داخلی) لایه نمایش (داشبورد تحت وب)

در این پروژه، پارامترهای زیر اندازه‌گیری و نمایش داده می‌شوند:

- ضربان قلب (BPM)
- سطح اکسیژن خون (SpO_2)
- دمای بدن
- دمای محیط
- رطوبت نسبی محیط

علاوه بر نمایش داده‌ها، برای هر پارامتر بازه‌های فیزیولوژیک تعریف شده و یک سیستم هشدار چندسطحی شامل حالت‌های OK، Warning، Critical و Sensor Fault پیاده‌سازی گردیده است. همچنین الگوریتم‌های نرم‌افزاری برای حذف جهش‌های ناگهانی، فیلتر میانگین نمایی (EMA) و تشخیص برداشتن انگشت از سنسور PPG توسعه داده شده‌اند.

تمرکز اصلی این تحقیق نه صرفاً ساخت یک مدار، بلکه طراحی یک معماری مهندسی پایدار است که تعامل میان سخت‌افزار، زمان‌بندی نرم‌افزار و نمایش بلادرنگ داده‌ها را به‌صورت هماهنگ مدیریت کند.

5-1- اهداف و آرمان‌های کلی تحقیق

هدف اصلی این تحقیق، طراحی و پیاده‌سازی یک سامانه پایش سلامت مبتنی بر اینترنت اشیاء است که بتواند چندین پارامتر زیستی و محیطی را به‌صورت بلادرنگ اندازه‌گیری، پردازش و نمایش دهد و در عین حال از نظر معماری نرم‌افزاری و سخت‌افزاری پایدار و قابل توسعه باشد. این هدف کلی به چند هدف عملیاتی و مهندسی قابل تفکیک است.

نخست، ایجاد یک ساختار ماژولار که در آن لایه‌های اندازه‌گیری، پردازش، ارتباط و نمایش تا حد امکان مستقل از یکدیگر طراحی شوند تا افزودن یا جایگزینی سنسورها در آینده نیازمند بازطراحی کامل سیستم نباشد. چنین معماری‌ای، سامانه را از یک نمونه آزمایشگاهی ساده به یک پلتفرم قابل توسعه تبدیل می‌کند.

دوم، طراحی یک معماری نرم‌افزاری غیرمسدودکننده (Non-Blocking) برای مدیریت هم‌زمان چند سنسور با نرخ‌های نمونه‌برداری متفاوت. یکی از آرمان‌های کلیدی این تحقیق، اثبات این نکته است که در سیستم‌های بلادرنگ، ساختار زمان‌بندی کد و حذف توابع مسدودکننده نقش تعیین‌کننده‌ای در پایداری داده‌ها دارند.

سوم، افزایش کیفیت سیگنال‌های زیستی از طریق به‌کارگیری تکنیک‌های پردازش اولیه نظیر حذف جهش‌های ناگهانی (Spike Rejection)، فیلتر میانگین نمایی (EMA) و الگوریتم تشخیص عدم تماس انگشت در سنسور PPG هدف در این بخش، نزدیک شدن رفتار سیستم به یک مانیتور واقعی و جلوگیری از نمایش مقادیر گمراه‌کننده است.

چهارم، تعریف بازه‌های فیزیولوژیک برای هر پارامتر و پیاده‌سازی یک سامانه هشدار چندسطحی. در بسیاری از پروژه‌های دانشجویی، داده‌ها صرفاً نمایش داده می‌شوند؛ اما در یک سامانه پایش سلامت، ارزش واقعی زمانی ایجاد می‌شود که سیستم بتواند خروج از محدوده طبیعی را تشخیص داده و بازخورد مناسب ارائه دهد. از این رو، تعریف سطوح OK، Warning، Critical و Sensor Fault یکی از اهداف کاربردی این پروژه بوده است.

در نهایت، یکی از آرمان‌های مهم این تحقیق، طراحی سیستمی مستقل از زیرساخت‌های ابری است؛ سیستمی که حتی در نبود اینترنت سراسری نیز بتواند در یک شبکه محلی عملکرد کامل داشته باشد. تجربه عملی محدودیت دسترسی به اینترنت در طول اجرای پروژه نشان داد که وابستگی کامل به سرویس‌های خارجی می‌تواند یک نقطه ضعف جدی در سامانه‌های IoT باشد. بنابراین استقلال عملیاتی، یکی از اصول طراحی این پروژه در نظر گرفته شد.

6-1- روش انجام تحقیق

روش انجام این تحقیق مبتنی بر رویکرد طراحی و پیاده‌سازی مهندسی بوده است. فرآیند توسعه در چند مرحله متوالی و تکرارشونده انجام شد که شامل مطالعه، طراحی اولیه، پیاده‌سازی، آزمون، عیب‌یابی و بهینه‌سازی بوده است.

در مرحله نخست، مطالعه منابع مرتبط با سامانه‌های پایش سلامت مبتنی بر IoT و بررسی دیتاشیت سنسورها و میکروکنترلر انجام شد. در این مرحله، معیارهایی نظیر هزینه، در دسترس بودن، دقت اندازه‌گیری و سازگاری ارتباطی برای انتخاب ماژول‌ها در نظر گرفته شد. نتیجه این بررسی‌ها انتخاب

ESP32 به عنوان هسته پردازشی و سنسورهای MAX30100، DS18B20 و DHT11 به عنوان منابع داده بود.

در مرحله دوم، طراحی سخت افزار و پیاده سازی اتصالات انجام شد. در این مرحله، چالش های عملی نظیر نیاز به مقاومت Pull-up در پروتکل One-Wire، طول سیم های I2C و کیفیت تغذیه مورد بررسی قرار گرفت. خطاهای مشاهده شده، به صورت مرحله ای تحلیل و اصلاح شدند تا یک ساختار سخت افزاری پایدار حاصل شود.

مرحله سوم به توسعه نرم افزار اختصاص یافت. در ابتدا ساختارهای ساده مبتنی بر delay مورد آزمایش قرار گرفت، اما به دلیل بروز اختلال در خواندن سنسور MAX30100، معماری برنامه به صورت کامل بازطراحی شد. استفاده از تابع millis برای مدیریت زمان، جداسازی وظایف هر سنسور و اجرای پیوسته تابع update در حلقه اصلی، هسته معماری نهایی را شکل داد.

در ادامه، برای بهبود کیفیت داده ها، الگوریتم های نرم افزاری پردازش اولیه پیاده سازی شدند. حذف جهش های ناگهانی، میانگین گیری نمایی و مکانیزم تشخیص عدم حضور انگشت از طریق تحلیل فاصله زمانی بین ضربان ها (Beat Timeout) بخشی از این فرآیند بود. سپس سیستم هشدار چندسطحی طراحی و در داشبورد تحت وب یکپارچه شد.

در مرحله نهایی، سیستم در سناریوهای مختلف شامل حالت استراحت، پس از فعالیت بدنی سبک، قطع تماس سنسور و کار مداوم چندساعته مورد آزمون قرار گرفت. داده های مشاهده شده با مقادیر فیزیولوژیک مرجع مقایسه و پایداری عملکرد ارزیابی شد. این رویکرد تکرار شونده آزمون و اصلاح، نقش کلیدی در رسیدن به نسخه نهایی سامانه ایفا کرد.

7-1- ساختار پایان نامه

این پایان نامه در پنج فصل تدوین شده است.

در فصل اول، مقدمه، پیشینه موضوع، تعریف مسئله، اهداف تحقیق و روش انجام آن ارائه شد تا چارچوب کلی پژوهش مشخص شود.

فصل دوم به بررسی مفاهیم نظری و مبانی فنی مرتبط با پروژه اختصاص دارد. در این فصل، اصول اینترنت اشیاء، ساختار سیستم های نهفته، مبانی سنسورهای مبتنی بر PPG، پروتکل های ارتباطی مانند I2C و One-Wire و اصول طراحی نرم افزار غیرمسدود کننده مورد بررسی قرار می گیرد.

فصل سوم به طراحی سخت‌افزار و معماری کلی سیستم می‌پردازد. در این فصل، انتخاب میکروکنترلر و سنسورها، نحوه اتصال ماژول‌ها، ملاحظات تغذیه و چالش‌های عملی پیاده‌سازی تشریح می‌شود.

فصل چهارم شامل طراحی نرم‌افزار، پیاده‌سازی الگوریتم‌ها، مدیریت زمان‌بندی، توسعه وب‌سرور داخلی و سیستم هشدار است. همچنین نتایج آزمون‌های عملی و ارزیابی عملکرد سامانه در این فصل ارائه می‌شود.

در فصل پنجم، جمع‌بندی کلی پروژه، دستاوردهای مهندسی، محدودیت‌ها و پیشنهادهای توسعه آینده مطرح می‌گردد.

این ساختار به‌گونه‌ای تنظیم شده است که خواننده ابتدا با مسئله و ضرورت آن آشنا شود، سپس با مبانی نظری و طراحی فنی مواجه گردد و در نهایت نتایج عملی و تحلیل عملکرد سامانه را مشاهده کند. چنین ترتیبی، مسیر منطقی شکل‌گیری ایده تا تحقق عملی آن را به‌صورت شفاف نمایش می‌دهد.

فصل دوم: مفاهیم اولیه و پیش زمینه

1-1- اینترنت اشیاء و سامانه‌های پایش سلامت

اینترنت اشیاء (Internet of Things - IoT) به شبکه‌ای از اشیای فیزیکی اطلاق می‌شود که از طریق حسگرها، ماژول‌های پردازشی و بسترهای ارتباطی قادر به جمع‌آوری، تبادل و تحلیل داده هستند بطوری که ویژگی‌های اصلی سیستم معرفی شده را دارا باشند.

در حوزه سلامت، IoT زمینه‌ساز شکل‌گیری مفهوم پایش سلامت از راه دور (Remote Health monitoring) شده است. در این رویکرد، داده‌های حیاتی بیمار بدون نیاز به حضور دائمی در مراکز درمانی ثبت و منتقل می‌شوند. چنین سامانه‌هایی معمولاً شامل سه لایه اصلی هستند: لایه حسگرها (Sensing Layer)، لایه پردازش و کنترل (Processing Layer) و لایه ارتباط و نمایش (User Interface).

در سامانه‌های پزشکی، پایداری و دقت اهمیت ویژه‌ای دارند. برخلاف بسیاری از کاربردهای عمومی IoT، در پایش سلامت هرگونه خطای اندازه‌گیری می‌تواند منجر به تفسیر اشتباه وضعیت بیمار شود. بنابراین علاوه بر سخت‌افزار مناسب، طراحی نرم‌افزار پایدار و مدیریت زمان‌بندی از اهمیت اساسی برخوردار است.

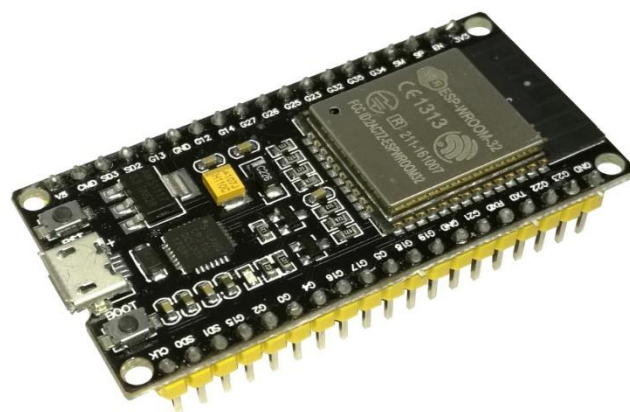
یکی از چالش‌های مهم در سامانه‌های IoT پزشکی، وابستگی به زیرساخت‌های ابری است. اگرچه استفاده از Cloud Platform ها امکان ذخیره‌سازی و تحلیل بلندمدت داده‌ها را فراهم می‌کند، اما در بسیاری از شرایط عملی (به‌ویژه در شبکه‌های محلی یا هنگام قطعی اینترنت) وابستگی کامل به فضای ابری می‌تواند عملکرد سیستم را مختل کند. به همین دلیل، معماری‌های مبتنی بر Web Server داخلی که پردازش و نمایش را در همان گره محلی انجام می‌دهند، رویکردی پایدارتر و مستقل‌تر محسوب می‌شوند. در پروژه حاضر، سامانه طراحی شده در همین چارچوب تعریف می‌شود؛ به‌طوری‌که ESP32 علاوه بر نقش پردازشی، وظیفه ایجاد یک Embedded Web Server را نیز بر عهده دارد.

انتخاب میکروکنترلر

در مرحله انتخاب یونیت پردازشی ، گزینه‌هایی مثل STM32 ، Raspberry Pi Pico و ESP32 مورد بررسی قرار گرفتند. با تحقیق‌هایی که انجام دادم و مقایسه امکانات، مشخص شد که ESP32 ترکیبی مناسب از توان پردازش بالا ، ارتباط Wi-fi داخلی و هزینه پایین را ارائه می دهد.

بر اساس Manual شرکت Espressif ، این میکروکنترلر دارای پردازنده دو هسته‌ای، پشتیبانی از Wi-fi و Bluetooth و همچنین رابط‌های ارتباطی متنوعی مانند I2C و SPI و UART هست. مهم‌ترین مزیت آن برای این پروژه وجود Wi-fi داخلی بود که نیاز به مازول جداگانه Wi-fi را حذف می کرد و پیچیدگی مدار را کاهش می داد.

همچنین در پباده سازی عملی مشخص شد که ESP32 توانایی مدیریت هم زمان خواندن سنسورهای I2C و راه اندازی یک وب سرور داخلی را دارد، البته به شرط آنکه ساختار برنامه به درستی زمان بندی شود.



شکل 1. ESP32 MCU

2-2- مبانی سنجش علائم حیاتی و سنسورهای مورد استفاده

فوتوپلتیسموگرافی (PPG) و اندازه‌گیری BPM و SpO_2

سنسور MAX30100 بر پایه تکنیک فوتوپلتیسموگرافی (Photoplethysmography – PPG) عمل می کند. در این روش، با تابش نور قرمز و مادون قرمز به بافت و اندازه‌گیری نور بازتاب‌شده، تغییرات حجم خون در مویرگ‌ها تشخیص داده می شود. این تغییرات به صورت یک سیگنال نوسانی ظاهر می شوند

برای محاسبه ضربان قلب (BPM)، فاصله زمانی بین قله‌های متوالی سیگنال PPG اندازه‌گیری می‌شود. در محاسبه SpO_2 ، نسبت جذب نور در دو طول موج مختلف مورد استفاده قرار می‌گیرد، زیرا هموگلوبین اکسیژن‌دار و بدون اکسیژن ضرایب جذب متفاوتی دارند.

با وجود سادگی ظاهری، سیگنال PPG بسیار حساس به حرکت (Motion Artifact)، فشار تماس انگشت و نور محیط است. بنابراین پردازش نرم‌افزاری و زمان‌بندی صحیح نمونه‌برداری نقش حیاتی در کیفیت خروجی دارد. در صورت اختلال در اجرای مداوم الگوریتم داخلی سنسور، خروجی ممکن است به صفر میل کند یا مقادیر غیرواقعی تولید شود.

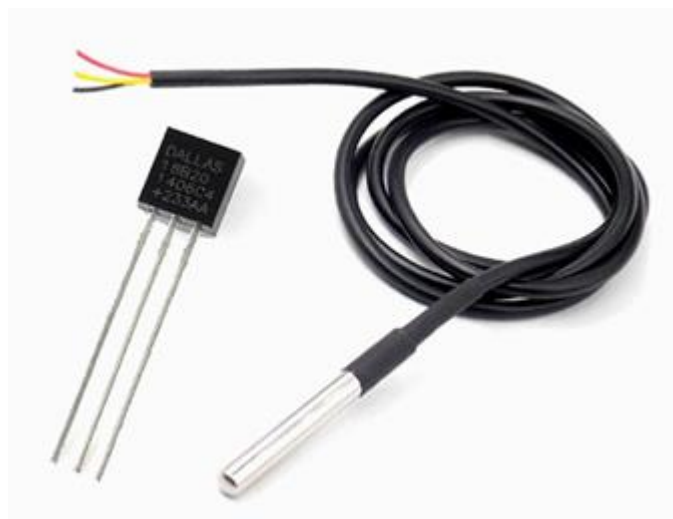


شکل 2. MAX30100 (BPM & SpO_2 Sensor)

سنجش دمای بدن با DS18B20

سنسور DS18B20 یک سنسور دیجیتال مبتنی بر پروتکل One-Wire است که امکان اندازه‌گیری دما با دقت مناسب را فراهم می‌کند. خروجی دیجیتال این سنسور باعث کاهش تأثیر نویز آنالوگ می‌شود. با این حال، پروتکل One-Wire نیازمند وجود مقاومت Pull-up روی خط داده است و در صورت عدم رعایت این موضوع، مقادیر خطا مانند $85^{\circ}C$ یا $-127^{\circ}C$ گزارش می‌شود.

همچنین این سنسور برای عملکرد صحیح نیازمند مقداردهی اولیه در بخش setup برنامه است. اجرای تابع initial begin منجر به عدم شناسایی دستگاه روی باس و تولید مقدار پیش‌فرض خطا می‌شود. این نکته نشان می‌دهد که تعامل صحیح میان لایه سخت‌افزار و نرم‌افزار برای دستیابی به داده معتبر ضروری است.



شکل 3. DS18B20 (Body Temperature Sensor)

سنجش شرایط محیطی با DHT11

سنسور DHT11 برای اندازه‌گیری دمای محیط و رطوبت نسبی استفاده می‌شود. این سنسور دارای دقت قابل قبول برای کاربردهای عمومی بوده و از طریق یک پروتکل تک‌سیمه با میکروکنترلر ارتباط برقرار می‌کند. داده‌های محیطی علاوه بر تکمیل تصویر فیزیولوژیک فرد، می‌توانند در تحلیل تغییرات علائم حیاتی نیز مؤثر باشند؛ زیرا شرایط محیطی بر عملکرد سیستم تنفسی و قلبی تأثیرگذار است.



شکل 4. DHT11 (Room Humidity and Temperature)

3-2- معماری نرم افزارهای بلادرنگ و برنامه نویسی غیرمسدودکننده

در سیستم‌های نهفته چندوظیفه‌ای، ساختار برنامه‌نویسی نقش تعیین‌کننده‌ای در پایداری سیستم دارد. استفاده از توابع مسدودکننده مانند delay باعث توقف اجرای سایر بخش‌های برنامه می‌شود و در سامانه‌هایی که نیازمند نمونه‌برداری مداوم هستند، این امر می‌تواند منجر به از دست رفتن داده یا اختلال در الگوریتم‌های داخلی شود.

در میکروکنترلری مانند ESP32 که به‌طور هم‌زمان وظیفه مدیریت Wi-Fi، اجرای Web Server و خواندن سنسورها را بر عهده دارد، استفاده از معماری غیرمسدودکننده (Non-Blocking Architecture) یک ضرورت محسوب می‌شود. در این رویکرد، به‌جای توقف برنامه برای مدت مشخص، از تابع millis() برای مدیریت زمان استفاده می‌شود و هر وظیفه در بازه‌های زمانی تعیین‌شده اجرا می‌گردد.

این ساختار امکان اجرای هم‌زمان چند سنسور با نرخ‌های نمونه‌برداری متفاوت را فراهم می‌کند. برای مثال، سنسور MAX30100 نیازمند اجرای پیوسته تابع update در حلقه اصلی است، در حالی که DS18B20 و DHT11 دارای زمان پاسخ طولانی‌تر هستند و می‌توانند در بازه‌های زمانی بزرگ‌تر خوانده شوند.

معماری غیرمسدودکننده همچنین بستر مناسبی برای پیاده‌سازی الگوریتم‌های پردازش سیگنال و سیستم هشدار فراهم می‌کند. در چنین ساختاری، می‌توان بدون اختلال در خواندن سنسورها، مقادیر را فیلتر کرده، بازه‌های ناهنجاری را بررسی نمود و وضعیت کلی سیستم را به‌روزرسانی کرد. در پروژه حاضر، بازطراحی حلقه اصلی برنامه و حذف کامل delay‌های طولانی، نقطه عطفی در رسیدن به عملکرد پایدار سیستم بود. این تجربه نشان داد که در سیستم‌های بلادرنگ، کیفیت معماری نرم‌افزار به اندازه انتخاب سنسور اهمیت دارد.

فصل سوم: طراحی و پیاده سازی سیستم

1-3-مقدمه

پس از بررسی مبانی نظری و پیش زمینه علمی در فصل قبل، در این فصل فرآیند طراحی و پیاده سازی عملی سامانه پایش سلامت ارائه می شود. هدف این فصل، تشریح دقیق تصمیم های مهندسی اتخاذ شده در مسیر توسعه سیستم و تحلیل چالش های عملی است که در فرآیند پیاده سازی بروز کرده اند.

در طراحی یک سامانه IoT پزشکی، صرف اتصال چند سنسور به یک میکروکنترلر کافی نیست. کیفیت عملکرد نهایی وابسته به هماهنگی میان لایه های مختلف سیستم، مدیریت منابع پردازشی، کنترل نویز، زمان بندی دقیق نرم افزار و ارائه خروجی قابل اعتماد است. از این رو، در این فصل ابتدا معماری کلی سیستم معرفی شده و سپس طراحی سخت افزار و نرم افزار به صورت مرحله ای تشریح می شود.

2-3-معماری کلی سیستم و مدل لایه ای

سامانه طراحی شده بر اساس یک مدل لایه ای چهار سطحی توسعه یافته است. این مدل با هدف افزایش ماژولار بودن، کاهش وابستگی بین بخش ها و تسهیل عیب یابی انتخاب شده است.

لایه اندازه گیری (Sensing Layer)

در این لایه، سنسورهای زیستی و محیطی قرار دارند که وظیفه جمع آوری داده های خام را بر عهده دارند. در این پروژه:

- MAX30100 برای اندازه گیری BPM و SpO_2

- DS18B20 برای دمای بدن

- DHT11 برای دما و رطوبت محیط

هر سنسور دارای نرخ نمونه برداری، پروتکل ارتباطی و محدودیت های عملیاتی متفاوتی است. این ناهمگونی یکی از چالش های اصلی طراحی سیستم بوده است.

لایه پردازش (Processing Layer)

این لایه توسط میکروکنترلر ESP32 پیاده‌سازی شده است. وظایف این لایه شامل: خواندن داده‌ها از سنسورها، مدیریت زمان‌بندی، اعمال فیلترهای نرم‌افزاری، تشخیص شرایط ناهنجار، تولید داده‌های نهایی برای نمایش

انتخاب ESP32 به دلیل وجود Wi-Fi داخلی، توان پردازشی مناسب و پشتیبانی از پروتکل‌های ارتباطی مختلف صورت گرفته است.

لایه ارتباطی (Communication Layer)

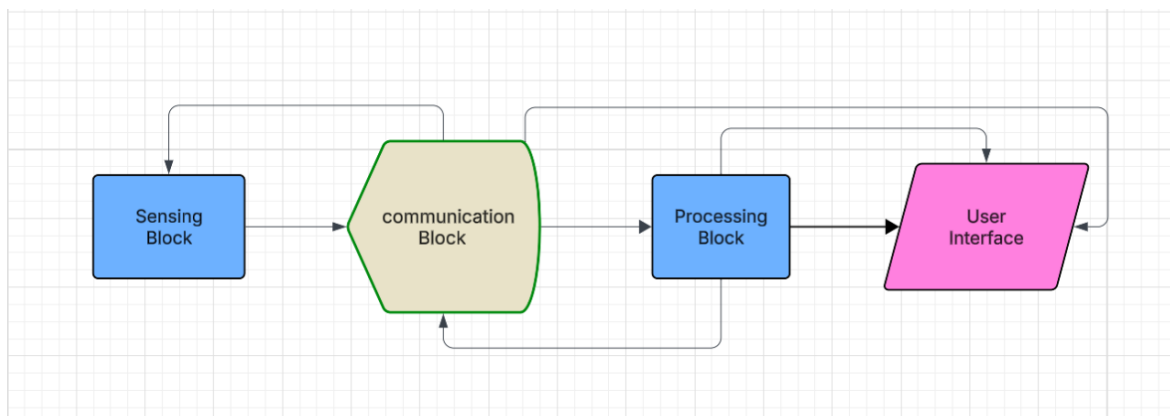
در این پروژه از Wi-Fi داخلی ESP32 برای راه‌اندازی یک Web Server داخلی استفاده شده است. در این معماری، دستگاه پس از اتصال به شبکه محلی یک IP دریافت می‌کند و کاربر می‌تواند با وارد کردن آن در مرورگر، داشبورد را مشاهده کند.

این طراحی باعث شده سیستم وابسته به Cloud نباشد و حتی در شرایط قطعی اینترنت سراسری نیز عملکرد کامل داشته باشد.

لایه نمایش (User Interface Layer)

داشبورد تحت وب طراحی شده شامل نمایش بلادرنگ پنج پارامتر اصلی و یک سیستم هشدار چندسطحی است. تولید صفحه HTML به صورت پویا توسط تابع SendHTML انجام می‌شود و داده‌ها هر یک ثانیه به‌روزرسانی می‌شوند.

معماری لایه‌ای انتخاب شده باعث شده تغییر در یک بخش (مثلاً جایگزینی سنسور) کمترین تأثیر را بر سایر بخش‌ها داشته باشد.



شکل 5. بلوک دیاگرام نحوه ارتباط بلوک‌های اصلی

در مرحله انتخاب واحد پردازشی، گزینه‌هایی مانند STM32، Raspberry Pi Pico و ESP32 مورد بررسی قرار گرفتند. معیارهای انتخاب شامل: توان پردازشی کافی، وجود Wi-Fi داخلی، مصرف توان مناسب، در دسترس بودن کتابخانه‌ها، هزینه پایین ESP32 با داشتن پردازنده دو هسته‌ای، ماژول Wi-Fi داخلی و پشتیبانی از I2C و One-Wire، گزینه‌ای مناسب برای این پروژه تشخیص داده شد.

طراحی اتصالات و پروتکل‌ها

اتصال MAX30100 از طریق پروتکل I2C انجام شد. در تست‌های اولیه، طول زیاد سیم‌ها و کیفیت تغذیه باعث بروز ناپایداری در ارتباط شد. با کوتاه کردن سیم‌ها و بهبود اتصالات، این مشکل برطرف گردید.

سنسور DS18B20 از پروتکل One-Wire استفاده می‌کند. بر اساس دیتاشیت، وجود مقاومت Pull-up به مقدار $4.7k\Omega$ بین خط داده و VCC ضروری است. در مراحل اولیه، عدم استفاده از این مقاومت باعث تولید مقادیر 85°C و -127°C شد. پس از افزودن مقاومت، پایداری داده‌ها به‌طور کامل بهبود یافت. سنسور DHT11 نیز از یک خط داده تک‌سیمه استفاده می‌کند و نیازمند رعایت زمان‌بندی دقیق در خواندن داده است.

ملاحظات عملی

در فرآیند پیاده‌سازی مشخص شد که عوامل زیر تأثیر قابل توجهی بر کیفیت داده‌ها دارند:

- کیفیت منبع تغذیه
- طول سیم‌های ارتباطی
- تماس صحیح انگشت با سنسور PPG
- نویز محیطی

این موارد نشان دادند که طراحی سخت‌افزار تنها محدود به شماتیک نیست، بلکه شرایط واقعی بهره‌برداری نیز باید در نظر گرفته شوند.

4-3- طراحی نرم افزار و معماری غیرمسدودکننده

برنامه شامل سه بخش اصلی setup، loop و توابع ماژولار برای هر سنسور است. در تابع setup عملیات مقداره‌ی اولیه سنسورها، اتصال Wi-Fi و راه اندازی Web Server انجام می‌شود.

در مراحل اولیه، از ساختارهای مبتنی بر delay استفاده شد. اما این روش باعث اختلال در عملکرد MAX30100 می‌شد، زیرا کتابخانه این سنسور نیازمند اجرای مداوم تابع update در حلقه اصلی است.

بازطراحی معماری

برای رفع این مشکل، معماری برنامه به صورت کامل به ساختار غیرمسدودکننده تبدیل شد. در این ساختار:

- هیچ delay طولانی وجود ندارد
 - از millis برای مدیریت زمان استفاده می‌شود
 - هر سنسور در بازه زمانی مناسب خود خوانده می‌شود
 - تابع update سنسور PPG به صورت پیوسته اجرا می‌شود
- این تغییر باعث حذف کامل مشکل صفر شدن BPM و SpO_2 در هنگام اجرای همزمان وب سرور شد.

مدیریت اختلاف نرخ نمونه برداری

یکی از چالش‌های اصلی، اختلاف زمان پاسخ سنسورها بود MAX30100 نیازمند نمونه برداری پیوسته است، در حالی که DS18B20 و DHT11 دارای تأخیر ذاتی در پاسخ هستند. با تفکیک زمان بندی هر سنسور، این تضاد حل شد.

این تجربه نشان داد که طراحی نرم افزار در سیستم‌های بلادرنگ به اندازه انتخاب سخت افزار اهمیت دارد.

5-3- طراحی الگوریتم‌های پردازش سیگنال و سیستم هشدار

یکی از مهم ترین بخش های طراحی سامانه، تبدیل داده های خام سنسورها به اطلاعات قابل اعتماد و قابل تفسیر است. به ویژه در مورد سیگنال های مبتنی بر PPG، داده های خروجی مستعد نویز حرکتی، تغییرات ناگهانی و خطاهای ناشی از تماس نامناسب هستند. بنابراین پردازش نرم افزاری نه به عنوان یک قابلیت اضافی، بلکه به عنوان بخشی ضروری از معماری سیستم در نظر گرفته شد.

در مرحله نخست، برای کاهش نوسانات ناگهانی در مقادیر BPM و SpO_2 ، مکانیزم حذف جهش (Spike Rejection) پیاده‌سازی شد. در این روش، اگر اختلاف مقدار فعلی با مقدار قبلی از یک آستانه مشخص فراتر می‌رفت و از نظر فیزیولوژیک قابل توجیه نبود، مقدار جدید به عنوان خطای احتمالی در نظر گرفته می‌شد و وارد فرآیند فیلتر نمی‌گردید. این کار از نمایش مقادیر غیر واقعی ناشی از حرکت دست جلوگیری کرد.

در گام بعدی، برای هموارسازی سیگنال از فیلتر میانگین نمایی (Exponential Moving Average – EMA) استفاده شد. این فیلتر با وزن‌دهی بیشتر به داده‌های اخیر و حفظ روند کلی تغییرات، نوسانات لحظه‌ای را کاهش داد بدون آنکه پاسخ دینامیکی سیستم بیش از حد کند شود. انتخاب ضریب هموارسازی بر اساس آزمون عملی انجام شد تا تعادلی میان پایداری و سرعت پاسخ برقرار گردد.

با وجود این بهبودها، در شرایطی که کاربر انگشت خود را از روی سنسور MAX30100 برمی‌داشت، مقادیر قبلی برای چند ثانیه ثابت باقی می‌ماندند و سیستم همچنان آن‌ها را معتبر تلقی می‌کرد. این رفتار از دیدگاه پایش سلامت قابل قبول نبود. برای رفع این مسئله، الگوریتم تشخیص عدم تماس بر اساس فاصله زمانی بین ضربان‌های تشخیص داده شده طراحی شد. اگر در بازه زمانی مشخصی هیچ Beat جدیدی ثبت نمی‌شد، سیستم نتیجه می‌گرفت که تماس قطع شده است و مقادیر BPM و SpO_2 را به صفر یا وضعیت Sensor Fault منتقل می‌کرد. این اصلاح باعث شد رفتار سیستم در شرایط واقعی بسیار طبیعی‌تر و قابل اعتمادتر شود.

در کنار پردازش سیگنال، طراحی سیستم هشدار چندسطحی نیز انجام شد. برای هر پارامتر، بازه‌های فیزیولوژیک تعریف گردید و وضعیت آن در یکی از چهار سطح OK، Warning، Critical یا Sensor Fault طبقه‌بندی شد. به منظور جلوگیری از هشدارهای کاذب، تغییر وضعیت تنها در صورتی اعمال می‌شد که مقدار خارج از محدوده برای چند نمونه متوالی تکرار گردد. این مکانیزم نوعی تأخیر منطقی ایجاد می‌کند که مانع از واکنش به نوسانات گذرا می‌شود.

وضعیت نهایی هر پارامتر در یک سطح کلی سیستم جمع‌بندی می‌شود و داشبورد بر اساس شدیدترین وضعیت فعال، رنگ و پیام مناسب را نمایش می‌دهد. این طراحی باعث شد سامانه از یک نمایشگر ساده داده به یک مانیتور فعال و واکنش‌گرا تبدیل شود.

6-3- ابزارهای توسعه و محیط پیاده‌سازی

توسعه نرم‌افزار با استفاده از محیط Arduino IDE انجام شد که به دلیل وجود کتابخانه‌های آماده برای ESP32 و سنسورهای مورد استفاده، فرآیند پیاده‌سازی را تسهیل می‌کند. کتابخانه DallasTemperature برای مدیریت الگوریتم داخلی PPG، کتابخانه MAX30100_PulseOximeter برای DS18B20 و کتابخانه DHT برای سنسور محیطی مورد استفاده قرار گرفتند.

در فرآیند توسعه، استفاده از Serial Monitor نقش مهمی در عیب‌یابی ایفا کرد. بسیاری از خطاهای اولیه، از جمله مقدار 127- در سنسور دمای بدن یا صفر شدن BPM، از طریق بررسی خروجی سریال و تحلیل مرحله‌ای کد شناسایی شدند. این رویکرد سیستماتیک در دیباگ باعث شد علت ریشه‌ای مشکلات مشخص شود، نه اینکه صرفاً با تغییرات تصادفی در کد برطرف گردند.

طراحی رابط کاربری تحت وب به صورت تولید رشته HTML در سمت سرور انجام شد. در مراحل اولیه، استفاده نادرست از کوتیشن‌های تو در تو در بخش JavaScript منجر به خطاهای کامپایل می‌شد. با بازطراحی ساختار رشته‌ها و استفاده صحیح از کاراکترهای escape، این مشکل برطرف شد. طراحی CSS ساده ولی هدفمند باعث شد داشبورد علاوه بر عملکرد مناسب، از نظر بصری نیز ساختار منسجم و حرفه‌ای داشته باشد.

7-3- معیارهای ارزیابی طراحی

برای ارزیابی عملکرد سامانه، چند معیار کلیدی در نظر گرفته شد. نخستین معیار، پایداری در خواندن داده‌ها طی زمان طولانی بود. سیستم در بازه‌های چندساعته مورد آزمون قرار گرفت تا مشخص شود آیا اجرای هم‌زمان وب‌سرور و سنسورها باعث افت عملکرد می‌شود یا خیر.

معیار دوم، صحت نسبی مقادیر اندازه‌گیری شده بود. مقادیر ضربان قلب و سطح اکسیژن خون در شرایط استراحت و پس از فعالیت سبک ثبت و با بازه‌های فیزیولوژیک مرجع مقایسه شدند. همچنین دمای محیط با یک دماسنج مرجع مقایسه گردید تا انحراف احتمالی مشخص شود.

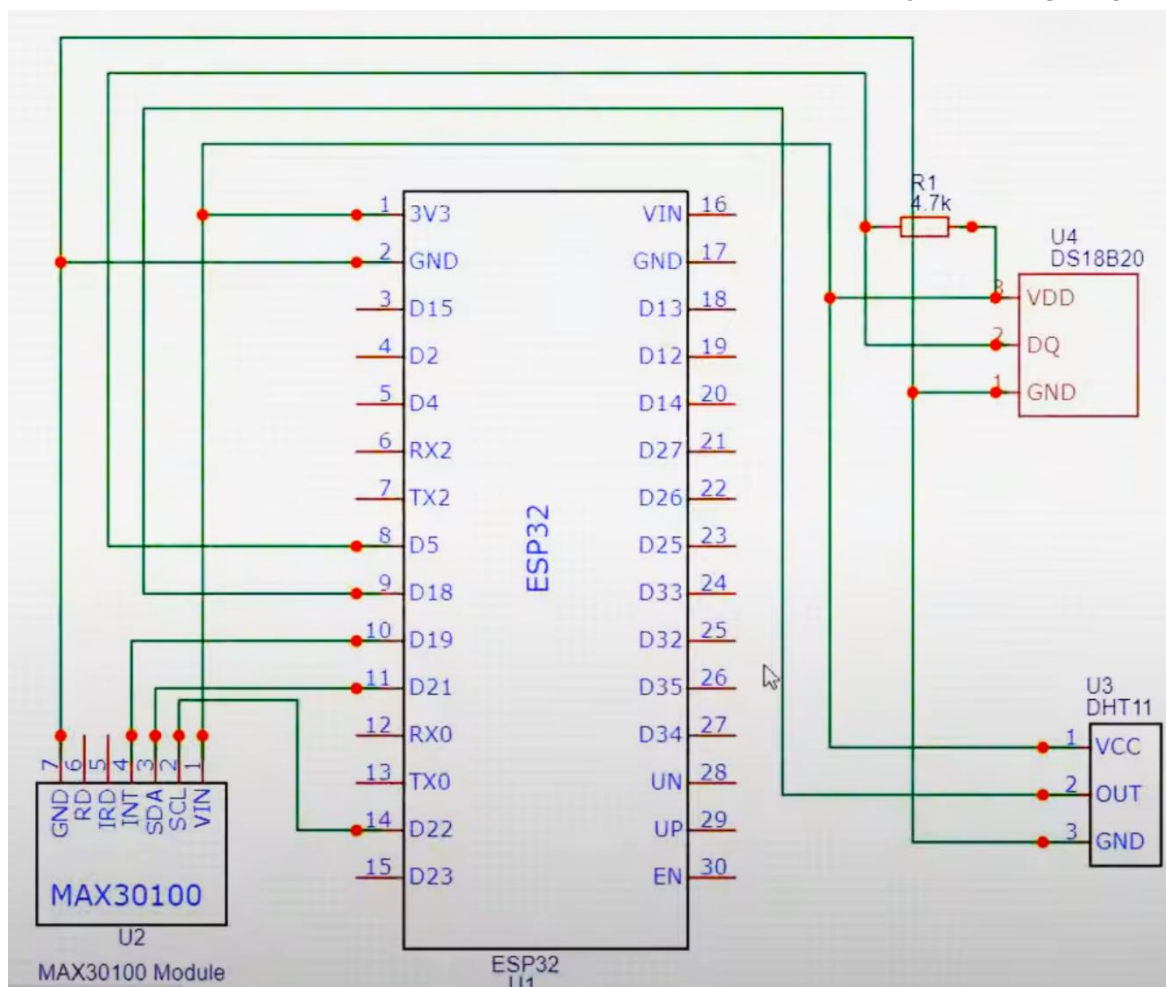
معیار سوم، رفتار سیستم در شرایط خطا بود. قطع سنسور دمای بدن، برداشتن انگشت از سنسور PPG و ایجاد نویز حرکتی از جمله سناریوهایی بودند که برای بررسی واکنش سیستم اجرا شدند. هدف این بود که سامانه نه تنها در شرایط ایده‌آل، بلکه در شرایط غیرعادی نیز رفتار قابل پیش‌بینی داشته باشد.

معیار چهارم، پاسخ‌گویی رابط کاربری و به‌روزرسانی بلادرنگ داشبورد بود. بررسی شد که آیا تأخیر شبکه یا پردازش داخلی باعث کندی نمایش می‌شود یا خیر.

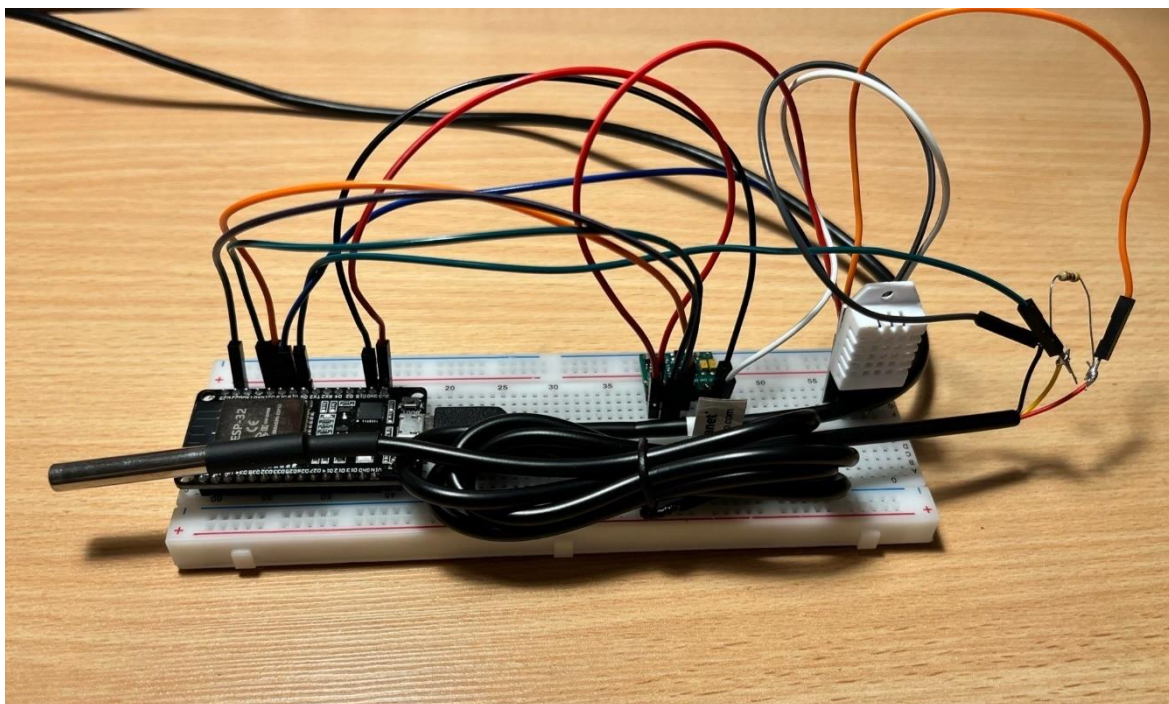
8-3- تحلیل طراحی و جمع‌بندی

تحلیل نهایی طراحی نشان می‌دهد که موفقیت سامانه بیش از آنکه وابسته به انتخاب سنسورها باشد، نتیجه طراحی صحیح معماری نرم‌افزار و مدیریت منابع پردازشی است. حذف ساختارهای مسدودکننده و استفاده از زمان‌بندی مبتنی بر millis نقطه عطف اصلی در رسیدن به عملکرد پایدار بود. همچنین تجربه خطاهای سخت‌افزاری نظیر نبود مقاومت Pull-up یا مقداردهی اولیه نادرست نشان داد که در سیستم‌های نهفته، کوچک‌ترین جزئیات می‌توانند اثر قابل توجهی بر خروجی داشته باشند.

در مجموع، طراحی انجام‌شده نشان می‌دهد که ترکیب معماری ماژولار، برنامه‌نویسی غیرمسدودکننده، پردازش اولیه سیگنال و طراحی مناسب رابط کاربری می‌تواند منجر به ایجاد یک سامانه پایش سلامت کم‌هزینه ولی پایدار شود. تجربه عملی این پروژه بیانگر آن است که فرآیند آزمون، خطا و بهینه‌سازی تدریجی بخش جدایی‌ناپذیر طراحی مهندسی است و رسیدن به نسخه نهایی نتیجه اصلاحات مرحله‌ای و تحلیل دقیق مشکلات بوده است.



شکل 6. شماتیک سیستم و اتصالات برقرار شده بین ماژول‌ها



شکل 7. شکل نهایی سیستم پیاده سازی شده بر روی بردبرد

فصل چهارم: پیاده سازی سامانه

1-4-مقدمه

پس از طراحی معماری سخت افزار و نرم افزار در فصل سوم، مرحله پیاده سازی عملی سامانه آغاز شد. هدف این مرحله، تحقق فیزیکی طراحی انجام شده و ارزیابی رفتار واقعی سیستم در شرایط مختلف بود. در این فرآیند، مشخص شد که بسیاری از چالش های سیستم های نهفته تنها در مرحله اجرا و آزمون عملی آشکار می شوند و مستقیماً قابل پیش بینی از روی طراحی نظری نیستند. پیاده سازی این سامانه نه تنها شامل اتصال ماژول ها و نوشتن کد، بلکه شامل فرآیند آزمون، خطا، تحلیل علت ریشه ای مشکلات و بازطراحی بخش هایی از نرم افزار بود. به همین دلیل، این فصل علاوه بر شرح اجرای سیستم، نشان دهنده مسیر تکامل آن از یک نمونه اولیه ناپایدار به نسخه ای پایدار و قابل اتکا است.

2-4- پیاده سازی سخت افزار و یکپارچه سازی ماژول ها

در مرحله نخست، کلیه ماژول ها شامل ESP32، MAX30100، DS18B20 و DHT11 مطابق شماتیک طراحی شده به یکدیگر متصل شدند. تغذیه سیستم از طریق پورت USB تأمین گردید و ولتاژ 3.3 ولت مورد نیاز سنسورها از خروجی رگولاتور داخلی برد ESP32 تأمین شد.

اتصال MAX30100

این سنسور از طریق پروتکل I2C به ESP32 متصل شد. پایه های SDA و SCL به پایه های تعریف شده در کد متصل شدند و زمین و تغذیه به صورت مشترک با سایر اجزا برقرار گردید. در تست های اولیه، مشاهده شد که افزایش طول سیم ها و عبور آن ها از نزدیکی خطوط تغذیه باعث ایجاد نویز و ناپایداری در داده های BPM می شود. با کوتاه سازی مسیرها و بهبود آرایش سیم کشی، این مشکل به طور کامل برطرف شد.

همچنین کیفیت تماس انگشت با سنسور تأثیر مستقیم بر شکل سیگنال PPG داشت. فشار بیش از حد یا تماس ناکافی منجر به نوسانات غیرعادی در مقادیر می شد. این مشاهده اهمیت تعامل کاربر با سامانه را نیز نشان داد.

اتصال DS18B20

این سنسور از طریق پروتکل One-Wire به ESP32 متصل شد. مقاومت Pull-up با مقدار $4.7\text{k}\Omega$ به یک خط داده و 3.3 ولت قرار گرفت. در مراحل اولیه، عدم استفاده از این مقاومت باعث بازگشت مقادیر خطا مانند 127°C می‌شد. پس از افزودن مقاومت و اطمینان از لحیم صحیح اتصالات، خواندن داده‌ها پایدار شد.

همچنین اجرای صحیح تابع مقداردهی اولیه در بخش setup نقش اساسی در شناسایی سنسور روی باس داشت. بدون این مرحله، میکروکنترلر هیچ دستگاهی را شناسایی نمی‌کرد.

اتصال DHT11

سنسور DHT11 به یکی از پایه‌های دیجیتال ESP32 متصل شد. این سنسور به دلیل ماهیت تک‌سیمه خود نیازمند رعایت دقیق زمان‌بندی در سطح نرم‌افزار است. در برخی تست‌های اولیه، خواندن سریع‌تر از حد مجاز باعث دریافت داده‌های نامعتبر می‌شد. با تنظیم بازه زمانی مناسب بین خواندن‌ها، این مشکل رفع گردید.

ملاحظات عملی

در حین پیاده‌سازی مشخص شد که موارد زیر تأثیر قابل توجهی بر عملکرد دارند:

- اشتراک صحیح زمین بین تمامی ماژول‌ها
 - جلوگیری از عبور خطوط I2C از کنار خطوط تغذیه
 - تثبیت مکان فیزیکی سنسور PPG هنگام تست
 - جلوگیری از تغییر ناگهانی بار منبع تغذیه
- این موارد گرچه در طراحی شماتیک ساده به نظر می‌رسند، اما در عمل تعیین‌کننده کیفیت عملکرد نهایی بودند.

3-4- پیاده‌سازی نرم‌افزار و ساختار اجرایی نهایی

ساختار اصلی برنامه

نسخه نهایی برنامه شامل سه بخش اصلی setup، loop و مجموعه‌ای از توابع ماژولار برای هر سنسور است. در تابع setup عملیات زیر انجام می‌شود:

- مقداردهی اولیه ارتباط سریال برای دیباگ
- اجرای sensors.begin برای DS18B20
- راهاندازی سنسور MAX30100 و تعیین نرخ نمونه‌برداری
- اتصال به شبکه Wi-Fi
- راهاندازی Web Server داخلی

ترتیب اجرای این مراحل اهمیت زیادی داشت. در تست‌های اولیه مشخص شد که در برخی شرایط، اتصال Wi-Fi پیش از مقداردهی کامل سنسورها باعث تأخیر در راهاندازی I2C می‌شود. با تغییر ترتیب مراحل، این مشکل برطرف شد.

معماری غیرمسدودکننده

در نسخه اولیه برنامه، استفاده از delay باعث توقف اجرای حلقه اصلی و در نتیجه اختلال در عملکرد MAX30100 می‌شد. این موضوع منجر به نمایش مقادیر صفر برای BPM و SpO₂ در هنگام فعال بودن وب‌سرور می‌گردید.

در نسخه نهایی، تمامی delay های طولانی حذف شدند و مدیریت زمان با استفاده از millis انجام شد. برای هر سنسور یک بازه زمانی مستقل تعریف شد و اجرای آن‌ها به صورت شرطی و غیرمسدودکننده انجام می‌گیرد. تابع update مربوط به MAX30100 در هر تکرار حلقه loop فراخوانی می‌شود تا الگوریتم داخلی آن بدون وقفه اجرا گردد.

این بازطراحی یکی از مهم‌ترین نقاط عطف پروژه محسوب می‌شود و موجب پایداری کامل خواندن داده‌ها در حضور Wi-Fi و وب‌سرور شد.

4-4- پیاده‌سازی سیستم هشدار و منطق تصمیم‌گیری

پس از تثبیت خواندن داده‌ها، مرحله پیاده‌سازی منطق هشدار آغاز شد. برای هر پارامتر بازه‌های نرمال، هشدار و بحرانی تعریف گردید. این بازه‌ها بر اساس مقادیر فیزیولوژیک عمومی انتخاب شدند.

برای جلوگیری از هشدارهای کاذب، شرط تغییر وضعیت تنها در صورتی فعال می‌شود که مقدار خارج از محدوده برای چند نمونه متوالی تکرار گردد. به عنوان مثال، اگر BPM به طور لحظه‌ای از حد تعریف شده عبور کند ولی در نمونه بعدی به محدوده طبیعی بازگردد، وضعیت سیستم تغییر نمی‌کند.

علاوه بر این، یک سطح جداگانه برای Sensor Fault در نظر گرفته شد. در صورتی که داده‌ای نامعتبر باشد (مانند قطع تماس سنسور PPG یا عدم پاسخ DS18B20)، وضعیت مربوطه به خطا تغییر می‌کند و از نمایش داده قدیمی جلوگیری می‌شود.

وضعیت کلی سیستم بر اساس شدیدترین وضعیت فعال تعیین می‌شود و در بالای داشبورد نمایش داده می‌شود.

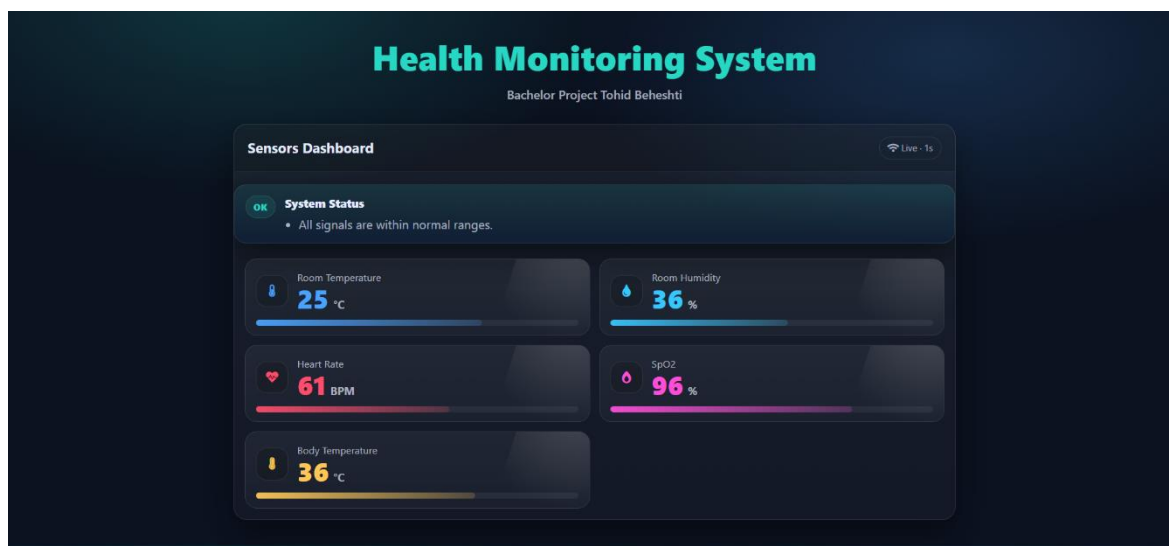
5-4- پیاده‌سازی رابط کاربری تحت وب

رابط کاربری به صورت یک صفحه HTML پویا تولید می‌شود که توسط تابع SendHTML ساخته شده و به درخواست کاربر ارسال می‌گردد. داده‌های سنسورها هر یک ثانیه به روزرسانی می‌شوند.

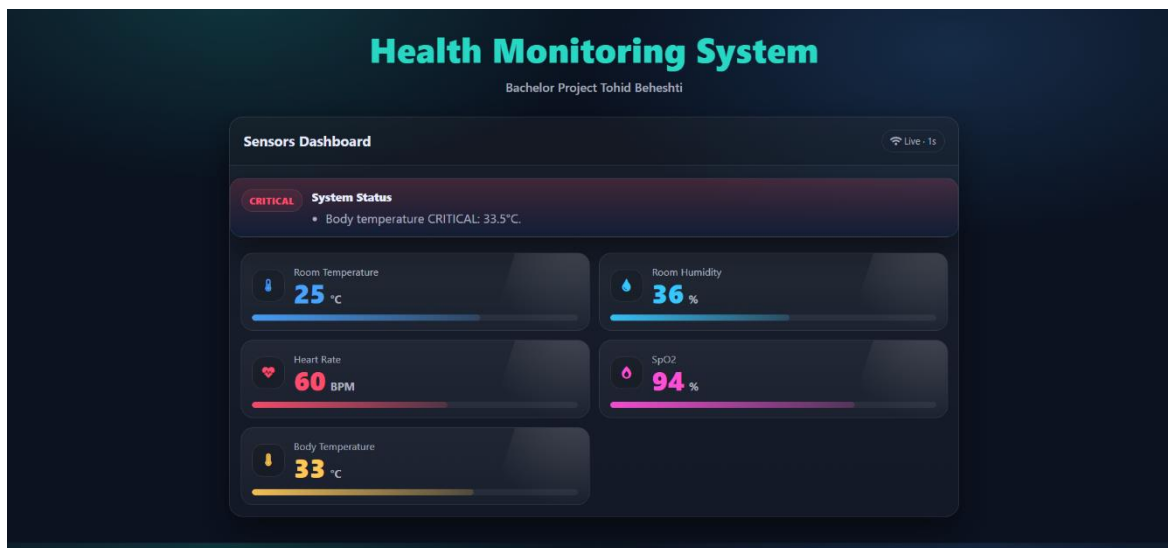
طراحی داشبورد بر اساس سادگی، خوانایی و نمایش واضح وضعیت‌ها انجام شده است. برای هر پارامتر یک کارت جداگانه با مقدار عددی و واحد مربوطه در نظر گرفته شده است. رنگ‌بندی کارت‌ها متناسب با وضعیت پارامتر تغییر می‌کند.

در بالای صفحه، یک نوار وضعیت کلی سیستم قرار دارد که می‌تواند در یکی از حالت‌های زیر باشد:

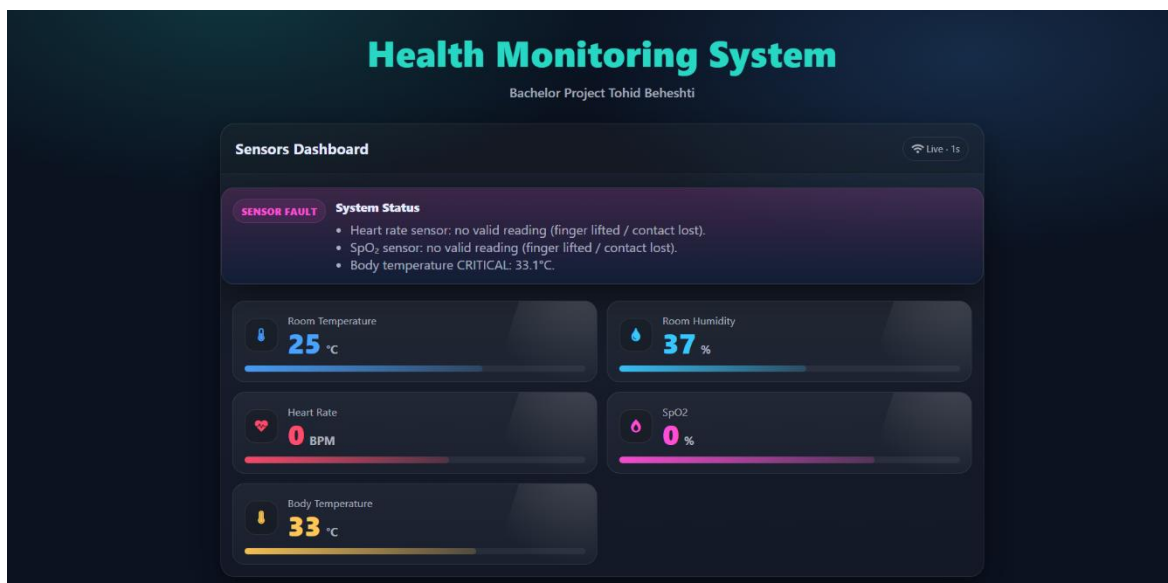
- OK (سبز)
- Warning (زرد)
- Critical (قرمز)
- Sensor Fault (بنفش)



شکل 8. نمای داشبورد در وضعیت عادی (OK)



شکل 9. نمایش وضعیت بحرانی (Critical Condition)



شکل 10. وضعیت خطای سنسور (Sensor Fault)

6-4- اجرای سناریوهای آزمون و بررسی عملکرد سیستم

پس از تکمیل پیاده‌سازی سخت‌افزار و نرم‌افزار، سامانه در مجموعه‌ای از سناریوهای عملی مورد آزمون قرار گرفت تا رفتار آن در شرایط واقعی ارزیابی شود. این آزمون‌ها با هدف بررسی پایداری، صحت نسبی داده‌ها، عملکرد سیستم هشدار و واکنش سامانه در شرایط غیرعادی طراحی شدند.

در نخستین سناریو، سیستم در حالت استراحت کامل کاربر مورد بررسی قرار گرفت. در این وضعیت، ضربان قلب در بازه 70 تا 90 ضربه در دقیقه ثبت شد و سطح اکسیژن خون بین 95 تا 99 درصد باقی ماند. دمای بدن در محدوده 36.2 تا 36.8 درجه سانتی‌گراد اندازه‌گیری شد که با مقادیر فیزیولوژیک طبیعی همخوانی دارد. در این حالت، داشبورد وضعیت کلی سیستم را در سطح OK نمایش داد و رنگ‌بندی عناصر گرافیکی نیز در حالت پایدار باقی ماند.

در سناریوی دوم، پس از انجام فعالیت بدنی سبک مانند راه رفتن سریع به مدت یک دقیقه، افزایش تدریجی مقدار BPM مشاهده شد که در برخی نمونه‌ها به محدوده 100 تا 110 رسید. پس از توقف فعالیت، کاهش تدریجی ضربان قلب و بازگشت آن به محدوده طبیعی ثبت گردید. این رفتار دینامیکی نشان می‌دهد که سیستم قادر به ثبت تغییرات واقعی فیزیولوژیک با تأخیر قابل قبول است. در صورتی که مقدار ضربان از آستانه تعریف‌شده عبور می‌کرد و این وضعیت برای چند نمونه متوالی پایدار می‌ماند، سطح هشدار به Warning یا Critical تغییر می‌یافت و نوار وضعیت در بالای داشبورد نیز رنگ مربوطه را نمایش می‌داد.

در سناریوی سوم، تماس انگشت با سنسور MAX30100 قطع شد. در نسخه‌های اولیه سیستم، مقادیر قبلی برای مدتی بدون تغییر باقی می‌ماندند که می‌توانست گمراه‌کننده باشد. در نسخه نهایی، پس از گذشت زمان تعیین‌شده بدون ثبت ضربان جدید، سیستم وضعیت Sensor Fault را فعال کرد و مقادیر BPM و SpO₂ به حالت نامعتبر منتقل شدند. این رفتار نشان می‌دهد که الگوریتم تشخیص عدم تماس به‌درستی عمل می‌کند و از نمایش داده‌های قدیمی جلوگیری می‌شود.

سناریوی دیگر شامل قطع موقت سنسور دمای بدن یا ایجاد خطا در اتصال آن بود. در این حالت، مقدار 127°C- درجه که نشانه عدم شناسایی سنسور است دریافت شد و سیستم بلافاصله وضعیت خطا را برای آن پارامتر فعال کرد. این واکنش نشان می‌دهد که مدیریت خطا به‌صورت سیستماتیک در نرم‌افزار لحاظ شده و سیستم در مواجهه با نقص سخت‌افزاری رفتار پیش‌بینی‌پذیر دارد.

7-4- ارزیابی پایداری در کار مداوم

یکی از آزمون‌های مهم، بررسی عملکرد سامانه در بازه زمانی طولانی بود. سیستم برای چندین ساعت به صورت پیوسته روشن نگه داشته شد و داده‌ها به طور مداوم در داشبورد تحت وب نمایش داده شدند. در نسخه‌های اولیه، پس از مدتی اجرای همزمان Wi-Fi و سنسورها باعث بروز اختلال در خواندن داده‌های MAX30100 می‌شد. با بازطراحی معماری نرم‌افزار و حذف کامل توابع مسدودکننده، این مشکل به طور کامل برطرف گردید.

در آزمون نهایی، هیچ‌گونه قطع ارتباط Wi-Fi یا افت محسوس در نرخ به روزرسانی مشاهده نشد. اجرای پیوسته تابع update در حلقه اصلی و تفکیک زمان‌بندی سنسورها موجب شد سیستم حتی در شرایط بار پردازشی هم‌زمان، پایداری خود را حفظ کند. این نتیجه نشان می‌دهد که معماری غیرمسدودکننده انتخاب‌شده در فصل طراحی، در مرحله اجرا نیز عملکرد مؤثری داشته است.

همچنین در دوره‌ای که دسترسی به اینترنت سراسری محدود بود، سامانه صرفاً در شبکه محلی مورد استفاده قرار گرفت. در این حالت نیز بدون نیاز به سرویس‌های ابری، داده‌ها در داشبورد نمایش داده شدند. این تجربه عملی، مزیت معماری مبتنی بر وب‌سرور داخلی را تأیید کرد و نشان داد که سیستم در شرایط واقعی بهره‌برداری انعطاف‌پذیر است.

8-4- تحلیل رفتار سیستم و محدودیت‌های نسخه فعلی

تحلیل عملکرد سامانه نشان می‌دهد که ترکیب پردازش سیگنال اولیه، مدیریت زمان‌بندی و سیستم هشدار چندسطحی موجب افزایش قابل توجه قابلیت اطمینان سیستم شده است. حذف جهش‌های ناگهانی و استفاده از میانگین‌نمایی باعث کاهش نویز در مقادیر BPM و SpO_2 گردید، در حالی که الگوریتم تشخیص قطع تماس از نمایش داده‌های غیرواقعی جلوگیری کرد.

با این حال، محدودیت‌هایی نیز در نسخه فعلی وجود دارد. نخست آنکه سیگنال PPG همچنان به حرکت دست حساس است و در صورت لرزش شدید، ممکن است برای مدت کوتاهی نوسانات غیرعادی مشاهده شود. دوم آنکه داده‌ها در حال حاضر ذخیره‌سازی بلندمدت ندارند و تحلیل تنها به سطح لحظه‌ای محدود است. همچنین سیستم هشدار تنها در سطح نمایش بصری عمل می‌کند و خروجی فیزیکی مانند Buzzer در این نسخه فعال نشده است.

این محدودیت‌ها نشان می‌دهند که اگرچه سامانه به یک سطح عملکرد پایدار رسیده است، اما همچنان ظرفیت توسعه در نسخه‌های آینده وجود دارد.

9-4- جمع‌بندی فصل پیاده‌سازی

فصل حاضر نشان داد که پیاده‌سازی عملی یک سامانه پایش سلامت مبتنی بر IOT نیازمند توجه هم‌زمان به جزئیات سخت‌افزاری و معماری نرم‌افزاری است. بسیاری از مشکلات مشاهده‌شده در مراحل اولیه، نه ناشی از انتخاب نامناسب سنسورها، بلکه ناشی از ساختار زمان‌بندی و مدیریت نادرست منابع بودند. بازطراحی حلقه اصلی برنامه و حذف توابع مسدودکننده نقش کلیدی در رسیدن به نسخه پایدار ایفا کرد.

نسخه نهایی سامانه توانست علائم حیاتی را با پایداری مناسب ثبت و در شرایط ناهنجار واکنش منطقی و قابل پیش‌بینی نشان دهد. پیاده‌سازی سیستم هشدار چندسطحی و طراحی داشبورد واکنش‌گرا، سامانه را از یک نمونه ساده نمایش داده به یک ابزار پایش فعال ارتقا داد.

در مجموع، مرحله پیاده‌سازی نشان داد که فرآیند مهندسی در سیستم‌های نهفته یک مسیر تدریجی مبتنی بر آزمون، تحلیل و اصلاح است و رسیدن به عملکرد قابل اتکا نتیجه مجموعه‌ای از بهبودهای مرحله‌ای بوده است.

فصل پنجم: نتیجه‌گیری و پیشنهادهای توسعه آینده

1-5- نتیجه‌گیری

در این پژوهش، طراحی و پیاده‌سازی یک سامانه پوشیدنی مبتنی بر اینترنت اشیاء برای پایش بلادرنگ علائم حیاتی انجام شد. هدف اصلی پروژه، توسعه سیستمی کم‌هزینه، مستقل از زیرساخت‌های ابری و در عین حال پایدار و قابل اعتماد بود که بتواند چندین پارامتر زیستی و محیطی را به‌صورت هم‌زمان اندازه‌گیری و تحلیل کند.

نتایج حاصل از پیاده‌سازی نشان داد که ترکیب میکروکنترلر ESP32 با سنسورهای MAX30100، DS18B20 و DHT11 می‌تواند بستری مناسب برای ساخت یک سامانه پایش سلامت فراهم آورد. سیستم طراحی شده قادر است ضربان قلب، سطح اکسیژن خون، دمای بدن و شرایط محیطی را با پایداری قابل قبول ثبت کرده و در یک داشبورد تحت وب به‌صورت بلادرنگ نمایش دهد.

یکی از مهم‌ترین دستاوردهای این پروژه، بازطراحی معماری نرم‌افزار به‌صورت غیرمسدودکننده بود. تجربه عملی نشان داد که استفاده از ساختارهای مبتنی بر delay در سیستم‌های بلادرنگ منجر به اختلال در عملکرد سنسورهای حساس به زمان، به‌ویژه MAX30100، می‌شود. با حذف کامل توابع مسدودکننده و استفاده از زمان‌بندی مبتنی بر millis، پایداری سیستم به‌طور چشمگیری افزایش یافت. این تغییر نشان داد که در سیستم‌های نهفته، کیفیت معماری نرم‌افزار به اندازه انتخاب سخت‌افزار اهمیت دارد.

افزون بر این، پیاده‌سازی الگوریتم‌های پردازش سیگنال شامل حذف جهش‌های ناگهانی، فیلتر میانگین نمایی و مکانیزم تشخیص قطع تماس انگشت، باعث شد خروجی سیستم رفتار طبیعی‌تر و قابل اعتمادتری داشته باشد. افزودن سیستم هشدار چندسطحی نیز سامانه را از یک نمایشگر ساده داده به یک ابزار پایش فعال ارتقا داد که قادر است خروج از محدوده‌های تعریف‌شده را تشخیص داده و بازخورد مناسب ارائه دهد.

در بعد سخت‌افزاری، تجربه خطاهایی نظیر عدم استفاده از مقاومت Pull-up برای DS18B20 یا مقداردهی اولیه نادرست سنسورها نشان داد که توجه به جزئیات کوچک در طراحی، نقش تعیین‌کننده‌ای در عملکرد کلی سیستم دارد. این تجربه عملی، اهمیت فرآیند آزمون و خطا و تحلیل علت ریشه‌ای مشکلات را در پروژه‌های مهندسی برجسته می‌کند.

به‌طور کلی، سامانه توسعه‌یافته توانست اهداف اصلی تحقیق را محقق سازد و به‌عنوان یک نمونه عملی از یک سیستم پایش سلامت مبتنی بر IoT عمل کند. هرچند این سامانه در سطح یک محصول تجاری نیست، اما به‌عنوان یک پلتفرم آموزشی و تحقیقاتی، قابلیت توسعه و ارتقای قابل توجهی دارد.

2-5- پیشنهادهای توسعه آینده

با وجود موفقیت نسخه فعلی سامانه، زمینه‌های متعددی برای توسعه و بهبود آن وجود دارد. نخستین مسیر توسعه می‌تواند افزودن قابلیت ذخیره‌سازی داده‌ها در حافظه محلی یا ارسال آن‌ها به یک پایگاه داده ابری باشد. ذخیره‌سازی بلندمدت داده‌ها امکان تحلیل روند تغییرات و استخراج الگوهای سلامت را فراهم می‌کند که برای کاربردهای پزشکی ارزشمند است.

مسیر دیگر، توسعه الگوریتم‌های تحلیل پیشرفته‌تر است. استفاده از روش‌های پردازش سیگنال پیشرفته یا حتی الگوریتم‌های یادگیری ماشین می‌تواند دقت تشخیص ناهنجاری‌ها را افزایش دهد و سیستم را از یک سامانه آستانه‌محور به یک سامانه تحلیل‌گر هوشمند ارتقا دهد.

بهبود طراحی سخت‌افزار و تبدیل نمونه بردوردی به یک برد مدار چاپی (PCB) اختصاصی نیز می‌تواند موجب کاهش نویز، افزایش پایداری و کوچک‌تر شدن ابعاد سامانه شود. این اقدام گامی مهم در جهت تبدیل پروژه به یک محصول قابل استفاده روزمره خواهد بود.

افزودن ماژول‌های تکمیلی مانند سنسور ECG، فشار خون یا شتاب‌سنج نیز می‌تواند دامنه پایش را گسترش دهد و تصویر جامع‌تری از وضعیت فیزیولوژیک کاربر ارائه کند. همچنین افزودن یک سیستم هشدار فیزیکی مانند Buzzer یا اعلان موبایلی می‌تواند واکنش سیستم را از سطح نمایش بصری فراتر ببرد.

در نهایت، طراحی یک اپلیکیشن موبایل اختصاصی برای ارتباط مستقیم با ESP32 و نمایش داده‌ها در قالب گراف‌های زمانی می‌تواند تجربه کاربری را بهبود دهد و سامانه را به یک پلتفرم کامل‌تر پایش سلامت تبدیل کند.

3-5- جمع‌بندی نهایی

پروژه حاضر نشان داد که طراحی یک سامانه پایش سلامت مبتنی بر اینترنت اشیاء، فرآیندی چندبعدی است که نیازمند درک هم‌زمان از سخت‌افزار، نرم‌افزار، پردازش سیگنال و تعامل کاربر با سیستم است. موفقیت نهایی این پروژه نه حاصل یک تصمیم واحد، بلکه نتیجه مجموعه‌ای از اصلاحات تدریجی و تحلیل دقیق مشکلات بوده است.

تجربه حاصل از این تحقیق بیانگر آن است که حتی در پروژه‌های کوچک مقیاس، رعایت اصول معماری نرم‌افزار، مدیریت زمان‌بندی و توجه به جزئیات سخت‌افزاری می‌تواند تفاوت میان یک سیستم ناپایدار و یک سامانه قابل اعتماد را رقم بزند. این پروژه علاوه بر دستیابی به اهداف فنی خود، زمینه‌ای برای توسعه‌های آینده و پژوهش‌های پیشرفته‌تر در حوزه پایش سلامت هوشمند فراهم کرده است.

1. Datasheet of MAX30102 Pulse Oximeter and Heart-Rate Sensor, Maxim Integrated, 2023.
2. ESP32 Technical Reference Manual, Espressif Systems, 2023.
3. Analog Devices (Maxim Integrated) – *MAX30100 Pulse Oximeter and Heart-Rate Sensor IC Datasheet*
4. DS18B20 – Digital Temperature Sensor Datasheet
5. DHT11 – Temperature & Humidity Sensor _ Technical specification Manual
6. M. M. Rahman, M. A. Qureshi and M. Z. U. Khan, “A Review on Smart Health Monitoring Systems using IoT,” *Health Information Science and Systems*, vol. 9, no. 1, pp. 1–15, 2021.
7. L. Atzori, A. Iera and G. Morabito, “The Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347-2376, 2015.

پیوست ها

پیوست 1- جدول آستانه‌های سیستم هشدار

پارامتر	وضعیت خطای سنسور (Sensor Fault)	محدوده بحرانی (Critical)	محدوده هشدار (Warning)	محدوده نرمال (OK)
ضربان قلب (BPM)	مقدار صفر یا عدم تشخیص ضربان	کمتر از 50 یا بیشتر از 120	50 – 59 یا 101 – 120	60 – 100 bpm
سطح اکسیژن خون (SpO ₂)	مقدار صفر یا قطع تماس انگشت	کمتر از 90%	90% – 94%	≥ 95%
دمای بدن (°C)	مقدار -127 یا 85 درجه (خطای سنسور)	کمتر از 35.0 یا بیشتر از 38.0	37.5 – 37.9	36.0 – 37.4
دمای محیط (°C)	مقدار غیرواقعی یا عدم پاسخ	کمتر از 18 یا بیشتر از 30	18 – 19 یا 27 – 30	20 – 26
رطوبت محیط (%)	مقدار خارج از بازه 0-100%	کمتر از 20 یا بیشتر از 70	20 – 29 یا 61 – 70	30 – 60

پیوست 2- جدول اتصالات ماژول ها به ESP32 (pin Assignment)

ماژول	پایه ماژول	پایه متصل به ESP32	توضیحات
MAX30100	SDA	GPIO21	ارتباط I2C
MAX30100	SCL	GPIO22	ارتباط I2C
MAX30100	INT	GPIO19	وقفه (Interrupt)
MAX30100	VCC	3.3V	تغذیه
MAX30100	GND	GND	زمین مشترک
DS18B20	VCC	3.3V	تغذیه
DS18B20	GND	GND	زمین مشترک
DS18B20	Signal (DQ)	GPIO5	دارای مقاومت Pull-up 4.7kΩ
DHT22	VCC	3.3V	تغذیه
DHT22	GND	GND	زمین مشترک
DHT22	Signal	GPIO18	ارتباط دیجیتال تک سیمه

پیوست 3- دسترسی به کد پروژه

جهت مستندسازی کامل و فراهم‌سازی امکان بازتولید سامانه، نسخه نهایی کد پروژه در GitHub منتشر شده است. آخرین نسخه پایدار سیستم از طریق لینک زیر قابل دسترسی می‌باشد:

<https://github.com/TohidBeh/Health-Monitoring-System.git>