بسم الله الرحمن الرحیم

گزارش و مستندات پروژه
کارخانه هوشمند
(Smart factor)

شبیه سازی سیستم هوشمند صنعتی، مانیتورینگ بلادرنگ و سیستم‌های نهفته

تیم پروژه:

منتظر نداف مقدم، علیرضا نوید، امیرحسین اسدی،

اشکان پور رستمی، آرین شفیع راد

استاد راهنما: جناب دکتر زرگری

دانشگاه تبریز - دانشکده مهندسی برق و کامپیوتر

نیم سال دوم ۱۴۰۴

چکیده

موضوع پروژه درس سیستم نهفته ما کارخانه هوشمند می باشد طوری که در این پروژه تلاش شده تا یک کارخانه هوشمند و صنعت نسل 4.0 که بر مبنای اتوماسیون و مانیتورینگ صنعتی است تا حدودی طراحی و پیاده‌سازی شود. این سیستم با یکپارچه‌سازی لایه‌های سخت‌افزاری و فریمورهای نهفته (Embedded Firmware) به زبان C و شبیه‌سازی در محیط Proteus 8، همراه با لایه مرکزی (Central System)، پایگاه داده و اپلیکیشن، امکان نظارت بلادرنگ بر فرایندهای تولید را فراهم می‌سازد.

زیرسیستم‌های اصلی این پروژه شامل مانیتورینگ ماشین‌آلات، سیستم ایمنی و اعلام حریق، کنترل کیفیت خودکار و مدیریت نیروی کار می‌باشد.

بخش ۱: مقدمه و کلیات پروژه

۱-۱- مقدمه و بیان مسئله

در صنعت، میدانیم گذار از سیستم‌های اتوماسیون سنتی به سمت کارخانه‌های هوشمند بسیار حائز اهمیت میباشد. نبود مانیتورینگ بلادرنگ در خطوط تولید سنتی موجب افزایش زمان توقف ناخواسته، خطرات ایمنی حریق و کاهش دقت در کنترل کیفیت محصولات می‌گردد. از این رو موضوع پروژه **کارخانه هوشمند** انتخاب شده که هم اهمیت این موضوع را نشان بدهد و هم بتوان تمرینی دانشگاهی در این خصوص باشد.

۱-۲- اهداف

هدف اصلی پروژه ارائه یک نمونه کاربردی و پیاده‌سازی‌شده از پلتفرم صنعتی با قابلیت‌های زیر است:

- مانیتورینگ ماشین‌آلات: نظارت بر شرایط عملیاتی دستگاه‌ها (دما، لرزش، وضعیت کارکرد).
- سیستم اعلام حریق: کشف سریع حریق و اعمال وقفه خودکار برای حفظ ایمنی.
- سیستم پایش وضعیت سالن تولید: مدیریت خودکار انرژی گرمایش سرمایش.
- کنترل کیفیت: ارزیابی سلامت و استانداردهای قطعات تولیدی در خط.
- مدیریت نیروی کار: مدیریت حضور و غیاب، مدیریت تسک‌ها و تردد پرسنل.

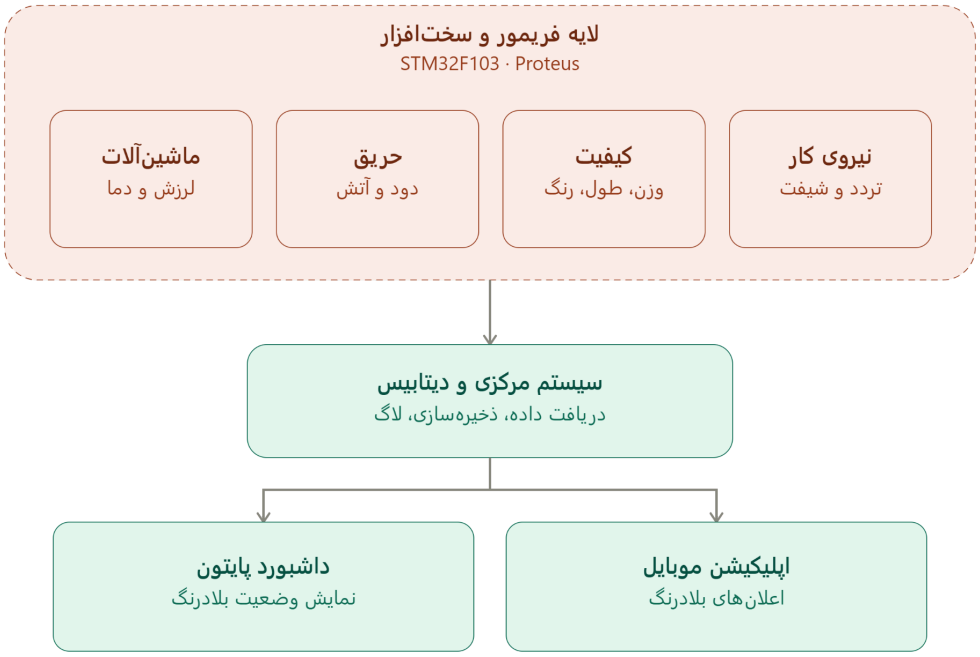
ساختار ماژولار: یک زیر سیستم مستقل در پوشه subsystems در نظر گرفته شده که شامل زیر مستندات، فریمور و شبیه سازی سخت‌افزار اختصاصی است.

بخش ۲: معماری کلی و تکنولوژی‌ها

۲-۱- معماری کلان سیستم

معماری این پروژه بر اساس الگوی سه لایه‌ای توزیع‌شده طراحی شده است:

- لایه فریمور و سخت‌افزار (Firmware/Hardware): مشتمل بر میکروکنترلرها، سنسورها و عملگرها که با زبان C توسعه داده شده و در محیط Proteus شبیه‌سازی شده‌اند.
- لایه سیستم مرکزی (Central System & DB): مسئول دریافت داده‌ها، مدیریت پیام‌ها، ذخیره‌سازی در دیتابیس و پردازش رویدادها.
- لایه کاربری (Python App): شامل اپلیکیشن و داشبورد مدیریتی جهت نمایش زنده وضعیت کارخانه به اپراتورها.



۲-۲- جدول تکنولوژی‌ها و ابزارها

بخش	تکنولوژی‌ها / زبان‌ها	توضیحات و ابزارها
فریمور نهفته (Firmware)	C ++, Assembly, C	برنامه‌نویسی MCUها، پردازش بلادرنگ سنسورها
شبیه‌سازی و سخت‌افزار	Proteus 8	طراحی مدارهای الکترونیکی و تست شبیه‌سازی
پایگاه داده و سرور	C / Central System DB	ذخیره لاگ‌ها، داده‌ها و مدیریت ارتباطات
رابط کاربری	Python mobile app	نمایش وضعیت خط تولید و اعلانات بلادرنگ

بخش ۳: پیاده‌سازی زیرسیستم‌ها

۳-۱- مانیتورینگ ماشین‌آلات

این زیرسیستم وضعیت سلامتی تجهیزات خط تولید را به صورت پیوسته پایش می‌کند. فریمور این بخش داده‌های آنالوگ/دیجیتال مربوط به دما و لرزش را دریافت کرده و در صورت تجاوز از آستانه مجاز، سیگنال هشدار صادر می‌کند. شبیه‌سازی کامل این بخش در پوشه firmware/WithProteus8 قرار دارد.

مسیر فریمور: subsystems/machine-monitoring/firmware

عملکرد:

i. نمونه‌برداری از لرزش‌سنج از طریق کانال صفر مبدل آنالوگ به دیجیتال (ADC1_IN0) انجام شده و مقدار خام ۱۲ بیتی خوانده می‌شود.

ii. جهت جلوگیری از افت سرعت به دلیل فقدان واحد محاسبات ممیز شناور در میکروکنترلر، محاسبات تبدیل به میلی‌متر بر ثانیه کاملاً بر پایه اعداد صحیح صورت می‌گیرد.

iii. مقدار لرزش مقیاس‌شده از طریق فرمول $vib_scaled = \frac{adc \times 140}{4095}$ محاسبه می‌شود که در آن بخش صحیح و اعشاری استخراج می‌گردد.

iv. ارزیابی سطح خطر بر اساس مقادیر خام ADC صورت می‌پذیرد؛ مقادیر کمتر از 819 معادل وضعیت NORMAL، مقادیر بین 819 تا 2073 معادل WARNING و مقادیر 2074 به بالا معادل وضعیت DANGER سنجیده می‌شوند.

v. نوار شدت لرزش مستقل از وضعیت خطر و بر اساس آستانه‌های vib_scaled کنترل می‌شود؛ به طوری که چهار ال‌ای‌دی روی پین‌های PB10 تا PB13 با عبور از مقادیر 0، 36، 70 و 105 به ترتیب روشن می‌شوند.

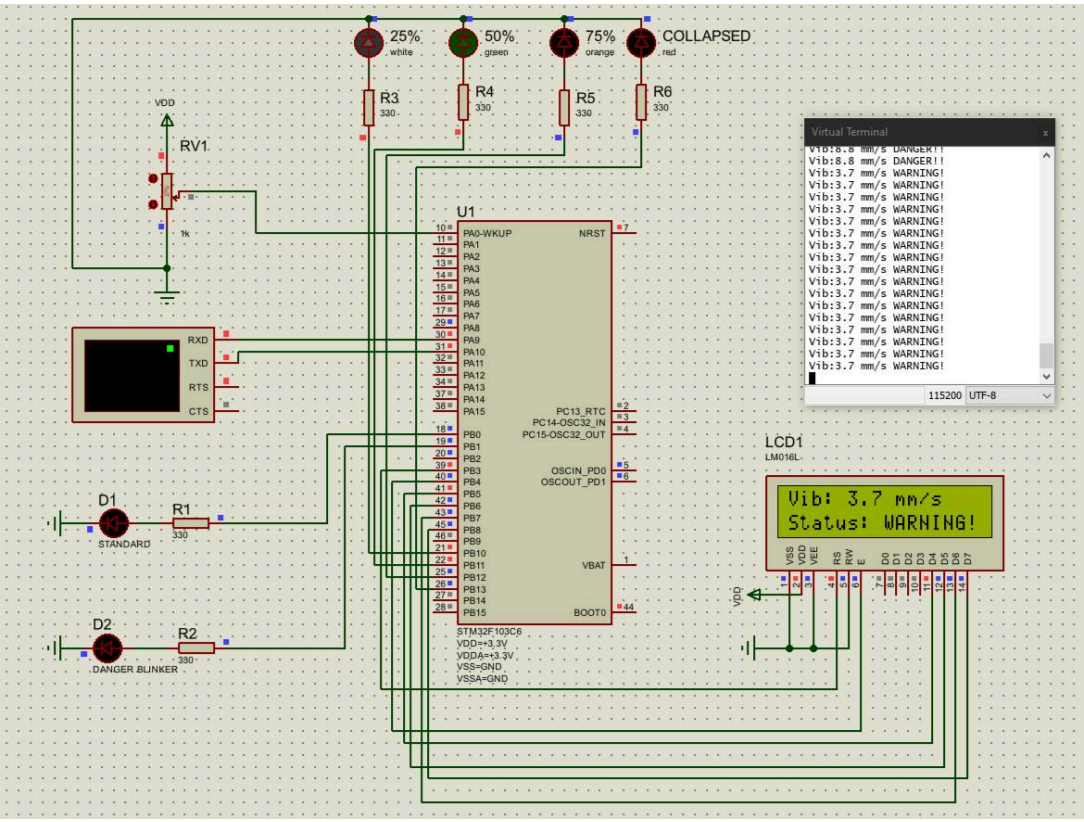
vi. سیستم هشدار بصری اصلی شامل دو ال‌ای‌دی وضعیت است؛ در وضعیت نرمال تنها ال‌ای‌دی سبز (PB0) روشن باقی می‌ماند.

vii. در حالت هشدار ال‌ای‌دی قرمز (PB1) با تاخیر غیر مسدود کننده 500 میلی‌ثانیه‌ای چشمک می‌زند و در حالت خطر (DANGER) سرعت چشمک‌زدن آن به تناوب سریع 100 میلی‌ثانیه افزایش می‌یابد.

viii. اطلاعات پردازش‌شده در هر چرخه فرمت‌بندی شده و همزمان از طریق پروتکل USART1 به ترمینال ارسال می‌شود.

ix. درایور سفارشی LCD در حالت 4 بیتی، داده‌ها را در قالب دو رشته مجزا شامل مقدار عددی لرزش و وضعیت متنی دستگاه روی نمایشگر چاپ می‌کند.

تصویر شماتیک این بخش را مشاهده می کنید.



۲-۳- زیرسیستم پایش محیط و اعلام حریق

بخش ایمنی وظیفه شناسایی دود - آتش و تنظیم دمای محیط بر حسب سنسورهای دما را دارد. منطق فریمور بر پایه سیستم‌های وقفه (Interrupt-driven) طراحی شده تا به محض کشف خطر، بدون تاخیر زمانی فرمان قطعی اضطراری و آلارم مرکز صادر شود.

مسیر فریمور: subsystems/fire-system/firmware

عملکرد:

i. بخش ایمنی حریق با پایش مداوم دو ورودی دیجیتال PIN_FIRE_1 و PIN_FIRE_2 روی پورت GPIOA عمل می‌کند.

ii. به محض تحریک هر یک از حسگرها، سیستم وارد حالت قفل اضطراری (is_system_locked = 1) شده و زمان وقوع حادثه به کمک تابع غیر مسدود کننده HAL_GetTick به روزرسانی می‌شود.

iii. در حالت قفل ایمنی (که تا ۵ ثانیه بعد از رفع خطر غیر فعال میشود)، برای جلوگیری از مخاطرات محیطی، عملگرهای کنترل اقلیم بلافاصله غیرفعال و خاموش می‌شوند.

iv. عملکرد اطفاء حریق کاملاً منطقه‌ای است؛ به طوری که تحریک حسگر اول تنها شیر برقی آب‌پاش اول (PIN_SPRINKLER_1) و تحریک حسگر دوم تنها آب‌پاش دوم (PIN_SPRINKLER_2) را فعال می‌سازد.

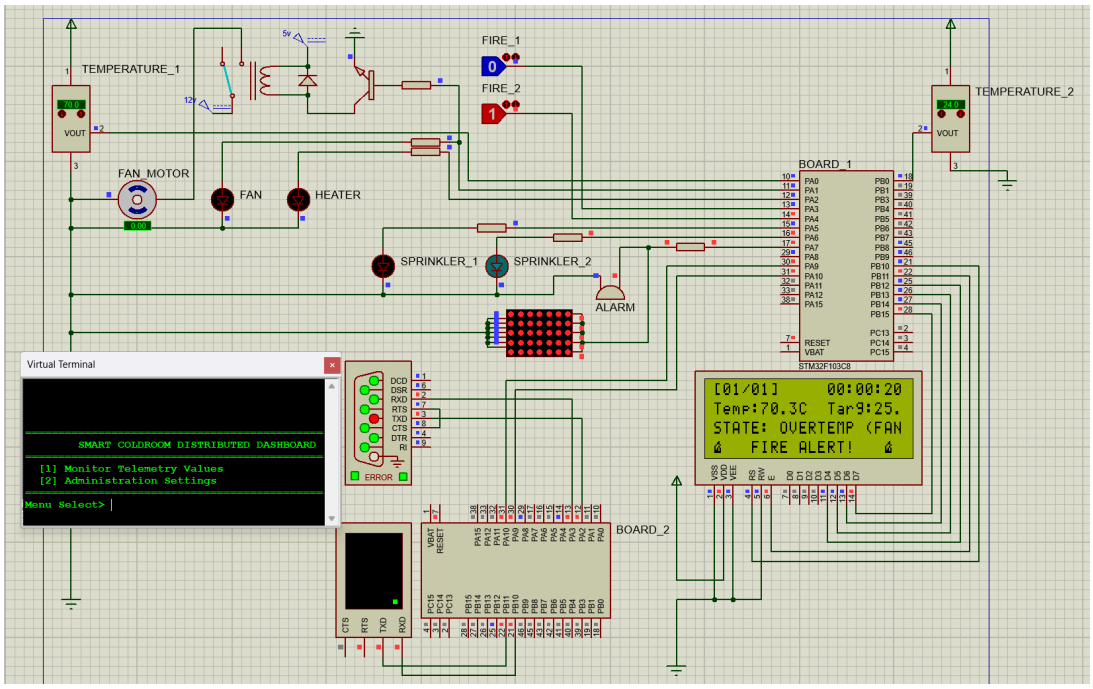
v. هشدار صوتی با دستور HAL_GPIO_TogglePin روی پایه آژیر پیاده‌سازی شده تا صدای آژیر در نرم‌افزار شبیه‌ساز پخش شود.

vi. پس از برطرف شدن حریق و صفر شدن ورودی‌ها، سیستم جهت اطمینان اپراتور وارد دوره تاخیر ایمنی شده و تا اتمام زمان ALARM_LOCK_DURATION_MS در حالت قفل باقی می‌ماند.

vii. پس از سپری شدن این زمان، قفل ایمنی سیستم و فلگ صداخفه (Mute) به صورت خودکار بازنشانی می‌گردند.

viii. همچنین کاربر می‌تواند از طریق دستورات UART، با دستور mute_alarm آژیر صوتی را قطع کند یا با دستور reset_sys تمامی خروجی‌های اطفاء حریق و متغیرهای مربوطه را به حالت اولیه بازگرداند.

تصویر شماتیک این بخش را مشاهده می کنید.



۳-۳- زیرسیستم کنترل کیفیت

در این ماژول، قطعات عبوری بر روی تسمه‌نقاله ارزیابی می‌شوند. کد فریمور با برقراری ارتباط با سنسورهای تشخیص ابعاد/سلامت، قطعات معیوب را شناسایی کرده و جک‌های خروجی را فعال می‌کند.

مسیر فریمور: subsystems/quality-control/firmware

عملکرد:

i. تشخیص ورود محصول و راه‌اندازی اولیه

- حالت انتظار (Idle): تا زمانی که کلید ON_OFF سیستم را فعال نکند، خروجی‌های LCD و سیستم در حالت غیرفعال/خاموش (LCD_STATE_OFF) باقی می‌مانند.
- سنسور حرکت: با ورود محصول به خط تولید، سنسور حرکت یک وقفه خارجی (EXTI) ایجاد کرده و فلگ motion_flag را فعال می‌کند.

ii. خواندن داده‌های سنسورها و آستانه‌گذاری

سیستم سه پارامتر اصلی را ارزیابی می‌کند:

- وزن: از طریق واحد ADC خوانده می‌شود و آستانه مجاز آن بین ۱۰۰ تا ۳۰۰۰ تعیین شده است
- طول: با ارسال دستور روی USART1 به سنسور اولتراسونیک محاسبه می‌شود؛ محدوده مجاز آن ۵ تا ۱۲ سانتی‌متر است

- رنگ: از پایه دیجیتال خوانده شده و با وضعیت COLOR_OK_STATE سنجیده می‌شود.

iii. کنترل و تفکیک فیزیکی محصول (موتور و رله‌ها)

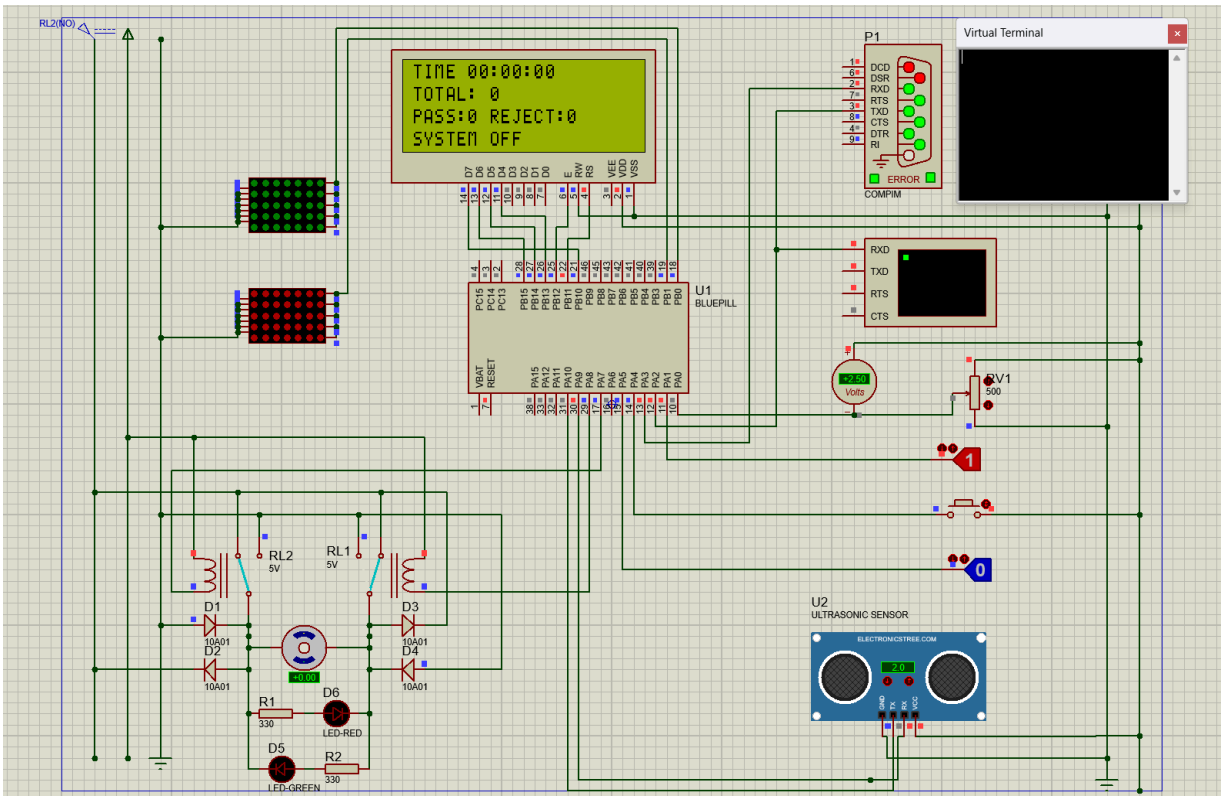
- محصول سالم (PASS): ال‌ای‌دی سبز روشن شده و رله‌های نوار نقاله غیرفعال می‌مانند تا محصول به مسیر عادی ادامه دهد.
- محصول خراب (REJECT): ال‌ای‌دی قرمز روشن می‌شود. یک مکانیزم زمان‌بندی‌شده دو مرحله‌ای با دو رله اجرا می‌شود:
  1. مرحله ۱: رله راست‌گرد (RELAY_FWD) به مدت ۵۰۰ میلی‌ثانیه فعال می‌شود تا نوار مجزا محصول خراب را خارج کند.
  2. مرحله ۲: رله چپ‌گرد (RELAY_REV) به مدت ۸۰۰ میلی‌ثانیه فعال می‌شود تا نوار به حالت اولیه بازگردد.

iv. ذخیره‌سازی تاریخچه

- سیستم یک آرایه از ساختارها به صورت بافر حلقوی (Circular Buffer) نگهداری می‌کند.
- زمان ثبت هر محصول بر اساس تابع HAL_GetTick به فرمت ساعت، دقیقه و ثانیه محاسبه و به همراه وضعیت (PASS/REJECT) در ۲۰ خانه آخر ذخیره می‌شود.

v. گزارش‌دهی و نمایشگر

- نمایشگر: زمان زنده، آمار کل، تعداد سالم، تعداد خراب و وضعیت آخرین محصول را به مدت ۲ ثانیه نمایش می‌دهد.
- ترمینال سریال: گزارش لحظه‌ای داده‌های سنسورها به همراه لیست کامل ۲۰ محصول آخر و زمان دقیق آن‌ها را ارسال می‌کند.



شکل شماتیک این بخش را مشاهده میکنید.

۴-۳- زیرسیستم مدیریت نیروی کار

این بخش مسئول ثبت زمان ورود/خروج، احراز هویت پرسنل بر روی خط تولید و تخصیص شیفت‌ها و تسک ها است.

مسیر فریمور: subsystems/workforce-management/firmware

عملکرد:

۱. ورود و احراز هویت کارگران و مدیران

- ورود کارگر: سیستم با دریافت اثر انگشت/کد از طریق UART1، درخواست REQ_NAME را به سرور (UART3) ارسال و پس از تایید، نام و پیام خوش‌آمدگویی را روی LCD نمایش می‌دهد.
- ورود مدیر: احراز هویت با نام‌کاربری و رمز عبور انجام شده و دسترسی به منوی مدیریتی (توقف خط، ثبت مرخصی، گزارش خطای فنی و دریافت اطلاعات) فعال می‌گردد.

۲. ثبت تردد، مرخصی و مکاتبات

- حضور و غیاب: ثبت ورود و خروج کارگر یا خروج اجباری توسط مدیر، مستقیما به پایگاه داده سرور منتقل می‌شود.
- مدیریت مرخصی: درخواست مرخصی بر اساس وضعیت پین ورودی پنل مدیریت (GPIOB_PIN_0) بررسی شده و پاسخ APPROVED یا REJECTED به سرور ارسال می‌شود.
- ارسال نامه‌ها: امکان ارسال گزارش خطا، نامه‌ها و درخواست‌های کاری از ترمینال به سرور فراهم شده است.

۳. مدیریت وظایف مازاد و اولویت‌بندی سنی

- تخصیص عادلانه کار: کارهای مازاد براساس گروه سنی کارگر (گروه‌های ۲، ۳ و ۴) در فازهای زمانی مشخص از سرور مرکزی تخصیص داده می‌شوند.
- ضرایب پاداش: پذیرش کار در زمان‌های متناسب با اولویت سنی، ضرایب پاداش متفاوت به همراه ثبت تاریخ در فایل‌های اختصاصی کارگر در پایگاه داده در پی دارد.

۴. پایش ایمنی و مدیریت پروتکل‌های اضطراری

- پایش ایمنی کارگر: در صورت عدم رعایت فاصله ایمنی یا پوشش ایمنی، هشدار صدایی/نوری فعال شده و تایمر تخصیص کار تا رفع خطا متوقف (FREEZE) می‌شود.
- پروتکل‌های بحران: قابلیت فعال‌سازی هشدارهای فوری مانند آتش‌سوزی، اورژانس پزشکی (حوادث صنعتی/شیمیایی) و تخلیه اضطراری کارخانه با تغییر وضعیت رله‌ها و نمایش پیام‌های بحران روی LCD تعبیه شده است.

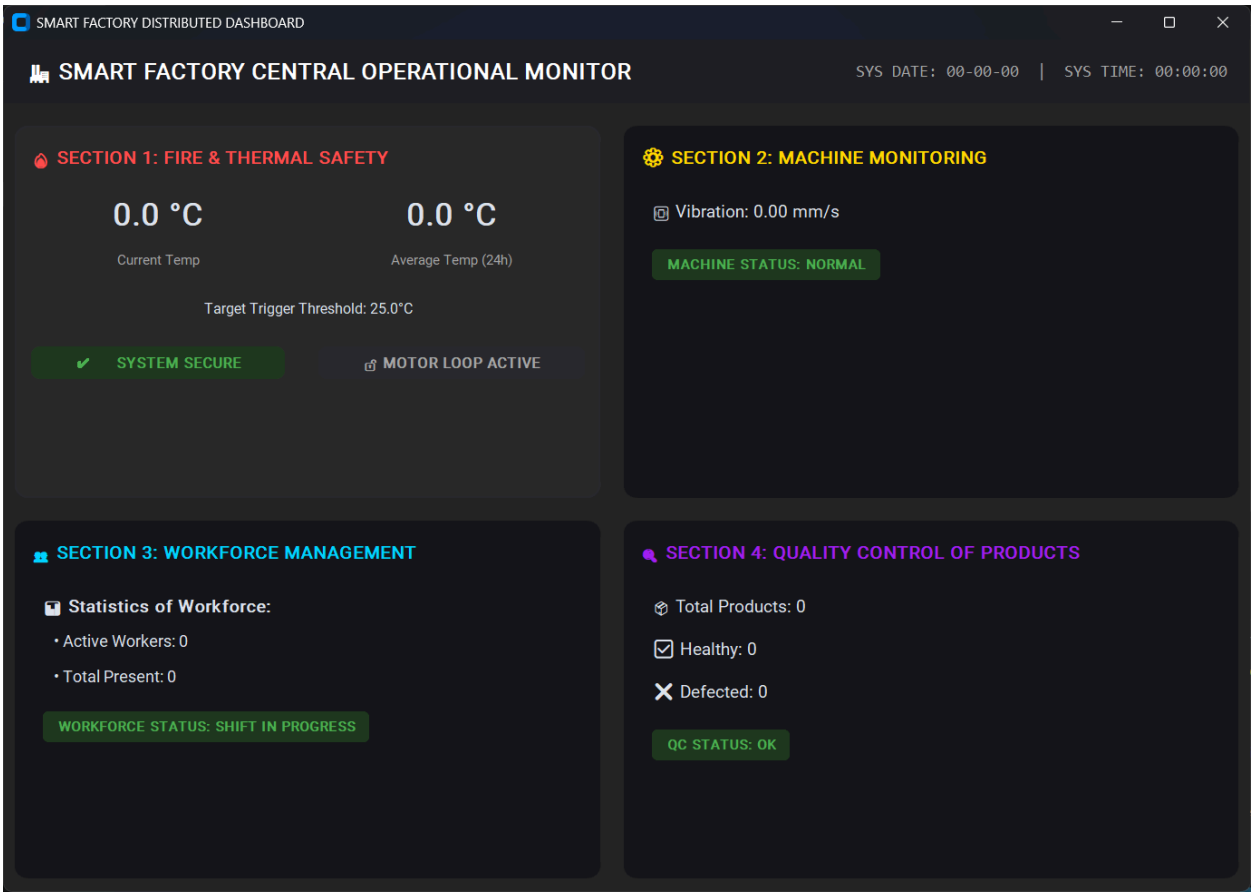
بخش ۴: سیستم مرکزی، پایگاه داده و اپلیکیشن پایتون

۴-۱- ساختار سیستم مرکزی و دیتابیس

پوشه Centeral System شامل کدهای مربوط به پردازش مرکزی و ارتباط با لایه‌های بالادستی است. پایگاه داده سیستم (database/) وظیفه ذخیره‌سازی لاگ‌های زمان‌بندی‌شده سنسورها، جدول کاربران و لیست رویدادهای ایمنی را برعهده دارد.

۴-۲- اپلیکیشن پایتون

اپلیکیشن پایتون برای دریافت اطلاعات بلادرنگ از زیر سیستم ها درنظر گرفته شده است.



۲-۴- راهنمای ساخت و توسعه و اجرای محلی

- دریافت کدها: کلون کردن مخزن پروژه از گیت‌هاب
- (git clone https://github.com/alznvd6/SMART_FACTORY.git).
- اجرای شبیه‌سازی: گشودن پروژه‌های Proteus 8 از پوشه زیرسیستم‌ها و اجرای شبیه‌سازی.
- کامپایل فریمور: کامپایل سورس‌کدهای و پروگرام کردن فایل Hex.
- صرفا برای اجرای محلی نیازی به کامپایل مجدد فایل های C نمیباشد و فایل hex اتومات در مسیر شبیه سازی برد پروتئوس قرار دارد.

بخش ۵: نتیجه‌گیری

در این پروژه تلاش شد یک نمونه کاربردی از مفهوم کارخانه هوشمند در مقیاس آزمایشگاهی طراحی و پیاده‌سازی شود؛ سیستمی که چهار زیرسیستم مانیتورینگ ماشین‌آلات، اعلام و اطفاء حریق، کنترل کیفیت و مدیریت نیروی کار را در قالب یک معماری سه‌لایه (فریمور/سخت‌افزار، سیستم مرکزی و پایگاه‌داده، و اپلیکیشن کاربری پایتون) به‌صورت یکپارچه کنار هم قرار می‌دهد.

محدودیت‌ها و کارهای آتی: پروژه در محیط شبیه‌سازی (Proteus) پیاده‌سازی و تست شده و پیاده‌سازی روی سخت‌افزار واقعی، ارتباط بی‌سیم (Wi-Fi/Bluetooth) به‌جای سریال، افزودن رمزنگاری در انتقال داده‌ها، و توسعه اپلیکیشن موبایل به‌عنوان گام‌های بعدی پیشنهاد می‌شود.
همچنین می‌توان با افزودن الگوریتم‌های یادگیری ماشین برای پیش‌بینی خرابی تجهیزات (Predictive Maintenance)، ارزش افزوده سیستم را در آینده افزایش داد.