به نام خدا

دانشکده مهندسی برق

دانشگاه صنعتی شریف

پروژه پایانی

سامانه هوشمند نگهبانی

سیستم‌های نهفته بلادرنگ

استاد:

جناب آقای دکتر ایمان غلام‌پور

نیمسال دوم ۱۴۰۵-۱۴۰۴

مقدمه و هدف پروژه

هدف این پروژه، طراحی و پیاده‌سازی یک سامانه نهفته کامل است که تصویر زنده دوربین را دریافت کرده، حضور انسان را تشخیص دهد و از طریق وب، ایمیل و پروتکل MQTT گزارش‌دهی کند. لازم است برد پس از روشن شدن بدون مداخله دستی شروع به کار کند و همه کارکرد های تعریف شده برای آن فعال باشد.

سخت‌افزار مورد نیاز: برد Orange Pi یا در صورت عدم دسترسی ماشین مجازی لینوکسی روی کامپیوتر شخصی

زبان برنامه‌نویسی الزامی: زبان **C** برای تمام بخش‌ها (به جز بخش پردازش تصویر و لایه Swagger که در ادامه توضیح داده می‌شود). **عدم استفاده از C در هر بخش، ۲۵٪ کسر نمره همان بخش را در پی دارد.**

معماری کلی سامانه

سامانه از سه قسمت اصلی تشکیل شده است:

۱. **برد OrangePi:** شامل وب‌سرور، خواندن سنسورهای سیستمی (دما و حافظه)، کلاینت MQTT، ارسال ایمیل و تمهیدات امنیتی به زبان C
۲. **ماژول پردازش تصویر:** تعداد افراد را تشخیص می دهد.
۳. **کامپیوتر شخصی:** راه‌اندازی MQTT Broker و مشاهده پیام‌های دریافتی در تاپیک مربوطه

الزامات امنیتی برای کل پروژه

لازم است موارد زیر در کل پروژه رعایت شوند و عدم رعایت آن ها کسر نمره دارد.

۱. ورود SSH فقط با کلید عمومی یا پسورد باشد. ورود root غیرفعال شود.
۲. ورود MQTT بدون احراز هویت غیرفعال شود و کاربر با پسورد اختصاصی ساخته شود.
۳. هیچ پسورد یا کلید API صریحا نباید به‌صورت hard-code در دستگاه باشد.

نکته درباره Swagger: از آن‌جا که Swagger UI به‌صورت بومی با FastAPI (پایتون) در دسترس است، مجازید یک **لایه نازک FastAPI** فقط به‌عنوان مستندساز و درگاه API داشته باشید، به شرطی که تمام منطق اصلی (خواندن دما، حافظه، شمارش افراد، اجرای دستورات) در برنامه C پیاده شده باشد.

بخش اول: وب‌سرور، سرویس خودکار و SSL (۲۵ نمره از ۱۰۰)

الف) راه‌اندازی وب‌سرور:

- یک وب‌سرور HTTP بنویسید که یک صفحه HTML شامل موارد زیر را میزبانی کند:
  - نام و شماره دانشجویی شما در عنوان صفحه
  - استریم زنده دوربین (اگر برد مجهز به دوربین نیست با استفاده از دوربین وبکم انجام دهید و یک مرحله استریم تصویر از لپ تاپ به برد اضافه شود.)
  - تعداد افراد تشخیص داده‌شده در لحظه
  - دمای پردازنده، مموری آزاد و درصد مصرف پردازنده به‌صورت زنده (به‌روزرسانی هر ۲ ثانیه)

ب) افزودن SSL با گواهی Self-Signed:

- گواهی self-signed با openssl بسازید. فیلد CN گواهی باید شماره دانشجویی شما باشد.
- صفحه فقط باید از طریق https در دسترس باشد و درخواست‌های http به https هدایت (redirect) شوند.

ج) سرویس systemd:

- فایل‌های service بنویسید به‌طوری‌که با روشن‌شدن برد، تمام برنامه‌ها (وب‌سرور، پردازش تصویر، کلاینت MQTT) بدون هیچ مداخله دستی اجرا شوند.
- سرویس‌ها باید در صورت crash به‌صورت خودکار restart شوند.
- ترتیب وابستگی سرویس‌ها (مثلاً پردازش تصویر پس از آماده‌شدن دوربین) باید با After و Requires مدیریت شود و در یک نمودار در گزارش این ترتیب را با ذکر علت توضیح دهید.

❖ **آزمایش‌های الزامی بخش اول: (در گزارش حتما به ترتیب شماره آزمایش و نتیجه را ذکر کنید.)**

شماره	آزمایش	خروجی مورد انتظار در گزارش
۱–۱	برد را reboot کنید و با systemd-analyze blame زمان بوت را اندازه بگیرید.	جدول زمان بوت کل و سهم سرویس‌های شما (تصویر از ترمینال)
۱–۲	پروسه وب‌سرور را با kill -9 بکشید.	تصویر لاگ journalctl که restart خودکار را نشان دهد.
۱–۳	یک بار برد را خاموش و سپس روشن کنید.	فیلم از بالا آمدن خودکار سامانه بدون لمس کیبورد
۱–۴	آدرس را به صورت http وارد کنید.	استفاده از قابلیت inspect در مرورگر و تصویر از redirect با کد ۳۰۱

۱–۵	گواهی self-signed برای صفحه html را باز کنید.	تصویر کلی گواهی و فیلد CN شامل شماره دانشجویی
۱–۶	صفحه HTML را باز کنید.	تصویری از این صفحه شامل شماره دانشجویی

بخش دوم: RESTful API و مانیتورینگ زنده (۲۵ نمره)

الف) طراحی API (منطق در C، مستندسازی با Swagger/FastAPI):

حداقل endpoint های زیر پیاده شوند:

عملکرد	METHOD	ENDPOINT
استریم زنده تصویر (MJPEG)	GET	**/API/V1/STREAM**
تعداد افراد فعلی + timestamp	GET	**/API/V1/PERSONS**
دمای CPU، حافظه آزاد، درصد بار CPU	GET	**/API/V1/TELEMETRY**
دستور اجرایی راه اندازی مجدد به صورت {"cmd":"reboot"}	POST	**/API/V1/COMMAND**
تاریخچه تشخیص‌ها (۵ رکورد آخر)	GET	**/API/V1/HISTORY**

- دما و حافظه باید از دایرکتوری مربوطه مستقیماً در کد C خوانده شود و استفاده از دستورات لینوکسی نمره ای ندارد.
- endpoint دستور اجرایی باید قابلیت دریافت دستور های مختلفی که در آینده ممکن است افزوده شود را داشته باشد.

ب) صفحه Swagger: تمام endpoint ها باید در Swagger UI قابل تست باشند. در گزارش، اسکرین‌شات اجرای زنده هر endpoint از داخل Swagger با پاسخ واقعی الزامی است.

❖ آزمایش‌های الزامی بخش دوم: (در گزارش حتما به ترتیب شماره آزمایش و نتیجه را ذکر کنید.)

شماره	آزمایش	خروجی مورد انتظار
۲–۱	دمای پردازنده را در سه حالت با نرخ نمونه‌برداری هر ۳۰ ثانیه در طول مدت ۵ دقیقه ثبت کنید: (الف) بیکار، (ب) فقط استریم، (ج) استریم + تشخیص فعال	نمودار دما—زمان با سه منحنی + جدول بیشینه دما و اسکرین شات از حالت انتهایی
۲–۲	مصرف حافظه برنامه C را در طول ۵ دقیقه استریم پیوسته با نرخ نمونه‌برداری هر ۵ ثانیه ثبت کنید.	نمودار بر حسب زمان + تحلیل کنید آیا نشتی حافظه دارید؟
۲–۳	با ابزار curl در حلقه، ۵۰ درخواست همزمان در طول مدت ۳۰ ثانیه به /api/v1/telemetry بزنید.	تغییرات دما، پردازنده و حافظه را گزارش کنید. تاخیر در پاسخ چقدر افزایش می یابد؟

۲–۴	کابل شبکه/وای‌فای را حین استریم قطع و پس از ۲ دقیقه وصل کنید.	توضیح رفتار سامانه، تصویر لاگ‌ها و نحوه بازیابی سامانه

بخش سوم: پردازش تصویر، ایمیل و MQTT (۲۵ نمره)

الف) تشخیص انسان (تنها بخش مجاز برای Python، ترجیحاً C/C++):

- با یک مدل سبک آماده تعداد افراد در تصویر شمارش شود.
- خروجی باید روی تصویر استریم با کادر و شمارنده نمایش داده شود.
- روی هر فریم خروجی باید شماره دانشجویی، تاریخ و ساعت زنده سیستم نوشته شود.
- نرخ فریم واقعی (FPS) اندازه‌گیری و روی تصویر نمایش داده شود.

ب) ارسال ایمیل:

- در صورت تشخیص یک یا چند انسان، ایمیلی شامل: تعداد افراد، timestamp، دمای لحظه‌ای پردازنده و تصویر پیوست‌شده لحظه تشخیص ارسال شود.
- ارسال ایمیل باید در کد C انجام شود.
- مکانیزم debounce پیاده کنید. حداکثر یک ایمیل در هر ۳۰ ثانیه ارسال شود. منطق این مکانیزم را در گزارش توضیح دهید.

ج) MQTT:

- Broker (Mosquitto) روی کامپیوتر شخصی راه‌اندازی شود.
- کلاینت MQTT روی برد به زبان C نوشته شود.
- ساختار topic الزامی: persons/<student_id>/home و telemetry/<student_id>/home
- پیام‌ها به‌صورت JSON شامل تعداد افراد، دما و timestamp باشند.
- ارسال با QoS = 1 و پیام LWT (Last Will and Testament) پیکربندی شود تا در صورت قطع ناگهانی برد، کامپیوتر مطلع شود.

❖ **آزمایش‌های الزامی بخش سوم: (در گزارش حتما به ترتیب شماره آزمایش و نتیجه را ذکر کنید.)**

شماره	آزمایش	خروجی مورد انتظار
۳–۱	دقت تشخیص را در ۴ وضعیت نوری بسنجید: نور روز، نور مصنوعی، نور کم، نور از پشت (backlight)	جدول دقت (تشخیص صحیح/کل) + نمونه تصویر هر وضعیت
۳–۲	با یک عکس چاپ‌شده از انسان (روی کاغذ یا موبایل) سیستم را فریب دهید.	آیا تشخیص می‌دهد؟ تحلیل و پیشنهاد راه‌حل
۳–۳	رزولوشن ورودی را در سه سطح دلخواه تغییر دهید.	جدول FPS، دمای CPU پس از ۵ دقیقه، مصرف حافظه و دقت در هر رزولوشن + نتیجه‌گیری درباره نقطه بهینه
۳–۴	Broker را حین کار خاموش کنید و پس از ۳ دقیقه روشن کنید.	نمایش پیام LWT
۳–۵	تأخیر انتها به انتها از لحظه ورود فرد به کادر تا دریافت پیام MQTT روی کامپیوتر را بسنجید. (با مقایسه timestamp)	میانگین و انحراف معیار تأخیر روی ۱۰ اندازه‌گیری
۳–۶	تلاش برای ورود ناشناس یا غیر مجاز به MQTT	تصویر از ناموفق بودن اتصال
۳–۷	تلاش برای ورود غیر مجاز SSH	تصویر از ناموفق بودن اتصال

بخش چهارم: قابلیت های تکمیلی (۲۵ نمره)

۱. **سیستم ضدسرقت:** یک «حالت نگهبانی» که از طریق API فعال یا غیرفعال می‌شود در صفحه HTML قرار دهید. در حالت فعال، تشخیص هر انسان سبب ارسال ایمیل فوری با عکس، ارسال پیام MQTT با topic اضطراری alarm/<student_id>/home می شود.

۲. **ذخیره تاریخچه (جعبه سیاه):** ذخیره تاریخچه تشخیص‌ها در SQLite با بافر حلقوی انجام شود و با API گزارش گیری از کل تعداد دفعات تشخیص انسان انجام شود.

۳. **Watchdog نرم‌افزاری:** اگر پروسه پردازش تصویر بیش از ۳۰ ثانیه فریم جدید نداد (مثلاً دوربین قطع شد) سیستم آن را تشخیص دهد و ایمیل هشدار «دستکاری دوربین» بفرستد و سرویس را ری‌استارت کند.

۴. **مدیریت حرارتی تطبیقی:** اگر دمای CPU از آستانه ای گذشت، سامانه به‌صورت خودکار FPS یا رزولوشن را کاهش دهد و این رخداد را از طریق ایمیل ارسال کند.

❖ **آزمایش الزامی بخش چهارم: (در گزارش حتما به ترتیب شماره آزمایش و نتیجه را ذکر کنید.)**

شماره	آزمایش	خروجی مورد انتظار
۴–۱	عملکرد حالت نگهبان	فیلم این عملکرد و تصاویر آن را گزارش کنید.
۴–۲	عملکرد جعبه سیاه	تصویر ثبت رخداد ها در دیتابیس را بیاورید.
۴–۳	عملکرد watchdog نرم افزاری	دوربین را قطع کنید و از عملکرد برد فیلم و تصویر تهیه کنید.
۴–۴	عملکرد مدیریت حرارت تطبیقی	با ابزار های موجود در لینوکس سعی کنید دمای پردازنده را افزایش داده و تصاویر عملکردی این بخش را درگزارش بیاورید.

موارد تحویلی

۱. همه کد ها
۲. گزارش PDF شامل معماری، توضیح کد، تمام جداول و نمودارهای آزمایش‌ها، تحلیل نتایج و مشکلاتی که با آن‌ها روبه‌رو شدید و نحوه حل آن‌ها.
۳. فیلم‌های درخواستی در آزمایش
۴. فایل‌های پیکربندی شامل سرویس ها، کانفیگ Mosquitto، گواهی SSL و ...

نکات پایانی

- کار گروهی مجاز نیست. پرسش و بحث کلی در گروه درس آزاد است.
- در صورت نبود Orange Pi، اجرای پروژه روی VM لینوکسی پذیرفته است.

در صورت سوال از جزییات پروژه با ایمیل sh.sadeghzadeh@gmail.com مطرح کنید.

موفق باشید!