

## سیستم‌های نهفته

استاد انصاری

عاطفه قندهاری - مبینا حیدری - نیکا قادری  
پاییز ۱۴۰۴



گزارش فاز اول پروژه

### سیستم امنیتی هوشمند

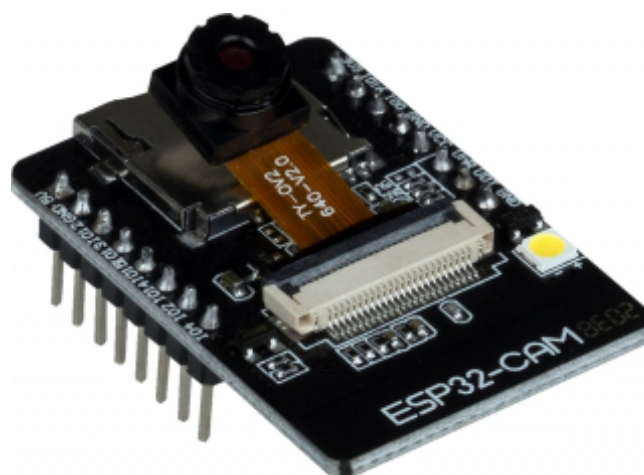
تاریخ گزارش: ۲۳ آبان ۱۴۰۴

#### فهرست مطالب

۱	۱ بررسی اجمالی پروژه
۱	۲ معماری سیستم و اجزای سخت‌افزاری
۲	۳ طراحی نرم‌افزار و استراتژی پیاده‌سازی
۲	۴ چالش‌های فنی
۲	۵ نتیجه‌گیری
۳	۶ پیوست: لیست قطعات سخت‌افزاری موردنیاز

#### ۱ بررسی اجمالی پروژه

هدف این پروژه، طراحی و پیاده‌سازی یک سیستم امنیتی هوشمند است که قابلیت‌های تشخیص حرکت، ایجاد بازدارندگی محلی و اطلاع‌رسانی *real-time* همراه با شواهد بصری را به‌صورت یکپارچه ارائه می‌دهد. هسته اصلی سیستم، برد توسعه *ESP32-CAM* است که یک میکروکنترلر قدرتمند، ماژول اتصال *WiFi* و دوربین را در ساختاری فشرده و کارآمد ادغام کرده است. با تشخیص حرکت توسط سنسور *Passive Infrared (PIR)*، سیستم به‌سرعت دو اقدام همزمان را انجام می‌دهد: فعال‌سازی هشدار محلی از طریق *buzzer* و چراغ هشدار متصل به ماژول *relay*، و در عین حال، گرفتن تصویر از محیط توسط دوربین و ارسال آن از طریق *WiFi* به یک *bot* در پیام‌رسان *Telegram*. این مکانیزم به کاربر امکان می‌دهد تا در لحظه وقوع رویداد، تأیید بصری فوری را از هر نقطه‌ای با دسترسی به اینترنت دریافت کند. برای افزایش امنیت و کنترل‌پذیری، یک رابط *keypad* به سیستم افزوده شده است که تنها با وارد کردن رمز عبور صحیح، امکان فعال یا غیر فعال کردن سیستم را فراهم می‌کند. این ترکیب هوشمند از پاسخ‌دهی سریع سخت‌افزاری، ارتباط شبکه‌ای پایدار و تعامل کاربرمحور، یک راه‌حل امنیتی جامع، متصل به اینترنت و کاملاً کاربردی برای محیط‌های مسکونی یا دفاتر کوچک ایجاد می‌نماید.



شکل ۱: برد *ESP32-CAM*

#### ۲ معماری سیستم و اجزای سخت‌افزاری

پردازنده مرکزی ماژول *ESP32-CAM* است که دارای پردازنده دوهسته‌ای *Tensilica LX6* با فرکانس تا  $240\text{ MHz}$ ،  $520\text{ KBRAM}$  و  $4\text{ MBPSRAM}$  برای بافرینگ تصاویر است. دوربین یکپارچه  $OV2640$  از فشرده‌سازی *JPEG* و رزولوشن‌های تا  $1200 \times 1600$  پیکسل

پشتیبانی می‌کند و آن را برای گرفتن عکس‌های واضح در شرایط نوری مختلف ایده‌آل می‌سازد. تشخیص حرکت توسط سنسور  $HC - SR501 PIR$  یا مدل‌های مشابه انجام می‌شود که به یک پین  $GPIO$  به عنوان ورودی متصل است. این سنسور بر اساس تغییرات تشعشعات مادون قرمز عمل می‌کند و عملکرد قابل اعتماد در محدوده تشخیص ۵-۷ متری را تضمین می‌کند. مکانیسم‌های هشدار محلی یک  $active\ piezo\ buzzer$  را شامل می‌شود که هنگام فعال‌سازی یک صدای بلند تولید می‌کند. همچنین یک یک ماژول  $5V\ relay\ single - channel$  نیز داریم که یک  $LED$  با روشنایی بالا یا لامپ خارجی را برای هشدار بصری متصل می‌کند.  $relay$  جریان و ولتاژ بالاتر دستگاه را از میکروکنترلر جدا می‌کند و از افت ولتاژ جلوگیری می‌کند. کنترل کاربر از طریق یک  $4 * 4\ keypad\ membrane$  ارائه می‌شود که با هشت پین دیجیتال متصل است و ورود رمز عبور (برای فعال‌سازی سیستم) را امکان‌پذیر می‌سازد. یک منبع تغذیه پایدار ۵V، که از طریق آداپتور یا باتری با تبدیل ولتاژ مناسب تأمین می‌شود، عملکرد مداوم تمام اجزا را تضمین می‌کند. از آنجایی که  $ESP32 - CAM$  فاقد رابط  $USB - to - serial$  محلی است، یک ماژول خارجی  $FTDI$  برای بارگذاری اولیه  $firmware$  و دیباگینگ سریال مورد نیاز است. این ماژول به پین‌های  $UART$  برد ( $U0RXD$  و  $U0TXD$ ) متصل می‌شود و یک رابط  $USB$  به کامپیوتر توسعه ارائه می‌دهد. در طول  $prototyping$ ، برد  $FTDI$  به طور گسترده برای  $flashing$  کد و نظارت خروجی سریال استفاده خواهد شد.

### ۳ طراحی نرم‌افزار و استراتژی پیاده‌سازی

نرم‌افزار با استفاده از  $Arduino IDE$  و هسته  $Arduino ESP32$  توسعه داده خواهد شد. برنامه به زبان  $C++$  نوشته شده و به صورت مدولار طراحی می‌شود؛ به گونه‌ای که توابع جداگانه‌ای برای خواندن سنسور، کنترل هشدار، مدیریت دوربین، ارتباط شبکه‌ای و کار با  $keypad$  تعریف می‌گردند. کتابخانه‌های اصلی مورد استفاده عبارتند از:  $ESP32Cam$  برای راه‌اندازی دوربین و ضبط فریم،  $WiFi.h$  برای برقراری ارتباط امن شبکه،  $UniversalTelegramBot$  برای ارتباط مبتنی بر  $HTTPS$  با  $Telegram Bot API$ ، و  $Keypad$  برای خواندن دقیق و قابل اعتماد صفحه کلید ماتریسی. اجرای برنامه با مرحله راه‌اندازی اولیه آغاز می‌شود: تنظیم پین‌های  $GPIO$ ، پیکربندی دوربین با پارامترهای بهینه مانند رزولوشن ( $640 * 480\ VGA$ ) و کیفیت متوسط  $JPEG$  (به منظور تعادل بین وضوح و حجم فایل)، و اتصال به یک  $access\ point\ WiFi$  از پیش تعیین شده. نظارت بر پین سنسور  $PIR$  یا از طریق  $polling$  در حلقه اصلی انجام می‌شود یا با استفاده از  $interrupt$  سخت‌افزاری برای واکنش سریع‌تر پیاده‌سازی خواهد شد. با تشخیص حرکت، سیستم به کمک قابلیت‌های چندوظیفگی  $FreeRTOS$  در  $ESP32$ ، دو اقدام موازی را اجرا می‌کند: یک  $task$  بلافاصله  $buzzer$  و  $relay$  را فعال می‌کند و  $task$  دیگر فریم را از دوربین ضبط کرده، آن را به فرمت  $JPEG$  تبدیل می‌کند و از طریق درخواست  $POST$  به  $Telegram bot$  ارسال می‌نماید. این درخواست شامل  $chat\ ID$  و  $API\ token$  دریافت شده از  $BotFather$  است. ادغام  $keypad$  با استفاده از یک ماشین حالت ( $state\ machine$ ) پیاده‌سازی می‌شود که کلیدهای فشرده شده را به ترتیب در یک بافر جمع‌آوری می‌کند، رشته وارد شده را با رمز عبور ذخیره شده در  $EEPROM$  مقایسه می‌کند و در صورت تطابق، حالت فعال/غیرفعال سیستم را تغییر می‌دهد. در صورت ورود نادرست، سیستم برای مدت کوتاهی قفل می‌شود تا از تلاش‌های مکرر جلوگیری شود. برای تضمین پایداری، مدیریت حافظه بسیار مهم است؛ بنابراین بافرهای تصویر بلافاصله پس از ارسال با فراخوانی  $esp\_camera\_fb\_return()$  آزاد می‌شوند و از تایمرهای  $watchdog$  برای جلوگیری از فریز سیستم در عملیات شبکه استفاده می‌گردد. همچنین، در صورت بروز خطا در آپلود تصویر، مکانیزم اتصال مجدد  $WiFi$  با الگوی  $exponential\ backoff$  و ارسال پیام‌های جایگزین ( $fallback$ ) پیاده‌سازی خواهد شد.

### ۴ چالش‌های فنی

برای اطمینان از عملکرد پایدار و قابل اعتماد سیستم، چندین چالش فنی مهم پیش‌بینی شده که هر یک نیازمند راه‌حل‌های دقیق هستند. حافظه  $RAM$  محدود  $ESP32 - CAM$  یکی از محدودیت‌های اصلی در پردازش تصاویر با رزولوشن بالا است؛ این مشکل با انتخاب رزولوشن پایین‌تر در زمان ضبط (مانند  $VGA$ ) و کاهش حداکثری تخصیص حافظه پویا برطرف خواهد شد. ناپایداری شبکه نیز می‌تواند ارسال اطلاع‌رسانی‌ها را مختل کند؛ به همین دلیل، منطق قوی برای اتصال مجدد به  $WiFi$  همراه با تنظیم آستانه‌های مناسب  $timeout$  در نرم‌افزار پیاده‌سازی می‌گردد. ورودی  $keypad$  به دلیل نویز الکتریکی و فشارهای مکانیکی کلیدها، نیازمند  $debouncing$  دقیق است تا از ثبت اشتباه یا تکراری کلیدها جلوگیری شود. این امر از طریق تأخیرهای نرم‌افزاری کنترل شده و اولویت‌بندی صحیح  $interrupt$ ها مدیریت خواهد شد. مصرف انرژی نیز، به‌ویژه در حالت استفاده از باتری، چالش مهمی است. برای بهینه‌سازی،  $ESP32$  در حالت عادی به  $light\ space\ sleep$  وارد می‌شود و تنها با دریافت  $interrupt$  از سنسور  $PIR$  بیدار می‌گردد؛ همچنین،  $peripherals$  غیرضروری مانند  $Bluetooth$  به‌طور کامل غیرفعال خواهند ماند. نویز الکترومغناطیسی ناشی از سوئیچینگ  $relay$  ممکن است بر عملکرد سنسور  $PIR$  تأثیر منفی بگذارد؛ این ریسک با جداسازی فیزیکی مسیرهای سیگنال، استفاده از کابل‌های شیلددار و افزودن خازن‌های  $decoupling$  در نقاط کلیدی مدار کاهش می‌یابد. در نهایت، امنیت ارتباطی نیز حیاتی است؛  $Telegram space bot$  در برابر دسترسی غیرمجاز با اعتبارسنجی دقیق منبع پیام‌های ورودی و محدودسازی دستورات قابل اجرا به کاربران از پیش تأیید شده محافظت خواهد شد.

### ۵ نتیجه‌گیری

در پایان، این پروژه یک سیستم امنیتی هوشمند، کم‌هزینه و کاملاً متصل به اینترنت را ارائه می‌دهد که با ترکیب تشخیص حرکت، هشدار محلی و اطلاع‌رسانی تصویری  $real - time$ ، امنیت محیط را به‌طور مؤثر ارتقا می‌بخشد. استفاده از برد  $ESP32 - CAM$  به عنوان هسته مرکزی، امکان پیاده‌سازی یک راه‌حل جمع‌وجور، قدرتمند و قابل توسعه را فراهم کرده است. ادغام اجزای سخت‌افزاری مانند سنسور  $PIR$ ،  $buzzer$ ،  $relay$  و  $keypad$  با نرم‌افزار ماژولار و بهینه شده، سیستمی پایدار، قابل کنترل و مقاوم در برابر خطا ایجاد می‌کند. قابلیت فعال‌سازی با رمز عبور و ارسال فوری تصاویر از طریق  $Telegram$ ، نه تنها تجربه کاربری را بهبود می‌بخشد، بلکه پاسخگویی سیستم را در شرایط واقعی تضمین می‌نماید.

## ۶ پیوست: لیست قطعات سخت‌افزاری موردنیاز

- ESP32-CAM module (with OV2640 camera)
- HC-SR501 PIR motion sensor
- Active 5V piezo buzzer
- Single-channel 5V relay module
- High-brightness LED or 5V warning lamp
- 4×4 membrane keypad
- FTDI USB-to-Serial programmer
- Stable 5V power supply ( 1A)
- Breadboard and jumper wires