گزارش پیشرفت هفته اول

پروژه ۱: پایش علائم حیاتی

آزمایشگاه اینترنت اشیا - DISCOVERY-STM32F746G

اعضای تیم:

درسا ضابطی - 403171051

یاسمن فرخی - 403110409

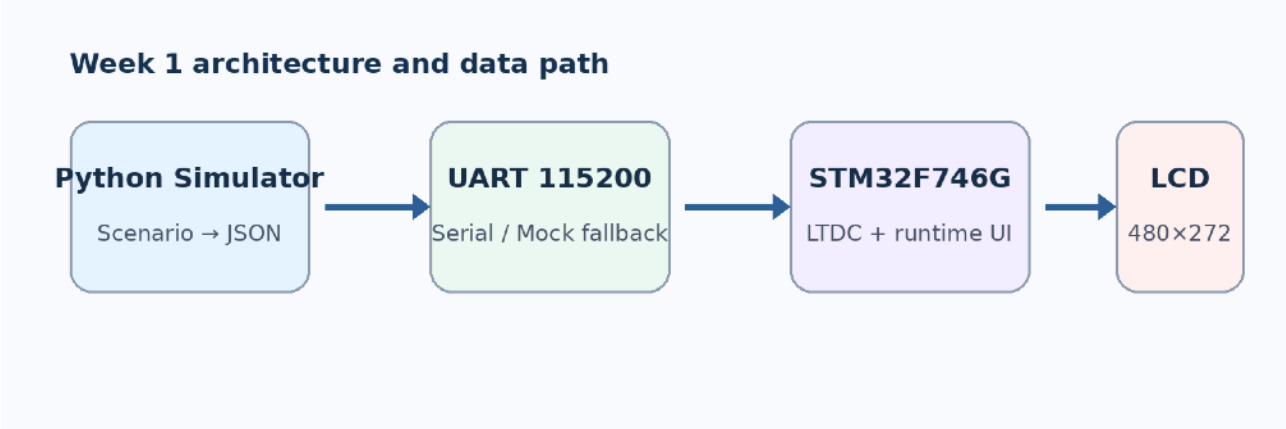
آراد ایزدی‌دوست - 403105758

۱. خلاصه اجرایی

هدف پروژه طراحی یک سامانه پایش علائم حیاتی است که داده‌های ضربان قلب (HR)، درصد اکسیژن خون (SpO_2)، دمای بدن و سیگنال ECG را دریافت و روی نمایشگر برد DISCOVERY-STM32F746G نمایش دهد. در فازهای بعدی، تشخیص وضعیت‌های غیرعادی و ارسال هشدار شبکه نیز به سامانه اضافه می‌شود.

در هفته اول، به‌جای ورود مستقیم به اتصال سنسورها، سه زیرسامانه به‌صورت موازی جلو رفت: شبیه‌ساز داده در پایتون توسط درسا، طراحی Assetهای رابط کاربری توسط فرخی، و up-Bring برد و LCD و یکپارچه‌سازی UI توسط آراد.

نتیجه هفته اول یک رابط کاربری واقعی روی LCD برد است که Splash Screen، پس‌زمینه سفید، Assetهای گرافیکی، قلب متحرک، Gauge دارای عقربه، دماسنج بزرگ‌تر و مقدار آزمایشی HR: 76 را نمایش می‌دهد.



معماری مورد استفاده برای حرکت از شبیه‌سازی *PC* به نمایش روی *STM32*

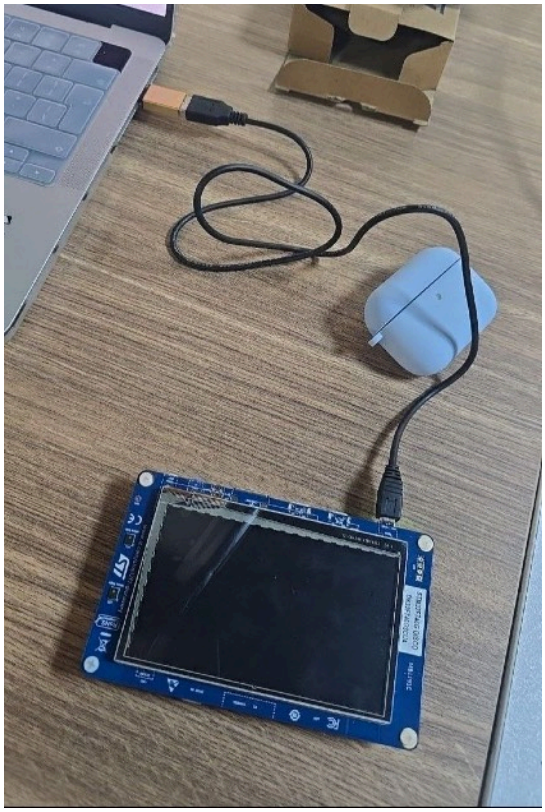
۲. تقسیم کار و خروجی هر عضو

عضو	مسئولیت هفته اول	خروجی تحویلی
درسا ضابطی	طراحی شبیه‌ساز علائم حیاتی و رفتار Mock/UART	تولید HR، Temp، SpO_2 و ECG در سناریوهای مختلف و خروجی JSON
یاسمن فرخی	طراحی Assetهای رابط کاربری	Heart animation، Gauge، Gauge، Arrow، Thermometer و فایل‌های گرافیکی قابل استفاده روی LCD
آراد ایزدی‌دوست	up-Bring برد، Flash/Build، راه‌اندازی LCD و Integration UI	LCD فعال، Layerهای LTDC، UI سفید، قلب متحرک شفاف، Splash Screen و HR آزمایشی

۳. کارهای آزاد: از up-Bring برد تا UI نهایی

۳-۱. اتصال و تایید سخت‌افزار

برد DISCO-STM32F746G از طریق پورت USB-Mini مربوط به LINK-ST به لپ‌تاپ متصل شد. ابتدا اتصال LINK-ST و ولتاژ Target بررسی شد و پس از رفع مشکل اتصال، برنامه‌ریزی برد با STM32CubeProgrammer انجام شد.

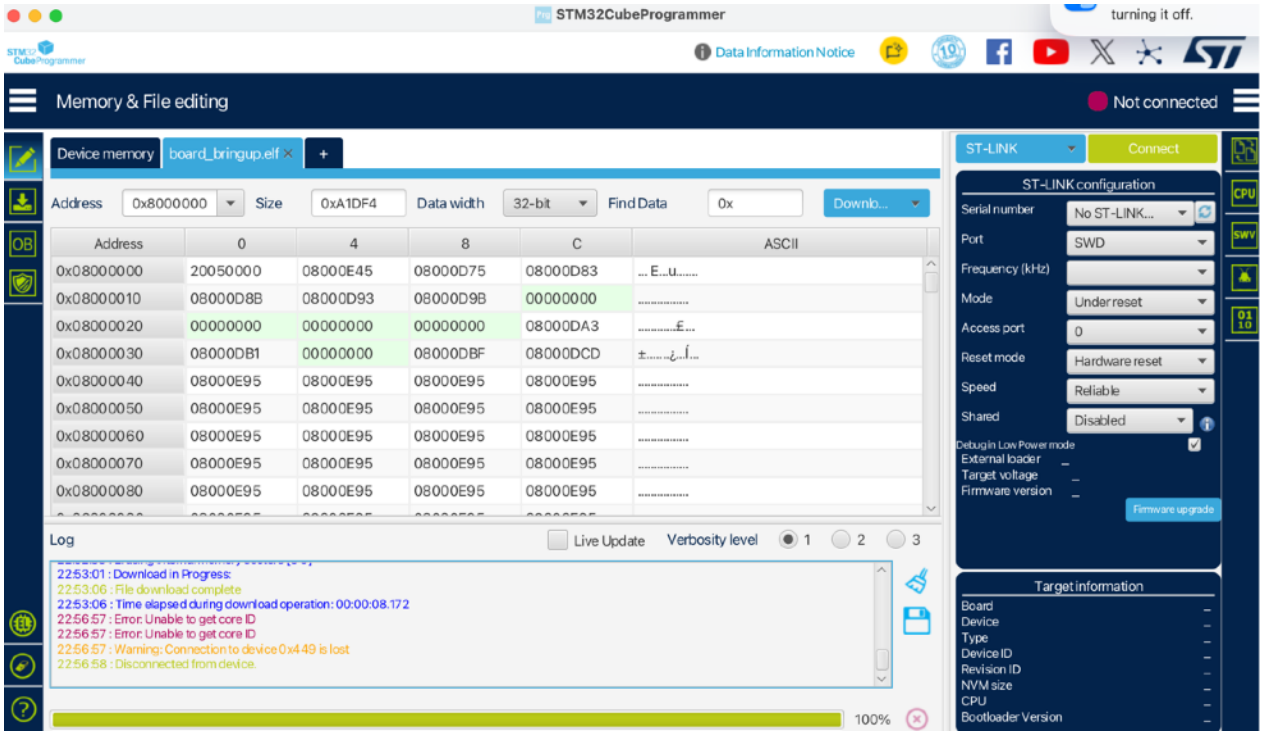


سمت راست: برنامه پیش‌فرض برد؛ سمت چپ: اتصال برد به لپ‌تاپ از طریق LINK-ST

برای اطمینان از صحت Toolchain و اجرای کد، یک برنامه metal-Bare ساده نوشته شد که LED سبز برد را با دسترسی مستقیم به رجیسترهای RCC و GPIO روشن و خاموش می‌کرد. موفقیت این تست نشان داد مسیر Build → ELF → Flash → Execute سالم است.

۳-۲. آماده‌سازی محیط Build و Flash

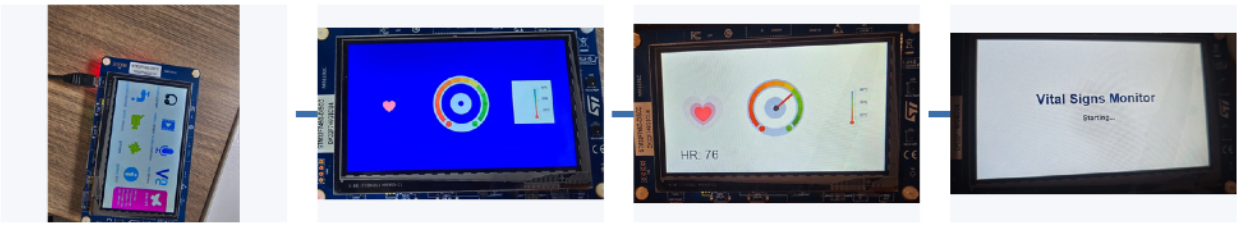
پروژه اولیه با CMake و GNU Arm Embedded Toolchain ساخته شد. در طول up LCD-Bring، مسیرهای HAL و CMSIS و تعریف‌های STM32F746xx و DRIVER_HAL_USE به Build اضافه شدند. برای Flash از STM32CubeProgrammer استفاده شد.



STM32CubeProgrammer در جریان تست و برنامه‌ریزی *ELF* روی برد

۳-۳. راه‌اندازی LCD با مثال رسمی ST

برای جلوگیری از نوشتن دستی درایور LCD، مثال رسمی STM32CubeF7 با مسیر 1Layer_Display_LTDC برای برد Discovery-STM32746G مبنا قرار گرفت. مثال رسمی ابتدا یک تصویر RGB565 با اندازه 272×480 را مستقیم از Flash روی LCD نمایش داد. سپس تصویر نمونه حذف شد و خروجی به یک صفحه تکرنگ و بعد به UI اختصاصی پروژه تبدیل شد.



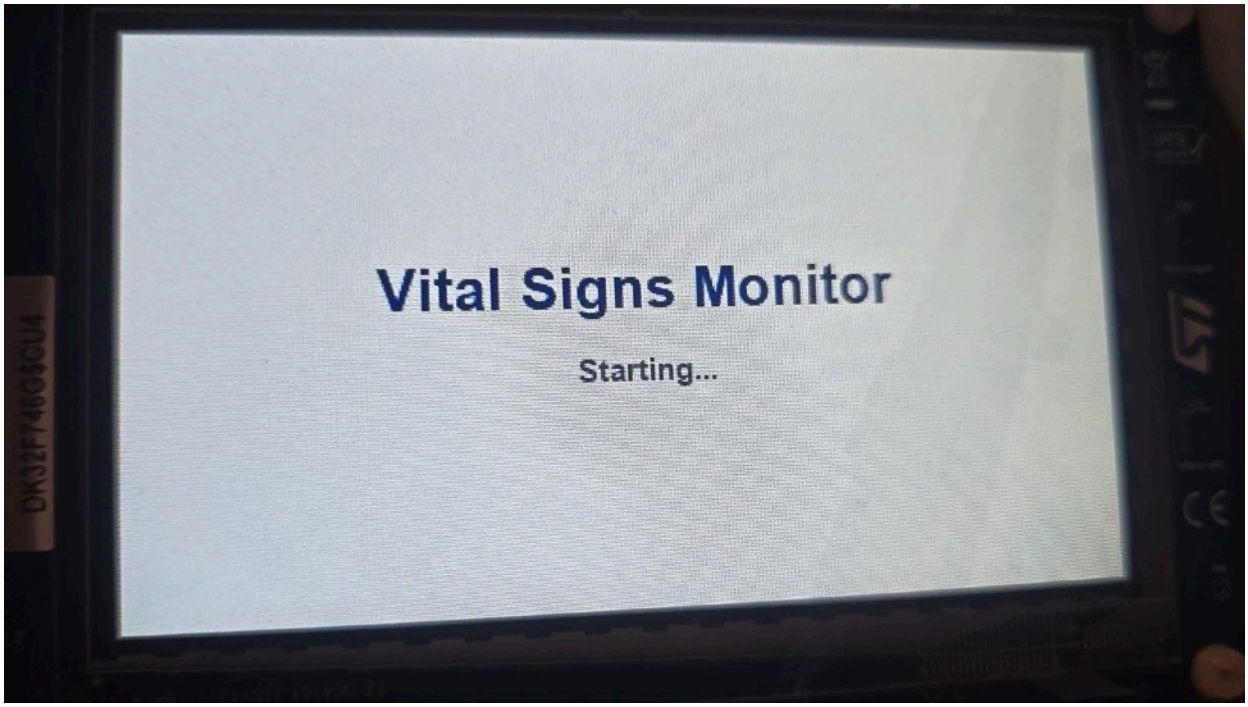
1. Official demo
2. First custom UI
3. Final Week 1 UI
4. Splash screen

تکامل خروجی *LCD* از مثال رسمی تا *UI* نهایی هفته اول

۴-۳. ساخت UI و Layerهای LTDC

رابط کاربری نهایی هفته اول از دو لایه LTDC استفاده می‌کند: لایه پایه برای UI ثابت با فرمت RGB565 و لایه جدا برای انیمیشن قلب با ARGB4444. استفاده از ARGB4444 امکان شفافیت پیکسلی را فراهم کرد و مربع رنگی اطراف قلب حذف شد.

- پس‌زمینه نهایی: سفید
- قلب: ۸ فریم متحرک روی Layer مستقل
- Gauge: دارای عقربه ثابت برای خواناتر شدن مفهوم
- Thermometer: بزرگ‌تر و واضح‌تر نسبت به نسخه اولیه
- HR: 76: مقدار آزمایشی ثابت برای اثبات امکان قرار دادن عدد روی UI
- Splash Screen: نمایش Vital Signs Monitor و Starting... هنگام راه‌اندازی، سپس ورود به Dashboard

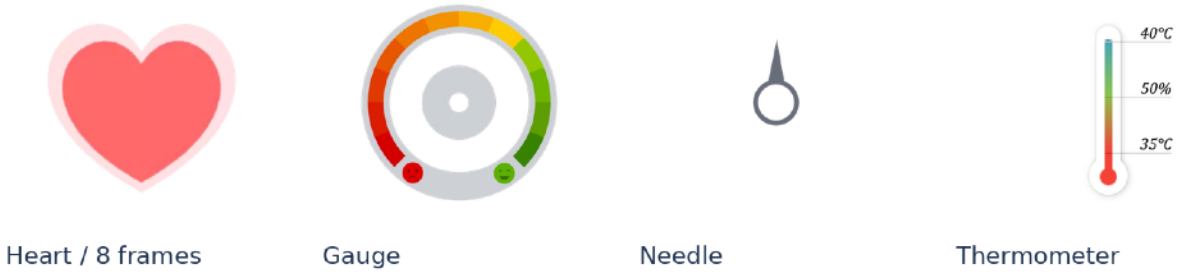


Splash Screen نهایی هنگام روشن شدن برد

Dashboard نهایی *Week 1*؛ مقدار *HR* در این مرحله آزمایشی و ثابت است

۴. کارهای فرخی: طراحی Assetهای رابط کاربری

یاسمن فرخی Assetهای اصلی UI را آماده کرد تا رابط کاربری از حالت تست فنی خارج و به یک Dashboard قابل ارائه نزدیک شود. فایل‌های تحویلی شامل نسخه‌های PNG و PSD و فریم‌های انیمیشن قلب هستند.



نمونه Assetهای تحویلی فرخی: *Heart*، *Gauge*، *Gauge Arrow* و *Thermometer*

- Heart: فایل Foreground، Background، PSD و ۸ فریم انیمیشن Heart1 تا Heart8
- Gauge: فایل PNG/PSD و فایل مجزای GaugeArrow
- Thermometer: فایل PNG/PSD و Bubble

در مرحله Integration، Assetها برای رزولوشن 272×480 Resize و به داده bit-16 مناسب LTDC تبدیل شدند. قلب به‌دلیل نیاز به Transparency با ARGB4444 و سایر بخش‌های ثابت UI با RGB565 استفاده شدند.

۵. کارهای درسا: Python Vital Signs Simulator

به‌دلیل نبود سنسور و تجهیزات فیزیکی در اختیار درسا، شبیه‌ساز طوری نوشته شد که بدون UART واقعی هم قابل اجرا و تست باشد. اگر PORT تعریف نشده باشد یا باز کردن/نوشتن پورت Serial خطا بدهد، برنامه به Mock mode می‌رود و داده‌ها را در ترمینال چاپ می‌کند. در پایتون این مدیریت خطا با except/try انجام شده است.

سناریو از داخل کد انتخاب می‌شود. برای مثال در نسخه فعلی می‌توان SCENARIO را روی Normal یا Tachycardia قرار داد. سپس HR، SpO_2 و Temp در محدوده متناسب با سناریو تولید می‌شوند و با حرکت تدریجی به سمت Targetهای تصادفی، نویز کوچک و ECG شبه‌فیزیولوژیک، خروجی از حالت کاملاً ثابت خارج می‌شود.

- Normal: HR حدود 68–82، SpO_2 حدود 96–99، Temp حدود 36.5–37.2
- Tachycardia: HR حدود 110–140، SpO_2 حدود 95–99، Temp حدود 36.5–37.3
- سناریوهای پیاده‌سازی‌شده دیگر: Low_SpO2، Bradycardia و Abnormal_Temp
- ساختار داده: temp، spo2، hr، scenario و ecg
- فرمت ارتباطی: یک خط JSON برای هر نمونه
- Baud Rate هدف برای UART: 115200

```
# dorsazabeti @ Dorsas-MacBook-Pro in ~/Documents/Uni/ij iot/Vital_Simulator on git:main o [22:51:11]
$ python3 main.py
Mock: {"scenario":"Tachycardia","hr":112,"spo2":97,"temp":36.8,"ecg":0.081}
Mock: {"scenario":"Tachycardia","hr":113,"spo2":97,"temp":36.9,"ecg":0.175}
Mock: {"scenario":"Tachycardia","hr":114,"spo2":97,"temp":36.9,"ecg":0.258}
Mock: {"scenario":"Tachycardia","hr":115,"spo2":97,"temp":36.9,"ecg":0.384}
Mock: {"scenario":"Tachycardia","hr":115,"spo2":97,"temp":36.9,"ecg":0.352}
Mock: {"scenario":"Tachycardia","hr":116,"spo2":97,"temp":36.9,"ecg":0.248}
Mock: {"scenario":"Tachycardia","hr":117,"spo2":97,"temp":36.8,"ecg":0.146}
Mock: {"scenario":"Tachycardia","hr":117,"spo2":98,"temp":36.8,"ecg":0.225}
Mock: {"scenario":"Tachycardia","hr":118,"spo2":98,"temp":36.8,"ecg":0.142}
Mock: {"scenario":"Tachycardia","hr":119,"spo2":98,"temp":36.8,"ecg":0.101}
Mock: {"scenario":"Tachycardia","hr":119,"spo2":98,"temp":36.8,"ecg":0.167}
Mock: {"scenario":"Tachycardia","hr":120,"spo2":98,"temp":36.8,"ecg":0.146}
Mock: {"scenario":"Tachycardia","hr":121,"spo2":98,"temp":36.8,"ecg":0.164}
Mock: {"scenario":"Tachycardia","hr":121,"spo2":98,"temp":36.8,"ecg":0.211}
Mock: {"scenario":"Tachycardia","hr":122,"spo2":98,"temp":36.8,"ecg":0.376}
Mock: {"scenario":"Tachycardia","hr":122,"spo2":98,"temp":36.9,"ecg":0.254}
Mock: {"scenario":"Tachycardia","hr":123,"spo2":97,"temp":36.9,"ecg":0.133}
Mock: {"scenario":"Tachycardia","hr":123,"spo2":97,"temp":36.9,"ecg":-0.004}
Mock: {"scenario":"Tachycardia","hr":124,"spo2":97,"temp":36.9,"ecg":-0.005}
Mock: {"scenario":"Tachycardia","hr":124,"spo2":97,"temp":36.9,"ecg":0.013}
Mock: {"scenario":"Tachycardia","hr":123,"spo2":97,"temp":36.9,"ecg":0.026}
Mock: {"scenario":"Tachycardia","hr":122,"spo2":97,"temp":36.9,"ecg":-0.004}
Mock: {"scenario":"Tachycardia","hr":121,"spo2":97,"temp":36.9,"ecg":-0.013}
Mock: {"scenario":"Tachycardia","hr":121,"spo2":97,"temp":36.9,"ecg":-0.019}
Mock: {"scenario":"Tachycardia","hr":121,"spo2":97,"temp":36.9,"ecg":-0.002}
Mock: {"scenario":"Tachycardia","hr":121,"spo2":97,"temp":36.9,"ecg":0.078}
Mock: {"scenario":"Tachycardia","hr":121,"spo2":97,"temp":36.9,"ecg":0.097}
Mock: {"scenario":"Tachycardia","hr":122,"spo2":97,"temp":36.9,"ecg":0.015}
Mock: {"scenario":"Tachycardia","hr":122,"spo2":97,"temp":37.0,"ecg":-0.043}
Mock: {"scenario":"Tachycardia","hr":123,"spo2":97,"temp":37.0,"ecg":0.771}
Mock: {"scenario":"Tachycardia","hr":123,"spo2":97,"temp":37.0,"ecg":-0.252}
Mock: {"scenario":"Tachycardia","hr":124,"spo2":97,"temp":37.0,"ecg":-0.0}
Mock: {"scenario":"Tachycardia","hr":125,"spo2":97,"temp":37.0,"ecg":0.029}
Mock: {"scenario":"Tachycardia","hr":125,"spo2":97,"temp":37.0,"ecg":0.186}
Mock: {"scenario":"Tachycardia","hr":126,"spo2":97,"temp":37.0,"ecg":0.305}
Mock: {"scenario":"Tachycardia","hr":125,"spo2":97,"temp":37.0,"ecg":0.114}
Mock: {"scenario":"Tachycardia","hr":125,"spo2":97,"temp":37.0,"ecg":0.016}
Mock: {"scenario":"Tachycardia","hr":124,"spo2":97,"temp":37.0,"ecg":0.021}
Mock: {"scenario":"Tachycardia","hr":123,"spo2":97,"temp":37.0,"ecg":0.037}
Mock: {"scenario":"Tachycardia","hr":123,"spo2":97,"temp":37.0,"ecg":0.017}
Mock: {"scenario":"Tachycardia","hr":122,"spo2":97,"temp":37.0,"ecg":-0.011}
Mock: {"scenario":"Tachycardia","hr":121,"spo2":97,"temp":37.0,"ecg":-0.02}
Mock: {"scenario":"Tachycardia","hr":121,"spo2":97,"temp":37.0,"ecg":-0.01}
Mock: {"scenario":"Tachycardia","hr":122,"spo2":97,"temp":37.0,"ecg":0.008}
Mock: {"scenario":"Tachycardia","hr":122,"spo2":97,"temp":37.0,"ecg":0.004}
```

```
Problems   Output   Debug Console   Terminal   Ports

# dorsazabeti @ Dorsas-MacBook-Pro in ~/Documents/Uni/ij iot/Vital_Simulator on git:main x [22:52:50]
$ python3 main.py
[*] No UART port selected. Running in MOCK mode.
[*] Available serial ports:
    /dev/cu.debug-console
    /dev/cu.Bluetooth-Incoming-Port
[*] Scenario: Normal
[*] Press Ctrl+C to stop.
Mock: {"scenario":"Normal","hr":74,"spo2":98,"temp":36.8,"ecg":-0.001}
Mock: {"scenario":"Normal","hr":74,"spo2":98,"temp":36.8,"ecg":0.315}
Mock: {"scenario":"Normal","hr":74,"spo2":98,"temp":36.8,"ecg":0.028}
Mock: {"scenario":"Normal","hr":73,"spo2":98,"temp":36.8,"ecg":0.001}
Mock: {"scenario":"Normal","hr":73,"spo2":98,"temp":36.8,"ecg":-0.01}
Mock: {"scenario":"Normal","hr":74,"spo2":98,"temp":36.8,"ecg":-0.02}
Mock: {"scenario":"Normal","hr":74,"spo2":98,"temp":36.8,"ecg":0.043}
Mock: {"scenario":"Normal","hr":74,"spo2":98,"temp":36.9,"ecg":1.174}
Mock: {"scenario":"Normal","hr":74,"spo2":98,"temp":36.9,"ecg":0.011}
Mock: {"scenario":"Normal","hr":75,"spo2":98,"temp":36.9,"ecg":0.348}
Mock: {"scenario":"Normal","hr":75,"spo2":98,"temp":36.9,"ecg":-0.012}
Mock: {"scenario":"Normal","hr":75,"spo2":98,"temp":36.9,"ecg":-0.029}
Mock: {"scenario":"Normal","hr":75,"spo2":98,"temp":36.9,"ecg":0.062}
Mock: {"scenario":"Normal","hr":75,"spo2":98,"temp":36.9,"ecg":0.114}
Mock: {"scenario":"Normal","hr":75,"spo2":98,"temp":36.9,"ecg":0.015}
Mock: {"scenario":"Normal","hr":75,"spo2":98,"temp":36.9,"ecg":0.008}
Mock: {"scenario":"Normal","hr":75,"spo2":98,"temp":36.9,"ecg":0.004}
Mock: {"scenario":"Normal","hr":75,"spo2":98,"temp":36.9,"ecg":0.15}
Mock: {"scenario":"Normal","hr":76,"spo2":98,"temp":36.9,"ecg":1.043}
Mock: {"scenario":"Normal","hr":77,"spo2":98,"temp":36.9,"ecg":0.026}
Mock: {"scenario":"Normal","hr":77,"spo2":98,"temp":37.0,"ecg":0.235}
Mock: {"scenario":"Normal","hr":77,"spo2":98,"temp":37.0,"ecg":1.151}
Mock: {"scenario":"Normal","hr":77,"spo2":98,"temp":37.0,"ecg":-0.021}
Mock: {"scenario":"Normal","hr":77,"spo2":98,"temp":37.0,"ecg":0.168}
Mock: {"scenario":"Normal","hr":77,"spo2":98,"temp":37.0,"ecg":0.543}
Mock: {"scenario":"Normal","hr":76,"spo2":98,"temp":37.0,"ecg":0.025}
Mock: {"scenario":"Normal","hr":76,"spo2":98,"temp":36.9,"ecg":0.312}
Mock: {"scenario":"Normal","hr":75,"spo2":98,"temp":36.9,"ecg":-0.152}
Mock: {"scenario":"Normal","hr":75,"spo2":98,"temp":36.9,"ecg":-0.024}
Mock: {"scenario":"Normal","hr":74,"spo2":98,"temp":36.9,"ecg":0.324}
Mock: {"scenario":"Normal","hr":74,"spo2":98,"temp":36.9,"ecg":0.0}
Mock: {"scenario":"Normal","hr":73,"spo2":98,"temp":36.9,"ecg":0.011}
Mock: {"scenario":"Normal","hr":73,"spo2":98,"temp":36.9,"ecg":0.031}
Mock: {"scenario":"Normal","hr":74,"spo2":97,"temp":36.9,"ecg":0.023}
Mock: {"scenario":"Normal","hr":74,"spo2":97,"temp":36.9,"ecg":0.044}
Mock: {"scenario":"Normal","hr":75,"spo2":97,"temp":36.9,"ecg":-0.03}
Mock: {"scenario":"Normal","hr":75,"spo2":97,"temp":36.9,"ecg":0.02}
Mock: {"scenario":"Normal","hr":75,"spo2":97,"temp":36.9,"ecg":0.335}
Mock: {"scenario":"Normal","hr":75,"spo2":97,"temp":36.9,"ecg":0.01}
Mock: {"scenario":"Normal","hr":75,"spo2":97,"temp":36.8,"ecg":-0.005}
Mock: {"scenario":"Normal","hr":75,"spo2":97,"temp":36.9,"ecg":0.122}
Mock: {"scenario":"Normal","hr":74,"spo2":97,"temp":36.9,"ecg":0.016}
Mock: {"scenario":"Normal","hr":74,"spo2":97,"temp":36.9,"ecg":0.02}
Mock: {"scenario":"Normal","hr":73,"spo2":97,"temp":36.9,"ecg":0.023}
Mock: {"scenario":"Normal","hr":73,"spo2":97,"temp":36.9,"ecg":0.018}
Mock: {"scenario":"Normal","hr":72,"spo2":97,"temp":36.9,"ecg":0.268}
Mock: {"scenario":"Normal","hr":71,"spo2":97,"temp":36.9,"ecg":-0.038}
Mock: {"scenario":"Normal","hr":71,"spo2":97,"temp":36.9,"ecg":-0.007}
^C
[!] Stopped.
```

خروجی *Mock* شبیه‌ساز: سناریوی *Normal* و *Tachycardia*

نمونه فریم JSON: {0.001-:"ecg",36.8:"temp",98:"spo2",74:"hr","Normal":"scenario"}

۶. چالش‌ها و تصمیم‌های فنی مهم

- مشکل Target Voltage و اتصال ST-LINK: با بررسی کابل/اتصال و CubeProgrammer برطرف شد.
- روش اولیه افزودن دستی وابستگی‌های LCD باعث خطاهای HAL/CMSIS شد؛ مسیر به مثال رسمی STM32CubeF7 تغییر کرد.
- برای تست اولیه، framebuffer تصویر از Flash خوانده شد تا SDRAM و پیچیدگی‌های اضافه وارد Week 1 نشوند.
- در اجرای runtime، دو متغیر hltdc_F باعث Shadow شدن Handle شده بود؛ با حذف تعریف محلی، تغییرات Layer در runtime فعال شد.
- انیمیشن قلب ابتدا با RGB565 مربع پس‌زمینه داشت؛ با Layer مستقل ARGB4444 و Pixel Alpha مشکل Transparency حل شد.
- TouchGFX در Week 1 استفاده نشد؛ برای رسیدن سریع و پایدار به LTDC، LCD working prototype مستقیم و مثال رسمی ST مبنا قرار گرفت.

۷. وضعیت پایان هفته اول

- Board bring-up و اجرای کد روی STM32F746NGH6: انجام شد
- Build با CMake و GNU Arm Toolchain: انجام شد
- Flash با STM32CubeProgrammer: انجام شد
- LCD 480×272 و LTDC: راه‌اندازی شد
- Assetهای UI فرخی: یکپارچه شد
- Heart animation شفاف: انجام شد
- Gauge needle و Thermometer اصلاح‌شده: انجام شد
- Splash Screen: انجام شد
- شبیه‌ساز Python درسا با Mock mode و سناریوها: آماده است
- نمایش عدد آزمایشی HR: 76 روی LCD: انجام شد

۸. پیشنهاد برای هفته دوم

هفته دوم می‌تواند با Integration واقعی داده شروع شود. مهم‌ترین Milestone بعدی این است که خروجی JSON شبیه‌ساز درسا از طریق UART به STM32 برسد و مقدار HR ثابت با مقدار واقعی دریافتی جایگزین شود.

- اتصال Python Simulator به UART واقعی برد و دریافت فریم JSON
- Parse کردن temp، spo2، hr و ecg در STM32
- Dynamic کردن HR و سپس SpO_2 و Temperature روی LCD
- هماهنگ کردن سرعت Heart animation با HR دریافتی
- در ادامه: رسم ECG، تشخیص وضعیت غیرعادی و هشدار شبکه

۹. جمع‌بندی

Week 1 با یک Prototype قابل نمایش بسته شد: برد برنامه‌ریزی می‌شود، LCD با LTDC کار می‌کند، UI اختصاصی روی نمایشگر آمده، Assetهای طراحی‌شده در سخت‌افزار واقعی استفاده شده‌اند و Python Simulator نیز مستقل از سنسور قابل اجراست. این خروجی پایه مناسبی برای شروع Integration داده در Week 2 است.