

بسمه تعالی



آزمایشگاه سخت افزار

گزارش نهایی

دکتر اجلالی

مهدی علیزاده ۹۹۱۰۱۹۲۳

نیما نجفی ۹۹۱۰۲۳۸۳

محمدحسین سلیمی ۹۹۱۰۱۷۳۸

دانشگاه صنعتی شریف

پاییز ۱۴۰۳

گزارش نهایی پروژه QT

مقدمه:

Qt یک Framework Application با قابلیت اجرا در سیستم عامل های مختلف است. با این نرم افزار میتوان نرم افزار هایی را توسعه داد که در پلتفرم های مختلف (هم نرم افزاری هم سخت افزاری) بدون کمترین تغییر در کد قابلیت اجرا داشته باشند. این Framework به طور پی شفرض از زبان های C++ و QML استفاده می کند. در این پروژه در ابتدا یک برنامه با رابط کاربری در پلتفرم QT نوشته می شود. در این برنامه بایستی تعدادی سنسور و عملگر به صورت مجازی تعریف کرد و از طریق اینترنت بتوان با این سیستم ارتباط برقرار کرد. در نهایت برنامه ی QT کامپایل شده را به همراه لایبرری های مورد نیاز بر روی یک image قرار می دهیم و image کاستوم خود را تولید می کنیم.

طراحی اپلیکیشن QT:

این قسمت چالش خیلی سختی نداشت و در همان مراحل ابتدایی تکمیل شد. این قسمت شامل تعریف یک سری سنسور و کلید بود که بتوانند از طریق وب سوکت دیتا به سمت سرور ارسال کنند و کنترل شوند. برای این کار از پروتکل websocket و پلاگین آن در QT استفاده کردیم و دیزاین های اولین برای قرارگیری سنسورهای مجازی را نیز تکمیل کردیم. در نهایت نیز با استفاده از قابلیت که خود QT در اختیار قرار می دهد این اپلیکیشن را compile و build کردیم تا یک نسخه ی باینری داشته باشید. لازم به ذکر است این بخش دقیقاً مطابق تخمین زمانی اولین انجام شد.

طراحی سرور:

در این قسمت نیز تنها باید یک سایت که هم فرانت و هم بکند داشته باشید دولوپ می کردیم که از وب سوکت پشتیبانی کند و کاربر بتواند از طریق ui به اپلیکیشن qt فرمان ارسال کند و همچنین دیتاهای مورد نیاز را دریافت کند. در این بخش نیز از django استفاده کردیم تا هم بتوانیم بخش های بکند را با استفاده از پلاگین های جنگو دولوپ کنیم و با استفاده از static های django بتوانیم بخش فرانت را بزیم. این بخش نیز چالش خاصی نداشت و توانستیم به موقع و طبق زمان بندی آن را انجام دهیم.

ساخت image و فرآیند boot:

در این قسمت ما باید ابتدا یک base image انتخاب می‌کردیم و سپس ابزارهای مورد نیاز خود برای اجرا اپلیکیشن qt و همچنین نمایش بر روی صفحه‌ی LCD را بر روی آن نصب می‌کردیم و در نهایت فایل بیلد شده‌ی اپلیکیشن را بر روی آن قرار می‌دادیم تا بتوانیم ایمج نهایی را بسازیم و بر روی رزبری پای بوت کنیم. این فرآیند به نحوی چالشی‌ترین بخش پروژه بود.

ما در ابتدا به پیشنهاد TA به سراغ ابزارهایی مانند boot to QT رفتیم ولی متوجه شدیم که این ابزار پولی است و ما لایسنس آن را نداریم. در ادامه به سراغ ابزارهای دیگر و روش‌های دیگر برای ساخت ایمیل رفتیم. ابتدا سعی کردیم از فیچرهایی که vmware در اختیار ما قرار می‌دهد استفاده کنیم و یک vm از روی یک image اولیه بسازی و ابزارهای مورد نیاز را بر روی آن نصب کنیم و در نهایت با استفاده از فیچر vmware تصویر نهایی را اکسپورت بگیریم و روی رزبری پای بوت کنیم. این کار را هم انجام دادیم ولی در نهایت متوجه شدیم از آنجایی که پروژه نیاز به صفحه نمایش دارد، نیاز به نصب یک gui داریم که با توجه به اینکه رزبری پای ما قدیمی بود و مقدار RAM کمی داشت، توانایی load کردن gui اوبونتو را نداشت.

در مرحله‌ی بعد به پیشنهاد TAها از image های کاستومی که برای رزبری پای طراحی شده اند به عنوان base image استفاده کردیم. مشکل gui حل شد.

مشکل بعدی‌ای که در این فرآیند داشتیم عدم تطبیق ورژن QT و shared library های روی رزبری پای با ورژن برنامه ی build شده‌ای بود که خودمان نوشته بودیم. مدت زیادی با این مشکل سر و کله زدیم تا در نهایت متوجه شدیم این ورژنی که به صورت دیفالت روی رزبری پای از طریق apt نصب می‌شود از بعضی از کامپوننت‌های پشتیبانی نمی‌کند. در نهایت برای حل مشکل یک base image رزبری پای را روی یک ماشین mount کردیم و لایبریهای مشترک و برنامه‌ی بیلد شده را از طریق qemu و chroot به ایمیل منتقل کردیم و در نهایت ایمج را تولید کردیم. همچنین تمامی این مراحل در اسکریپتی که در کنار پروژه، قرار دارد و می‌توان به صورت اتوماتیک به هر base ایمج و با هر ورژنی ایمج مربوطه را بیلد و استفاده کرد.