بسمه تعالی

دانشکده مهندسی کامپیوتر دانشگاه صنعتی امیرکبیر

(پلی‌تکنیک تهران)

درس مبانی اینترنت اشیا نیمسال دوم ۱۴۰۰ – ۱۴۰۱

تمرین سری چهارم

دانشجویان محترم توجه داشته باشند که تنها موظف به پاسخگویی به یکی از نسخه‌های تمرین هستند. لذا در صورت انتخاب نسخه‌ی نرم‌افزاری (شبیه‌سازی) نیازی به انجام نسخه سخت‌افزاری نخواهند بود و یا در صورت انتخاب نسخه‌ی سخت‌افزاری نیازی به انجام نسخه‌ی نرم‌افزاری نیست.

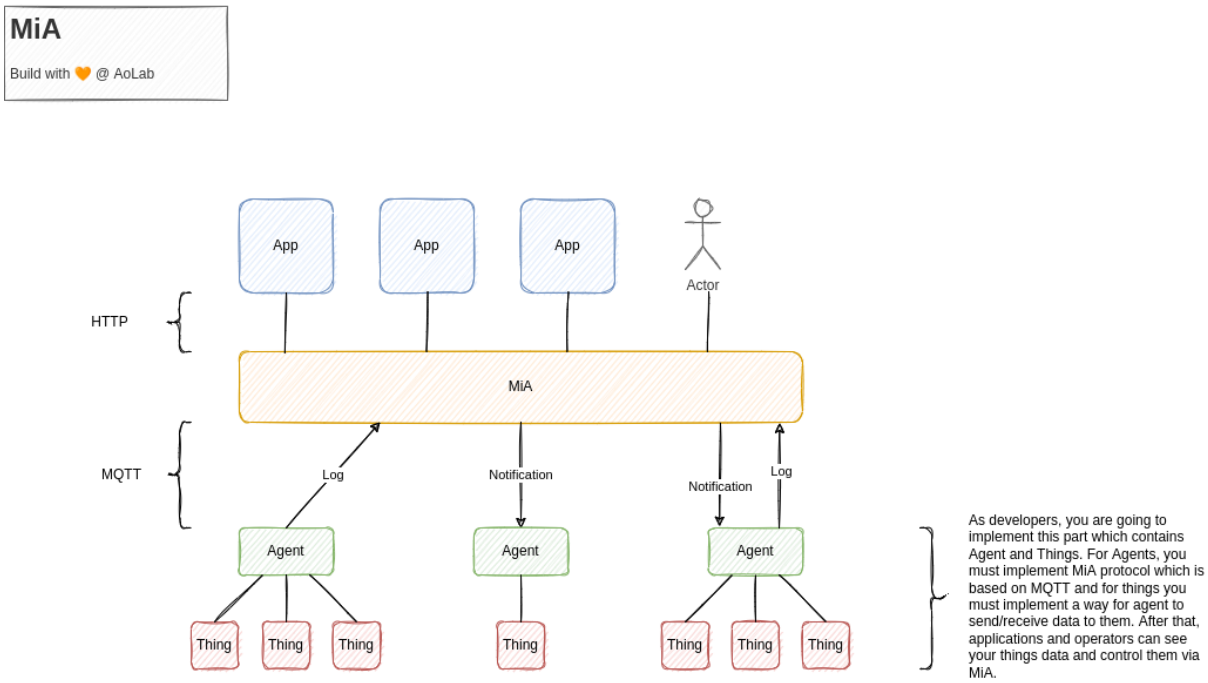
۱ مقدمه

آزمایشگاه اینترنت اشیا دانشکده مهندسی کامپیوتر در تابستان ۹۵ تصمیم گرفت تا پیاده‌سازی یک میان‌افزار برای پروژه هوشمندسازی دانشکده مهندسی کامپیوتر به صورت آزمایشی صورت دهد. قرار بود این میان‌افزار بتواند نیازمندی‌های پروژه‌ی هوشمندسازی دانشکده مهندسی کامپیوتر را برآورده کرده و پیاده‌سازی برنامه‌های کاربردی را ساده‌تر کند. با توجه به محدودیت منابع قرار بود این پیاده‌سازی منابع سخت‌افزاری زیادی را مصرف نکرده بتواند حتی روی یک Raspberry Pi نیز اجرا شود.

این میان‌افزار Middleware in Action یا اختصارا MiA نام گرفت و با توجه به سادگی زبان پایتون قرار شد از این زبان استفاده شود. این میان‌افزار دو نقش[1] دارد. نقش اول کاربرانی هستند که با استفاده از این میان‌افزار ساختمان هوشمند را نظارت کرده و پارامترهایی مانند شدت‌نور، دما و رطوبت را کنترل می‌کنند. این افراد نیاز دارند تا داده‌های حسگرها را به صورت تجمیع شده داشته باشند و از سوی دیگر بتوانند به واسطه عملگرها سیستم‌هایی مانند تهویه، سرمایش و گرمایش را کنترل کنند. نقش دوم برنامه‌نویسان هستند که قصد دارند اشیا جدیدی را به این میان‌افزار اضافه کرده یا با استفاده از رابط برنامه‌نویسی آن برنامه‌های کاربردی را توسعه دهند. نیاز این گروه وجود رابط‌های برنامه‌نویسی مناسب در سمت سخت‌افزار و نرم‌افزار است. برای ارتباط نرم‌افزاری یک پروتکل محبوب HTTP است که در MiA از آن استفاده شد و برای ارتباط سخت‌افزار نیاز به یک پروتکل دو طرفه و همزمان[2] وجود دارد که برای اجرا روی سخت‌افزارهایی با منابع محدود نباید پیچیدگی زیادی داشته باشد. در MiA از پروتکل MQTT برای ارتباط با اشیا استفاده می‌شود.

پروژه هوشمندسازی دانشکده مهندسی کامپیوتر شامل هوشمندسازی دو فضای مجزا بود. فضای اول اتاق سابق شورا دانشکده و فضای دوم آزمایشگاه اینترنت اشیا بود. این میان‌افزار با استفاده از مفهوم Tenant اجازه می‌دهد دو سرور مجزای MiA با استفاده از یک گارگزار[3] MQTT بتوانند فعالیت کنند و داده‌های اشیا را بدون تداخل جمع‌آوری کنند. ساختار طراحی شده در پروژه هوشمندسازی شامل تعدادی حسگر و عملگر بود که با استفاده از یک شبکه حسگر بی‌سیم به یک گره‌ی Raspberry Pi متصل می‌شدند. این گره اطلاعات این حسگرها و عملگرها را جمع‌آوری کرده و با توجه به توانایی اتصال به شبکه IP می‌توانست آن‌ها را به میان‌افزار انتقال دهد. با توجه به این معماری دو مفهوم Thing و Agent در MiA تعبیه شدند. Thing یا شی حسگرها و عملگرها را نمایندگی می‌کند و Agent گره‌ای با توانایی اتصال به شبکه IP را نمایندگی می‌کند. اشیا می‌توانند انواع مختلفی داشته باشند و هر یک از این انواع ویژگی‌های خاص خود را دارد. به طور مثال لامپ یک عملگر دو وضعیتی (خاموش یا روشن) است و یک حسگر دما یک حسگر است که دما را در قالب یک عدد گزارش می‌کند. در MiA هر Thing یک شناسه دارد که می‌بایست تنها میان شناسه‌های اشیا با همان نوع روی یک Agent یکتا باشد و از سوی دیگر هر Agent نیز یک شناسه یکتا دارد.

[1] Actor
[2] Realtime
[3] Broker



شکل ۱: معماری MiA در یک سیستم انتها به انتها اینترنت اشیا

از آنجایی که MiA قصد دارد حداقل منابع را مصرف کند بسیاری از اطلاعات در حافظه RAM نگهداری می‌شوند و تنها داده‌های اشیا روی پایگاه‌داده‌ای قرار می‌گیرند. به ترتیب MiA می‌تواند تنها با یک کارگزار MQTT و یک پایگاه‌داده‌ای اجرا شود. میان‌افزار MiA از یک ساختار داده‌ای و پروتکل مشخص برای ارتباط با Agentها استفاده می‌کند اما پیاده‌سازی این Agentها و چگونگی ارتباط آن‌ها با اشیا بر عهده برنامه‌نویسان سخت‌افزاری است و تنها برای سادگی کار این افراد تعدادی SDK ابتدایی توسط MiA ارائه شده است.

با توجه به قابل اطمینان نبودن شبکه میان Thingها و Agentها و از سوی دیگر امکان خرابی در Agentها هر میان‌افزاری نیاز به یک مکانیزم پیدا کردن سرویس[۴] دارد. در MiA Agentها به صورت دوره‌ای لیست اشیا خود را ارسال می‌کنند و به این ترتیب سرور آخرین وضعیت هر Agent و اشیا متصل به آن را داخل حافظه خود خواهد داشت.

۲ لینک‌های مرتبط

- کد منبع میان‌افزار MiA
- کد منبع کتابخانه ارتباطی میان‌افزار MiA به زبان پایتون

۳ بخش تئوری

۱. پروتکل‌های MQTT، CoAP و HTTP را از منظر معماری، کابردها و ...مقایسه کنید.

۲. بزرگ‌ترین مشکل CoAP و MQTT را جداگانه بررسی کرده و برای هر کدام راه‌حل پیشنهاد شده را توضیح دهید.

۳. در پیاده‌سازی لایه Application در دنیای IoT نیاز هست که دستگاه‌های انتهایی[۵] را بتوانیم به گونه‌ای شناسایی کنیم.

[4] Service Desicovery
[5] end-device

(شناسه دستگاه باید منحصر به فرد باشند) در دنیای واقعی عموما این شناسه دستگاه شامل چه اطلاعاتی هستند؟

۴. امتیازی. یکی از چالش‌های پیاده‌سازی در لایه Application، در نظر گرفتن مدل اطلاعاتی[۶] مشخص برای اطلاعات دریافتی از سنسور است. برای مثال فرض کنید سنسورهای دما، رطوبت خاک، درب و تشخیص حرکت داریم. بهتر است برای هر کدام از این سنسورها مدل اطلاعاتی جداگانه در نظر بگیریم یا اینکه یک مدل اطلاعاتی برای همه سنسورها در نظر بگیریم؟ یک روشی که برای حل این مشکل پیشنهاد شده است، استفاده از SenML هست. این روش را توضیح دهید.

۴ بخش عملی

گروه‌های سخت‌افزاری می‌بایست پیاده‌سازی یک Agent را روی بردی که در اختیار دارند، انجام بدهند. برای اینکار نیاز است که ارتباط MQTT با سرور برقرار شده و اطلاعات حسگرها در قالب Thing ارسال شده و دستورات مورد نظر عملگرها که آن‌ها نیز در قالب Thing هستند، دریافت شود. قسمت مهم دیگر در یک Agent ارسال دوره‌ای لیست اشیا متصل و وضعیت آن‌ها است. در MiA به این عملیات‌ها به ترتیب عملیات Log، Notification و Ping می‌گوییم. برای جزئیات چگونگی پیاده‌سازی این سه مورد اینجا را ببینید.
بعد از پیاده‌سازی می‌توانید برد خود را به نسخه‌ای از MiA که بالا آورده‌اید متصل کنید. برای بالا آوردن MiA اینجا را ببینید. سپس می‌توانید با استفاده از درخواست‌های HTTP اطلاعات خودتان را ارسال و دریافت کنید.

۱.۴ روشنایی هوشمند

یکی از موارد استفاده اینترنت اشیا در شهر هوشمند و بحث روشنایی هوشمند شهر است. به همین دلیل نیاز است تا در سطح شهر میزان روشنایی خیابان سنجیده شود. شما قرار است به کمک سنسور LDR این عملکرد را شبیه سازی کنید. برنامه‌ای بنویسید که میزان شدت نور LDR را در بازه بسته ۰ تا ۱۰۰ اندازه‌گیری کرده و هر ۱۰ ثانیه، آن را برای میان‌افزار MiA ارسال کند.
دقت داشته برای معرفی LDR باید از مدل مناسب (light) استفاده کنید که در این آدرس می‌توانید اطلاعات آن را مشاهده کنید.
در این تمرین می‌خواهیم کنترل چراغ‌های هوشمند شهر را شبیه‌سازی کنیم. ۴ عدد LED را به عنوان چراغ هوشمند در نظر میگیریم که هر یک را به صورت مستقل به عنوان یک Thing در نظر گرفته شده و به MiA شناسایی می‌شوند. سپس بوسیله API موجود در MiA یک درخواست به MiA برای روشن کردن یکی از LEDها ارسال می‌شود و سپس چراغ روشن می‌گردد.
دقت داشته برای معرفی LEDها باید از مدل مناسب (lamp) استفاده کنید که در این آدرس می‌توانید اطلاعات آن را مشاهده کنید.

۵ نحوه تحویل تمرین

۱. این تمرین در ۲ بخش تئوری و عملی طراحی شده است. برای بخش‌های تئوری یک فایل ارائه تهیه کرده و از روی آن پاسخ خود را در قالب یک ویدیو ضبط کنید. برای هر سؤال قسمت عملی هم یک ویدیو کوتاه حداکثر ۳ دقیقه‌ای تهیه کنید که شامل دو بخش زیر باشد.

آ) یک فیلم کوتاه از نحوه عملکرد سیستم

[۶] Data Model

ب) یک فیلم کوتاه از کد و توضیح بخش‌های مهم کد

۲. تحویل تمرین در قالب سه فایل ویدئویی انجام می‌شود، این فایل ویدویی شامل پیاده‌سازی و عملکرد Agent شما، پیاده‌سازی میزان روشنایی شهری و پیاده‌سازی کنترل چراغ‌های هوشمند است. دانشجویان شبیه‌سازی در کنار موارد ذکر شده، باید پیاده‌سازی مربوط به بهبود خواسته شده در MiA را نیز ارائه دهند.

۳. در هر ویدئو باید مشخص شده باشد که این فایل متعلق به شما است. برای مثال قبل از توضیح مراحل انجام کار، یک فایل word حاوی نام افراد گروه، شماره دانشجویی و بخش مربوطه بر روی سیستم نشان دهید که مشخص کند این ویدئو توسط شما ضبط شده است.

۴. تمرین در قالب یک فایل zip تحویل داده شود و باید برای هر مرحله، یک فایل ویدئو به همراه کد وجود داشته باشد (به جز سوالات تشریحی). در صورت عدم تحویل کد نمره‌ی بخش مربوطه به طور کامل صفر لحاظ خواهد شد. همچنین نحوه نام‌گذاری فایل zip نهایی باید به‌صورت زیر باشد:

HW1_studentNumber.zip

که در آن StudentNumber شماره دانشجویی سرگروه می‌باشد. (مثال: HW1_9631079)

۵. دقت کنید که حجم فایل zip شده نهایی، حداکثر ۳۰۰ مگابایت باشد. برای کاهش حجم ویدیوها توصیه می‌شود از نرم‌افزار ZD Soft Screen Recorder استفاده نمایید.

۶. تمامی ویدئوهای ضبط شده باید قابل پخش با آخرین نسخه نرم‌افزار KMPlayer باشد.

۷. مهلت تحویل تمرین ۷ تیر ۱۴۰۱ است. برای اطلاع از سیاست‌های تاخیر به شیوه‌نامه مراجعه نمایید.

۸. در صورت عدم رعایت موارد ذکر شده، نمره مربوط به بخش خوانایی کسر خواهد شد.

موفق و موید باشید

این سند برپایه بسته XɜPersian گونه 24.2 توسعه پیدا کرده است. نگارش شده به تاریخ ۱۴ خرداد ۱۴۰۱