

Sensorfusion i Sakernas Internet

Simon Gustafsson
dat13sgu@student.lu.se

Victor Nilsson
dat13vni@student.lu.se

Sammanfattning

Internet of Things, eller *sakernas internet* anses vara nästa stora koncept inom IT-branschen. Denna rapport utvärderar och visar konceptet kring sakernas internet med användning av Microsofts molnplattform Azure. Målet med rapporten var att bevisa vilka möjligheter systemet har kommersiellt för Combitech och andra potentiella klienter. Resultatet blev en fusion av tre olika sensorer, som kombinerat med två mikrokontroller, kommunicerar med Azure's IoT-hubb. Sensordata vidare distribuerades sedan till en molntjänst som lyssnar på hubben och sammanfogar data till en ström för vidare distribuering till en regelmotor och en visualiseringsapplikation för sensordata.

Diskussion

Utifrån de resultat som uppnåtts kunde slutsatser dras kring vilket protokoll som var lämpligast att använda vid kommunikationen mellan enhet och hubb, hur säkerheten kan förbättras och hur framtida utvecklingsmöjligheter ser ut. Under utvecklingen implementerades kommunikationskanaler över protokollen MQTT och HTTP. MQTT blev att föredra då dess publish/subscribe-mönster inte var beroende av den senaste API-versionen och behövde uppdateras med jämna mellanrum. MQTT var även mer energieffektiv än HTTP då enbart en stor uppkoppling mot server behövdes istället för flera som i HTTP.

En av de större frågorna inom IoT är säkerheten och hur system ska anpassas för att motverka tillgängligheten hos obehöriga. En utvecklare bör ta hänsyn till aspekten att den data som flödar inom ett IoT-system kan hamna i orätta händer och anpassa systemet utefter det genom att tillägga mekanismer som försvårar åtkomst i form av krypteringar och olika verifieringsmekanismer.

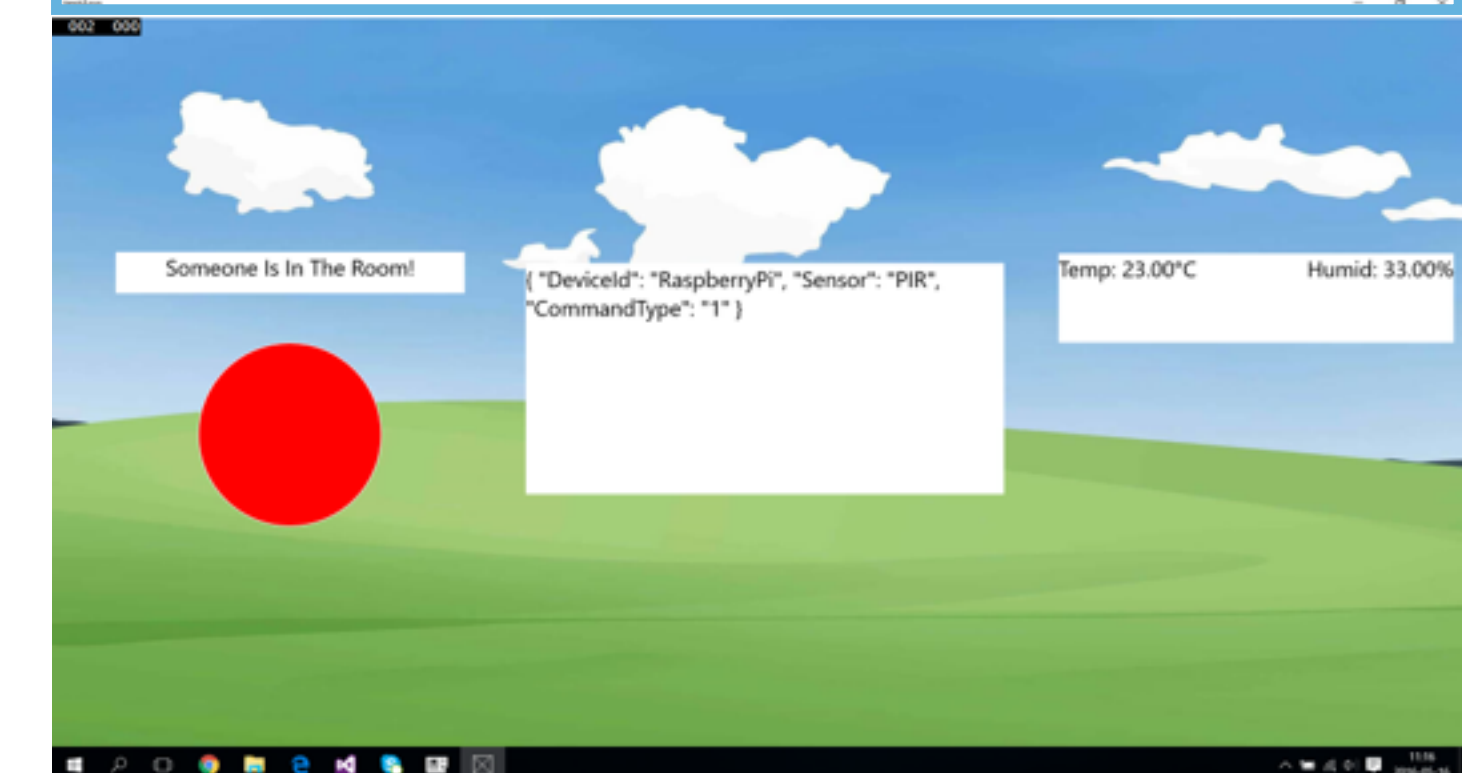
Framtida utvecklingsmöjligheter av systemet är goda och utvidgning av systemet är möjligt då stöd för anslutning av maximalt 500 enheter finns och att molntjänsten är modulärt uppbyggd. Systemet är en prototyp och kommer användas för att demonstrera konceptet både internt hos Combitech och för eventuella affärsklienter.

Problem

- Hur ska sensordata skickas från enhet till moln för snabbast möjliga leverans, utan dataförlust och med integriteten i bibehåll?
- Hur ska sensordata lagras för att försvåra åtkomsten för obehöriga användare?
- Hur ska sensordata visualiseras?
- Vilken typ av sensordata är lämplig att hämta från användaren utifrån ett integritetsperspektiv?
- Hur kan hämtad sensordata påverka individens integritet?
- Är det ekonomiskt hållbart att använda IoT inom fastighetsmarknaden utifrån aspekterna: kostnad, efterfrågan och miljö?
- Hur ska systemet konstrueras för parallell hantering av en variation av sensordata från olika mikrokontrollerenheter?
- Hur skapar man ett aggregerat meddelande utifrån flera olika sensorer samtidigt i en molnbaserad tjänst?

Lösning

Det färdiga systemet består av två mikrokontroller, en konfigurerad med en DHT-sensor och en PIR-sensor samt en konfigurerad med en Reed-Switch-sensor. Mikrokontrollerna kommunicerar via MQTT-protokollet med en IoT-hubb där sensordata avläses av en molntjänst för vidare distribution till en UWP-applikation och en regelmotor. I molntjänsten sammanfogas sensordata till en ström som via en kö distribueras till regelmotorn som, beroende på data i strömmen reagerar och skickar ut ett meddelande till UWP-applikationen.



Metod

Utvecklingen har varit uppdelad i faserna: informationsinsamling, konfiguration av sensorer, implementation av kommunikationskanaler mellan enheter och IoT-hubb, implementation av molntjänst och visualisering av sensordata. Utvecklingen har varit agil med användning av Kanbans hantering av sprintar och Extreme-programmerings parprogrammering. Den hårdvara som används är Espressifs mikrokontroller ESP8266-12E, sensorer i form av en temperatur- och luftfuktighets-mätare, en PIR-sensor och en Reed-Switch-sensor och en Raspberry-Pi för visualisering av sensordata. Implementationen av mikrokontrollerna genomfördes med hjälp av Arduino SDK och programmeringsspråket Arduino. Utvecklingen har varit testdriven där implementationer kontinuerligt testats för att säkerställa att metoderna fungerade korrekt.

