

Westerdals Oslo ACT

TEK301 – Maskin-till-maskin kommunikation

Oppgave 3 Av: Robert Mattias Molin

Prosjektuppgiften

När jag skulle bestämma mig för vilket IoT koncept jag skulle välja som faller under genren säkerhetsbranschen så kom jag att tänka på att det den senaste tiden har det varit en del inbrott i källarförråden där jag bor, så då kom jag lite in på att göra ett enkelt övervaknings system som skulle kunna hålla koll på all oönskad aktivitet.

Projektet består i dagsläget av en Particle Photon som mikrokontroller och en PIR sensor (passive infrared sensor) för att upptäcka rörelser i omgivningen. Samt en magnet sensor som kan placeras på en dörr/ingång för att registrera om dörren är i öppet eller stängt tillstånd. Lösningen har också en LED som lyser när PIR sensorn registrerar en rörelse förändring i omgivningen, Denna tog jag egentligen med i lösningen av demonstrationssyfte för video demo.

Som front-end har jag valt att använda mig utav Blynk som är en mycket användarvänlig app som ger fina visuella representationer av sensor data, och finns även tillgänglig för både Android och IOS. Azure IoT Hub som används som back-end för att lagra sensor data som skickas från projekt lösningen.

Vid ett verkligt scenario där lösningen skulle fysiskt placeras ut så skulle ett skyddande hölje (casing) behövas som skydd mot sabotage och annat som skulle kunna skada elektronikkomponenterna.

Även tillgång till internet uppkoppling är ju essentiellt, då antingen i form av en trådlös anslutning eller via en nätverkskabel. Och som skydd mot strömavbrott kan till exempel en batterihållare 4xAA, 2x2 med switch (finns i Particle dev kit) användas som backup.

Jag har valt att jobba med Particle Photon som mikrokontroller av flera anledningar. För det första så har den ett väldigt fint gränssnitt att jobba emot med bra bibliotek och en stabil cloud baserad utvecklingsmiljö.

Men den kanske viktigaste aspekten är av säkerhetsmässiga skäl. Denna informationen nedan är hämtad från Particle's dokumentation (se referens [6.](#))

- **Secure communication:** Ensuring that all communications between the device and the Particle cloud are authorized and encrypted
- **Hardware abstraction:** Providing a single, unified interface to the device, regardless of the underlying hardware architecture
- **Application enablement:** Exposing a feature-rich API that is used by developers to write applications for the device
- **Over-the-air updates:** Allowing rapid remote changes to code running on the device while providing resilience in poor connectivity environments

Notera att jag har markerat två av punkterna i orange, dessa två egenskaperna är framförallt det som gör Particle's Photon till ett förstahandsval. Just det att enheterna kan uppdateras "Over-the-air" förenklar ju också möjligheterna att utvidga projektet med ännu fler enheter som alla kan uppdateras efter behov utan att fysisk närvaro krävs. Även det att all kommunikation mellan cloud och enhet är auktoriserad och krypterad är en viktig del av valet av rätt mikrokontroller.

Vägen vidare

Möjligheterna är många, men för att vidareutveckla lösningen har jag som plan att även lägga två sensorer till, en ljud-sensor som kan ta upp ljud vid avvikande tidpunkter på dygnet. Och även en flamsensor som kan upptäcka tecken på brand.

Ja jag har faktiskt som avsikt att bygga vidare på lösningen, med dessa två komponenter som nämns ovan så tycker jag det mer börjar likna en komplett lösning.

Det skall tilläggas att jag nu använder en gratis subscription på mitt Microsoft Azure konto och därav så begränsas antalet end-points man kan ha, så i dags läget kan jag inte använda funktionerna Stream Analytics och Power BI för att visualisera min sensor data i realtid direkt på Azure IoT Hub, så detta skulle jag även vilja implementera framöver.

Sist men inte minst! För att göra lösningen mera mobil så finns även Particle's Electron som ett alternativ då det ej är möjligt med en Wi-Fi baserad lösning. Den använder ett eget SIM kort och kommunicerar över 2G/3G nätet och drivs med hjälp av ett batteri med lång levnadstid. En nackdel här kan ju vara att den inte skulle passa sig så bra i områden där de mobila nätverken har dålig täckning, något som värt att fundera över.

Instruktioner och nödvändigheter

Utvecklingsmiljö:

- Particle Build IDE, eller Particle Dev offline IDE.
- Ett Microsoft Azure konto, och en Azure IoT Hub integrering med ett message storage för lagring av sensor data som skickas från applikationen, (se referens [2.](#))

Libraries:

- <blynk.h> – Läggts till under fliken Libraries i Particle Dev miljö

Hardware:

- Particle Photon
- Breadboard
- Jumper wires
- Resistor 1stk 220 ohm
- PIR sensor
- Magnet sensor
- LED 1st

Till att börja med så krävs det att man skapar ett konto på Particle's hemsida:

<https://www.particle.io>

Med det gjort så har man fri tillgång till Particle Build som är online utvecklingsverktyget som används i denna lösning.

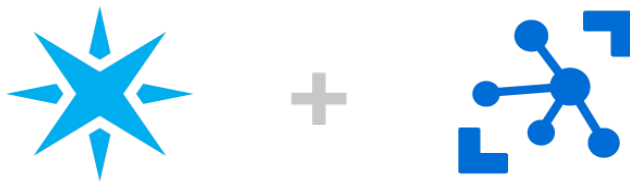


Bild tagen från: <https://docs.particle.io/tutorials/integrations/azure-iot-hub/>

När allt detta är på plats så skapar man själva integreringen genom fliken Console i Particle Dev miljö.

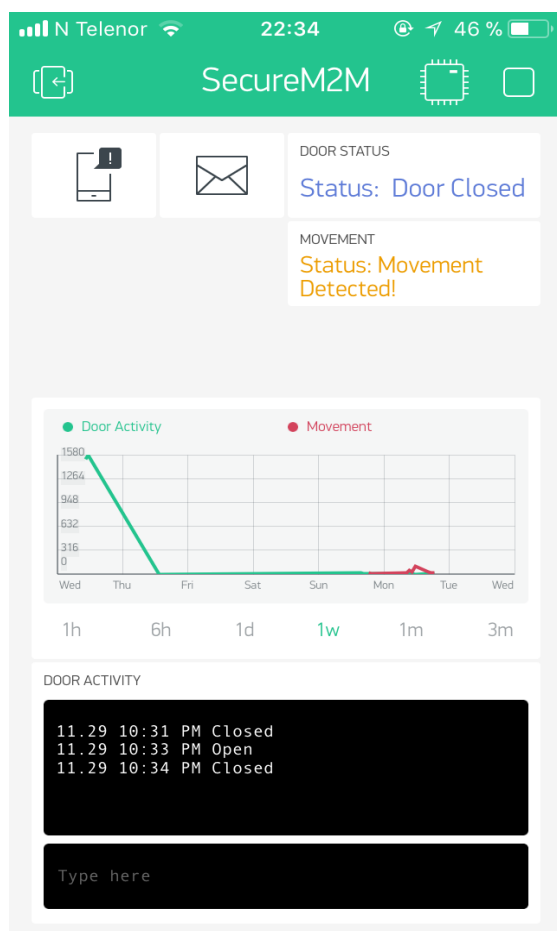
Men för att få tillgång till denna funktionaliteten så måste man först skapa ett gratis Azure konto, eller ett betal konto för "high rollers".

Allt detta kan sättas upp relativt enkelt, det är egentligen bara att följa guiden uppifrån och ned så finns allting fint förklarat på följande adress:

<https://docs.particle.io/tutorials/integrations/azure-iot-hub/>

Därefter krävs det också att man har applikationen Blynk installerad på telefonen, antingen för Android eller IOS. Denna finns tillgänglig i App Store för IOS och på Google Play för Android.

I min lösning används IOS versionen. Nedan finns en skärmbild på mitt projekt i applikationen;



Applikationen har funktionalitet för att sända notiser till ägarens telefon vid förändrat tillstånd då dörren öppnas och stängs, se Door Status.

Vid ett tillfälle då PIR sensorn upptäcker en rörelse tänds en LED och ett mail skickas till registrerad email adress i koden.

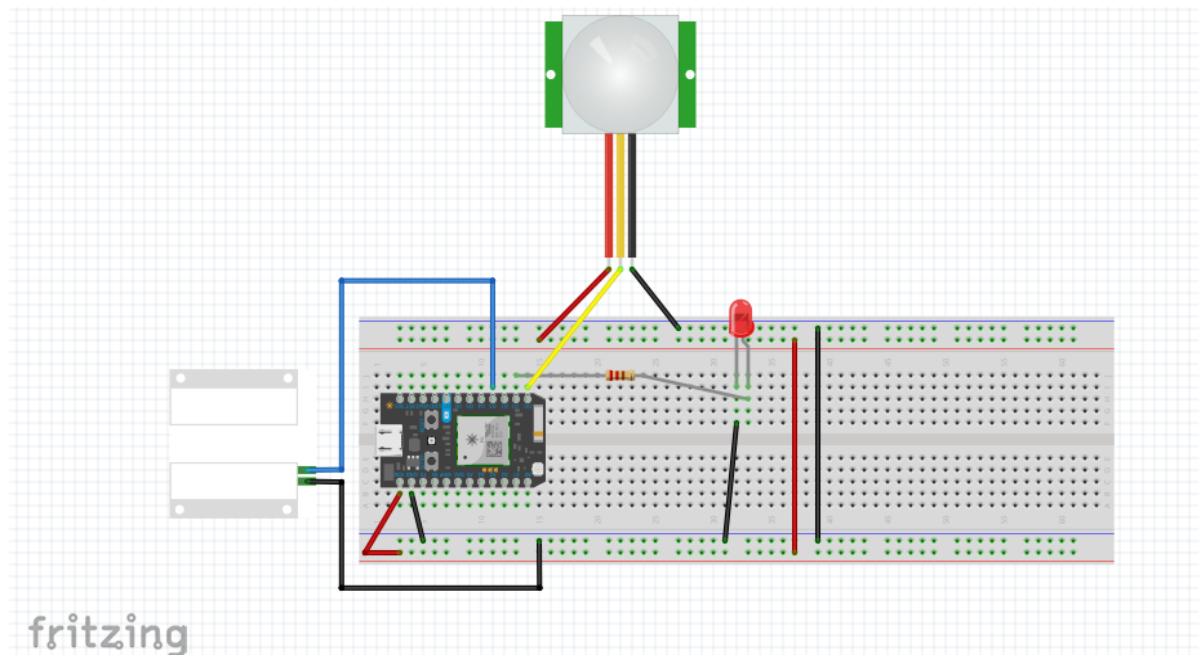
Tidsgrafnen sparar all aktivitet som sker över en vis tid, och kan ses som två linjer i grafen.

I terminalen längst ned kan all dörr aktivitet följas med datum, klockslag och status (Open eller Closed).

En QR kod är skickad till föreläsares email med instruktioner för bruk, appen kan i praktiken användas bara genom att ladda ned Blynk och använda denna kod, magi!

På nästa sida följer instruktioner för uppkoppling av fysiska komponenter.

Uppkopplings instruktioner



Till själva uppkopplingen av lösningens komponenter:

GND (-) är markerat med svart jumper wire, och **VCC (+)** är markerat med **röd** wire. Något som är värt att notera här är att **VCC** tas från **pin VIN** istället för **3V3 pin**. Detta för att få ström i närheten av **5V**.

Till att börja med, starta med att koppla magnet sensorn (komponenten längst till vänster på bilden) med en wire till GND (-), och den andra **blå** wire till **pin D3** som på bilden ovan.

Därefter koppla in PIR sensorn (den vita komponenten överst på sketchen), där **röd** wire (**VCC**) går till **(+)** på breadboard. Koppla sen **gul** wire (**OUT**) till **pin D0** på Photon enheten. Till sist, koppla svart wire till GND (-) på breadboard.

Avslutningsvis så kopplas resistorn från **pin D1** på Photon enheten till höger ben på LED'n som kan placeras på breadboard efter eget behag, sen allra sist koppla en svart wire från (GND) till vänster ben på LED'n.

Det var det färdigt! bra jobbat.

Referenslista

1. Tutorials integration med Azure IoT Hub (Online) Tillgänglig från:
<https://docs.particle.io/tutorials/integrations/azure-iot-hub/>
(2017-11-29)
2. Spara data till Azure IoT Hub (Online) Tillgänglig från:
<https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-store-data-in-azure-table-storage>
(2017-11-29)
3. Visualisera data med Stream Analytics (Online) Tillgänglig från:
<https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-live-data-visualization-in-power-bi>
(2017-11-29)
4. Particle Electron för vidare utveckling av lösning (Online) Tillgänglig från:
<https://docs.particle.io/guide/getting-started/intro/electron/>
(2017-11-29)
5. Blynk Guide (Online) Tillgänglig från:
<http://docs.blynk.cc/#hardware-set-ups-particle>
(2017-11-29)
6. Information angående firmware Particle Photon (Online) Tillgänglig från:
<https://docs.particle.io/guide/tools-and-features/system-firmware/>
(2017-11-29)
7. PIR Sensor tutorial (Online) Tillgänglig från:
<https://particle.hackster.io/ingo-lohs/mypir-sensor-informs-me-via-ifttt-e8b628>
(2017-11-29)
8. Tutorial med magnet sensorer (Online) Tillgänglig från:
<https://particle.hackster.io/BianchiRider/garage-door-opener-with-blynk-a0530d>
(2017-11-29)