



UPPSALA
UNIVERSITET

TVE-F 17 029 juni

Examensarbete 15 hp
Juni 2017

Internet of Things

Uppkopplade sensorer med Raspberry Pi

Jenny Bengtsson

Filip Nykvist

Alvin Ljung



UPPSALA
UNIVERSITET

Abstract

Internet of Things - Uppkopplade sensorer med Raspberry Pi

Jenny Bengtsson, Filip Nykvist och Alvin Ljung

**Teknisk- naturvetenskaplig fakultet
UTH-enheten**

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
<http://www.teknat.uu.se/student>

I detta projekt, som gjorts på uppdrag av IoT Sverige, har en produkt skapats åt deras Demorum för att visa hur Internet of Things fungerar och vad man kan använda det till. Rapporten beskriver hur arbetet har utformats fram till den färdiga produkten som, utifrån deras önskemål, består av en koldioxid- och en rörelsesensor kopplade till en enkortsdator. Med hjälp av kodade skript skickas den insamlade datan till en server placerad på ett så kallat moln. I grafer på en hemsida kan man därefter se datan illustrativt och det enda man då behöver är en dator med internetuppkoppling.

Handledare: Johan Fernquist och Torbjörn Fängström
Ämnesgranskare: Rikard Emanuelsson
Examinator: Martin Sjödin
ISSN: 1401-5757, TVE-F 17 029 juni

Populärvetenskaplig Sammanfattning

IoT, sakernas internet, har under ett par år varit väldigt omdiskuterat. Enkelt beskrivet är detta ett koncept som innebär att anordningar som sensorer och processorer i maskiner, fordon, o.s.v. kopplas upp mot internet. Detta gör det möjligt att övervaka olika system för att undvika skador. Ett exempel är en temperatursensor som kan hålla koll i sommarboenden för att undvika frusna rör på vintern.

Arbetet görs i samarbete med IoT Sverige, ett programkontor som arbetar för att samla aktörer, öka samverkan mellan branscher och finansiera innovationsprojekt inom området IoT. Syftet med projektet är att tillverka ett system som de kan använda i sitt "Demorum" för att visa vad IoT kan användas till. Resultatet blev två komponenter, vardera utrustade med en minidator, en rörelsesensor och en koldioxid sensor, som är utplacerade i olika rum. Dessa samlar in data som visas på en hemsida. Hemsidan är åtkomlig för alla med en dator, mobil eller annat verktyg med internetuppkoppling. Den visar grafer med insamlad data i realtid samt en rörelse över ett tidsspänn.

Denna produkt är endast till för att visa hur man enkelt kan använda IoT på ett roligt sätt, men vidare skulle den även kunna användas för annan nytta. Genom att se till att ventilationen är inställd på ett effektivt sätt skulle miljön i arbetslokaler kunna förbättras.

Innehåll

1	Introduktion	5
1.1	Bakgrund	5
1.1.1	Internet Of Things	5
1.1.2	Uppdragsgivare	5
1.2	Produktmål	6
1.3	Problembeskrivning	6
2	Teori	7
2.1	Sensor	7
2.2	Datamoln	7
2.3	Server	8
2.4	Databas	8
3	Utförande	9
3.1	Förberedelser	9
3.2	Experimentell metod	10
3.3	Tillvägagångssätt	10
3.3.1	Sensorer	10
3.3.2	Raspberry Pi	11
3.3.3	Molnlagring	11

3.4	Hemsida	11
3.5	Test av produkten	12
4	Källkritik	12
5	Resultat	12
5.1	Raspberry Pi	13
5.1.1	Hämta och skicka data	13
5.1.2	Robusthetsanpassning	14
5.2	Moln	14
5.2.1	Skicka information till servern	15
5.2.2	Hämta och visualisera datan	16
5.3	Extern klient	16
5.3.1	Hemsida	16
5.4	Exempel	17
6	Diskussion	18
6.1	Motivering av centrala beslut	18
6.1.1	Sensorerna	18
6.1.2	DigitalOcean	18
6.1.3	LAMP-stack	18
6.1.4	Extern Klient	18

6.2	Framtida Utveckling/Förbättring	19
6.2.1	Optimering	19
6.2.2	Säkerhet	19
6.3	Alternativa Metoder	20
6.3.1	Sensorer	20
6.3.2	Presentation/Representation av Datan	20
7	Slutsats	21
7.1	Måluppfyllelse	21
7.2	Användningsområden	21
8	Referenser	23

1 Introduktion

“Om vi hade datorer som visste allt som fanns att veta om saker genom att använda data som de samlat in utan hjälp från oss, skulle vi kunna spåra och beräkna allt och därmed kraftigt minska avfall, förluster och kostnader. Vi skulle veta när saker och produkter behövde bytas ut, repareras eller påminnas om ifall de är nya eller för gamla.”

- Kevin Ashton, Radio Frequency, 1999

1.1 Bakgrund

Citatet ovan kommer från en expert på digital innovation, Kevin Ashton, som förutspådde att nästa version av Internet skulle komma att handla om data som skapats av saker och inte av människor. Digital innovation, också kallat digital transformation, är studien om hur digital teknik kan förbättra funktioner och processer både på företags- och samhällsnivå. Han ska även ha myntat uttrycket Internet of Things (IoT) ¹.

1.1.1 Internet Of Things

IoT, sakernas internet, har under ett par år varit väldigt omdiskuterat. Enkelt beskrivet är detta ett koncept som innebär att anordningar som sensorer och processorer i maskiner, fordon, o.s.v. kopplas upp mot internet. I och med detta kan man skapa ett situationsanpassat beteende och smarta lösningar för samhället både i vardagen och på arbetsplatser².

1.1.2 Uppdragsgivare

Detta projekt görs i samarbete med det strategiska innovationsprogrammet IoT Sverige som är en del av Vinnova, Energimyndigheten och Formas gemensamma satsning på strategiska innovationsområden. Programmet har sedan våren 2016 valt

¹IT-kanalen, “Undersökning av Internet Of Things”

²Forbes, “A Simple Explanation Of ‘The Internet Of Things’”

att fokusera på innovativ samhällsnytta med syftet att skapa förutsättningar för internationell konkurrenskraft och hållbara lösningar på globala samhällsutmaningar. IoT Sveriges uppgift som programkontor är att samla aktörer, öka samverkan mellan branscher och finansiera innovationsprojekt inom området IoT³. Ett sätt IoT Sverige har valt för att påvisa IoTs användningsområden och hur man kan använda dessa verktyg för att generera beslutsstöd är bygget av ett s.k. demorum". Målet är att detta rum ska fyllas med flera olika produkter som kan demonstrera olika funktioner för att tillsammans ge en helhetsvision om hur ett smart samhälle skulle kunna se ut och utvecklas.

1.2 Produktmål

Efter projektperioden ska det finnas ett fungerande och pålitligt system till IoT Sverige som de kan använda i sitt demorum för att visa vad det finns för möjligheter med IoT. Den ska kunna demonstrera hur IoT kan användas för att lösa problem samt dess roll i att utveckla smarta samhällen.

Produktmålet såsom definierat av IoT Sverige är att skapa en sensorlösning efter deras behov. Lösningen ska bestå av en enkortsdator, Raspberry Pi(RPi), ansluten till en koldioxid- och en rörelsesensor som ska kunna placeras ut för att mäta dessa värden. Denna data ska sedan på lämpligt sätt kunna tas del av från annan plats. Målet är att ha ett fungerande system där det oberoende av placering går att se datan förändras i realtid samt nå data från specifika tidsintervall.

1.3 Problembeskrivning

Problemformuleringen för projektet är hur, och med vilka verktyg och programvaror, ska mätdata från sensorerna samlas in, lagras, skicka och presenteras. Utifrån önskemål från IoT Sverige skulle lösningen baseras på RPi och en molnlösning. Därför var de centrala problemen:

- Hur ska RPi:n hämta informationen från sensorerna?
- Hur ska data från RPi:n föras över till molnet och hur ska detta ske automatiskt?

³IoT Sverige, "Om IoT Sverige"

- Hur ska data tas emot på molnet?
- Hur ska data hanteras på molnet?
- Hur ska data presenteras?

2 Teori

I följande stycke presenteras en generell teoretisk bakgrund för de huvudsakliga moment lösningen bygger på.

2.1 Sensor

En sensor är en anordning som används för att mäta olika parametrar som till exempel rörelse, koldioxidhalt, temperatur eller tryck. Olika sensorer mäter sina givna parametrar på olika sätt. Ett vanligt sätt att mäta rörelse är genom infraröd strålning, så kallade PIR-sensorer. De mäter förändringar i temperaturen som ändras i och med att vi avger värme genom infraröd strålning⁴. Precis som med rörelse finns det många sätt att mäta koldioxidhalten i luften och även där kan infraröd strålning användas.

2.2 Datamoln

I och med den tekniska utvecklingen som lett mot den ständigt uppkopplade människan så har det så kallade molnet blivit en viktig del inom tekniken. Molnet är ett externt serverutrymme för datortjänster, som till exempel program, processortid, applikationer, e-post och lagring av datafiler, som är tillgängliga via internet från enskilda datorer. Idag har det blivit vanligt att programvaror, hårddiskar samt servrar flyttas ut från användares personliga datorer eller mobiler till molnens servrar. En fördel med detta är att en datoranvändare får tillgång till alla sina sparade filer och program från vilken extern dator som helst så länge det finns internetuppkoppling. Dessutom är programmen och filerna sparade även om den personliga datorn skulle råka ut för till exempel skada eller stöld. Molnet och olika molntjänster är intressant

⁴Bilddr, "Did It Move? Detecting Motion With PIR + Arduino"

för många företag. Det är nu för tiden vanligt att många företag lägger komponenter från sina IT-avdelningar på molnet. Dessutom innebär det stor flexibilitet i och med att lagringsutrymme och datorprogram kan anpassas efter företagets behov som i perioder kan vara skiftande⁵.

Det finns många olika företag som erbjuder molnlagring och andra molntjänster. Flera, till exempel DigitalOcean, Jelastic och Vultr vänder sig lite mer till specifika användare som program- och datautvecklare och har tjänster som också inkluderer processortid. Andra molntjänster är mer anpassade för privatpersoners vardagliga fillagring som till exempel iCloud, OneDrive och Dropbox. Det som varierar mellan olika företags tjänster kan vara priser, uppladdningstid och lagringsutrymme.

2.3 Server

En server, även kallad värddator, är ett datorsystem som över till exempel ett datornätverk betjänar andra system (klienter). Server kan också, beroende på sammanhang, syfta på en fysisk dator eller en viss programvara den kör. En variant av server är en webbserver som lagrar information som är tillgänglig genom webben och levererar den till besökare. Termen används både om servern (datorn) som används och om det program som sköter materialet och kommunikationen. Programmen ser till att kommunikationen med webbsidorna hanteras rätt så att ens förfrågningar tas emot, behandlas och kan köras på de flesta operativsystem(OS). Ett OS är ett lager av mjukvara på en dator som agerar som grund för datorns program. Det är kärnan i OS:et som har till uppgift att övervaka datorns resurser och fördela de resurserna till program som behöver dem, t.ex. minne och processorkraft. Det finns flera olika OS. Några av de vanligaste är Microsoft, Mac OS X och Linux/GNU⁶.

2.4 Databas

En databas är en mängd data samlad och organiserad i dataregister. Det finns olika slags databaser för olika syften, till exempel relationsdatabaser där man lagrar data i olika tabeller, även kallat relationer. DB2, Oracle och MySQL är exempel på några relationsdatabaser. Gemensamt för dessa är att de använder programspråket SQL för att hantera datan.

⁵Nationalencyklopedin, "Molnet"

⁶HowStuffWorks, "What's Ubuntu, and How is it Different From Linux?"

3 Utförande

Nedan, i figur 1, illustreras de komponenter och tankegångar som var utgångspunkten vid starten av projektet.



Figur 1: En illustration av starten av projektet. Till vänster och höger är de komponenter som var utgångspunkten för projektet och mitten illustrerar de tankegångar som förekom för hur problembeskrivningen skulle lösas. Orden med större text representerar några av de komponenter som var mer kritiska för projektets framgång

3.1 Förberedelser

För att kunna utföra projektet krävdes en del förkunskap gällande programmering samt förståelse för hur Internet fungerar och är strukturerat. I Civilingenjörsprogrammet i Teknisk Fysik ingår bland annat kursen *Programmeringsteknik I, 1TD433* som ger grundkunskaper i programmeringsspråket Java. Det är ett väldigt grundläggande språk som gjort det enkelt att även kunna förstå och hantera de andra språken som använts för detta projekt. Utöver Programmeringsteknik I har även kunskaperna kring programmering utvecklats i kurserna *Beräkningsvetenskap I, II samt III* som även dessa ingått i programmet. För mer grundläggande kunskap om hur Internet och IoT fungerar så lästes även kursen *Internet of Things, 1DT094* som ges på Uppsala Universitet.

Mycket tid lades i början av projektet på att läsa in området och studera hur liknande projekt utformats för att få en idé om vilket tillvägagångssätt som skulle användas.

3.2 Experimentell metod

En experimentell metod med fokus på utveckling har använts under hela projektets gång för att nå de produktspecifika målen. En sådan metod var passande för projektet delvis för att det har använts mycket programmering i olika program som på bättre och sämre sätt interagerar med eller påverkar varandra. Därav behövde de testas med varandra och justeras vid eventuella konflikter i programmen.

Genom att ställa frågor utformade efter problemet har olika idéer utformats och därefter testats för att få fram den bästa metoden.

3.3 Tillvägagångssätt

Skript utvecklades^{7,8} för att ta emot data från sensorerna och sedan skicka dem vidare till molnet där en webserver sattes upp med en databas för att lagra informationen. Molnet var kopplat till en IP-adress vilket gjorde att en enkel hemsida kunde skapas för att presentera data i grafer och tabeller. En del löste sig direkt med den valda metoden medan andra lösningsmetoder fick testas och den bäst lämpade valdes. Nedan presenteras komponenterna som användes i projektet.

3.3.1 Sensorer

För att mäta rörelse användes en PIR-sensor. PIR-sensorn i projektet är från Kjell & Company och har förmåga att mäta rörelse upp till cirka sju meter bort med 100° vinkel. Den har justerbar fördröjning samt känslighet och drivs av en spänningskälla på minst 5,0 volt⁹. Syftet med sensorn var primärt är att identifiera om och inte hur många som vistas i rummet.

Koldioxidhalten mättes med en K30-sensor från SenseAir. Den mäter mängden koldioxid i luften genom SenseAirs unika metod som bygger på principen av infraröd absorption av strålning som kallas icke-dispersiv infraröd teknik. Sensorn har en kapacitet att mäta från 0 – 5000 ppm_{vol} med en noggrannhet på ± 30 ppm och drivs

⁷CO2 Meter, “Application Note AN137: Raspberry Pi UART Interface to K-30 CO2 Sensor”

⁸Raspberry Pi Learning Resources, “Parent Detector”

⁹Kjell & Company, “Luxorparts Rörelsedetektor för Arduino”

av en spänningskälla på minst 4,5 V ¹⁰.

3.3.2 Raspberry Pi

På RPi:n placerades de skript som samlar in data. Även skript som kontaktar databasen och transporterar data dit lades där.

3.3.3 Molnlagring

För att kunna nå informationen oberoende av placering så blev molnlagring en central del i projektet. För detta projekt användes DigitalOcean. Det är en leverantör som erbjuder molnlagring för speciellt data- och programutvecklare. En privat virtuell maskin sattes upp, eller en *droplet* som det kallas hos DigitalOcean¹¹, där fanns möjligheten att lagra och hantera allt från olika filer till uppbyggda databaser. På molnet installerades en LAMP-konfiguration. LAMP är en akronym för Linux, Apache, MySQL och PHP. Det är en open source plattform för webbutveckling som använder ett OS med Linuxkärna (här användes en distribution av Linux/GNU vid namn Ubuntu, version 16.04), Apache2 som webbserver, MySQL 5.7.17 som databas samt PHP 7.0.15 som det objektorienterade skriptspråket. Skripten på molnet användes för att ta emot och lägga till data till databasen samt för att hantera visualiseringen av data. Ett skydd sattes även upp på dropleten för att förhindra att utomstående kunde komma åt privat information lagrad där.

3.4 Hemsida

För att nå data och se den oberoende av placering så lades informationen ut på en hemsida där datan presenteras i grafer skapade med hjälp av Google Visualisation API.

¹⁰SenseAir, "Product Specification"

¹¹ DigitalOcean, DigitalOcean"

3.5 Test av produkten

Under större delen av tiden, så fort sensorerna var uppkopplade, har produkten varit igång och testats i olika miljöer. Till en början skedde det i demorummet. När produkten var utvecklad i första steget, för att kunna göra allt det som förväntades, så placerades produkten ut på olika platser på universitetet under olika långa perioder. Testen gjordes för att identifiera problem som dataförlust och uppkopplingsproblem.

4 Källkritik

Under projektets gång har ett flertal källor använts för bakgrundsinformation och specifikationer för produkter och programvaror. Dessa källor anses vara pålitliga och inga slutsatser har dragits utifrån information hämtad från dessa. Därför är inga av källorna kritiskt innehållsgranskade.

5 Resultat

Den slutgiltiga produkten inkluderar:

- Två RPi enheter med:
 - Rörelse-och koldioxidsensor
 - Robustanpassat Pythonskript för insamling och sändning av mätdata
 - Pythonskript för identifiering- och sändning av IP-adressen via SMTP
 - SSH för fjärranslutning
- Server, hemsida och databas på molnlagringstjänsten DigitalOcean
 - PHP-skript för mottagning och lagring i databasen
 - MySQL-databas
 - PHP-skript för hämtning ur databasen
 - Javascript för visualisering av den hämtade datan
 - HTML/CSS-kod för hemsidan

Händelsekedjan har delats in i tre huvudsakliga steg: RPi, molnlagring och externa klienter. Till de två RPi enheterna kopplades de två sensorerna vars data skickas till molnet. På molnet hanteras datan och informationen presenteras på en webbsida. Webbsidan kan nås från alla externa klienter med en internetuppkoppling och presenterar informationen i grafer och tabeller. RPi enheterna och molnet inkluderar flera komponenter som diskuteras i djupare detalj under respektive rubrik nedan och presentationen av data under externa klienter. I figur 2 visas en övergripande bild över komponenterna som produkten består av.



Figur 2: Övergripande bild över komponenterna för produkten

5.1 Raspberry Pi

Insamling, behandling och skickande av mätdata hanteras av RPi-enheterna och beskrivs i rubrikerna nedan.

5.1.1 Hämta och skicka data

RPi enheten kör kontinuerligt ett Pythonskript som initieras vid uppstart och hämtar mätvärden från sensorerna. Rörelsevärdet är en kumulativ summa av antalet rörelser som registrerats under en period på 5 minuter. Tillsammans skickas då den momentana koldioxidhalten, i parts per million (ppm), och tiden då mätningen gjordes. Dessa mätvärden konverteras först till ett format som databasen kan hantera och skickas därefter iväg. Mätvärdena skickas till dropletens IP-adress som hanteras

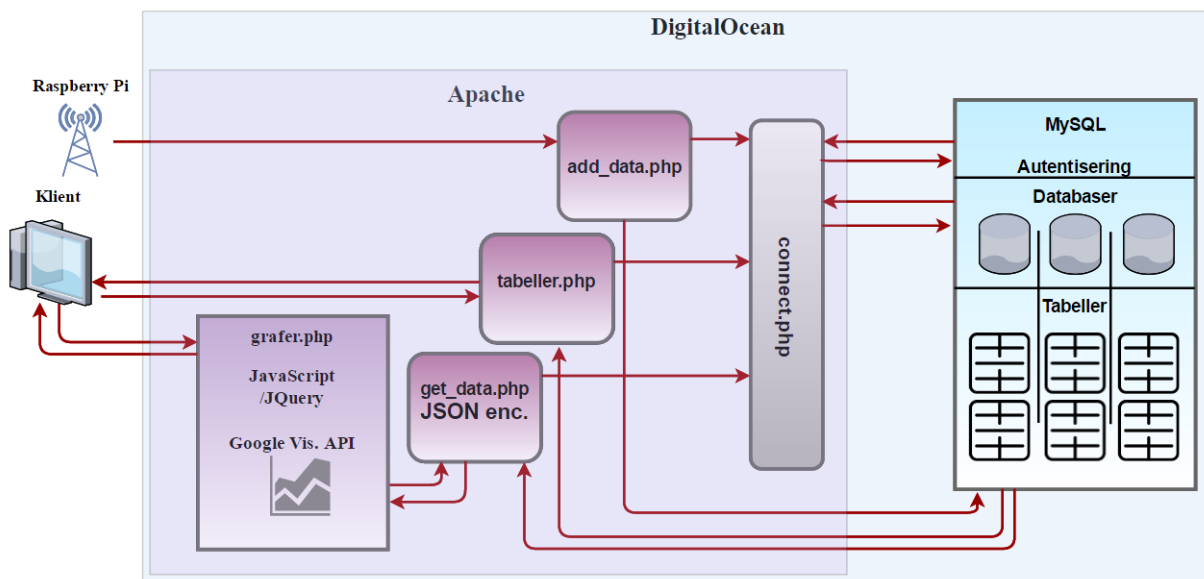
av ett PHP-skript som tar emot en enligt HTTP-protokollet definierad *GET-metod* från Python-skriptet.

5.1.2 Robusthetsanpassning

Pythonskriptet som samlar in och skickar data är anpassat så att eventuella problem med internet och anslutning inte skall påverka de mätvärden som presenteras på hemsidan. Tillsammans med mätvärden sparas även tiden för insamling. Vid eventuella anslutningsproblem sparas alla mätvärden tillsammans med tiden och skickas när anslutningen är tillbaka. Skriptet kontrollerar också att samma data inte skickas flera gånger. Skriptet initieras vid upp- eller omstart och alla mätvärden som sparats då de ej är skickade blir de första som skickas medan insamling av ny data börjar.

5.2 Moln

Den installerade LAMP-konfigurationen, likt figur 3, ger möjligheten att samla in informationen från vår RPi och spara denna i SQL-databasen med hjälp av PHP. Apache gav sedan möjlighet att nå denna information genom HTTP-protokollet.



Figur 3: Översiktsbild för serversidan

5.2.1 Skicka information till servern

När RPi skickar sin HTTP-förfrågan att lägga in data i databasen hanteras denna transaktion med CGI-skriptet `add_data.php`.

`add_data.php` ansluter i sin tur till `connect.php` som har som uppgift att autentisera åtkomst till samlingen med databaser, samt att välja en specifik databas som skall användas. När autentiseringen har skett väljer `add_data.php` tabell att lägga in den skickade datan i. `add_data.php` förväntar sig specifika variabler från den skickade datan, samt förväntar sig också att dessa datatyper korresponderar med datatyperna för fälten i de valde tabellerna. Den data skriptet är konstruerad för att hantera är den klientbaserade tiden i UNIX-tid vid vilken datan samlades, CO_2 mätvärdet samt rörelsesensornas värde. MySQL hanterar dock tidsfälten med DATETIME variabeln då denna underlättar val av specifik data som ska hämtas eller läggas till. Tidsvariabeln konverteras därför i `add_data.php` skriptet till ett DATETIME format.

5.2.2 Hämta och visualisera datan

När en klient ber om graferna skickas den till *grafer.php*. Den använder sig av *get_data.php* skript som på motsatt sätt till *add_data.php* använder sig av *connect.php* och sedan istället för att sätta in specifik information i en viss tabell istället extraherar specifik information från en viss tabell. Denna data ska sedan visualiseras med hjälp av integrerad JavaScript kod som använder JQuery API för att hämta data från *add_data.php* filerna. API står för *Application Programming Interface* och är ett gränssnitt som specificerar hur applikationsprogram kommunicerar med programvara. JavaScript koderna har valts att behandla en JSON-kodning av datan. JSON, *JavaScript Object Notation*, är ett textbaserat format som är lätt att hantera i JavaScript. Denna JSON-kodning är integrerad i *get_data.php* filerna och levereras alltså i korrekt format till JavaScript koden.

Liknande, när en klient ber om *tabeller.php*, går den via *connect.php* för autentisering och val av databas. *tabeller.php* väljer sedan en tabell den vill hämta specifik data från och visar denna datan i en HTML/CSS baserad tabell. Här sker alltså ingen ändring till JSON format av datan mellan, då *tabeller.php* ej använder JavaScript.

JavaScript koden använder sig sedan av Googles Visualiserings-API för att underlätta en visualisering av datan.

5.3 Extern klient

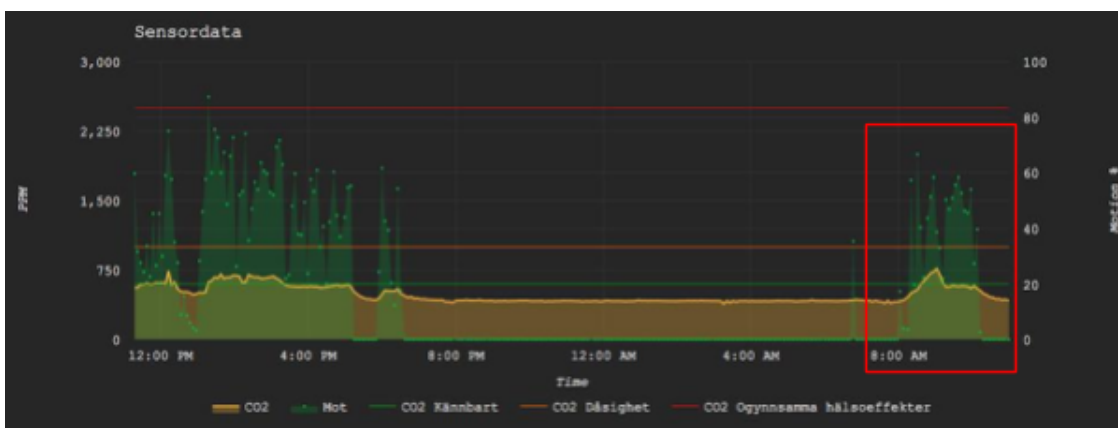
När en extern klient ansluter till hemsidan hämtar den hem den kod som är skriven i HTML, CSS och JavaScript vilken sedan kompileras i klienten. I detta skiljer den sig från CGI-skripten såsom PHP, då denna verkar på serversidan.

5.3.1 Hemsida

Mätdata som kontinuerligt samlas in visualiseras på en webbsida som går att nås för vem som helst med internetuppkoppling. Mätdata presenteras i tabeller, två interaktiva grafer och en mätare där signifikanta CO_2 -nivåer är utmarkerade.

5.4 Exempel

Mycket testning har gjorts under projektets gång, men när alla komponenter bedömdes väl utvecklade så kördes ett sista test. Produkten placerades ut i ett klassrum på Ångströmlaboratoriet under ett dygn. Syftet var att se att data samlades in och skickades som önskat och att det gick att ta del av och följa datan från hemsidan genom tabellerna och graferna. Resultatet från testet visas i figur 4 nedan.



Figur 4: Momentanbild från hemsidan efter att produkten testats under ett dygn. Den röda rutan markerar morgonen där koldioxiden tydligt förändras då rörelse detekteras

Från figur 4 ses tydligt att koldioxidhalten förändras vid rörelse. På Ångströmlaboratoriet är lektionerna vanligtvis två timmar långa med en paus efter halva tiden. I det markerade området på grafen syns att lektionen har startat vid 8.15 och därefter rör sig koldioxidhalten uppåt under en timme. Därefter går den neråt, vilket är då de har paus. Sedan håller den sig stabil 45 minuter vilket kan bero på att de har valt att ha dörren öppen eller att en del studenter valt att avvika under andra timmen av lektionen. Därefter sjunker halten ytterligare något och rör sig ner mot den tidigare stabila nivån som var under natten.

6 Diskussion

6.1 Motivering av centrala beslut

Under projektets gång behövde flera gånger viktiga beslut tas. De mest centrala besluten diskuteras nedan.

6.1.1 Sensorerna

Koldioxidensorn som använts är sponsrad av SenseAir. PIR-sensorn valdes då känslighetsområdet passar syftet och för dess låga pris. Dessutom är PIR-sensorn flitigt använd av andra och många exempellösningar till kod finns tillgängliga på diverse forum som kunde modifieras för projektets syfte.

6.1.2 DigitalOcean

DigitalOcean valdes som service för datamolnstjänster främst då det fanns erfarenhet av denna på kontoret eftersom den redan används av IoT Sverige. Den uppfyllde önskemål i fråga om komponenter och kapacitet samt bra stöd för det önskade OS.

6.1.3 LAMP-stack

En LAMP-stack valdes, detta då det är en välkänd konfiguration för liknande syften vilket innebär att det finns bra stöd i form av foruminlägg vid eventuella problem. De olika komponenterna som användes i LAMP valdes då de alla är välkända och kompatibla med varandra. För de flesta installerades den senaste uppdateringen då det är den mest stabila och välutvecklade.

6.1.4 Extern Klient

En hemsida är passande för att visualisera samt kommunicera informationen då det är ett redskap tillgängligt för alla. En variation av visualiseringstyper som tabeller,

grafer och mätare användes för att ge en övergripande blick i ett tidsperspektiv samt för att informera om vad informationen betyder genom olika färger och markerade områden. Google Charts användes för att illustrera dessa grafer eftersom verktyget var lätt att använda ihop med den uppbyggda databasen.

6.2 Framtida Utveckling/Förbättring

För de flesta lösningar finns utvecklingsmöjligheter som rör bland annat funktion och utseende, men den tidsbegränsade perioden gör att det är svårt att hinna med alla förbättringar. Nedan beskrivs en del utvecklingar som skulle kunna förbättra produkten och dess utformning.

6.2.1 Optimering

Skripten som utformats skulle möjligtvis kunna optimeras för att minska arbetsbelastningen för servern och klienter. Vissa kodsegment kan ta längre tid att exekvera och då kan det vara fördelaktigt att skriva om dem så att de inte kräver lika mycket arbete att utföra. Det har även funnits en del tankar kring att lägga vissa delar i egna skript eftersom de inte används hela tiden.

6.2.2 Säkerhet

Under första veckan utsattes dropleten för ett dataintrång där webbservern användes i en DOS-attack. En DOS-attack, Denial Of Service, går ut på att skicka så mycket data till en server att den inte kan hantera alla förfrågningar. Som resultat till den ökade nätverksanvändningen från partitionen stängde DigitalOceans interna säkerhetsansvariga ner trafiken och dropleten behövde förstöras. Säkerhet är centralt när det kommer till internettjänster och säkerheten har förbättrats sen attacken men det finns alltid vidare anpassningar som kan göras. På DigitalOceans hemsida finns många länkar till hur dropletens säkerhet skulle kunna förstärkas.

6.3 Alternativa Metoder

Många alternativa lösningar har stötts på under projektets gång som har varit mer eller mindre välanpassade för projektet. Hade tidsbilden varit annorlunda finns det dock en del lösningar som kunde ha varit att föredra.

6.3.1 Sensorer

För ännu mer specifik information gällande rörelse skulle man kunna införskaffa en mer avancerad sensor som kan mer information än om det bara är rörelse. Med hjälp av detektorer skulle det gå att identifiera antalet personer i ett klassrum istället för att bara meddela att det är någon slags rörelse.

6.3.2 Presentation/Representation av Datan

Det främsta alternativet till nuvarande metod skulle vara att använda tjänster såsom Kibana och Grafana för visualiseringen av den lagrade datan. De är anpassningsbara visualiserare där man lätt och interaktivt kan skapa och omplacera grafer med ett användarvänligt GUI, Graphical User Interface. För att kunna implementera dessa istället för Google-Graphs skulle en omstrukturering av databasen krävas då dessa tjänster baseras på att variabeln tid hanteras unikt i databasen. Alternativ till MySQL för att kunna använda dessa applikationer skulle vara databaser såsom Influx DB eller Graphite.

Med denna förändring skulle visualiseringen av datan både bli mer estetiskt tilltalande, lättare att använda, men framförallt lättare att förändra vilken och hur datan presenteras interaktivt och i realtid.

7 Slutsats

7.1 Måluppfyllelse

Målet med projektet som formulerats av IoT Sverige var att i slutet av projekttiden ha ett fungerande och pålitligt system som demonstrerar möjligheten med IoT lösningar i form av en sensorenhet vars information kan när varifrån som helst.

Resultatet av projektet uppfyller alla mål. Sensordata samlas in och skickas på ett pålitligt och kontinuerligt sätt utan externt ingripande. Informationen lagras, transporteras och visualiseras på ett sådant sätt att den går att nås på annan plats och är endast beroende av internetuppkoppling. Därmed har IoT Sverige en fungerande produkt att använda i sitt demorum. Den demonstrerar väl hur IoT-lösningar kan användas för att bygga ett interaktivt digitalt samhälle samt förse användare med underlag för beslut. I kombination med andra IoT-lösningar skapar den en helhetsbild för hur dessa lösningar skulle kunna användas och implementeras. IoT Sverige tycker att den fyller dessa mål väl i sin slutliga produktform och har låtit den vara en demonstration för IoT-lösningar och IoT Sveriges visioner även utanför demorummet.

7.2 Användningsområden

Syftet med det här projektet har varit att få produkten att fungera och kunna leverera ett visuellt resultat, men det finns även användningsområden för produkten som skulle göra nytta. Området för uppkopplade enheter växer och möjligheten att ta del av information av olika slag är snarare regel än undantag. För just denna produkt ser vi tre möjliga sätt att använda produkten för någon slags nytta.

- Genom att registrera den aktuella koldioxidhalten kan ventilationen enkelt justeras för att optimera arbetsmiljön i ett rum.
- Tack vare att den historiska datan för koldioxidhalten går att följa så kan man energieffektivisera genom att anpassa ventilationen genom att t.ex. stänga av eller sänka den när den inte behöver användas.
- Dessa ovanstående punkter kan tillsammans utvecklas vidare för att skapa så kallade smarta rum där ventilationen kan anpassas till tidigare data och själv

styras för att uppnå den mest optimerade och effektiva ventilationen.

Produkten kan lämpligtvis placeras ut på skolor, kontor och andra allmänna platser där luften kan spela stor roll för arbetsinsatserna.

8 Referenser

Bil提高. "Did It Move? Detecting Motion With PIR + Arduino." Senast ändrad 2011. Hämtad 24 mars 2017. http://bildr.org/2011/06/pir_arduino/

CO2Meter. ApplicationNoteAN137: RaspberryPiUARTInterfacetoK-30CO2Sensor. Hämtad 2 April 2017. <http://www.co2meters.com/Documentation/AppNotes/AN137-K30-sensor-raspberry-pi-uart.pdf>

Digital Ocean. Digital Ocean.Senast ändrad 2017. Hämtad 24 mars 2017. <https://www.digitalocean.com>

Forbes. "A Simple Explanation Of The Internet Of Things." Senast ändrad 2014. Hämtad 24 mars 2017. <https://www.forbes.com/sites/jacobmorgan/2014/05/13/simple-explanation-internet-things-that-anyone-can-understand/#78d42e6a1d09>

HowStuffWorks. "What's Ubuntu, and How is it Different From Linux?" Senast ändrad 2011. Hämtad 3 mars 2017. <http://computer.howstuffworks.com/ubuntu.htm>

IoT Sverige. "Om IoT Sverige." Senast ändrad 2017. Hämtad 24 mars 2017. <https://iotsverige.se/om-iot-sverige/>

IT-kanalen. "Undersökning av Internet Of Things." Senast ändrad 2015. Hämtad 23 mars 2017. <http://it-kanalen.se/undersokning-av-internet-of-things>.

Kjell Company. "Luxorparts Rörelsedetektor för Arduino". Hämtad 10 April 2017. https://www.kjell.com/se/sortiment/el-verktyg/elektronik/arduino/moduler/luxorparts-rorelsedetektor-for-arduino-p87892?gelid=CjwKEAjw2qzHBRChloWxgoXDpyASJAB-01Io0IPzpgBnkZrqOV1GAuvPNI_By1oDfdkVZy6AV3NQv6hoCkt7w_wcB

Myndigheten för samhällsskydd och beredskap. "Planering för ett hållbart och robust samhälle." Senast ändrad 2009. Hämtad 28 mars 2017. <https://www.msb.se/sv/Forebyggande/Samhallsplanering/>.

Nationalencyklopedin. "Molnet." Hämtad 11 april 2017. <http://www.ne.se/uppslagsverk/encyklopedi/lang/molnet>

Raspberry Pi learning Resources. "Parent Detector". Hämtad 30 Mars 2017. <https://www.raspberrypi.org/learning/parent-detector/worksheet/>

SenseAir, "Product Specification". Senast ändrad 2015. Hämtad 24 mars 2017. <http://www.senseair.com>

[content/uploads/2015/03/CO2-Engine-K30_PSP110-R7.pdf](#)