

Tallinna Ülikool
Digitehnoloogiaste Instituut

Arduino ühendamine Microsoft IoT-ga

Bakalaureusetöö

Autor: Martin Kütt

Juhendaja: Jaagup Kippar

Autor: „2017

Juhendaja: „2017

Instituudi direktor: „2017

Tallinn 2017

Autorideklaratsioon

Deklareerin, et käesolev seminaritöö on minu töö tulemus ja seda pole kellegi teise poolt varemkaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, seisukohad, kirjandusallikatest ning mujalt pärinevad andmed on viidatud.

.....(kuupäev)(autor)

Lihlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina _____ (sünnikuupäev: _____)

(autori nimi)

1. annan Tallinna Ülikoolile tasuta loa (lihlitsentsi) enda loodud teose

(lõputöö pealkiri)

mille juhendaja on _____,

(juhendaja nimi)

säilitamiseks ja üldsusele kättesaadavaks tegemiseks Tallinna Ülikooli Akadeemilise Raamatukogu repositooriumis.

2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tallinnas/Haapsalus/Rakveres/Helsingis, _____

(digitaalne) allkiri ja kuupäev

Sisukord

Sissejuhatus	5
1. Valdkonna üldtutvustus.....	6
1.1 Arduino	6
1.2 Internet of Things (asjade internet).....	7
1.3 Microsoft Azure	8
2. Ülevaade olemasolevast materjalist	9
3. Omapoolsed katsetused, lõplik lahendus, kogemused ja analüüs tulemustest.....	10
3.1 Katsetused ja tekkinud probleemid.....	10
3.2 Lõplik lahendus.....	12
3.2.1 Arduino.....	12
3.2.2 Python.....	13
3.2.3 Microsoft Azure.....	13
3.2.4 Pythoni sidumine Microsoft Azure'iga	15
3.2.5 Power BI.....	19
3.3 Kogemused	22
3.4 Analüüs tulemustest.....	23
Kokkuvõte	24
Summary.....	25
Kasutatud allikad	26
Lisad	31
Lisa 1. Arduino kood.....	31
Lisa 2. Pythoni kood.....	32

Sissejuhatus

Tänapäeval oleme juba üsnagi harjunud sellega, et väga paljud asjad meie ümber on ühte või teist moodi internetiga ühenduses. Enamustel meist on olemas nutifonid mida me igapäevaselt kasutame. Samuti kasutab enamus meist igapäevaselt arvuteid. Igapäevaseid seadmeid on internetiga ühendatud leidub aga veel. On olemas külmkapid mis on võimalik internetti ühendada ja millel on ükste peal displeid, uuemad autod on konstantselt ühenduses GPS'iga (ülemaailmse asukoha määramisega) ning mida on võimalik ka internetti ühendada, sellised autod on võimelised juba ise peale õnnetust abi kutsuma. Samuti edastatakse ka telepilti praegu üle internetikaabli mis jookseb digiboksi, uuematel telekatel ka otse telekasse. Selliseid seadmeid tekib aina juurde. Kõik need seadmed tervikuna on asjade internet.

Käesoleva bakalaureusetöö autori (edaspidi tekstis autor) huvi antud teema vastu tekkis siis kui autor käis robotika aines. Huvi hakkas just pakkuma see kuidas luua mingi seade ja kuidas seda arvutist juhtida saab. Autor ei olnud aga kindel mis teemat ta täpsemalt käsitleda tahaks. Selle probleemiga pöördus ta oma tulevase juhendaja, Jaagup Kippari poole, kes soovitas just seda teemat.

Käesoleva bakalaureusetöö peamiseks eesmärgiks on leida võimalus kuidas ühendada Arduino Microsofti asjade interneti platvormiga Azure. Autor otsib selleks internetist materjale ning analüüsib kas nende materjalide põhjal on võimalik koostada selline lahendus. Töö võiks pakkuda huvi inimestele kes tegelevad asjade internetiga ning informaatika tudengitele, kes on valinud endale robotika aine.

Bakalaureusetöö koosneb kolmest peatükist. Esimeses peatükis tutvustatakse antud valdkondi, mida töös käsitletakse, nimelt Arduinot, Internet of Things'i (asjade interneti) ning Microsoft Azure'i pilveplatvormi. Teises peatükis antakse ülevaade olemasolevatest materjalidest ning analüüsib neid. Kolmandas peatükis räägitakse tehtud katsetest, luuakse töötav lahendus, autor räägib töö käigus omandatud kogemustest ning lõpuks analüüsitakse saadud tulemusi.

1. Valdkonna üldtutvustus

Antud teemaga on seotud kolm suurt osa. Selles peatükis tutvustab autor esiteks Arduino platvormi. Arduino representeerib kasutaja enda loodud füüsilist seadet mis edastab mingit sorti andmeid. Internet of Things (asjade internet) on andmete kogu millele on võimalik ligi pääseda igalt poolt, iga kell ning peaaegu iga seadmega millel on interneti ühendus, seda kutsutakse ja pilveks. Viimasena on Microsoft Azure. Microsoft Azure ühendab ja seob omavahel eeltoodud kaks osa üheks suureks tervikuks, pakkudes sellist potentsiaali mille piiranguks on inimese enda kujutlusvõime.

1.1 Arduino

Arduino on pangakaardi suurune avatud lähtekoodiga (nii tarkvara kui ka riistvara) elektroonika arendusplatvorm. Arduino on loodud selleks, et nii algajal kui ka professionaalil oleks võimalik lihtsalt ja odavalt seadmeid programmeerida mis suudavad oma ümbruskonnaga suhelda, korjata sensorite abiga infot ning selle info abil juhtida teisi elektroonilisi komponente (näiteks LED-id, mootorid, alarmid jne). Samuti on Arduino üsnagi odav, suurusjärg 20€. Kui see aga ei sobi siis kuna on tegemist avatud lähtekoodiga, on igalühel võimalus ka oma arduino ehitada. (Arduino)

Arduino projekt sai alguse Itaaliast, Ivreast 2002. aastal. Arduino nimi tuleneb Itaalia linnas Ivreas asuvast pubist nimega Bar di Re Arduino kus osad projekti autorid kohtumas käisid. Pubi ise on aga nime saanud Itaalia kuninga Arduin of Ivrea (võimul 1002-1014) järgi. (Kushner, 2011)

Kõige populaarsem Arduino variant on Arduino Uno. Samuti kasutab autor just Arduino Unot oma katsetes. Arduino Unol on Atmega328P mikrokontroller mille taktisageduseks on 16 MHz, 32 KB välmälu (*Flash Memory*), 1KB püsिमälu (*EEPROM*) ja 2kB juhupöördlusmälu (*SRAM*). Arduino saab toite kas üle USB kaabli või kasutades välist toiteblokki. Toitebloki kasutamisel peab aga vool jääma 6V ja 20V vahele, soovitatavalt 7V – 12V. Samuti võivad olla ühendatud samaaegselt noo toiteblokk kui ka USB juhe. Sellisel juhul võtab arduino oma toite automaatselt toiteblokilt. (Võsandi, 2013)

Arduinol on kokku kakskümmend sisendit/väljundit millega on võimalik seadmetega ühendada, neliteist digitaalväljundit millest kuus on PWM (*Pulse-Width Modulation*)

väljundit ning kuus analoogsisendit. Samuti asuvad plaadil 5V ja 3.3V väljundid ning kolm maandust. (Võsandi, 2013)

Arduinol on oma arenduskeskkond IDE (*Integrated development environment*) mis on kirjutatud Javas. Ka arenduskeskkond üles ehitatud nii, et ka algajad, kes ei pruugi programmeerimisest palju teada, saaksid seal hakkama. Arduino IDE keel põhineb C ja C++ baasil ning on mingil määral lihtsustatud. (Oolep, 2013)

1.2 Internet of Things (asjade internet)

Asjade internet on süsteem mis koosneb omavahel ühenduses olevatest masinatest, objektidest, loomadest või inimestest, keda või mida on võimalik unikaalselt eristada. Samuti on neil võime omavahel suhelda ilma, et kontakt oleks inimeselt inimesele või inimeselt arvutile. (Rouse, 2016)

Sõna „asi“ asjade interneti kontekstis võib tähendada inimest kellele on paigaldatud südamestimulaator, kiibistatud lemmikloom või auto millel on sensorid mis annavad teada sellest, kui rehvide rõhk langeb liiga madalale. Põhimõtteliselt iga asi millele on võimalik määrata IP aadress ning millel on võimalus andmeid üle interneti saata ja vastu võtta. (Rouse, 2016)

Mõiste „Internet of Things“ võeti esmakordselt kasutusele 1999. aastal Kevin Ashtoni poolt. Kuigi mõiste võeti kasutusele alles 1999. aastal, oli selle arendusega tegeletud juba aastakümneid. (Rouse, 2016)

Esimeseks internetiga ühendatud seadmeks, peale arvuti, sai 1982. aastal Coca-Cola müügiautomaat. Automaat loodi Carnegie Melloni ülikoolis. Automaat oli suuteline andma ülevaate olemasolevatest jookidest ning anda infot kas joogid on külmad või mitte. (Maxey, 2016)

2008. ja 2009. aasta vahel ületas internetiga seotud seadmete arv maailma rahvaarvu. Täna on kasutusel erinevate andmete järgi 6,4-17,7 miljardit internetiga ühendatud seadet, kui arvestada, et maailmas on ligikaudu seitse miljardit inimest, teeb see 0,9-2,5 seadet inimese kohta. Aastaks 2020 ennustatakse aga, et see arv on 28-30 miljardit. Trend näitab igatahes seda, et see arv on endiselt kasvamas. (Nordrum, 2016)

1.3 Microsoft Azure

Microsoft Azure on Microsofti poolt loodud pilveplatvorm. Azure'i platvormi tutvustati avalikusele esmakordselt aastal 2008, toona veel Windows Azure nime all. Kättesaadavaks sai platvorm aga kaks aastat hiljem, aastal 2010. Azure ei olnud siis veel üldsegi nii suur nagu ta praegu on. Azure pakkus platvormi .NET, SQL ja Microsofti enda Live teenustele. Paljud inimesed olid „pilve“ üle skeptilised, ilmselt inimesed ei tahnud mida see üldse tähendab. (Callaway, 2015)

Aastatega täienes Azure'i funktsionaalsus märgatavalt. Tulid uued teenused ning vanud teenuseid arendati ning optimeeriti edasi. Aastal 2014 muudeti platvormi nimi Microsoft Azure'iks, mis on kasutusel tänapäevani. (Azure, Upcoming Name Change for Windows Azure, 2014)

Tänapäevaks on Azure'ist saanud maailmas üks juhtivamaid pilveplatvorme. Platvorm pakub võimalust luua andmebaase, virtuaalmasinaid, salvestada andmeid, analüüsida andmeid ja veel palju muud. Azure'i on võimalik kasutada kõigil, nii firmadel kui ka tavalistel inimestel. Vahe seisneb lihtsalt mastaabis. Azure võimaldab nii tavainimestel kui ka firmadel kasutada ja töödelda suurel hulgal andmeid. (Azure, Azure Products)

Azure'is käib kõik vastavalt mastaabile, mida suurem, võimsam või kiirem valitud teenus on, seda rohkem tuleb selle eest kuus maksta. Teenuste eest tuleb aga ainult siis maksta kui teenust ka realselt kasutatakse. Kui mingi teenus mis peaks olema kuutasuga aga seda ei ole sisse lülitatud, ei pea ka selle eest raha maksma. Kõik toimib samamoodi nagu kodus elektri tarbimine, makstakse selle eest mida tarbitakse. See muudab platvormi väga mitmekülgses. (Jr & teised, 2015)

2. Ülevaade olemasolevast materjalist

Autor alustas materjali otsimisega lootuses, et leiab mitmeid terviklike õpetusi kuidas ühendada Arduino Uno Microsoft Azure'iga, kasutades selleks arvutit millel on Windowsi operatsioonisüsteem. Seda kahjuks ei juhtunud. Autor leidis küll materjale kus oli kasutatud küll Arduinot aga ülejäänud erines autori eeldustega.

Väga palju leidis autor materjale milles oli kirjeldatud kuidas ühendada Arduino Raspberry Pi-ga. Samuti oli mitmeid allikaid mis näitasid kuidas ühendada Arduino (või Arduinole sarnane platvorm) kas üle WiFi või lokaalse võrgu (LAN) Raspberry Pi-ga. Nendest materjalidest ei olnud just väga palju kasu, kuid need materjalid andsid autorile ülevaate mida on teised inimesed kasutanud, et saada töötav lahendus. (Whitehouse, 2016; Center)

Katsete käigus ilmnis ka materjale mis olid aegunud. Autor otsis lahendust probleemile kuidas ühendada Python Microsoft Azure'iga, leides, et Pythonile on täiesti olemas Microsoft Azure SDK moodul. Edasi uurides, leidis autor ka materjale kus oli kirjas täpselt see mida autor otsis. Järele proovides selgus, et näited ei tööta. Lähemalt uurides leidis autor, et Pythoni moodulist on välja antud uus versioon mille ülesehitus on muutunud, muutes materjali aegunuks. Uuendatud versiooni kohta autor materjale ei leidnud. (faister, 2014; Loogman, 2014)

Kuna autor ühtegi tervikliku materjali ei leidnud mis oleks rahuldanud kõiki kriteeriume, pidi autor väga palju vahelülisid otsima. Üheks kõige olulisemaks kohaks kust autor infot sai, oli Stack Overflow. Just sealt leidis autor vastuseid sellistele küsimustele mida polnud kuskil mujal kirjeldatud. Stack Overflows saavad kasutajad küsida abi küsimusi mis on seotud programmeerimisega. Autor tundis kergendust kui leidis, et ka teised kasutajad on olnud silmitsi täpselt samade, väga spetsiifiliste probleemidega, ning leidub inimesi kes on seal oma lahendusi jaganud. (BlueCastle, 2016; hardillb, 2016; aladeen, 2016)

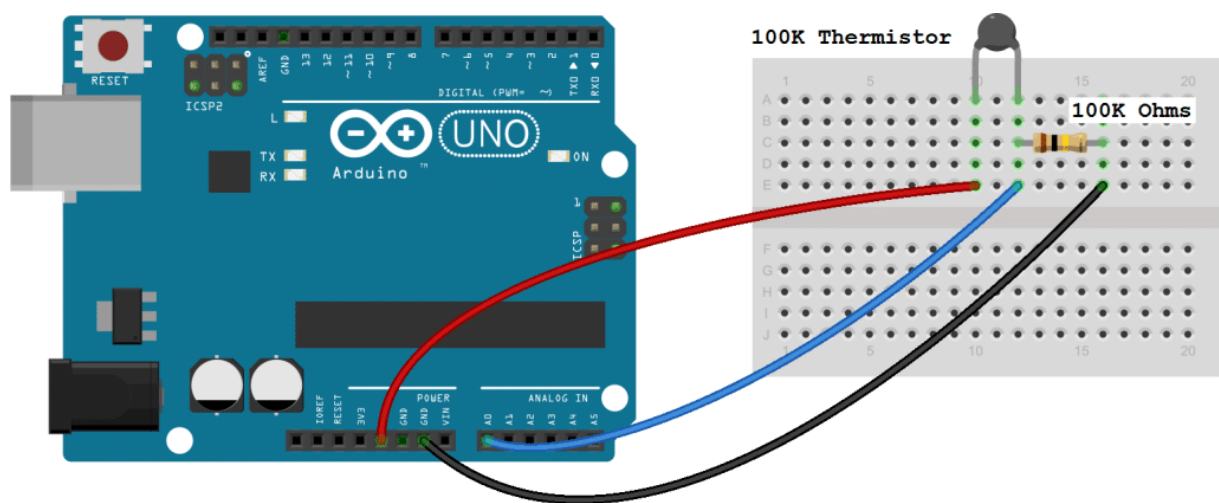
Autori arvates oli materjale antud teema kohta piisavalt. Ei leidunud küll täpset lahendust nagu autor oleks soovinud, kuid samas sai autor palju mõtteainet mida on võimalik veel sarnase lahendusega korda saata. Õnneks olemasolev materjal suutis autori õigele teele suunata, et jõuda töötava lahenduseeni. (Hilscher, 2016; Mackes, 2015)

3. Omapoolsed katsetused, lõplik lahendus, kogemused ja analüüs tulemustest

Antud peatükis annab autor ülevaate tehtud katsetustest, annab ülevaate mida ta tegi, et saavutada töötav lahendus. Samuti räägib autor milliseid kogemused ta katsete käigus omandas ning lõpuks analüüsib autor saadud tulemusi.

3.1 Katsetused ja tekkinud probleemid

Katsetused algasid Arduino Uno, maketeerimisplaadi, TMP103 temperatuuri anduriga, hiljem juba termperatuuri ja õnhuniiskuse anduri DHT22-e ning viie juhtmega, millega komponendid omavahel ühendada. Kõigepealt tuli kõik komponendid õieti omavahel ühendada ning ka Arduino ise üle USB kaablu arvuti külge ühendada. Peale seda pidi Autor kirjutama koodi, mis oleks võimeline lugema ning ka edastama andurilt saadud infot üle seriali, ehk edastama selle info üle USB kaabli (vt Joonis 1). (Basics)



Joonis 1. Algne TMP103 ühendus Arduinoga (TMP103 on 10k ohmine, samuti kasutati ka 10k ohmist takistit) (Basics)

Järgmisena tuli see info kuidagi arvutist vastu võtta. Autor otsustas kasutada selleks Pythoni versiooni 2.7.13. Autor valis Pythoni sellepärast, et on seda varem õppinud ning isegi robotika aines kasutanud. Python tuli kasuks juba sellepärast, et Pythonil on serial moodul mis võimaldab üpriski kerge vaevaga suhtlust üle seriali. Selleks pidi autor kirjutama vastava koodi, et saada Arduino anduri info arvutisse.

Esimeseks probleemiks kujunes Microsoft Azure. Autor kasutas algselt kontol loomiseks koodi, mille ta sai Microsoft Imagine'i platvormilt. Sealt platvormilt on võimalik (üli)õpilastel võimalik ligi pääseda Microsofti toodetele nagu näiteks kõige uuem Windows. Sealt saadud võtmega sai autor küll luua andmebaasi, kuid millelegi muule vajalikule ligi ei pääsenud. Microsoft Azure ütles sellepeale „*Update your subscription*“. Autoril ei jäänudki muud üle kui minna üle *Free trial*-i peale. (Azure, Microsoft Imagine Azure Offer)

Järgmiseks takustuseks kujunes autorile Pythoni ühendamine IoT Hubiga. Autor proovis alguses kasutada ühes materjalis olnud lahendust, kuid edutult. Alguses ei saanud autor milles viga võib olla, sest kõik tundus nagu olemas olevat. Autor isegi installis moodulit mitu korda uuesti, arvates, etäkki läks midagi vahepeal kaduma. Mooduli enda lähemalt uurides selgus, et moodulit on uuendatud ning vanad funktsioonid ei tööta enam. See selgus loomulikult pärast mitut tundi peavalu. Autor pidi seega teistsuguse lahenduse leidma. Autor valis selleks paho-mqtt mooduli. (faister, 2014)

Veel üheks väikseks probleemiks kujunes autorile üllatuseks DHT22 sensor. Need sensorid saabusid robotika klassi siis kui autor oli oma tööga poole peal. Kuna autor nägi uues sensoris rohkem potentsiaali kui analoogsel sensoril, mida ta seni oli kasutanud, otsustas autor hakata kasutama uut DHT22 sensorit. Kui autor selle sensori Arduino külge ühendas ja vaatas mis numbreid see näitab, tekkisid autoril kohe küsimused. Nimelt väitis sensor, et robotika klassis on 25 kraadi sooja, õhuniiskus tundus aga olevat mõistlikkuse piires. Seda autor kindlasti ei uskunud. Autor mõõtis õhutemperatuuri voltmeetriga üle (autori üllatuseks oli voltmeetril temperatuurimõõtja olemas). Selgus, et temperatuurandur näitab keskeltläbi neli soojakraadi rohkem kui tegelikult. Autor testis ka teisi DHT22 sensoreid, kõik näitasid konstantselt sama temperatuuri. Autor ise arvab, et see viga võib tulla „*DHT sensor library*“-st mida on vaja, et selle anduriga mõõte kätte saada. Sellest tulenevalt pidi autor lahutama Arduino koodis temperatuurilt neli kraadi „ $(t-4)$ “ (vt Lisa 1. Arduino kood).

Viimaseks suuremaks probleemiks kujunes autoril Microsoft Azure'is Steam Analytics. Stream Analytics edastab andmed, mis saadetakse IoT Hubi, ning edastab need SQL serverisse. Autor ühendas ilusti sissetuleva infoallika IoT Hubiga ning väljamineva infoallika oma andmebaasi ja tabeliga. Isegi pärast seda ei jõudnud andmed andmebaasi kohale. Pärast väikest uurimist selgus, et sellel teenusel on „Start“ nupp mis selle tööle paneb. Millegipärast eeldas autor, et see teenus hakkab automaatselt tööle. Andmeid aga ikka läbi ei tulnud, ent oli märgata edusamme. Nüüd hakkas teenus vigu andma, et mingi tabeli väli on puudu ning tabelisse ei saa andmeid lisada. Pärast mõningast uurimist, selgus, et tuleb *Query* ära muuta.

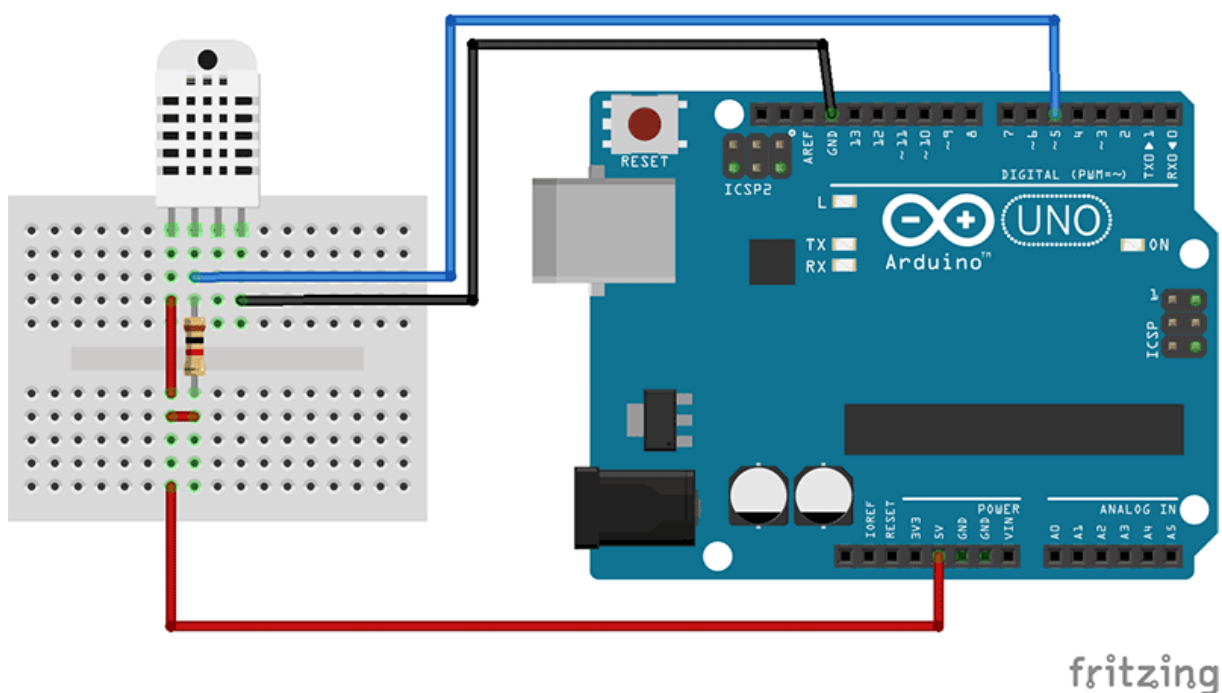
Sinna tuli täni asemel panna need tabeli väljad kuhu andmeid soovitakse lisada. Autor arvab, et seda pidi ta tegema sellepärast, et saadetas info ei olnud ära märgitud ID välja, tabelis oli see väli aga olemas, kindel ta selles muidugi ei ole.

3.2 Lõplik lahendus

Lõplikuks lahenduseks kasutas autor juba eelpool mainitud riistvara. Arduino Uno, maketeerimisplaati, DHT22 temperatuuri ja õhuniiskuse andurit ning viite juhet. Samuti kasutab autor lahenduses Pythoni versiooni 2.7.13.

3.2.1 Arduino

Esmalt tuli andur ühendada õieti Arduinoga (vt Joonis 2). Samuti on vaja installeerida enda Arduino arendusplatvormile „*DHT sensor library*“, et sensori käest infot saada. Peale seda kirjutab autor Arduinole vajaliku programmi mis hakkaks edastama saadud andmeid üle seriali. Nagu eespool mainitud sai, tuli autoril DHT22 sensorit kasutades õige temperatuuri saamiseks saadud temperatuurist maha lahutada neli kraadi. Järgmisena asus autor tegelema Pythoniga. Lõpliku Arduino koodi leiab lisadest (vt Lisa 1. Arduino kood). (codebender_cc)



Joonis 2. DHT22 ühendamine arduinoga (Hoyt, 2014)

3.2.2 Python

Antud Pythoni lahenduse kasutamiseks oli vaja kahte moodulit kasutada: paho.mqtt ning serial. Juhul kui neid ei ole olemas siis tuleb need moodulid tõmmata. Seada oli väga lihtne teha üle käsurea kasutades Pythoni enda pip-i (package manager) kasutades järgmist kahte koodinäidet (vt Koodinäide 1). (Light, 2015)

```
pip install paho-mqtt
```

```
pip install pyserial
```

Koodinäide 1. Pythoni moodulite installeerimise käsklused

Juhul kui peaks tulema neid käsklusi kasutades sõnum: „'pip' is not recognized as an internal or external command, operable program or batch file.“, tähendab see seda, et Windows ei tea kus asub „pip.exe“, et käsklust läbi viia. Selle lahendamiseks kasutas autor järgmist käsurea käsklust (vt Koodinäide 2). (fr1tz, 2016)

```
setx PATH "%PATH%;C:\Python27\Scripts"
```

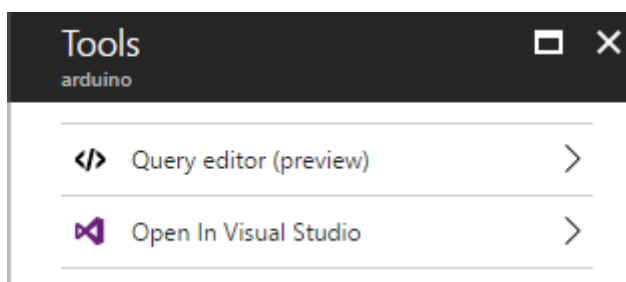
Koodinäide 2. Käsklus, peale mida on võimalik kasutada „pip“ käsklust

3.2.3 Microsoft Azure

Järgmisena asus autor tegelema Microsoft Azure'iga. Et pääseda ligi kõigile vajalikele võimalustele mida Azure pakub, pidi autor tegema endale „Free trial“ konto. Konto loomiseks on vaja isikut tuvastada läbi krediitkaardi (sobib ka deebetkaart millel on krediitkaardi funktsionaalsus). See süsteem on selleks, et Azure'i ei oleks võimalik kuritarvitada. Hea on see, et *Free trial*-iga tuli kaasa 170€ krediiti mida on võimalik 30 päeva jooksul kasutada, peale seda krediit kaob. See andis autorile hea võimaluse kasutada vajalike teenuseid ilma raha maksmata. Samuti kui 30 päeva läbi saab siis ei hakata automaatselt kontolt raha võtma ilma konto omaniku nõusolekuta. (Azure, Azure Free Trial)

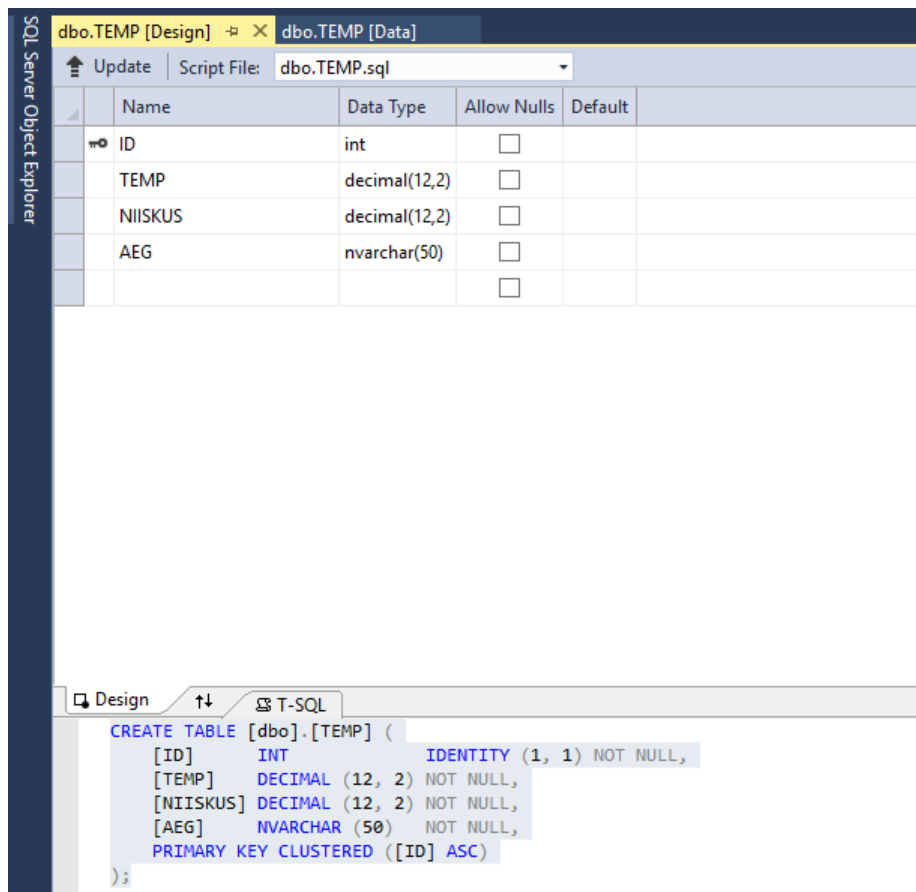
Peale konto loomist tegi autor esimese asjana SQL serveri. Serverile tuli anda nimi mis on unikaalne, tuli luua uus *Resource group*, määrata serveri asukoht ning serveri maht. Autor kasutas serveri asukohaks North Europe. Peale seda oli võimalik luua sinna serverisse ka andmebaas kuhu lõpuks kõik info salvestatakse. Andmebaasi loomine käis nagu serveri loomine. Ka andmebaasilt tuleb anda nimi, *Resource group*, mis loodi serveri loomise ajal, valida eelnevalt tehtud server ning *Pricing tier*-is mahuks *Basic* variant. Ülejäänud valikud

võivad jääda nii nagu nad on. Andmebaasi tegi autor ka tabeli, milles on väljad ID, TEMP, NIISKUS ja AEG, et oleks võimalik pärast andmeid ka mingil määral sorteerida ning ka välja kuvada. Mugavat tabeli loomise varianti Azure'is aga kahjuks ei ole (vt Joonis 3).



Joonis 3. Azure'is andmebaasile ligipääsu variandid

Võimalus on küll kasutada Azure'i enda *Query editor*-i kuid autor leidis, et see on liiga tülikas ja ebamugav. Teine võimalus on installida Microsoft Visual Studio, et andmebaasi hallata. Seda autor ka tegi, ning installis endale Visual Studio 2017 Community versiooni, antud versioon on Microsofti poolt täiesti tasuta saadaval. Samuti töötab selleks 2015 Update 2 ja 2013 Update 4 versioon Visual Studiost. Peale seda tuli autoril kindlaks teha see, et Visual Studiost oleks olemas „SQL Server Data Tools“, ilma selleta ei olnud võimalik Visual Studiost andmebaasiga siduda. Ühendamiseks tuli sisestada loodud andmebaasi andmed. Autor kasutas Visual Studiost selleks, et teha muudatusi algsele tabelile ning vaadata kas ja millisel kujul andmed tabelisse jõuavad (vt Joonis 4). (Calbimonte, 2015; Kess & teised, 2017)

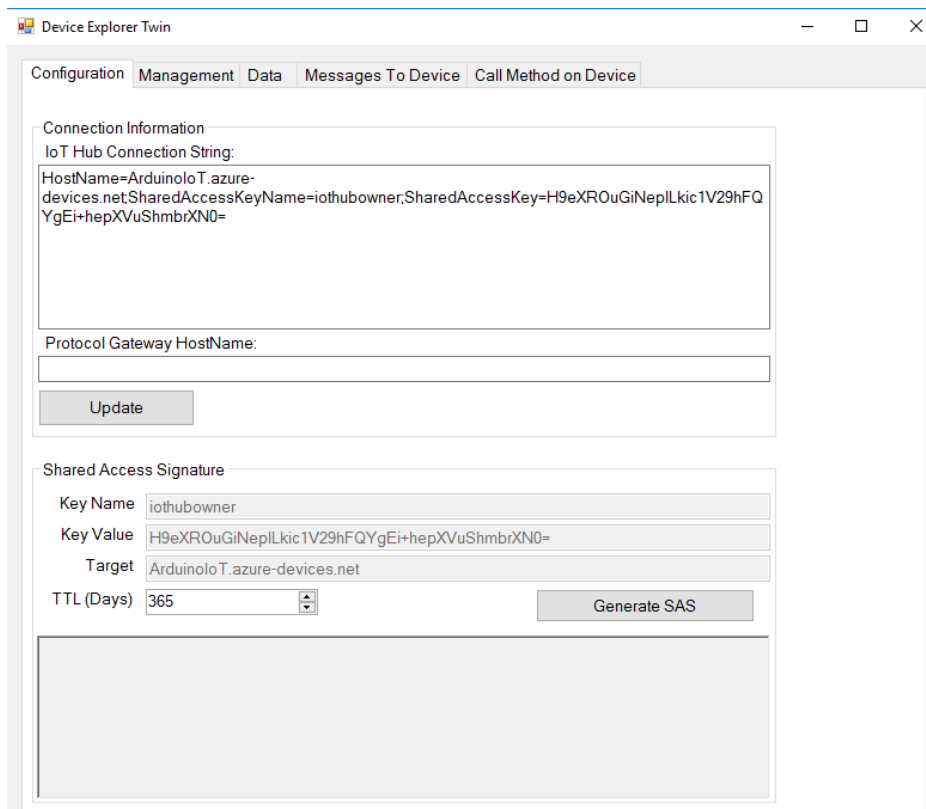


Joonis 4. Autori loodud tabeli ülesehitus

Et andmed jõuaksid Azure'i kohale, tuli luua endale IoT Hub imeline teenus. Just IoT Hubile oli vaja andmed saata, et neid oleks võimalik kuhugi veel edasi saata. Iot Hubi loomine käis sama moodi nagu serveri ja andmebaasi loomine. Autor valis endale Free variandi IoT Hubist. (Basak, 2015)

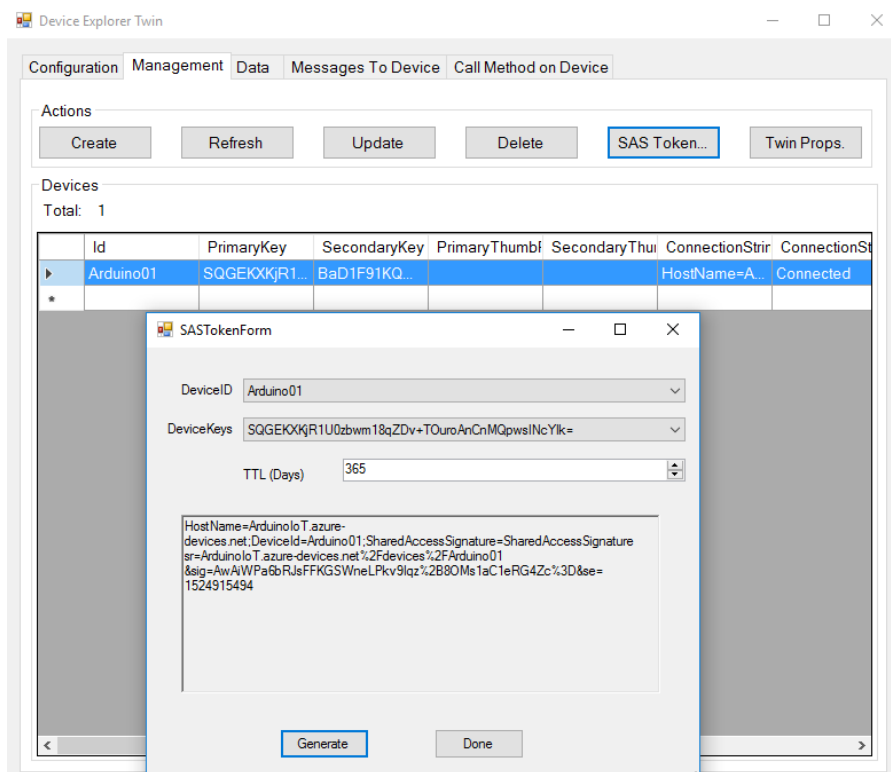
3.2.4 Pythoni sidumine Microsoft Azure'iga

Pythoni ja IoT Hubi omavahel sidumine oli ilmselt kõige keerulisem asi. Kõigepealt oli vaja tõmmata selline programm nagu Device Explorer GitHub'ist. See programm lihtsustas tunduvalt edasist katsetamist. Esmalt tuli võtta IoT Hubist, Shared access policies alt vajutada iothubowneri peale. Sellepeale avaneb aken kus oli „*Connection string—primary key*“. See väärtus tuli kopeerida Device Exploreri IoT Hub Connection String aknasse ning vajutada Update nuppu. See ühendas Device Exploreri IoT Hubiga (vt Joonis 5).



Joonis 5. Device Exploreri IoT Hubiga ühendamine

Järgmisena tuli lisada sinna uus seade, mida saai teha „*Management*“ alt ning vajutada „*Create*“. Seadmele tuli määrata vabal valikul nimi ning soovitatavalt jätta linnuke „*Auto Generate Keys*“ peale ning vajutada „*Create*“. Kui vastav seade on menüüsse tekkinud siis on nüüd vaja sellele kuidagi andmeid saata. Selleks on vaja kuidagi ühendus tekitada. Selleks tuleb esmalt valida vastav seade menüüst ning vajutada „*SAS Token*“ nupule ning genereerida uus võti. Võtme TTL võiks olla 365 päeva. Kui võti on genereeritud siis tuleks kõik info mis alumisse kastikesse tekkis kuhugi tekstifaili kopeerida. Seda infot läheb kindlasti pärast vaja. Kui see info aga peaks mingil moel ära kaduma siis annab alati uue võtme genereerida (vt Joonis 6). (ThingLabs.io)



Joonis 6. Ligipääsuvõtme genereerimine

Seda kõike oli ka võimalik ilma Device Explorerita teha, juhul kui see programm ei peaks töötama või ei ole soovi seda kasutada. Selle kohta leidis autor küll materjale, kuid otsustas seda ise mitte kasutada. (npm)

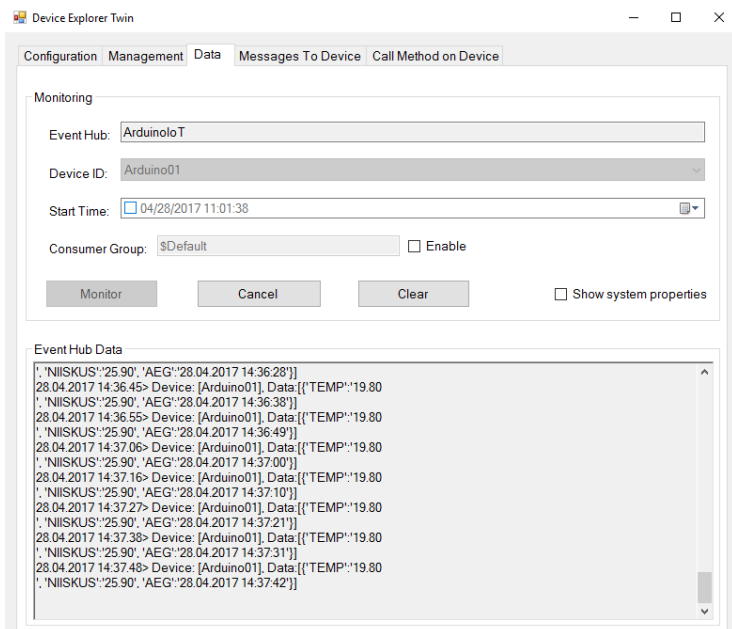
Nüüd tuli võtta SAS Tokeni poolt genereeritud Shared access siganture ja lisada see Pythoni koodi *password*-i välja. Kindlasti pidi see algama „*SharedAccessSignature sr=*“, muidu see ei töötanud. Samuti tuli sisestada „*username*“ välja kujul „*<IoT hubi nimi>.azure-devices.net/<seadme nimi>*“. Andmed mida edastatakse peavad kindlasti olema JSON vormingus. Ühe viimase asjana mida on Pythoni juures vaja, on sertifikaat. (Vokkarne & teised, 2017)

Sertifikaati ei pidanud õnneks ise genereerima. Sobivad ka DigiCert-i poolt väljastatud sertifikaadid. Antud juhul kasutas autor Baltimore CyberTrust Root nimelist sertifikaati. See on täiesti tasuta saadaval ning tuleb endal lihtsalt alla tõmmata. (DigiCert)

Nüüd tuleb see sertifikaat lisada ka Pythoni koodi. Koodilõik peab lõpuks välja nägema „*client.tls_set("<sertifikaadi nimi>.crt")*“. Lõplik Pythoni kood on leitav lisadest (vt Lisa 2. Pythoni kood).

Lõpuks oli võimalik testida kas andmed jõuavad IoT Hubi või mitte. Selleks tuli appi uuesti Device Explorer. Device Exploreri alt tuli valida *Data* ning vajutada „*Monitor*“ nupule. Nüüd

hakkas Device Explorer kuulama sõnumeid mis IoT Hubile saadetakse. Kui autor Pythoni koodi käima pani siis hakkasid ilmuma ka vastavad sõnumid. See tähendas, et kõik töötab ilusti ja andmed jõuavad IoT Hubi kohale (vt Joonis 7). Nüüd oli vaja andmed kuidagi edasi saata SQL serverisse. (Basak, 2015)



Joonis 7. Saadetud andmete kontrollimine

Et IoT Hubist jõuaksid andmed SQL andmebaasi, on vaja kasutada Stream Analytics job'i. Selle loomine käib samamoodi nagu ka eelmistel Azure'i moodulite loomine. Peale selle loomist avaneb aken kus tuleb määrata sisend ja väljund. Sisendi tüüp peab olema „Data Stream“ ning Source-ist tuli valida IoT hub, seejärel tuli valida sinna alla tekkinud kastist enda loodud IoT hubi nimi. Endpoint peab kindlasti olema Messaging, muidu ei lastud saadetud andmeid läbi. Väljundiks tuli panna loodud SQL andmebaas. Et elu natuke lihtsamaks teha valis autor Import optioni alt valiku „Use SQL database from current subscription“, see valik täitis järgnevad kaks välja automaatselt ära. Lõpuks tuli täita andmebaasi kasutajanimi, parool ning ka tabeli nimi kuhu andmed salvestada. Viimane asi mis Steam Analytics'i juures teha tuli oli Query. Kuna autor kasutas enda tabelis ID välja ning saadetavas infos ei olnud ID ära märgitud, ei sobinud vaikimisi olnud query. Autor tegi sellele paar väikest muudatust (vt Koodinäide 3). Samuti ei tuleb Stream Analytics job peale neid muudatusi käima panna, muidu ei juhtunud mitte midagi. Nüüd jõudsid kõik andmed lõpuks ka edukalt SQL andmebaasi. (Jayakody, 2016)

```
SELECT
    TEMP, NIISKUS, AEG
INTO
    ArduinoToSQL
FROM
    DataStream
```

Koodinäide 3. SQL Query

3.2.5 Power BI

Viimase asjana otsustas autor need andmed mis said andmebaasi salvestatud, kuidagi ka välja kuvada. Selleks valis autor järjekordselt Microsofti loodud rakenduse, Power BI. Power BI kasutamiseks tuli autoril luua uus konto. Tavalise hotmaili kontoga mida autor kasutas Azure'is, ei olnud võimalik Power BI-sse kontot luua. Power BI soovis konto loomiseks töö emaili. Autor kasutas selleks oma TLÜ emaili kontot. Samuti pidi autor aktiveerima 60 päevase *Trial* perioodi mis võimaldab ligipääsu „Pro“ versioonis olevale funktsionaalsusele. Seda on vaja selleks, et oleks võimalus ligi pääseda Azure SQL andmebaasile ning ka paljule muule. (Saxton, 2017)

Peale konto loomis installeeris autor omale Power BI Desktop programmi, et seadistada milliseid andmeid ja kuidas kuvatakse. Esimese asjana tuli lahti võtta installeeritud programm ja äsja loodud kontoga sisse logida. Järgmisena tuli valida „*Get Data*“ ning sealt „*Azure SQL database*“. Nüüd tuli sisestada serveri nimi ning valida „*Direct Query*“ valik, peale seda tuli vastava andmebaasi andmetega sisse logima (*Database* valiku alt). Peale seda tuli valida tabel mille andmeid kasutatakse ja lõpuks oli võimalik alustada andmete välja kuvamisega (vt Joonis 8).

Navigator

Display Options

- iotarduino.database.windows.net [1]
 - arduino [2]
 - sys.database_firewall_rules
 - TEMP**

TEMP

Preview downloaded on laupäev

ID	TEMP	NIISKUS	AEG
1	20	22	27.04.2017 13:53:12
2	20	21,9	27.04.2017 13:53:23
3	20	21,9	27.04.2017 13:53:33
4	20	22	27.04.2017 13:53:44
5	20	21,9	27.04.2017 13:53:54
6	20	21,9	27.04.2017 13:54:05
7	20	22	27.04.2017 13:54:16
8	24,1	99,9	27.04.2017 13:54:26
9	25,8	99,9	27.04.2017 13:54:37
10	22,8	99,9	27.04.2017 13:54:47
11	21,4	99,9	27.04.2017 13:54:58
12	20,8	99,9	27.04.2017 13:55:08
13	20	92,6	27.04.2017 13:55:19
14	20,3	63,3	27.04.2017 13:55:29
15	20,5	53,5	27.04.2017 13:55:40
16	20,5	43,4	27.04.2017 13:55:51
17	20,4	35,4	27.04.2017 13:56:01
18	20,4	29,6	27.04.2017 13:56:12
19	20,4	27,3	27.04.2017 13:56:22
20	20,4	25,9	27.04.2017 13:56:33
21	20,4	25	27.04.2017 13:56:43
22	20,4	24,5	27.04.2017 13:56:54
23	20,3	23,9	27.04.2017 13:57:04

Select Related Tables

Load Edit Cancel

Joonis 8. Tabeli valimine

Autor otsustas kasutada nii temperatuuri kui ka õhuniiskuse näitamiseks kasutada poolikut rõngasdiagrammi ja paigutada need kõrvuti (vt Joonis 11). Nende diagrammide kohale otsustas autor lisada kuupäeva ja kellaaja millal viimased andmed kohale jõudsid, alla paigutas autor joondiagrammi mis näitab 60 viimast õhuniiskuse ja temperatuuri mõõtmist. (Munro, 2015)

Selleks, et viimaseid andmeid kätte saada, tuli menüüribalt valida Modeling ning luua kolm *New Measure*'it. Et saada kätte viimane kellaeg ja kuupäev, tegi autor järgmise funktsiooni (vt Koodinäide 4).

```
Last Time = CALCULATE (MAX (TEMP [AEG] ) )
```

Koodinäide 4. Viimase kuupäeva ja kellaaja arvutamise funktsioon

Viimase temperatuuri ja õhuniiskuse arvutamise funktsioonid tulid üksteisele väga sarnased, lihtsalt tabeli väli sai erinev (vt Koodinäide 5). Et neid funktsioone kasutada saaks, tuli sätetest, *DirectQuery* alt panna linnuke valikule „*Allow unrestricted measures in DirectQuery mode*“. Peale seda tuli programm taaskäivitada. (JeffDuzak, 2016; Sean, 2016)

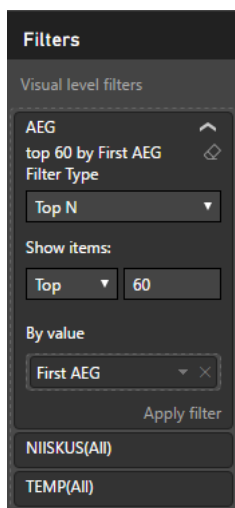
```
Last Temp = CALCULATE (DISTINCT (TEMP [TEMP] ) ;  
FILTER (ALL (TEMP [AEG] ) ; [AEG] = MAX (TEMP [AEG] ) ) )
```

```
Last Niiskus = CALCULATE (DISTINCT (TEMP [NIISKUS] ) ;  
FILTER (ALL (TEMP [AEG] ) ; [AEG] = MAX (TEMP [AEG] ) ) )
```

Koodinäide 5. Viimase temperatuuri ja õhuniiskuse arvutamise funktsioonid

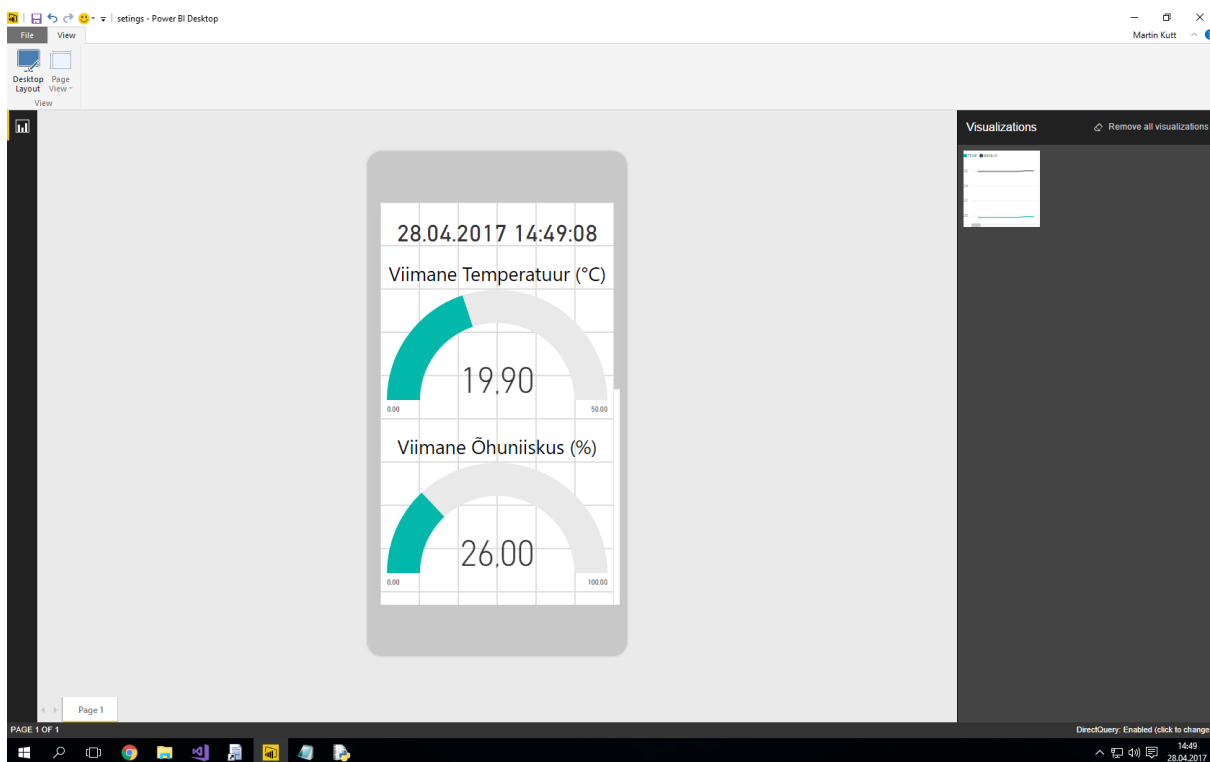
Et saada alumisel diagrammil ainult viimased 60 viimast mõõtmist, tuleb kasutada filtrit. Diagramm kasutab kolme tabeli välja: aeg, temp ja niiskus. Filtrit tuli panna aja alla. Aja alt

rippmenüüst tuli valida „Top N“, märkida ära kui palju tulemusi kuvada ning lõpuks lohistada tabelis asuv aja lahter viimasesse valikusse (vt Joonis 9). Peale seda tegi autor mõned visuaalsed muudatused oma soovide järgi, nagu pealkirjade muumine jne. Seda kõike on võimalik teha kui valida diagramm ning vajutada värvirulli ikooni peale.



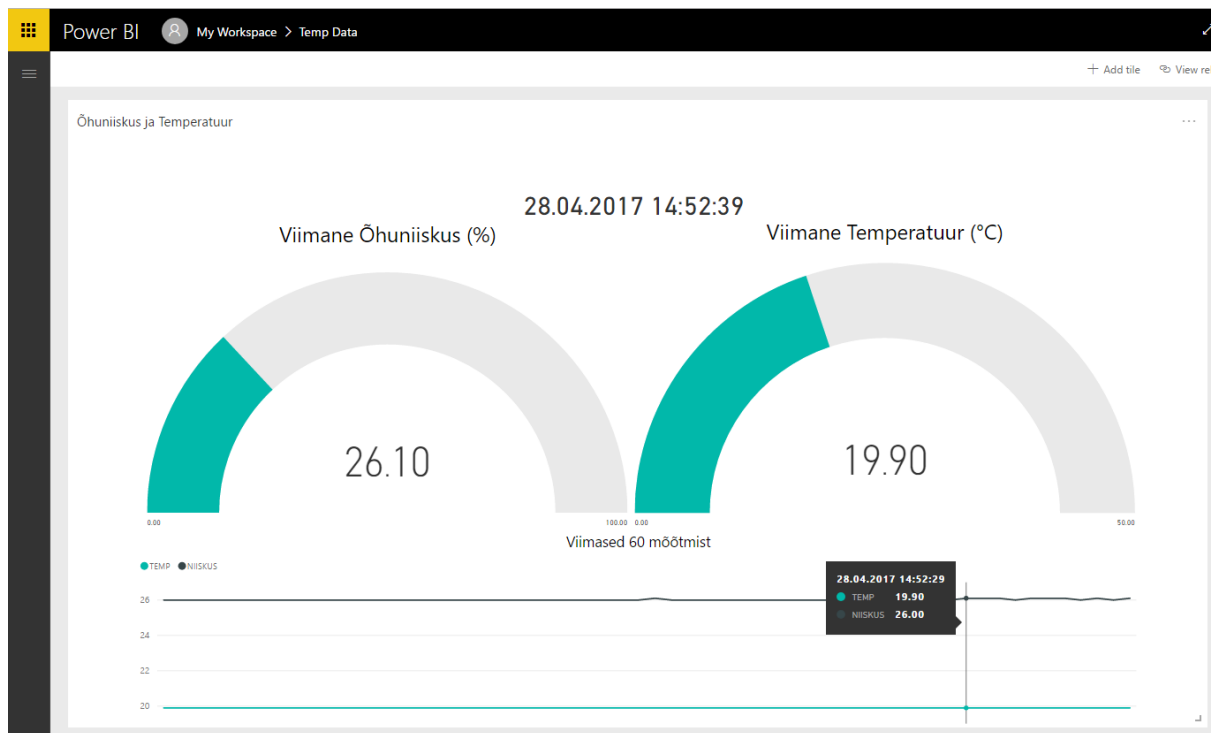
Joonis 9. 60 viimase tulemuse välja kuvamine aja järgi

Power BI võimaldab teha ka nutitelefonidele mõeldud varianti andmete kuvamisest. Selleks tuli valida View alt Phone Layout ning lihistada lohistada valmistatud diagrammid vastavalt soovile telefoni ekraanile (vt Joonis 10). (Sparkman, 2017)



Joonis 10. Mobiilivaate loomine

Viimase asjana tuli valmistatud diagrammid ka interneti üles laadida, selleks tuli valida Publish. Nüüd oli võimalik kas läbi brauseri või üle telefoni rakenduse ning PowerBI-sse sisse logides näha oma viimaseid andmeid (vt Joonis 11). Boonuseks on see, et kui lehekülge värskendada siis uuenevad ka kuvatud andmed.



Joonis 11. Lõplik disain andmete väljakuvamisel

3.3 Kogemused

Autor omandas katsetuste käigus palju uusi kogemusi. Üks peamisi asju mis autor õppis oli see, et ühtegi materjali kohta ei tasu uskuda, et antud lahendus töötab sajaprotsendiliselt. Isegi kui materjali vanus on üks kuni kaks aastat, võib see tähendada seda, et antud materjal on juba praeguseks hetkeks aegunud. Tehnoloogia areneb tänapäeval nii kiiresti, et nõuded teatud lahendustele võivad päevapealt muutuda.

Materjale, milles on kirjas mingil ajahetkel töötanud lahendus ei tasu koht huviorbiidist välja vista. Neid materjale tuleb analüüsida ning leida põhjus, miks nende lahenduskäik enam ei tööta. Selline lähenemine andis autorile kätte alguspunkti kust oli võimalik edasi minna. Samuti aitab selline lähenemine kindlaks teha millised tükid on veel täielikust lahendusest puudu, mida on vaja juurde otsida ning mis suunas tasuks edasi liikuda.

Autor sai ka täiesti uue arusaama kuidas töötab asjade internet. Enne selle töö tegemist puudus autoril kokkupuude sarnaste lahendustega, veel vähem selliste lahenduste loomisega. Autor lähenes probleemidele varasemalt omandatud kogemuste ning loogilise mõtlemisega, mis viisid autori ka lõpuks sihini.

Uusi kogemusi omandas autor ka Microsoft Azure'i kohta. Alguses tundus Azure'i platvorm väga suure ning hirmuäratavana. Töö edenedes leidis autor aga, et see platvorm on väga hästi üles ehitatud, samuti ei olnud seal midagi mis oleks olnud ülemäära keeruline. Azure on ka väga turvaline. Kõrvaliste isikute ligipääs andmetele on tehtud vägagi keeruliseks. See aga tähendas seda, et ka andmete saatmine sinna oli keerulisem kui autor algselt eeldas.

3.4 Analüüs tulemustest

Katsetuste käigus valmis lahendus, mis on võimeline üle USB juhtme arvutiga ühendatud Arduinolt saadud andmed edastama läbi Microsoft Azure IoT teenuse SQL serverisse. Samuti on kogutud andmed võimalik pärast välja kuvada, kasutades samuti Microsofti poolt loodud Power BI rakendust. Autor ise jäi saadud tulemustega rahule ning töö alguses püstitatud probleem sai seega lahenduse.

Kui on soov kasutada seda lahendust mitme Arduino ühendamiseks, tuleks ilmselt lahendust natuke kohandada. Tabelisse lisada näiteks „*deviceID*“ nimeline väli. Sinna välja saadaks iga Arduino oma unikaalse nime või koodi mis on sisse kodeeritud infosse, mille Arduino üle seriali edasi saadab. Nii oleks võimalik ka andmeid lõpuks sorteerida. Samuti võimaldab see andmeid hoopis teisiti välja kuvada Power BI-d kasutades. Nii oleks võimalik temperatuure ja õhuniiskust võrrelda kahe seadme vahel.

Kokkuvõte

Käesoleva bakalaureusetöö eesmärgiks oli leida võimalus ühendada Arduino Microsofti asjade interneti lahendusega Azure'i platvormil. Käesoleva töö tulemusena valminud lahendus võimaldab USB-ga arvuti külge ühendatud Arduino poolt kogutud andmed saata läbi Microsoft Azure IoT teenuse SQL andmebaasi, kust on võimalik need andmed pärast ka välja kuvada.

Autor tutvus erinevate materjalidega, eeldades alguses, et keegi on varem sarnase lahenduse juba teinud ning selle ka kirja pannud. Nii aga ei läinud. Autor leidis küll ligilähedasi materjale sellele, mida tegelikult vaja oleks läinud. Need materjalid andsid aga autorile kätte alguspunkti kust oli võimalik edasi minna. Samuti sai autor nendest materjalidest piisavalt infot, et luua arusaam mida on sellise lahenduse jaoks üldse vaja ja mida järgmisena otsida. Nii jõudis autor tükk tüki haaval ka oma lõpliku lahenduseni.

Autor soovitab sarnase süsteemi loojatel teha natuke eeltööd. Tuleks ära määrata kui suurt süsteemi vaja läheb, samuti milliseid Azure'i teenuseid selleks vaja läheb. Azure on lõppude lõpuks ikkagi tasuline platvorm. Seal on küll teenuseid mida on võimalik väikeses koguses ka tasuta kasutada, kuid kui sellest väheks jääb, tuleb hakata juba maksma. Kui luua sarnane süsteem nagu loodi selle bakalaureuse töö käigus siis tuleks kindlasti arvestada, et selles on kasutatud tasulisi teenuseid.

Bakalaureusetöö kirjutamise käigus omandas autor palju uusi teadmisi asjade interneti kohta ning selle kasutamisest ja seadete sidumisest sellega. Samuti sai autor uue arusaama kuidas täpsemalt liiguvad andmed seadmest Microsoft Azure-i.

Summary

Title: Connecting Arduino with Microsoft Internet of Things

The purpose of this Bachelor Thesis was to find a way to connect an Arduino through a computer to the Internet of Things solution on the Microsoft Azure platform. The result is that the completed solution takes the data from the Arduino, transfers it to the computer using an USB cable. After that the Python code takes over and sends the data to the Microsoft IoT Hub, from which the data is sent to a SQL server also located on the Azure platform. Lastly the data is also taken and visualized by Microsoft's Power BI service.

To achieve this, the author had to find and study plenty of guides and other sources. At first the author was hoping to find a complete guide on how to solve the problem from a single source. Sadly that didn't happen. The author did find sources that were only approximate matches to what he was looking for. These did give the author a point where to start from. Also the author manages to gather enough information from these sources to get a grasp of what is needed in order to create such solution, where to go next and what to look for. Doing so, the author managed to put all the puzzle pieces together and create a working solution.

The author suggests that before building an Internet of Things solution, it would be best to do some research. Consider how big the system is going to be and which Azure services are needed for that. At the end of the day Azure is a platform where you will be billed for the services that you use. There are services that are free, when used in small amounts but when that is not enough then you need to start paying. If building a solution that is similar to the one found in this thesis, it is important to understand that some services will cost money.

During the development of this thesis gave the author a new understanding of how the Internet of Things really works and how to connect devices to it. Also how the data moves from the device and into the Microsoft Azure.

Kasutatud allikad

- aladeen. (16. 04 2016. a.). *Python paho-MQTT connection with azure IoT-Hub*. Kasutamise kuupäev: 25. 04 2017. a., allikas Stack Overflow:
<http://stackoverflow.com/questions/36660410/python-paho-mqtt-connection-with-azure-iot-hub>
- Arduino. (kuupäev puudub). *Introduction*. Kasutamise kuupäev: 27. 04 2017. a., allikas Arduino: <https://www.arduino.cc/en/guide/introduction>
- Azure, M. (24. 03 2014. a.). *Upcoming Name Change for Windows Azure*. Kasutamise kuupäev: 28. 04 2017. a., allikas Microsoft Azure: <https://azure.microsoft.com/en-us/blog/upcoming-name-change-for-windows-azure/>
- Azure, M. (kuupäev puudub). *Azure Free Trial*. Kasutamise kuupäev: 23. 04 2017. a., allikas Microsoft Azure: <https://azure.microsoft.com/en-us/offers/ms-azr-0044p/>
- Azure, M. (kuupäev puudub). *Azure Products*. Kasutamise kuupäev: 29. 04 2017. a., allikas Microsoft Azure: <https://azure.microsoft.com/en-us/services/>
- Azure, M. (kuupäev puudub). *Microsoft Imagine Azure Offer*. Kasutamise kuupäev: 29. 04 2017. a., allikas Microsoft Azure: <https://azure.microsoft.com/en-us/offers/ms-azr-0144p/>
- Basak, A. (14. 10 2015. a.). *Exploring the Microsoft Azure IoT Hub*. Kasutamise kuupäev: 26. 04 2017. a., allikas DZone IoT: <https://dzone.com/articles/exploring-the-microsoft-azure-iot-hub>
- Basics, C. (kuupäev puudub). *Make an Arduino Temperature Sensor (Thermistor Tutorial)*. Kasutamise kuupäev: 19. 04 2017. a., allikas Circuit Basics:
<http://www.circuitbasics.com/arduino-thermistor-temperature-sensor-tutorial/>
- BlueCastle. (17. 02 2016. a.). *Python MQTT connection to Azure Iot Hub*. Kasutamise kuupäev: 25. 04 2017. a., allikas Stack Overflow:
<http://stackoverflow.com/questions/35452072/python-mqtt-connection-to-azure-iot-hub>

- Calbimonte, D. (21. 09 2015. a.). *Working with Visual Studio and SQL Azure databases*. Kasutamise kuupäev: 23. 04 2017. a., allikas SQLShack:
<https://www.sqlshack.com/working-with-visual-studio-and-sql-azure-databases/>
- Callaway, R. (02. 11 2015. a.). *A Brief History of Cloud-Based Integration in Microsoft Azure*. Kasutamise kuupäev: 27. 04 2017. a., allikas QuickLEarn:
<http://www.quicklearn.com/blog/2015/11/02/historyofthecloud/>
- Center, W. D. (kuupäev puudub). *Azure data uploader*. Kasutamise kuupäev: 20. 04 2017. a., allikas Windows Dev Center: <https://developer.microsoft.com/en-us/windows/iot/samples/azuredatauploader>
- codebender_cc. (kuupäev puudub). *How to Use DHT-22 Sensor - Arduino Tutorial*. Kasutamise kuupäev: 26. 04 2017. a., allikas Instructables:
<http://www.instructables.com/id/How-to-use-DHT-22-sensor-Arduino-Tutorial/>
- DigiCert. (kuupäev puudub). *DigiCert Trusted Root Authority Certificates*. Kasutamise kuupäev: 28. 04 2017. a., allikas DigiCert: <https://www.digicert.com/digicert-root-certificates.htm>
- faister. (26. 11 2014. a.). *How to send sensor data from Raspberry Pi to Azure Event Hub using a Python script*. Kasutamise kuupäev: 21. 04 2017. a., allikas Faister, say what?:
<http://blog.faister.com/how-to-send-sensor-data-from-raspberry-pi-to-azure-event-hub-using-a-python-script/>
- fr1tz. (27. 09 2016. a.). *'pip' is not recognized as an internal or external command*. Kasutamise kuupäev: 28. 04 2017. a., allikas Stack Overflow:
<http://stackoverflow.com/questions/23708898/pip-is-not-recognized-as-an-internal-or-external-command>
- hardillb. (3. 05 2016. a.). *python mqtt script on raspberry pi to send and receive messages*. Kasutamise kuupäev: 25. 04 2017. a., allikas Stack Overflow:
<http://stackoverflow.com/questions/37006863/python-mqtt-script-on-raspberry-pi-to-send-and-receive-messages>
- Hilscher, K. (17. 07 2016. a.). *Azure IoT Swimming Pool*. Kasutamise kuupäev: 17. 04 2017. a., allikas Arduino Project Hub: <https://create.arduino.cc/projecthub/khilscher/azure-iot-swimming-pool-f1a68c>

- Hoyt, K. (14. 11 2014. a.). *Circuit Friday: DHT22*. Kasutamise kuupäev: 27. 04 2017. a., allikas Kevin Hoyt: <http://www.kevinhoyt.com/2014/11/14/circuit-friday-dht22/>
- Jayakody, P. (02. 03 2016. a.). *Processing IoT Device Data with Azure and Power BI* . Kasutamise kuupäev: 25. 04 2017. a., allikas Hackster.io: <https://www.hackster.io/peejster/processing-iot-device-data-with-azure-and-power-bi-c00e35>
- JeffDuzak. (17. 02 2016. a.). *Function 'FORMAT' is not supported in DirectQuery mode*. Kasutamise kuupäev: 28. 04 2017. a., allikas Microsoft Power BI: <https://community.powerbi.com/t5/Desktop/Function-FORMAT-is-not-supported-in-DirectQuery-mode/td-p/19356>
- Jr, R. B., & teised. (30. 06 2015. a.). *Introducing Microsoft Azure*. Kasutamise kuupäev: 27. 04 2017. a., allikas Microsoft Azure: <https://docs.microsoft.com/en-us/azure/fundamentals-introduction-to-azure>
- Kess, B., & teised. (30. 03 2017. a.). *Install Visual Studio and SSDT for SQL Data Warehouse*. Kasutamise kuupäev: 27. 04 2017. a., allikas Microsoft Azure: <https://docs.microsoft.com/en-us/azure/sql-data-warehouse/sql-data-warehouse-install-visual-studio>
- Kushner, D. (26. 10 2011. a.). *The Making of Arduino*. Kasutamise kuupäev: 28. 04 2017. a., allikas IEEE Spectrum: <http://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino>
- Light, R. (01. 02 2015. a.). *paho-mqtt 1.1*. Kasutamise kuupäev: 24. 04 2017. a., allikas Python: <https://pypi.python.org/pypi/paho-mqtt/1.1>
- Loogman, O. (11. 10 2014. a.). *The Internet of Things with Arduino, Azure Event Hubs and the Azure Python SDK* . Kasutamise kuupäev: 21. 04 2017. a., allikas Kloud Blog: <https://blog.kloud.com.au/2014/10/11/the-internet-of-things-with-arduino-azure-event-hubs-and-the-azure-python-sdk/>
- Mackes, M. (30. 09 2015. a.). *Pool Controller*. Kasutamise kuupäev: 10. 04 2017. a., allikas Arduino Project Hub: https://create.arduino.cc/projecthub/mmackes/pool-controller-8dfa69?ref=tag&ref_id=relays&offset=0

- Maxey, K. (25. 02 2016. a.). *How a Coke Machine and the Industrial Internet of Things Can Give Birth to a Planetary Computer*. Kasutamise kuupäev: 30. 04 2017. a., allikas Engineers Rule: <http://www.engineersrule.com/how-a-coke-machine-and-the-industrial-internet-of-things-can-give-birth-to-a-planetary-computer/>
- Munro, E. (13. 09 2015. a.). *Sending our Temperature data to the cloud for Power BI*. Kasutamise kuupäev: 27. 04 2017. a., allikas GCITS: <http://gcits.com.au/sending-our-temperature-data-to-the-cloud-for-power-bi/>
- Nordrum, A. (18. 08 2016. a.). *Popular Internet of Things Forecast of 50 Billion Devices by 2020 Is Outdated*. Kasutamise kuupäev: 27. 04 2017. a., allikas IEEE Spectrum: <http://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated>
- npm. (kuupäev puudub). *npm install iotHub-explorer*. Kasutamise kuupäev: 23. 04 2017. a., allikas npm: <https://www.npmjs.com/package/iotHub-explorer>
- Oolep, K. (2013). *Jälgimissüsteemi automatiseerimine GSM/GPRS ühenduse kaudu*. Kasutamise kuupäev: 23. 04 2017. a., allikas Eesti Ettevõtluskõrgkool Mainor: <https://is.eek.ee/download.php?t=kb&dok=p17gb9fgnl105018bsomvmqf1fihb.pdf>
- Rouse, M. (07 2016. a.). *Internet of Things (IoT)*. Kasutamise kuupäev: 29. 04 2017. a., allikas IoT Agenda: <http://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT>
- Saxton, A. (12. 04 2017. a.). *Self-service sign up for Power BI*. Kasutamise kuupäev: 27. 04 2017. a., allikas Power BI: <https://powerbi.microsoft.com/en-us/documentation/powerbi-service-self-service-signup-for-power-bi/#power-bi-pro-60-day-trial>
- Sean. (20. 05 2016. a.). *Best method to get a single value answer from a table*. Kasutamise kuupäev: 28. 04 2017. a., allikas Microsoft Power BI: <https://community.powerbi.com/t5/Desktop/Best-method-to-get-a-single-value-answer-from-a-table/td-p/37285>
- Sparkman, M. (06. 03 2017. a.). *Create reports optimized for the Power BI phone apps*. Kasutamise kuupäev: 26. 04 2017. a., allikas Microsoft Power BI: <https://powerbi.microsoft.com/en-us/documentation/powerbi-desktop-create-phone-report/>

ThingLabs.io. (kuupäev puudub). *Lab02: Setting Up an Azure IoT*. Kasutamise kuupäev: 20. 04 2017. a., allikas ThingLabs.io: <http://thinglabs.io/workshop/cs/nightlight/setup-azure-iot-hub/>

Whitehouse, R. (08. 05 2016. a.). *Building an Azure IoT Demo Device with Raspberry Pi*. Kasutamise kuupäev: 25. 04 2017. a., allikas RPW Consulting: <http://www.rpw-consulting.com/building-an-azure-iot-demo-device-with-raspberry-pi/>

Vokkarne, R., & teised. (31. 01 2017. a.). *Send and receive messages with IoT Hub*. Kasutamise kuupäev: 24. 04 2017. a., allikas Microsoft Azure: <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-devguide-messaging#read-device-to-cloud-messages>

Võsandi, L. (31. 10 2013. a.). *Arduino Uno*. Kasutamise kuupäev: 28. 04 2017. a., allikas Lauri's blog: <https://lauri.võsandi.com/arduino/arduino-uno.html>

Lisad

Lisa 1. Arduino kood

```
#include "DHT.h"

#define DHTPIN 5      // mis digitaalset väljundit kasutatakse
#define DHTTYPE DHT22  // DHT 22  (AM2302), AM2321
DHT dht(DHTPIN, DHTTYPE);

void setup() {
    Serial.begin(9600);
    dht.begin();
}

void loop() {
    // Loeb õhuniiskust
    float h = dht.readHumidity();
    // Loeb temperatuuti (vaikimisi Celsiuses)
    float t = dht.readTemperature();
    Serial.print(h);
    Serial.print(" ");
    Serial.println(t-4);
    // Ootab 5 sekundit enne järgmist mõõtmist.
    delay(5000);
}
```

Lisa 2. Pythoni kood

```
from paho.mqtt import client as mqtt
import serial
import datetime
import time

ser = serial.Serial('COM3', 9600)

def on_connect(client, userdata, flags, rc):
    print "Connected with result code: %s" % rc

client.subscribe("devices/Arduino01/messages/servicebound/feed
back")

def on_disconnect(client, userdata, rc):
    print "Disconnected with result code: %s" % rc

def on_message(client, userdata, msg):
    print " - ".join((msg.topic, str(msg.payload)))
    client.publish("devices/Arduino01/messages/events",
"REPLY", qos=1)

def on_publish(client, userdata, mid):
    print "Sent message "
```



```

client = mqtt.Client("Arduino01", mqtt.MQTTv311)

client.on_connect = on_connect

client.on_disconnect = on_disconnect

client.on_message = on_message

client.on_publish = on_publish

client.username_pw_set(username="ArduinoIoT.azure-
devices.net/Arduino01",

                        password="SharedAccessSignature sr=ArduinoIoT.azure-
devices.net%2Fdevices%2FArduino01&sig=Jmq1OdYNMRmVQTI
EFurzUQbLgj05FYKa66uYnj2U9EU%3D&se=1524574065")

client.tls_set("sertifikaat.cer")

client.connect("ArduinoIoT.azure-devices.net", port=8883)

ser.flushInput()

while True:

    ts = time.time()

    temp = ser.readline()

    data = temp.split(" ")

    client.publish("devices/Arduino01/messages/events/",
                  "{ 'TEMP': '"+data[1]+'', 'NIISKUS': '"+data[0]+'',
                  'AEG': '"+datetime.datetime.fromtimestamp(ts).strftime('%d.%m.%
Y %H:%M:%S')+'"}")

    #time.sleep(5)

```