**iSPAM**

**IoT Smart Postal Announcement Message**



Tekninen raportti

INTIM17A6

2019

Alix Kalliojärvi

Miska Kivioja

Kimmo Koivusalo

Jonne Laitinen

SISÄLLYS

1 JOHdanto 1

2 toimintaperiaate 2

2.1 Käytetyt järjestelmät 2

2.2 Toimintaperiaatteen kuvaus 3

3 tiedonsiirron tekninen toteutus 5

3.1 Tiedonsiirto Arduino MKR1000 5

3.2 Tiedonsiirto ohjelmallisesti 6

3.3 Tiedonsiirto NodeRed ja Watson 7

3.3.1 iSpam Notification Control -sovellus 7

3.3.2 Sovelluksen prosessit ja toimintakuvaus 8

4 TULOKSET 9

5 YHTEENVETO 10

lähteet 11

Liitteet

1. JOHdanto

Älypostilaatikko, iSPAM – IoT Smart Post Announcement Message, on konsepti, joka toteutettiin HAMK:n Älykkäät Järjestelmät opintokokonaisuuden harjoitustyönä keväällä 2019.

Harjoitustyön määrityksenä oli tuotesuunnittelutyönä ideoita “ällistyttävä robotti” TiES:n organisoimaan kilpailuun. Harjoitustyössä tuli toteuttaa vähintään yksi toiminto ideoidusta robotista ja lisäksi kytkeä toiminto hyväksikäyttämään liikennöintiä verkon yli, sekä hyödyntämään anturitiedon siirtoa ja päätöksentekoa IoT-alustassa. Tuoteideaksi valittiin älypostilaatikko, jonka peruskonsepti on esitetty kuvassa 1.

Kuva, joka sisältää kohteen sisä, taivas

Kuvaus luotu automaattisesti

1. Älypostilaatikko iSPAM konseptin idea.

Moduulin harjoitustyössä iSPAM -älylaatikosta toteutettiin liiketunnistus, jonka teknisenä ratkaisuna käytettiin luminanssi-ilmaisua. Tehtävänannon mukaisesti iSPAM -älypostilaatikko liitettiin IBM Watson IoT –alustaan, jonka avulla toteutettiin päätöksenteko ja käskyjen välittäminen älylaatikolle. Lopputuloksena iSPAM havaitsee laatikon kannen aukaisun ja kykenee erillisten määritysten mukaisesti päätöksentekoon ja kommunikointiin älylaatikon käyttäjän kanssa, sekä ilmaisemaan postinjakajalle äänimerkillä postin vastaanoton. Tarkempi tekninen toimintaperiaatteen kuvaus on esitetty luvussa 2.

Laite- ja ohjelmointiympäristönä käytettiin Arduinoa ja sen omaa ohjelmointiympäristöä, joka pohjautuu C ja C++ ohjelmointikieleen. Älypostilaatikon liittäminen verkkoon ja sen etäohjaus toteutettiin IBM Watson Cloud – pilvipalvelun IoT alustan ja NodeRed:n avulla.

1. toimintaperiaate
   1. Käytetyt järjestelmät

Älypostilaatikon tekninen toteutus voidaan jakaa seuraaviin osiin: Komponentit (Arduino), Komponenttien ohjelmointi (Arduino) sekä verkkotoiminnallisuus ja sen parametrointi. Älypostilaatikko iSPAM:n toteutettiin käytännössä Arduino MKR1000 kehityskortin, Pmod ALS –valosensorin, PMod AMP2 –vahvistimen sekä MKR SD Proto Shield – muistikortinlukijan avulla.

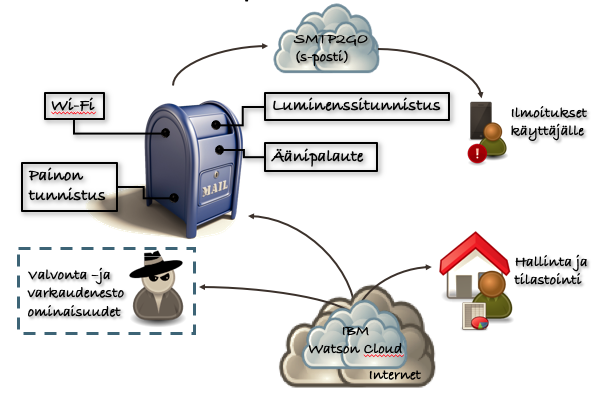


1. Arduino MKR 1000

Komponenttien ohjelmointi toteutettiin Arduinon omassa kehitysympäristössä. Ohjelmallisesti toteutuksessa tukeuduttiin Arduinon valmiisiin kirjastoihin ja esimerkkeihin. Ohjelmointikielenä Arduinossa käytetään C ja C++ johdannaista ohjelmointikieltä, jonka rakenne voidaan jakaa kolmeen osaan; funktioihin, muuttujiin ja varsinaiseen sovelluksen runkoon. Sovelluksen runko on jaettu kahteen osaan; Setup ja loop. Setup:ssa suoritetaan ohjelman perusalustukset, jotka suoritetaan kertaalleen, aina käynnistettäessä laitetta ja loop:ssa varsinainen laitteen toimintaan liittyvät ja toistuvat toimenpiteet. (Arduino, 2019)

Älypostilaatikon liittämiseen verkkoon käytettiin IBM Watson Cloud IoT alustaa, jonka ratkaisu koostuu useista komponenteista, joidenka yhteistoiminnalla saadaan aikaan yhtenäinen ja luotettava pilvipalvelu. IoT-alustan avulla voidaan mm. vuokrata palvelimia, toteuttaa virtuaalikonetoimintoja, hallinnoida sovellusten ja applikaatioiden toimintaa, analysoida ja hallita käytössä olevaa infrastruktuuria. (IBM Watson Cloud IoT Platform, 2019)

NodeRed on selainpohjainen ohjelmointityökalu laitteiden, sovellusliittymien ja online-palvelujen yhteen liittämiseen esimerkiksi IBM Watson Cloud IoT alustan kanssa. Yksi NodeRed:n eduista on graafinen käyttöliittymä, jolla erilaisten toimintojen käskytys ja ohjelmointi sekä tietovirtojen jäsentely laitteiden ja pilvipalvelun välillä on helppoa. (NodeRed, 2019)

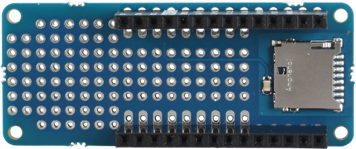


1. iSPAM älypostilaatikon eri järjestelmät.
   1. Toimintaperiaatteen kuvaus

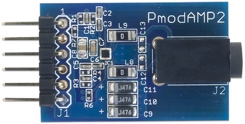
Älypostilaatikon alustana toimii Arduino MKR 1000 -mikrokontrolleri

johon muu elektroniikka on liitetty. Arduino MKR 1000 on toiminnallisuuksiltaan Arduino Zeroa vastaava mikro -ohjain johon lisätty WLAN-tekniikkaa tukeva langaton tiedonsiirtomoduuli. (Arduino, 2019)

Mikro-ohjaimeen on liitetty MKR SD Proto Shield -muistikortinlukija, johon liitetylle muistikortille älypostilaatikko taltioi lokitiedot laitteen ollessa käytössä. Lisäksi muistikortille on taltioitu laatikon käyttöön asetetut äänitiedostot. Äänitiedostojen audio ajetaan laatikon käyttäjälle kuuloaistivasteeksi laatikon sisältämän ulkoisen äänilähteen kautta, jota varten kokonaisuus sisältää Pmod AMP2- digitaalisen audiovahvistimen. Vahvistinmoduuli sisältää tyypillisen 3,5mm:n jakkiliittimen, johon kaiutin on kytketty. (Arduino, 2019) (Digilent, 2019)

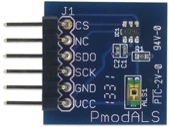


1. MKR SD Proto Shield.



1. Pmod AMP2.

Älypostilaatikon kannen sisäpuolelle kiinnitetty Pmod ALS-sensori on yksinkertainen, valoon reagoiva sensori, joka muuntaa vastaanottamansa valon luminanssin digitaaliseksi arvoksi. Luminanssiarvo määrittää käytännössä sen onko laatikon kansi auki vai kiinni. Auki ollessaan sensori saa tietyn raja-arvon ylittävän luminanssiarvon, jonka perusteella Arduinoon ohjelmoitu logiikka ryhtyy jatkotoimenpiteisiin. (Digilent, 2019)



1. Pmod ALS.

Valosensorin tuottama arvo raportoidaan IBM Watson-pilvipalveluun jatkuvalla syklillä. Laatikon avaamista indikoiva arvo jatkokäsitellään NodeRedissä. Laatikon avaamista indikoiva arvo toimii myös triggerinä merkkiäänelle, joka ajetaan audiovahvistimen ja kaiuttimen avulla käyttäjälle.

NodeRed tuottaa käskyt takaisin postilaatikolle, joiden tehtävänä on antaa laatikon omistajalle tietoa iSPAM:n käytöstä. NodeRed luo käskyn saapuneesta postista ja julkaisee sen MQTT-protokollaa hyödyntäen Arduinolle takaisin. Ilmoitus jaetaan käyttäjälle Arduinon saatua käskyn NodeRediltä, jolloin Arduino lähettää sähköpostin SMTP2GO -sähköpostipalvelimen kautta laatikon omistajalle.

NodeRed antaa laatikon omistajalle sähköpostitse myös varoitusilmoituksen auki jääneestä postilaatikosta. Lisäksi NodeRediin on luotu websocket -väylä älypostilaatikon http -palveluun, lähinnä jatkokehitystä ennakoiden.

SMTP2GO-palvelinyhteys on toteutettu Arduinon WLAN-väylää hyödyntäen. WLAN-yhteyden läpi saadaan myös NTP-palvelimelta aikatieto järjestelmän lokia varten, sekä tietysti laitekokonaisuuden tiedonsiirtoon MQTT-protokollalla.

Kuvassa 7 on esitetty iSPAM -älypostilaatikon käynnistysvaiheessa suoritettavat alustukset ja kuvassa 8 esitetään toimintalogiikka prosessikaavion avulla. Tiedonsiirto Arduinon logiikan, IBM Watson Cloud IoT-alustan, ja NodeRedin kesken on kuvattu tarkemmin luvussa 3.

Kuva, joka sisältää kohteen näyttökuva

Kuvaus luotu automaattisesti

1. MKR1000 Setup logiikka.

Kuva, joka sisältää kohteen näyttökuva, rakennus, merkki, ulko

Kuvaus luotu automaattisesti

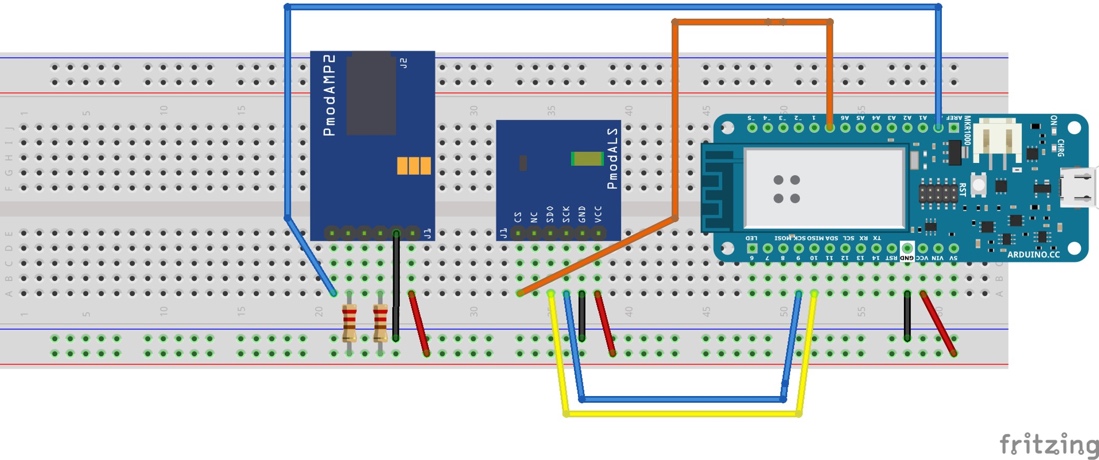
1. Älypostilaatikon toimintaperiaate.
2. tiedonsiirron tekninen toteutus

iSPAM-älypostilaatikon tiedonsiirtoratkaisut käsittävät digitaalisen tiedonsiirron Arduinon ja siihen liitettyjen moduulien välillä, tiedonsiirron laitteen ja pilvipalvelun välillä MQTT-protokollalla, ohjauksen NodeRed/Cloud-ympäristössä, sekä liikenteen älypostilaatikon ja laatikon omistajan välillä SMTP-protokollaa hyödyntäen.

Älypostilaatikon ohjauslogiikka on laadittu Arduinon omassa IDE:ssa seuraavia kirjastoja hyödyntäen.

* + <SPI.h>
  + <SD.h>
  + <AudioZero.h>
  + <WiFi101.h>
  + <MQTT.h>
  + <MQTTClient.h>
  + <NTPClient.h>
  + <WiFiUdp.h>
  1. Tiedonsiirto Arduino MKR1000

Arduinoon liitetyt moduulit on kytketty alla olevan kuvan periaatteen mukaisesti.



1. Arduinon kytkennät

ALS-valosensori ja muistikortinlukija hyödyntää SPI-muotoista sarjaliitäntää ja AMP2-vahvistin on kytketty analogiseen lähtöön. Verkkoyhteyteen käytetään MKR1000:n omaa WiFi -piiriä.

* 1. Tiedonsiirto ohjelmallisesti

Ennen varsinaista tiedonsiirtoa laite yhdistää ennalta määritettyyn langattomaan verkkoon käynnistyessään. Jos langaton verkkoyhteys onnistuu, laite autentikoi yhteyden IBM Watson Cloud IoT Platformin kanssa. Tiedonsiirto Watsonin ja laitteen välillä on salaamatonta, mutta autentikointiin käytetään laitekohtaista auth ID:tä.

Varsinainen tiedonsiirto alkaa yhteyksien muodostuttua. Viestiliikenne laitteen ja pilven välillä on toteutettu MQTT-protokollalla käyttämällä Arduinon MQTT-kirjastoa. Ohjelma tilaa IBM Watson Cloud IoT-alustan MQTT-topicin joka on tarkoitettu laitteen ohjaamiseen ("*iot-2/cmd/+/fmt/json*") ja alkaa julkaisemaan event-topicissa mittaamiaan tietoja pilveen. Julkaistavia tietoja ovat valon määrä (luminenssi), laatikon tila (kiinni/auki) ja langattoman verkon kuuluvuus (RSSI).

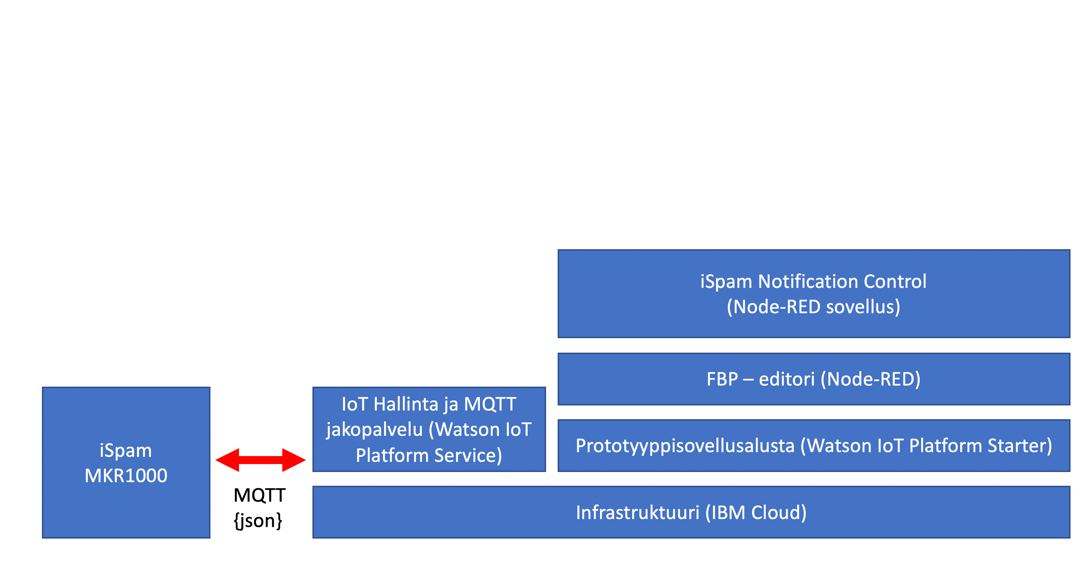
Laitteen ja käyttäjän välinen sähköposti on toteutettu käyttämällä SMTP2GO-sähköpostipalvelinta. Sisällöltään ennalta määrätyt sähköpostit lähetetään pilveltä tulevalla käskyllä. Käskyt ovat MQTT-viestejä, joissa itse käsky on topicissa ja payload on tyhjä. Esimerkiksi MQTT-viesti, jossa topic on “*iot-2/cmd/sendmail/fmt/json”* johtaa sähköpostin lähettämiseen käyttäjälle, jossa ilmoitetaan postilaatikon avautuneen.

Äänipalaute on muistikortilta luettava WAV-äänitiedosto. Äänen toistamiseen käytetään AudioZero –kirjastoa. Äänitiedoston on oltava WAV-muodossa määritelmillä 88200 Hz, 8-bit, unsigned PCM mono.

* 1. Tiedonsiirto NodeRed ja Watson

Arduino MKR1000 -mikro-ohjain on yhdistetty IBM:n Watson Cloud IoT alustaan. IoT-alusta on yhteydessä IBM:n pilvipalvelussa olevaan NodeRed-sovellukseen. Viestiliikenne ohjaimen ja sovelluksen välillä on MQTT-protokollalla välitettyä json-muotoista dataa.

IBM Watson Cloud IoT Platform on mikro-ohjainlaitteiden hallinta-alusta. Alustalla on mahdollista hallita iot-laitteiden yhteyksiä, tietovirran tallentamista ja analysointia sekä laitteiden valvontaa. Alustaan on mahdollista yhdistää sovelluksia, joilla iot-laitteilta saatavaa tietoa voidaan käsitellä, jalostaa ja esittää. Kuvassa 10 on esitelty iSPAM-sovelluksessa käytettävät Watson-alustan komponentit.



1. IBM Watson Cloud IoT-ympäristö iSPAM-sovelluksessa.

iSPAM:n mikro-ohjain yhdistetään TCP/IP -protokollaa käyttäen IBM Watson Cloudiin. Ohjain lähettää alustalle MQTT-protokollaa käyttäen json-muodossa olevaa tietoa postilaatikon kannen tilasta, luminenssiarvosta ja WLAN-verkon signaalin voimakkuudesta.

* + 1. iSPAM Notification Control -sovellus

Pilvipalvelussa, jossa IBM Watson Cloud IoT-alusta on, on myös prototyyppisovellusalusta Watson IoT Platform Starter. Sovellusalustalle on rakennettu iSPAM Notification Control -sovellus (myöhemmin iSNC). Sovelluksen tehtävänä on valvoa postilaatikon kannen tilaa ja ohjata postilaatikkoa lähettämään ilmoituksia sähköpostitse käyttäjälle. Sovellus on tehty FBP-editori Node-RED:llä.

FBP (flow-based programming) on ohjelmointiparadigma, jossa syötteenä saatu tieto prosessoidaan verkotettujen, peräkkäisten, haaroittuvien ja rinnakkaisten prosessien toimesta. Niiden väliset yhteydet ovat riippumattomia prosessien toiminnasta. FBP-verkkoa on helppo muokata ja prosesseja uudelleen verkottamalla voidaan luoda uusia sovelluksia ilman tarvetta muokata itse prosesseja. (Wikipedia 2019)

Kun laatikon avaustapahtuma vastaanotetaan iSNC:lle, se antaa käskyn laatikon ohjaimelle lähettää sähköposti käyttäjälle laatikon aukaisusta. Sovellus alkaa samalla hetkellä laskemaan ennalta asetettua aika-arvoa. Jos tämän ajan kuluttua loppuun ohjain lähettää edelleen kansi auki -tapahtumaa, sovellus käskee ohjainta lähettämään varoitussähköpostin käyttäjälle auki unohtuneesta kannesta.

* + 1. Sovelluksen prosessit ja toimintakuvaus

iSPAM event -prosessi vastaa yhteyden muodostamisesta IBM Watson Cloud IoT-alustalla olevaan MKR1000-laitteeseen. Yhteys suojataan käyttämällä sovelluskohtaista avainta (API-key). Prosessi kuuntelee laitteen kaikkia, json-muodossa olevia tapahtumia ja välittää ne eteenpäin. iSPAM-sovellus käyttää toimitusvarmuustasoa (QoS) nolla kaikessa tietoliikenteessä.

SendMail\_OpenClose -prosessi lukee MKR1000:n lähettämästä paketista laatikon kannen tilatietoa. Prosessi on silmukka, joka suoritetaan aina viestin vastaanoton yhteydessä. Silmukan lopussa se lähettää json-paketin, jossa on ohjeet liipaisuprosessille lähettää joko avausilmoitus tai varoitus.

Prosessi hakee muistista edellisen suorituksen aikana tallennetun tiedon kannen tilasta sekä tiedot onko postin saapumisilmoitusta tai avoimen kannen varoitusta lähetetty, muuttujiin. Prosessi tallentaa myös silmukan suorituksen alkuajankohdan muuttujaan.

Prosessin vastaanottamaa kannen tilatietoa verrataan kolmea ehtolausetta vastaan. Jos vastaanotettu tilatieto on yksi (1) ja muistissa oleva epätosi, asetetaan hälytysajankohta muuttujaan lisäämällä suorituksen alkuaikaan ennalta määrätty hälytysraja. Kannen tilatieto tallennetaan muistiin ja avausilmoitus asetetaan tosi-tilaan.

Jos vastaanotettu tilatieto on yksi (1) ja muistissa oleva tosi otetaan edellisen avauksen ajankohta muistista ja verrataan sitä käynnissä olevan silmukan suoritusajankohtaan. Mikäli edellisestä aukaisusta on kulunut hälytysrajan mittainen aika, asetetaan varoitus tosi-tilaan.

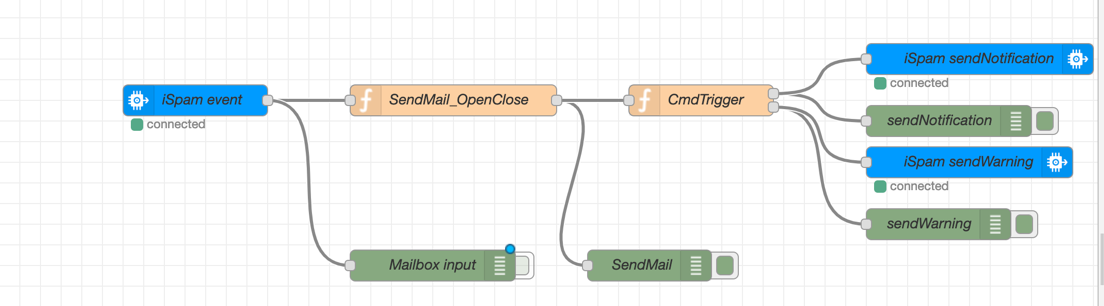
Jos vastaanotettu tilatieto on nolla (0) ja muistissa oleva tosi tallennetaan muistiin tilatiedon muutos.

CmdTrigger -prosessi lukee edellisen prosessin lähettämän ohjepaketin. Jos avausilmoitus on tosi-tilassa, se lähettää viestin sisällön ensimmäisestä lähdöstä. Jos varoitus on tosi-tilassa se lähettää viestin sisällön toisesta lähdöstä. Muussa tapauksessa prosessi ei lähetä mitään.

iSPAM sendNotification ja iSPAM sendWarning prosessit muodostavat iSPAM event -prosessin tavoin MKR1000 -laitteelle. Prosessit lähettävät MQTT-komentoviestin CmdTrigger -prosessin ohjaamana. Ensimmäinen prosesseista on yhteydessä ohjaimen *sendmail* -topiciin, joka ohjaa ilmoituspostin lähettämistä. Toinen on yhteydessä *sendWarning* -topiciin, joka ohjaa varoituspostin lähettämistä.

Mailbox input, SendMail, sendNotification ja sendWarning -prosessit ovat vianselvitysprosesseja jotka vastaanottavat tietovirran edelliseltä prosessilta ja tulostavat sen konsolinäytölle.

Kuvassa 11 on sovelluksen verkkokuva ja jokaisen prosessin lähdekoodi sekä yhteysprosessien asetukset ovat raportin liitteessä 1.



1. iSPAM Notification -sovelluksen FBP-verkkokuva
2. TULOKSET

Älypostilaatikon tekniseksi toteuttavaksi toiminnoksi valittiin liikkeentunnistus luminenssin avulla. Käytännössä valitulla tekniikalla havaitaan laatikon kannen aukaisu ja sulkeutuminen. Valoanturin toiminta kyettiin toteuttamaan halutulla tavalla ja laite saatiin suorittamaan halutta ilmaisu valon määrän muutoksella.

Arduinoon ohjelmoitu logiikka toimii erinomaisesti käyttötarkoitukseensa. Sensoreilta tuleva data sietää ohjelmallisesti häiriöitä riittävällä tasolla, joka luo pohjaa älylaatikon toimintavarmuudelle. Sensorien tiedonsiirto Arduinolle toimii luotettavasti. Valittu pilvipalveluratkaisu, ohjaustoiminnot, ja sähköpostipalvelin ovat kolmannen osapuolen tuottamia palveluja, jotka ovat universaalisti käytössä, ja täten varsin toimintavarmoja ratkaisuja älypostilaatikon tarpeisiin. Laatikon operatiivisessa käytössä ajettava ohjelmasilmukka on todettu testiajoissa hyvin soveltuvaksi prosessiksi älylaatikon tiedonsiirron perustalle.

IBM Watson Cloud IoT-alustaan ja NodeRediin ohjelmoidut toiminnallisuudet toteuttavat tiedonsiirron ja tiedonkäsittelyn halutulla tavalla. Yhteys älypostilaatikkoon toimii edellä kuvatulla tavalla, ja Arduinolle annettavat komennot toimivat tarkoituksenmukaisesti niitä tiedonsiirtomenetelmiä käyttäen, mitä alussa valittiin älypostilaatikon käyttöön.

1. YHTEENVETO

Harjoitustyön suunnittelun ja toteutuksen aikana IoT ympäristö ja sen tarjoamat mahdollisuudet erilaisten esineiden, elementtien ja toimintojen liittämisestä internettiin tulivat hyvin esiin. Omaperäisen ja ainutlaatuisen, erityisesti ällistyttäväksi robotiksikin luonnehdittavan, aiheen keksiminen oli kuitenkin haastavaa. Melko laajalla otannalla tarkasteltaessa ensivaikutelmaksi jäi, että nykypäivänä miltei kaikki asiat, esineet tai toiminteet ovat jo jollain tavalla liitettynä verkkoon. Tai ainakin joku on jo sitä kokeillut. Aiheen valinnassa pyrittiin löytämään sovellus, jota voisi ajatella käytettävän ja jonka tuotekehitys voitaisiin viedä käytäntöön asti todellisena kaupallisena laitteena.

Toteutettavaksi konseptiksi valittiin älypostilaatikko. Älypostilaatikolle määriteltiin ensisijaiset toiminteet, joita olivat ääni-ilmaisu, kannen aukaisun tunnistaminen sekä kommunikointi omistajan kanssa sähköpostilla.

Aiheen valinta tehtiin moduulin alkuvaiheessa ja tarkasteltaessa koko moduulin opintosisältöä verrattuna tehtyyn ryhmätyöhön, koettiin aiheen valinta onnistuneeksi. Kurssin opetus tuki sisällöllisesti ryhmätyötä, erityisesti NodeRed ja IBM Watson osuuksissa.

Ryhmätyön toteutuksen aikana koko ryhmälle syntyi kattava kuva siitä, miten IoT -ympäristö pääperiaatteellisesti toimii ja millaisia sovellutuksia sen avulla on tehty ja voitaisiin tehdä. Kurssin aikana opittiin uutta ja syvennettiin jo opittuja taitoja Arduino:sta ja Tinkercad:sta, NodeRed:stä, IBM Watson Cloudista:sta sekä yleisesti komponenteista, piirilevytekniikasta ja niiden suunnittelusta.

Älypostilaatikko rakennettiin ja toteutettiin fyysisesti tekniikan osalta toimivaksi. Rakenteellisesti älypostilaatikosta jätettiin varsinainen ”postilaatikko” pois ja toteutuksessa keskityttiin teknisen ratkaisun toimivuuteen. Käytännön toteutuksen tekeminen Arduino:lla ja sen eri komponenteilla oli merkittävästi opettavaisempaa kuin Tinkercad:lla simuloituna. Teknisenä toteutukseksi valittu liiketunnistus luminenssin avulla, sekä äänen toistaminen onnistui hyvin ja älypostilaatikko saatiin suorittamaan halutut toiminteet.

Arduino kehityskorttina käytetyn MKR1000 ohjelmointi toimimaan halutulla tavalla onnistui verrattain helposti laajojen kirjastojen, sekä erilaisten ohjesivustojen avulla. Ohjelmallisesti älypostilaatikon toiminnoista toteutettiin mm. lokitiedoston kirjoittaminen, äänitiedoston toistaminen, wifi-yhteys, sähköpostitoiminnallisuus sekä liittyminen IBM Watson Cloud IoT -alustaan. Ohjelmointiosuus tuki hyvin jo aiemmilta kursseilta opittua ohjelmointitaitoa.

IBM Watson Cloud IoT-alusta ja NodeRed vaiheessa kyettiin viemään kurssin aikana opittuja asioita käytäntöön konkreettisesti. Älypostilaatikko kyettiin liittämään verkkoon ja laatikon toimintojen perusteella voitiin tehdä analyysiä ja päätöksiä, jotka kyettiin välittämään takaisin MKR1000 kehityskortille toteutettavaksi. Esimerkkinä tällaisesta toiminnosta on mm. kannen avaamisesta ja/tai sen auki jäämisestä syntyvä sähköposti ja sen lähetyksen päätös.

Jatkotyöskentelynä tuotekehitystä voitaisiin jatkaa eteenpäin lisäämällä laatikkoon painoanturit saapuneen postimäärän todentamiseksi. Laatikon toimintaa voidaan melko yksinkertaisin toimin kehittää havaitsemaan eri postituskerrat sekä analysoimaan mahdollisia varkauksia esim. painon vähenemisenä. Käyttäjän näkökulmasta laatikon toimintojen muokkaaminen ja ohjaaminen verkkosivuston osalta olisi myös jatkokehiteltävissä laajemmaksi. Verkkosivujen kautta käyttäjä voisi määrittää esim. reagointi- ja kynnyksiä, sähköpostiviestin sisältöä, äänimerkkejä tai ääniviestejä postinkantajalle sekä tilastoitavia asioita. Yksityisyydensuojan ja tulevan tiedustelulain nyanssien vaikutukset videokuvan välittämiseen sekä live-kuva ja ääniyhteyden avaamiseen postinkantajan kanssa lienee syytä jättää toteutettavaksi myöhemmissä kehitysversioissa.

Kokonaisuudessaan ryhmätyössä opittiin valitun laitteen teknisen, ohjelmallisen ja toiminnallisen rakenteen toteuttaminen ja liittäminen IoT -alustaan. Tavoitteena ryhmätyössä oli valmistaa älypostilaatikko, joka havaitsee kannen aukaisun, kykenee ääni-ilmaisuun sekä kommunikointiin ja päätöksentekoon IoT -alustan avulla. Asetetut tavoitteet saavutettiin täysimääräisinä ja ryhmätyö todettiin erittäin onnistuneeksi ja mielekkääksi.

lähteet

Arduino (2019). <https://www.arduino.cc/reference/en/#page-title> Arduino IDE:n opas. Haettu 14.3.2019 osoitteesta https://arduino.cc

Arduino (2019). <https://store.arduino.cc/arduino-mkr1000> MKR 1000 tuotekuvaus. Haettu 14.3.2019 osoitteesta https://arduino.cc

Arduino (2019). <https://store.arduino.cc/mkr-sd-proto-shield> MKR SD Proto Shield tuotekuvaus. Haettu 14.3.2019 osoitteesta https://arduino.cc

Digilent (2019). <https://store.digilentinc.com/pmod-amp2-audio-amplifier/> AMP2-audiovahvistimen tuotekuvaus. Haettu 14.3.2019 osoitteesta https://store.digilentinc.com

Digilent (2019). <https://store.digilentinc.com/pmod-als-ambient-light-sensor/> ALS-valosensorin tuotekuvaus. Haettu 14.3.2019 osoitteesta https://store.digilentinc.com

IBM Cloud (2019). <https://console.bluemix.net/docs/overview/ibm-cloud-platform.html#whatis> IBM Cloud yleisesittely. Haettu 15.3.2019 osoitteesta <https://console.bluemix.net>

NodeRed (2019). <https://nodered.org/#features> NodeRed toiminnat. Haettu 15.3.2019 osoitteeta <https://nodered.org/>

Wikipedia (2019). <https://en.wikipedia.org/wiki/Flow-based_programming> Flow-Based Programming. Haettu 15.3.2019 osoitteesta <https://en.wikipedia.org>

LIITTEET

LIITE 1: iSNC lähdekoodi ja asetukset

LIITE 2: Arduinon lähdekoodi