



Ville Karppinen

Pill dispenser documentation

Pill dispenser

Metropolia University of Applied Sciences
Bachelor of Engineering
Information and Communication Technology
Embedded Systems Programming
12.12.2025

Abstract

Author(s):	Ville Karppinen
Title:	Pill dispenser documentation
Number of Pages:	11 pages
Date:	12 December 2025
Degree:	Bachelor of Engineering
Degree Programme:	Information and Communication Technology
Specialisation option:	Smart IoT Systems: Embedded IoT Devices
Instructor(s):	Keijo Länsikunnas, Responsible Supervisor

Contents

1	Johdanto.....	1
2	Projektin tavoitteet.....	1
3	Toimintaperiaatteet.....	2
3.1	Käynnistys.....	2
3.1.1	Laitteiston alustus.....	2
3.1.2	LoRaWAN-yhteyden tarkistus.....	3
3.1.3	Tilan alustus.....	3
3.2	Pääsilmukka (Main loop).....	4
3.2.1	ISR-tapahtumien käsittely.....	4
3.3	Tilakone: run_sm().....	4
3.3.1	Kalibrointi.....	5
3.3.2	Pillerin putoamisen käsittely.....	6
3.3.3	Virran katkaisun käsittely.....	7
4	Järjestelmän arkkitehtuuri.....	7

1 Johdanto

Tämä projekti toteuttaa sulautettuihin järjestelmiin perustuvan pillerinannostelijan, joka sisältää Raspberry Pi Pico -ohjainpiirilevyn, EEPROM-muistin, LoRaWAN-moduulin, askelmoottorin ja sen ohjaimen sekä annostelijan mekaaniset osat ja sensorit (optinen sensori ja piezosensori). Projektin on toteutettu Metropolitan Embedded Systems Programming -kurssilla.

Laitteen tarkoitus on annostella pilleri automaattisesti ennalta määritellyin väliajoin (projektissa 30 sekuntia), rekisteröidä annostelun onnistuminen piezosensorilla ja tallentaa laitteen tila EEPROM-muistiin. Tämän lisäksi annostelija välittää LoRaWAN-verkon kautta tilamuutokset, kuten käynnistymisen, kalibroinnin, virran katkeamisen askelmoottorin pyöriessä, onnistuneen tai epäonnistuneen pillerin annostelun sekä pillerien täytön niiden loputtua.

Automatisoidun pillerinannostelijan tavoitteena on tukea käyttäjiä, kuten muistisairaita, jotka saattaisivat muuten unohtaa tai sekoittaa lääkityksensä, ja siten parantaa turvallisuutta ja elämänlaatua. Järjestelmä vähentää myös hoitohenkilökunnan kuormitusta, koska annostelua ja seurantaa voidaan valvoa etänä.

2 Projektin tavoitteet

Projektiin tavoitteena oli tehdä pilleriannostelija, josta löytyy ainakin seuraavat ominaisuudet:

- Annostelijan käynnistyessä odotetaan painikkeen painallusta ja vilkutetaan LED-valoa.
- Kun nappia painetaan, annostelija kalibroidaan käänämällä pyörää vähintään yksi täysi kierros niin, että anturin aukon sisältävä lokero kohdistuu pudotusaukkoon.
- Kalibroinnin jälkeen ohjelma pitää LED-valon päällä, kunnes käyttäjä painaa nappia, joka aloittaa ajastetun pillerien annostuksen 30 sekunnin välein.
- Pyörän käentyessä piezosensoria käytetään havaitsemaan, annosteltiinko pilleri vai ei.
- Jos pilleriä ei havaita, LED-valo vilkkuu 5 kertaa.

- Kun kaikki pillerit on annosteltu (pyörä käännynt 7 kertaa), ohjelma alkaa alusta (odotus kalibroinnille jne.).
- Laite muistaa tilan käynnistysten ja virrankatkosten yli.
- Laite yhdistyy LoraWAN-verkkoon ja raportoi tilan muutoksista (käynnistys, pilleri annosteltu tai ei annosteltu, annostelija tyhjä, virrankatko moottorin pyöriessä jne.).
- Laite pystyy havaitsemaan, jos se on resetoitu tai virta on katkennut moottorin pyöriessä. Laite pystyy kalibroimaan itsensä automaattisesti virrankatkon jälkeen kesken pyörähdyksen ilman, että jäljellä olevia pillereitä annostellaan.

3 Toimintaperiaatteet

Tässä luvussa kuvataan pillereiden annosteluohjelmiston keskeiset toimintaperiaatteet. Luku käsittelee laitteen käynnistystä, tilojen alustusta, keskeisiä syötteitä ja ohjelmiston tilakonetta. Lisäksi kuvataan, miten järjestelmä reagoi erilaisiin tilanteisiin, kuten virran katkeamiseen tai epäonnistuneeseen pillerin pudotukseen.

3.1 Käynnistys

Käynnistysvaiheessa järjestelmä suorittaa kaikki välttämättömät laitteisto- ja ohjelmistotason alustukset.

3.1.1 Laitteiston alustus

Käynnistyksen alussa alustetaan kaikki laitteen fyysiset I/O-komponentit, jotta ohjelmisto voi käyttää niitä. Alustettavia komponentteja ovat:

- sarjaportti (stdio/UART).
- painikkeet
- LED-valot
- UART ja I2C
- askelmoottorin ohjauspinnit
- piezo- ja optofork-sensorit

3.1.2 LoRaWAN-yhteyden tarkistus

Seuraavaksi järjestelmä testaa, saadaanko LoRa-moduuliin yhteys UARTin kautta. Mikäli moduuli vastaa odotetusti, laite yrittää liittyä LoRaWAN-verkkoon. Jos LoRaWAN-verkkoon liittyminen onnistuu niin tilamuutokset voidaan tämän jälkeen lähetää LoRaWAN-verkon kautta.

- UART-yhteyden tarkistus.
 - Laite lähetää AT-komennon LoRa-moduulille varmistaakseen, että se vastaa odotetusti.
 - Moduulin version tarkistus.
 - DevEUI-tunnisteen tarkistus.
- Liittymisparametrien asettaminen.
 - Laite määrittää liittymiseen käytettävän AppKey:n.
 - Protokollaluokan: Class A.
 - Portin.
- OTAA (Over-The-Air-Activation)-liittymisyritys (5 yritystä).
 - Mikäli liittyminen onnistuu niin tilaviestit välitetään LoRaWAN-verkon kautta.
 - Mikäli liittyminen ei onnistu niin laite toimii offline-tilassa.

3.1.3 Tilan alustus

Muuttujat on tallennettu EEPROM-muistiin arvona ja arvon inversiona, jotta muistin eheys voidaan varmistaa. Tässä vaiheessa tarkistetaan EEPROM-muistin sisältämien tilojen eheys. Tämä määrittää aloittaako järjestelmä täysin alusta vai jatkaako se siitä, mihin toiminta ennen virran katkaisua jää:

- Mikäli EEPROM-tila on korruptoitunut niin kaikki keskeiset tilat alustetaan uudelleen ja laite siirtyy initial_state.
- Mikäli tila on validi niin tarkistetaan pyörökö askelmoottori virran katketessa.
 - Jos moottori pyöri niin kirjataan tämä lokiin muistiin EEPROM:in ja välitetään tieto LoRaWAN-verkon kautta. Tämän jälkeen käynnistetään joko kalibrointi tai uudelleenkalibrointi riippuen siitä pysähtyikö moottori kesken kalibroinnin tai kesken uudelleenkalibroinnin tai pillereiden annostelun.

- Jos moottori ei pyörinyt niin tarkistetaan kalibrointitala ja ladataan EEPROM:sta viimeisin tila.

3.2 Pääsilmukka (Main loop)

Pääsilmukka koostuu kahdesta osasta: ISR keskeyystapahtumien käsittelystä ja tilakoneen ajamisesta.

3.2.1 ISR-tapahtumien käsitteily

Keskeytysjono sisältää painikkeiden ja piezo-sensorin tapahtumat. Painalluksia käsitellään vain initial- ja idle-tiloissa, jotta muiden tilojen aikana ei voi keskeyttää prosessia.

- SW_R painettu:
 - Jos initial/idle-tilassa niin tulostetaan kaikki lokitapahtumat
- SW_M painettu:
 - Jos kalibroitu ja idle-tilassa niin siirrytään pill_dispensing-tilaan.
 - Muuten tulostetaan: "Calibrate first."
- SW_L painettu:
 - Jos initial/idle-tilassa niin siirrytään kalibrointiin.

3.3 Tilakone: run_sm()

Tilakone ohjaa koko ohjelman toimintaa. Jokaiselle tilalle on oma funktionsa, ja tilasta toiseen siirrytään next_state-function kautta.

next_state:

- Tallentaa uuden tilan EEPROM-muistiin.
- Päivittää nykyisen tilan.

Alla tiivistetty kuvaus tiloista:

- initial_state
 - Keskimmäinen LED-valo vilkkuu sen merkiksi, että laite odottaa kalibointia.
- calib_state
 - Askelmoottori pyörii vähintään yhden kokonaisen kierroksen verran, jotta saadaan steps_per_rev. Optofork tunnistaa vähintään kaksi reunaa. Tämän jälkeen tehdään realignment ja siirrytään idle-tilaan.
- idle_state
 - Odotetaan käyttäjän painavan nappia.
- pill_dispensing_state
 - Ajetaan moottoria yhden kierroksen verran ja tunnistetaan mahdollinen pillerin putoaminen piezo-sensorin avulla.
- waiting_pill_to_drop_state
 - Odotetaan pillerin putoamista, jos se ei pill_dispensing-tilassa kerennyt tapahtumaan.
- pill_not_dropped_state
 - Keskimmäinen LED-valo vilkkuu 5 kertaa sen merkiksi, ettei pilleri pudonnut.
- wait_turn_state
 - Odotetaan seuraavaa annostelukierrosta (30 sekuntia testissä).
- re_calib_state
 - Virrankatkaisun jälkeinen uudelleenkalibrointi.

Tilojen tapahtumat käydään tarkemmin läpi flowchartissa.

3.3.1 Kalibrointi

Kalibrointi perustuu optofork-sensorin toimintaan, joka tunnistaa kaksi laskevaa reunaa moottorin pyöriessä. Moottori pyörii vähintään yhden täyden kierroksen, jotta tarkka kokonaisaskelmäärä saadaan talteen stepper_motor.c-scriptin globaaliin staattiseen steps_per_rev-muuttujaan. Sitten kokonaisaskelmäärä tallennetaan EEPROM-muistiin, josta se voidaan ladata virran katkaisun jälken, jos sovellus on kalibroidussa tilassa. Tämän jälkeen suoritetaan vielä uudelleenkohdistusmenettely, jotta saadaan annostelupyörän pilleripaikka kohdistettua tarkasti annostelijan pohjan suuaukolle.

Rekalibroinnissa askelmoottori käännyy vastakkaiseen suuntaan. Tämä tila käynnistyy jos virta on katkennut kesken moottorin pyörimisen ja ollaan joko pillerin käsittely tilassa (pill_dispensing_st) tai rekalibrointtilassa (re_calib_st). Tämän jälkeen suoritetaan vielä

uudelleenkohdistusmenettely. En ollut projektin palautukseen mennessä huomioinut, että uudelleenkohdistusmenettely vaati eri askelmääärän toisesta suunnasta tullessa, mutta olen tämän korjannut jälkeenpäin. Uudelleen kohdistusmenettelyn jälkeen palataan vielä sille paikalle minne jäätii pillerien pudotuksessa, eli askelmoottori ottaa niin monta vuoroa vastaan eteenpäin, kuin niitä oli keretty tekemään pillerin annostelu tilassa.

3.3.2 Pillerin putoamisen käsittely

Pillerin putoamisen havainnointi perustuu piezosensorin toimintaan, joka havaitsee nopean mekaanisen väähdyksen ja reagoi siihen hetkellisenä jännitepiikkinä. Pillerin pudotus tunnistetaan kahdessa vaiheessa:

- Heti askelmoottorin pyörähdyksen jälkeen.
- Viivästetysti pillerin putoamisen odottamiseen tarkoitettussa tilassa.

Jos piezosensori tunnistaa pillerin putoamisen niin ensiksi luetaan EEPROM-muistista montako pilleriä on vielä jäljellä. Jos pillereitä on enemmän kuin 0 niin vähennetään pilleriän määrää EEPROM-muistissa sekä tehdään lokiin merkintä onnistuneesta pillerin annostelusta ja välitetään tapahtumailmoitus LoRaWAN-verkon kautta.

Jos pillereitä ei ole enää jäljellä niin tehdään lokiin EEPROM-muistiin merkintä epäonnistuneesta pillerin annostelusta sekä välitetään tapahtumailmoitus LoRaWAN-verkon kautta.

Jos piezosensori ei tunnista pillerin putoamista niin tehdään tilailmoitus lokiin EEPROM-muistiin tilanteesta sekä välitetään tilailmoitus LoRaWAN-verkon kautta.

Tapauksissa joissa piezosensori tunnisti pillerin putoamisen siirrytään seuraavaksi odotustilaan (wait_turn_st).

Tapauksissa joissa piezosensori ei tunnistanut pillerin putoamista siirrytään ensiksi pilleri ei pudonnut tilaan (pill_not_dropped_st). Tässä tilassa keskimmäinen LED-valo vilkkuu 5 kertaa sen merkiksi, ettei pilleri pudonnut. Tämän jälkeen vasta siirrytään odotustilaan (wait_turn_st).

3.3.3 Virran katkaisun käsittely

Virran palattua sovellus palaa siihen tilaan, jossa se oli ennen virran katkaisua EEPROM-muistin perusteella. Jos virta katkaisiin askelmoottorin pyöriessä joko pillerin käsittely tilassa (pill_dispensing_st) tai kesken rekalibroinnin (re_calib_st) niin silloin rekalibrointi suoritetaan uudelleen. Jos askelmoottori pysähtyy kesken kalibroinnin niin silloin kalibrointi suoritetaan uudelleen.

Rekalibroinnin jälkeen askelmoottori ottaa niin monta askelta eteenpäin, kuin oli keretty tekemään pillereitä annostellessa EEPROM-muistiin tallennetun moottorin vuoron muuttujan perusteella.

4 Järjestelmän arkkitehtuuri

Tässä luvussa esitellään projektin keskeiset laitteistokomponentit.

- ohjausyksikkö
 - controller PCB (Raspberry Pi Pico -piirilevy)
 - PicoProbe debugger
- muisti
 - I2C EEPROM (AT24C256)
- yhteysmoduuli
 - LoRaWAN-moduuli (LoRa-E5)
- mekaaninen annostelujärjestelmä
 - annostelijan pohja
 - annostelupyörä
 - askelmoottori
- sensorit
 - optinen sensori (optofork)
 - piezosensori
- käyttöliittymä
 - painikkeet
 - LED-valot
- liitännät
 - UART

- ADC
- I2C.

5 Projektin jako

Projekti toteutettiin yksilötyönä osana Metropolitan Embedded Systems Programming -kurssia. Vastasin projektin kaikista osa-alueista, mukaan lukien suunnittelu, ohjelmointi, testaus sekä dokumentaatio.

Suunnittelin projektin niin, että ensimmäisenä ohjelmoin toiminnallisuudet vähimmäisvaatimusten täyttämiseksi ja tämän jälkeen oli vuorossa edistyneempien toiminnallisuuksienvuoro. Sitä mukaa kun projektissa tuli vastaan toistuvaa logiikkaa niin kapseloin näitä uudelleen käytettäviksi funktioiksi. Jaoin ohjelmiston eri osa-alueet omiin lähdekooditiedostoihinsa rakenteen selkeyden ja ylläpidettävyyden parantamisen vuoksi.

