**Edupoli**

**TIETO-JA TIETOLIIKENNETEKNIIKAN PERUSTUTKINTO**

**Sulautetut sovellukset ja projektityöt,30 osp**

KitchenTimer – Keittiöajastin Arduinolla

**Toni Seppänen**

**17.12.2018**

[1 Ammattitaitovaatimukset 1](#_Toc531248159)

[2 Kuvaus laitteesta 1](#_Toc531248160)

[3 Kuvaus elektroniikasta 2](#_Toc531248161)

[4 Kuvaus ohjelmasta 3](#_Toc531248162)

[5 Linkit, ohjeet, inspiraatiolähteet 4](#_Toc531248163)

[6 (YTO Kestävän kehityksen edistäminen) 5](#_Toc531248164)

[7 Projektikansio 6](#_Toc531248165)

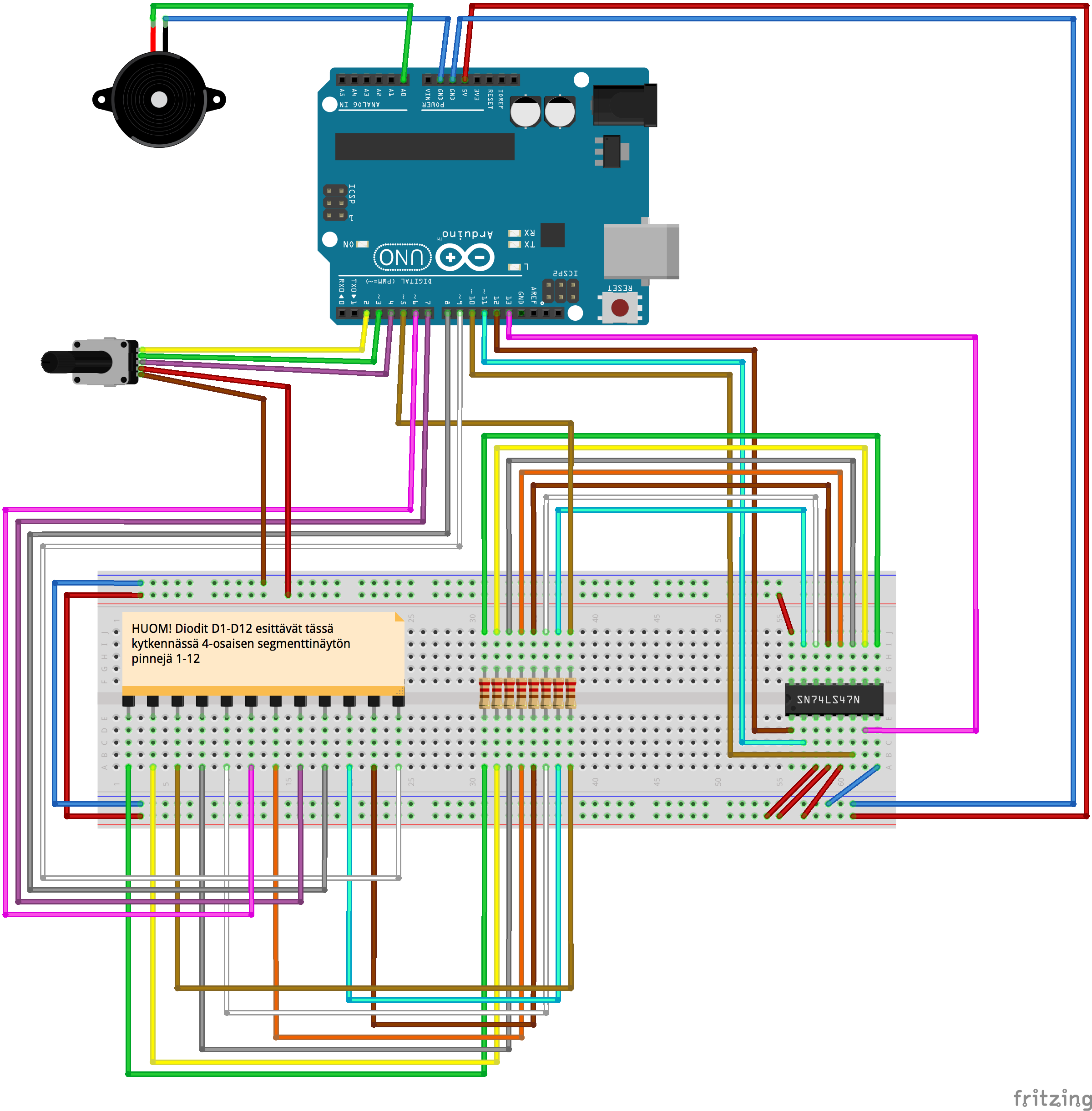
# Ammattitaitovaatimukset

<https://eperusteet.opintopolku.fi/#/fi/esitys/3397335/reformi/tutkinnonosat/3436381>

# Kuvaus laitteesta

Laitteen tarkoituksena on toimia keittiöajastimena/munakellona eli ilmoittaa, tässä tapauksessa pienellä musiikkikappaleella, kun määritetty aika on kulunut loppuun ja näin ollen pyrkiä estämään pohjaan palaneet ruoat.

Toiminnasta on pyritty tekemään mahdollisimman yksinkertainen, joten käyttäjällä on vain yksi kontrolleri (pulssi-enkooderi/pulssianturi), josta tehdään kaikki tarvittavat valinnat. Painiketta pyörittämällä myötäpäivään lisätään ajastukseen 30 sekuntia ja vastapäivään sitten vähennetään vastaavasti 30 sekuntia. Sitten kun haluttu aika on määritetty, painiketta painetaan ja ajastus alkaa. Jos on tarvetta tauolle niin painiketta painamalla ajanotto pysähtyy siihen saakka kunnes painiketta painetaan uudemman kerran.

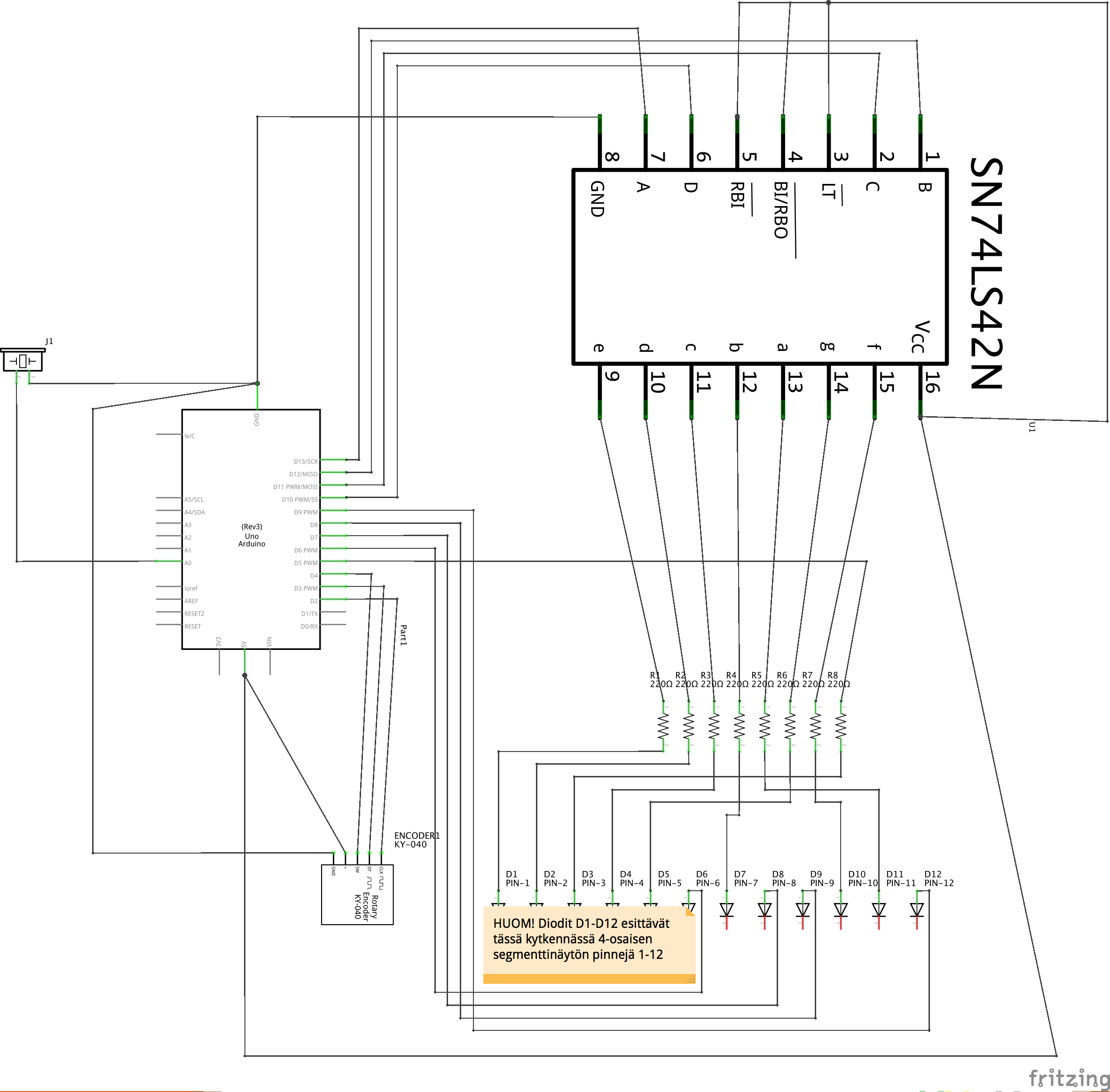


# Kuvaus elektroniikasta

Ajanoton näyttönä toimii yhteisanodilla oleva 4-osainen 7-segmenttinäyttö, punaisilla ledeillä. Tätä on ohjaamassa SN74LS42-piiri.

Ainoan kontrollerin virkaa toimittaa Keyes KY-040 pulssi-enkooderi. Tässä on sisäänrakennettuna 10k vastukset kiertopotikalle (ei kuitenkaan painiketoiminnolle, tällöin pitää käyttää ulkoista tai sitten input\_pullup käskyä), joita voi halutessaan käyttää. Tämä edellyttää siinä tapauksessa 5V ja GND-kierrättämistä pulssi-enkooderin kautta. Tätä tapaa on myös käytetty tässä keittiöajastimessa

Äänimerkkien antamisessa toimii ihan tavallinen piezo-kaiutin.



Eaglellä en tällä kertaa tehnyt piirilevyä, koska meidän koulun jyrsin on toiminut viime aikoina epäluotettavasti. En halunnut laittaa edes suunnitelmaa tästä, koska en voinut olla varma, että se olisi toiminut ennen testaamista.

# Kuvaus ohjelmasta

**Yleistä**

Keittiöajastin on toimintaperiaatteeltaan yksinkertainen ja näin ollen olen myös pyrkinyt pitämään ohjelmarakenteen yksinkertaisena ja selkeänä, jotta voin palata sen pariin myöhemminkin. Koodia olen viilaillut ja tehostanut sitä mukaan kun olen jotain uutta keksinyt tai oivaltanut. Tällä hetkellä ohjelma sisältää 258 riviä koodia sekä muutaman erillisen kirjaston. Käyn alla lyhyesti läpi ohjelman toimintaa ja kokonaisuudessaan koodi löytyy githubista (linkki lopussa).

**Ohjelmiston rakenne**

Käytännössä ohjelma toimii tilakoneen tavoin eli se odottaa käyttäjän toimia ja sen jälkeen nostaa oikean ”lipun” (tässä tapauksessa **timerMsg**) ja sen perusteella tapahtuu asioita.

|  |
| --- |
| void loop() { |
|  |  |
|  | // timerMsg = 0 | Ajastimen demo-moodi | Timer demo mode |
|  | // timerMsg = 1 | Ajastimen asetus | Set timer |
|  | // timerMsg = 2 | Ajastin on päällä | Timer is on |
|  |  |
|  | if (timerMsg == 0) { |
|  | timerDemo(); |
|  | } |
|  |  |
|  | if (timerMsg == 1) { |
|  | setKitchenTimer(); |
|  | } |
|  |  |
|  | if (timerMsg == 2) { |
|  | timerOn(); |
|  | } |
|  |  |
|  | if (rotBTN.pressed() && timerMsg == 0) { |
|  | timerMsg = 1; |
|  | } |
|  | } |

Käytössä olevat tilat:

**timerMsg = 0**

Aluksi pyöritellään ajastinta vain demo-moodissa (**timerMsg = 0**) joka kutsuu **timerDemo()** funktiota ja näyttää näytöllä numerot 1234. Tästä tietää, että laite on käyttövalmiina eikä mitään ajastustoimintaa ole kesken.

**timerMsg = 1**

Kun laite on käyttövalmiina ja näytössä näkyy numero ”1234” voidaan painikkeen painalluksella siirtyä asettamaan ajastinta. Nyt **timerMsg** lippu muuttuu ja kutsutaan funktiota **setKitchenTimer()** jossa päästään asettamaan haluttu aikamäärä ajastimeen. Kun haluttu aika on asetettu voidaan painikkeen painamisella käynnistää ajastin (**timerMsg = 2**)

**timerMsg = 2**

Kun haluttu aika on asetettu, voidaan taas painaa painiketta ja **millis()** -funktio käynnistyy ja aika lähtee rullaamaan. Aikaa mitataan **millis()** -funktion avulla. Sitten kun aika on kulunut loppuun, kuuluu piezo-summerista ensin pieni musiikinpätkä (tällä hetkellä joululaulu) ja sitten ”lippu” (**timerMsg**) palaa takaisin demo-moodiin (**timerMsg = 0**)

Ajan juoksu on myös mahdollista keskeyttää hetkeksi painamalla kerran painiketta jolloin mennään ”pause tilaan” ja painamalla uudelleen ajanjuoksu jatkuu. Tämä toimii samalla periaatteella kuin **timerMsg** ”lippu”, mutta vaihtaa vain **pause** lipun tilaa – ***pause = 0 🡪 pause = 1*.** Käytännössä tuossa tapahtuu se, että näytön numeroiden päivitys loppuu, mutta silti **millis()** funktio pyörii taustalla valmiina odottamassa jatkoa.

**Käytetyt kirjastot**

Projektissa on käytetty muutamia kirjastoja toiminnan avuksi ja selkeyttämiseksi.

[*libSeg47.h*](https://github.com/tonzik/libSeg47)

Kirjasto ohjaa 4-osaisen 7-segmenttinäytön toimintaa, SN47LS42-piirin avulla. Kirjasto syntyi (tietysti opettajan ja muiden opiskelijoiden avustuksella) siitä syystä kun yritin opetella kuinka kirjastoja Arduinolle tehdään. Kirjasto on myös tehty tämä projekti silmällä pitäen.

[*libButton.h*](https://github.com/tonzik/libButton)

Kirjasto ohjaa painikkeen painallusten toimintaa. Tämä on opettajamme Jani Lehdon käsialaa kun hän perehdytti meitä siihen kuinka kirjastoja tehdään. Mielestäni kirjasto on yleishyödyllinen ja siitä saa helposti mallia niin otin se mukaan tästä syystä.

[*libRTTTL\_Tunes.h*](https://github.com/tonzik/libRTTTL_Tunes)

Kirjasto soittaa vanhoista kännyköistä tuttua pimputusta jota silloin soittoääniksi kutsuttiin. Tämän ohjelman koodin löysin valmiiksi netistä, mutta tein siitä kirjaston helpomman käytön saavuttamiseksi.

# Linkit, ohjeet, inspiraatiolähteet

*Kuvaile työtapaasi. Mistä sait ideoita. Mieti miten sait muilta apua. Mieti miten itse osasit muita auttaa. Mikä oli haastavaa työn organisoinnissa. Mikä sujuu sinulta luontevasti.*

Idea tähän projektiin lähti ihan tarpeesta. Meillä hajosi keittiön ajastin ja ajattelin, että tälläinen projekti olisi ihan hyvä oppimista varten ja mahdollisesti jos kaikki menee hyvin niin lopputulos jopa sitten varsinaiseen käyttöön asti.

Aluksi mietitytti kuinka lähteä ohjelmaa rakentamaan ja mitä kaikkea sen pitäisi pitää sisällään. Laitteen toimintaperiaatehan on idealtaan yksinkertainen, mutta näin vasta hiljattain Arduinon pariin siirtyneen näkökulmasta katsoen työtä ja ajattelua vaadittiin jo enemmän kuin pelkän ledin sytyttämiseen. Onneksi apua sai pyytämällä niin opettajalta kuin muiltakin opiskelijoilta.

Harmaita hiuksia ehkä eniten tässä projektissa aiheutti **millis()**-funktio. Sain tämän kuitenkin toimimaan loppujen lopuksi haluamallani tavalla. Ehkä en kuitenkaan ihan vielä ole sinut sen kanssa, mutta sanotaan, että tulemme jo jonkin verran toimeen. Toinen ongelma liittyi myös osittain millikseen eli tuo pause-toiminto aiheutti ihmetystä. Sekin saatiin kuitenkin toimimaan kunhan sitä aikansa ihmetteli.

Se mitä tämän projektin suhteen on oppinut auttaa myös tulevissa projekteissa. Huomasin, että koodin kommentoinnista on apua siinä vaiheessa kun projektiin palaa uudelleen toisen idean kanssa. Pääsee heti kärryille mitä tapahtui missäkin kohdassa. Myös versionhallinta (tässä tapauksessa GitHub) on ollut oiva työkalu uusien koodi-ideoiden kokeilussa. Myöskin se oli hyvä ajatus, että ei heti lähdetä rönsyilemään ominaisuuksien kanssa ennen kuin ollaan Arduinon kanssa enempi kärryillä. Ainakin omassa tapauksessa se toimi mielestäni.

Olen ehkä saanut sellaisia tietoja ja taitoja tämän projektin avulla, että voin mahdollisesti auttaa muita joissakin asioissa. Ainakin osaan Googlesta etsiä ja löytää erilaisia ratkaisuja ☺

# (YTO Kestävän kehityksen edistäminen)

Tämä osio vain 2018 aloittaneille joilta eHOPSista puuttuu arvosana kyseisestä kurssista.

<https://eperusteet.opintopolku.fi/#/fi/esitys/3397335/reformi/tutkinnonosat/4209965>

Pakolliset osaamistavoitteet, 1 osp

Opiskelija osaa

* toimia kestävän kehityksen periaatteiden mukaisesti
* ottaa huomioon elinkaariajattelun periaatteet
* pohtia ratkaisuja eettisistä näkökulmista.

Olen tämän osion suorittanut erillisellä kurssilla.

# Projektikansio

[Linkki projektikansioon githubissa](https://github.com/tonzik/KitchenTimer)

* kytkentäkaavio
* ohjelmiston lähdekoodit
* tämä dokumentti (Word)