



# ESP32-pohjainen ajanottojärjestelmä rally-tokoon

Roosa-Maria Rovio

OPINNÄYTETYÖ

Lokakuu 2025

Sähkö- ja automaatiotekniikka

Automaatiotekniikka

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Sähkö- ja Automaatiotekniikan Tutkinto-ohjelma  
Automaatiotekniikka

ROVIO, ROOSA-MARIA:  
ESP32-pohjainen Ajanottojärjestelmä Rally-tokoon

Opinnäytetyö 31 sivua, joista liitteitä 3 sivua  
Syyskuu 2025

---

Tässä opinnäytetyössä suunniteltiin sähköinen ajanottojärjestelmä rally-tokokilpailuihin, korvaamaan manuaalista käsikelloa. Opinnäytetyönä suunniteltiin kustannustehokas, kompakti ja langattomasti toimiva ratkaisu, joka on helposti käyttöönotettavissa ulkokentillä ilman verkkovirtaa.

Järjestelmä perustuu ESP32-mikrokontrolleriin ja hyödyntää PIR-liiketunnistimia ajanoton käynnistämiseen ja pysäyttämiseen. Mittaustulokset tallennetaan laitteeseen CSV-muodossa, ja ne ovat nähtävissä ESP32:n luoman selainkäyttöliittymän kautta. Järjestelmää pystyy käyttämään ilman internetyhteyttä, mikä mahdollistaa sen käytön kenttäolosuhteissa.

Ohjelmointi toteutettiin Arduino IDE -ympäristössä käyttäen C++-kieltä sekä avoimen lähdekoodin kirjastoja. Tuloksena syntyi helposti laajennettava ajanottojärjestelmä, joka tukee tulevaisuudessa esimerkiksi integraatiota kilpailujärjestelmiin kuten Virkku. Järjestelmä täyttää rally-tokon käytännölliset tarpeet.

## **ABSTRACT**

Tampereen Ammattikorkeakoulu  
Tampere University of Applied Sciences  
Degree Programme in Electrical and Automation Engineering  
Automation Engineering

ROVIO, ROOSA-MARIA:  
ESP32-Based Timing System for Rally Obedience

Bachelor's thesis 31 pages, appendices 3 pages  
September 2025

---

This thesis presents the development of an electronic timing system for rally obedience competitions, replacing traditional manual stopwatch timing. The thesis was to create a cost-effective, compact, and wireless solution that could be easily deployed in outdoor environments without a power supply.

The system is based on an ESP32 microcontroller and uses PIR motion sensors to start and stop the timing automatically. The measured times are stored in CSV format on the device and can be viewed via a web-based interface hosted directly by the ESP32. The system operates independently in Access Point mode, eliminating the need for an external internet connection.

The software was developed using the Arduino IDE in C++, leveraging several open-source libraries. The result is a standalone, extendable timing solution that can be integrated in the future with competition management systems such as Virkku. The system meets both the technical and practical needs of rally obedience.

---

Keywords: ESP32, timing system, microcontroller, embedded system

## SISÄLTÖ

1	JOHDANTO .....	1
2	TAVOITTEET .....	3
3	VAATIMUKSET .....	5
	3.1 Fyysinen laitteisto.....	5
	3.2 Ohjelmisto .....	6
4	SOVELLUS .....	9
	4.1 Kehityskohteet.....	10
5	OHJELMISTOKEHITYS.....	11
	5.1 Integraatio .....	12
6	TEKNINEN TOTEUTUS JA KOMPONENTTIVERTAILU.....	13
	6.1 Mikrokontrollereiden vertailu .....	13
	6.2 Liiketunnistin .....	14
	6.3 Kotelointi .....	16
	6.4 Akkuratkaisu .....	17
	6.5 Tietoturva .....	18
7	KÄYTTÄJÄKYSELY.....	20
	7.1 Seuratausta ja nykyiset käytännöt.....	20
	7.2 Kiinnostus .....	20
	7.3 Palaute .....	21
	7.4 Johtopäätökset kyselyyn .....	21
8	JATKOKEHITYS .....	22
9	KUSTANNUSARVIO.....	24
10	YHTEENVETO .....	27
11	LIITEET.....	29
12	LÄHTEET.....	32

## 1 JOHDANTO

Tämän opinnäytetyö suunnitelma kehittää rally-tokossa käytettävän ajanoton tarkkuutta, vaihtamalla se manuaalisesta käsikellosta sähköiseksi ajanottojärjestelmäksi. Tarkoituksena on kehitellä taloudellisesti kannattava laitteisto ja ohjelma täyttämään rally-tokon tarpeet.

Rally-toko on koiraurheilulaji, jossa yhdistyvät koiran hallinta, ohjaajan ja koiran välinen yhteistyö sekä tarkkuus ja suorituksen laatu. Ajanotto ei perinteisesti ole ollut lajin keskiössä. Siitä huolimatta aika vaikuttaa lopulliseen sijoitukseen tasapelin sattuessa. Kun vuonna 2026 sääntömuutos siirtää ajanoton alkamaan heti kehäalueelle siirtymisestä, nousee ajankäyttö entistä keskeisempään rooliin. Samalla tämä muutos avaa ovia sähköisten järjestelmien hyödyntämiselle myös rally-tokon kentillä.

Sähköisen ajanoton yleistyminen muissa koiraurheilulajeissa, kuten agilityssä, on jo osoittanut sen hyödyt: mittaustarkkuus paranee, inhimilliset virheet vähenevät ja kilpailun järjestely helpottuu, kun osa tehtävistä voidaan automatisoida. Näin ollen myös rally-tokossa sähköinen ajanotto voidaan nähdä välineenä lajin kehittymiselle – ei vain teknisenä parannuksena, vaan myös toiminnan sujuvoittajana ja harrastajien tasa-arvoisen kohtelun mahdollistajana.

Eräs merkittävä hyöty sähköisessä ajanotossa liittyy talkootyön vähentämiseen. Manuaalinen ajanotto vaatii erillisen talkoilijan, jonka tehtävänä on kellottaa jokainen kilpailusuoritus tarkasti oikeasta hetkestä alkaen ja lopettaen. Automaattinen, liikkeentunnistukseen perustuva järjestelmä poistaa tämän tarpeen kokonaan. Yhdessä kilpailupäivässä voi olla jopa 60 suoritusta, ja ajanottajan on oltava jatkuvasti läsnä ja tarkkaavainen. Kun tämä työ saadaan korvattua automaatiolla, voidaan vapauttaa yksi henkilö muihin tehtäviin – tai jättää kokonaan pois. Tämä ei ole pieni asia, sillä vapaaehtoisten saatavuus on monilla seuroilla jatkuva haaste. Talkoolaisia ei ole helppo löytää. Jos seura järjestää esimerkiksi kisat kerran kuussa, ja ajanottajalle kertyy päivän aikana 6–8 tuntia työtä, puhutaan

vuositasolla noin 72 talkootunnin vähentymisestä yhden henkilön osalta. Suuremmassa seurassa, jossa järjestetään useampia tapahtumia tai useita kehien ajanottoa rinnakkain, vaikutus kasvaa entisestään.

Sähköinen ajanotto voi myös parantaa kilpailujen toistettavuutta ja yhdenmukaisuutta. Manuaalisesti käynnistetty kello on aina riippuvainen talkoilijan reaktiokyvystä, ja vaikka erot voivat olla pieniä, niillä voi olla merkitystä esimerkiksi tasapistetilanteissa. Automaattinen ajanotto, joka perustuu objektiiviseen liikkeentunnistukseen, takaa sen, että jokaiselle kilpailijalle aika mitataan samalla logiikalla.

On kuitenkin huomattava, että sähköinen ajanotto ei ole itseisarvo. Sen tuomien hyötyjen on vastattava siihen liittyviä kustannuksia, teknistä osaamista ja käyttöönottoon vaadittavaa vaivaa. Tässä työssä esitetty ratkaisu – ESP32-pohjainen langaton järjestelmä – on suunniteltu nimenomaan vapaaehtoistoiminnan ja pienempien seurojen resurssit huomioon ottaen. Se tuo sähköisen ajanoton hyödyt ilman suuria investointeja tai monimutkaisia asennuksia.

Lopulta sähköinen ajanotto on osa suurempaa muutosta, jossa koiraurheilun kilpailutapahtumat siirtyvät askel askeleelta kohti digitaalisempaa ja automaattisempaa toimintaa. Niin kuin agilityssa on jo menty.

## 2 TAVOITTEET

Urheilukisoissa on jo käytössä olevia sähköisiä ajanottojärjestelmiä, kuten valokennoihin perustuvia ratkaisuja, RFID-siruja ja transpondereita sekä kamerajärjestelmiä tarkan maaliintulon määrittämiseksi, mutta haluan että suunnittelemani järjestelmä olisi kustannustehokkaampi, kompaktimpi ja käyttäjäystävällinen.

Kustannustehoikkaus on tärkein tavoitteeni, jotta mahdollisimman monella seuralla voisi olla mahdollisuus käyttää sähköistä ajanottojärjestelmää. Rally-toko kisoja järjestävät seurat toimivat vapaaehtoistoimintana ja voittoa tavoittelemattomina yhdistyksinä, eikä seurojen tavoite ole haalia kassavirtaa vaan edistää jäsenistölle mahdollisuus harrastamiseen. Tästä syystä seuroilla ei ole välttämättä taloudellista mahdollisuutta panostaa sähköiseen ajanottojärjestelmään, kun se ei ole kisojen järjestämisen kannalta ~~kumminkaan~~ vaatimus.

Toiseksi tärkein tavoite on mahdollisimman kompakti laitteisto olisi mahdollisimman kompakti, koska lämmin säilytystila on myös seuroilla rajallista. Kukaan rallytokovastaava tuskin haluaa kotinsa nurkkaan mitään isoa ajanottojärjestelmää, jos seuroilla ei ole ollenkaan lämmintä säilytystilaa, koska osalla seuroista on vaan ulkokenttä ja mahdollisesti pieni varasto, osalla ei välttämättä ollenkaan omia tiloja.

Kolmantena järjestelmän tavoitteista olisi langaton toteutus, mutta tästä voidaan tinkiä, jos kustannustehoikkaus kärsii liikaa. Litiumioniakkupohjainen järjestelmä on ulkokenttiä ajatellen järkevin toteuttaa.

Lisäksi tavoitteena on, että järjestelmä ei vaadi käyttäjältään syvällistä teknistä osaamista. Kaikki asennus- ja käyttöönotto on suunniteltu mahdollisimman suoraviivaiseksi. Käyttäjän ei tarvitse osata ohjelmoida tai konfiguroida laitteita, vaan laite toimii heti käyttövalmiina akkujen ollessa ladattuna.

Erityistä huomiota pyrittiin myös kiinnittämään siihen, että järjestelmä ei häiritse koiria kilpailutilanteessa. Tämä tarkoitti mm. äänettömyyttä, huomaamatonta

muotoilua ja sitä, ettei antureita tai koteloita sijoiteta liian näkyvästi radalle. Tavoitteena on siis järjestelmä, joka ikään kuin "häviää taustalle" kilpailun aikana.

### 3 VAATIMUKSET

Tässä osuudessa käydään läpi laitteiston teknillisiä vaatimuksia ja mahdollisia esimerkkiratkaisuja anturoinnista. Kerrotaan mikrokontrollerista ja miten päädyttiin valitsemaan se opinnäytetyöprojektiin.

#### 3.1 Fyysinen laitteisto

Teknisiltä vaatimuksiltaan järjestelmän on pystyttävä mittaamaan aikaa millisekuntien tarkkuudella, havaitsemaan liike ilman fyysistä kontaktia, toimimaan langattomasti ja tallentamaan mittaustulokset paikallisesti. Lisäksi käyttöliittymän on oltava selkeä ja saavutettavissa tavallisella verkkoselaimella, esimerkiksi puhelimella tai tabletilla. Järjestelmässä käytetään ESP-WROOM-32-mikrokontrolleria, joka tarjoaa hyvän suorituskyvyn ja sisältää sisäänrakennetut Wi-Fi- ja Bluetooth-yhteydet. Sen 240 MHz:n kellotaajuus ja kaksi suoritinydintä mahdollistavat rinnakkaisen tietojenkäsittelyn, mikä tekee siitä sopivan reaaliaikaiseen ajanottoon. Lisäksi se tarjoaa useita GPIO-liitäntöjä anturien ja muiden laitteiden kytkemiseen. (David Watson 2020)

Liikkeen havaitsemiseen käytetään HC-SR501 PIR-liiketunnistinta, joka on edullinen, helposti saatavilla ja soveltuu erityisesti liikkeen tunnistamiseen. Anturi reagoi lämpösäteilyn muutoksiin näkökentässään ja antaa digitaalisena ulostulona korkean signaalin (HIGH), jonka ESP32 tunnistaa. Ajanoton logiikka perustuu kahteen erilliseen liikkeen havaitsemiseen – ensimmäinen käynnistää ajanoton ja toinen pysäyttää sen. Tämä mahdollistaa yksinkertaisen ja häiriöttömän toiminnan ilman käyttäjän aktiivista osallistumista.

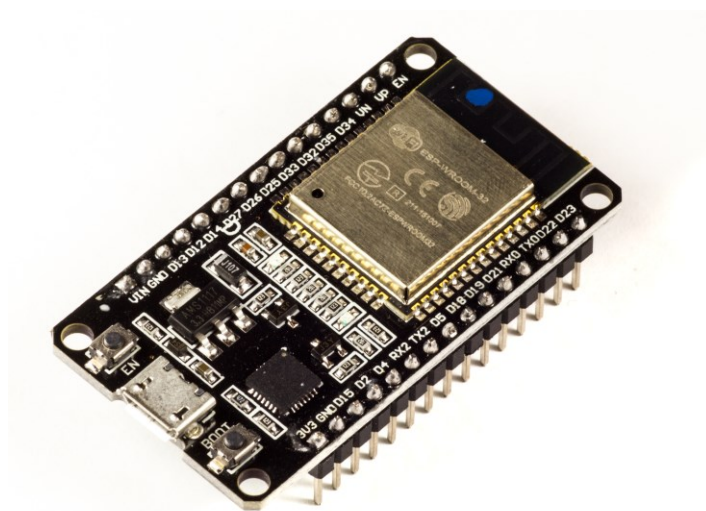
Mittausdata tallennetaan ESP32:n sisäiseen flash-muistiin SPIFFS-tiedostojärjestelmää hyödyntäen CSV-muodossa, mikä mahdollistaa tulosten katselun verkkosivun kautta. Web-käyttöliittymä toteutetaan ESP32:n sisäisellä web-palveli-

mella, joka näyttää tallennetut ajat selkeänä taulukkona. Järjestelmä toimii itsenäisesti langattomana Access Point -verkossa, johon käyttäjä voi liittyä ilman ulkoista internetyhteyttä.

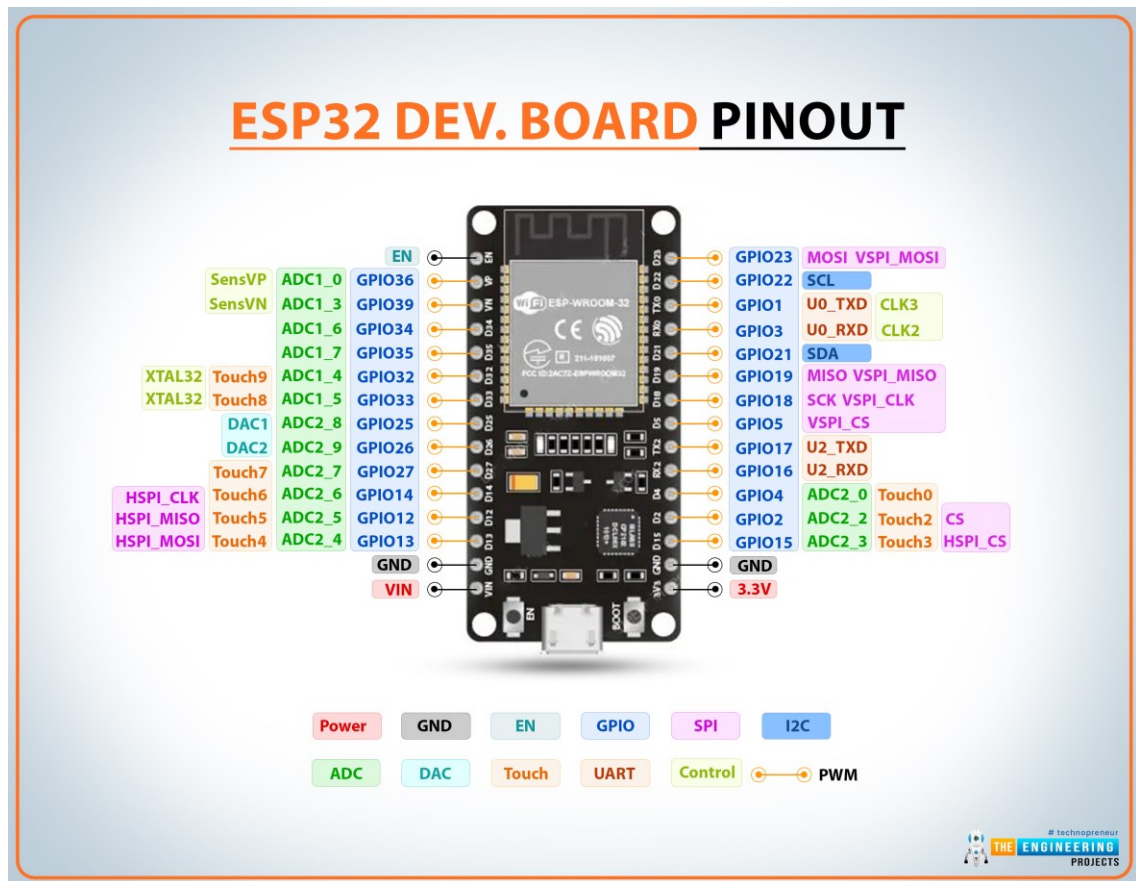
Koko järjestelmä suunnitellaan toimimaan akkuvirralla, jotta se olisi käyttökelpoinen myös ulkokentillä, joissa sähköä ei ole saatavilla.

### 3.2 Ohjelmisto

ESP32 on kiinalainen mikrokontrolleri, joka sisältää mahdollisuuden bluetooth- ja wifi-yhteyden luomiseen. Malleja on useita mutta kyseinen malli on ESP-WROOM-32 jota aion käyttää tulevan suunnitelmani keskiössä.



KUVA 1. ESP32 mikrokontrolleri (Wikipedia 2018)



KUVA 2. Liitinkytkentä kuva (James Wilson, 2020)

Kuvassa on esitetty WROOM-32 liitinkytkentä kaavio, jonka perusteella osataan kytkeä oikein ja ohjelmoida koodi oikeisiin pinneihin. Kaaviosta voidaan nähdä, kuinka pinnit on suunniteltu monikäyttöisiksi.

TAULUKKO 1. Pinnien selitykset (James Wilson, 2020)

VIN	Virran sisään syöttö
GND	Maadoitus
3.3V	Ulostulo
GPIO1..39	Yleiskäyttöisiä I/O pinnejä
ADC	Analogian lukija
DAC	Tuottaa analogista jännitettä digitaalisesta arvosta

TOUCH	Kapasitiivinen kosketusanturi
SPI	Sarjaviestintää (esim. Näyttö)
PWM	Ohjataan esim. servoja
UART	Sarjaviestintää toisen kontrollerin kanssa
I2C	Antureille ja muille laitteille
EN	Ohjauspinni, Reset

Opinnäytetyössäni päädyin käyttämään ESP32-mikrokontrolleria, koska se on monipuolinen ja omaa hyvän suorituskyvyn pieneen kokoon sovitettuna. Ja koska mikrokontrollerissa on sisäänrakennettu Wi-Fi- ja Bluetooth-yhteys, se tekee siitä sopivan käyttö tarkoituksiini.

ESP32:n suorituskyky on riittävän tarkka ajanseuranta, koska se pystyy käsittelemään tietoa rinnakkain. Tämän ansiosta se pystyy mittaamaan aikaa, käsittelemään sitä ja lähettämään sen ilman pitkiä viiveaikoja. Mikrokontrollerin useampi GPIO-pinni paikka mahdollistaa useiden laitteiden liittämisen, mitä mahdollisesti tarvitaan ajanottolaitteistosta.

Ohjelmiston tekemiseen aion käyttää Arduino IDEä ja konfiguroida koodin sopivaksi ESP32-mikrokontrollerille.

Projektissa hyödynnettiin useita avoimen lähdekoodin kirjastoja, jotka laajentavat ESP32-mikrokontrollerin ominaisuuksia erityisesti verkkopalvelimen ja tiedonsiirron osalta. Verkkopalvelimen toteuttamiseksi käytettiin ESPAsyncWebServer-kirjastoa, joka mahdollistaa tehokkaan ja rinnakkaisen HTTP-pyyntöjen käsittelyn ilman, että mikrokontrollerin pääohjelmasilmutta kuormittuu tarpeettomasti.

Lisäksi ESPAsyncWebServer vaatii toimiakseen riippuvuuskirjaston ESPAsyncTCP. Molemmat kirjastot löytyvät Arduino IDE:n kirjastohaun (Library Manager) kautta, mistä ne voidaan asentaa suoraan ilman manuaalista latausta.

Kirjastojen asentamisen jälkeen ne otetaan käyttöön lisäämällä ohjelmakoodin alkuun `#include`-lauseet, minkä jälkeen ESP32 pystyi toimimaan tehokkaana verkkopalvelimena ja palvelemaan käyttöliittymän suoraan selaimelle ilman erillisiä ulkoisia palvelimia. (Rui Santos ja Sara Santos)

## 4 SOVELLUS

Ajanottolaitteiston tallennusratkaisussa käytetään ESP32-mikrokontrollerin sisäistä Flash-muistia, johon tulokset tallennetaan CSV-muotoisena SPIFFS-tiedostojärjestelmää hyödyntäen. Tämä mahdollistaa pysyvän tallennuksen ilman ulkoista SD-korttia ja pitää laitteiston rakenteen yksinkertaisena ja kompaktina. Kukin mittausta tallennetaan omalle rivilleen millisekuntitarkkuudella, ja tiedot säilyvät laitteen muistissa uudelleenkäynnistyksistä huolimatta.

Tiedonhallinnan kannalta järjestelmässä hyödynnettiin mahdollisimman yksinkertaista ja kevyttä tiedostomuotoa – CSV-tiedostoja. Tämä mahdollistaa yhteensopivuuden lähes kaikkien taulukkolaskentaohjelmien kanssa, ja siten mittaustietoja voidaan helposti siirtää edelleen esimerkiksi Exceliin, Google Sheetsiin tai tulevaisuudessa mahdollisesti suoraan kilpailun pöytäkirjoihin.

Koska SPIFFS-tiedostojärjestelmä ei ole rajattoman tallennuskyvyn omaava, mittauksia voidaan halutessa myös automaattisesti kierrättää: vanhimmat rivit poistetaan, kun tietty raja ylittyy, mikä pitää järjestelmän kevyenä. Kilpailussa on koirakkomäärä maksimissaan 60 koirakkoa, josta vähän ylempää arvoa voitaisiin pitää raja-arvona rivien poistoon.

Käyttäjäraja-apinta on toteutettu ESP32:n sisäisellä web-palvelimella, joka toimii Access Point -tilassa. Tämä tarkoittaa, että käyttäjä voi yhdistää puhelimensa tai muun mobiililaitteen suoraan laitteen muodostamaan Wi-Fi-verkkoon ilman ulkopuolista internetyhteyttä. Käyttöliittymä toimisi selaimessa, eikä se vaadi erillisiä sovelluksia tai ohjelmia. Käyttöliittymä näyttäisi mittaustulokset selkeänä taulukkona. Tulevaisuudessa käyttöliittymään voidaan lisätä painikkeita esimerkiksi tietojen nollaamiseen tai CSV-tiedoston lataamiseen.

Järjestelmä on suunniteltu laajennettavaksi niin, että mittaustulokset voidaan siirtää tai integroida muihin tietojärjestelmiin, kuten Suomen Kennelliiton Virkku-järjestelmään. Tämä edellyttää Virkulta ohjelmointirajapintaa (API), jonka avulla

ESP32 voi lähettää ajanottotulokset suoraan tietokantaan tai pilvipalveluun kilpailun järjestämistä varten. Mikäli rajapinta on saatavilla, tulokset voidaan lähettää automaattisesti heti mittauksen jälkeen, mikä vähentää manuaalista työtä ja virhemahdollisuuksia. Järjestelmä voidaan päivittää tukemaan HTTP POST -lähetyksiä tai MQTT-protokollaa, mikäli Virkku tai muu vastaava järjestelmä tukee näitä yhteystapoja.

#### **4.1 Kehityskohteet**

Rally-tokon ajanotto on perinteisesti toteutettu manuaalisesti, käyttäen käsikelloja, joita käynnistetään ja pysäytetään manuaalisesti. Tämä menetelmä on edullinen ja teknisesti kevyt, mutta se on altis inhimillisille virheille, kuten epätarkalle reaktioajalle, viivästyneelle kellon käynnistykselle tai virheelliselle pysäytykselle. Lisäksi manuaalinen ajanotto vaatii kilpailun toimihenkilöiltä jatkuvaa tarkkaavaisuutta.

Joissakin koiraurheilulajeissa on otettu käyttöön sähköisiä ajanottojärjestelmiä, jotka perustuvat valokennoihin. Nämä ratkaisut tarjoavat tarkempaa mittausta, mutta ovat usein kalliita, tilaa vieviä ja vaativat tarkkaa asettelua ennen käyttöä.

Mahdollisia kehityskohteita tulevaisuudessa ovat mm. järjestelmän automaattinen synkronointi ulkoisten aikapalvelimien kanssa (esim. NTP), tulosten automaattinen siirto kilpailun hallintajärjestelmiin kuten Virkku. Lisäksi voidaan kehittää robustimpi kotelointi säänkestäväksi sekä mahdollisuus akun varaustason etäseurantaan. Käyttöliittymään voidaan lisätä toimintoja, kuten erillinen tuomaritila, tulosten vienti PDF- tai Excel-muodossa.

## 5 OHJELMISTOKEHITYS

Projektissa käytettiin useita nykyaikaisia teknologioita ja ohjelmointikieliä, jotka yhdessä muodostavat helposti käyttöönotettavan ja luotettavasti toimivan ajanottojärjestelmän. Järjestelmän ytimessä on ESP32 (ESP-WROOM-32) -mikrokontrollerialusta, joka tarjoaa kaksiytimisen prosessorin ja sisäänrakennetut langattomat yhteydet, mikä mahdollistaa sekä tiedonsiirron että käyttöliittymän toteuttamisen ilman ulkoisia verkkolaitteita.

Ohjelmiston kehittämiseen hyödynnettiin Arduino IDE:tä, joka soveltuu erityisesti sulautettujen laitteiden ohjelmointiin ja tarjoaa laajan valikoiman valmiita kirjastoja ESP32:n eri ominaisuuksien hallintaan. Varsinainen ohjelmointikieli on C++, jolla on toteutettu ajanottologiikka, antureiden lukeminen, tiedon tallennus sekä web-palvelimen toiminta.

Mittausdatan tallennus on toteutettu hyödyntämällä SPIFFS-tiedostojärjestelmää, jonka avulla data tallentuu ESP32:n Flash-muistiin CSV-muodossa pysyvästi ja on helposti luettavissa myöhemmin. Käyttäjäräjäpinta on toteutettu selaimessa toimivalla HTML-pohjaisella sivulla, joka esittää mittaustulokset taulukossa selkeästi ja reaaliaikaisesti.

Langaton yhteyskäyttö on toteutettu Wi-Fi Access Point -tilassa, jolloin laite luo oman verkkonsa, johon käyttäjä voi yhdistää mobiililaitteellaan ilman erillistä internetyhteyttä tai reititintä.

Kokonaisuutena käytetyt teknologiat mahdollistavat itsenäisesti toimivan, käyttäjätasavälisen ja kenttäolosuhteissa helposti käyttöönotettavan järjestelmän, joka ei vaadi ulkopuolisia palveluita tai monimutkaisia asennuksia.

## 5.1 Integraatio

Tietojen siirto laitteen ja web-sovelluksen välillä on toteutettu ESP32-mikrokontrollerin sisäisen web-palvelimen avulla. ESP32 toimii Access Point -tilassa ja luo oman langattoman verkon, johon käyttäjä voi yhdistää mobiililaitteella tai tietokoneella. Kun käyttäjä avaa selaimen ja siirtyy laitteen IP-osoitteeseen, ESP32 palvelee HTML-pohjaista käyttöliittymää suoraan omasta muististaan.

Mittausdata tallennetaan reaaliajassa ESP32:n Flash-muistiin SPIFFS-tiedostojärjestelmään CSV-muodossa. Kun käyttäjä avaa käyttöliittymän, web-palvelin lukee tallennetut tiedot tiedostosta ja generoi HTML-sivun, joka sisältää mittaustulokset taulukkona. Tiedonsiirto tapahtuu siis HTTP-protokollan yli selaimen ja laitteen välillä, eikä vaadi ulkopuolista internet-yhteyttä. (David Watson 2020)

## 6 TEKNINEN TOTEUTUS JA KOMPONENTTIVERTAILU

Tässä luvussa esitellään ajanottojärjestelmän keskeiset tekniset ratkaisut, joita ovat mikrokontrollerin ja anturien valinta, laitteiston kotelointi, virtalähde, sekä tietoturvaan liittyvät ratkaisut. Valintaperusteissa painotettiin kenttäkäytön asettamia vaatimuksia: kustannustehokkuus, langattomuus, helppokäyttöisyys, kompakti rakenne sekä käyttövarmuus ulko-olosuhteissa. Kunkin osa-alueen suunnittelussa pyrittiin ratkaisuihin, jotka tukevat mahdollisimman vaivatonta käyttöönottoa sekä helppoa huollettavuutta ilman erityistä teknistä osaamista.

### 6.1 Mikrokontrollereiden vertailu

ESP32 on osa nykyaikaisten mikrokontrollerien sukupolvea, jotka tarjoavat sekä tehokkaan prosessointikyvyn että sisäänrakennetut langattomat yhteydet. ESP32:n kehitti Espressif Systems vuonna 2016, ja siitä tuli nopeasti suosittu erityisesti IoT-sovelluksissa sen edullisuuden, suorituskyvyn ja monipuolisuuden vuoksi (Espressif Systems, 2024).

ESP32:sta löytyy kaksiytiminen prosessori, useita GPIO-liitäntöjä, analogisia ja digitaalisia signaalinkäsittelyominaisuuksia sekä Wi-Fi- ja Bluetooth-yhteydet. Tämä yhdistelmä tekee siitä ihanteellisen valinnan sulautettuihin järjestelmiin, joissa tarvitaan sekä datankeruuta että tiedonsiirtoa.

Mikrokontrolleria valittaessa vertailussa olivat mukana myös Arduino Uno, Raspberry Pi Pico sekä STM32-sarjan mikrokontrollerit. Arduino Uno on tunnettu helppokäyttöisyydestään ja laajasta tukimateriaalistaan, mutta se ei tarjoa sisäänrakennettua langatonta yhteyttä eikä riitä suorituskyvyltään reaaliaikaiseen mittaukseen ja käyttöliittymän palvelemiseen. Raspberry Pi Pico tarjoaa ARM Cortex-M0+ ytimen ja toimii hyvin kevyissä projekteissa, mutta ilman lisämoduuleja se ei sisällä Wi-Fi-yhteyttä. STM32-sarjan mikrokontrollerit ovat tehokkaita ja ammattimaisessa käytössä suosittuja, mutta niiden ohjelmointi ja käyttöönotto vaatii

enemmän asiantuntemusta. Taulukossa 2 on esitetty näiden mikrokontrollerien ominaisuuksia vertailun tueksi.

TAULUKKO 2. Mikrokontrollereiden vertailu

Mikrokontrolleri	Prosessori	Wi-Fi	GPIO
ESP32	2x Xtensa 240 MHz	Kyllä	30+
Arduino Uno	1x ATmega328 16 MHz	Ei	14
Raspberry Pi Pico	2x Coretex-M0+ 133 MHz	Ei	26
STM32 (F103)	1x Coretex-M3 72 MHz	Ei	37

## 6.2 Liiketunnistin

Liiketunnistus on ajanottolaitteiston keskeinen osa, sillä se käynnistää ja pysäyttää mittauksen ilman käyttäjän manuaalista toimintaa. Tässä työssä käytetty HC-SR501 PIR-liiketunnistin (Passive Infrared) tunnistaa lämpösäteilyn muutoksia näkökentässään. Se reagoi liikkeisiin ja antaa digitaalisena ulostulona korkean signaalin, kun liike havaitaan.

Vertailussa pohdittiin myös anturien vaihtoehtoja ja mahdollisuutta käyttää kahden PIR-anturin sijasta yhdistettyä ratkaisuanturia (esim. IR-katkaisua). Kuitenkin PIR-anturin etuna on sen laaja saatavuus, yksinkertainen logiikka ja hyvä yhteensopivuus ESP32:n kanssa ilman monimutkaista kalibrointia.

PIR-anturin toimintaperiaate perustuu siihen, että kaikki lämpöä säteilevät kohteet, kuten ihmiset ja eläimet, lähettävät infrapunasäteilyä. Anturissa on kaksi pyroelektristä kennoa, jotka havaitsevat säteilyn muutoksen. Kun kohde liikkuu näkökentässä, toinen kenno havaitsee enemmän säteilyä kuin toinen, jolloin syntyy havaittava jännite-ero, joka laukaisee signaalin ESP32:lle (Adafruit, 2024).

PIR-antureilla on useita etuja: ne ovat halpoja, helposti saatavilla, vähän virtaa kuluttavia ja toimivat luotettavasti ulkona ja sisällä, kunhan sääolosuhteet eivät ole äärimmäisiä. Verrattuna valokennoihin, ultraääniantureihin tai laserpohjaisiin ratkaisuihin, PIR on yksinkertaisempi ja taloudellisempi.



KUVA 3. PIR-liiketunnistin

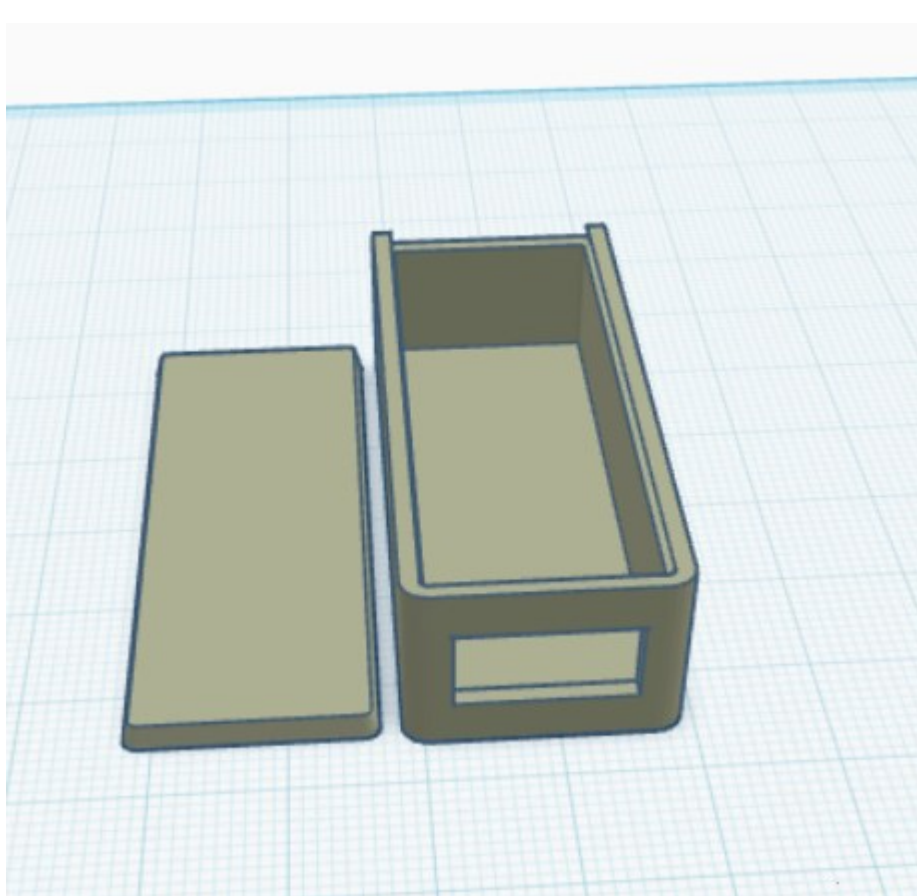
Valokennot ovat erittäin tarkkoja ja yleisesti käytetty esimerkiksi juoksukilpailuissa, mutta ne vaativat huolellista kohdistusta ja voivat häiriintyä kirkkaassa auringonvalossa. Ultraäänianturit mittaavat etäisyyksiä, mutta voivat heijastua väärin esimerkiksi karvaisista tai epätasaisista kohteista. Lasersensorit tarjoavat korkean tarkkuuden, mutta ovat kalliita ja herkkiä mekaanisille häiriöille.

Kokonaisuutena PIR-anturi on paras valinta tähän sovellukseen sen yksinkertaisuuden, edullisuuden ja riittävän herkkyyden vuoksi. Se ei tarvitse jatkuvaa kalibrointia, ja sen reaktioaika on riittävä rally-tokon kaltaiseen käyttöön, jossa huipputason millisekuntitarkkuus ei ole ratkaiseva, vaan johdonmukaisuus ja häiriötömyys. Anturin tunnistus etäisyys on säädettävissä 3-7 metrin välille, joka luo joustoa radalle tulemisen kulkukäytävän kokoon.

### 6.3 Kotelointi

Fyysisen laitteiston suunnittelussa korostui erityisesti kestävyys ja kompaktius. Koska ajanottojärjestelmä on tarkoitettu käytettäväksi ulkona, esimerkiksi hiekkakentillä, nurmikolla tai halliolosuhteissa, sen koteloinnin tuli suojata elektroniikkaa pölyltä, kosteudelta ja mekaanisilta iskuilta.

Kotelo suunniteltiin 3D-tulostettavaksi PLA-muovista, joka on kevyt mutta riittävän kestävä kohtuullisiin kenttäolosuhteisiin. Protomalli kotelosta näkyy kuvassa.



KUVA 4. Ajanottojärjestelmän 3D-tulostettava kotelo

ESP32 sijoitetaan kotelon keskiosaan, ja PIR-anturit asennetaan kotelon etusivulle, jolloin ne saivat hyvän näkökentän kohdetta kohti. Anturit kiinnitettiin liimalla ja ruuveilla, ja niiden johtimet suojattiin kutistesukalla.

Laitteiston koko pidettiin mahdollisimman pienenä: kokonaiskoko jäi alle 10 x 10 x 5 cm. Kansi on helposti irrotettavissa liu'uttamalla.

## 6.4 Akkuratkaisu

Koska järjestelmä on suunnattu ulkokentille, joissa sähköverkkoa ei välttämättä ole, akkuvirta oli ensisijainen virtalähde. Projektissa tutkittiin eri akkuteknologioita, kuten litiumioni (Li-ion), litium-polymeeri (LiPo) sekä 18650-sarjan sylinteriakut. Lopulliseksi valinnaksi tuli Li-ion-akku (esim. 18650-kenno), joka yhdistää hyvän kapasiteetin, uudelleenladattavuuden ja laajan saatavuuden.

ESP32:n tyypillinen virrankulutus vaihtelee käytön mukaan. Lepotilassa virrankulutus voi olla alle 1 mA, mutta aktiivisessa Wi-Fi-yhteydessä ja antureita käyttäessä se voi nousta yli 100 mA:iin. Käyttöaika riippuu akkujen kapasiteetista: yhdellä 3000 mAh akulla laite toimii useita tunteja yhtäjaksoisesti. Käyttöaikaa voidaan pidentää käyttämällä kahta akkua sarjassa tai parissa sekä optimoimalla ohjelmointia virransäästötilojen hyödyntämiseksi (esim. deep sleep -tila) (Random Nerd Tutorials, 2024).

Lataus toteutetaan mikro-USB-liitännän kautta, ja mukana on TP4056-latauspiiri, joka mahdollistaa turvallisen latauksen ja suojaa ylivirralla. Latauksen tilaa voidaan seurata yksinkertaisella LED-indikaattorilla tai tulevaisuudessa lisäämällä akun varaustason mittaus ja näyttö käyttöliittymään.

Aurinkopaneelin käyttö lisävirtalähteenä olisi mahdollisuus erityisesti pitkissä ulkokisoissa. Yksinkertainen 6V/1W-paneeli voisi pidentää akun kestoa useilla tunteilla, etenkin jos se yhdistetään vara-akkuun. Tällainen ratkaisu voisi olla hyödyllinen erityisesti paikoissa, laitteisto olisi kisapäivän aikana käytössä 5 tuntia, josta aktiivista aikaa olisi ainakin 4 tuntia.

Yhteenvedona voidaan sanoa, että akkupohjainen ratkaisu tekee järjestelmästä aidosti siirrettävän ja kenttäkelpoisen. Energiankulutuksen optimointi on mahdollista ohjelmallisesti, ja tarvittaessa järjestelmää voidaan laajentaa aurinkolatauksella tai vaihtoakulla.

Akkujärjestelmässä pohdittiin myös erillisen virransyöttömoduulin lisäämistä, joka näyttäisi jäljellä olevan kapasiteetin prosentteina käyttäjälle. Tämä voitaisiin toteuttaa OLED-näytöllä tai vaihtoehtoisesti käyttöliittymään lisättävällä mittarilla, joka laskee käyttöajan ja kulutuksen perusteella arvioidun varaustilan.

## 6.5 Tietoturva

Tietoturva on olennainen osa kaikkia langattomia järjestelmiä, myös sulautetuissa ratkaisuissa. Koska ESP32 toimii Access Point -tilassa ja tarjoaa avoimen käyttöliittymän, on tärkeää minimoida riski, että ulkopuolinen pääsee käsiksi laitteen hallintaan tai muokkaamaan tietoja.

Tässä järjestelmässä Wi-Fi-verkko on suojattu salasanalla (WPA2), mikä estää satunnaiset liittymisyrietykset. Käyttäjän tarvitsee kirjautua verkkoon käyttämällä tunnettuja tunnuksia. Vaikka kyseessä ei ole julkinen palvelin, paikallinen verkko voi silti olla haavoittuva, mikäli laitteeseen liitetään esimerkiksi tulosten vienti verkkoon tai pilvipalveluun.

Tällä hetkellä HTTP-pohjainen käyttöliittymä ei ole salattu (ei HTTPS), mikä tarkoittaa, että tietoliikenne ei ole salattu.

Lisäksi voidaan toteuttaa roolipohjainen käyttöoikeusmalli: esimerkiksi "tuomari" voisi nähdä ja nollata tulokset, mutta "katsoja"-käyttäjä ei voisi muuttaa tietoja. Tämä vaatisi käyttöliittymään kirjautumistoiminnon sekä sessionhallinnan. Tietoturvaa voi myös parantaa ottamalla käyttöön aikakatkaisun, jolloin laite sulkee yhteyden automaattisesti, mikäli käyttäjä ei toimi tietyssä ajassa. Tiedostojen muokkaus voidaan rajata vain fyysisesti kytkettävään käyttöön (esim. USB-kaapelin kautta), ja käyttöliittymässä voidaan näyttää vain tuloksia, ei hallintatoimintoja.

Vaikka tietoturvariskit ovat suhteellisen pieniä suljetussa kenttäkäytössä, on hyvä suunnitella laitteisto siten, että mahdollinen laajentaminen verkko- tai pilvipalveluun ei vaadi järjestelmän täydellistä uusimista. Näin kehitys voi jatkua hallitusti ja turvallisesti, jos järjestelmää otetaan käyttöön laajemmin tai virallisissa kilpailuissa.

## 7 KÄYTTÄJÄKYSELY

Järjestelmän suunnittelun tueksi toteutettiin sähköinen kysely, jonka tarkoituksena oli kartoittaa rally-toko seurojen kiinnostusta sähköisen ajanottojärjestelmän käyttöön sekä kerätä palautetta ja toiveita kehitystyöhön liittyen. Kyselyyn saatiin yhteensä yksitoista vastausta eri puolilta Suomea. Vastauksia tarkasteltiin sekä määrällisesti että sisällöllisesti. Kyselyä jaettiin Facebook-ryhmään, joka oli tarkoitettu rally-tokon koetoimitsijoille.

### 7.1 Seuratausta ja nykyiset käytännöt

Vastaajista kymmenen ilmoitti edustavansa monilajiseuraa, ja yksi toimi seurassa, joka järjestää vain yhden lajin toimintaa.

Kysyttäessä, onko seuroilla jo käytössä sähköistä ajanottojärjestelmää muissa lajeissa, vastaukset jakaantuivat. Useat vastaajat ilmoittivat käyttävänsä tai tuntevansa sähköiset järjestelmät esimerkiksi agilityssä. Toisaalta kolme seuraa ilmoitti, ettei sähköistä ajanottoa ole käytössä lainkaan.

### 7.2 Kiinnostus

Keskeinen kysymys oli, olisivatko seurat kiinnostuneita käyttämään sähköistä ajanottojärjestelmää nimenomaan rally-tokossa. Vastaukset osoittavat yleistä kiinnostusta. Kuusi vastaajaa ilmaisi kiinnostuksensa suoraan "Kyllä"-vastauksella tai vastaavalla, ja kaksi vastausta sisälsi ehdollisen kiinnostuksen, kuten "Jos se on helposti ja edullisesti toteutettavissa, niin miksipä ei". Kaksi vastaajaa suhtautui asiaan kielteisesti.

### 7.3 Palaute

Kyselyssä oli mukana myös avoimia kysymyksiä ja “Sana vapaa” -kenttä, joiden kautta saatiin arvokasta laadullista tietoa käyttäjien näkökulmista. Vastauksissa toistuivat erityisesti seuraavat teemat:

- Hinta ja saavutettavuus:

“Pieni seura eli hinta on tietysti isoin kysymys. Helppokäyttöisyys toinen eli sellainen mitä kuka tahansa pystyy käyttämään.”

- Kompakti koko ja helppo käyttö kisatilanteessa:

“Agilityn järjestelmä on iso, hankala ja työläs. Miten voisi järjestää ajanoton siten, että se on pieni ja helppo, edullinen eikä häiritse koirakkoa?”

- Hyödyt kyseenalaistettuina:

“Miksi tätä tarvittaisi? Yhden talkoilijan poisjätö ei ole iso etu.”

- Toiveet integraatiosta Virkku-järjestelmään:

“Kerääkö laitteisto talteen kelloitetut ajat, jotta ne voi sieltä noukkia tarvittaessa Virkkuun? Onko mietitty suoraan yhteyttä Virkkuun?”

### 7.4 Johtopäätökset kyselyyn

Kyselyn perusteella voidaan todeta, että sähköisen ajanottojärjestelmän käyttöönotolle rally-tokossa on aitoa kysyntää, mikäli ratkaisu on kustannustehokas, helppokäyttöinen ja fyysisesti kompakti. Erityisesti vapaaehtoispuolelta toimivien seurojen tarpeet ja resurssit on otettava huomioon, minkä vuoksi kehitetyn järjestelmän yksinkertaisuus ja edullisuus ovat keskeisiä vahvuuksia.

Opinnäytetyö kokonaisuutena mielestäni vastaa jo kysytyihin kysymyksiin.

Mutta tämän projektin aikana ei tulla toteuttamaan Virkku integraatiota, mutta jos tätä jatkojalostaisi tästä, niin se olisi ehdottomasti tärkeä ominaisuus, jonka haluaisin saada liitettyä.

## 8 JATKOKEHITYS

Yksi selkeä jatkokehitysmahdollisuus rally-tokon ajanottojärjestelmälle on sen integroiminen osaksi Suomen palveluskoiraliiton Virkku-järjestelmää. Virkku toimii keskitettynä alustana kilpailujen ilmoittautumisiin, tulosten tallentamiseen ja kilpailudatan hallintaan, ja sen käyttö on laajaa eri palveluskoiraliiton koiraurheilulajien keskuudessa. Integraation myötä sähköinen ajanotto ei olisi vain paikallinen työkalu, vaan osa valtakunnallista kilpailujärjestelmää.

Teknisesti ESP32-mikrokontrolleriin perustuva järjestelmä tarjoaa erinomaiset valmiudet tällaisen yhteyden toteuttamiseen. ESP32 tukee HTTP- ja MQTT-pohjaista tiedonsiirtoa, ja sen ohjelmointiin voidaan lisätä tuki REST-tyyppiselle rajapinnalle. Mikäli Virkku-järjestelmä tulevaisuudessa tarjoaa avoimen ohjelmointirajapinnan (API), ajanottotulokset voitaisiin lähettää automaattisesti pilvipalveluun tai suoraan Kennelliiton tietokantaan.

Integraation myötä ajanottolaite voisi lähettää tulokset oikeaan koirakkoon yhdistettynä ilman manuaalista välikäsitelyä. Tämä vähentäisi toimitsijoiden työkuormaa merkittävästi ja nopeuttaisi tulospalvelun läpimenoaikaa. Lisäksi tuloksia voitaisiin näyttää reaaliajassa esimerkiksi yleisölle mobiililaitteiden tai verkkosivun kautta, mikä parantaisi kilpailukokemusta ja läpinäkyvyyttä.

Toteutus vaatisi käytännössä kolme asiaa: Virkku-rajapinnan, tunnistautumismekanismin (esimerkiksi API-avain) sekä internet-yhteyden kisapaikalla. Nykyinen järjestelmä toimii Access Point -tilassa ilman ulkoista verkkoa, mutta ESP32 voidaan asettaa toimimaan myös Station-tilassa, jolloin se liittyy olemassa olevaan Wi-Fi-verkkoon tai mobiilitukiasemaan. Tämä mahdollistaisi suoran tiedonsiirron ulospäin.

Vaikka Virkku-integraatio ei ole osa tämän työn toteutusta, sen tekninen mahdollisuus on suunnittelussa huomioitu. Koko järjestelmä on rakennettu modulaarisesti siten, että myöhemmin lisättävät ominaisuudet – kuten JSON-tiedonsiirto, suoritusten tunnistus tai tulosten kuittaus – voidaan ohjelmallisesti ottaa käyttöön

ilman suuria muutoksia järjestelmän ytimeen. Mikäli Virkku avaa rajapintansa kehittäjille, integraatio voidaan toteuttaa tehokkaasti.

Yhteenvetona voidaan todeta, että Virkku-yhteensopivuus nostaisi ajanottojärjestelmän käytettävyyden uudelle tasolle. Se mahdollistaisi saumattoman kilpailuprosessin alusta loppuun ja vahvistaisi sähköisen ajanoton asemaa osana rallytokon tulevaisuutta. Erityisesti isompien kilpailujen ja useita ratoja samanaikaisesti pyörittävien seurojen näkökulmasta automaattinen tulosintegraatio on merkittävä askel kohti sujuvampaa ja modernimpaa kisatoimintaa.

## 9 KUSTANNUSARVIO

Kokonaiskustannuksia tarkasteltaessa voidaan todeta, että järjestelmän toteuttaminen on varsin edullista verrattuna moniin kaupallisiin ajanottoratkaisuihin. Laskelemien perusteella kokonaisuus voidaan rakentaa noin 35–40 euron investoinnilla (TAULUKKO 2). Kustannuksista merkittävimmän osuuden muodostavat keskeiset elektroniikkakomponentit, eli mikrokontrolleri ja liiketunnistimet. Yksittäisten komponenttien hinnat vaihtelevat hieman toimittajasta ja ostopaikasta riippuen, mutta hintataso pysyy johdonmukaisesti edullisena eurooppalaisilla jälleenmyyjillä.

Akkuratkaisun ja siihen liittyvän latausmoduulin osuus kustannuksista on suhteellisen pieni, mutta niiden valinnalla on suuri merkitys laitteen käytettävyyteen. Myös koteloinnin kustannukset jäävät vähäisiksi, sillä 3D-tulostamalla valmistettu kuori voidaan toteuttaa muutaman euron filamenttikustannuksella. Pienet tarvikkeet, kuten johdot, kutistesukat ja kiinnitysruuvit, muodostavat vain marginaalisen osuuden kokonaiskustannuksista.

Kun kokonaisuutta arvioidaan, voidaan todeta, että kustannusrakenne on hyvin linjassa työn alkuperäisen tavoitteen kanssa: ratkaisu on aidosti kustannustehokas ja mahdollistaa teknologian hyödyntämisen myös pienillä, vapaaehtoistoimintaan perustuvilla seuroilla. Kustannukset eivät muodosta estettä järjestelmän käyttöönotolle, vaan hinta jää huomattavasti matalammaksi kuin perinteisissä, kaupallisissa ajanottolaitteissa, joiden hinnat nousevat usein satoihin euroihin. Näin ollen prototyypin kustannusarvio osoittaa, että järjestelmän toteuttaminen on realistista ja taloudellisesti perusteltua. Lisäksi se luo hyvän pohjan jatkokehitykselle, sillä tulevat laajennukset, kuten aurinkopaneelin lisääminen tai integrointi kilpailunhallintajärjestelmiin, voidaan toteuttaa ilman merkittävää kustannusten kasvua.

TAULUKKO 2.

Komponentti	Hinta	Ostopaikka
ESP32	11,50e	Botland store
PIR-liiketunnistin	2x 5,95e	Elektor
Li-ion akku 18650	0-80-1.35e	Nkon
TP4056 latausmoduuli	4.06e	Mouser
PLA-filaamentti	4e	3D Fuel
Pienet komponentit (johdot yms)	~3,00e	Triopak
Mikro-USB-kaapeli	2e	Prisma

Edellä esitetty kustannusarvio osoitti, että järjestelmän rakentaminen on mahdollista noin 40 euron materiaalikuluilla. Tämä kuvaa hyvin prototyypin ja harrastelijatason toteutusta, jossa työtä ei hinnoitella. Mikäli laite kuitenkin tuotteistettaisiin ja saatettaisiin markkinoille valmiina tuotteena, kustannusrakenne muuttuisi olennaisesti.

Ensimmäinen merkittävä lisätekijä on työ. Laitteen kokoaminen, ohjelmointi, testaus ja pakkaus. Jos työn kustannus sivukuluineen arvioidaan 20–25 euroksi tunnilta. Tämän lisäksi yritystoimintaan sisältyy yleiskuluja, kuten hallinto, työkalut, varastointi ja asiakaspalvelu, jotka katetaan tuotteiden myyntihinnoista. Käytännössä tämä tarkoittaa, että valmistuskustannusten päälle lisätään usein 50–100 prosentin kate, jotta toiminta olisi kannattavaa.

Kun materiaalikulut (40 €) ja työ (45 €) yhdistetään, muodostuu tuotteen valmistuskustannukseksi noin 85 euroa. Kun tähän lisätään yrityksen kate ja yleiskulut (esimerkiksi 60 %), nousee välisumma noin 135 euroon. Lopullinen kuluttajahinta sisältää lisäksi arvonlisäveron (25.5 %), mikä nostaa hinnan arviolta noin 170 euroon.

Näin ollen voidaan todeta, että mikäli järjestelmä tulisi myyntiin valmiina tuotteena, sen kuluttajahinta asettuisi todennäköisesti noin 200 euron haarukkaan. Tämä on moninkertainen verrattuna itse rakennettuun prototyyppiin, mutta edelleen selvästi edullisempi kuin monet markkinoilla olevat kaupalliset ajanottoratkaisut, joiden hinnat nousevat satoihin euroihin.

## 10 YHTEENVETO

Tässä opinnäytetyössä suunniteltiin kustannustehokas, kompakti ja langattomasti toimiva ajanottojärjestelmä rally-tokokilpailuja varten. Perinteinen manuaalinen ajanotto korvattiin ESP32-mikrokontrolleriin perustuvalla sähköisellä ratkaisulla, joka hyödyntää liiketunnistimia ajanoton käynnistämiseen ja pysäyttämiseen ilman käyttäjän aktiivista osallistumista.

Järjestelmä tallentaa mittaustulokset ESP32:n sisäiseen muistiin CSV-muodossa ja tarjoaa selainkäyttöliittymän, jonka avulla tulokset voidaan tarkastella ilman ulkoista internetyhteyttä. Tärkeimpinä tavoitteina olivat alhaiset kustannukset, pieni fyysinen koko ja mahdollisuus käyttää laitteistoa kenttäolosuhteissa akkuvirralla. Ohjelmointi toteutettiin Arduino IDE -ympäristössä käyttäen C++-kieltä ja avoimen lähdekoodin kirjastoja, kuten ESPAsyncWebServer. Järjestelmä on laajennettavissa esimerkiksi tulosten siirtoon kilpailunhallintajärjestelmiin, kuten Virkuun.

Projektiin lopputulos täytti asetetut tavoitteet, ja taloudellisesti halvempi ratkaisu olisi toteutettavissa tämän opinnäytetyön pohjalta. Jatkokehityskohteita ovat järjestelmän automaattinen synkronointi ulkoisiin tietojärjestelmiin.

Yhteenvetona voidaan todeta, että järjestelmän toteutus vastaa hyvin rally-tokon vaatimukseen: se on nopea asentaa, helppo käyttää ja tarkka mittauksissaan. Lisäksi se vähentää toimihenkilöiden kuormitusta, parantaa kisojen toistettavuutta ja luo mahdollisuuden tarkempaan tulosseurantaan.

Projekti toimi samalla myös oppimiskokemuksena sulautettujen järjestelmien ja IoT-tekniologioiden soveltamisesta käytännön ongelmanratkaisuun. Työn aikana syvennyttiin sekä laitteistojen valintaan, virransyöntiin että käyttöliittymäkehitykseen. Näiden osa-alueiden hallitseminen on suoraan sovellettavissa myös muihin automaattioratkaisuihin – esimerkiksi teollisuuteen, ympäristön seurantaan tai etäohjattaviin järjestelmiin.

Jatkokehityksessä tärkeintä olisi parantaa järjestelmän vikasietoisuutta, lisätä automaattista raportointia ja mahdollistaa tiedonsiirto suoraan kilpailunjärjestäjien

järjestelmiin. Integraatio, päivitysmahdollisuus (OTA) ja tietoturva ovat osa-alueita.

## 11 LIITEET

LIITE 1. Koodi

```
#include <WiFi.h>
#include <ESPAsyncWebServer.h>
#include <FS.h>
#include <SPIFFS.h>

const int startPin = 14; // Lähtöanturi
const int stopPin = 27; // Maalianturi

bool timerRunning = false;
unsigned long startTime = 0;
unsigned long endTime = 0;

// WiFi-asetukset (ESP32 toimii Access Pointina)
const char* ssid = "AjanottoESP";
const char* password = "salasana123";

// Tiedostopolku SPIFFS:ssä
const char* fileName = "/tulokset.csv";

// Web-palvelin
AsyncWebServer server(80);

void setup() {
  Serial.begin(115200);

  // Anturit
  pinMode(startPin, INPUT);
  pinMode(stopPin, INPUT);

  // Käynnistä SPIFFS
  if (!SPIFFS.begin(true)) {
```

```

Serial.println("SPIFFS alustus epäonnistui.");
return;
}

// Käynnistä Wi-Fi Access Point
WiFi.softAP(ssid, password);
Serial.print("Yhdistä verkkoon: ");
Serial.println(ssid);
Serial.print("IP-osoite: ");
Serial.println(WiFi.softAPIP());

// Luo web-sivu
server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
  File file = SPIFFS.open(fileName);
  if (!file) {
    request->send(200, "text/plain", "CSV-tiedostoa ei löytynyt.");
    return;
  }

  String html = "<html><head><meta charset='utf-8'><title>Ajanottotulokset</ti-
tle></head><body><h2>Ajanottotulokset</h2><table
border='1'><tr><th>#</th><th>Aika (ms)</th></tr>";
  int i = 1;
  while (file.available()) {
    String line = file.readStringUntil('\n');
    html += "<tr><td>" + String(i++) + "</td><td>" + line + "</td></tr>";
  }
  html += "</table></body></html>";
  request->send(200, "text/html", html);
  file.close();
});

server.begin();
}

```

```

void loop() {
  // Lähtö
  if (!timerRunning && digitalRead(startPin) == HIGH) {
    startTime = millis();
    timerRunning = true;
    Serial.println("Liike havaittu lähtöalueella – ajanotto käynnissä");
    delay(500); // Estä tuplahavainto
  }

  // Maali
  if (timerRunning && digitalRead(stopPin) == HIGH) {
    endTime = millis();
    unsigned long elapsed = endTime - startTime;
    Serial.print("Maali saavutettu. Aika: ");
    Serial.print(elapsed);
    Serial.println(" ms");

    // Tallenna CSV
    File file = SPIFFS.open(fileName, FILE_APPEND);
    if (file) {
      file.println(elapsed);
      file.close();
    }

    timerRunning = false;
    delay(2000); // Pieni viive ennen uutta mittausta
  }
}

```

[Forms-kyselyn vastaukset](#)

## 12 LÄHTEET

The Engineering Projects. ESP32 Pinout, Datasheet, Features & Applications. Viitattu 20.4.2025. Saatavilla: <https://www.theengineeringprojects.com/2020/12/esp32-pinout-datasheet-features-applications.html>

GitHub. ESP-WROOM-32-Doc. Viitattu 20.4.2025. Saatavilla: <https://github.com/anandsummer28/ESP-WROOM-32-Doc>

Wikipedia. ESP32. Viitattu 20.4.2025. Saatavilla: <https://en.wikipedia.org/wiki/ESP32>

Wilson, J. ESP-WROOM-32 Pinout Diagram. Viitattu 20.4.2025. Saatavilla: <https://randomnerdtutorials.com/esp32-pinout-reference-gpios/>

Rui Santos ja Sara Santos. ESP32 Async Web Server – ESPAsyncWebServer Library. Random Nerd Tutorials. Saatavilla: <https://randomnerdtutorials.com/esp32-async-web-server-espasyncwebserver-library/>. Luettu 21.6.2025.

Adafruit Industries. (2024). *Using a PIR w/Arduino – PIR Motion Sensor Guide*. Adafruit Learning System. Viitattu 3.7.2025. Saatavilla: <https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/using-a-pir-w-arduino>

Random Nerd Tutorials. (2024). *ESP32 Deep Sleep with Arduino IDE and Wake Up Sources*. Viitattu 3.7.2025. Saatavilla: <https://randomnerdtutorials.com/esp32-deep-sleep-arduino-ide-wake-up-sources/>