

KYMENLAAKSON AMMATTIKORKEAKOULU
Tietotekniikan koulutusohjelma / ohjelmistotekniikka

Miikka Mäkinen

MOOTTORIOHJATTU KAMERAKÄÄNTÖPÄÄ

Opinnäytetyö 2011

TIIVISTELMÄ

KYMENLAAKSON AMMATTIKORKEAKOULU

Ohjelmistotekniikka

MÄKINEN, MIIKKA

Moottoriohjattu kamerakääntöpää

Opinnäytetyö

37 sivua + 9 liitesivua

Työn ohjaaja

Yliopettaja Paula Posio

Toimeksiantaja

Rai-Kuva Ky

Lokakuu 2011

Avainsanat

kameravalvonta, kamerat, ohjelmointi,
sulautettu tietotekniikka, mikro-ohjaimet

Kamerakääntöpääksi kutsutaan moottoroitua yksikköä, jolla siihen liitettyä kameraa voidaan kääntää haluttuun suuntaan, yleensä vaaka- ja pysty akselien suhteen. Tällaisia apuvälineitä on pienin eroavaisuuksin ollut käytössä sekä teollisuuden videovalvontaa että isompien studioiden tuotantojärjestelmissä.

Opinnäytetyön tavoitteena oli suunnitella ja toteuttaa Haminan Teatterin käyttöön edullinen moottoriohjattu kääntöpää videokuvauksessa hyödynnettäville järjestelmäkameroille.

Ennen työhön ryhtymistä useiden olemassa olevien järjestelmien teknisiä yksityiskohtia analysoitiin eri menetelmin, muun muassa takaisinmallinnuksella. Analyysi ulottui ohjelmistoteknisten näkökulmien lisäksi myös mekaanisiin sekä sähkötekniisiin ratkaisuihin. Esikeskusteluissa asetettiin täytettävät vaatimukset ja käsiteltiin niiden toteuttamismahdollisuuksia. Tehdyn esitutkimuksen perusteella syntyi päätös aloittaa järjestelmän suunnittelutyö.

Tuloksena valmistuivat toteutusvalmiit mallit kääntöpään prototyypille. Samalla työ osoitti uusien työkaluohjelmistojen ja kehitysympäristöjen korkean laadun. Kun tämän lisäksi otetaan huomioon Internetissä prototyypipalveluita tuottavan sektorin lisääntynyt tarjonta ja siitä seurannut hintakilpailu, yksityisillä henkilöillä on yhä paremmat edellytykset haastaa olemassa olevia kaupallisia tuotteita jäämättä laadullisesti altavastajiksi.

ABSTRACT

KYMENLAAKSON AMMATTIKORKEAKOULU

University of Applied Sciences

Software Engineering

MÄKINEN, MIIKKA

Bachelor's Thesis

Supervisor

Commissioned by

October 2011

Keywords

Motorized Pan/Tilt Unit for a SLR Camera

37 pages + 9 pages of appendices

Paula Posio, Senior Lecturer

Rai-Kuva Ky

camera surveillance, cameras, programming,
embedded information technology, microcontrollers

A Motorized pan/tilt unit is a device used to control the vertical and horizontal rotation of the camera attached to it. These kinds of units are fairly usual in video surveillance and studio systems.

The main objective of this final year project was to design and implement a cost effective motorized pan/tilt unit for an SLR camera to be used by the Theater of Hamina.

The technical details of the existing systems were analyzed with various methods, including reverse-engineering. In addition to that, an extended analysis of mechanical and electrical solutions was carried out. The general requirements were set up on the pre-conversations, and the decision to begin the system development was made as a result of the pre-research.

As a result, an implementation-ready motor head design was produced. This project also proved the high quality of modern integrated development environments and designing tools. This, combined with the fact that there is a growing number of prototype service providers available on the Internet and the prices are getting lower, makes it possible for almost everyone to challenge the commercial products without having to automatically be the underdog in a qualitative sense.

ALKUSANAT

Tahtoisin kiittää KyAMK:n tekniikan toimipisteen opettajia, perusaineiden opettajista alakohtaisiin. Lisäkiitos heille, jotka uskaltavat vielä vaatia oppilailtaan tuloksia. Siitä aiheutuneet panostukset ensimmäisten vuosien perusopintoihin, etenkin matemaattisten aineiden kohdalla, loivat vakaan pohjan, johon jatko-opinnoissa oli helppo tukeutua.

Henkilökohtainen kiitos entiselle koulutusohjelmavastaavalle Teemu Saarelaiselle, jonka myötävaikutus mahdollisti suuntautumisen projektiopinnoissani sulautettujen järjestelmien ohjelmointiin ja toteutukseen.

Kiitos myös Visi-Systems Oy:n Ilkka Sorsalle ja Jouni Wichille. He lainasivat prototyypisuunnittelussa käytetyt kameraohjaimet, digitaalioskilloskoopit ja muut mittalaitteet sekä rauhallista toimistotilaa tulosten analysointiin.

Tärkein viimeiseksi. Heidi, luulit että unohtaisin. Kiitos kaikesta ja Douglas Adamsin sanoin - Terve ja kiitos kaloista!

Miikka Mäkinen

Haminassa 17.10.2011

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

ALKUSANAT

TERMIT JA LYHENTEET

7

1 JOHDANTO

10

1.1 Taustatietoja

10

1.2 Työn rajaus

11

2 KÄÄNTÖPÄÄKAMERAJÄRJESTELMÄT

12

3 SULAUTETUT JÄRJESTELMÄT

14

4 VALITUT RATKAISUT

16

5 TOTEUTUS

17

5.1 Sähkötekniset ratkaisut

17

5.2 Mekaaniset ratkaisut

20

5.3 Ohjelmistotekniset ratkaisut

23

5.3.1 Esimerkki 1, SerialClass-luokan funktio pelcoCheck

26

5.3.2 Esimerkki 2, MemoryClass-luokan funktio EEPROMRead.

26

5.3.3 Esimerkki 3, MotorClass-luokan funktio motorRun

27

6 ARVIOINTI JA JOHTOPÄÄTÖKSET

30

6.1 Ongelmakohdat

30

6.2 Mitä opittiin?

31

6.3 Mitä muuttaisin?

31

6.4 Johtopäätökset

32

7 PELCO PROTOKOLLAN DOKUMENTAATIO

33

8 YHTEENVETO

34

LÄHTEET

35

LIITTEET

Liite 1. How to Obtain and Use Pelco Protocols

Liite 2. Kamerakääntöpään sähkökuvat

Liite 3. 3D-kuvat mekaanisista ratkaisuista

Liite 4. Ohjelmiston luokkakaavio

Liite 5. "Preset"-toiminnon sekvenssikaavio

TERMIT JA LYHENTEET

Arduino

Arduino on yhdysvaltalaisen Atmel Corporationin mikrokontrollereihin pohjautuva avoimen lähdekoodin prototyypialusta, joka perustuu joustavaan ja helppokäyttöiseen laitteistoon ja ohjelmistoon. [1]

askelmoottori

Sähkömoottorin tyyppi, jossa pyöriminen toteutetaan askelittain eli steppeinä.

assembly

Matalan tason ohjelmointikielten alaluokka. Synonyymi termille "symbolinen konekieli".

breakout board

Piirilevy, johon integroidun hyvinkin pienen komponentin kytkennät on levitetty fyysisesti laajemmalle alueelle helpommin hyödynnettäviksi sähköisiksi kytkentäpisteiksi.

CCTV

(Closed-circuit television) Tarkoittaa analogisten videokameroiden hyödyntämistä suljetun verkon kuvansiirrossa. Termiä käytetään erityisesti puhuttaessa valvontakamerajärjestelmistä.

IDE

(Integrated development/design environment) ohjelmointiympäristö

NEMA

(National Electrical Manufacturers Association) yhdysvaltalainen järjestö, jonka määrittelemillä standardeilla kuvataan esimerkiksi askelmoottorien fyysistä kokoa. NEMA17 tarkoittaa 1,7" x 1,7", NEMA24 2.4" x 2.4" kokoista moottorin etulevyä. Tämä ei kuitenkaan aseta rajoituksia moottorin pituudelle.

optohaarukka

Komponentti, jossa on samassa rakenteessa sekä fototransistori että valodiodeja. Rakenteen välissä on hahlo, johon osuva este katkaisee fototransistorille suunnatun valonsäteen. Ratkaisu toimii rajakytkimen tapaan, mutta ilman mekaanisesti kuluvia osia.

parikaapeli

Yleinen kaapelityyppi, jossa käytetään häiriöiden minimoimiseksi yhtä tai useampaa paria toistensa ympäri kierrettyjä johtimia.

Pelco

Nykyisin Schneider Electric S.A.:n alaisuudessa toimiva videovalvontasektorin yritys, jolla on vakiintunut asema alan ratkaisujen tuottajana.

Pelco D

Yleisesti käytössä oleva Pelcon (ks. edel.) kehittämä käännettävien valvontakameroiden ohjausprotokolla.

Pololu

Pololu Robotics And Electronics, Inc. on Yhdysvalloissa sijaitseva sulautettuihin järjestelmiin erikoistunut yritys, jonka kehittämiä mikrokontrollereiden kytkentäalustoja ja koekytkentälevyjä käytettiin pohjana tämän opinnäytetyön projektissa.

PTZ

(Pan, Tilt ja Zoom) Käytetään yleisesti kuvaamaan valvontakameran ominaisuuksia tarkoittaen, että kamera on etäohjauksella käännettävissä ja että kameran objektiivin polttoväli on myös muutettavissa.

RS-485, RS-232

(Recommended Standard) RS-485 ja RS-232 ovat Electronics Industries Associationin standardeja, jotka muun muassa määrittelevät näiden sarjaliikennöintityyppien sähköiset ominaisuudet. RS-485 on tyypillisesti käytössä teollisuudessa (esimerkiksi kameroiden ohjauksessa) ja RS-232 on tuttu esimerkiksi kotitietokoneiden sarjaliikenneporteista.

SVN

SVN eli Subversion on ohjelmistokehityksessä usein käytössä oleva versionhallintajärjestelmä.

Trac

Trac Integrated SCM & Project Management on avoimeen lähdekoodiin perustuva projektinhallintaohjelmisto.

1 JOHDANTO

Tämän opinnäytetyön aihe on "Moottorihjattu kamerakääntöpää". Kääntöpää suunniteltiin puomikameran ohjaukseen Haminan Teatterin käyttöön. Tilaajana projektissa toimi Rai-Kuva Ky:n Vesa Hovi.

Valmiiden suunnitelmien lisäksi työn päämääränä oli osoittaa, että joissakin tapauksissa ennen monimutkaisia kytkentöjä ja erikoiskomponentteja vaatineet järjestelmät on mahdollista toteuttaa uusilla nopeilla mikrokontrollereilla ja huomattavasti yksinkertaisemmilla sähköisillä kytkennöillä. Toiveena on tätä kautta osaltaan lisätä alan opiskelijoiden kiinnostusta sulautettuihin järjestelmiin ja niiden ohjelmointiin sekä innostusta etsiä ja toteuttaa itse vaihtoehtoisia ratkaisuja kaupallisille tuotteille.

1.1 Taustatietoja

Haminan Teatteri toteuttaa kokeellisia produktioita, joiden tallentamiseen käytetään uusiin järjestelmäkameroihin sisällytettyjä videokuvausominaisuuksia ja korkealuokkaisia objektiiveja. Syksyllä 2010 avuksi hankittiin kamerapuomi, jolla kameralle saatiin lisää ulottuvuutta ja toivottua ammattimaisuutta kuva-ajoihin.

Tässä vaiheessa Vesa Hovi kuuli, että etsin opinnäytetyön aihetta, ja otti yhteyttä. Kamerapuomin päähän sijoitettavan kameran liikkeisiin tahdottiin lisää tarkkuutta, vaakataason ja pystytason käännöt ja, mikäli mahdollista, alhaalta ohjattava objektiivin tarkennus. Kaluston osalta tahdottiin varautua mahdollisiin tulevaisuudessa toteutettaviin lyhytelokuviin. Vaikka tähän tarkoitukseen oli olemassa valmiita studiokäyttöön suunniteltuja videokameroiden kääntökehikoita, niiden rakenteellinen laatu tai ominaisuudet eivät vastanneet käyttötarkoitusta. Tarvittavien ominaisuuksien löytyessä myös ostohinta nousi harrastelijateatterin ulottumattomiin, noin kahteen tuhanteen euroon ja yli. [2]

Syntyi idea perinteiseen valvontakameratekniikkaan perustuvan kääntöpään prototyypin toteuttamisesta. Tämän myötä esimerkiksi edullisia kolmansien osapuolten ohjaimia voitaisiin hyödyntää, jolloin järjestelmän oheislaitteiden kustannukset saataisiin painettua alas. Myöskään itse kääntöpään teknisissä ratkaisuissa ei tarvitsisi tyytyä niin suuriin kompromisseihin, kun projektiin sijoitettava pääoma kohdistuisi suoraan komponentteihin ja materiaaleihin. Valmiin tuotteen kuluiksi arvioitiin alustavasti noin 500 euroa.

1.2 Työn rajaus

Työssä käsitellään kääntöpään sähkösuunnitelmia, mekaanisia ratkaisuja ja ohjelmiston toteutusta.

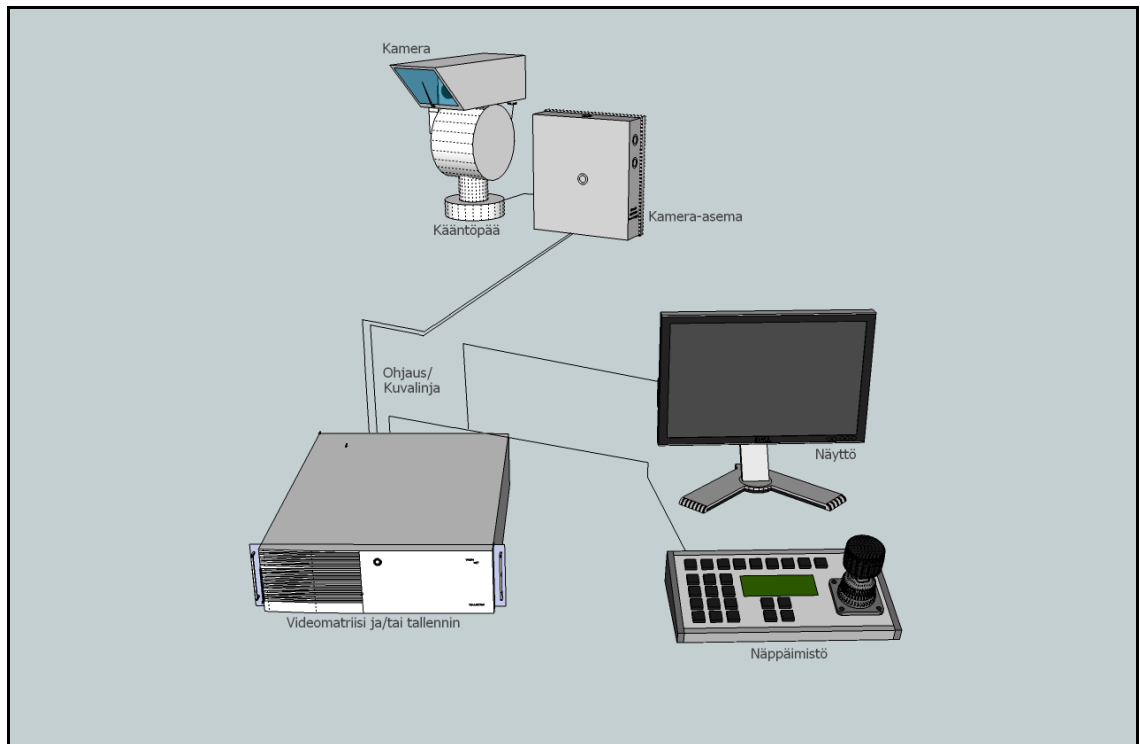
Sähkösuunnitelmia on koestettu käytännössä niiltä osin, kuin Pololun suunnittelemat prototyyppialustat niitä vastaavat.

Ohjelmisto on testattu ja läpikäyty niiltä osin, kuin se saatavilla olevilla komponenteilla oli mahdollista. Järjestelmä ohjautuu testatulla kolmannen osapuolen ja Pelcon alkuperäisellä näppäimistöllä. Samoin moottoriohjaimille ajettava pulssitieto on todettu mittalaitteilla paikkansa pitäväksi.

Mekaanisten ratkaisujen pohjalta ei ole toteutettu prototyyppiä, eikä työn suunnittelussa esille nousseiden mekaanisten ratkaisujen toteutuskelpoisuuteen voida siksi ottaa ehdotonta kantaa. Sama koskee luvattujen spesifikaatioiden paikkansa pitävyyttä askelmoottoreiden osalta.

2 KÄÄNTÖPÄÄKAMERAJÄRJESTELMÄT

Perinteisillä analogisilla kamerajärjestelmillä on pitkä historia ja eri valmistajien laitteiden keskinäinen yhteensopivuus on nykyisin kohtalaisen hyvä. Kuvassa 1 on esimerkki ns. CCTV-videovalvontajärjestelmästä.



Kuva 1. Perinteisen videovalvontajärjestelmän komponentit

Näppäimistöllä välitetään komennot kääntöpääkameralle ja videomatriisille.

Videomatriisi toimii suuremmissa järjestelmissä ohjaajana, joka reitittää valitut kamerakuvat valittuihin monitoreihin. Pienemmissä, mutta yli yhden kameran järjestelmissä erillistä videomatriisia ei tarvita, vaan mainitut toiminnot voidaan yleensä hoitaa tallentimen avulla. Yhden kameran järjestelmissä näitä kamerakuvia uudelleen reitittäviä toimintoja ei käytetä.

Yleensä kääntöpää ja sen moottoreita ohjaava kamera-asema ovat olleet erillisiä yksiköitä. Tosin viime vuosina integroidut mallit ovat nousseet kilpailemaan perinteisten mallien rinnalle asennuksen ja kaapeloinnin helppoudella.

Vaikka videovalvontajärjestelmien kuvansiirrossa on miltei alusta alkaen ollut voimassa eri valmistajien laitteiden yhteensopivuuden takaavat standardit, ohjauslinjojen osalta tilanne ei ole ollut sama. Jokaisella valmistajalla oli omansa, yleensä RS-485-väylään perustuva mutta erilaisella viestirakenteella toteutettu ohjausprotokolla. Mikäli kahden eri valmistajan järjestelmiä yhdistettiin, niiden välisiä viestejä tulkitsivat kalliit protokollamuuntimet.

Vuonna 2001 Pelco otti askeleen kohti avoimuutta ja julkisti järjestelmissään käyttämänsä Pelco D -ohjausprotokollan. Samalla sallittiin, että kolmannet osapuolet voivat käyttää protokollaa. [Liite 1]

Tämä oli ratkaiseva käännekohta, sillä vaikka viesteiltään salaamaton ohjauskäskykanta on helppo analysoida ja dokumentoida sarjaliikenteen takaisinmallinnuksella (menetelmä testattiin ja todettiin toimivaksi tämän projektin yhteydessä), sitä ei voida luvatta hyödyntää kaupallisissa tuotteissa.

Tuolla julkistuksella Pelco sai protokollalleen erityisaseman ulkopuolisten kehittäjien tuotteissa ja varmisti samalla, että useimmista nykyään myytävistä kameraohjausjärjestelmistä löytyy jonkinasteinen tuki sen omille tuotteille Pelco D -protokollan muodossa.

Viime vuosina osa suurista toimijoista on toteuttanut vastaavanlaisia julkistuksia huomattavasti nykyaikaisemmillekin standardeilleen, mutta Pelco D -protokollan asema lähestulkoon alan standardina on pysynyt muuttumattomana.

3 SULAUTETUT JÄRJESTELMÄT

"Sulautettu järjestelmä on rajattuun tarkoitukseen tehty laite, jossa on pieni tietokone." [3,9]

"Useimmat sulautetut järjestelmät käsittelevät tarvittavat tiedot välittömästi. Sulautetut järjestelmät ovat tyypillisesti reaktiivisia järjestelmiä. Ne ovat jatkuvassa interaktiossa ympäristönsä kanssa ja reagoivat ympäristön määräämällä tahdilla." [3,9]

On tapauksia, joissa järjestelmän lokerointi sulautettuihin järjestelmiin ja perinteisiin tietokoneisiin voi olla hankalaa. Mikäli järjestelmällä voi tehdä melko rajatun skaalan ennalta määrättyjä toimintoja, joiden muuttaminen toisenlaisiksi ei ole erityisen helppoa, sen voidaan katsoa olevan sulautettu järjestelmä. [3,9]

Puhuttaessa sulautetuista järjestelmistä ohjelmoijan näkökulmasta kooditason toteutuksessa on huomattavasti tiukemmat toimintarajat verrattuna esimerkiksi perinteisiin x86-alustoilla toteutettaviin projekteihin.

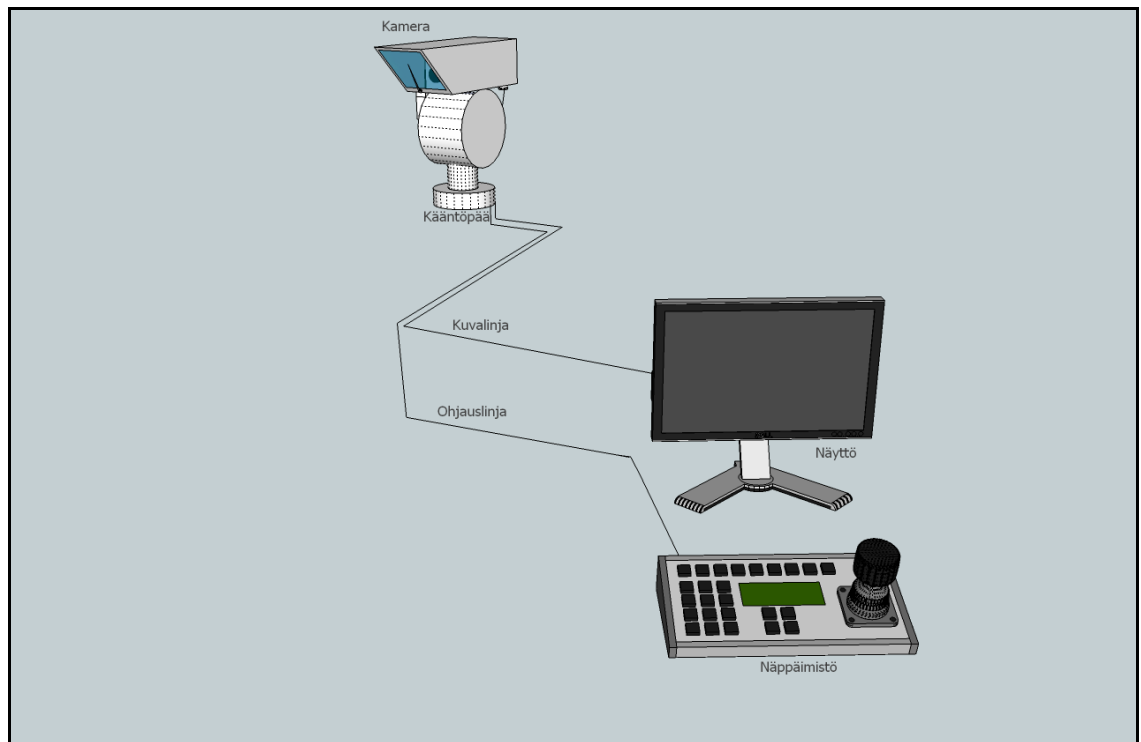
Käytössä olevan muistin rajallisuus, toteutettavan koodin viiveettömyys ja mahdollisimman korkea tehokkuus ovat puoltaneet tiettyjen usein toistettavien osien optimointia assembyllä. Myös globaalit muuttujat ovat olleet suosittuja johtuen niihin kohdistuvien viittausten nopeudesta ja helppoudesta.

Voidaankin todeta, että PC-maailmassa pitkään käytössä olleen validin luokkaperustaisen koodin tuottaminen on haastavaa sulautetuissa ympäristöissä. Vasta suurta suosiota saavuttaneen Arduinon sulautettujen alustojen ja kehitysympäristön ilmestyminen markkinoille yhdistettynä uusiin tehokkaisiin mikroprosessoreihin on auttanut keskivertoharrastajien ajatusmaailman siirtämisessä luokkatyyppisen ajattelun suuntaan.

Mikrokontrollereiden laskentatehon lisääntyessä korkein prioriteetti ei enää ole välttämättä koodin tehokkuus, sillä suoritusnopeus on huippuunsa virittämättömälläkin koodilla riittävän korkea. Ennen vähemmän tärkeät seikat, kuten koodin luettavuus ja rakenteen selkeys, ohjelmointityön järkevä aikataulutus ja mahdollisen interaktiivisen järjestelmän käytettävyys ovat alkaneet painaa vaa'assa enemmän kuin äärimmäisyyksiin viety optimointi.

4 VALITUT RATKAISUT

Kuvasta 2 ilmenee, miten kääntöpää päätettiin toteuttaa käyttäen perinteisiä videovalvontajärjestelmissä esiintyviä ratkaisuja mutta poistamalla ketjusta kaikki ylimääräiset komponentit ja sijoittamalla kamera-asema kääntöpään kanssa samaan yksikköön.



Kuva 2. Videovalvontajärjestelmän komponentit, minimitoteutus

Sarjaliikennemuodoksi valittiin kameraohjaimien kanssa yhteensopiva RS-485 ja ohjausprotokollaksi Pelco D.

Mikrokontrolleriksi valittiin Atmel Corporationin atmega328-piiri sijoitettuna Pololun koekytentälevylle. Yhdistelmä oli sähköisesti riittävän lähellä Arduinon valmiita piirilevyjä, joskin fyysisesti huomattavasti pienempi. Kyseiselle mikrokontrollerille löytyi myös suora tuki Arduino IDE:stä. Lisäeduksi luettiin moduulinen ja yksinkertainen rakenne sekä alustaan, jolla on laaja käyttäjäkunta, liittyvä ryhmäpotentiaali, mikäli tuote päätettäisiin siirtää julkiseksi avoimen lähdekoodin toteutukseksi.

5 TOTEUTUS

5.1 Sähkötekniset ratkaisut

Alusta asti oli selvää, että kääntöpäätä käytettäisiin studio-olosuhteiden lisäksi ulkona, mahdollisesti haastavissakin ympäristöissä ja akkukäytöllä. 230 voltin järjestelmä ja sen vaatima invertteri suljettiin pois laskuista huonon hyötysuhteen ja käytöstä aiheutuvan heikentyneen sähköturvallisuuden vuoksi. Suoraan matalajännitteellä toimiva kääntöpää olisi huomattavasti turvallisempi ja mahdollistaisi pidemmät käyttöajat akustolla sekä yksinkertaistaisi kytkentää.

Käyttövoimaksi valittiin perinteinen lyijyakku nykyisistä akkutekniikoista parhaan "Ah/euro"-suhteen, sen muita tekniikoita paremman pakkaskestävyyden ja helpon saatavuuden vuoksi. Jännitteeksi valittiin mahdollisimman korkea mutta yleinen ja yhdellä akkuyksiköllä saavutettavissa oleva. 12 voltin jännite täytti asetetut ehdot (myös studiokäytössä tarvittavien muuntajien osalta).

Koska liikkeiden vaadittu tarkkuus oli korkea, moottorityypiksi valittiin askelmoottori. Kuvassa 3 näkyvästä taulukosta on luettavissa valitun moottorityypin sähköiset arvot; 12 voltin käyttöjännite sekä 300 mA:n käämikohtainen virta. Ero listassa esimerkiksi seuraavalla rivillä olevaan 6 voltin 42BYG401-tyyppin moottoriin on akselilta mitattavan voiman suhteen selkeä. Näin on siitakin huolimatta, että kyseisen moottorin 800 mA:n virrankulutus on yli kaksinkertainen 42BYG304-02-tyyppiin verrattuna. Myös tämä seikka puoltaa valitun 12 voltin käyttöjännitteen paremmuutta suhteessa 6 volttiin tässä nimenomaisessa sovelluksessa.

Model	No. of Phase	Voltage (V)	Current (A)	Resistance (Ω)	Inductance (mH)	Holding torque(g.cm)	Rotor Inertia(g.cm ²)	Weight(Kg)	Lead style	Wire diagram	Figure diagram	Figure L	Figure L1	Figure L2
42BYG015	4	12	0.3	39	16	1150	20	0.2	AWG26 UL1007	a	1	34	16	500
42BYG016	4	12	0.16	75	45	900	20	0.2	AWG26 UL1007	c	1	34	17	400
42BYG021(JB)	2	9.1	0.24	38	44	800	20	0.2	AWG26 UL1007	b	1	34	24	500
42BYG023-05	2	12	0.4	30	36	2000	20	0.2	AWG26 UL1007	b	1	34	18	400
42BYG023-14A	2	9.9	0.45	22	28	1950	20	0.2	NO LEAD	b	2	34	19	
42BYG032A(JB)	2	15.3	0.33	46.5	58	2000	20	0.2	NO LEAD	b	2	34	20	
42BYG034A	2	14	0.4	35	44	2200	20	0.2	NO LEAD	b	2	34	24	
42BYG228	4	12	0.4	30	25	2000	22	0.24	AWG26 UL1007	a	1	38	24	480
42BYG304-02	2	12	0.3	40	80	2800	30	0.26	AWG26 UL1569	b	1	44	20	1200
42BYG401	4	6	0.8	7.5	5	2300	36	0.29	AWG26 UL1007	a	1	48	13.5	110

*Above in the table arranges in order only is the representative product.

2.FIGURE DIMENSIONS (UNIT=MM)

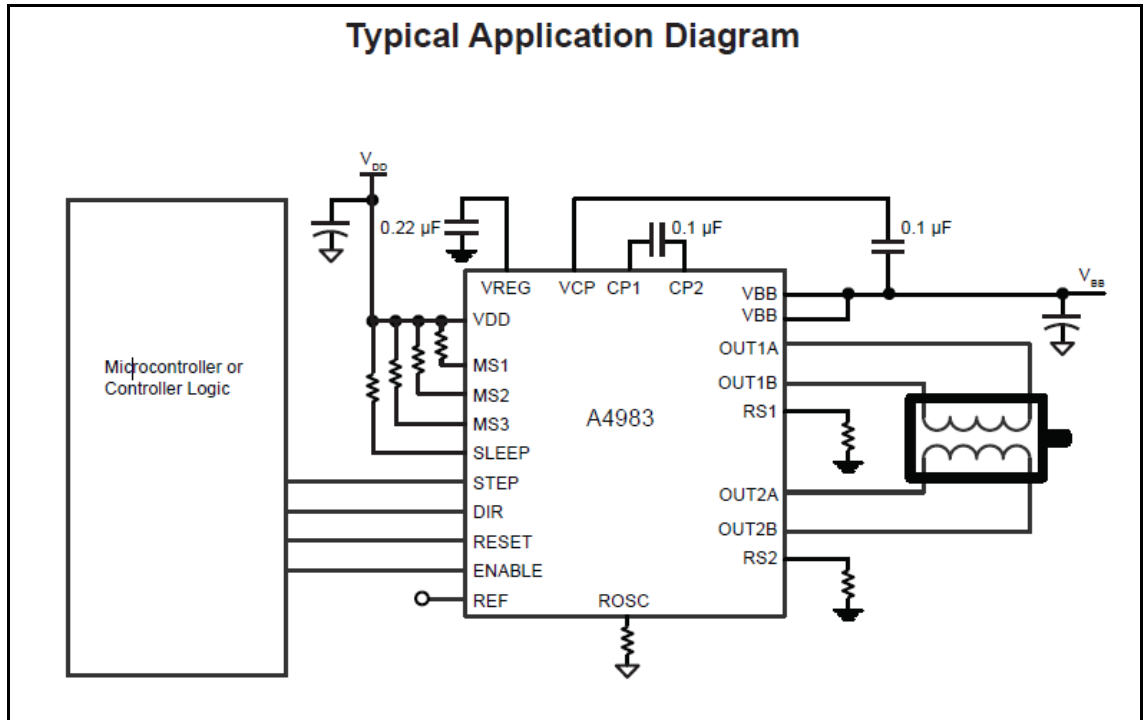
3.WIRING DIAGRAM

Kuva 3. Valitun NEMA 17, 42BYG304-02 -askelmoottorin tekniset ominaisuudet [4]

Mikrokontrollerin ja muiden sähköisten komponenttien, pois lukien askelmoottorit, käyttöjännite oli reguloitava viiteen volttiin. Regulaattoriksi valittiin suunnitellulla jännitealueella pienihäiriöinen L7805 [5,5]. Selvästi regulaattorin ulostulojännitettä korkeampi akkujännite takaisi riittävän jännitteen mikrokontrollerille moottorin aiheuttamasta kuormasta huolimatta. Siksi ei myöskään nähty tarpeelliseksi mikrokontrollereiden ja oheiskomponenttien syöttämistä erillisellä akulla mahdollisten häiriöiden poistamiseksi.

Kääntöpään asentotiedon laskemiseen edullisin ja hyvin yleinen vaihtoehto käyttövoiman ollessa normaali sähkömoottori on kameran asentoa mittaava potentiometri. Käytettäessä tarkkoja askelmoottoreita asentotieto luetaan yleensä ääriasentoihin sijoitetuilla rajakytkimillä tai optohaarukoilla.

Tästä syystä projektin asentotietojen mittaustavaksi valittiin jälkimmäinen optohaarukoilla, joiden käyttöikää pidettiin mekaanisia rajakytkimiä korkeampana.



Kuva 4. A4983-askelmoottoriohjain suojaelektronikalla ja sen kytkentä [6,1]

Askelmoottorin ja mikrokontrollerin välinen kommunikointi päätettiin aluksi hoitaa yksinkertaisella ja yleisesti käytössä olevalla ULN2003-askelmoottoriohjaimella [7]. Suojaelektronikan puute pakotti kuitenkin harkitsemaan vaihtoehtoja uudestaan ja ULN2003:n sijasta päätettiinkin käyttää nykyaikaisempaa A4983-piiriä. Jonkin verran ULN2003-ohjainta kalliimpaan piiriin sisältyy virta- ja lämpösuojaukset sekä mahdollisuus tarvittaessa säätää askelmoottorin steppien pituutta.

Piirilevyn arvioidut valmistuskustannukset komponentteineen asettuvat viidenkymmenen ja sadan euron välille. Vaikka komponenteille on laskettavissa tarkat hinnat, todelliset kustannukset riippuvat siitä, kuinka paljon kolmansien osapuolten piirilevyjä ja ns. breakout boardeja joudutaan hyödyntämään laitteen toteutuksessa. Valittujen askelmoottorien kappalehinnat olivat tarkastushetkellä toimittajasta riippuen noin 15 euroa + toimituskulut.

Liitteestä 2 löytyvät tarkemmat sähkösuunnitelmat.

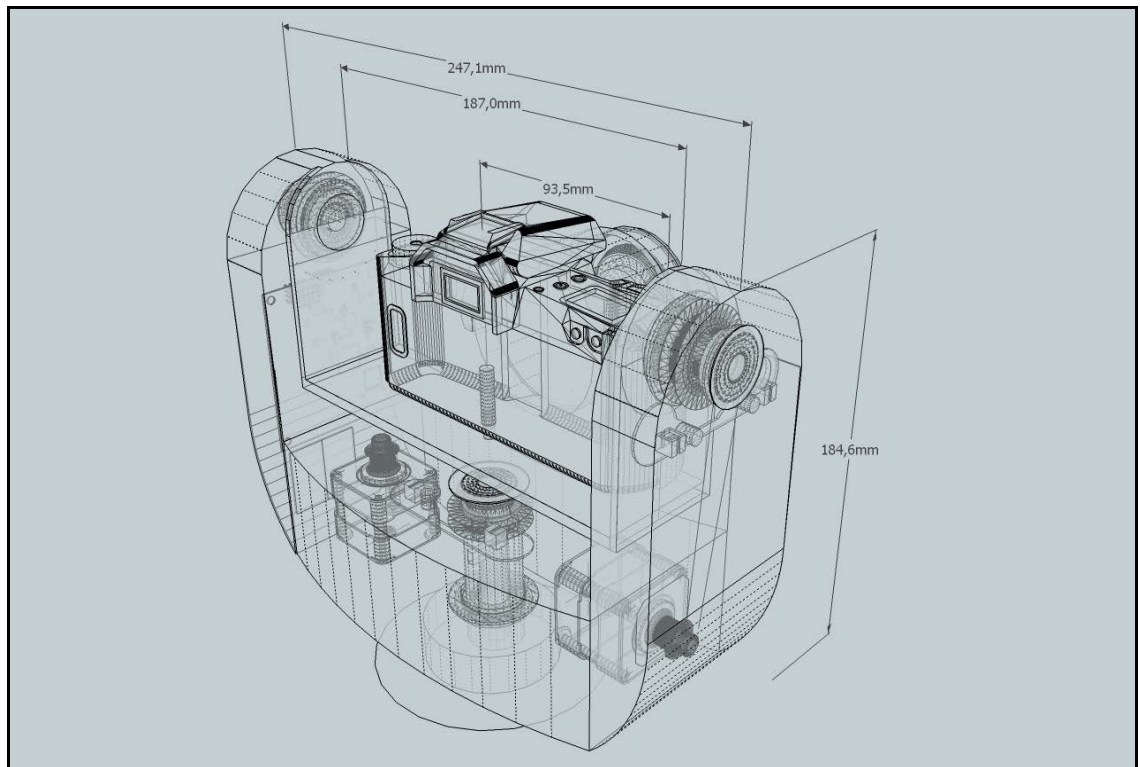
5.2 Mekaaniset ratkaisut

Kamerapuomi, johon kääntöpää oli tarkoitus liittää, oli kantokyvyltään 3,6 kg. Kamera objektiivineen painaa noin 2 kg, joten itse kääntöpään maksimipainoksi määräytyi pyöreästi 1,6 kg.



Kuva 5. ProAm DVC250 -kamerapuomi asetti kääntöpäälle tiukat painorajat [8]

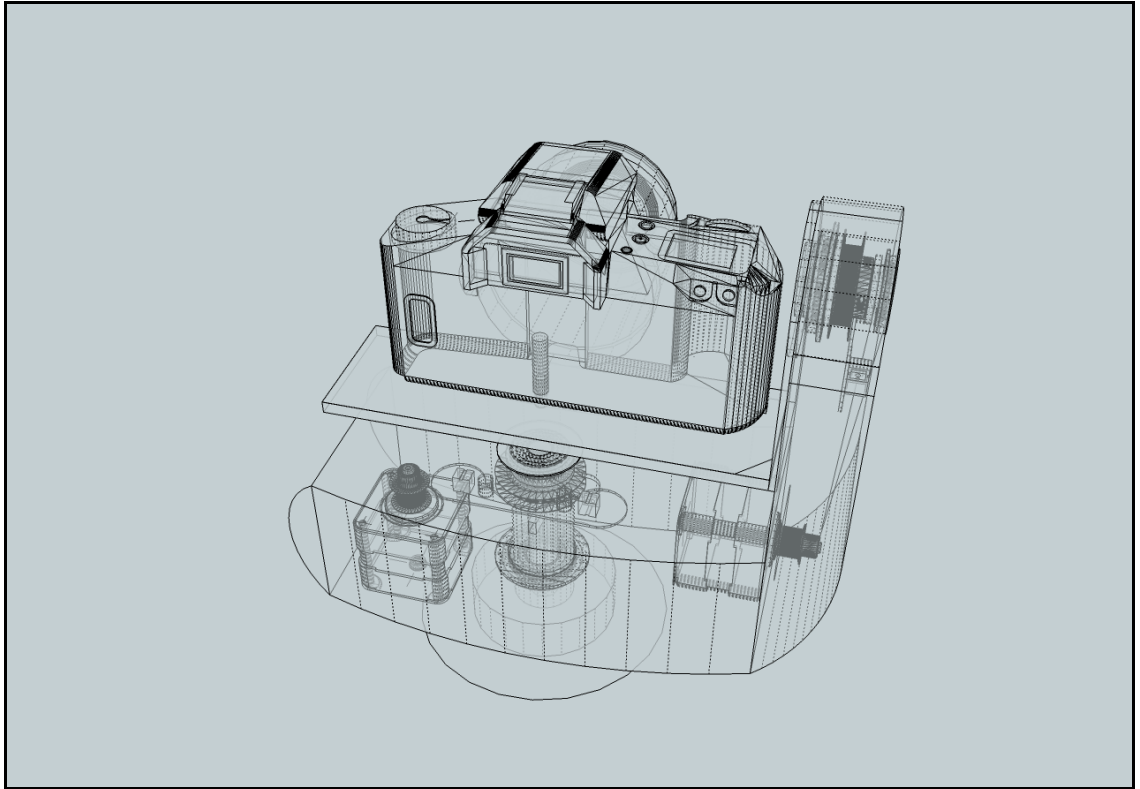
Kääntöpään valmistusmateriaaliksi suunniteltiin päärungon osalta 3 - 5 mm haponkestävää terästä. Lopullisen tarkan paksuuden määrittelee ulkopinnoissa käytetty materiaali. Mikäli ulkopintoihin valitun materiaalin jäykkyys ei riitä tukemaan varsinaista päärunkoa mekaanisessa rasituksessa (ulkopintojen materiaali on ohutta alumiinia tai polykarbonaattia), päärungon levypaksuutta on luultavasti kasvatettava kolmesta millistä ylöspäin. Varsinaisia lukuja laskelmia materiaalivalintojen osalta ei ole tehty.



Kuva 6. Kääntöpää

Kuvasta 6 näkyy 3D-luonnos kääntöpäästä suhteessa kameraan. Kuvassa olevat mittasuhteet ovat tarkkoja. Moottorit vastaavat todellisia NEMA 17 -askelmoottoreita ja kamera vastaa mitoiltaan järjestelmäkameroiden ammattimalleja.

Aluksi harkinnassa oli toispuoleinen kamerataso ja laakerointi, joka olisi keventänyt kääntöpäätä mahdollistamalla vasemmasta sivusta avonaisen kehikon ja poistamalla näin toisen laakeripakan akseleineen. Perinteisesti vastaavia ratkaisuja on käytetty kamerajärjestelmissä, joissa painopiste on lähellä akselia ja laakeripakkaa ja suurta vääntömomenttia ei pääse syntymään. Järjestelmäkameran paino (isolla objektiivilla 2 kg ja yli) ja painopisteen etäisyys laakeroinnista olisi vaatinut kameratason vahvistamista, ja liikkeiden tarkkuuden epäiltiin siitä huolimatta heikentyvän. Yhdeltä puolelta laakeroitu malli hylättiin, koska saavutetut pienet voitot keveydessä nähtiin liian suurina menetyksinä muissa ominaisuuksissa. Kuvassa 7 on esitetty hylätty suunnitelma.



Kuva 7. Kääntöpää kevennettyinä

Valitun 42BYG304 -askelmoottorin voima siirretään 1/10-välityksellä hihnalla kameran kiinnitystasoon. Tämä vastaa suoraan muunnettuna 28 kg:n nostovoimaa yhden senttimetrin päässä akselin halkaisijan keskipisteestä tai 2,8 kg:n nostovoimaa kymmenen senttimetrin päässä. Valittu moottorityyppi on kompromissi pitovoiman ja painon suhteen. Tällöin moottorien yhteispaino nousee hieman yli puoleen kiloon mutta on vielä hyväksyttävissä rajoissa. Rungolle ja muille komponenteille jää tällöin 1,1 kilogramman maksimipainoraja.

Yleensä käytetyn rattailta välitetyn suoravedon sijasta valittiin hammastettu hihnavedo, joka minimoi moottorista kamerakiinnikkeeseen siirtyviä värinöitä, on erittäin tarkka ja hiljainen sekä säästää mekaanisia komponentteja. Ratkaisun pitäisi jakaa nopeisiin liikkeenmuutoksiin liittyviä impulsseja laajemmalle aikavälille, mikä johtuu hihnaan aiheutuvasta pienestä venymästä. Vastaavia ratkaisuja on käytössä esimerkiksi mustesuihkutulostimissa ja tasokannereissa.

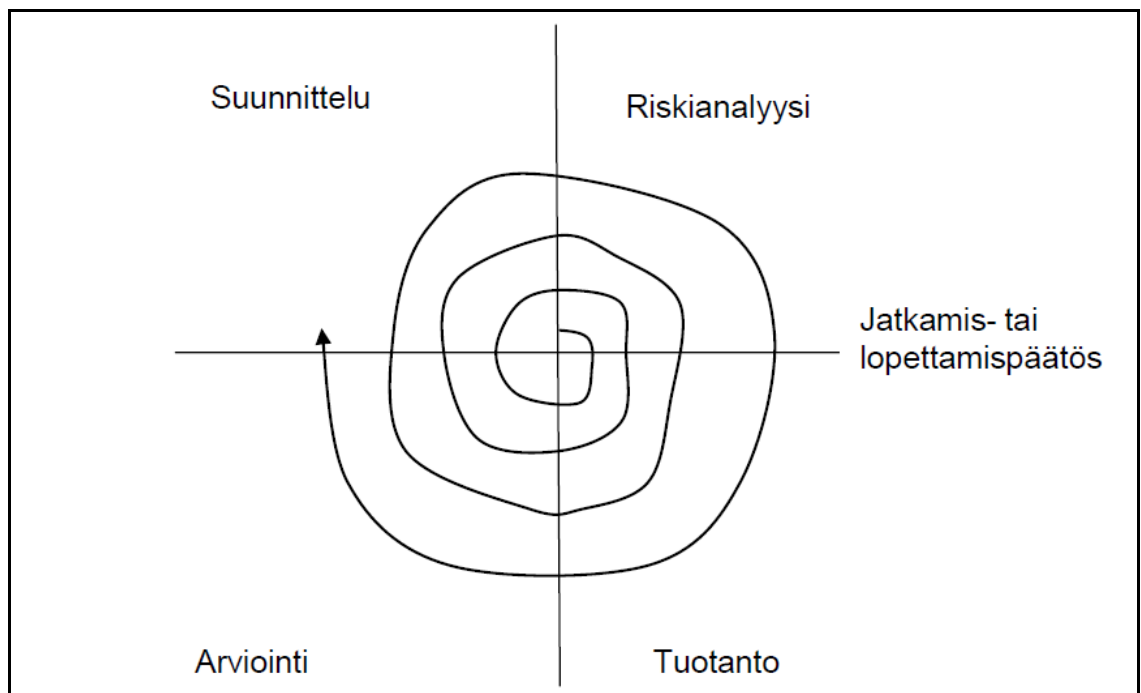
Hammashihnavedolla saadaan rakenteesta mahdollisimman kevyt. Koska erillisiä ratasvälityksiä tai vaihteistoja ei käytetä, käytettävien komponenttien määrä ja paino saadaan pidettyä alhaisena. Liitteestä 3 löytyvät tarkemmat kuvat suunnitellusta rakenteesta.

5.3 Ohjelmistotekniset ratkaisut

Kun aletaan analysoida, millaista ohjelmistosuunnittelumallia projektissa käytettiin, seikat kuten kooditason toteutuksen aikataulusuunnitelma ja testausvaihe viittaisivat lähinnä ketteriä menetelmiä puoltaviin malleihin. Kehitysvaiheessa jokainen päivä pyrittiin päättämään versioon, jossa lisätyt toiminnot ja osat kääntyivät ongelmitta eivätkä vaarantaneet muun ohjelman eheyttä. Seuraava päivä alkoi yleensä edellisen käännöksen syöttämällä sulautettuun kehitysympäristöön, mitä seurasi tulosten havainnointi. Näiden havaintojen pohjalta toteutettiin mahdolliset korjaukset, minkä jälkeen toteutettiin taas mahdolliset uusien toimintojen lisäykset, ja sykli alkoi alusta.

Projektia varten perustettu Subversion-palvelin ja siihen integroitunut lähdekoodiin kohdistuneita muutoksia visuaalisesti esittänyt Trac-projektinhallintaohjelmisto olivat kriittisen tärkeitä työvälineitä.

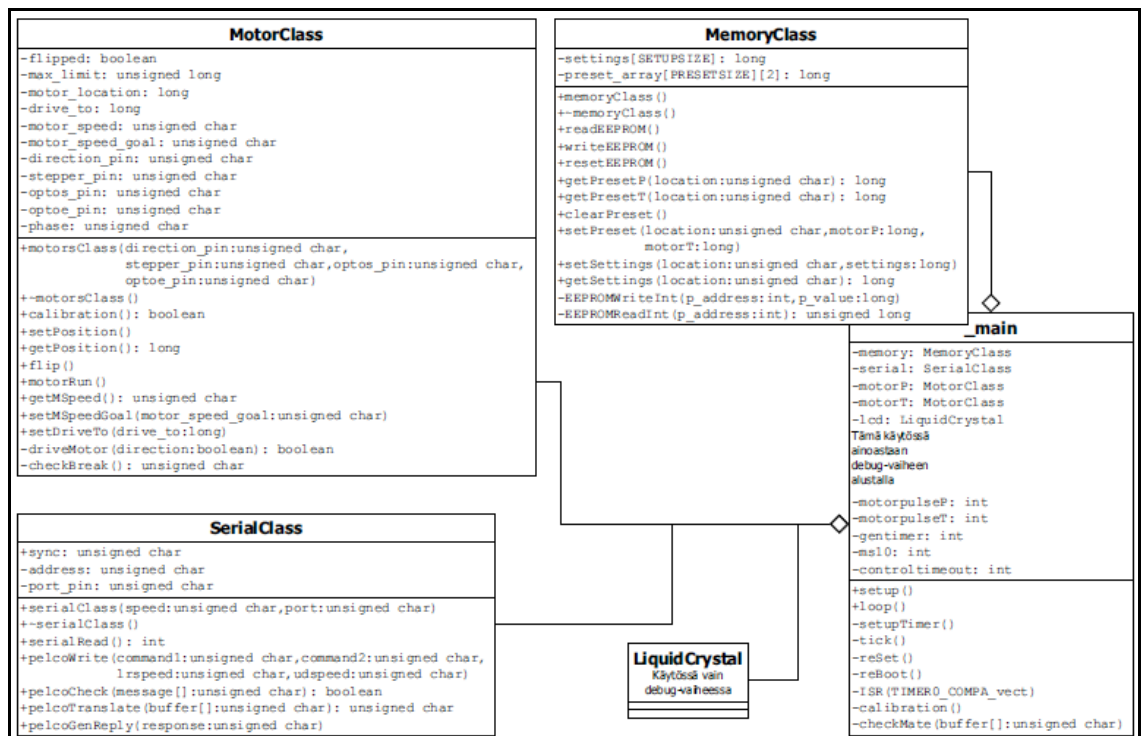
Mikäli projektia lähdetään viemään toteutusvaiheeseen, voitaisiin käytettyä mallia tarkentaa spiraalimalliin ja sen ensimmäiseen iteraatiokerrokseen. Tarkasteltaessa kyseistä mallia sijoituttaisiin tilanteeseen, joka on juuri ennen varsinaisen prototyypin valmistamista eli tuotantoa, riskianalyysin jälkeen. Kuvassa 8 esiintyy käsitelty spiraalimalli.



Kuva 8. Spiraalimalli [9,12]

Vaikka Arduinon IDE:llä tuotettava koodi ei täysin toteuta olio-ohjelmoinnista tuttua dynaamisuutta, vaan esimerkiksi new- ja delete-metodit puuttuvat ja ns. luokat toteuttavat oliot määritellään staattisesti aina ennen varsinaista pääsilmuukkaa (loop()), voidaan selkeyden vuoksi puhua luokkarakenteesta. Ohessa on esimerkki olioiden liittämisestä luokkiin Arduino-ympäristössä.

```
MemoryClass memory;
SerialClass serial (0xff, 0x01, memory.getSetting(1),
  RX_OR_TX);
MotorClass motorP
  (M1_DIR, M1_STEP, OPTOM1_START, OPTOM1_END);
MotorClass motorT
  (M2_DIR, M2_STEP, OPTOM2_START, OPTOM2_END);
```



Kuva 9. Ohjelmiston luokkakaavio (liitteessä 4 täysikokoisena)

Kuten kuvasta 9 voidaan päätellä, jako luokkiin toteutettiin fyysisen rakenteen perusteella moottoreihin, sarjaliikennepuoleen ja muistin käsittelyyn. Myös suurimman moottoriluokan keskeinen asema ilmenee kuvasta.

Pääluokassa "**_main**" on yhteensä 9 funktiota. Siitä käsin luodaan muiden luokkien oliot, toteutetaan asetusten ja kellokeskeytysten lisäksi ajonaikaisen pääsilman kierto sekä tulkitaan muista luokista palautettua tietoa esimerkiksi saaduista Pelco D -protokollan ohjauskäskyistä.

MemoryClass-luokassa on yhteensä 13 funktiota. Luokka huolehtii ajonaikaisessa ja pysyvässä muistissa olevien asetustietojen hausta ja tallennuksesta.

MotorClass-luokassa on yhteensä 12 funktiota. Luokka toteuttaa moottorien ohjaukseen liittyvää asentotiedon tallennusta, moottorikohtaisten optoerottimien tilatietojen tarkistuksia sekä moottorien varsinaista pyörittämistä eritasoisilla funktioilla.

SerialClass-luokassa on yhteensä 7 funktiota. Luokka lukee sarjaliikenneväylää ja toteuttaa Pelco D -protokollan mukaista sarjaliikenneviestien varmentamista.

Luokkien funktioiden koot vaihtelevat pienimpien asetusfunktioden (settereiden) yhden rivin ratkaisusta pisimpien funktioiden viidenkymmenen rivin toteutuksiin. Valmis mikrokontrollerille ladattava käännetty binäärikoodi vie maltilliset 5900 tavua käytössä olevasta reilusta 16000 tavusta. Tämä mahdollistaisi pienemmällä 8 kilotavun muistilla saatavan Atmelin 168-sarjan mikrokontrollerin käytön, mutta ei jättäisi juurikaan laajennusvaraa mahdollisille lisäominaisuuksille.

Ohessa seuraa kolme esimerkkiä selostuksineen ohjelmassa esiintyvistä funktioista ja niiden toteutuksista Arduino-kehitysympäristössä. Ensimmäiset kaksi ovat pieniä apufunktioita, kolmas moottoreita ohjaavan luokan pääfunktio. Vaikka bittitason operaatioita pyritään usein Arduinon kehitysympäristössä piilottamaan näkyvistä valmiisiin mukana seuraaviin funktioihin ja kirjastoihin, on niiden tunteminen ja opiskelu välttämätöntä. Tästä esimerkkinä on pieni mutta tärkeä ohjelman muistinkäsittelyyn liittyvä funktio, `EEPROMReadint`.

5.3.1 Esimerkki 1, SerialClass-luokan funktio pelcoCheck

```
bool SerialClass::pelcoCheck(unsigned char* message){
    return ((message[1] + message[2] + message[3]
    + message[4] + message[5]) % 256) == message[6];}
```

Oheisessa funktiossa tarkastetaan, onko luetussa viestissä Pelco D -protokollan mukainen tarkistussumma kunnossa. Protokollan tarkistussumman muodostus on yksinkertaista eikä sisällä virheenkorjausta. Tavut 1, 2, 3, 4 ja 5 summataan ja niistä muodostetaan luvun 256 jakojäännös, joka tallennetaan tavuun 6. Viestiä luettaessa sama lasku toistetaan ja saatua tulosta verrataan 6. tavuun. Jos arvot täsmäävät, viesti todetaan virheettömäksi.

5.3.2 Esimerkki 2, MemoryClass-luokan funktio EEPROMRead.

```
unsigned long MemoryClass::EEPROMReadInt(int addr){
    return (((EEPROM.read(addr) << 0) & 0xFF)
    + ((EEPROM.read(addr + 1) << 8) & 0xFF00)
    + ( ((EEPROM.read(addr + 2) << 0) & 0xFF)
    + ((EEPROM.read(addr + 3) << 8) & 0xFF00))*65536);}
```

Esimerkki yleisesti käytössä olevasta yksinkertaisesta, mutta hyödyllisestä menetelmästä lukea long-tyyppinen muuttuja pysyvästä muistista. Yli yhden tavun pituista muuttujaa ei oletusarvoisesti osata hakea useammasta pysyvän muistin lohkoista, vaan se pitää tallennettaessa pilkkoa ja haettaessa jälleen koota. Long vaatii neljä tavun mittaista muistipaikkaa.

Yhden rivin jako osiin:

```
EEPROM.read(addr + 1)
```

1. Haetaan arvo muistipaikasta (addr + 1).

```
<< 8
```

2. Siirretään arvo kahdeksan bittiä vasemmalle.

```
& 0xFF00
```

3. Tyhjennetään maskilla alemmat 8 bittiä.

5.3.3 Esimerkki 3, MotorClass-luokan funktio motorRun

```
void MotorClass::motorRun(void){
    bool mot_result = true;
    if (this->motor_speed_goal>this->motor_speed)
        this->motor_speed++;
    if (this->motor_speed_goal<this->motor_speed)
        this->motor_speed--;
```

Moottorin ajonopeuteen kohdistuvat muutokset toteutetaan aina mahdollisimman pehmeillä siirtymillä. Nopeutta kuvaa kaksi arvoa, moottorin nykyinen nopeus eli motor_speed ja moottorin haluttu nopeus eli motor_speed_goal. Nopeuteen tehdään jokaisella kierroksella yhden yksikön lisäyksiä tai poistoja, kunnes ollaan halutussa moottorinopeudessa. Tarkoituksena on välttää moottoreilla ohjattaviin kappaleisiin (kamera ja objektiivi) kohdistuvia suuria liikkeen muutoksia.

```
//----- joystick drive
if (this->drive_to==-9999 && motor_speed_goal!=0){
    if (this->motor_speed_goal<0)mot_result = driveMotor(0);
    if (this->motor_speed_goal>0)mot_result = driveMotor(1);
}
//----- joystick drive ends
```

Mikäli käyttäjä kääntää kameraohjainta, moottoria ajetaan haluttuun suuntaan. Ehdossa esiintyvä drive_to-muuttujan arvo "-9999" vastaa nollatilaa. Tällöin ns. "preset"-toiminto eli muistiin esiasetettuihin asentoihin ajo ei ole käytössä ja käskyt luetaan suoraan käyttäjältä.

```
//----- preset drive
else{
    if (this->drive_to!=-9999 && this->drive_to != this->motor_location){
        //drive parameters should be in sustainable range
        if (this->drive_to>this->max_limit)
            this->drive_to=this->max_limit;
        if (this->drive_to<0)this->drive_to = 0;

        long tmp = abs(this->drive_to - this->motor_location);
        bool result = false;
        if (tmp > 0x40)tmp = 0x40;
        if (tmp < 0x01)tmp = 0x01;
        this->motor_speed_goal = tmp;
        if (this->motor_location < this->drive_to)
            mot_result = driveMotor(0);
        if (this->motor_location > this->drive_to)
            mot_result = driveMotor(1);
    }
}
```

"Preset"-toiminto on tila, jossa kääntöpäälle on aikaisemmin asetettu tiettyyn muistipaikkaan moottoreiden senhetkiset asentotiedot, ja nyt moottorit pyritään ajamaan uudestaan kyseisiin asentoihin. Haettavan asentotiedon tarkastetaan löytyvän saavutettavissa olevalta alueelta eli 0 - max_limit.

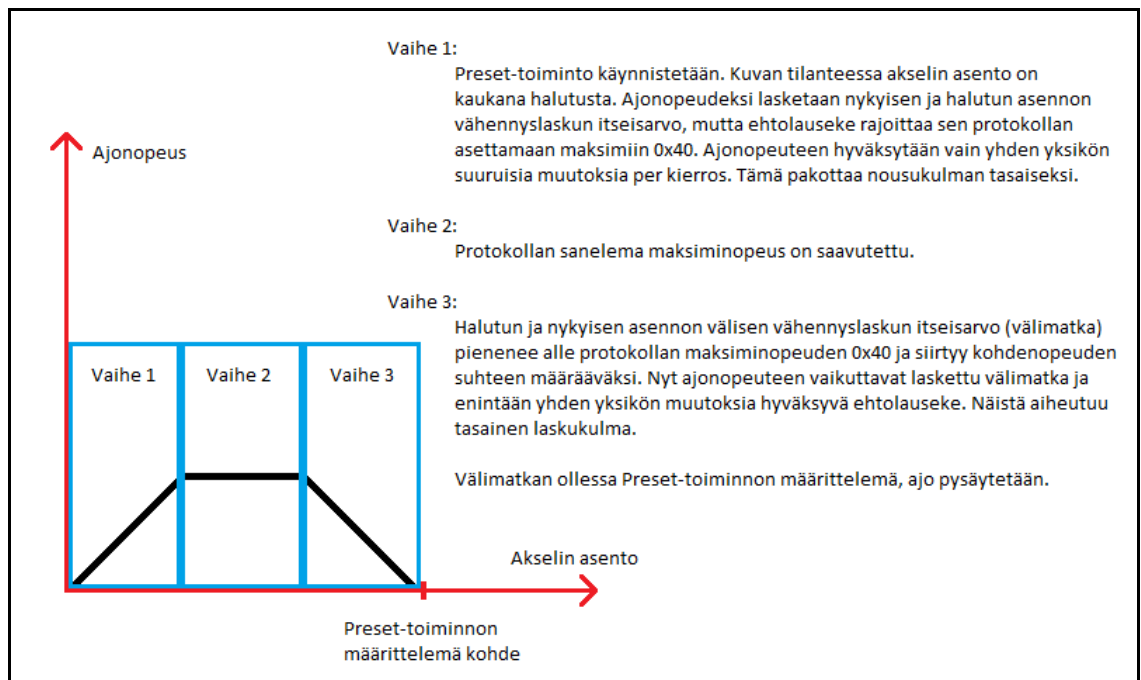
Tämän jälkeen määritellään ajonopeus yksinkertaisella vähennyslaskun itseisarvolla, jonka tulos on halutun asennon ja nykyisen asennon välimatka. Tämä kahden pisteen välinen matka määrittelee ajonopeuden. Mitä pidempi etäisyys, sitä suurempi ajonopeus. Nopeutta rajoitetaan kuitenkin niin, että maksiminopeus on protokollan mukainen 0x40 ja miniminopeus 0x01. Tästä johtuen pienet siirtymät hoidetaan aina pienillä nopeuksilla, ja lähestyttäessä asetettua kohdetta välimatka pienenee ja samalla haluttua ajonopeutta hidastetaan. Muutokset tehdään muuttujaan motor_speed_goal, joten funktion alussa käsiteltävä liikenopeuden muutoksia hidastava lauseke pätee myös tähän. Lopussa todetaan tarvittavan liikkeen suunta ja ohjataan matalan tason driveMotor-funktio ajamaan moottoria.

```
//location equals preset
if (this->drive_to == this->motor_location){
    this->drive_to = -9999;
    this->motor_speed_goal = 0x00;
    this->motor_speed = 0x00; // quick stop, go past soft drive
}
//----- preset drive ends
}
```

Ollaan saavutettu haluttu muistista haettu asento. Toiminto lakkautetaan ja moottorit pysäytetään. Jälkimmäinen motor_speed-muuttujan nollaus on varotoimi. Tässä vaiheessa asetetun nopeuden pitäisi olla automaattisesti 0x00. Kuvassa 10 esitetään ajonopeutta määrittelevät vaiheet etäisyyden ja ajonopeuden suhteen.

```
//calibrate on the fly
if (!mot_result){
    byte tmp = checkBreak();
    if (tmp == 1){//start break detected
        this->motor_location = 0;
    }
    if (tmp == 2){//end break detected
        this->max_limit = this->motor_location;
    }
}
}
```

Lopuksi funktio toteuttaa kalibroinnin, mikäli asennon havaitaan saavuttaneen toisen maksimirajoista. Tämä täydentää jokaisessa käynnistyksessä suoritettavaa askelten laskemista ääriasentojen välillä. Vaikka tämän siirtymien välillä vaadittujen askelten määrän pitäisi olla täysin muuttumaton, seikat, kuten muutokset käännettävän kameran painossa objektiivin vaihtojen yhteydessä tai puomiin kohdistuvat nopeat heilahdukset, saattavat aiheuttaa muutaman mitatun askeleen eroja tuloksissa. Mikäli ollaan alkurajassa, moottorin asennon on aina oltava 0; mikäli ollaan loppurajassa, moottorin senhetkinen asento asetetaan suurimmaksi mahdolliseksi eli max_limit-arvoksi.



Kuva 10. "Preset"-toiminnon ajonopeuden muutosten eri vaiheet

Käsitelty motorRun on yksi MotorClass-luokan tärkeimmistä funktioista, käytännössä luokan pääfunktion asemassa oleva moottorihjauksia toteuttava funktio. Se on selkeyden vuoksi pilkottu lyhyempiin osiin. Liitteessä 5 sekvenssikaavio "preset"-toiminnosta, johon kyseinenkin funktio liittyy.

6 ARVIOINTI JA JOHTOPÄÄTÖKSET

Projekti onnistui suhteellisen hyvin. Alkuvaiheessa työhön oli tarkoituksena liittää myös prototyypin toteutus käytännössä, mutta projektiin liittyneet pienet vastoinkäymiset johtivat tämän siirtämiseen mahdolliseen lähitulevaisuuteen.

6.1 Ongelmakohdat

Kamerapuomin asettama 1,6 kg painoraja osoittautui hyvin haasteelliseksi, ja kyseinen rajoite heijastui käytännössä kaikkiin suunnittelun osa-alueisiin. Perinteisiä edellisistä projekteista totuttuja varmuusrajoja ei materiaalien riittävän jäykkyyden varmistamiseksi voitu toteuttaa. Prototyypivaiheeseen tehtävien materiaalivalintojen toimivuus onkin syytä varmistaa erillisillä luku- ja lujuuslaskelmilla ennen toteutusta.

Rungon lisäksi myös moottoreiden valinnassa keveys oli tärkeysjärjestyksessä korkeimmalla. Projektin alussa suunnitellut moottorit olivat lopulta korvattava pienemmillä NEMA 17 -standardin toteuttavilla askelmoottoreilla ja hävitty vääntömomentti jouduttiin korvaamaan 1/10-välitykselle mitoitetuilla hammashihnapyörillä.

Mahdollisuutta kameran tarkennuksen etäohjaukseen tutkittiin. Paljastui, että protokolla ei tue muuttuvanopeuksista eli analogista tarkennusmoottorin ajoa, vaikka mitään teknisiä rajoituksia sille ei ole. Edellisestä johtuen tarkennuksen ohjaus olisi tunnoton ja epätarkka, eikä sitä voisi hyödyntää suunnitellussa käyttötarkoituksessa kameran objektiivin tarkentamiseksi. Periaatteessa rajoitusta voisi kiertää laajentamalla Pelco D -protokollaan omaa käskykantaan, mutta se vaatisi ohjauslaitteiden uudelleen ohjelmointia. Tämä taas voi pahimmassa tapauksessa muodostua työmäärältään suuremmaksi kuin koko ohjauslaitteen suunnittelu itse alusta alkaen olisi. Ominaisuuden käyttöön saamista ei siksi pidetty realistisena tavoitteena.

Mahdollisen ulkopuolisen rahoituksen tärkeyttä ei myöskään pidä väheksyä. Nyt projekti toteutettiin täysin omakustanteisena. Rahoituksen puute oli myös pääasiallinen syy päätökseen jättää prototyyppi tältä erää toteuttamatta.

Yksikköhinnaksi arvioitu 500 euroa on mahdollisuuksien rajoissa puhuttaessa valmiista tuotteesta, mutta prototyypin kustannuksiksi on asiaa uudelleen tarkasteltaessa arvioitu karkeasti noin 1500 - 2000 euroa, josta rungon suunnittelu materiaaleineen veisi pääosan.

6.2 Mitä opittiin?

Kokemuksesta johtuen helpoin osa-alue oli ehdottomasti varsinainen ohjelmointityö. Kiitos Pelcolta saatujen tarkkojen D-protokollaa koskevien teknisten dokumentaatioiden, koodiin oli helppo lisätä vaaditut ominaisuudet ja sarjaliikenteessä ilmaantuneet epäselvyydet selvisivät nopeasti. Tämä todisti käytännössä tarkkojen teknisten spesifikaatioiden tärkeyden työskenneltäessä esimerkiksi rajapintojen parissa. Myös käytetty mikrokontrolleri osoittautui valittuun kohteeseen riittävän tehokkaaksi, eikä koodin luettavuutta tarvinnut vaarantaa.

Kokonaisuutena valittu projekti käsitteli ohjelmistotekniikan opintolinjalla läpi käytyjä asioita elegantisti käytännössä alkaen matemaattisista perusopinnoista ja siirtyen niistä elektroniikan, digitaalitekniikan, mikroprosessorien sekä signaalinkäsittelyn kautta sulautettujen järjestelmien ohjelmointiin.

6.3 Mitä muuttaisin?

Projektin olisi voinut jakaa useamman eri alan opiskelijan projektiksi tai yhdistetyksi työksi. Ohjelmoinnin lisäksi projekti vaati 3D-malleja ja sähkösuunnitelmia. Rakennustekniikan opiskelijoilla olisi riittävä koulutus rakenteen lujuusteknisten ominaisuuksien selvittämiseksi, ja muotoilulla olisi kokemusta itse kappaleiden suunnittelusta ja pääsy nopeita prototyyppejä tuottaviin 3D-tulostusyksiköihin.

Projektin toteuttaminen lyhyemmässä ajassa olisi vaatinut ulkopuolista rahoittajaa.

6.4 Johtopäätökset

Sähköisen suunnitelman toteutunut yksinkertaisuus, ohjelmistosuunnittelun vaivattomuus ja saatavien ilmaisten 3D-suunnitteluohjelmistojen hyvä käytettävyys todistivat, että kaupalliset tuotteet on mahdollista haastaa. Kiitos harrastajille tarjolla olevien alihankkijoiden, haastaja ei myöskään välttämättä jää enää ominaisuuksien tai laadun osalta altavastaajaksi, vaan valmistunut tuote voi olla edullisemmän hinnan lisäksi myös teknisesti parempi.

Vaikka kääntöpäätä ei saatu vielä toteutettua käytännössä, sen ominaisuudet vaikuttavat lupaavilta. Valmiin tuotteen yksikköhinta voi hyvinkin asettua suunnitellulle 500 euron tasolle ja itse laitteen teoreettinen ohjaustarkkuus on korkea. Lisäksi sähköisten kytkentöjen yksinkertaisuus ja yleisesti saatavilla olevat komponentit tekisivät valmiista laitteesta helposti huollettavan.

Tarkoituksena on palata kääntöpään pariin taloudellisesti paremmalla ajankohdalla tai mikäli ulkopuolinen taho kiinnostuu toteuttamaan prototyypin käytännössä. Yksi paikallinen yritys on ilmaissut mielenkiintonsa laitteen tekniikan käyttämiseen varastokentän pinnanmuutoksia mittaavan laser-etäisyysmittarin ohjauksessa, mutta toteutusaikataulu tai tekniset yksityiskohdat eivät ole vielä tarkentuneet.

7 PELCO PROTOKOLLAN DOKUMENTAATIO

Tässä työssä käytettiin apuna Pelco D -protokollan kirjallista dokumentaatiota. Kuitenkaan dokumentaatioon liittyviä taulukoita tai kuvia ei voitu käyttää aihetta käsittelevissä luvuissa. Tämä johtuu ehdoista, joiden hyväksyminen oli edellytys dokumentaation käyttöön saamiseksi. Lähdeluettelossa ei myöskään ole suoraa viittausta osoitteeseen, josta materiaali olisi mahdollista ladata. Käyttölupa on anottava tapauskohtaisesti ja jakelusta pidetään rekisteriä.

Ilmoitus ensimmäisellä sivulla on hyvin yksiselitteinen: "Those receiving this protocol cannot redistribute the protocol without the expressed written consent of Pelco. [10]" Vapaasti suomennettuna edellisessä viittauksessa lukee: "He, jotka vastaanottavat tämän protokollan, eivät saa uudelleen levittää sitä ilman Pelcon kirjallista suostumusta." Kyseistä vaadittua kirjallista lupaa haettiin, mutta sitä ei valitettavasti myönnetty.

Pelco D-protokollan rinnalla on olemassa vanhempi, Pelco P -protokolla. Tällä protokollalla ei sinänsä ole merkitystä aiheen toteutuksen kannalta; se pitää sisällään samoja toimintoja pienin poikkeuksin ja löytyy usein laitteesta Pelco D -protokollan ohessa. Työn osalta kyseistä protokollaa ja sen käskykantaan olisi vertailtu nykyiseen Pelco D-protokollaan lähinnä historiallisista näkökohdista. Kysyttäessä Pelco P -protokollan teknisiä dokumentteja vastaus oli kieltävä. Kyseisen protokollan jakelua rajoitetaan, eikä opinnäytetyö täyttänyt sen jakelusta määrättyjä kriteereitä myöntävän päätöksen osalta.

8 YHTEENVETO

Tarkoituksena oli toteuttaa edullinen kamerakääntöpää Haminan Teatterin mahdollisten lyhytelokuvien ammattimaiseen kuvaamiseen ja koeluonteisiin projekteihin.

Järjestelmä suunniteltiin teoriatasolla mahdollisimman valmiiksi. Toteutuksen seuraava vaihe olisi ensimmäisen kuviin perustuvan prototyypin valmistaminen.

LÄHTEET

- 1) Arduino. Saatavissa: <http://arduino.cc>. [viitattu 29.9.2011]
- 2) Hovi, Vesa. Keskustelu projektin kulusta ja lähtöasetelmista 6.10.2011. Hamina: Rai-Kuva Ky:n toimitilat.
- 3) Karvinen, T. & Karvinen, K. 2009. Sulautetut - Opi rakentamaan robotteja ja muita sulautettuja järjestelmiä. Helsinki: Readme.fi
- 4) Changzhou Jingbo Motors Co. 2005. "Step Motors". Saatavissa: <http://www.czcoact.com/english/pro-h5.asp> [viitattu 29.09.2011]
- 5) STMicroelectronics, 2011. L78xx Positive voltage regulator ICs Rev 24. Saatavissa: http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/CD00000444.pdf. [viitattu 03.10.2011]
- 6) Allegro MicroSystems, Inc. 2008 Rev. 1. A4983 DMOS Microstepping Driver with Translator. Saatavissa: http://www.allegromicro.com/en/Products/Part_Numbers/4983/4983.pdf. [viitattu 29.09.2011]
- 7) STMicroelectronics, 2007. ULN200xA Seven darlington array, Rev. 6. Saatavissa: http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/CD00001244.pdf. [viitattu 29.09.2011]
- 8) ProAm Cranes & Equipment, 2011. ProAm® DVC250 Camera Crane - tuotetiedot. Saatavissa: <http://www.proamcranes.com/shop/product.php?productid=16134>. [viitattu 06.10.2011]
- 9) Rautio, P. & Määttä, T. 2010. Toiminnallinen määrittelydokumentti ketterästi. Opinnäytetyö. Saatavissa: https://publications.theseus.fi/bitstream/handle/10024/7469/maatta_tiina_rautio_pirkko.pdf. [viitattu 03.10.2011]

10) Hamilton, E. 2008. Pelco PTZ Protocols, D Protocol Version 5.0.1. Saatavissa: <http://www.pelco.com/sites/global/en/sales-and-support/support-services/obtaining-pelco-protocol.page>. [viitattu 29.09.2011]

LIITTEET

Liite 1. How to Obtain and Use Pelco Protocols

Liite 2. Kamerakääntöpään sähkökuvat

Liite 3. 3D-kuvat mekaanisista ratkaisuista

Liite 4. Ohjelmiston luokkakaavio

Liite 5. "Preset"-toiminnon sekvenssikaavio



How to Obtain and Use Pelco Protocols

HELPFUL TIPS FOR PELCO PRODUCTS

Tip No.: 01-3043

Date: January 18, 2001

Product Affected: Pelco Protocols

As the manufacturer of the largest base of CCTV camera positioning devices, discreet dome systems, and video/matrix control systems in the industry, Pelco is proud to make our most popular forms of system interface information available to the public. Depending on system needs, there are three relevant protocols for interfacing third-party equipment to Pelco systems:

Controlling multiple cameras via a matrix switcher: ASCII protocol

The Pelco ASCII interface protocol is used to control our most popular matrix switching systems, and the pan/tilt or dome devices that are connected to them. Camera positioning commands sent to the switcher will be translated into the appropriate Coaxitron®, or RS-422 commands as system needs and switcher capabilities dictate (see table below). Using this protocol, a third party system may control pan/tilt/zoom functions; activate presets, patterns, or alarms; activate relays; or call-up cameras to monitors. Depending upon the capabilities of the switcher, this protocol may be used to activate macros and/or control multiplexers and VCRs that are interfaced to the switcher. Common applications for this protocol are for Access Control systems, Phone Line Transmission systems, or PC Control of both cameras and switcher. Interface via the ASCII protocol may require the use of an optional data translator.

Switcher Model Series	Data Translator	PTZ Interface Options	Mux/VCR control via Switch
CM6700	Not Required	Coaxitron, or RS-422	No
CM8500	CM8500DT	Coaxitron, Only	No
CM9500	Not Required	Coaxitron, Only	No
CM9740	CM9760DT or DT4	RS-422 Only	Yes
CM9760	CM9760DT or DT4	RS-422 Only	Yes

Controlling multiple cameras via a Genex® Multiplexer: G protocol

The Pelco “G” protocol is used to control our Genex multiplexers, and the pan/tilt or dome devices that are connected to them. Camera positioning commands sent to the multiplexer will be translated into the appropriate Coaxitron, or RS-422 commands, as system needs dictate. Common applications for this protocol are for Access Control systems, Phone Line Transmission systems, or PC Control of both cameras and multiplexer.

Direct Pan/Tilt/Zoom control of one or more camera locations: RS-422 "D" Protocol

The "D" protocol is used in Pelco's Spectra[®], and Esprit[™], positioning systems as well as the LRD41Cxx series receiver/drivers. "D" protocol can be used where direct RS-422 control of a dome or receiver/driver is necessary without switcher control. Up to 32 positioning devices may be addressed on a single RS-422 line. Note that **receiver devices may not be addressed by more than one control source**. In other words, you cannot have a Pelco system controlling a dome via Coaxitron, and another device controlling it via RS-422. If the user requirements are to have a Pelco system and the third party device controlling the same camera, then the interface should be with the Pelco system controller (multiplexer or switcher) and it will generate appropriate commands to the camera. This protocol is more difficult to implement than the ASCII protocol detailed above. Most developers will find it much more convenient to use the ASCII protocol.

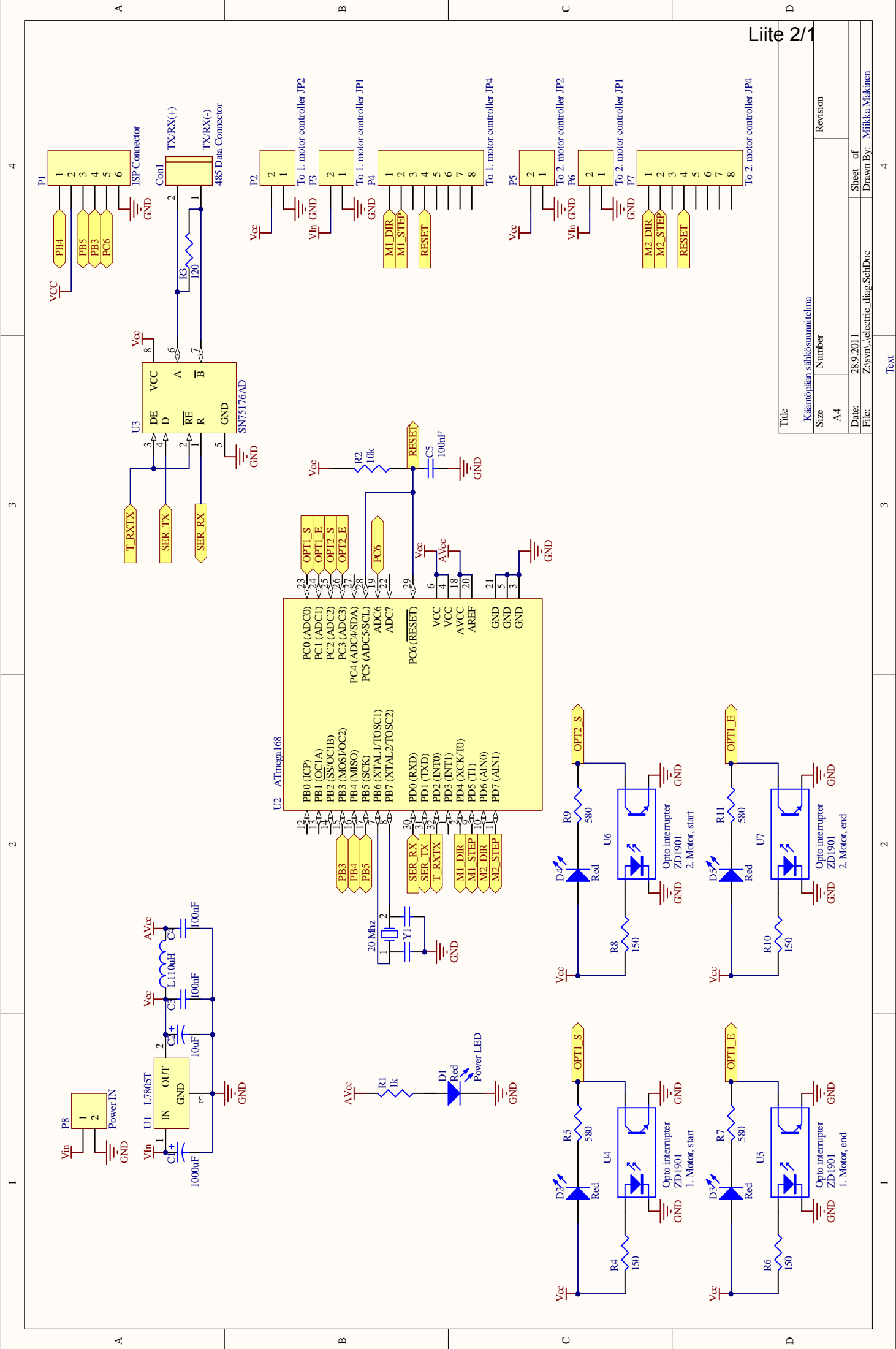
Certain Pelco products support an RS-422 protocol referred to as "P" protocol. This protocol is not published or supported for third-party use, as it is very difficult to implement without system interaction problems. From a functional standpoint, all third-party RS-422 control issues may be handled by the "D" protocol without compromise to system flexibility or performance. Pelco will eventually discontinue the use of this protocol in new products.

Requests for one or more of these protocol documents should be made via e-mail to techsupport@pelco.com. Requests should include the Company and individual name(s) making the request (with real and e-mail addresses), and a description of the intended application. We do not withhold this interface information from companies making competitive products. Contact information is maintained so that we might notify users if protocol changes are made. Application information is requested to ensure that we recommend the appropriate protocol, and so expectations may be understood and met.

Pelco makes no claims, expressed or implied, regarding the usefulness of these protocols, their implementation, or their correctness. Any use of these protocols is the sole responsibility of the agency or company implementing the protocols. While Pelco will answer limited questions about the protocols, we do not provide programming assistance.

Persons or companies who intend to use the Pelco name or trademarks should first consult the document "Policy regarding use of Pelco Trademarks" which can be found at www.pelco.com.

The contents of these documents and the function of the protocols are subject to change without notice.



Liite 2/1

Title	
Käyttöpain sähkösäunnilma	
Size	Number
A4	
Date:	28.9.2011
File:	Z:\svml\electric_diag\SchDoc
Sheet_of	4
Drawn By:	Miikka Mäkinen

Text

