

AS-0.3200 Automaatio- ja systeemitekniikan projektityöt: S11-15 LED-monimuuttujasäätö

Mikko Tuohimaa

13.5.2011

Sisältö

1	Johdanto	1
2	Systeemin kuvaus	1
2.1	Arkkitehtuuri	1
2.2	Kamera	2
2.3	Arduino Uno	2
2.4	Analogiaelektroniikka	3
2.5	PC	4
2.5.1	Liitännät	4
2.5.2	Ohjelman alusta ja rakenne	5
2.5.3	Säätöalgoritmi	6
3	Työn eteneminen	6
4	Tulokset	7
5	Yhteenveto	7

1 Johdanto

Pintaverisuonia voidaan kuvata indosyaniinivihreän (ICG) avulla, joka on suurikokoinen orgaaninen molekyyli. Se sitoutuu tiukasti veriplasman proteiineihin ja sillä on erikoiset optiset ominaisuudet: sen absorptiomaksimi on 800 nanometrin kohdalla ja emissiomaksimi puolestaan 835 nm:n kohdalla, joten valaisemalla ihoa pienemmällä aallonpituudella ja kuvaamalla sitä alipäästöfiltterin läpi, nähdään vain ICG:n tuottama toisioemissio. Koska molekyyli sitoutuu vereen, saadaan näin näkyviin verisuonisto varsin vähällä invaasiolla. Infrapunavalon tunkeutuu kudokseen ainakin muutaman senttimetrin syvyydelle, joten kuvaaminen ei rajoitu pinnan hiussuoniin. [1] [2]

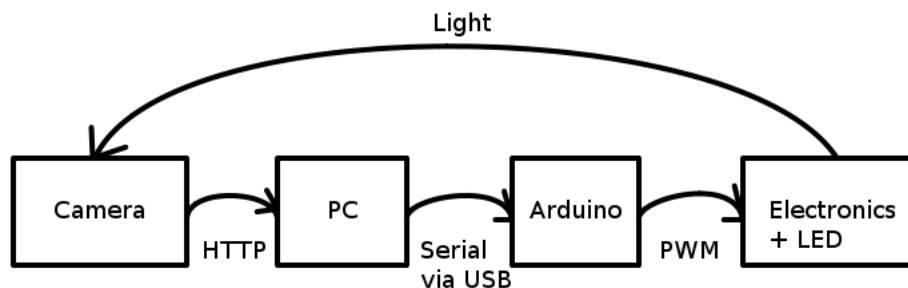
ICG:n puoliintumisaika elimistössä on potilaasta riippuen 150-180 sekuntia [1], mikä asettaa haasteita jatkuvalla kuvaamiselle esim. intravaskulaaristen operaatioiden aikana. Tässä työssä kehitettiin automaattisesti säätävä IR-LED-valaisin, joka adaptoituu ICG:n vasteeseen ajan funktiona tuottaen mahdollisimman selkeän kuvan.

2 Systemin kuvaus

2.1 Arkkitehtuuri

Kuvassa 1 on esitetty systemin kommunikaatioarkkitehtuuri. Erillisiä osajärjestelmiä ovat:

- Kamera: Elphel 353
- PC python-tulkilla
- Arduino Uno
- Tätä työtä varten erikseen rakennettu analogiaelektroniikka

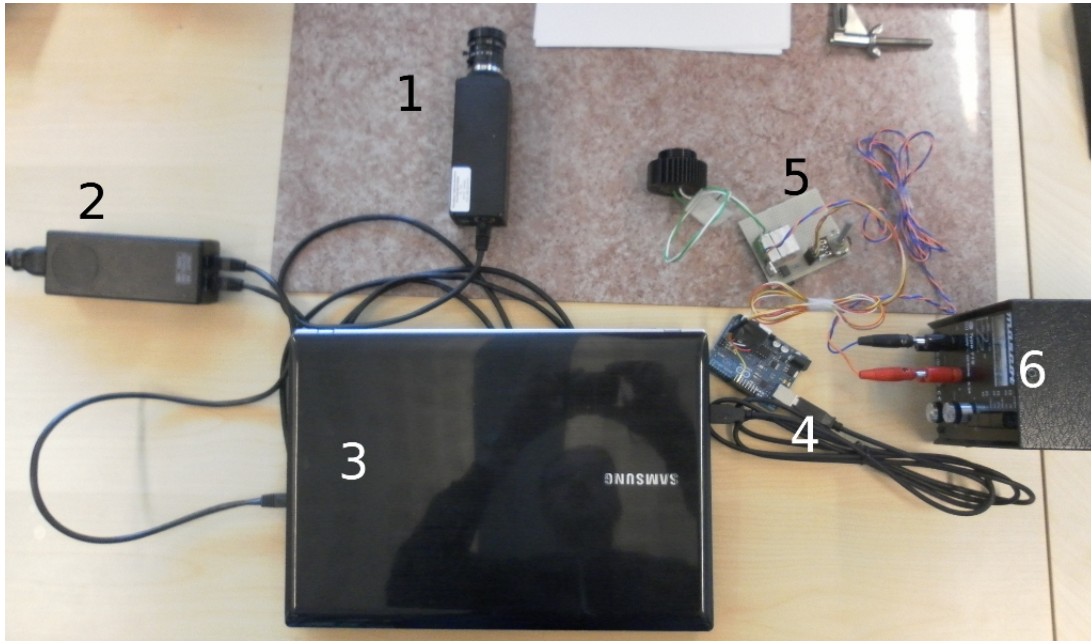


Kuva 1: Systemin kommunikaatioarkkitehtuuri

PC lukee kuvan histogrammi-informaation kameralta käyttäen Ethernet-yhteyttä, suorittaa säätöalgoritmin ja antaa ohjauskäskyn (haluttu pulssisuhde) Arduinolle

USB-portin kautta. Arduino muuttaa ohjauskäskyn pulssinleveysmodulaatiosignaaliksi ja analogielektroniikka vahvistaa sen LED-valaisimen vaatimalle tasolle.

Todellinen systeemi näkyy kuvassa 2.



Kuva 2: Todellinen systeemi. Kuvassa: 1. kamera, 2. kameran teholähde, 3. PC, 4. Arduino Uno, 5. custom-elektroniikka, 6. 12V jännitelähde.

2.2 Kamera

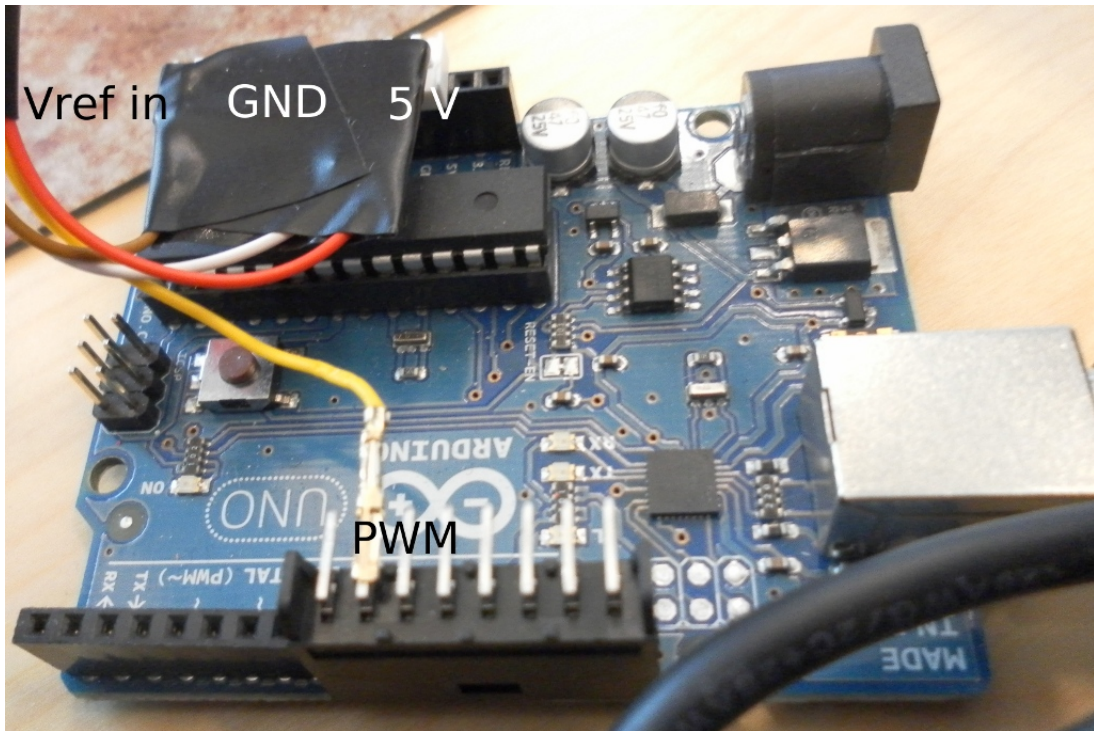
Kamerana on käytetty Elphelin mallia 353 [3]. Se on sekä raudaltaan että ohjelmistoltaan avoin, mikä helpottaa erilaisten custom-sovellusten tekoa. Se pitää sisällään FPGA:n, joka pyörittää minimalistista web-palvelinta PHP-tuella. Tätä käytettiin hyväksi ja palvelimelle kirjoitettiin PHP-ohjelma, joka pyydettäessä hakee kameran muistista senhetkisen komposiittihistogrammidatan ja tarjoilee sen välilyönnein erotettuina desimaalikoodattuina arvoina.

2.3 Arduino Uno

Arduino [4] on avoimen arkkitehtuurin projekti, jonka referenssituote Uno on. Se on tarkoitettu prototyypialustaksi, johon se onkin omiaan: kehitystyö on nopeaa ja vaivatonta, koska piirilevyltä ja ohjelmistosta löytyy valmiina tuki suurimmalle osalle usein tarvittavista ominaisuuksista, kuten sarjamuotoiselle kommunikaatiolle USB:n yli sekä digitaalisten ja analogisten sisään- ja ulostulojen monipuoliseen hallitsemiseen. Arduinon voi myös ohjelmoida vaivattomasti USB:n kautta. Tässä projektissa piiriä käytetään vain USB:n yli kommunikoitavan PWM-pulssisuhdearvon muuttamiseen PWM-signaaliksi.

Arduino-piiri saa käyttöjännitteensä USB-liitännästä. Seuraavia I/O- ja muita pinnejä on käytetty:

- *Digital Pin 9*: PWM-signaali analogiaelektronikalle
- *Digital Pin 8*: jos tämä pinni on kytkettynä GND-pinniin, piiri toimii manuaalimoodissa; jos ei, on automaattinen moodi valittuna.
- *Analog In 3*: manuaalimoodin input-signaali potentiometriltä
- *5V*: 5 voltin käyttöjännite manuaalimoodin potentiometrille
- *GND*: maataso sekä PWM:ää että potentiometriä varten

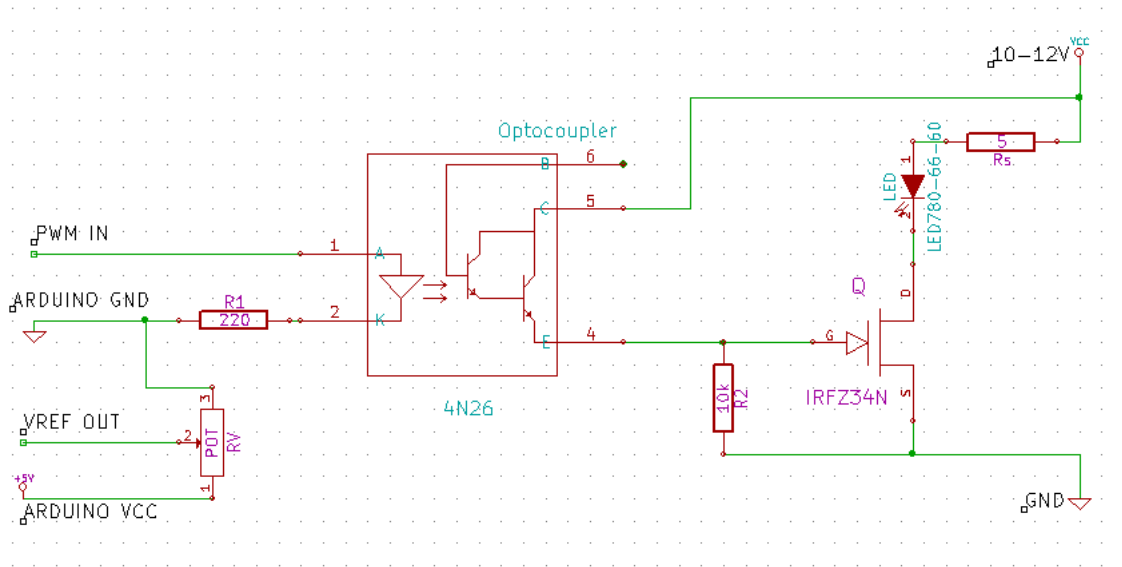


Kuva 3: Arduino Uno liittimet kytkettyinä.

2.4 Analogiaelektronikka

Kuvassa 4 näkyy rakennetun elektronikan kytkentäkaavio ja kuvassa 5 itse piirilevy. Piirin tarkoitus on vahvistaa Arduinolta tuleva PWM-signaali niin, että IR-LEDin voi liittää suoraan piiriin ilman mitään ylimääräisiä kytkentöjä. Tätä varten piiriltä löytyy myös 5 ohmin tehoetuvastus LEDin virran rajoittamiseksi.

Arduino on optisesti erotettu korkeamman käyttöjännitteen piiristä. Käytetyn IR-LEDin nimellinen jännitehäviö on 9 V ja maksimivirta 600 mA. Etuvastus on mitoitettu niin, että LEDin maksimivirta saavutetaan 12 V käyttöjännitteellä, jolloin



Kuva 4: Projektia varten toteutetun analogipiirin kytkentäkaavio

vastuksen nimellisen tehonkeston varmuusvara on vielä yli kahdeksankertainen. Haluttua maksimivirtaa voi siis säätää tietyissä rajoissa käyttöjännitettä muuttamalla, mutta toki myös maksimipulssisuhdetta ohjelmallisesti rajoittamalla. Piiri on pyritty suunnittelemaan niin, että mahdollisissa kaapelien oikosulkutapauksissakin LEDin virta on aina vastusten rajoittamaa. Toisin sanoen kumpikaan LEDin johtimista ei ole suoraan käyttöjännitteessä tai maassa kiinni.

Piirissä käytetty kytkinelementti on hyvin alhaisen johtavan moodin jännitehäviön omaava HEXFET (*IRFZ34N*). Nimellisesti se kestää 40 ampeerin virran, mutta se tuskin tulee tarpeeseen. Alhaisesta jännitehäviöstä ja kytkintoiminnallisuudesta johtuen FET ei tarvitse jäähdytinelementtiä, mutta se on joka tapauksessa sijoitettu piirin reunalle jäähdytyksen mahdollistamiseksi.

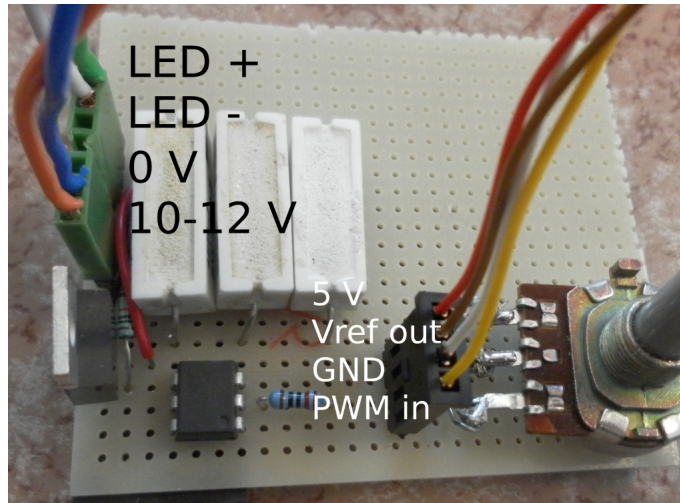
Piirissä on myös yksi potentiometri, jolla voi manuaalisesti säätää Arduinon tarjoaman PWM-signaalin pulssisuhdetta. Tämä tehtiin lähinnä debug- ja mittaustarkoituksiin.

12 voltin käyttöjännitteellä piirin hyötysuhde (LEDin kuluttama teho jaettuna kokonaisteholla) on n. 75%; hukatehosta suurin osa (1,8 W maksimipulssisuhteella) muuttuu lämmöksi etuvastuksessa.

2.5 PC

2.5.1 Liitännät

PC keskustelelee kameran kanssa Ethernetin välityksellä. Kamera kytketään suoraan PC:n Ethernet-porttiin ja tietokoneen IP-asetukset on konfiguroitava sisäverkkokoalulle 192.168.0.xxx. Kameran IP on oletuksena 192.168.0.9. Kameraa pystyy käyttämään ja manipuloimaan sekä sen HTTP-serverin että suoran SSH- tai TELNET-



Kuva 5: Projektia varten toteutettu analogiapiiri ja sen liitännät

yhteyden kautta. Kameran sisäisen FPGA:n käyttöjärjestelmänä on riisuttu Linux. Arduinon kanssa puolestaan keskustellaan virtuaalisen sarjaportin kautta. Fyysisesti Arduino liitetään USB-porttiin, ja tunnistettuaan laitteen PC luo sitä vastaavan virtuaalisen COM-portin. Käyttöä varten on tunnistettava kyseinen portti ja tallennettava sen tunniste konfiguraatitiedostoon. Tämä toimii sekä Windows- että Linux-alustalla, mutta portin tunniste vaihtelee alustasta riippuen.

2.5.2 Ohjelman alusta ja rakenne

Säätöön tarkoitettu ohjelma on tehty Pythonilla ja sitä on ajettu *Python(x,y)* -sovelluksella Windows-ympäristössä. Käyttöjärjestelmällä tai virtuaalikoneella ei ole merkitystä, mutta *pySerial*-moduuli tulee olla asennettuna sarjaliikennettä varten. Ohjelma on säikeistetty ja jokseenkin modulaarinen; olio-ohjelmointia on hyödynnetty.

Ohjelma koostuu pääohjelmasta ja muutamasta lisämoduulista. Jako on seuraavanlainen:

- *control.py*: main-funktio, muutama apufunktio mm. konfiguraation parsimiseen sekä säikeistetty Controller-luokka, joka on vastuussa itse säädöstä.
- *SerialInputMonitor.py*: moduuli sisääntulevan sarjaliikenteen monitorointiin. Säikeistetty.
- *HistogramMonitor.py*: moduuli histogrammidatan visualisoimiseen. Ei käytetty, koska kameran “Advanced Webcam” -sovellus sopii paremmin tarkoitukseen.
- *config.conf*: sisältää mm. kameran IP-osoitteen, käytetyn COM-portin tunnisteen sekä muita konfiguraatioparametreja.

2.5.3 Säätoalgoritmi

Säätoalgoritmi on PID-johdannainen ja perustuu kameran saatavaan histogrammi-dataan. Olennaista on välttää informaation hukkaaminen, joten ylivalottumisen esto on prioriteettilistassa korkeimpana - siis suurimman painokertoimen virhetermiä muodostettaessa saa histogrammin yläpään painotettu keskiarvo. Toisena prioriteettina on alivalottumisen esto, ja kolmantena histogrammin painopisteen säätäminen haluttuun kohtaan (tällä hetkellä kirkkausarvo 172). Säätoalgoritmin voi esittää kaavana:

$$e_n = a \cdot \sum_{i=240}^{255} (i - 239) \cdot h_{n,i} + b \cdot \sum_{i=0}^{10} (11 - i) \cdot h_{n,i} + c \cdot \left(L - \frac{\sum_{i=0}^{255} i \cdot h_{n,i}}{\sum_{i=0}^{255} h_{n,i}} \right) \quad (1)$$

$$s_n = s_{n-1} + P \cdot e_n + D \cdot \frac{de}{dt}, \quad (2)$$

jossa e_n on virhetermin arvo hetkellä n , $h_{n,i}$ on histogrammin alkion i arvo ajan hetkellä n , a, b, c, P, D painokertoimia, L haluttu histogrammin painopiste ja s_n ohjauksen (eli halutun pulssisuhteen) arvo hetkellä n .

Lyhyesti tiivistettynä ohjausarvoa siirretään joka iteraatio havaittua virhettä korjaavaan suuntaan. Aika-askel on 250 millisekuntia. Kameran ominaisuudet sallisivat suuremmankin säätoataajuuden, mutta sitä ei nähty tarpeelliseksi.

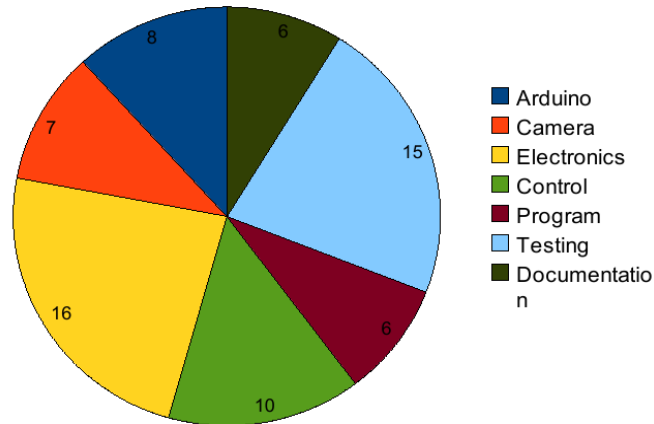
3 Työn eteneminen

Työ päästiin aloittamaan kunnolla vasta huhtikuun puolella, kun kaikki laitteet saatiin paikalle. Tätä ennen oli saatu tehtyä arduino-kommunikaatio ja myös runko pääohjelmalle sekä tutustuttu kirjallisuuteen. Heti laitteiden saapumisen jälkeen n. yksi työpäivä käytettiin kameran liittämiseen PC:een ja sen ominaisuuksien tutkimiseen. Todettiin, että jo lähes valmiiksi tehtyä histogrammimonitoria ei tarvitakaan toiminnan visualisoimiseksi, koska kamerassa löytyy valmiina *Advanced Webcam* -sovellus, joka näyttää myös histogrammidatan kuvan ohessa.

Yleisesti ottaen lähes kaikkia projektin osa-alueita kehitettiin enemmän tai vähemmän rinnakkain. Ainoastaan Arduinon koodi saatiin lopputilaansa aikaisin, jo maaliskuussa. Työ oli itsenäistä, osittain koska ohjaaja oli harvoin paikalla mutta myös koska projektin suuntaviivat olivat alusta asti selvät eikä erityisiä ongelmia esiintynyt.

Kokonaisuudessaan, hieman yllättäenkin, suurin osa käytetystä ajasta kului elektronikan rakentamiseen, vaikka varsinaisia iteraatiokierroksia ei suunnittelussa esiintynytäkään. Myös yleiseen testaukseen ja parametrien optimointiin kului paljon aikaa. Loppuaika jakautui melko tasan eri laitteiden välisen kommunikaation toteuttamiseen, ohjelmointiin ja itse säätoalgoritmin toteuttamiseen. Arvioitu tuntitilanne ennen tämän dokumentin kirjoittamista on esitetty kuvassa 6. Loppuraportin sekä

koodien siistimisen ja kommentoinnin myötä ”dokumentaatio”-kohtaan on lisättävä n. 7h, jolloin yhteistuntimääräksi tulee n. 75h. Kyseessä on arvio tehokkaasta työajasta; toki tunteja kului työhuoneella useampiakin.



Kuva 6: Työtuntijakauma työn eri osa-alueiden kesken. Kaaviosta puuttuu tämän dokumentin tekemiseen sekä koodien siistimiseen ja kommentointiin käytetty aika, 7h.

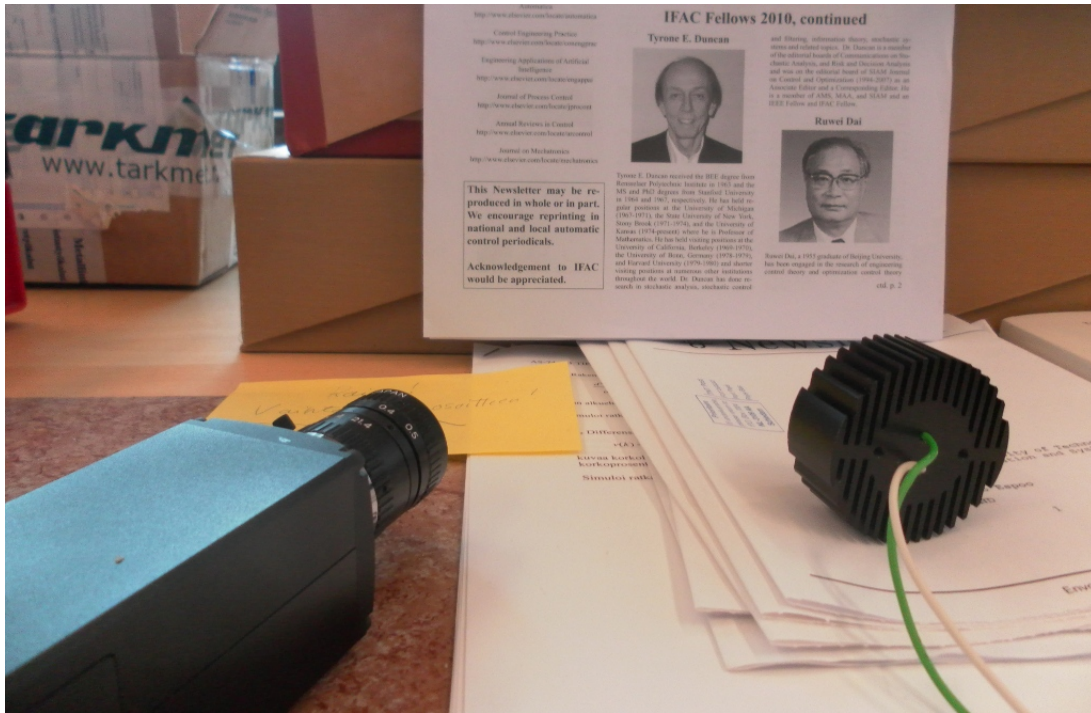
4 Tulokset

Alkuperäistä käyttötarkoitusta, ICG-kuvantamista varten tarvittavia laitteita (IR-filtterit, varjoaine, koe-eläin tai -henkilö) ei ollut saatavilla, siispä systeemin suorituskykyä mitattiin kuvan 7 osoittamalla koejärjestelyllä. Kameran valotus säädettiin sellaiseksi, että LEDin ollessa pois päältä kameran edessä ollut paperi juuri ja juuri näkyi - tämä oletettiin toivotuksi vasteeksi IR-filtteriä käytettäessä. Kameran valotus pidettiin kiinteänä koko kokeen ajan.

Systeemin käyttäytymistä tutkittiin siirtymällä manuaalimoodista automaattimoodiin sekä ohjausarvosta 0 että 255 (täysi pulssisuhde), ja myös siirtämällä LEDiä lähemmäs tai kauemmas. Systeemin vaste oli nopea ja kaikissa tapauksissa yli- tai alivalotus eliminoidiin nopeasti, 1-4 iteraation kuluessa eli yleensä alle sekunnissa. Vähemmän tärkeä histogrammin painopisteen siirto asettui n. 5 sekunnin kuluessa. Systeemi oli koko testin ajan stabiili, värähtelyjä ei esiintynyt. Kuvassa 8 näkyy kamerakuva nolla- ja maksimipulssisuhteella sekä säädön tulos.

5 Yhteenveto

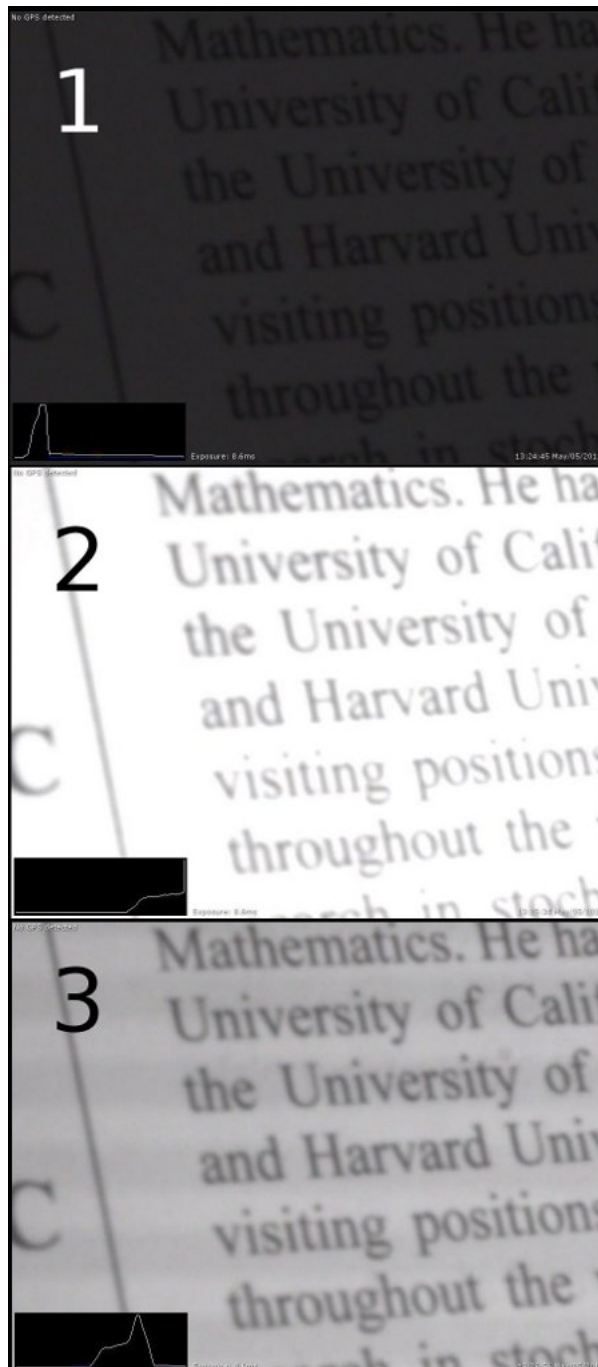
Kehitettiin kamerapohjainen säätöjärjestelmä LED-valaisimelle. Projekti sisälsi kommunikaatorajapintojen luontia ja konfigurointia, säätöalgoritmin kehitystä ja toteutusta, elektroniikkasuunnittelua ja -toteutusta sekä yleistä ohjelmointia. Tulokset olivat hyviä, laite toimii vaatimusten mukaisesti. Työhön kului n. 75 tehokasta työtuntia.



Kuva 7: Koejärjestely - kamera katsoo paperia, jota systeemin ohjaama IR-LED valaisee.

Mahdollisia jatkokehityksen kohteita:

- Arduinon ja custom-elektroniikan integrointi, kenties kokonaan oman alustan luominen sisältäen elektroniikan, mikrokontrollerin ja RS232/USB-konvertterin
- Laitteen kotelointi ja liitinten standardointi
- On kenties mahdollista eliminoida PC kokonaan ja ajaa säätöalgoritmia suoraan mikrokontrollerissa. Tämä kuitenkin edellyttää Ethernet-rajapinnan luomista mikrokontrollerille.



Kuva 8: Suorituskykytestin tulos: systeemi säätää valaistuksen kameralle sopivaksi nopeasti ja tarkasti. Kuva 1: LED pois päältä. Kuva 2: LEDiä ajetaan maksimipulsisuhteella. Kuva 3: LED säädetty optimaaliseksi.

Viitteet

- [1] Owens, S. L. Indocyanine green angiography. *British Journal of Ophthalmology*, 1996. Vol. 80. S. 263–266.
- [2] Wikipedia. Indocyanine Green. http://en.wikipedia.org/wiki/Indocyanine_green. Viitattu 10.05.2011.
- [3] Elphel website. <http://www3.elphel.com/>.
- [4] Arduino Website. <http://www.arduino.cc/>.