

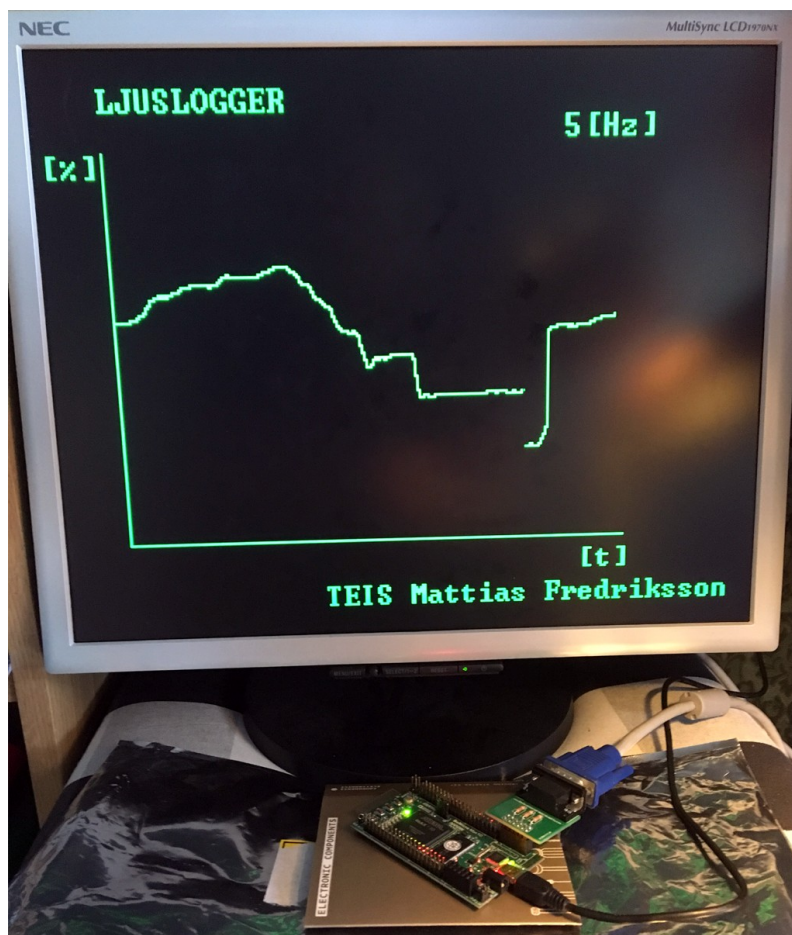
## TEIS

## Teknisk rapport

Författare	Datum	Uppgift	
Mattias Fredriksson	2016-01-03	C Engineeringjob – konstruktionsrapport ljuslogger	
E-postadress	Kontrollerad	Version	Fil
<a href="mailto:mattias.fredriksson@gmail.com">mattias.fredriksson@gmail.com</a>		A4	mattias_fredriksson_c_engineeringjob_konstruktionsrapport

# TEIS

## Ljuslogger med visuell datapresentation



### Sammanfattning

Rapporten beskriver system för ljusloggning med visuell datapresentation på VGA-skärm.

# Innehållsförteckning

1. Kravspecifikation.....	3
2. Förstudie.....	3
3. Projektplan.....	4
4. Testprotokoll.....	4
5. Konstruktion.....	4
5.1 Hårdvara.....	5
5.1.1 Utvecklingsplattform.....	5
5.1.2 Arkitektur.....	5
5.1.3 Hierarki.....	6
5.1.2 Minneskarta.....	6
5.1.2 Delsystem.....	7
5.1.2.1 ALTERA MAX 10 FPGA.....	7
5.1.2.1 CPU NIOS II/e IP.....	7
5.1.2.2 SDRAM.....	7
5.1.2.3 Fotoresistor.....	7
5.1.2.4 ADC.....	7
5.1.2.5 SPI DAC.....	8
5.1.2.6 VGA_IP.....	8
5.1.2.7 TIMER_IP.....	8
5.2 Mjukvara.....	8
5.2.1 Drivrutiner VGA_IP.....	8
5.2.1.1 Grafikfunktioner VGA_IP.....	9
5.2.1 Drivrutiner modular_adc_0.....	10
5.2.1 Drivrutiner TIMER_IP.....	10
5.2.2 Program: Ljusdatalogger.....	10
5.2.2.1 Beskrivning av program.....	10
5.2.2.2 Flödesschema för program.....	11
5.2.2.3 Funktioner i program.....	11
6. Verktyg.....	12
6.1 Inställningar av BSP.....	12
7. Verifiering.....	12
7.1 Testprotokoll efter verifiering.....	13
8. Analys.....	14
8.1 Varningar.....	14
8.2 Minimering av kod.....	14
8.2.1 Ej optimerad kod.....	14
8.2.2 Optimerad kod.....	14
8.3 Buggar.....	15
9. Validering.....	15
9.1 Bilder från Validering.....	15
9.2 Ifyllt testprotokoll efter validering.....	16
9.3 Film.....	16
10. Sammanfattning och förbättringar.....	17
11. Versionshistorik.....	17
Referenser.....	17
Bilagor.....	18

# 1. Kravspecifikation

Kravspecifikation för konstruktion av ljusdatalogger beskrivs i tabell 1.1.

Tabell 1.1. Kravspecifikation för ljusdatalogger.

Krav ID	Beskrivning	Utfört
<b>Plattform</b>		
1	Avsedd utvecklingsplattform är BeMicro Max 10 med tillhörande VGA-adapter från AGSTU AB.	Ja
<b>Funktion</b>		
2	<p>Ljusdatalogger med visuell presentation av data på VGA-skärm ska utvecklas och stöda skärmupplösningen 320 x 240 pixlar. Detta medför även utveckling av drivrutiner för VGA-grafik till plattformen.</p> <p>Konstruktionen ska vid uppstart visa en bild som presenterar systemet och sedan efter knapptryck ljusloggningsgraf.</p> <p>Samplingshastigheter: 5, 2, 1, 0,1 Hz och så fort som möjligt.</p> <p>Användaren ska kunna cykla igenom de olika samplingshastighet via knapptryckning på knapp 1. Samtidigt visas aktuellt samplingsläge med LEDS på kortet och på skärmbilden.</p>	Ja
<b>Leveranskrav</b>		
3	En förstudie med tidsplan och budgeterad kostnad (den här rapporten).	Ja
4	<p>Slutleveransen ska vara en .rar-fil innehållande två mappar med följande dokument:</p> <p>Mappen konstruktionsbeskrivning: Konstruktionsrapport (med TEIS-rapportregler och enligt <a href="http://agstuworld.se/TEIS/2015TEIS/2015_kurs_C/V12_Ingenjorsjobb_start/C_ing.pdf">http://agstuworld.se/TEIS/2015TEIS/2015_kurs_C/V12_Ingenjorsjobb_start/C_ing.pdf</a>)</p> <p>Mappen konstruktionsdokument: Alla C-filer i (software/BSP och Projekt mapp) Alla IP-komponenter (IP mapp) SOF- och SOPCINFO-filen</p>	Ja

## 2. Förstudie

Inget av de tidigare projekten har skapat någon realvärldsteknisk lösning likt den ljusdatalogger vilken tillämpar sensorer som är tänkt att utvecklas här, de har mer fokuserat på spelindustrin.

Dock har majoriteten av de tidigare projekten utvecklat omfattande VGA-grafik vars tillämpning är lika oavsett om de används för att skriva ut en graf eller rita ut ett rymdskepp.

De mest framträdande utvecklingsprojekten vilka har bidragit med underlag till ljusloggern är

Meteorstorm av Lars Bengtsson, där finns förutom en VGA-tillämpning även en bra LCD-drivrutin och Rymdkrig av Rafi Normanian, vars spel onekligen verkar kul att spela.

### 3. Projektplan

Tabell 3.1. Projektplan.

Tidsplan och budgeterad kostnad för ljusdatalogger

Aktivitet \ 2015 vecka x	V48	V49	V50	V51	V52	V53*	Timmar** [h]	Kostnad** [kr]
Kravspecifikation och förstudie	4						4	2000
Bekanta sig med systemarkitektur	2						2	1000
Konstruera drivrutiner för VGA och ADC	6	12	10				28	14000
Konstruera ljusdatalogger			2	12	10		24	12000
Verifiering					1	1	2	1000
Validering					1	1	2	1000
Slutrapport						10	10	5000
Summor:	12	12	12	12	12	12	72	36000

\* 2015 innehåller 53 veckor vilket inträffar ungefär vart 5 år.

\*\* Var vecka avses 12 timmar avsättas för projektet a´ 500 kr/h.

Utveckling kommer ske parallellt med annat projekt till samma kund där tidsfördelningen mellan projekten är 28 respektive 12 timmar per vecka i vardera projektet vilket visas i tabell 3.1.

Projektet har utförts inom ramen för tidsplanen.

### 4. Testprotokoll

Tabell 4.1. Testprotokoll.

Testfall	Beskrivning	Resultat	OK
1	Skärmmminnet rensas vid uppstart		
2	Presentationsbilden för systemet skrivs ut på VGA-skärmen		
3	Vid tryck på knapp 1 ritas bakgrunden till grafen upp och loggning med utskrift i realtid visas. LED1 är släckt.		
4	Vid ytterligare ett tryck på knapp 1 byts uppdateringsfrekvensen för loggningen till 5 Hz. LED1 lyser.		
5	Vid ytterligare ett tryck på knapp 1 byts uppdateringsfrekvensen för loggningen till 2 Hz. LED 2 lyser		
6	Vid ytterligare ett tryck på knapp 1 byts uppdateringsfrekvensen för loggningen till 1 Hz. LED1 och 2 lyser.		
7	Vid ytterligare ett tryck på knapp 1 byts uppdateringsfrekvensen för loggningen till 0.1 Hz. LED4 lyser.		
8	Vid ändring av ljusstyrka mot fotoresistorn ändras grafens presentation på skärmen på sådant vis att det uppfattas relevant för betraktaren.		

Testprotokollet i tabell 4.1 har kravspecifikationen till grund.

### 5. Konstruktion

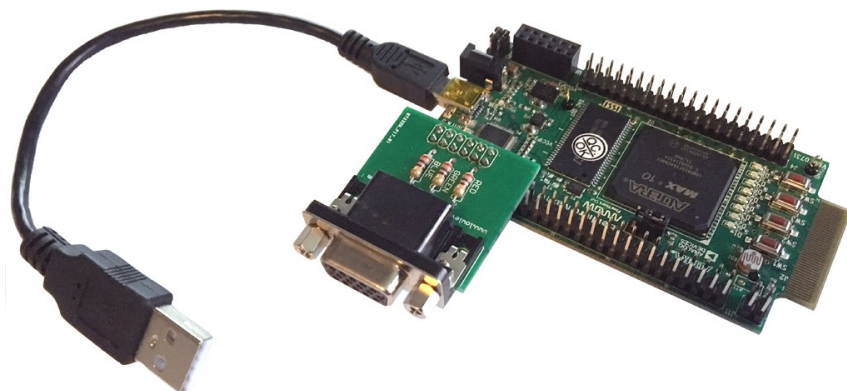
Under avsnitt 5 beskrivs den hård- och mjukvara som används för konstruktion av ljusdataloggern.

## 5.1 Hårdvara

I avsnittet 5.1 beskrivs den hårdvara och IP (immaterialrättsprodukter) som tillämpas för konstruktion av ljusdataloggern. IP för konfiguration av FPGA är sammansatt av TEIS AB som ett mikrodatorsystem för att i samverkan med den fysiska hårdvaran på BeMicro Max 10-kortet bilda en anpassningsbar utvecklingsplattform.

### 5.1.1 Utvecklingsplattform

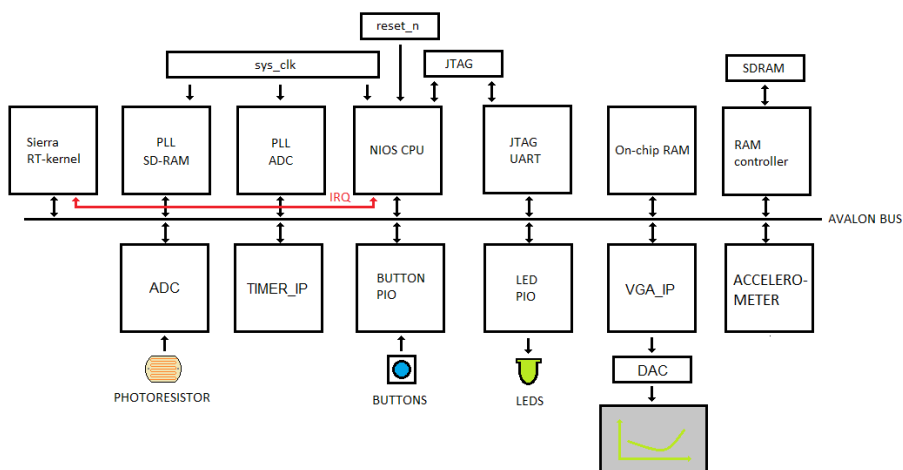
BeMicro Max 10 från Arrow Eletronics med VGA-adapter-kort från AGSTU installerat i PMOD A&B, vilket kan ses i figur 5.1, tillämpas som utvecklingsplattform.



Figur 5.1. BeMicro Max 10 med VGA-adapter i PMOD A&B.

### 5.1.2 Arkitektur

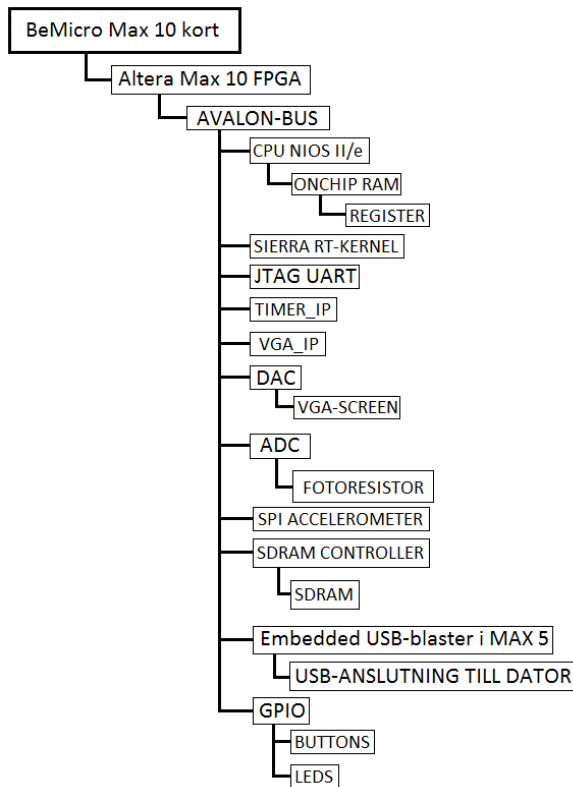
I figur 5.1 visas utvecklingsplattformens systemarkitektur i sin helhet, dock tillämpas den ej till fullo i konstruktionen av ljusdataloggern.



Figur 5.2. Systemarkitektur.

### 5.1.3 Hierarki

I figur 5.3 visas hierarkin hos systemet.



Figur 5.3. Hierarki.

### 5.1.2 Minneskarta

I figur 5.4 visas minneskarta för systemet.

Device	Address Range	Size (bytes)	Attributes	View
jtag_uart_0	0x01203680 - 0x01203687	8	printable	jtag_uart_0
sysid_qsys_0	0x01203678 - 0x0120367F	8		sysid_qsys_0
modular_adc_0_sequencer_csr	0x01203670 - 0x01203677	8		modular_adc_0_sequencer_csr
TIMER_HW_IP_0	0x01203660 - 0x0120366F	16		TIMER_HW_IP_0
ADC_PLL	0x01203650 - 0x0120365F	16		ADC_PLL
SDRAM_PLL	0x01203640 - 0x0120364F	16		SDRAM_PLL
pio_key_1_3	0x01203630 - 0x0120363F	16		pio_key_1_3
pio_LED	0x01203620 - 0x0120362F	16		pio_LED
spi_accelerometer	0x01203600 - 0x0120361F	32		spi_accelerometer
modular_adc_0_sample_store_csr	0x01203400 - 0x012035FF	512		modular_adc_0_sample_store_csr
onchip_memory2_0	0x01201000 - 0x01201FFF	4095	memory	onchip_memory2_0
IP_VGA_0	0x01000000 - 0x011FFFFFF	2097152		IP_VGA_0
sdram_controller	0x00800000 - 0x00FFFFFF	8388608	memory	sdram_controller
sierra_0	0x00000000 - 0x0000003FF	1024		sierra_0

Figur 5.4. Minneskarta från linkern i Eclipse bsp-editor.

## 5.1.2 Delsystem

Här presenteras de komponenter på utvecklingsplattformen vilka tillämpas i ljusdataloggern.

### 5.1.2.1 ALTERA MAX 10 FPGA

MAX 10 FPGA\_n som finns på BeMicro Max 10 kortet har 8000 Logiska element, 414 kbit M9K minne, 256 kbit flash (användardefinierbart), 2 PLL:er, 24 st 18 x 18 multipliers, 1st internt ADC-block med 17 tillgängliga kanaler och 1 temperaturmätare samt 250 st GPIO varav 80 finns tillgängliga via pinnar och kantanslutningar. M9K minnet är true dual block minne med register input [1] & [4].

### 5.1.2.1 CPU NIOS II/e IP

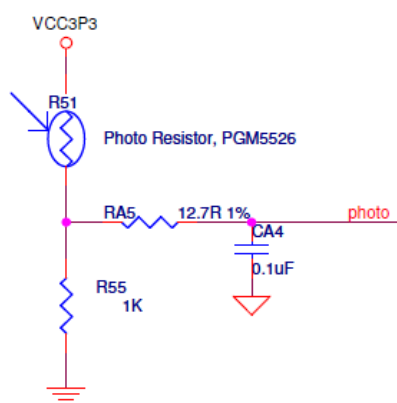
NIOS II/e är en 32-bitar RISC CPU\_IP från Altera som FPGA:n konfigureras med [5]. CPU:n utnyttjar systemklockan på 50 Mhz vilket återfinns i "system.h"-filen.

### 5.1.2.2 SDRAM

8 MB SDRAM ISSI IS42S16400J-7TL finns tillgängligt på kortet vilka adresseras via SDRAM-kontroller [1].

### 5.1.2.3 Fotoresistor

Fotorresistorn PGM5526 återfinns på BeMicro Max 10-kortet, vilket visas i ett urklipp från schemat i figur 5.5, ansluten på FPGA:ns interna ADC på pinne J9. Den varierar i resistans mellan 8(mörkt)-20(ljust) kohm beroende av ljusstyrka den utsätts för [3]. Den kommer i den här tillämpning att förse systemet med ljusdata från sin omgivning via ADC:n.



Figur 5.5. Schema för fotoresistorns anslutning på BeMicro Max 10-kortet.

### 5.1.2.4 ADC

ADC:en som finns integrerad i FPGA:n har en bugg där VVC-railen till en I/O-bank är ansluten till fel spänningsnivå, 3.3 V, vilket ger värden vilka ej överensstämmer med de som kan mätas upp med

oscilloskop. För att åtgärda buggen behöver den omanslutas till 2.5 V [1].

Hänsyn behöver tas vid tillämpning av exempelvis temperaturmätare där det är viktigt med skalan för mätningen. Vid tillämpning av fotoresistorn är dock skillnaden negligerbar för ADC:n.

Den integrerade ADC:n har en 12-bitars omvandling med 18 kanaler av vilka 17 är tillgängliga utanför FPGA:n [1]. Den kan utföra upp till 1 miljon mätningar per sekund beroende av systemklockan/PLL. Spänningsnivån är här 0-3.3V men skulle varit 0-2.5 V för fullgod funktion. Upplösningen är 12 bitar, dvs  $2^{12} = 4096$  vilket ger  $1 \text{ LSB} = 3.3 \text{ V} / 4096 = 805.66 \mu\text{V}$  för 0-3.3 V och  $1 \text{ LSB} = 2.5\text{V} / 4096 = 610.35\mu \text{ V}$  vid 0-2.5 V.

#### 5.1.2.5 SPI DAC

DAC:en som finns på BeMicro-kortet är en AD5681 med spänningsnivå 2.5 V och 12-bitars omvandling upp till 50 Mhz. Interface via SPI [7]. Outputen spänningsområde är  $2 \times V_{\text{ref}}$ , i det här fallet 5 V.

#### 5.1.2.6 VGA\_IP

VGA\_IP är en VGA-kontroller utvecklad av TEIS för VGA-tillämpningar med upplösning 640 x 320. Den har validerats på Alteras FPGA:er och är anpassad att fungera tillsammans med NIOS II och avalonbussen. Den kräver en systemklocka på 50 Mhz och en 25 Mhz klocka för VGA\_CLK samt en extern DAC för att fungera [8].

För tillämpning på BeMicro Max 10 kortet via SPI DAC krävs att en VGA-adapter från AGSTU installeras i PMOD A&B, se figur 5.1.

#### 5.1.2.7 TIMER\_IP

TIMER\_IP är en timer för tillämpning i Alteras FPGA:er, utvecklad av TEIS för att mäta tid, den kan max hantera 32 bitar tidsdata.

### 5.2 Mjukvara

I avsnitten nedan beskrivs drivrutiner och övriga mjukvara utvecklade för ljusdatalogger-tillämpningen.

#### 5.2.1 Drivrutiner VGA\_IP

Drivrutiner för att utnyttja VGA\_IP med upplösning 320 x 240 har tagits fram:

```
VGA_WRITE(adress, data) //skriver 3 bitar data till adress i minnet
```

```
VGA_READ(adress) //läser data från adress i minnet
```

där

```
adress = x + y * 640 //adress anpassad för 320 x 640 skärmupplösning
```

```
Data = Integer mellan 0-7 för RGB-färg, exempelvis 0 = svart och 7 = vit.
```

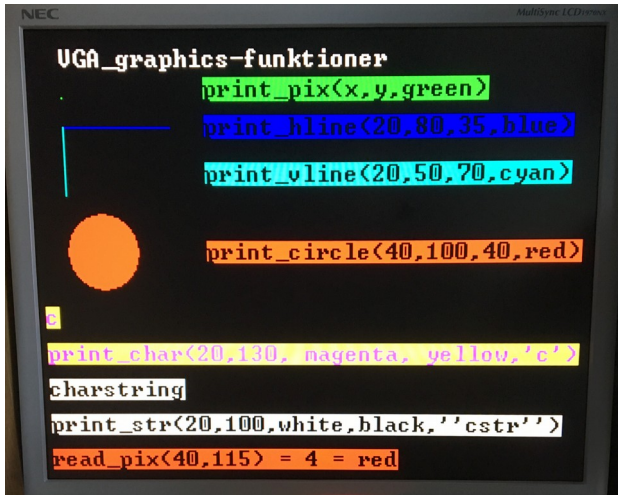
För mer ingående beskrivning se "VGA\_sw\_user\_manual.pdf" vilken ligger i mappen doc under



### 5.2.1.1 Grafikfunktioner VGA\_IP

Grafikfunktionerna beskrivs mer ingående i "VGA\_sw\_user\_manual.pdf" vilken ligger i mappen doc under VGA\_IP.

I figur 5.6. visas en demonstrationsbild för de utvecklade grafikfunktionerna vilka sammanfattas kort i tabell 5.1 nedan.



Figur 5.6. Utvecklade grafikfunktioner för VGA\_IP.

Tabell 5.1. Grafikfunktioner för VGA\_IP.

Funktion	Beskrivning	Tidsåtgång	enhet
print_pix(alt_u16 x,alt_u16 y, alt_u8 rgb)	Skriver ut data (3 bitar) för en pixel på position x,y i grafikminnet, dvs på skärmen, data = färgen i rgb.	1519	cykler
print_hline(alt_u16 x,alt_u16 x_end,alt_u16 y,alt_u8 rgb)	Skriver ut en horisontell linje på skärmen från x till x_end i y med färgen rgb. Tillämpar funktionen print_pix(); Tidsåtgången beror av linjens längd.	111225	cykler
print_vline(alt_u16 x,alt_u16 y, alt_u16 y_end, alt_u8 rgb);	Skriver ut en vertikal linje mellan y och y_end i x med färgen rgb. Tillämpar funktionen print_pix();. Tidsåtgången beror av linjens längd.	68094	cykler
print_circle(alt_u16 radie, alt_u16 x_mitt, alt_u16 y_mitt, alt_u8 rgb)	Skriver ut en med färgen rgb fylld cirkel vars radie utgår från mitten av x_mitt och y_mitt. Tillämpar funktionen print_pix();. Tidsåtgången beror av cirkelns storlek.	66404069	cykler
clear_screen(alt_u8 rgb);	Skriver ut färgen rgb till hela skärmen. Tillämpar print_pix();. Används för att rensa hela grafikminnet. Tidsåtgången beror av skärmapplösningen.	1444164155	cykler
print_char(alt_u16 x, alt_u16 y, rgb, rgb_bg, 'c')	Skriver ut ett ASCII-tecken i skärmminnet vilket hämtas från VGA_font.h.	230583	cykler
print_str(alt_u16 x, alt_u16 y, rgb, rgb_bg, "charstring")	Skriver ut en av användaren definierad sträng med ASCII-tecken i skärmminnet. Tecknen hämtas från VGA_font.h. Tillämpar print_char();. Tidsåtgången beror på strängens längd.	2325236	cykler
alt_u8 read_pix(alt_u16 x_read, alt_u16 y_read)	Läser data (3 bitar) för en pixel i positionen x och y.	1540	cykler

För att visa en skärmbild där samtliga funktioner demonstreras kan funktion nedan användas:

```
print_VGA_functions() // skriver ut en demonstrationssida för grafikfunktionerna
```

### 5.2.1 Drivrutiner modular\_adc\_0

En förenklad tilläggsdrivrutin till den ordinarie drivrutinen har utvecklats för att initiera och läsa av den ADC-kanal där fotoresistorn är ansluten.

```
ADC_ENABLE_CSR() // konfigurerar ADC:n att sampla 1 gång  
ADC_ENABLE_SSR() // konfigurerar ADC:n att sampla hela tiden  
ADC_READ() // läser av den ADC-kanal där fotoresistorn är ansluten och  
            returnerar 32 bitar data.
```

### 5.2.1 Drivrutiner TIMER\_IP

Tillämpning av drivrutiner för TIMER\_IP beskrivs mer ingående i ”TIMER\_SW\_USER\_MANUAL.pdf” dock görs en kort sammanfattning nedan.

```
TIMER_START // startar timern  
TIMER_STOP  // stoppar timern  
TIMER_READ  // läser av tidsdata  
TIMER_RESET // resetar timern
```

### 5.2.2 Program: Ljusdatalogger

Huvudprogrammet vilket har utvecklats för ljusdataloggerfunktionalitet hos plattformen.

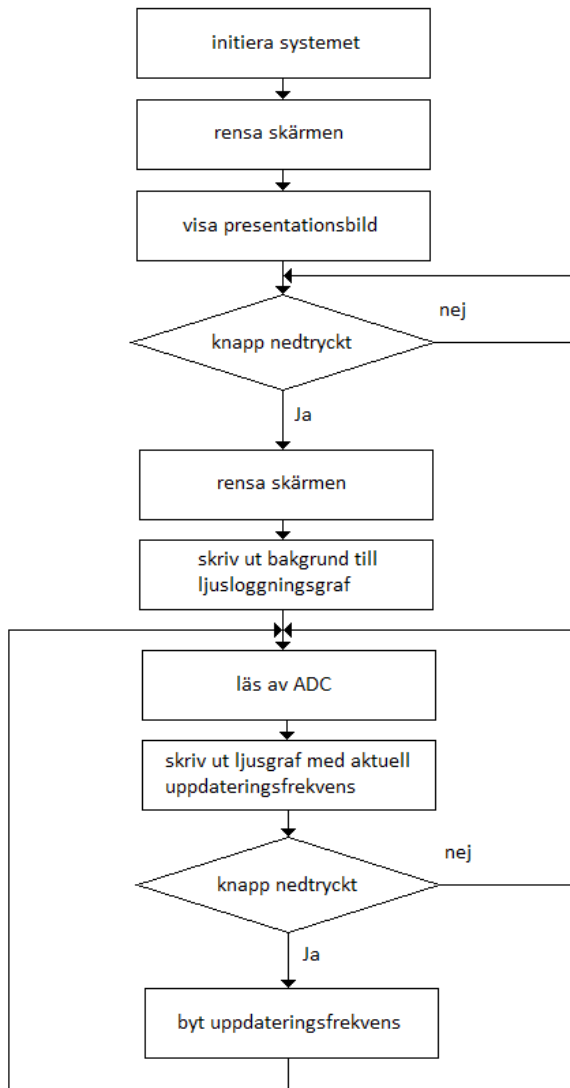
#### 5.2.2.1 Beskrivning av program

I enlighet med kravspecifiaktionen har en ljusdatalogger med visuell presentation av data på VGA-skärm har utvecklats med stöd för skärmupplösningen 320 x 240 pixlar.

Konstruktionen visar en bild som presenterar systemet vid uppstart och efter knapptryck ljusloggningsgraf. Samplingshastigheter för ljusloggningen är 5, 2, 1, 0,1 Hz och så fort som systemet kan skriva ut data (AFAP).

Användaren ska kunna cykla igenom de olika samplingshastighet via knapptryckning på knapp 1. Samtidigt visas aktuellt samplingsläge binärt med LEDS på kortet och decimalt på skärmbilden med ljusintensitetsgrafen vilket visas i figur 5.4.

### 5.2.2.2 Flödesschema för program



Figur 5.4. Flödesschema för ljusdatalogger.

### 5.2.2.3 Funktioner i program

Nedan är de funktioner vilka deklarerats och tillämpas i main-satsen för att rita ut bakgrund och visa grafen.

```
ADC_print_bg_graph()    // skriver ut grafens bakgrund med x- och y-axel samt  
                        beskrivande text såsom konstruktörens namn på skärmen
```

```
ADC_print_graph(sample_rate) // skriver ut grafen med aktuell uppdateringsfrekvens på  
                             skärmen
```

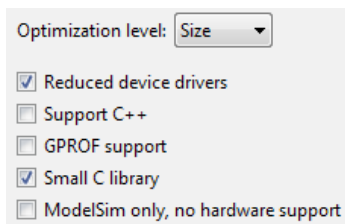
där

sample\_rate = uppdateringsfrekvenser; så fort som möjligt (AFAP), 5, 2, 1 och 0,1 Hz.

## 6. Verktyg

Verktyget som har använts vid utveckling är Nios II 15.0 Software Build Tools for Eclipse.

### 6.1 Inställningar av BSP



Figur 6.1. Inställningar av BSP i Eclipse.

Figur 6.1 visar det inställningar vilka ha gjorts i verktyget för BSP.

## 7. Verifiering

Analys av tidsåtgången för att rita ut en punkt i grafen med rätt uppdateringsfrekvens gjordes initialt med den i hårdvaran inbyggda timern. Dock bidrog användandet av printf och som alternativ även alt\_printf för utskrift av resultat att värdena avvek från normalt då funktionerna slöade ner programmet rejält. Därefter tillämpades timern och strategiskt placerade variabler vars resultat lästes av i debuggern. Timern resetas var gång en utskrift sker.

Utskrift av pixel i grafen vid 1 Hz uppdateringsfrekvens görs var 50090662 cykel efter mätning med den interna timern. Det motsvarar  $50090662 \times 20 \text{ ns} = 1,002 \text{ sekunder}$  då en cykel vid 50 MHz tar 20 ns vilket visas i figur 7.2.

(0) key_in	1
(0) key_counter	2
sample_rate	0x00ffffe4
(0) sample_rate[0]	0
(0) sample_rate[1]	10000000
(0) sample_rate[2]	25000000
(0) sample_rate[3]	50000000
(0) sample_rate[4]	500000000

Name	Value
X+Y =? "\$sp"	0x00ffffcc
X+Y =? "\$pc"	0x00800830
X+Y =? "TIMER_DATA_STOPP"	50090662
X+Y =? "offset"	50
+ Add new expression	

Figur 7.1. Debugg av uppdateringsfrekvenser.

Figur 7.2. Uppdateringsfrekvens vid 1 Hz i cykler.

Name	Value
(0) sample_rate	25000000
(0) adc_data	1708
(0) VGA_adc_data	'j'
106 'j'	

Figur 7.3. Kontroll av data ut till skärmen vid 2 Hz uppdateringsfrekvens och anpassning av data, notera att debuggern registrerar alt\_u8 som char, 106 motsvarar 'j' i ASCII-tabellen.

## 7.1 Testprotokoll efter verifiering

Tabell 7.1. Testprotokoll efter verifiering.

Testfall	Beskrivning	Resultat	OK
1	Skärminnet rensas vid uppstart	Svart skrivs till grafikminnet	OK
2	Presentationsbilden för systemet skrivs ut på VGA-skärmen	Svart och spår av grönt i grafikminnet	OK
3	Vid tryck på knapp 1 ritas bakgrunden till grafen upp och loggning med utskrift i realtid (eller så fort som systemet hinner med att skriva ut på skärmen) börjar. LEDs är släckta.	Grafen ritas ut med en uppdateringshastighet som ej är definierad. Kontroll i debuggern ger 625498 cykler * 20 ns = 0,125 sekunder mellan var utskrift.	OK
4	Vid ytterligare ett tryck på knapp 1 byts uppdateringsfrekvensen för loggningen till 5 Hz. LED1 lyser.	Grafen ritas ut med en uppdateringshastighet om 5 Hz vilket motsvarar 0,2 s. LED1 lyser. Kontroll i debuggern ger 10011987 cykler * 20 ns = 0,200 sekunder.	OK
5	Vid ytterligare ett tryck på knapp 1 byts uppdateringsfrekvensen för loggningen till 2 Hz. LED 2 lyser	Grafen ritas ut med en uppdateringshastighet om 2 Hz vilket motsvarar 0,5 s. LED2 lyser. Kontroll i debuggern ger 25105924 cykler * 20 ns = 0,502 sekunder.	OK
6	Vid ytterligare ett tryck på knapp 1 byts uppdateringsfrekvensen för loggningen till 1 Hz. LED1 och 2 lyser.	Grafen ritas ut med en uppdateringshastighet om 1 Hz vilket motsvarar 1 s. LED1 och 2 lyser. Kontroll i debuggern ger 50090662 cykler * 20 ns = 1,002 sekunder.	OK
7	Vid ytterligare ett tryck på knapp 1 byts uppdateringsfrekvensen för loggningen till 0.1 Hz. LED4 lyser.	Grafen ritas ut med en uppdateringshastighet om 0,1 Hz vilket motsvarar 10 s. LED3 lyser. Kontroll i debuggern ger 500041060 cykler * 20 ns = 10,00 s.	OK
8	Vid ändring av ljusstyrka mot fotoresistorn ändras grafens presentation på skärmen på sådant vis att det uppfattas relevant för betraktaren.	Kontroll i debuggern ger värden vilka verkar överensstämma med den uppfattade verkliga förändringen i ljusintensitet.	OK

## 8. Analys

### 8.1 Varningar

Description	Resource	Path	Location	Type
Warnings (9 items)				
assignment makes integer from pointer without a cast	sierra_task.c	/C_engjobb_final_bsp/drivers/src	line 64	C/C++ Problem
assignment makes integer from pointer without a cast	sierra_task.c	/C_engjobb_final_bsp/drivers/src	line 65	C/C++ Problem
assignment makes integer from pointer without a cast	sierra_task.c	/C_engjobb_final_bsp/drivers/src	line 70	C/C++ Problem
dereferencing type-punned pointer will break strict-aliasing rules [-Wstrict-aliasing]	sierra_sem.c	/C_engjobb_final_bsp/drivers/src	line 117	C/C++ Problem
dereferencing type-punned pointer will break strict-aliasing rules [-Wstrict-aliasing]	sierra_task.c	/C_engjobb_final_bsp/drivers/src	line 115	C/C++ Problem
implicit declaration of function 'alt_irq_register' [-Wimplicit-function-declaration]	sierra.c	/C_engjobb_final_bsp/drivers/src	line 48	C/C++ Problem
passing argument 3 of 'alt_irq_register' from incompatible pointer type	altera_modular_adc.c	/C_engjobb_final_bsp/drivers/src	line 131	C/C++ Problem
unused variable 'info_t' [-Wunused-variable]	sierra_sem.c	/C_engjobb_final_bsp/drivers/src	line 103	C/C++ Problem
unused variable 'retval' [-Wunused-variable]	sierra.c	/C_engjobb_final_bsp/drivers/src	line 247	C/C++ Problem
Infos (1 item)				
expected 'alt_isr_func' but argument is of type 'void (*)(void *)'	alt_legacy_irq.h	/C_engjobb_final_bsp/HAL/inc/priv	line 61	C/C++ Problem

Figur 8.1. Kvarvarande varningar.

Kvarvarande varningar härrör från sierra-rt-kärnan och ADC:n. RT-kärnan tillämpas inte i konstruktionen, även om försök gjordes för att få igång IRQ-hantering med sierran för kontroll av knapptryckningar. Den ensamma varningen från ADC:n berör även den IRQ-hanteringen. Ingen av de kvarvarande varningarna påverkar konstruktionens funktion.

### 8.2 Minimering av kod

#### 8.2.1 Ej optimerad kod

```
Mattias@Mattias-mini /cygdrive/d/teis/c/HW_C_engjobb_final/software/C_engjobb_final
$ nios2-elf-size C_engjobb_final.elf
text    data    bss     dec      hex filename
74644   6968   1484   83096   14498 C_engjobb_final.elf
```

Figur 8.2. .elf-filens storlek före optimering.

Kontroll av storleken hos .elf-filen kan göras m.h.a NIOS II command shell och kommandot "nios2-elf-size <mitt projekt.elf>".

Den ej optimerade koden upptar 83096 bytes vilket visas i figur 8.2.

#### 8.2.2 Optimerad kod

För full optimering tillämpas följande inställningar i BSP-editorn [11]:

- Ingen C++ support
- Inget clean\_exit
- Inget exit
- Enable lightweight device driver API
- Optimization level satt till "Size"

- stdio.h utelämnas, istället används #include "sys/alt\_stdio.h" och alteras alt\_-varianter.

```
Mattias@Mattias-mini /cygdrive/d/teis/c/HW_C_engjobb_final/software/C_engjobb_final
$ nios2-elf-size C_engjobb_final.elf
   text    data    bss     dec     hex filename
 17468    320    1412    19200   4b00 C_engjobb_final.elf
```

Figur 8.3. .elf-filens storlek efter optimering.

Den optimerade koden upptar 19200 bytes vilket visas i figur 8.3.

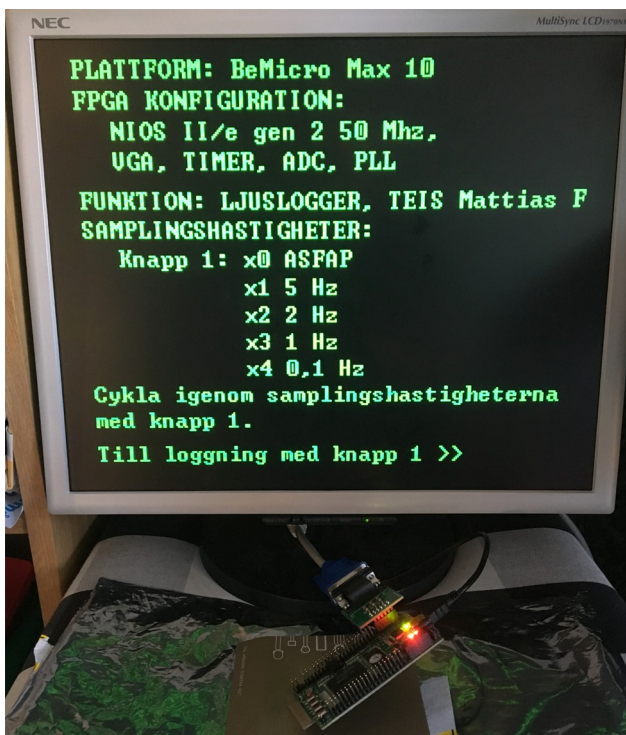
## 8.3 Buggar

De finns inget maxvärde för ljusstyrkan i grafen vilket kan leda till att då fotoresistorn utsätts för mycket starkt ljus sker utskrift utanför diagrammets område, dock ej utanför skärmens synliga område då grafikfunktionernas inbyggda skydd förhindrar det.

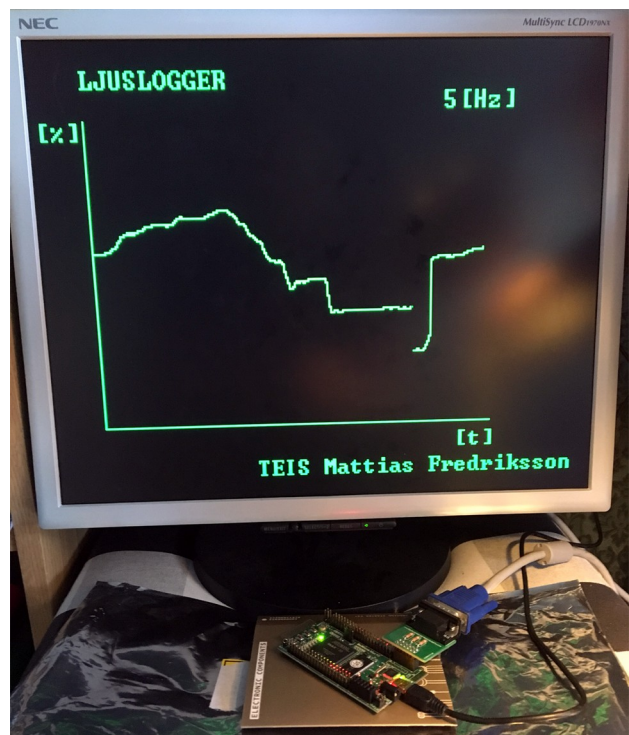
Hålls knappen inne för att cykla mellan de olika samplingshastigheterna väntar enbart systemet på att knappen ska släppas upp igen. Detta beror av IRQ-rutinen vilken kunde skapa ett interrupt vid knapptryck samt ett vid uppsläppt knapp och eventuellt kringgått problemet ej fungerade och ersattes i konstruktionen med pollande av knappar istället. Ytterligare felsökning av IRQ:n omfattades ej i tidsplanen.

## 9. Validering

### 9.1 Bilder från Validering



Figur 9.1. Validering av presentationsbild.



Figur 9.2. Validering av graf.



## 9.2 Ifyllt testprotokoll efter validering

Tabell 9.1. Testprotokoll efter validering.

Testfall	Beskrivning	Resultat	OK
1	Skärminnet rensas vid uppstart	Svart skrivs till grafikminnet	OK
2	Presentationsbilden för systemet skrivs ut på VGA-skärmen	Svart och spår av grönt i grafikminnet	OK
3	Vid tryck på knapp 1 ritas bakgrunden till grafen upp och loggning med utskrift i realtid visas. LED1 är släckt.	Grafen ritas ut med en uppdateringshastighet som ej kan uppmätas manuellt men är under 0,17 s. Kontroll i debuggern ger 625498 cykler * 20 ns = 0,125 sekunder.	OK
4	Vid ytterligare ett tryck på knapp 1 byts uppdateringsfrekvensen för loggningen till 5 Hz. LED1 lyser.	Grafen ritas ut med en uppdateringshastighet som har uppmätts manuellt* till 0,17 sekunder. 5 Hz motsvarar 0,2 s. LED1 lyser. Kontroll i debuggern ger 10011987 cykler * 20 ns = 0,200 sekunder.	OK
5	Vid ytterligare ett tryck på knapp 1 byts uppdateringsfrekvensen för loggningen till 2 Hz. LED 2 lyser	Grafen ritas ut med en uppdateringshastighet som har uppmätts manuellt* till 0,47 sekunder. 2 Hz motsvarar 0,5 s. LED2 lyser. Kontroll i debuggern ger 25105924 cykler * 20 ns = 0,502 sekunder.	OK
6	Vid ytterligare ett tryck på knapp 1 byts uppdateringsfrekvensen för loggningen till 1 Hz. LED1 och 2 lyser.	Grafen ritas ut med en uppdateringshastighet som har uppmätts manuellt* till 0,96 sekunder. 1 Hz motsvarar 1 s. LED1 och 2 lyser. Kontroll i debuggern ger 50090662 cykler * 20 ns = 1,002 sekunder.	OK
7	Vid ytterligare ett tryck på knapp 1 byts uppdateringsfrekvensen för loggningen till 0.1 Hz. LED4 lyser.	Grafen ritas ut med en uppdateringshastighet som har uppmätts manuellt* till 9,98 sekunder. 0,1 Hz motsvarar 10 s. LED3 lyser. Kontroll i debuggern ger 500041060 cykler * 20 ns = 10,00 s.	OK
8	Vid ändring av ljusstyrka mot fotoresistorn ändras grafens presentation på skärmen på sådant vis att det uppfattas relevant för betraktaren.	Grafen varierar med en ihopsittande linje mellan 20-85 % av diagrammets yta i y-led beroende av exponeringsgraden för ljus. Dock överstigs diagrammet då en lysande LED från en mobil läggs rätt över fotoresistorn.	OK (med anmärkning)

\* Manuell mätning med digitalt tidtagarur, avvikelserna för mätfel, dels av den digitala, men främst av mänskliga faktorn torde överstiga 5 %.

## 9.3 Film

Under länken ned återfinns en film publicerad på youtube vilken beskriver systemets funktion:

<https://youtu.be/B2A96lmOR1I>



## 10. Sammanfattning och förbättringar

Åtgärdande av buggarna under avsnitt 8.3. hade polerat något men i stort inte påverkat konstruktionen vilken i nuläget fungerar mycket bra.

Skalan och utseendet på grafens bakgrund skulle kunna justeras något för ett mer estetiskt tilltalande gränssnitt.

Det är synd att det inte finns en SD-kortplats på BeMicro-kortet, då hade ljusdata kunnat sparas även där för att få lite volymer på datan.

Felet vilket beskrivs under 8.1 där IRQ:n inte ville fungera upptog en del tid och med stor besvikelse fick den tillämpningen överges.

Efter att ha arbetat något med VHDL kan nu C kännas något begränsande just för att det saknar parallelliteten. I VHDL är det lätt att bara lägga till ytterligare en process om det behöver göras flera saker samtidigt, exempelvis polla knappar och skriva ut en graf parallellt. Dock fortskrider inte programmeringen lika snabbt och en tillämpning som ljusloggern hade tagit betydligt längre tid att utföra i ren VHDL.

## 11. Versionshistorik

Tabell 11.1. Versionshistorik.

Version	Datum	Ansvarig	Beskrivning
A4	2016-01-03	MF	Sammanställt rapport.
A3	2015-12-27	MF	Lagt till optimering av kod
A2	2015-12-23	MF	Lagt till delsystem
A1	2015-12-21	MF	Upprättat konstruktionsrapport för ljuslogger.
A0	2015-12-10	MF	Upprättat kravspecifikation, tidsplan och förstudie för ljusdatalogger i enlighet med C uppgift engineeringjob.

## Referenser

[1] 2015-12-21

[http://www.alterawiki.com/wiki/BeMicro\\_Max\\_10](http://www.alterawiki.com/wiki/BeMicro_Max_10)

[2] 2015-12-21

[http://www.alterawiki.com/uploads/e/ec/BeMicro\\_Max\\_10-Schematic\\_A4-20141008.pdf](http://www.alterawiki.com/uploads/e/ec/BeMicro_Max_10-Schematic_A4-20141008.pdf)

[3] 2015-12-22

<http://www.cypax.dk/pdf/PGM5526x.pdf>

[4] 2015-12-23

[http://www.alterawiki.com/uploads/3/34/BeMicroM10\\_Getting\\_Started\\_14\\_0\\_update.pdf](http://www.alterawiki.com/uploads/3/34/BeMicroM10_Getting_Started_14_0_update.pdf)

[5] 2015-12-23

<https://www.altera.com/content/dam/altera->

[www/global/en\\_US/pdfs/literature/hb/nios2/n2cpu\\_nii5v1.pdf](http://www.global/en_US/pdfs/literature/hb/nios2/n2cpu_nii5v1.pdf)

[6] 2015-12-23

[https://www.altera.com/en\\_US/pdfs/literature/hb/max-10/ug\\_m10\\_adc.pdf](https://www.altera.com/en_US/pdfs/literature/hb/max-10/ug_m10_adc.pdf)

[7] 2015-12-23

<http://www.farnell.com/datasheets/1861845.pdf>

[8] Datasheet\_IP\_VGA.pdf (bifogat med projektet)

[9] TIMER\_SW\_USER\_MANUAL.pdf (bifogat med projektet)

[10] VGA\_sw\_user\_manual.pdf (bifogat med projektet)

[11] 2015-12-27

[http://agstuworld.se/TEIS/2015TEIS/2015\\_kurs\\_C/V7\\_NIOSII\\_tool\\_Kod\\_optimering/Theory\\_7/presentation.html](http://agstuworld.se/TEIS/2015TEIS/2015_kurs_C/V7_NIOSII_tool_Kod_optimering/Theory_7/presentation.html)

## Bilagor

Datasheet\_IP\_VGA.pdf (i mappen \konstruktionsdokument\ljusdatalogger\ip\IP\_VGA\doc)

Kort\_beskrivning\_IP\_VGA.pdf (i mappen \konstruktionsdokument\ljusdatalogger\ip\IP\_VGA\doc)

VGA\_sw\_user\_manual.pdf (i mappen \konstruktionsdokument\ljusdatalogger\ip\IP\_VGA\doc)

TIMER\_SW\_USER\_MANUAL.pdf (i mappen  
\konstruktionsdokument\ljusdatalogger\ip\TIMER\doc)