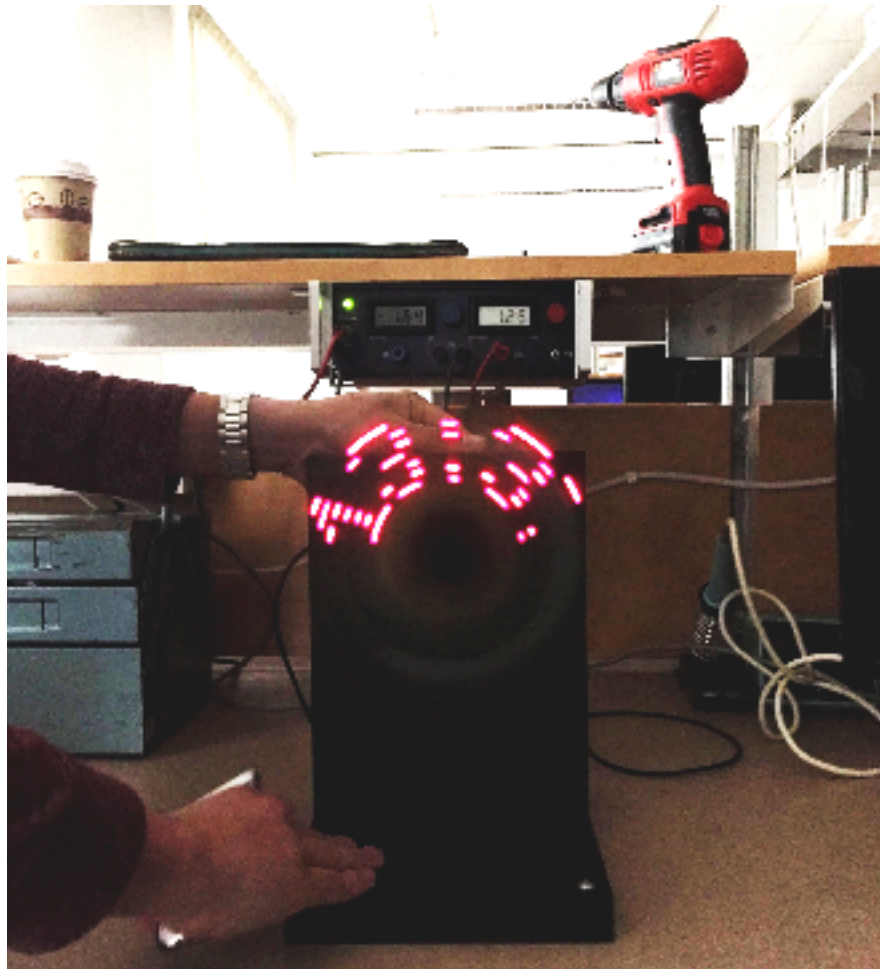


# Projektrapport - Roterande digital LED-klocka



**Kurs:** EITF 11 - Digitala projekt

**Elever:** Axel Truedsson, Erik Lundström, Sara Åkerblom

**Handledare:** Bertil Lindvall

**Datum:** 2017-05-17

**Lunds Tekniska Högskola**

# Sammanfattning

Denna rapport beskriver kraven, arbetsprocessen och resultatet av implementering av en roterande LED-klocka. Avslutningsvis förs en diskussion runt arbetet och möjliga framtida förbättringar.

Kraven på klockan är huvudsakligen följande: visa tiden vid rotation, ställa in starttid enligt HH:MM samt att klockan ska kunna hålla tiden. I projektet har en rad olika hårdvarukomponenter använts som vars egenskaper och funktioner beskrivs i teoridelen.

Klockans knappar gör det möjligt för användaren att ställa in tiden. Genom att trycka på den gula knappen kan användaren skifta mellan att ställa in timme eller minut. De nästföljande två knapparna (efter den gula) gör det möjligt att öka och minska timme/minut. Den sista knappen används för att starta tidsräkningen i processorn.

Arbetsprocessen var inledningsvis väldigt teoretisk med framtagning av kopplingsschema och kravspecifikation. Programmeringen av mjukvaran skedde i Atmel Studio 7.0 och efter succesiv felsökning med JTAG kunde slutligen önskat resultat uppnås.

# Innehållsförteckning

<b>1 Inledning</b>	<b>3</b>
1.1 Syfte	3
<b>2 Kravspecifikation</b>	<b>3</b>
2.1 Funktionella krav	3
2.2 Kvalitativa krav	3
<b>3 Teori</b>	<b>3</b>
3.1 Hårdvara	4
3.1.1 Processor	4
3.1.2 Mönsterplatta	4
3.1.3 LED-lampa	4
3.1.4 Knapp (strömbrytare)	4
3.1.5 Kristall	4
3.1.6 JTAG	4
3.1.7 Light Sensor AMS 302	4
3.1.8 Batteri	4
3.1.9 Resistorer	4
3.2 Mjukvara	5
<b>4 Användning &amp; instruktionsmanual</b>	<b>5</b>
4.1 Starta klockan	5
4.2 Ställ in tid	5
4.3 Stäng av klockan	5
<b>5 Arbetsprocess</b>	<b>5</b>
5.1 Planering	6
5.2 Konstruktion av hårdvara	6
5.3 Programmering av mjukvara	6
<b>6 Resultat</b>	<b>7</b>
<b>7 Diskussion</b>	<b>7</b>
<b>9 Källor</b>	<b>7</b>
<b>Appendix</b>	<b>7</b>

# 1 Inledning

## 1.1 Syfte

Syftet med denna rapport är att beskriva och redogöra för projektet "roterande digital LED-klocka". Denna projektrapport utgör en del av dokumentationsarbetet i kursen EITF11 Digitala Projekt vid Lund Tekniska Högskola. Kursens övergripande mål är att ge studenten en ökad förståelse och ett begrepp om hur konstruktionsarbete går till samt prestera en fungerande prototyp med tillhörande dokumentation. Med en fungerande prototyp syftar rapporten till att lyckas utveckla en prototyp som når eller försöker nå de funktionella krav och kvalitativa krav som ställs på prototypen i den här rapporten.

Den här rapporten behandlar projektets olika delar i form av en kravspecifikation, bakomliggande teori, en användarmanual, en beskrivning av arbetsprocessen och slutligen resultat och en diskussion kring resultatet.

## 2 Kravspecifikation

Här följer de funktionella och kvalitativa krav som ställs på produkten. Dessa har använts som ledstång och utgångspunkt vid utvecklandet av produkten.

### 2.1 Funktionella krav

1. Klockan ska kunna visa tiden vid rotation med hjälp av LED-lamporna.
2. Användaren ska kunna justera och ställa in tiden.
3. Klockan ska uppdateras i realtid.
4. Tiden ska visas i digital form enligt HH:MM.
5. Siffrorna ska uppdateras varje minut.
6. Prototypen ska ha fyra knappar som ska kunna användas vid inställning av tiden.
7. Klockan ska kunna pausas för att ställa om tiden.

### 2.2 Kvalitativa krav

1. Klockarmen ska rotera med 1500 hz.
2. Siffrorna ska tydligt kunna urskiljas.
3. Tidsfördröjningen ska inte påverka tidsinställningen med mer än två sekunder.

## 3 Teori

Här följer en sammanfattad beskrivning av den teori som använts under projektets gång. Avsnittet är uppdelat i två delar. Första delen behandlar de hårdvarukomponenter som används och den andra delen behandlar mjukvaran.

### 3.1 Hårdvara

Nedan beskrivs de hårdvarukomponenter som utgör prototypen.

#### 3.1.1 Processor

För projektet används en 8 bits-processor av modellen Atmega 16.

#### 3.1.2 Mönsterplatta

Detta är plattan som utgör klockarmen. På denna monterades komponenterna.

#### 3.1.3 LED-lampa

Prototypen har åtta LED-lampor som lyser rött då de kopplas till ström.

#### 3.1.4 Knapp (strömbrytare)

Prototypen består av fyra knappar som gör det möjligt att välja mellan timmar eller minuter, öka eller minska tiden samt pausa eller starta tidsräkningen. Utöver detta finns även en strömbrytare som är kopplad till motorn.

#### 3.1.5 Kristall

En 16 MHz kristall har använts för att skapa en elektrisk signal med exakt frekvens. Denna frekvens har använts för att få en mer exakt tid än processorns egen tidsräknare. Detta för att få en så stabil varvtid som möjligt.

#### 3.1.6 JTAG

En AVR JTAG används under projektet för felsökning.

#### 3.1.7 Light Sensor AMS 302

En ljussensor som gör det möjligt att registrera varje gång klockarmen passerar. På så sätt är det möjligt att kartlägga när LED-lamporna ska lysa respektive inte lysa.

### 3.1.8 Batteri

För att klockan ska kunna rotera utan att trassel med lösa sladdar uppstår behövs det ett batteri som försörjer alla hårdvarukomponenter på mönsterplattan med ström. Batteriet som används har spänning 6 volt.

### 3.1.9 Resistorer

På ett antal ställen på mönsterplattan har resistorer av olika storlek monterats för att komponenterna ska förses med rätt spänning.

## 3.2 Mjukvara

Källkoden för prototypens mjukvara har utvecklats i språket C i programmet AtmelStudio 7. En kopia av källkoden återfinns i Appendix B.

Programmets huvudsakliga kod utgörs av en main-metod som möjliggör att klockan räknar tiden från en starttid som användaren ställer in. Koden innehåller även de attribut och metoder som gör det möjligt att lamporna tänds och släcks i ett mönster så att tiden visas vid rotation. Attribut som motsvarar lampornas mönster, för att kunna avbilda siffror, lagras i vektorer. Varje siffra för 0-9 inklusive ett kolontecken lagras i en 11x5 matris. Metoder för kalibrering och inställning av tiden är kopplade till de insignaler som kommer från knapparna.

## 4 Användning och instruktionsmanual

Nedan följer ett antal manualer för de funktioner som är möjliga hos prototypen.

### 4.1 Slå på klockan

För att starta klockan kopplas ström till klockan. Klockan är nu i pausläge och tiden kan ställas in.

### 4.2 Ställ in tid

Den gula knappen används för att ställa in timme eller minut. Skifte mellan dessa två sker genom en knapptryckning. Knappen närmast den gula knappen används för att öka tiden. En knapptryckning ökar tiden en minut eller en timme. Knappen näst längst ifrån den gula knappen används för att minska tiden. En knapptryckning minskar tiden en minut eller en timme.

### 4.3 Starta klockan

Tryck på paus/start-knappen, som startar prototypens tidräkning. Tryck sedan på strömbrytaren som sitter på motorns baksida. Klockan ska nu rotera och visa tiden som ställts in. För varje minut som går ska tiden uppdateras medan klockarmen roterar.

## 4.4 Stäng av klockan

Stäng av rotationen genom att slå av strömbrytaren på motorns baksida. Bryt sedan strömmen till klockan.

# 5 Arbetsprocess

Här följer en beskrivning av hur arbetet har fortskridit och dess tre huvudsakliga faser.

## 5.1 Planering

Projekt inleddes under i januari 2017 med planering och förberedelse av projektet. Det initiala planeringsarbetet bestod av utforska vilka möjligheter som finns och fatta beslut om vad projektet ska prestera. Efter diskussion beslutades det att projektet ska prestera en roterande LED-klocka. Sedan specificerades de krav som projektet ställer på prototypen. Det fortsatta planeringsarbetet, med interna och externa deadlines, har sedan haft sin utgångspunkt från dessa. Kraven gjorde det även möjligt i ett tidigt skede kunna identifiera vilka hårdvarukomponenter som kommer krävas för prototypen, samt vilka problem som skulle kunna uppstå under projektets gång.

Som en del av planeringsarbetet producerades ett kopplingsschema som en övergripande ritning av hur prototypen ska vara uppbyggd. Först när projekttid och kopplingsschema godkänkts av handledare inleddes det konkreta arbetet med konstruktion och programmering av prototypen.

## 5.2 Konstruktion av hårdvara

Med utgångspunkt från det godkända kopplingsschemat inleddes konstruerandet av prototypen. Komponenterna löddes fast och kopplades samman i mönsterplattan. De olika kopplingarna testades sedan med hjälp av JTAG. De fel som upptäcktes åtgärdades och behövdes i några fall lödas om.

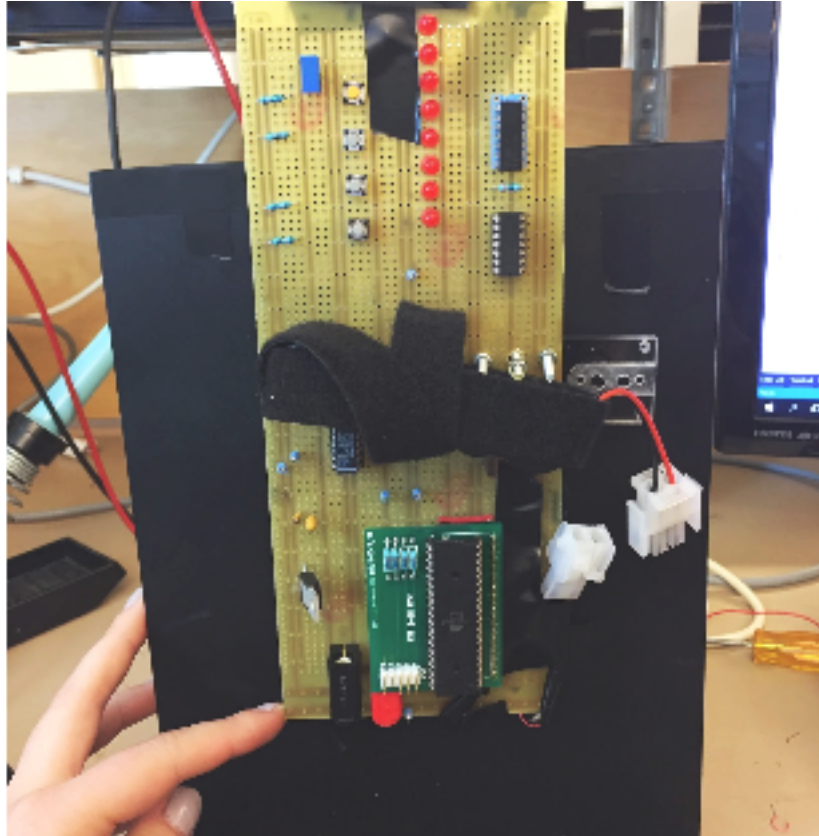
## 5.3 Programmering av mjukvara

När hårdvaran var testad och fungerade enligt önskemål började källkoden utvecklas. Källkoden började utvecklas småskaligt, till exempel genom att deklarerat attribut och parametrar som skulle kunna tänkas behövas. För att koden skulle vara översiktlig och lättarbetad användes beskrivande namn för deklARATIONERNA och kommentarer i koden.

Källkoden testades löpande med hjälp av JTAG och fel modifierades successivt för att uppnå önskat resultat.

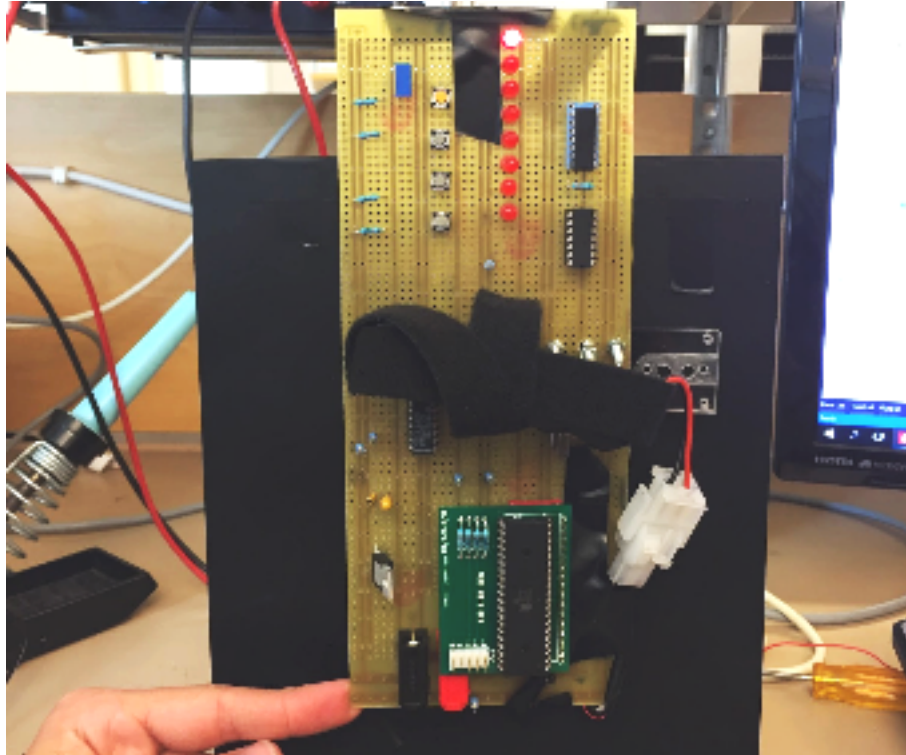
## 6 Resultat

Nedan presenteras resultat av prototypen i form av en serie bilder och beskrivande bildtext.  
**Se hemsida för video!**

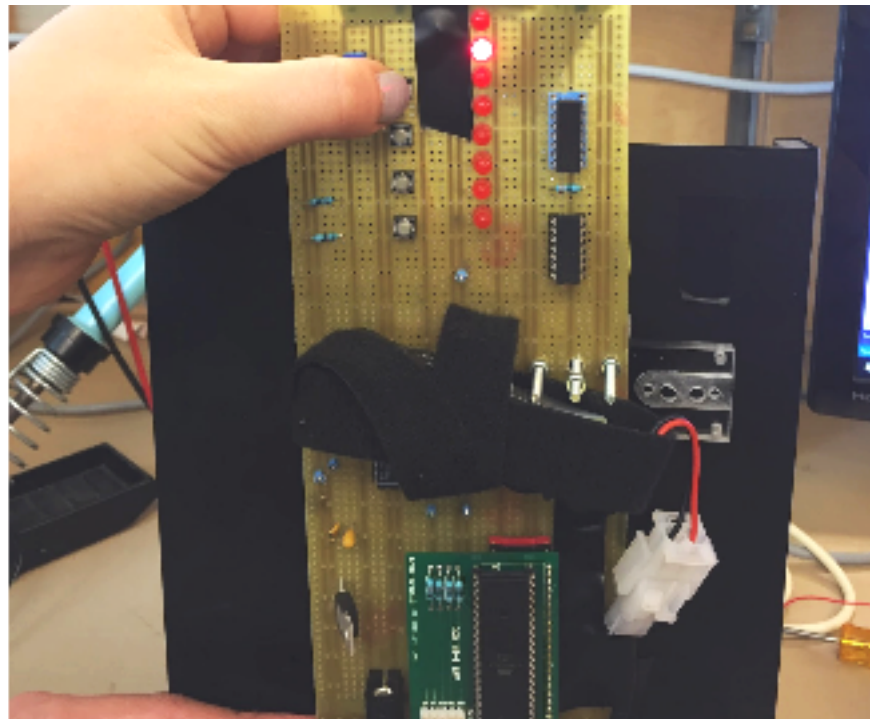


*Figur 1: Prototypens ursprungsläge med batteriet urkopplad. Överst i bild syns 8 LED som visar tiden och till vänster om LED sitter 4 knappar som används för att konfigurera önskad starttid.*

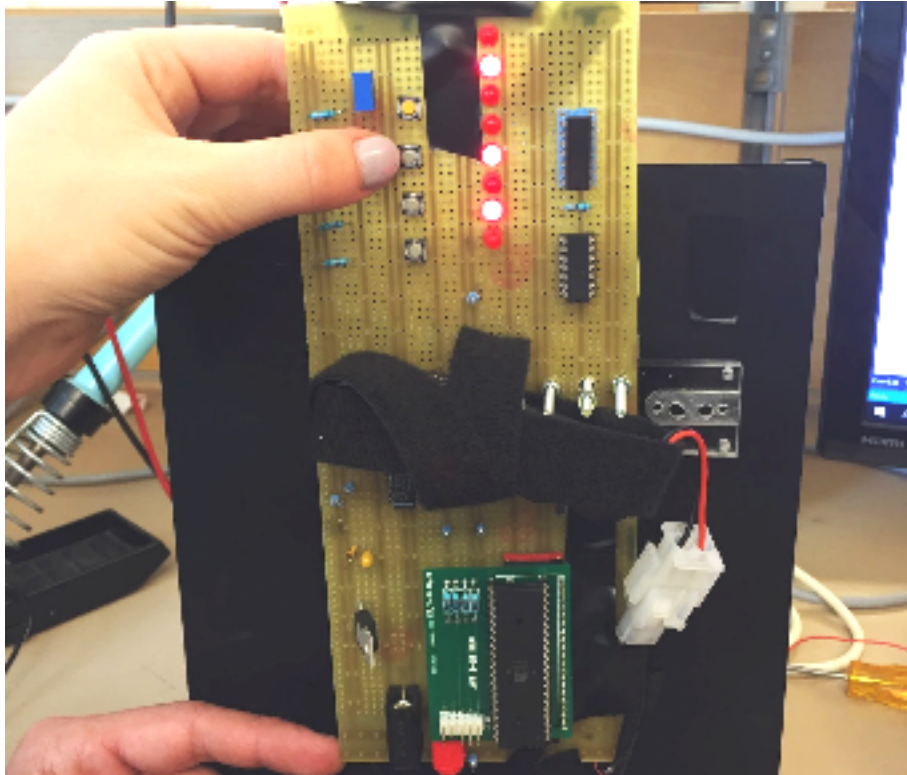




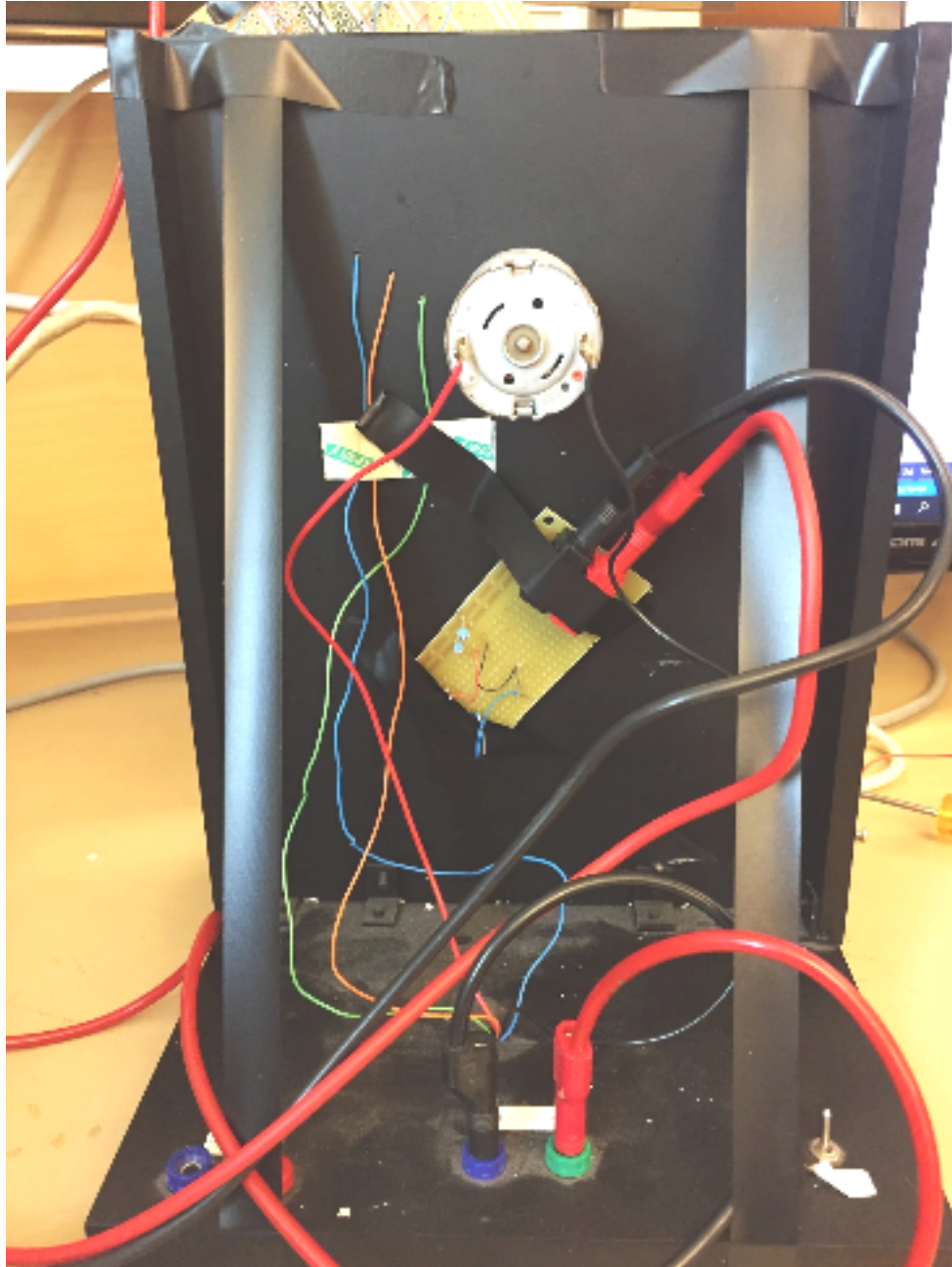
Figur 2: Prototypen med batteriet inkopplat. I inkopplat ursprungsläge lyser den översta lampan, vilket indikerar att prototypen väntar på att användare ska ställa in timme.



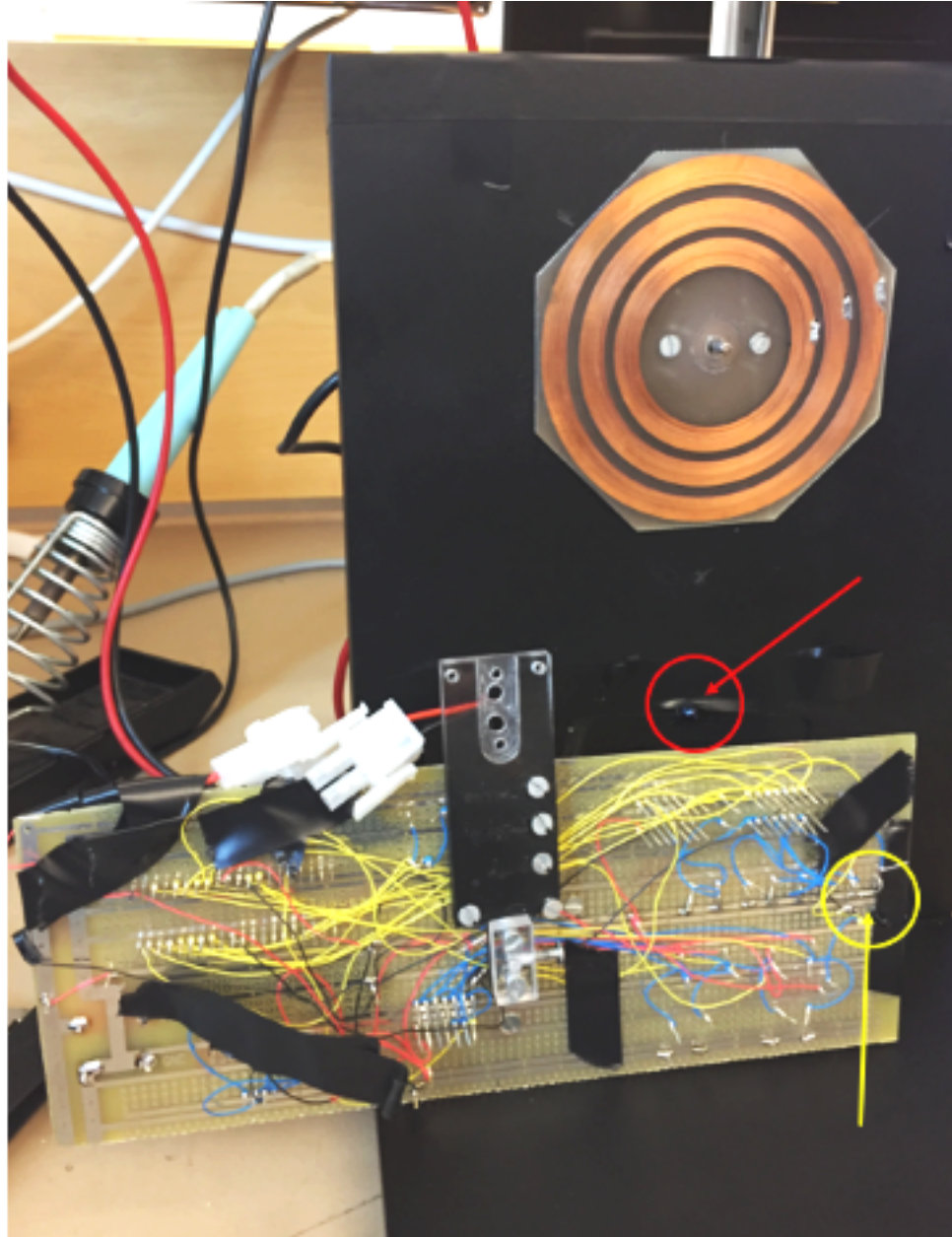
Figur 3: Genom att trycka på gula knappen, börjar andra lampan lysa och användaren tillåts ställa in minuter.



*Figur 4: Genom att klicka på den andra knappen (uppifrån) kan användaren öka timme/minut. Vilken siffra användaren är på beskrivs binärt av de 6 undre lamporna (läses nerifrån). Genom att tryck på den tredje knappen (uppifrån) kan användaren minska timme/minut.*



*Figur 5: Prototypens baksida. De nedre kablarna driver motorn, de övre ljussensorn.*



*Figur 6: Prototypen med klockarmen avmonterad. I den röda cirkeln sitter LED som sänder ut IR-ljus. I den gula cirkeln sitter en ljussensor som registrerar IR-ljus.*



## 7 Diskussion

Samtliga studenter i gruppen anser att arbetet har utvecklat förmågan att planera och driva ett projekt framåt. Då gruppen saknade stora delar av nödvändig kunskap inför projektet, ägnades en stor del av projektets inledande fas åt faktainsamling. Framförallt inom programmering i språket C.

Under kursen har gruppen erhållit stöd och feedback från handledare, men gruppen har också haft mycket användning av de databladsmaterial som finns tillgängliga på kurshemsidan.

## 9 Källor

[1] Datablad till processor Atmega 16

<<http://www.eit.lth.se/fileadmin/eit/courses/edi021/datablad/Processors/ATmega16.pdf>>

Hämtad: 2017-05-09

[2] Datablad till Voltage Regulator MCP 1827

<<http://www.eit.lth.se/fileadmin/eit/courses/edi021/datablad/Analog/voltage/mcp1827.pdf>>

Hämtad: 2017-05-09

[3] Datablad till Inverterare Hex Smith Trigger Inverter

<<http://www.eit.lth.se/fileadmin/eit/courses/edi021/datablad/Logik/74HC/74HC14.pdf>>

Hämtad: 2017-05-09

[4] Datablad till QUADRUPLE 2-INPUT POSITIVE-OR GATES

<<http://www.eit.lth.se/fileadmin/eit/courses/edi021/datablad/Logik/74HC/74HC32.pdf>>

Hämtad: 2017-05-09

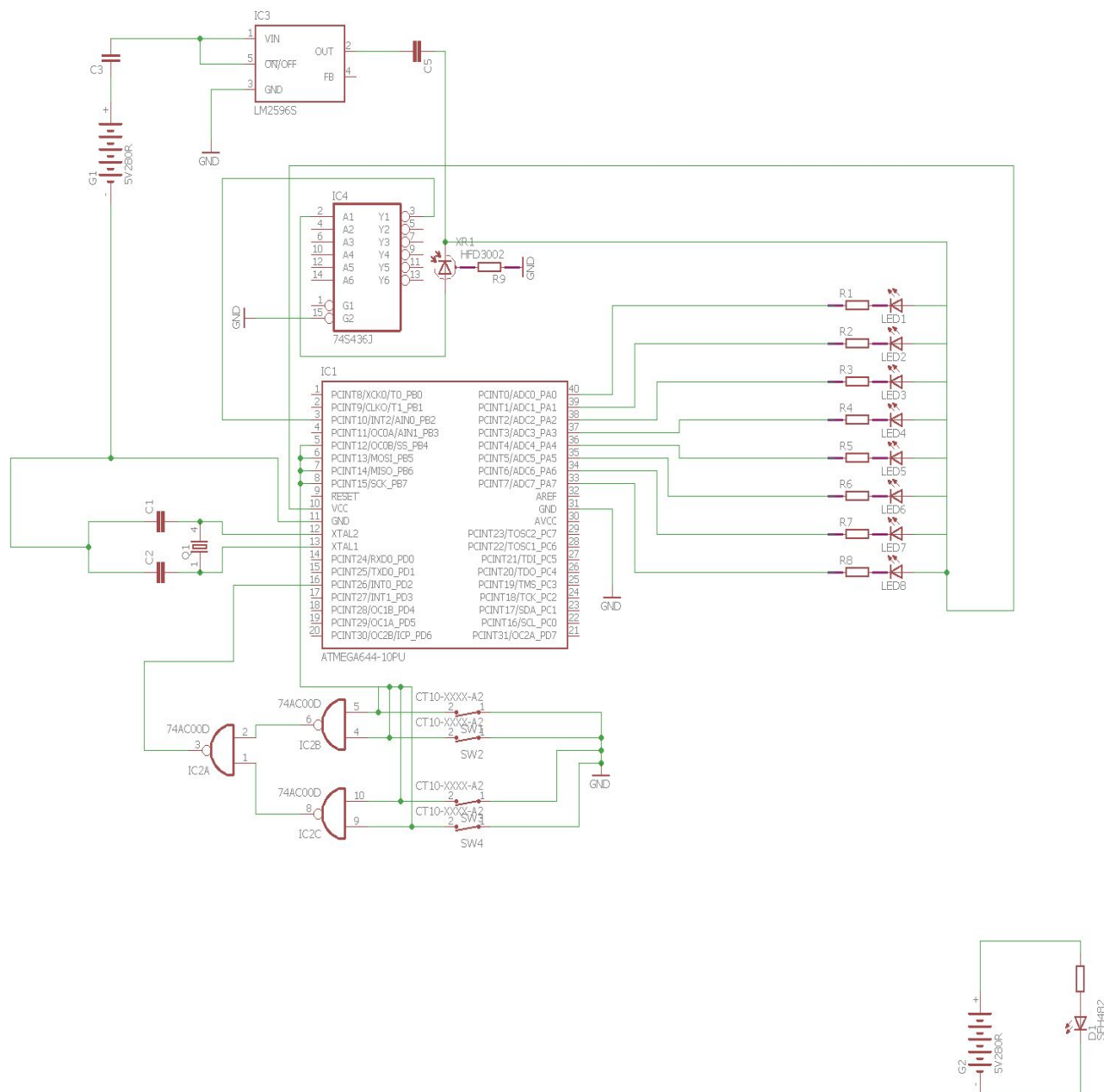
[2] Allmän information angående programmering i C

<[www.AVRFreaks.net](http://www.AVRFreaks.net)>

Hämtad: 2017-05-09

# Appendix

## Appendix A - Kopplingschema



Figur 1: Till vänster är konstruktionen som är monterad på klockarmen. I höger nedre hörn är kopplingschemat av IR-LED som svarar mot ljussensorn.

## Appendix B - Källkod

```
/*
 * LEDNEW2.c
 *
 * Created: 2017-04-24 13:58:57
 * Author : kem14sak
 */

// IMPORTS(?)
#include <avr/io.h>
#include <avr/interrupt.h>
#define F_CPU 16000000UL //16MHZ
#include<util/delay.h>

//FUNCTIONS
void settime();
void timeRunner();
void counter();
void nbrvector();
void showclock();
void displaytime();

//VARIABLES
unsigned int LAMPA = 0;
unsigned int HOUR = 0;
unsigned int MIN = 0;
unsigned int SEC = 0;
unsigned int HOURORMIN = 1; //0=timmar, 1=minuter
unsigned int PRESSEDBUTTON;
unsigned int blsPressed = 0; // sant falskt
unsigned int pauseTime = 0;
unsigned int count = 0;
unsigned int count1 = 0;
unsigned int count2 = 0;
unsigned int count3 = 0;
unsigned int place = 0;

unsigned int playpause = 0; //0 = PAUS
unsigned int hourfirst = 0;
unsigned int hoursecond = 0;
unsigned int minfirst = 0;
unsigned int minsecond = 0;
unsigned int onepos = 0;
unsigned int roundtime = 0;
static int numbers[11][5] = {
    {193, 190, 190, 190, 193}, //0
```

```

        {238, 222, 128, 254, 254},           //1
        {222, 188, 186, 182, 206},         //2
        {221, 190, 182, 182, 201},         //3
        {135, 247, 247, 247, 128},         //4
        {133, 182, 182, 182, 184},         //5
        {193, 182, 182, 182, 217},         //6
        {191, 191, 188, 179, 143},         //7
        {201, 182, 182, 182, 201},         //8
        {207, 183, 183, 183, 128},         //9
        {255, 255, 204, 255, 255}         //dots
};

// INTERRUPTS FROM BUTTONS
ISR(INT0_vect){
    blsPressed = 1;
    PRESSEDBUTTON = PINB;
    _delay_ms(20);
}

//INTERRUPTS FROM LIGHT SENSOR
ISR(INT2_vect){
    roundtime = count2;
    onepos = roundtime/60;
    count2=0;
    count3=0;
    GIFR = 1<<INTF2;
}

/*INTERRUPTS FOR TIME*/
ISR(TIMER1_OVF_vect){
    TCNT1 = 0xffff - 5;
    if(playpause ==1){
        count = 1;
        //counter();
    }
}

//MAIN
int main(void)
{
    SEC = 0;
    GICR = 1<<INT0|1<<INT2; // ENABLE INTO & INT2
    MCUCR = 1<<ISC01 |1<<ISC00; //RISING EDGE ON INTO
    MCUCSR = 1<<ISC2; //RISING EDGE ON INT2
    GIFR = 1<<INTF2;

    TCCR1B|= (1 << CS12);//(0 << CS10); //Prescaler 1024, skalar ner klockan --> 15625 Hz
    TIMSK |= (1 << TOIE1);
    TCNT1 =0xffff - 5;

```



```

sei();//Enable global interrupt

DDRA = 0xff; //PORTA output (lampa) 0 = LYSER
PORTA = 0xff; //PORTA '1' (släckt)
DDRB = 0x00; //PORTB input (knapp)
PORTB = 0xff; //PORTB internall pullup (ska reagera då vi trycker
LAMPA = 127;
PORTA = LAMPA;
playpause = 0;

```

```
//DET SOM SKA KÖRAS HELA TIDEN
```

```

while (1){
    if (blsPressed==1){
        PRESSEDBUTTON = PRESSEDBUTTON & 0b11110000;
        if(PRESSEDBUTTON== 0b10000000){//OM PLAY/PAUS-KNAPP BLIR INTRYCKT
            if(playpause == 0){
                playpause=1; //1 = PLAY
                showclock();
            } else {
                playpause = 0; //0 = PAUS
            }
        }
        settime();
    }
    if (count == 1){
        counter();
    }
}
}

```

```
// KNAPPAR
```

```

void settime(){
    blsPressed=0;
    if(playpause ==1){

    } else {
        PRESSEDBUTTON = PRESSEDBUTTON & 0b11110000;
        if(PRESSEDBUTTON == 0b00010000 ){//OM GUL KNAPP BLIR INTRYCKT
            if(HOURORMIN == 0){
                MIN = 191 - LAMPA; //SPARAR MINUTER
                HOURORMIN =1;//TIMME
                LAMPA= 127 - HOUR; //LADDAR TIMMAR
            }
            else {
                HOUR = 127 - LAMPA;//SPARAR TIMMAR
                HOURORMIN = 0;//MINUT
                LAMPA= 191 - MIN; //LADDAR MINUTER
            }
        }
    }
}

```

```

    PRESSEDBUTTON = 0;
  }
  if(PRESSEDBUTTON == 0b00100000){ //VI VILL ÖKA TIDEN
  if(HOURORMIN == 0){
    if(LAMPA == 132){//SKA BLI 0 MIN
      LAMPA = 191;
    }
    else {
      LAMPA = LAMPA -1;
    }
  }
  if(HOURORMIN == 1){
    if(LAMPA == 104){//SKA BLI 0 TIMMAR
      LAMPA = 127;
    }
    else{
      LAMPA = LAMPA -1;
    }
  }
  PRESSEDBUTTON = 0;
  }
  if(PRESSEDBUTTON == 0b01000000){ //VI VILL MINSKA TIDEN
  if(HOURORMIN == 0){
    if(LAMPA == 191){//SKA BLI 59 MIN
      LAMPA = 132;
    }
    else {
      LAMPA = LAMPA +1;
    }
  }
  if(HOURORMIN == 1){
    if(LAMPA == 127){//SKA BLI 23 TIMMAR
      LAMPA = 104;
    }
    else{
      LAMPA = LAMPA +1;
    }
  }
  }
  PRESSEDBUTTON = 0;
  }
  PORTA = LAMPA;
}

```

```

//UPDATE TIME
void timeRunner(){
  SEC ++;
  if (SEC >= 60)
  {

```

```

        MIN ++;
        showclock();
        SEC = 0;
        if (MIN >= 60)
        {
            HOUR ++;
            MIN = 0;
            if (HOUR>=24)
            {
                HOUR = 0;
            }
        }
    }
}

//RÄKNA TID
void counter(){
count1++;
count2++;
if(count1 >= 12496){
    count1=0;
    timeRunner();
}
if ( count2 % onepos == 0){
    count3++;
    displaytime();
}
}

void displaytime(){
    //Första
    if (count3 >=16 && count3 <= 20){
        PORTA = numbers[hourfirst][place];
        place++;
    }
    //ANDRA
    else if (count3 >=22 && count3 <= 26){
        PORTA = numbers[hoursecond][place];
        place++;
    }
    //PRICKAR
    else if (count3 >=28 && count3 <= 32){
        PORTA = numbers[10][place];
        place++;
    }
    //TREDJE
    else if (count3 >=34 && count3 <= 38){
        PORTA = numbers[minfirst][place];
        place++;
    }
}

```

```
}  
//FJÄRDE  
else if (count3 >=40 && count3 <= 44){  
    PORTA = numbers[minsecond][place];  
    place++;  
}  
//RESET  
else {  
    PORTA = 0xff;//Stäng av alla lampor  
    place = 0;  
}  
}
```

```
//STARTA SNURRANDE KLOCKA
```

```
void showclock(){  
    hourfirst = HOUR/10;  
    hoursecond = HOUR%10;  
    minfirst = MIN/10;  
    minsecond = MIN%10;  
}
```