```
//-------------------------------------------------------------
//Remotebud
//-------------------------------------------------------------
#include <avr/signal.h>
#include <avr/io.h>

//Define all variabels
int pause = 0;
int Wftrigg = 1;
unsigned char Lighton = 1;
int Signal;
int Factor = 3;
int Comparevector[100];
int Signalvector1[100];
int Signalvector2[100];
int Signalvector3[100];
int Cvpos = 0;
int Svpos1;
int Svpos2;
int Svpos3;
int Stillingame[4];
int Oneorzero = 1;
int Endlimit = 500;              //Signal ends with 500 ones
long Pausetimer = 0;
int Pausetimes;
int Record=0;
int Recbutton;
int Recposition=1;
int Changebutton;
int Blink;
int Startblink=0;
unsigned char Motorinstuktion[4];
int mipos=0;
int forward=0;
int Motoron=0;
int Lampon=0;
int Curtain=1;


void Count(){                                                  //Count the incomming signal
    if(Signal == Oneorzero){
        if(((Comparevector[Cvpos]) >= Endlimit)   ){       //End of signal either way
            if(Record==1){
                Savesignal();
                Clean();
            }else if(((Cvpos == Svpos1)&&(Stillingame[1]==1))|||((Cvpos == Svpos2)&&(
Stillingame[2]==1))|||((Cvpos == Svpos3)&&(Stillingame[3]==1))){          //Check if
signallength are the same
                Action();
                Clean();
            }else{
                Clean();
            }
        }else{                                                          //Counting up one
              Comparevector[Cvpos] = (1 + (Comparevector[Cvpos]));
        }
    }else{                                                          //Change in insignal
           ++Cvpos;
           Comparevector[Cvpos] = 1;
           if (Cvpos==99){
                Clean();
            }else{
                Compare(); // Compare disabled
```

```
                Oneorzero = Signal;
            }
        }
}
void Compare(){                                         //Check if signals still
are equal
    if(Record==0){
        if((((Comparevector[(Cvpos-1)]&(0xFF))+Factor) > (Signalvector1[(Cvpos-1)]&(
0xFF)))&&((((Comparevector[(Cvpos-1)]&(0xFF))-Factor) < (Signalvector1[(Cvpos-1)]&(
0xFF))))){
            //Signal1 is correct so far
        }else{
            Stillingame[1]=0;
        }
        if((((Comparevector[(Cvpos-1)]&(0xFF))+Factor) > (Signalvector2[(Cvpos-1)]&(
0xFF)))&&((((Comparevector[(Cvpos-1)]&(0xFF))-Factor) < (Signalvector2[(Cvpos-1)]&(
0xFF))))){
            //Signal2 is correct so far
        }else{
            Stillingame[2]=0;
        }
        if((((Comparevector[(Cvpos-1)]&(0xFF))+Factor) > (Signalvector3[(Cvpos-1)]&(
0xFF)))&&((((Comparevector[(Cvpos-1)]&(0xFF))-Factor) < (Signalvector3[(Cvpos-1)]&(
0xFF))))){
            //Signal3 is correct so far
        }else{
            Stillingame[3]=0;
        }
        if ((Stillingame[1]+Stillingame[2]+Stillingame[3])==0){
            Clean();        //Start over from the begining
        }
    }
}
void Trigg(){                                           //Signal start is trigged
    Wftrigg=0;
    Comparevector[0]=1;
    Oneorzero=0;
    if (Recbutton==0){
        Record=1;
    }
}
void Clean(){                                           //Signal was wrong. start
over
    Cvpos=0;
    Wftrigg=1;
    Stillingame[1]=1;
    Stillingame[2]=1;
    Stillingame[3]=1;
}
void Action(){                                          //Signal was right do what
shall be done
    if(Stillingame[1]==1){
        if(Lampon==(0x02)){     //Turn off light
            Lampon=0;
        }else{ //Turn on light
            Lampon=(0x02);
        }
        PORTA=(PORTA|Lampon);
        pause=1;
        Blink=4;
        Pausetimes=1;
        Startblink=1;
    }else if(Stillingame[2]==1){
```

```
        if (Curtain==0){
            Startblink=1;
            forward=1;
            Pausetimes=75;
            pause=1;
            Motoron=1;// Turn motor on
            Curtain=1;
        }
    }else{
        if (Curtain==1){
            Startblink=1;
            forward=0;
            Pausetimes=75;
            pause=1;
            Motoron=1;// Turn motor on
            Curtain=0;
        }
    }
}
void Pause(){                                                    //Handle all timebased
    ++Pausetimer;
    if(Startblink==1){
        Lighton=0;
        Blinkled();
        Startblink=0;
    }else if (Motoron==1){
        if (Pausetimer==5000){
            --Pausetimes;
            Pausetimer=0;
            Runmotor();
        }
    }else if (Blink==1){
        if ((Pausetimes==5)||(Pausetimes==3)||(Pausetimes==1)){
            if (Pausetimer==30000){
                --Pausetimes;
                Pausetimer=0;
                Blinkled();
            }
        }else{
            if (Pausetimer==5000){
                --Pausetimes;
                Pausetimer=0;
                Blinkled();
            }
        }
    }else if(Blink==2){
        if (Pausetimes==4){
            if (Pausetimer==30000){
                --Pausetimes;
                Pausetimer=0;
                Blinkled();
            }
        }else{
            if (Pausetimer==5000){
                --Pausetimes;
                Pausetimer=0;
                Blinkled();
            }
        }
    }else if(Blink==3){
        if (Pausetimes==6){
            if (Pausetimer==30000){
                --Pausetimes;
```

```
                Pausetimer=0;
                Blinkled();
            }
        }else{
            if (Pausetimer==5000){
                --Pausetimes;
                Pausetimer=0;
                Blinkled();
            }
        }
    }else if(Blink==4){ //Action blink
        if (Pausetimer==10000){
            --Pausetimes;
            Pausetimer=0;
            Blinkled();
        }
    }else if(Blink==5){ //Save blink
        if (Pausetimer==50000){
            --Pausetimes;
            Pausetimer=0;
            Blinkled();
        }
    }
    if (Pausetimes==0){
        pause=0;
        PORTA=Lampon;
        PORTB=0;
        Pausetimer=0;
        Motoron=0;
    }
}
void Blinkled(){                                        //Switch led
    if(Lighton==1){
        PORTB=(0x00); //Turn off light
        Lighton=0;
    }else{
        PORTB=(0x01);//Turn on light
        Lighton=1;
    }
}
void Savesignal(){                                      //Store Signal
    if (Recposition==1){
        Svpos1=0;
        while(Cvpos>=Svpos1){
            Signalvector1[Svpos1]=Comparevector[Svpos1];
            ++Svpos1;
        }
        --Svpos1;
    }else if(Recposition==2){
        Svpos2=0;
        while(Cvpos>=Svpos2){
            Signalvector2[Svpos2]=Comparevector[Svpos2];
            ++Svpos2;
        }
        --Svpos2;
    }else{
        Svpos3=0;
        while(Cvpos>=Svpos3){
            Signalvector3[Svpos3]=Comparevector[Svpos3];
            ++Svpos3;
        }
        --Svpos3;
    }
```

```
    Record=0;
    pause=1;
    Blink=5;
    Pausetimes=1;
    Startblink=1;
}
void Changesignal(){                                    //Change Record-channel
    ++Recposition;
    if (Recposition==4){
        Recposition=1;
    }
    Blink=Recposition;
    pause=1;
    if (Blink==1){
        Pausetimes=6;
    }else if(Blink==2){
        Pausetimes=7;
    }else{
        Pausetimes=11;
    }
    Startblink=1;
}
void Runmotor(){                                        //Stepmotor one step
    if (forward==1){
        ++mipos;
        if (mipos==4) mipos=0;
    }else{
        --mipos;
        if (mipos==-1) mipos=3;
    }
    PORTA=(Motorinstuktion[mipos]);
    PORTA=(PORTA|Lampon);
}
void Init(){                                            //Initialize
//Configure interupts
//_SEI();   //Set globale interupt enable
OCR0=0xc8;   // Define resolution
TCCR0=0x09;   //shure yet, compare match, no prescaler
TIMSK=0x02; // enables interupt on match
//Set globale interupt enable
SREG=(SREG|(0x80));
//Set i/o pins
DDRA = 0xff;
DDRB = 0xff;
DDRD = 0x00;
//load stored signal
//setup motorinstuktion White, yellow, red, Blue
Motorinstuktion[0]=(0x30);
Motorinstuktion[1]=(0x90);
Motorinstuktion[2]=(0xc0);
Motorinstuktion[3]=(0x60);
//
Stillingame[1]=1;
Stillingame[2]=1;
Stillingame[3]=1;
}

//Clear Timer on Compare Match (CTC) interrupt
void SIG_OUTPUT_COMPARE0(void){
    PORTD=PIND;
    if ((PORTD&1)==0){   //Read signal
       Signal=0;
    }else{
```

```
        Signal=1;
    }
    Recbutton=(PORTD&4);                                //Read RecButton
    Changebutton=(PORTD&8);                             //Read ChangeButton

    if(pause==1){
        Pause();                                         //Count down pause timer
    }else if(Changebutton==0){                          //Change Recordsignal
        Changesignal();
    }else if(Wftrigg==1){
        if(Signal==0){                                  //Trigg
            Trigg();                                     //Start counting
        }
    }else{
        Count();                                         //Count upp Comparevector
    }
    SREG=(SREG|(0x80));
}

void main(){                                                    //Main
    Init();
    while(1){   //Infinit main loop
    }
}
```