

Projekt: Zigbee kommunikation

Dato: 24-02-2010

Afleverings Dato: 04-06-2009

Titel:

**Projekt rapport
for
Zigbee kommunikation**

6. semester projekt i WEM1 ved Ingeniørhøjskolen i Århus

Gruppe 2:

06709 Sean Koch Nyekjær (SN)

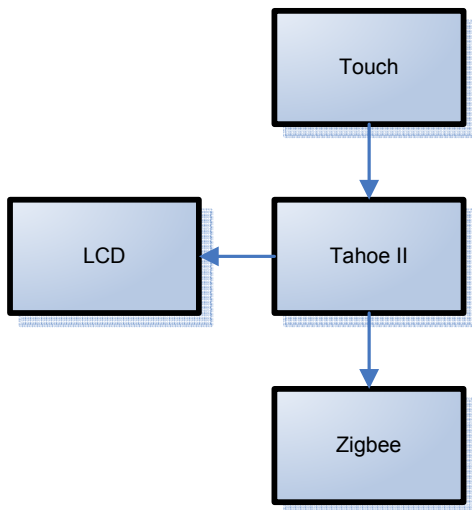
06843 Nicolaj Hougaard Andersen (NA)

Lære: Michael Alrøe

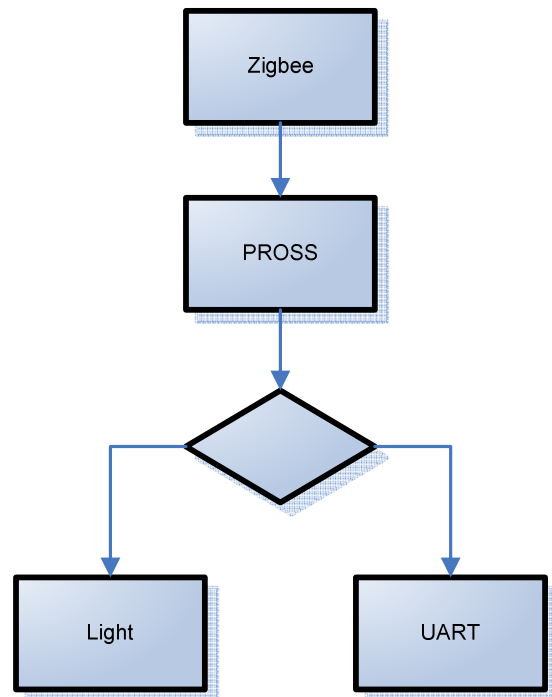
Indledning

Denne rapport beskriver udviklingen af et Zigbee kommunikations projekt, lavet i WEM1 6.semester på IHA. Projektet er lavet af os for at prøve at skabe vise den grundlæggende teori omkring Zigbee og hvordan man eventuelt kan kommunikere via dette med C# og micro frameworket.

Tahoe



PC-kort



Dia. 1

Ovenfor er indsat et simplificerende diagram over systemet når det er sat op og køre.

Der vil på LCD displayet af controller enheden være et visuelt design til brugeren der viser 3 knapper og en slider.

De tre knapper er udformet på touch og ligger ovenpå de visuelle knapper.

De tre knapper bruges til at aktivere og deaktivere tre lysdioder. Slideren bruger vi til at vise vi også kan bruge Zigbee til at sende data til en pc. På Pc'en bliver der vist et procent tal styret af slideren.

Vi har herunder indsat et billede fra LCD displayet.



Push on the buttons above to control the diodes



Use the slider function to control the winapp

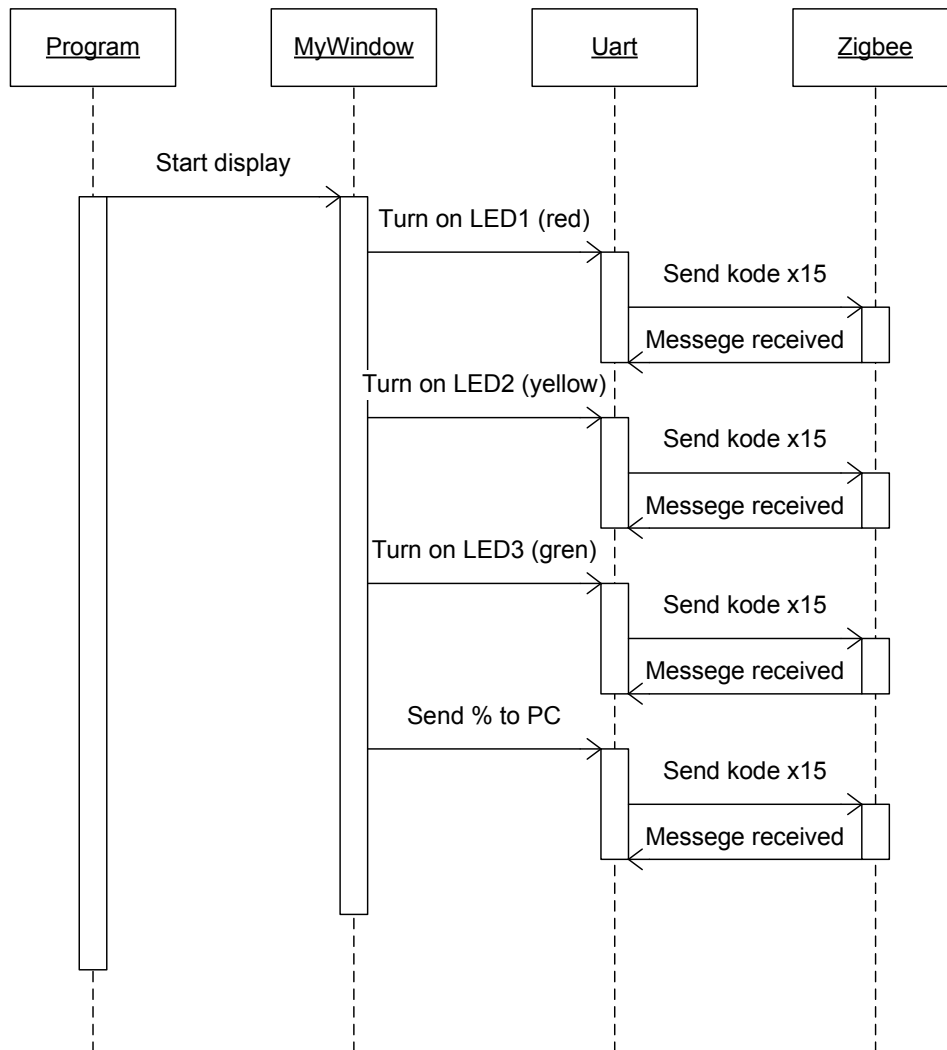
Indholdsfortegnelse

Indledning.....	2
Indholdsfortegnelse.....	4
System beskrivelse	5
Arkitektur Design.....	5
Hardware Design	6
Opsætning af koordinator og end device.....	7
Software Design.....	9
Uart.....	9
Zigbee	10
MyWindow	15
Pc baseret program	17
Test	18
Der trykkes på tænd/sluk knap	18
Slideren slides.....	18
Fremtid	19
Konklusion	19

System beskrivelse

Arkitektur Design

Sekvensdiagram.



Hardware Design

Til at udvikle vores Zigbee system, har vi brugt en række forskellige hardware dele, som vi herunder gerne vil beskrive.

Tahoe II:



Tahoe II er et udviklings kit til micro framework 3.0 programmering.

Tahoe II indeholder en række forskellige komponenter så som et farve LCD display med touch, og sokkel til Zigbee modul. Begge disse er noget som vi i vores projekt bruger.

Vi har også valgt at bruge det påmonteret LCD farve display, til at vise hvilken tilstand dioder er i samt styre lys dioderne via et Zigbee modul der passer til soklen på Tahoe II kittet.

Zigbee moduler:

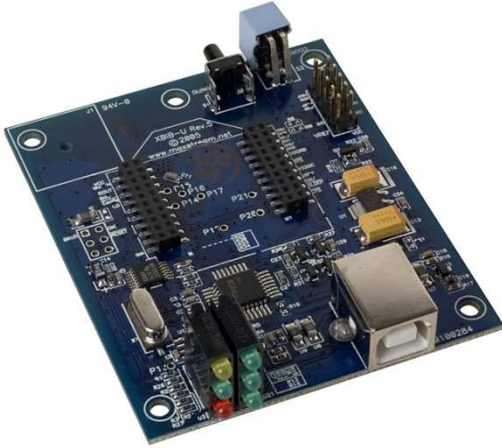
For at kommunikere via Zigbee er det nødvendigt med et Zigbee modul som skal fungere som end device, samt et modul der skal fungere som koordinator i den anden ende.

Til dette formål bruger vi et XBee modul som kan ses på billedet herunder.



Koordinatoren er sat på et medfølgende hardware kort der er lavet til USB tilslutning se næste afsnit.

USB tilslutnings enhed for Zigbee modul:

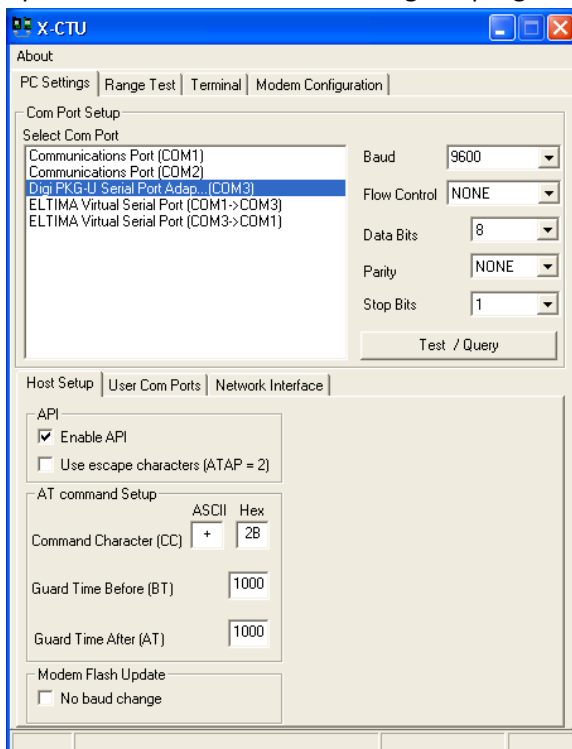


Denne enhed bruger vi til koordinator delen af systemet.

På printet påsættes et Zigbee modul magen til det der bruges på Tahoe II kittet. På printet er også et 10 pins stik som vi bruger til at påsætte vores tre lysdioder som skal styres. Kortet tilsluttes via USB, som virker som strømforsyning og som en virtuel serielport. Til Pc'en sendes det procent tal som slideren på Tahoe II kittet har sendt.

Opsætning af koordinator og end device

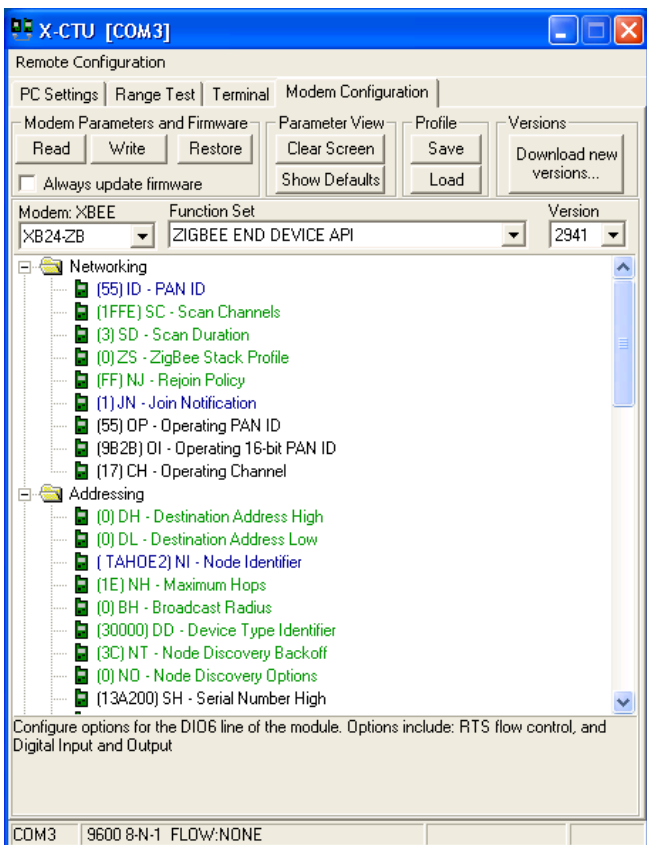
Koordinatoren er den som starter Zigbee netværket, hvor end devices og routerne tilslutter sig. Til at opsætte koordinatoren har vi brugt en program fra Digi der hedder X-CTU.



Da vi fik koordinatormodulet var det opsat til at køre med API kommandoer. Modulet kunne også have været opsat til at køre med AT kommandoer. Ved at bruge API kommandoer kommer man tættere på hardwaren, vi har valgt at bruge API kommandoer da man her kan sende opsætningskommandoer til de forskellige Zigbee devices i netværket.

Koordinatoren som sidder i pc-kortet er opsat til et PAN ID på 0x55, hvilket definerer hvilket netværksnummer Zigbee netværket til egner sig. Vi har sat koordinatoren til at acceptere alle devices der prøver på at joine vores netværk. Operating channel er sat til 0x17 som var det standard kanal valg. Sikkerheden er deaktiveret da vi er i gang med en prototype. Serielporten er opsat til en baudrate på 9600, ingen paritet, CTS flow control aktiveret. Det er på koordinatoren vores lysdioder sidder de styres fra end devicen.

End devicen der sidder på Tahoe II boardet, den har fået Node Identifier TAHOE2 så den kan identificeres af koordinatoren. End devicen prøver automatisk at tilslutte til et Zigbee netværk med et PAN ID på 0x55 og en operating channel på 0x17. Serielporten er opsat til en baud rate på 9600, ingen paritet, CTS flow control aktiveret.



Software Design

Uart

Denne klasse bruges til at opsætte uart porten på Tahoe II kittet
Denne uart port er her hvor Zigbee modulet er monteret.
Herunder vil vi komme med en kort forklaring på hver af de metoder vi bruger i denne klasse.

Uart:

Konstruktoren i Uart klassen bruges til at sætte hvilken COM port der skal benyttes og herefter kalde serialport_init() metoden til at opsætte porten.

```
public Uart(string COM)
{
    COMport = COM;
    serialport_init();
}
```

Serialport_init:

Denne metode kaldes i den ovenforstående Uart konstruktor. Formålet med metoden er at opsætte de forskellige parametre for COM porten.

Der opsættes en baud rate på 9600, en paritet på ingen, en 8 data bit, og en stop bit.

Det er Zigbee modul i standard opsætning.

```
public void serialport_init()
{
    serialPort = new SerialPort(COMport, 9600, Parity.None, 8, StopBits.One);
    serialPort.Handshake = System.IO.Ports.Handshake.RequestToSend;
    serialPort.WriteTimeout = 2000;
    serialPort.DataReceived += new
    System.IO.Ports.SerialDataReceivedEventHandler(serialport_DataReceived);
    serialPort.Open();
}
```

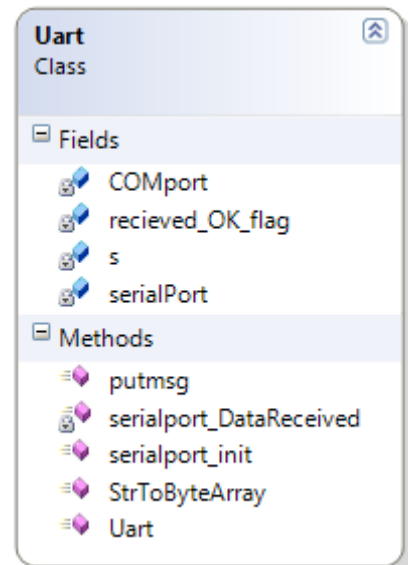
StrToByteArray:

Denne funktion bruges til at omligge en streng til et byte array. Dette gøres da serialport.write funktionen skal have et byte array og ikke en streng som man kan i almindelig .net.

```
public static byte[] StrToByteArray(string str)
{
    return System.Text.Encoding.UTF8.GetBytes(str);
}
```

Putmsg:

Denne funktion bruges til at sende meddelelsen ud på uart porten til Zigbee modulet.
Dette gør vi 15 gange da handshake med det sovende Zigbee modul ikke virker.
Vi er med de 15 gange sikre på at beskeden blev sendt.



Hvis der skulle ske at svar bliver modtaget fra koordinatoren vil den hoppe ud af while lykken og der vil ikke blive sendt flere beskeder.

```
public static void putmsg(byte[] msg,byte cycles)
{
    byte cnt = 0;
    while (recieved_OK_flag == false)
    {
        serialPort.Write(msg, 0, msg.Length);
        cnt++;
        if(cnt == cycles)
        {
            break;
        }
    }
    recieved_OK_flag = false;
}
```

Serialport_DataReceived:

Denne funktion bruges til at modtage meddelelsen fra uart porten kommende fra Zigbee modulet. Hvis den modtaget besked er over fire karaktere lang, og første karakter indeholder 0x7E, bliver vores recive ok flag sat højt.

```
private static void serialport_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
    int cnt = serialPort.BytesToRead;
    byte[] inputdata = new byte[cnt];
    serialPort.Read(inputdata, 0, cnt);

    if (cnt > 4)
    {
        if (inputdata[0] == 0x7E)
        {
            recieved_OK_flag = true;
            Debug.Print("Message Recieved!");
        }
    }
}
```

Zigbee

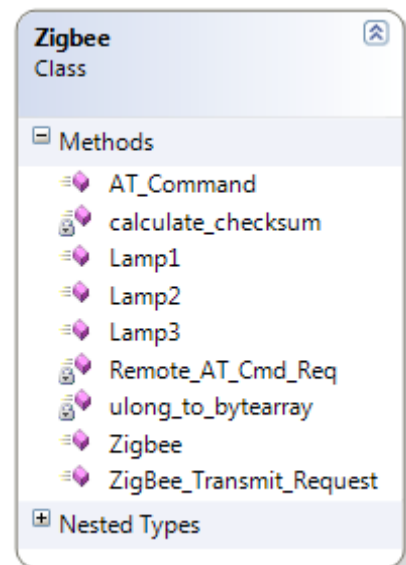
Denne klasse bruges til at kontrollere hvad der sendes fra Zigbee modulet samt hvortil.

Det er også i denne klasse opsætningen af Zigbee modulet foregår, som specificeret i databladet.

Zigbee:

Denne konstruktor bruges til at oprette klassen Uart.

```
public Zigbee()
{
    Uart uart = new Uart("COM2");
}
```



Zigbee_transmit_Request:

Denne funktion bruges til at sende et byte array til modtageren på Zigbee netværket.

Koordinator adressen er som kan ses i skemaet herunder 0x0000000000000000 samt at RF data er en vores payload som sendes til koordinatoren.

Opsætningen er gjort som i skemaet herunder

Frame Fields		Offset	Example	Description	
API Packet	Start Delimiter	0	0x7E		
	Length	MSB 1	0x00	Number of bytes between the length and the checksum	
		LSB 2	0x16		
	Frame-specific Data	Frame Type	3	0x10	
	Frame-specific Data	Frame ID	4	0x01	Identifies the UART data frame for the host to correlate with a subsequent ACK (acknowledgement). If set to 0, no response is sent.
		64-bit Destination Address	MSB 5	0x00	Set to the 64-bit address of the destination device. The following addresses are also supported: 0x0000000000000000 - Reserved 64-bit address for the coordinator 0x000000000000FFFF - Broadcast address
			6	0x13	
			7	0xA2	
			8	0x00	
			9	0x40	
			10	0x0A	
			11	0x01	
		LSB 12	0x27		
		16-bit Destination Network Address	MSB 13	0xFF	Set to the 16-bit address of the destination device, if known. Set to 0xFFFE if the address is unknown, or if sending a broadcast.
			LSB 14	0xFE	
		Broadcast Radius	15	0x00	Sets maximum number of hops a broadcast transmission can occur. If set to 0, the broadcast radius will be set to the maximum hops value.
		Options	16	0x00	All other bits must be set to 0.
	RF Data	17	0x54	Data that is sent to the destination device	
		18	0x78		
		19	0x44		
		20	0x61		
		21	0x74		
		22	0x61		
		23	0x30		
		24	0x41		
Checksum	25	0x13	0xFF - the 8 bit sum of bytes from offset 3 to this byte.		

Skema taget fra datablade /manualen til Zigbee modulet s. 89

AT_Command:

Denne funktion bruges til at sende en AT kommando til det tilsluttede Zigbee modul.

Opsætningen er gjort som i skemaet herunder

Frame Fields		Offset	Example	Description	
A P I P a c k e t	Start Delimiter	0	0x7E		
	Length	MSB 1	0x00	Number of bytes between the length and the checksum	
		LSB 2	0x10		
	Frame-specific Data	Frame Type	3	0x17	
		Frame ID	4	0x01	Identifies the UART data frame for the host to correlate with a subsequent ACK (acknowledgement). If set to 0, no response is sent.
		64-bit Destination Address	MSB 5	0x00	Set to the 64-bit address of the destination device. The following addresses are also supported: 0x0000000000000000 - Reserved 64-bit address for the coordinator 0x000000000000FFFF - Broadcast address
			6	0x13	
			7	0xA2	
			8	0x00	
			9	0x40	
			10	0x40	
			LSB 12	0x22	
		16-bit Destination Network Address	MSB 13	0xFF	Set to the 16-bit address of the destination device, if known. Set to 0xFFFE if the address is unknown, or if sending a broadcast.
			LSB 14	0xFE	
	Remote Command Options	15	0x02 (apply changes)	0x02 - Apply changes on remote. (If not set, AC command must be sent before changes will take effect.) All other bits must be set to 0.	
	AT Command	16	0x42 (B)	Name of the command	
		17	0x48 (H)		
	Command Parameter	18	0x01	If present, indicates the requested parameter value to set the given register. If no characters present, the register is queried.	
	Checksum	18	0xF5	0xFF - the 8 bit sum of bytes from offset 3 to this byte.	

Remote_AT_Cmd_Req:

Denne funktion bruges til at sende en AT kommando til et vilkårligt modul på Zigbee netværket. Opsætningen er gjort som i skemaet herunder

Frame Fields		Offset	Example	Description	
A P I P a c k e t	Start Delimiter	0	0x7E		
	Length	MSB 1	0x00	Number of bytes between the length and the checksum	
		LSB 2	0x10		
	Frame-specific Data	Frame Type	3	0x17	
	64-bit Destination Address	Frame ID	4	0x01	Identifies the UART data frame for the host to correlate with a subsequent ACK (acknowledgement). If set to 0, no response is sent.
			MSB 5	0x00	Set to the 64-bit address of the destination device. The following addresses are also supported: 0x0000000000000000 - Reserved 64-bit address for the coordinator 0x000000000000FFFF - Broadcast address
			6	0x13	
			7	0xA2	
			8	0x00	
			9	0x40	
			10	0x40	
			11	0x11	
			LSB 12	0x22	
		16-bit Destination Network Address	MSB 13	0xFF	Set to the 16-bit address of the destination device, if known. Set to 0xFFFE if the address is unknown, or if sending a broadcast.
	LSB 14		0xFE		
Remote Command Options	15	0x02 (apply changes)	0x02 - Apply changes on remote. (If not set, AC command must be sent before changes will take effect.) All other bits must be set to 0.		
AT Command	16	0x42 (B)	Name of the command		
	17	0x48 (H)			
Command Parameter	18	0x01	If present, indicates the requested parameter value to set the given register. If no characters present, the register is queried.		
Checksum		18	0xF5	0xFF - the 8 bit sum of bytes from offset 3 to this byte.	

LampX:

Denne funktion bruges til at tænde og slukke for de enkelte lamper. Dette er eksempel fra Lamp1

```
public void Lamp1(bool ONOFF)
{
    if (ONOFF)
    {
        Remote_AT_Cmd_Req(Digital_IO_control.High, AT_Cmd.Ascii1,
        Destination_64bit.Coordinator, Destination_16bit.Broadcast);
    }
    else
    {
        Remote_AT_Cmd_Req(Digital_IO_control.Low, AT_Cmd.Ascii1,
        Destination_64bit.Coordinator, Destination_16bit.Broadcast);
    }
}
```

```
    }  
}
```

Ulong_to_bytearray:

Denne funktion bruges til at tage et unsigned 64 bit tal og konvertere det til et bytearray.

```
private byte[] ulong_to_bytearray(ulong x, byte bytecnt)  
{  
    byte[] result;  
  
    result = new byte[bytecnt];  
    for (byte i = 0; i < bytecnt; i++)  
    {  
        result[i] = (byte)(x >> (((bytecnt - i)*8)-8));  
    }  
  
    return result;  
}
```

Calculate_checksum:

Denne funktion bruges til at udregne en checksum til vores API frame.

Funktionen bliver kaldt inden start delimitoren og længden bliver sat ind, da checksummen ikke indeholder information om dette.

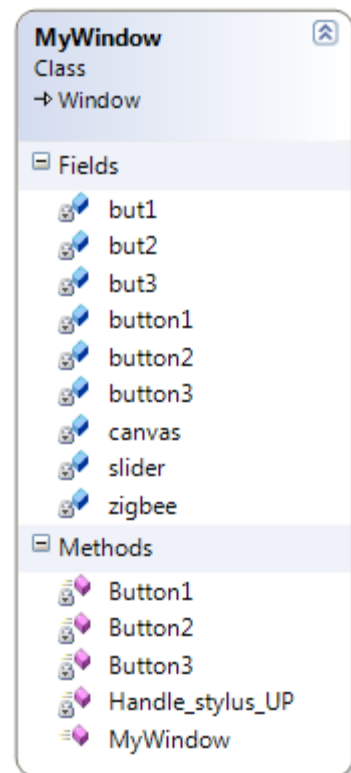
Checksummen bliver udregnet ved man tager summen, af bytearrayet eksklusiv start delimitoren og længden. Herefter returneres de nederste 8 bit, og hvorefter 0xFF bliver trukket fra resultatet.

```
private byte calculate_checksum(byte[] msg)  
{  
    byte length, checksum;  
    uint checksum_calc = 0;  
  
    length = (byte)msg.Length;  
    for (int i=0; i < length; i++)  
    {  
        checksum_calc = checksum_calc + msg[i];  
    }  
  
    checksum = (byte)checksum_calc;  
    return checksum;  
}
```

MyWindow

Denne klasse bruges til at opsætte det display, som brugeren kontrollerer de 4 enheder med.

Der opsættes det grafiske samt touch funktioner til de tre knapper og sliden specificeres her.



MyWindow:

Denne konstruktør bruges til at opsætte det visuelle på skærmen så som tekst og billeder til knapperne.

Dette gøres ved at oprette et canvas hvor vi putter tekst og billeder ind som children.

Ud over dette oprettes events til touch knapperne og til canvasen.

```
public MyWindow()  
{  
    //Sets text1  
    Font font = Resources.GetFont(Resources.FontResources.small);  
    Text text1 = new Text(font, "Push on the buttons above to");  
    text1.ForeColor = Color.Black;  
    Canvas.SetLeft(text1, 80);  
    Canvas.SetTop(text1, 130);  
  
    //Sets text2  
    Text text2 = new Text(font, "control the diodes");  
    text2.ForeColor = Color.Black;  
    Canvas.SetLeft(text2, 80);  
    Canvas.SetTop(text2, 140);  
  
    //Sets text3  
    Text text3 = new Text(font, "Use the slider function to control the winapp");  
    text3.ForeColor = Color.Black;  
    Canvas.SetLeft(text3, 40);  
    Canvas.SetTop(text3, 200);  
  
    //Button1 image  
    Image button1 = new Image(Resources.GetBitmap(Resources.BitmapResources.RedB_black));
```



```

Canvas.SetLeft(button1, 30);
Canvas.SetTop(button1, 50);
button1.StylusUp += new StylusEventHandler(Button1);

//Button2 image
Image button2 = new
Image(Resources.GetBitmap(Resources.BitmapResources.YellowB_black));
Canvas.SetLeft(button2, 130);
Canvas.SetTop(button2, 50);
button2.StylusUp += new StylusEventHandler(Button2);

//Button3 image
Image button3 = new
Image(Resources.GetBitmap(Resources.BitmapResources.GreenB_black));
Canvas.SetLeft(button3, 230);
Canvas.SetTop(button3, 50);
button3.StylusUp += new StylusEventHandler(Button3);

//Slider image and slider background
slider.Bitmap = Resources.GetBitmap(Resources.BitmapResources.slide);
Canvas.SetLeft(slider, 40);
Canvas.SetTop(slider, 180);
Bitmap slider_bg_bmp = new Bitmap(240,10);
slider_bg_bmp.DrawRectangle(Color.White, 1, 0, 0, 240, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0);
slider_bg_bmp.Flush();
Image slider_bg = new Image(slider_bg_bmp);
Canvas.SetLeft(slider_bg, 40);
Canvas.SetTop(slider_bg, 180);

// Add the text controls to the window
this.Child = canvas;
canvas.Children.Add(text1);
canvas.Children.Add(text2);
canvas.Children.Add(text3);
canvas.Children.Add(button1);
canvas.Children.Add(button2);
canvas.Children.Add(button3);
canvas.Children.Add(slider_bg);
canvas.Children.Add(slider);

//Make event for stylus press on canvas
canvas.StylusUp += new StylusEventHandler(Handle_stylus_UP);
}

```

ButtonX:

Denne funktion bruges til at tænde og slukke for koordinatorens dioder, samt ændre billedet i canvasen. Dette er eksempel fra Button1

```

void Button1(object sender, StylusEventArgs e)
{
    Image img = (Image)sender;
    but1 = !but1;
    Zigbee.Lamp1(but1);
    if (but1)
    {
        img.Bitmap =
Resources.GetBitmap(Resources.BitmapResources.RedB_black);
    }
    else
    {
        img.Bitmap = Resources.GetBitmap(Resources.BitmapResources.RedB);
    }
    Debug.Print("Button1");
    Invalidate();
}

```

Handle_stylus_UP:

Denne funktion bruges til rette på slideren hvis der trykkes indenfor et bestemt område på canvasen, hvorefter teksten sendes til koordinatoren.

```
void Handle_stylus_UP(object sender, StylusEventArgs e)
{
    base.OnStylusUp(e);

    int x;
    int y;

    e.GetPosition((UIElement)this, out x, out y);

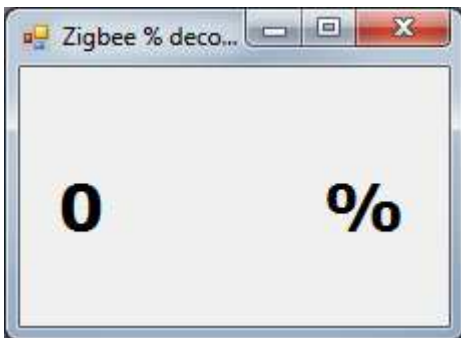
    if (y < 210 && y > 155 && x < 280 && x > 40) //Hvis der er trykket inden for sliderens
    område
    {
        Canvas.SetLeft(slidebar, x);
        //Calculates the %
        int procent = (int)((double)((x - 40) / 240.0) * 100.0);
        //Send it to the coordinator
        Zigbee.Zigbee_Transmit_Request(procent.ToString(),
        Zigbee.Destination_64bit.Coordinator,
        Zigbee.Destination_16bit.Broadcast);
        Debug.Print("Slider captured!");
    }

    Debug.Print(x.ToString() + "x, " + y.ToString() + "y");

    Invalidate();
}
```

Pc baseret program

Vi har lavet et program der kan vise outputtet fra slideren, det er skrevet i c#. Det er bare et lille program der kan decode en Zigbee API streng og vise data indholdet.



Test

Der trykkes på tænd/sluk knap

ID: 1	INIT: SKN
Beskrivelse: Når systemet er startet op og Zigbee modulerne har fået kontakt til hinanden. Trykkes der på en tænd/sluk knap.	
Succes Kriterier: Dioden til den tilhørende knap på koordinatoren toggles.	
Forudsætninger: Zigbee end devicen har kontakt til koordinatoren.	

Step	Input	Forventet Output	Ok
1.	Der trykkes på en tænd/sluk knap	Dioden i den samme farve som knappen tændes eller slukkes på koordinatoren.	OK

Test Resumé: Testen forløb tilfredsstillende men beskeden ikke altid blev modtaget da vi har nogle problemer med handshake.

Slidern slides

ID: 2	INIT: SKN
Beskrivelse: Når systemet er startet op og Zigbee modulerne har fået kontakt til hinanden. Slides slidern.	
Succes Kriterier: Vores PC program modtager beskeden	
Forudsætninger: Zigbee end devicen har kontakt til koordinatoren. Pc programmet er startet.	

Step	Input	Forventet Output	Ok
2.	Slidern slides	PC programmet viser sliderens procent.	OK

Test Resumé: Testen forløb tilfredsstillende men beskeden ikke altid blev modtaget da vi har nogle problemer med handshake.

Fremtid

I en fremtidig version vil der kunne bygges flere enheder på. Samt enhederne kan være af forskellige typer, så som radiator kontrol, TV, Radio ja mulighederne er mange.

Displayet vil kunne indeholde flere sider en til hver funktion så som, Lys, Varme, TV, Radio osv.

Ud over alle forbedringerne skal der også findes en løsning på handshaket så det ikke er nødvendigt at sende beskeden 15 gange.

Konklusion

Dette projekt har ikke det største produkt at vise men da vi skulle vælge et projekt, valgte vi at prøve noget med Zigbee og sætte os lidt nærmere ind i det. Vi mener også vi har vist at vores viden indenfor design i micro framework og brug af dette.

Vi har haft lidt problemer med handshake fra et sovende Zigbee modul, hvilket vi valgte at gå uden om ved at sende beskeden 15 gange, da vi så mente at beskeden ville være blevet modtaget.

De 15 gange er ikke en optimal løsning, men vi har ikke kunne finde en løsning på problemet, og ud fra de test vi har fortaget virker det til at løsningen med de 15 gange til at fungere.

Vi har under projektet skulle undersøge en række ting omkring Zigbee for at blive klogere omkring denne kommunikationsform. Ud fra dette vi kalde projektet vellykket da det var en del af det vi ønskede med projektet.