function Decoder(bytes, port) {

//LT33222-L or LT22222-L Decode

if(port==0x02)

{

var hardware= (bytes[10] & 0xC0)>>6;

var mode0= bytes[10] & 0xff;

var mode= bytes[10] & 0x3f;

var decode = {};

if(hardware=='0')

{

decode.Hardware\_mode="LT33222";

decode.DO3\_status=(bytes[8] &0x04)? "L":"H";

if(mode0=='1')

{

decode.DI3\_status= (bytes[8] &0x20)?"H":"L";

}

}

else if(hardware=='1')

{

decode.Hardware\_mode= "LT22222";

}

if(mode!=6)

{

decode.DO1\_status= (bytes[8] &0x01)? "L":"H";

decode.DO2\_status= (bytes[8] &0x02)? "L":"H";

decode.RO1\_status= (bytes[8] &0x80)? "ON":"OFF";

decode.RO2\_status= (bytes[8] &0x40)? "ON":"OFF";

if(mode!=1)

{

if(mode!=5)

{

decode.Count1\_times= (bytes[0]<<24 | bytes[1]<<16 | bytes[2]<<8 | bytes[3])>>>0;

}

decode.First\_status= (bytes[8] &0x20)? "Yes":"No";

}

}

if(mode=='1')

{

decode.Work\_mode= "2ACI+2AVI";

decode.AVI1\_V= parseFloat(((bytes[0]<<24>>16 | bytes[1])/1000).toFixed(3));

decode.AVI2\_V= parseFloat(((bytes[2]<<24>>16 | bytes[3])/1000).toFixed(3));

decode.ACI1\_mA= parseFloat(((bytes[4]<<24>>16 | bytes[5])/1000).toFixed(3));

decode.ACI2\_mA= parseFloat(((bytes[6]<<24>>16 | bytes[7])/1000).toFixed(3));

decode.DI1\_status= (bytes[8] &0x08)? "H":"L";

decode.DI2\_status= (bytes[8] &0x10)? "H":"L";

}

else if(mode=='2')

{

decode.Work\_mode= "Count mode 1";

decode.Count2\_times= (bytes[4]<<24 | bytes[5]<<16 | bytes[6]<<8 | bytes[7]) >>>0;

}

else if(mode=='3')

{

decode.Work\_mode= "2ACI+1Count";

decode.ACI1\_mA= parseFloat(((bytes[4]<<24>>16 | bytes[5])/1000).toFixed(3));

decode.ACI2\_mA= parseFloat(((bytes[6]<<24>>16 | bytes[7])/1000).toFixed(3));

}

else if(mode=='4')

{

decode.Work\_mode= "Count mode 2";

decode.Acount\_times= (bytes[4]<<24 | bytes[5]<<16 | bytes[6]<<8 | bytes[7]) >>>0;

}

else if(mode=='5')

{

decode.Work\_mode= " 1ACI+2AVI+1Count";

decode.AVI1\_V= parseFloat(((bytes[0]<<24>>16 | bytes[1])/1000).toFixed(3));

decode.AVI2\_V= parseFloat(((bytes[2]<<24>>16 | bytes[3])/1000).toFixed(3));

decode.ACI1\_mA= parseFloat(((bytes[4]<<24>>16 | bytes[5])/1000).toFixed(3));

decode.Count1\_times= bytes[6]<<8 | bytes[7];

}

else if(mode=='6')

{

decode.Work\_mode= "Exit mode";

decode.Mode\_status= bytes[9] ? "True":"False";

decode.AV1L\_flag= (bytes[0] &0x80)? "True":"False";

decode.AV1H\_flag= (bytes[0] &0x40)? "True":"False";

decode.AV2L\_flag= (bytes[0] &0x20)? "True":"False";

decode.AV2H\_flag= (bytes[0] &0x10)? "True":"False";

decode.AC1L\_flag= (bytes[0] &0x08)? "True":"False";

decode.AC1H\_flag= (bytes[0] &0x04)? "True":"False";

decode.AC2L\_flag= (bytes[0] &0x02)? "True":"False";

decode.AC2H\_flag= (bytes[0] &0x01)? "True":"False";

decode.AV1L\_status= (bytes[1] &0x80)? "True":"False";

decode.AV1H\_status= (bytes[1] &0x40)? "True":"False";

decode.AV2L\_status= (bytes[1] &0x20)? "True":"False";

decode.AV2H\_status= (bytes[1] &0x10)? "True":"False";

decode.AC1L\_status= (bytes[1] &0x08)? "True":"False";

decode.AC1H\_status= (bytes[1] &0x04)? "True":"False";

decode.AC2L\_status= (bytes[1] &0x02)? "True":"False";

decode.AC2H\_status= (bytes[1] &0x01)? "True":"False";

decode.DI2\_status= (bytes[2] &0x08)? "True":"False";

decode.DI2\_flag= (bytes[2] &0x04)? "True":"False";

decode.DI1\_status= (bytes[2] &0x02)? "True":"False";

decode.DI1\_flag= (bytes[2] &0x01)? "True":"False";

}

if(bytes.length!=1)

return decode;

}

else if(port==5)

{

var freq\_band;

var sub\_band;

if(bytes[0]==0x01)

freq\_band="EU868";

else if(bytes[0]==0x02)

freq\_band="US915";

else if(bytes[0]==0x03)

freq\_band="IN865";

else if(bytes[0]==0x04)

freq\_band="AU915";

else if(bytes[0]==0x05)

freq\_band="KZ865";

else if(bytes[0]==0x06)

freq\_band="RU864";

else if(bytes[0]==0x07)

freq\_band="AS923";

else if(bytes[0]==0x08)

freq\_band="AS923\_1";

else if(bytes[0]==0x09)

freq\_band="AS923\_2";

else if(bytes[0]==0x0A)

freq\_band="AS923\_3";

else if(bytes[0]==0x0F)

freq\_band="AS923\_4";

else if(bytes[0]==0x0B)

freq\_band="CN470";

else if(bytes[0]==0x0C)

freq\_band="EU433";

else if(bytes[0]==0x0D)

freq\_band="KR920";

else if(bytes[0]==0x0E)

freq\_band="MA869";

if(bytes[1]==0xff)

sub\_band="NULL";

else

sub\_band=bytes[1];

var firm\_ver= (bytes[2]&0x0f)+'.'+(bytes[3]>>4&0x0f)+'.'+(bytes[3]&0x0f);

var tdc\_time= bytes[4]<<16 | bytes[5]<<8 | bytes[6];

return {

FIRMWARE\_VERSION:firm\_ver,

FREQUENCY\_BAND:freq\_band,

SUB\_BAND:sub\_band,

TDC\_sec:tdc\_time,

}

}

}