

```

#define ADDRESS "2";           // eigene Adresse
#define DESTINATION "1";      // Zieladresse
#define TRENN "!";           //Trennzeichen für die Nachricht

byte laengenachricht;        //Variable für Nachrichtenlänge

sub SendeNachricht(string versenden)

{
    //Sendevorgang
    laengenachricht = ArrayLen(versenden);
    SetHSOutputBuffer(0, laengenachricht, versenden);
    SetHSOutputBufferOutPtr(0);
    SetHSOutputBufferInPtr(laengenachricht);
    SetHSState(HS_SEND_DATA);
    SetHSFlags(HS_UPDATE);
}

task main()
{
    SetSensorLowspeed(IN_2);           //Ultraschallsensor
    SetSensorType(IN_4, SENSOR_TYPE_HIGHSPEED); //Verbindung zum zweiten
NXT
    SetHSState(HS_INITIALISE);
    SetHSFlags(HS_UPDATE);
    SetHSInputBufferInPtr(0);
    /*
    Verschiedene Variablen für die ein und ausgehende Nachricht.
    Variablen für Ziel und Trennzeichen.
    */
    string empfangen, versenden, tmp, dest, tren, adre, message;

    bool token = false; //Variable für Token, dieser NXT besitzt den
Token
    bool send = false; //Variable ob der NXT die Nachricht bereits
gesendet hat

    int distance, i, y, x, addr, adreascii, trenascii; //Variablen um den
empfangenen String in seine
//Einzelteile -Adresse, Trennzeichen,
Nutzdaten zu unterteilen
    adre = ADDRESS;
    dest = DESTINATION;
    tren = TRENN;
    adreascii = StrIndex(adre, 0);
    trenascii = StrIndex(tren, 0);

    while(true)
    {

        i = 0;

        if(!token) //besitzt der NXT den Token nicht, versucht er zu

```

```

empfangen
{
    Wait(50);
    laengenachricht = HSInputBufferInPtr();
    ClearScreen();
    if(laengenachricht > 0)
    {
        GetHSInputBuffer(0, laengenachricht, empfangen); //empfangen der
Nachricht
        addr = StrIndex(empfangen, 0); //ASCII der ersten Stelle des
empfangenen Strings
        TextOut(0, LCD_LINE4, empfangen);

        if(addr == adreascii) //prüfen der ersten Stelle des empfangen
Strings /Adresse
        {
            while(i < laengenachricht)
            {
                if(trenascii == StrIndex(empfangen, i) && y == 0)
//Adresse/Trennzeichen von
                {
                    //Nutzlast
trennen
                    y = i + 1;
                    i++;
                }
                else if(trenascii == StrIndex(empfangen, i) && y != 0)
                {
                    x = i;
                    x = x - y;
                    message = SubStr(empfangen, y, x);
                    i++;
                }

                else
                {
                    i++;
                }
            }

            TextOut(0, LCD_LINE1, message); //Nutzlast ausgeben

            SetHSInputBufferInPtr(0);

            Wait(50);
            empfangen = ""; //Nachricht leeren
            token = true; //der NXT darf nun Nachrichten senden

            ClearScreen();
        }

    }

}

else if(token && !send) //wenn der Token im Besitz ist und noch

```

```

keine
{
    //Nachricht gesendet wurde, wird die
    Nachricht
    distance = SensorUS(IN_2); //zusammengestellt und versendet
    tmp = NumToStr(distance);
    versenden = StrCat(dest,tren,tmp,tren);
    SendeNachricht(versenden);
    NumOut(0, LCD_LINE2, distance);

    send = true;
}

else //wenn der Token im Besitz ist und die
    Nachricht //bereits gesendet wurde, wird der Token
    {
        gelöscht
        token = false;
        send = false;

    }

}

```

Quelle: <http://www.mindstormsforum.de/viewtopic.php?t=5865>