

Messen, Steuern u. Regeln mit dem STR9-ComStick

Ausgehend von der Beispielanwendung ADC habe ich in
C:\Programme\Hitex\HiTOP52-STR9-comStick\Examples\STR9\Hitex-
STR912\GNU\ADC\Source\main.c den Quelltext

```
#include "defines.h"

#define global extern /* to declare external variables and functions */
#include "91x_lib.h"

#include "main.h"
#define GPIO_Alt1 0x01

#include <stdio.h>
#include <string.h>

#define TxBufferSize (countof(TxBuffer) - 1)
#define RxBufferSize 0xFF

/* Private macro -----*/
#define countof(a) (sizeof(a) / sizeof(*(a)))

/* Private variables -----*/
UART_InitTypeDef UART_InitStructure;
u8 TxBuffer[] = "UART1 - 1,8,N,1@115.2k communication without flow control\n\r";
u8 RxBuffer[RxBufferSize];
u8 NbrOfDataToTransfer = TxBufferSize;
u8 TxCounter = 0;
u8 RxCounter = 0;

GPIO_InitTypeDef GPIO_InitStructure;
TIM_InitTypeDef TIM_InitStructure;
ADC_InitTypeDef ADC_InitStructure;

/* Private function prototypes -----*/
void SCU_Configuration(void);
void GPIO_Configuration(void);
void ADC_Configuration(void);
void UART1_Configuration(void);
static void Delay(u32 nCount);

int main (void)
{

    u16 Conversion_Value = 0;
```

```

/* Configure the system clocks */
SCU_Configuration();
/* Configure the GPIOs */
GPIO_Configuration();
/* Configure and start the ADC */
ADC_Configuration();

UART1_Configuration();

/* endless loop */
while (1)
{
    /* Wait until conversion completion */
    while(ADC_GetFlagStatus(ADC_FLAG_ECV) == RESET);

    /* Get the conversion value */
    Conversion_Value = ADC_GetConversionValue(ADC_Channel_6);

    /* Clear the end of conversion flag */
    ADC_ClearFlag(ADC_FLAG_ECV);

    sprintf((u8 *)&TxBuffer,"ADC 6: %5u\n",Conversion_Value);
    TxCounter = 0;
    NbrOfDataToTransfer = 14;

    while(NbrOfDataToTransfer--)
    {
        UART_SendData(UART1, TxBuffer[TxCounter++]);
        while(UART_GetFlagStatus(UART1, UART_FLAG_TxFIFOFull) != RESET);
    }

    /* Turn OFF led connected to P9.0 pin */
    GPIO_WriteBit(GPIO9, GPIO_Pin_0, Bit_SET);

    /* Insert delay */
    Delay(0x7FFFF);

    /* Turn ON led connected to P9.0, pin */
    GPIO_WriteBit(GPIO9, GPIO_Pin_0, Bit_RESET);

    /* Insert delay */
    Delay(0x7FFFF);
}
}

```

abgeändert in

```

#include "defines.h"

#define global extern /* to declare external variables and functions */
#include "91x_lib.h"

#include "main.h"
#define GPIO_Alt1 0x01

#include <stdio.h>
#include <string.h>

#define TxBufferSize (countof(TxBuffer) - 1)
#define RxBufferSize 0xFF

/* Private macro -----*/
#define countof(a) (sizeof(a) / sizeof(*(a)))

/* Private variables -----*/
UART_InitTypeDef UART_InitStructure;
u8 TxBuffer[] = "UART1 - 1,8,N,1@115.2k communication without flow control\n\r";
u8 RxBuffer[RxBufferSize];
u8 NbrOfDataToTransfer = TxBufferSize;
u8 TxCounter = 0;
u8 RxCounter = 0;

GPIO_InitTypeDef GPIO_InitStructure;
TIM_InitTypeDef TIM_InitStructure;
ADC_InitTypeDef ADC_InitStructure;

/* Private function prototypes -----*/
void SCU_Configuration(void);
void GPIO_Configuration(void);
void ADC_Configuration(void);
void UART1_Configuration(void);
static void Delay(u32 nCount);

int main (void)
{

    u16 Conversion_Value = 0;

    /* Configure the system clocks */
    SCU_Configuration();
    /* Configure the GPIOs */
    GPIO_Configuration();
    /* Configure and start the ADC */
    ADC_Configuration();

    UART1_Configuration();

```

```

/* endless loop */
while (1)
{
    /* Wait until conversion completion */
    while(ADC_GetFlagStatus(ADC_FLAG_ECV) == RESET);

    /* Get the conversion value */
    Conversion_Value = ADC_GetConversionValue(ADC_Channel_6);

    /* Clear the end of conversion flag */
    ADC_ClearFlag(ADC_FLAG_ECV);

    //send u16-data
    UART_SendData(UART1,(u8)(Conversion_Value >> 8));
    while(UART_GetFlagStatus(UART1, UART_FLAG_TxFIFOFull) != RESET);
    UART_SendData(UART1,(u8)Conversion_Value);
    while(UART_GetFlagStatus(UART1, UART_FLAG_TxFIFOFull) != RESET);

    /* Insert delay to help SerialPort*/
    Delay(0xFFF);
}
}

```

Mit der im Internet frei erhältlichen Microsoft Visual C# 2005 Express Edition habe ich die Conversion_Value-Werte graphisch dargestellt:

```

using System;
using System.IO.Ports;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace Normal01
{
    public partial class Form1 : Form
    {
        private double xmin, xmax, ymin, ymax;
        private double xStreckfaktor, yStreckfaktor;
        double yA=0;
        static SerialPort _serialPort = null;
        Boolean com_is_open = false;
        int i_wait = 1;

        public Form1()
        {
            InitializeComponent();
            xmax = 10;
            xmin = -.2;
            ymax = 1150;

```

```

    ymin = -50;
    xStreckfaktor = this.ClientRectangle.Width / (xmax - xmin);
    yStreckfaktor = this.ClientRectangle.Height / (ymax - ymin);
}

public int xX(double x)
{
    return (int)Math.Round(xStreckfaktor * (x - xmin));
}

public int yY(double y)
{
    return (int)Math.Round(yStreckfaktor * (ymax - y));
}

public double Xx(int x)
{
    return x / xStreckfaktor + xmin;
}

public double Yy(int y)
{
    return ymax - y / yStreckfaktor;
}

public void Graph2(Graphics g, Color c, int delay) //delay >=1 wählen
{
    Font fn = new Font("Verdana", 12);
    Brush br = new SolidBrush(Color.Black);
    double x, dx, y = 500;
    int n1, n=1, xvon, yvon, xbis, ybis;
    Pen pen = new Pen(c, 1);
    dx = (xmax - xmin) / this.ClientRectangle.Width;
    x = 0; xvon = xX(x);
    try
    {
        n1 = _serialPort.ReadByte();
        y = (n1 << 8) + _serialPort.ReadByte();
        if (y < 0 || y > 1024)
        {
            g.DrawString("Port-Error: y out of range", fn, br, 250, 30);
            n1 = _serialPort.ReadByte();
            y = 65;
        }
    }
    catch (TimeoutException)
    {
        y = 100;
        g.DrawString("TimeoutException1", fn, br, 50, 70);
    }
    yvon = yY(y);

```

```

x += dx;
while (x <= xmax)
{
    xbis = xX(x);
    try
    {
        n1 = _serialPort.ReadByte();
        y = (n1 << 8) + _serialPort.ReadByte();
    }
    catch (TimeoutException)
    {
        y = 100;
        g.DrawString("TimeoutException1", fn, br, 50, 70);
    }
    if (n == delay)
    {
        ybis = yY(y);
        g.DrawLine(pen, xvon, yvon, xbis, ybis);
        x += dx; xvon = xbis; yvon = ybis;
    }
    n = (n % delay) + 1;
}
}

public void Koordinatensystem(Graphics g, double dx, double dy, double hier_yAchse)
{
    Font fn = new Font("Verdana", 12);
    Brush br = new SolidBrush(Color.DimGray);
    Pen pen = new Pen(Color.Red, 1/*Strichdicke*/);

    Point[] pxA =
    {
        new Point(xX(xmax),yY(0)),
        new Point(xX(xmax) - 18,yY(0) - 9),
        new Point(xX(xmax) - 18,yY(0) + 9),
        new Point(xX(xmax),yY(0))
    };
    int h = menuStrip1.Height;
    Point[] pyA =
    {
        new Point(xX(hier_yAchse) + 9, yY(ymax) + 18 + h),
        new Point(xX(hier_yAchse), yY(ymax)-2 + h),
        new Point(xX(hier_yAchse) - 9, yY(ymax) + 18 + h),
        new Point(xX(hier_yAchse) + 9, yY(ymax) + 18 + h)
    };

    g.DrawLine(pen, xX(xmin), yY(0), xX(xmax), yY(0)); //x-Achse
    g.FillPolygon(br, pxA); //Pfeil der x-Achse

    g.DrawLine(pen, xX(hier_yAchse), yY(ymin), xX(hier_yAchse), yY(ymax)); //y -Achse
    g.FillPolygon(br, pyA); //Pfeil der y-Achse
}

```

```

double x = dx;
while (x < xmax)
{
    g.DrawString(x.ToString("#.#"), fn, br, xX(x), yY(0) + 3);
    g.DrawRectangle(pen, xX(x), yY(0), 1, 3);
    x += dx;
}
x = -dx;
while (x > xmin)
{
    g.DrawString(x.ToString(), fn, br, xX(x), yY(0) + 3);
    g.DrawRectangle(pen, xX(x), yY(0), 1, 3);
    x -= dx;
}
x = dy;
while (x < ymax)
{
    g.DrawString(x.ToString("0.#####"), fn, br, xX(hier_yAchse) + 5, yY(x) );
    g.DrawRectangle(pen, xX(hier_yAchse), yY(x), 3, 1);
    x += dy;
}
x = -dy;
while (x > ymin)
{
    g.DrawString(x.ToString("0.#####"), fn, br, xX(hier_yAchse) + 5, yY(x));
    g.DrawRectangle(pen, xX(hier_yAchse), yY(x), 3, 1);
    x -= dy;
}
}

private void ClosePort()
{
    if (_serialPort != null)
    {
        _serialPort.Close();
        _serialPort = null;
        com_is_open = false;
    }
}

private void OpenPort()
{
    _serialPort = new SerialPort("COM3", 115200, Parity.None, 8, StopBits.One);
    if (_serialPort != null)
    {
        _serialPort.ReadTimeout = 50;
        _serialPort.Open();
        _serialPort.DiscardInBuffer();
        com_is_open = true;
    }
}

```

```

    }
}

private void Form1_Paint(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics;
    Font fn = new Font("Verdana", 12);
    Brush br = new SolidBrush(Color.DimGray);
    Koordinatensystem(g, xmax / 5, ymax / 5, yA);
    if (com_is_open)
    {
        g.DrawString("measurement: busy", fn, br, 50, 30);
        Graph2(g, Color.Blue, i_wait);
        g.DrawString("ready", fn, br, 50, 50);
    }
    else
    {
        g.DrawString("com_is_open = false;", fn, br, 50, 30);
        g.DrawString("delay = " + i_wait.ToString(), fn, br, 50, 50);
    }
    ClosePort();
}

private void Form1_SizeChanged(object sender, EventArgs e)
{
    ClosePort();
    xStreckfaktor = this.ClientRectangle.Width / (xmax - xmin);
    yStreckfaktor = this.ClientRectangle.Height / (ymax - ymin);
    this.Refresh();
}

private void Menu_close_Click(object sender, EventArgs e)
{
    ClosePort();
    this.Close();
}

private void messenToolStripMenuItem_Click(object sender, EventArgs e)
{
    ClosePort();
    OpenPort();
    this.Refresh();
}

private void delayToolStripMenuItem_Click(object sender, EventArgs e)
{
    ClosePort();
    i_wait += 1;
    this.Refresh();
}

```



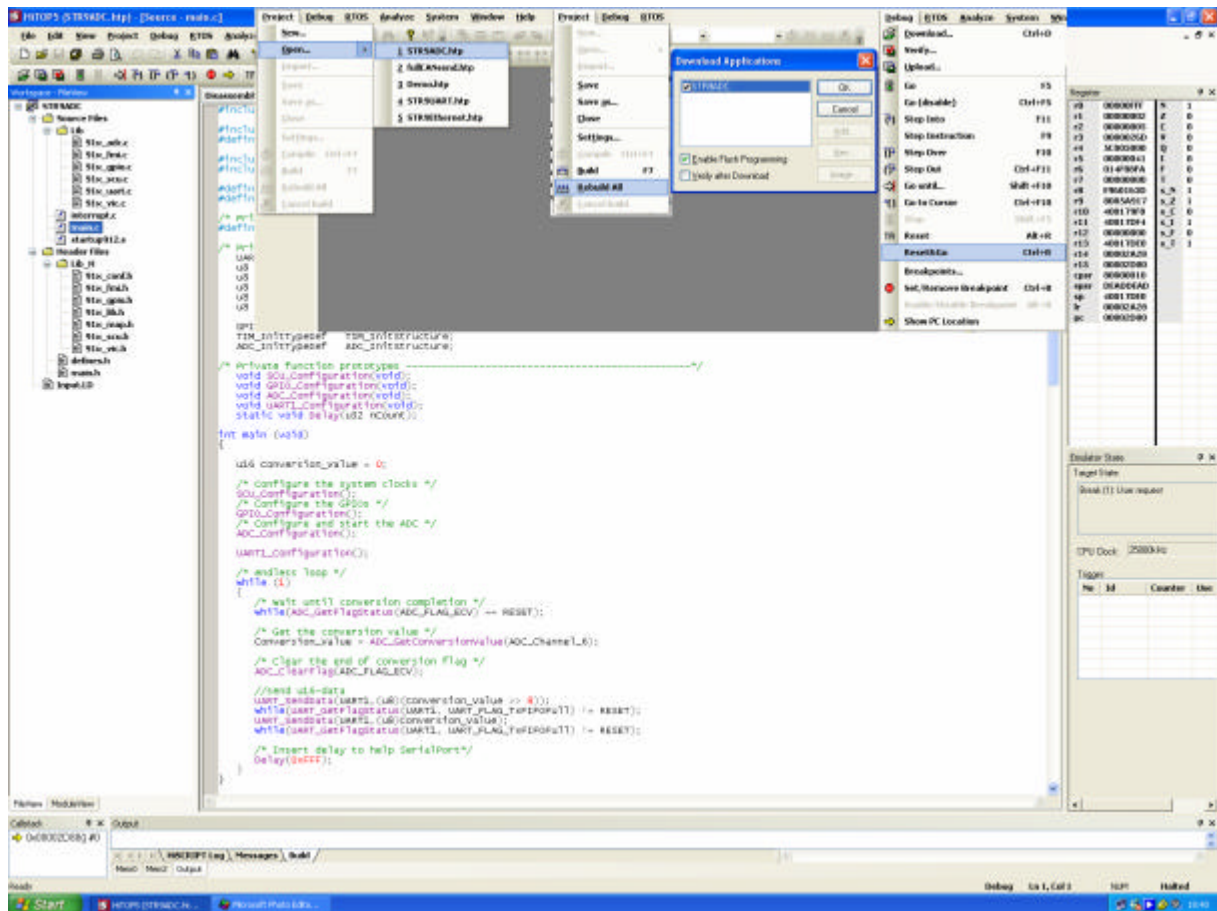
```

private void delayToolStripMenuItem1_Click(object sender, EventArgs e)
{
    ClosePort();
    i_wait -= 1;
    if (i_wait <= 0) i_wait = 1;
    this.Refresh();
}
}
}
}

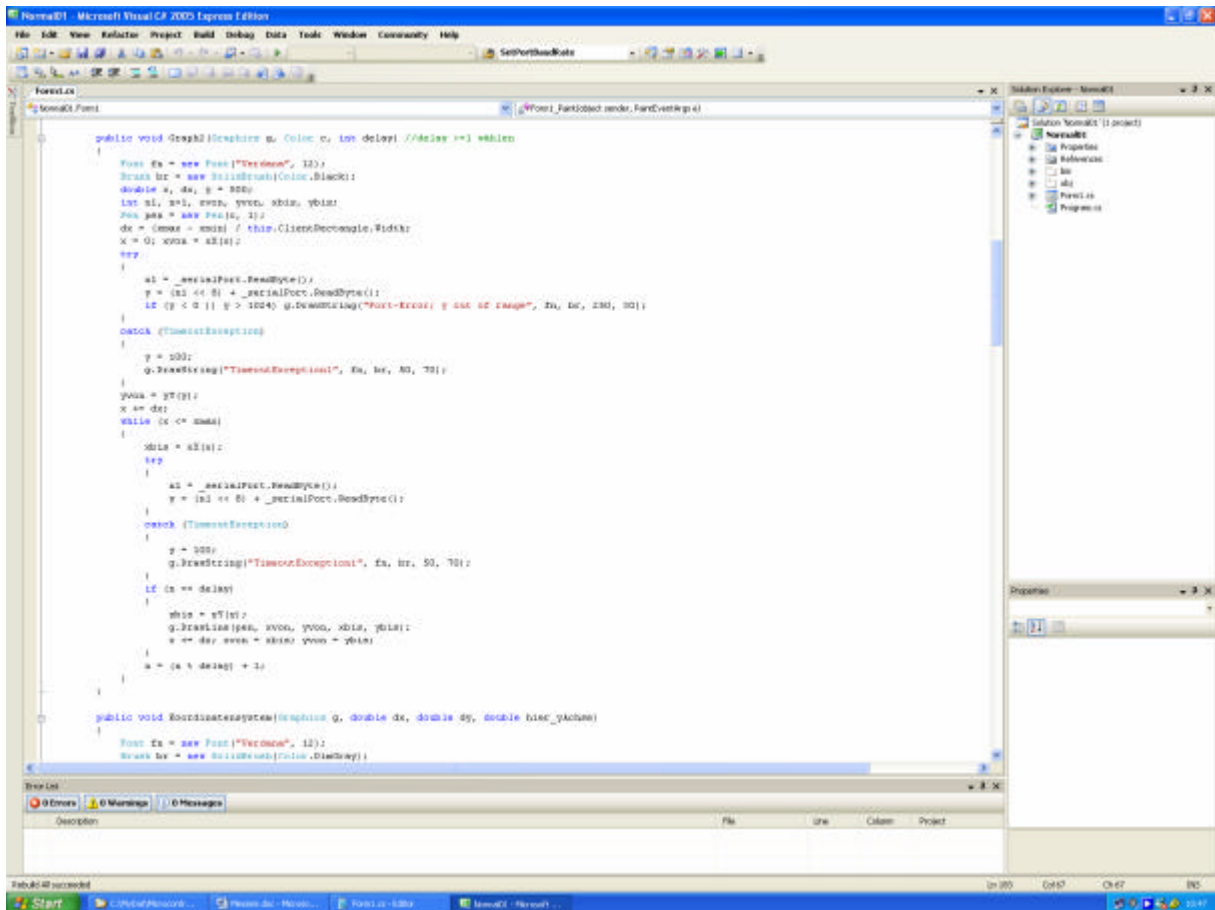
```

Ich habe dabei bewusst auf Thread's verzichtet, um wenig an Geschwindigkeit zu verlieren.

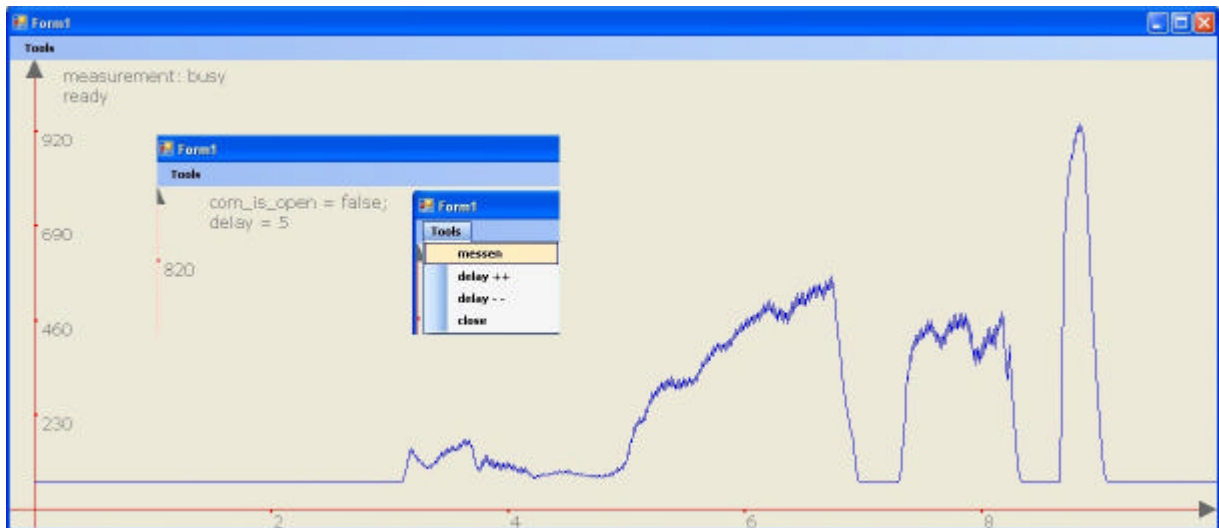
Als erstes startet man HiTOP5 wie das Bild zeigt.



Dann startet man Visual C# 2005.

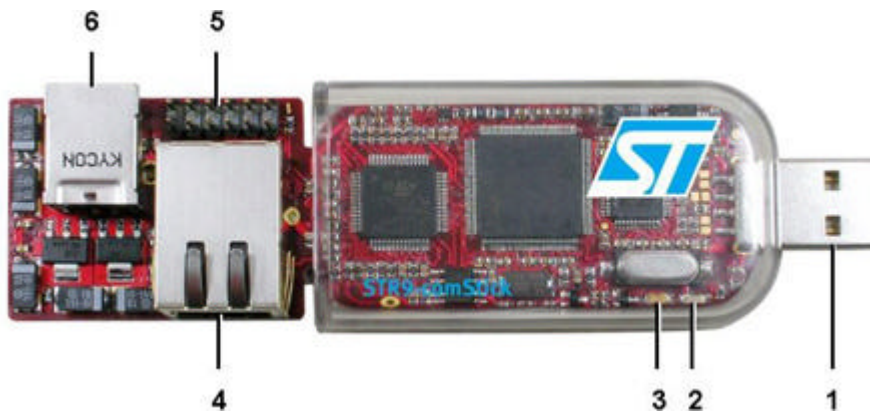


und erhält z.B.



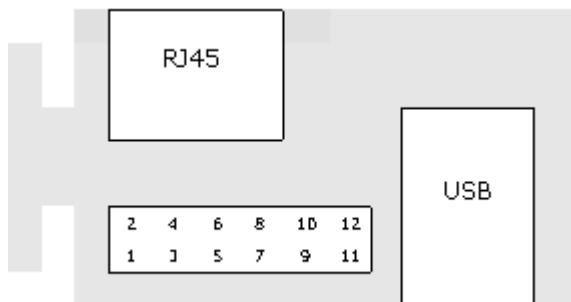
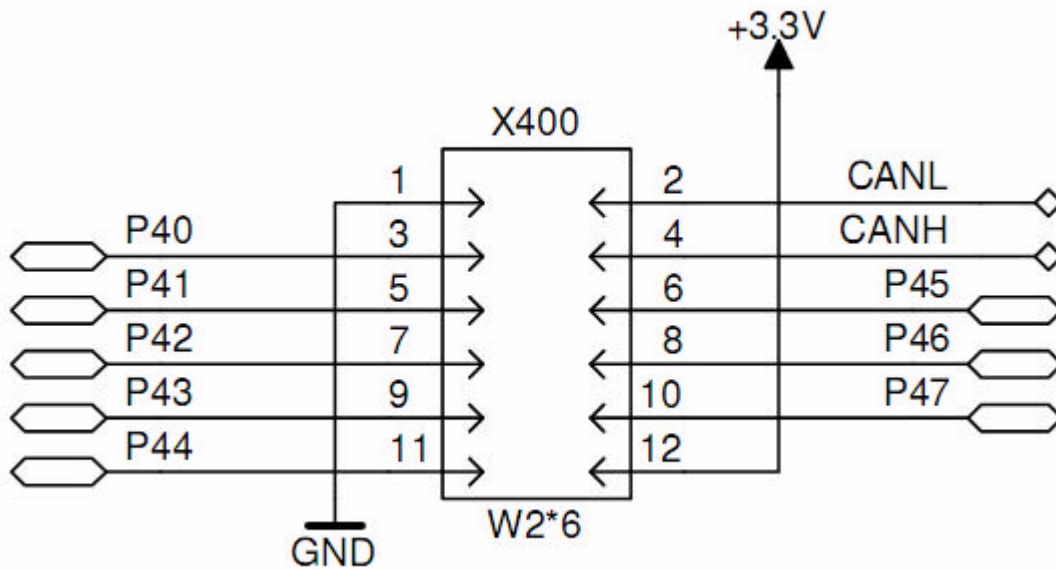
Das Muster habe ich einfach dadurch erhalten, dass ich mit dem Finger die Stiftleiste (5 Extension Header) beim LAN-Anschluss des STR9-ComSticks berührt habe. Über die Haut gelangen dann zufällig elektrische Signale an Pin8. Berühren Sie bei dieser

Methode wegen körpereigenen Ladungen zuerst das Metallgehäuse des LAN-Anschlusses.



- 1 USB Target Connector
- 2 COM LED (green)
- 3 USER LED (red)
- 4 RJ45 Ethernet Connector
- 5 Extension Header
- 6 USB Connector Type B

Die analogen Signale zwischen 0V und 1,85V führt man über Pin1(Masse) und Pin8 dem Stick zu. $U=+1,85V$ an Pin8 wird in den Conversion_Value-Wert 1023 umgewandelt.



Falls Ihr COM-Port nicht COM3 ist müssen Sie

```
private void OpenPort()
{
    _serialPort = new SerialPort("COM3", 115200, Parity.None, 8, StopBits.One);
    if (_serialPort != null)
    {
        _serialPort.ReadTimeout = 50;
        _serialPort.Open();
        _serialPort.DiscardInBuffer();
        com_is_open = true;
    }
}
```

ändern oder einen neuen Menüpunkt im Programm hinzufügen um dies dynamisch zu ermöglichen.

Zum weiteren Ausprobieren eignen sich Stellen wie

```
/* Insert delay to help SerialPort*/  
Delay(0xFF);
```

oder

```
_serialPort.ReadTimeout = 50;
```

oder

```
_serialPort.DiscardInBuffer();
```

Viel Spaß: Edgar Marx

edgarmarx@t-online.de