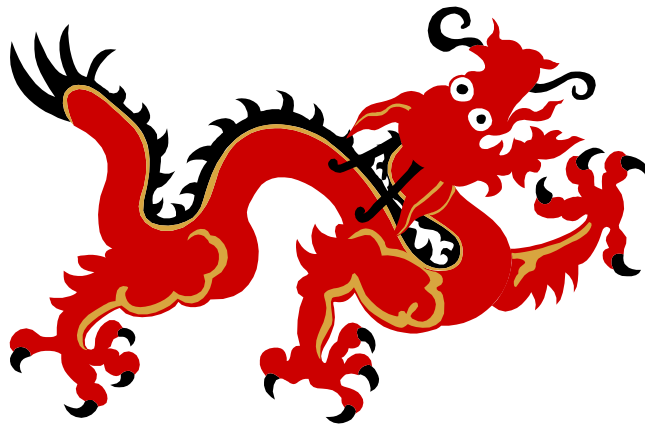

fischertechnik ROBO TX Controller

umFish50.DLL

Documentation of Version 5.0

Ulrich Müller



Inhaltsverzeichnis

Overview	3
ROBO TX Controller	3
Common	3
Sensor Inputs	3
Actor Outputs	3
Installation	3
umFish50.DLL - List of Functions	4
Using umFish50.DLL	6
Common	6
Robo TX Controller Test	7
C#	10
VB.NET	11

Copyright Ulrich Müller. Dokumentname : umFish50e.doc. Druckdatum : 18.07.2010

Overview

ROBO TX Controller

Common

One Master at a time. Connected to the Master Extension Connectors : 0 – 8 Extensions (same hardware, but configured as Extension). Text display with some lines of different size. Near the display 2 buttons for operation the controller. Connection via USB (mini plug) alternatively Bluetooth.

Sensor Inputs

The controller comes with 8 universal inputs, which must be configured for their special use. In addition there are 4 counter inputs to be use as quick counters for the encoder motors and as normal digital inputs.

Universal Inputs

D5K : Digital 5 kOhm (switch, reed contact, photo transistor)

D10V : Digital 10 V (trail sensor ; cable red/green to plus/ground, yellow/blue to two different universal inputs)

A5K : Analog 5 kOhm (NTC, photo resistor, potentiometer)

A10V : Analog 10 V (color sensor, voltage)

Distance : Distance sensor : 3 – 400 cm. Cable : red to 9V, green to ground, black to an universal input.

Actor Outputs

The controller owns 8 outputs with one line and ground O Outputs (lamps, beeper, magnet, motors with one direction). They can be combined to 4 motor connectors M Outputs (motors XS, XM, Encoder, Power).

Installation

The simple way : an installed **Robo Pro** v2.4. and higher (ftMscLib.DLL in \RoboPro. Alternatively : Download from fischertechnik.de | Compting | Downloads "**PC Programming ROBO-TXC ...**". It contains the driver for the TX Controller, ftMscLib.DLL (used by umFish50.DLL) and the test tool RoboTxTest.exe.

Download umFish50.ZIP with umFish50.DLL and the declaration for using umFish.DLL in several different programming languages. ftMscLib.DLL and umFish50.DLL must be placed in \Windows ..\System32, under Windows 7/64bit in ...\SysWOW64.

umFish50.DLL - List of Functions

umFish50.DLL is thought to be used for construction of own language specific class libraries like the FishFace Libraries. umFish50.DLL supports the ROBO TX Controller (and only this) connected via USB or Bluetooth in the "online" mode (program is running on the PC).

umFish50.DLL combines the function of ftMscLib.DLL functions for a more simple use :

- MultiMediaTimer
 - Control of the M-Output and the C-Inputs. Stop of a motor when reached its goal, defined by the counter.
 - Cancel running motor operations with the Esc-Key.
- Combining ftMscLib.DLL function to sensor/actor oriented methods.
- Configuration of the universal input while first use.
- Inserting of Sleeps in time critical program sections.

All methods return on failure the same code txError.

Parameters

ctrlId : Controller Id (0 - 8)

inpNr : Number of an Universal Input (0 - 7 (11))

cntNr : Number of an Counter Input (0 - 3)

outNr : Number of an O Output (0 - 7)

motNr : Number of a M Output (0 - 3)

OnOff : Power for a O Output (0 - 512)

speed : Speed for a M Output (0 - 512)

Common Constants

ctxError Return Value : method failed

ctxOff, ctxOn, ctxLeft, ctxRight : Switching M / O Outputs

ctxFull 512, ctxHalf 444 : Power to be used

ctxMain, ctxExt1 : Name of Master / Extension (0 / 8)

ctxI1 ... : Number of a Universal Input (0 - 8, - 11, if C Inputs are used as switches)

ctxC1 ... : Number of a Counter Input (0 - 3)

ctxM1 ... : Number of a M Output (0 - 3)

ctxO1 ... ; Number of an O Output (0 - 7)

Connection to the TX Controller

int **txOpenController**(ComNr)
 Connection to the used Master TX Controller
 ComNr : Number of the COM connection. Can be found with RoboTxTest.EXE
 or the Windows device manager (TX Controller must be connected).

int **txCloseController**()
 Close the existing connection to the TX Controller

Universal Inputs

Universal inputs are configured in concordance to the calling method on time of the first access after an OpenController

- int **txGetAnalog**(ctrlId, inpNr)
Read of the actual analogous value from the addressed universal input configured to A5K (NTC, photo resistor, potentiometer)
- int **txGetDistance**(ctrlId, inpNr)
Read of the actual distance value from the addressed universal input configured to Dist (distance sensor)
- int **txGetInput**(ctrlId, inpNr)
Read of the actual digital state of the addressed universal input configured to D5K (switch, reed contact, photo transistor)
- int **txGetTrack**(ctrlId, inpNr)
Read of the state of a "half" track of a trail sensor. TRUE(1) means on a black trail configured to D10V (trail sensor)
- int **txGetVoltage**(ctrlId, inpNr)
Read of the actual voltage value from the addressed universal input configured to A10V (color sensor, voltage)

Counter Inputs

- int **txGetCounter**(ctrlId, cntNr)
Read of the actual counter value of the addressed counter
- int **txClearCounter**(ctrlId, cntNr)
Reset the addressed counter to zero

M / O Outputs

An RobMotor is an real encoder motor with end switch on an universal input with the same number the motor is connected to M-Output. Alternatively an normal motor with impulse switch at the C-Input with the same number as the motor is connected to. End switch like encoder motor.

- int **txSetLamp**(ctrlId, outNr, OnOff)
Switch of an O-Output to the power value of OnOff(0 (off) to 512)
- int **txSetMotor**(ctrlId, motNr, direction, speed)
Switch of an M-Output to direction left, right, off and a speed value (0 - 512)
- int **txStartMotor**(ctrlId, motNr, direction, speed, iCount, robMode)
Switch of an RobMotor to direction left, right, off an a speed value (0 - 512) for counting iCount impulses. The motor is switched off asynchronous, if the number of impulses is reached (or exceeded) or the end switch is set.
- int **txIsMotorReady**(ctrlId, motNr)
State of a motor startet with txStartMotor. Returns -1 if running, number of impulses(0 - ...) exceeding iCount if stopped.

System Functions

- void **txSleep**(mSek)
Halt of the actual Thread for noted milli seconds
- int **txEsc**()
State of the ESC key (0 not pressed)
- int **txGetTickCount**()
Read the actual tick counts (milli seconds since midnight)

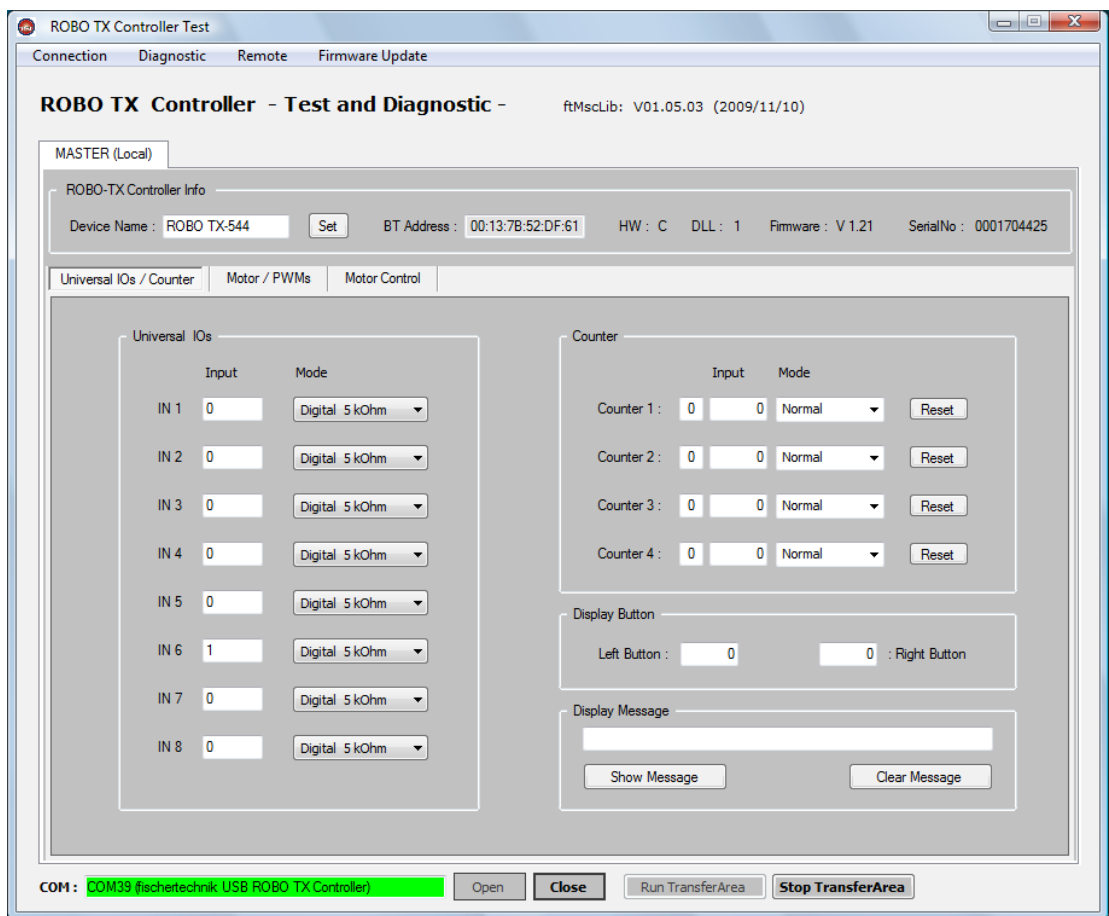
Using umFish50.DLL

Common

umFish50.DLL is delivered as compiled DLL and as VC++ project (www.ftcomputing.de/zip/umFish50.zip). Additionally there are some "H-Files" (declarations, classes) for use in different programming languages. They are supplemented by a typical HelloRobo program. Note : The COM connection number of the TX Controller may differ from the samples. Then it must be changed.

umFish50.DLL first is a base for the development of own libraries. Direct using may be complicated because of not DoEvents (interrupt to process Windows messages) is contained.

Robo TX Controller Test



The Test Panel delivered with the **PC Programming ROBO-TXC ...** from
www.fischertechnik.de | Computing | Downloads.

C#

```
using System;
using System.Collections.Generic;
using System.Text;
using um = umFish50.umFish50;

namespace HelloFish50CS {
    class Program {
        static void Main(string[] args) {
            int res = 0;
            Console.WriteLine("HelloFish50-Test startet on COM39");
            res = um.txOpenController(39);
            if ((uint)res == um.ctxError) {
                Console.WriteLine("OpenError");
                Console.Read();
                return;
            }
            Console.WriteLine("--- For Blinking : I3-Switch");
            while (um.txGetInput(um.ctxMain, um.ctxI3) == 0)
                um.txSleep(44);
            Console.WriteLine("Bei der Arbeit");
            for (int i = 0; i < 10; i++) {
                Console.WriteLine("Round : " + i.ToString());
                um.txSetLamp(um.ctxMain, um.ctxO5, um.ctxFull);
                um.txSetLamp(um.ctxMain, um.ctxO6, um.ctxOff);
                um.txSleep(444);
                um.txSetLamp(um.ctxMain, um.ctxO5, um.ctxOff);
                um.txSetLamp(um.ctxMain, um.ctxO6, um.ctxFull);
                um.txSleep(333);
            }
            um.txSetLamp(um.ctxMain, um.ctxO5, um.ctxOff);
            um.txSetLamp(um.ctxMain, um.ctxO6, um.ctxOff);
            Console.WriteLine("Done");
            um.txCloseController();
            Console.WriteLine("--- FINIS : Enter Key ---");
            Console.Read();
        }
    }
}
```

The original project is constructed with C# 2005, it can easily be converted to 2008 and 2010. It will run on 32bit system without problems. On Windows 7/64bit it only will work with a new C# 2010, in which you can copy this code.

The TX Controller are added "on the fly" an switch on I3 and lamps on O5 and O6. The TX Controller itself is connected via COM39 (you can see this on the Test Panel left bottom corner and change it with the device manager of Windows).

VB.NET

```
Imports cs = System.Console
Imports um = HelloFish50VB.umFish50

Module HelloFish50

    Sub Main()
        Dim res As Int32 = 0
        cs.WriteLine("HelloFish50-Test an COM4 gestartet")
        res = um.txOpenController(4)
        If res = um.ctxError Then
            cs.WriteLine("OpenError")
            cs.Read()
            Return
        End If
        cs.WriteLine("--- Zum Blinken : I3-Taster")
        Do
            um.txSleep(44)
        Loop While um.txGetInput(um.ctxMain, um.ctxI3) = 0
        cs.WriteLine("Bei der Arbeit")
        For i As Integer = 1 To 10
            cs.WriteLine("Runde : " & i)
            um.txSetLamp(um.ctxMain, um.ctxO5, um.ctxFull)
            um.txSetLamp(um.ctxMain, um.ctxO6, um.ctxOff)
            um.txSleep(444)
            um.txSetLamp(um.ctxMain, um.ctxO5, um.ctxOff)
            um.txSetLamp(um.ctxMain, um.ctxO6, um.ctxFull)
            um.txSleep(333)
        Next
        um.txSetLamp(um.ctxMain, um.ctxO6, um.ctxOff)
        cs.WriteLine("Das war's")
        um.txCloseController()
        cs.WriteLine("--- FINIS : Enter-Taste ---")
        cs.Read()
    End Sub

End Module
```

The original project is constructed with VB 2005, it can easily be converted to 2008 and 2010. It will run on 32bit system without problems. On Windows 7/64bit it only will work with a new VB 2010, in which you can copy this code.

The TX Controller are added "on the fly" an switch on I3 and lamps on O5 and O6. The TX Controller itself is connected via COM4 (you can see this on the Test Panel left bottom corner and change it with the device manager of Windows).