

# RoboTXkey App



Table of Contents

USB-Driver Installation .....3

ROBO TX Controller firmware Update .....3

Write and compile a C program for the ROBO TX Controller .....5

Installing a program on the ROBO TX Controller .....6

Installing the RoboTXkey App on an Android-Device.....7

Pairing devices and using the RoboTXkey App .....7

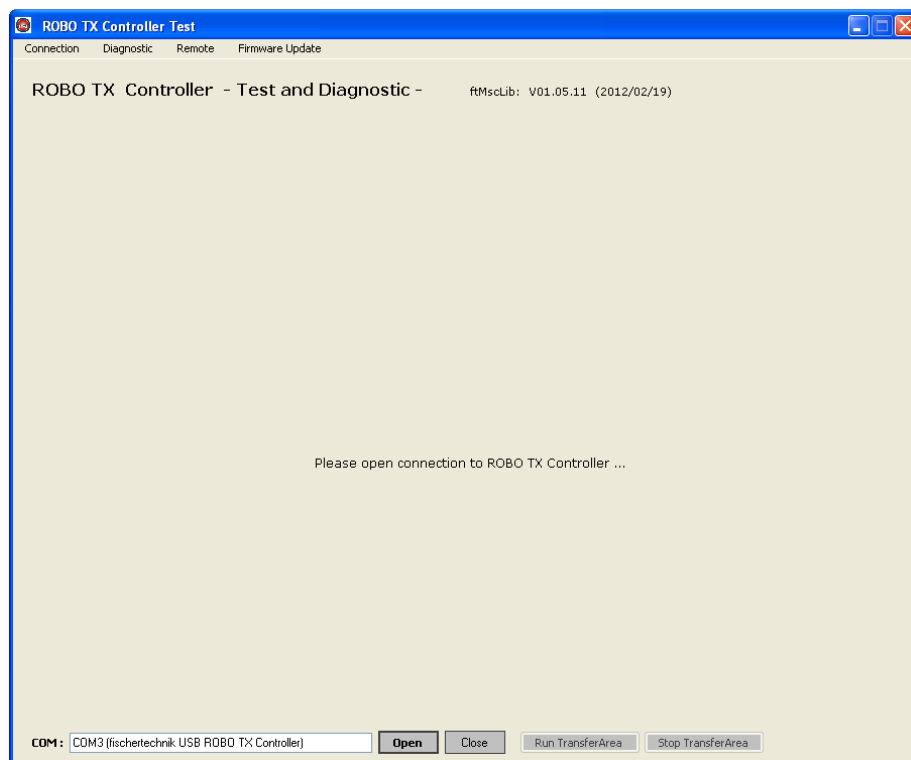
## USB-Driver Installation

Connect your Fischertechnik Model via USB cable with your PC. To work with the ROBO TX Controller on a PC, a USB-Driver is required. If not already installed, please have a look at Fischertechnik's web site ([www.fischertechnik.de](http://www.fischertechnik.de)). Go to *Downloads* and *Computing*. You will find USB-driver-installation guidelines for various Windows Systems. Please proceed accordingly.

## ROBO TX Controller firmware Update

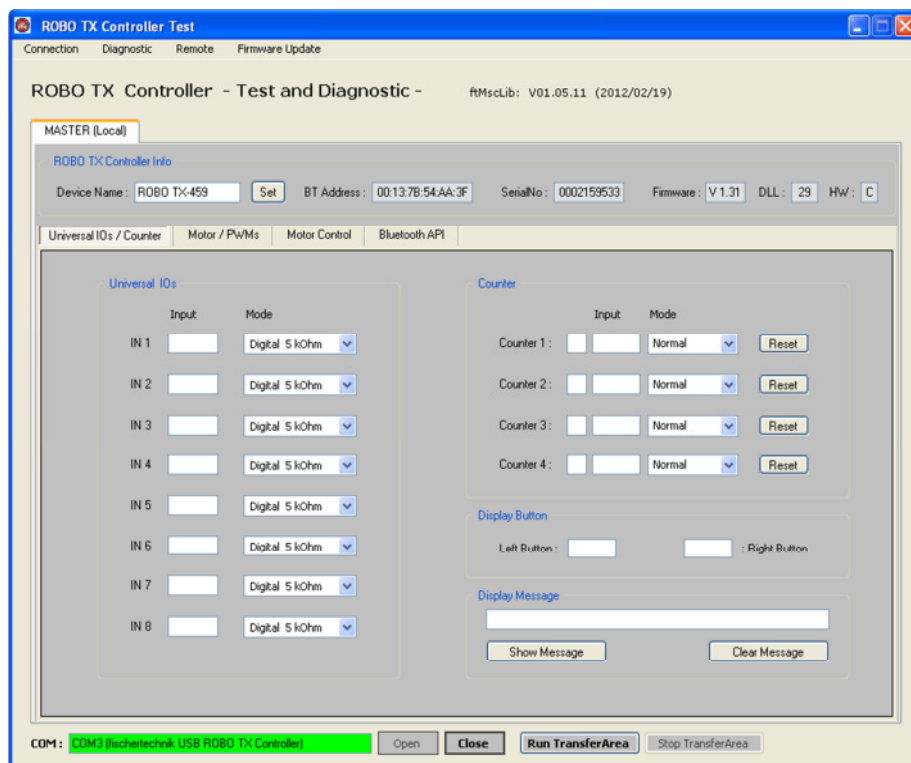
On the ROBO TX Controller firmware version 1.3.2 or higher is required. If you already familiar with firmware updates using ROBO Pro, skip this section and perform the firmware update with ROBO Pro.

If you aren't familiar with this topic yet, navigate to the folder *RoboTxTest* in the *Fischertechnik.zip* package and start the executable "RoboTxTest.exe" (see Figure 1).



**Figure 1:** RoboTxTest screenshot.

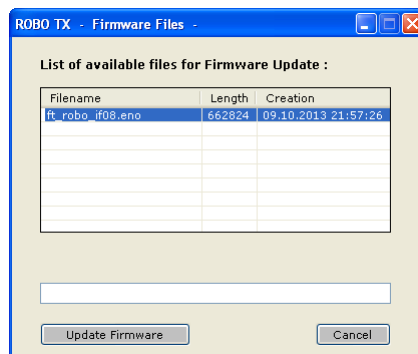
Click on "Open" at the bottom of the window. This will open the screen in Figure 2. (Please check, if a Fischertechnik Controller is connected to one of your PC's COM ports).



**Figure 2:** Screenshot of RoboTxTest with connection established.

With a connection up and running, navigate to the menu bar and press “Firmware Update” at the top of the window. A dialog will pop up and the folder with the new firmware needs to be selected.

The window in Figure 3 will open. Select the latest firmware version and press “Update Firmware”. The firmware will be updated immediately.



**Figure 3:** Screenshot of the dialog to update the firmware.

With the updated firmware the ROBO TX Controller may be used with the RoboTXkey App.

## Write and compile a C program for the ROBO TX Controller

An example program written in C for the ROBO TX Controller is called “Example.c” and can be found in the folder *Programs\Source\Example*. It maps the buttons of the RoboTXkey App to the outputs of the controller (see Figure 4 and Figure 5).



**Figure 4:** Schematic mapping of buttons.

The example C program can easily be adapted to various requirements. To make changes to the program open the file “Example.c” and scroll down to the code block beginning at line 284. The code there should be the outline shown in Figure 5.

```

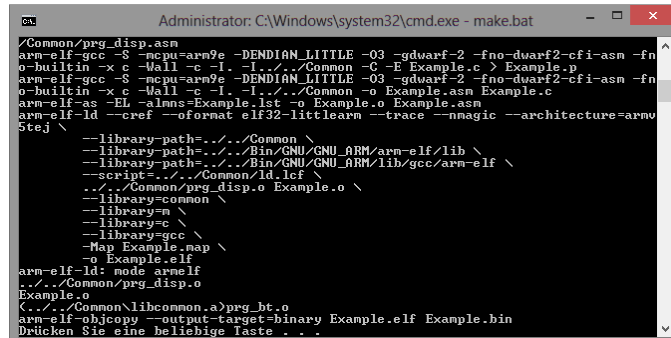
/* EXAMPLE */
/* */ if (keys[3] == 1) {
/* */     p_ta->output.duty[0] = DUTY_MAX;
/* */     p_ta->output.duty[1] = 0;
/* */ }
/* */ } else if (keys[5] == 1) {
/* */     p_ta->output.duty[0] = 0;
/* */     p_ta->output.duty[1] = DUTY_MAX;
/* */ }
/* */ } else {
/* */     p_ta->output.duty[0] = 0;
/* */     p_ta->output.duty[1] = 0;
/* */ }
/* */ ...
/* */
/* EXAMPLE */

```

**Figure 5:** Source code outline.

This part of the program file can be changed as required. (When looking at the C Code please remember that in C arrays have a lower bound of zero, while the soft keys in the RoboTXkey App start with ‘1’).

All changes to the C Program are optional. The C program can be used without any changes. It is just necessary to compile the program. Execute the batch file “make.bat” in the same folder. A window as shown in Figure 6 will open.



```

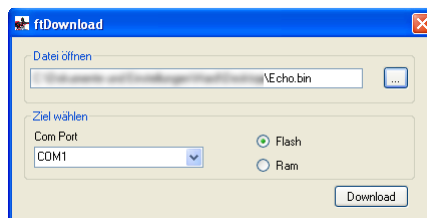
C:\Common\prg_disp.asm
arm-elf-gcc -S -mcpu=arm9e -DENDIAN_LITTLE -O3 -gdwarf-2 -fno-dwarf2-cfi-asm -fno-
o-builtin -x c -Wall -c -I. -I../Common -G -E Example.c > Example.p
arm-elf-gcc -S -mcpu=arm9e -DENDIAN_LITTLE -O3 -gdwarf-2 -fno-dwarf2-cfi-asm -fno-
o-builtin -x c -Wall -c -I. -I../Common -o Example.asm Example.c
arm-elf-as -EL -altns=Example.lst -o Example.o Example.asm
arm-elf-ld -cref -oformat elf32-littlearm --trace --nmagic --architecture=armv
Step) \
--library-path=../Common \
--library-path=../Bin/GNU/GNU_ARM/arm-elf/lib \
--script=../Common/ld.lcf \
../Common/prg_disp.o Example.o \
--library=common \
--library=c \
--library=gcc \
-Map Example.map \
-o Example.elf
arm-elf-ld: note: armelf
../Common/prg_disp.o
Example.o
C:\Common\libcommon.a)prg_bt.o
arm-elf-objcopy --output-target=binary Example.elf Example.bin
Drücken Sie eine beliebige Taste . . .
  
```

**Figure 6:** Screenshot of the output after compiling the program.

After successful compilation of the C program, the executable file can be loaded onto the ROBO TX Controller as described in the next section.

## Installing a program on the ROBO TX Controller

Once a program is successfully compiled, the generated binary file can be installed on the ROBO TX Controller. To do so, execute the “ftDownload.exe” from the folder *Downloader*. The window shown in Figure 7 will open. Select the binary file (\*.bin) of the compiled C program and check the respective COM Port (the COM Port connected to the controller). Then press “download” at the lower right corner of the window.



**Figure 7:** Screenshot of ftDownload right before downloading the program.

On the ROBO TX Controller, start the program. The RoboTXkey App is now able to interact with the controller.

## Installing the RoboTXkey App on an Android device

To install the RoboTXkey App on an Android-Device, scan the QR-Code in Figure 8 and download the app from Google Play.

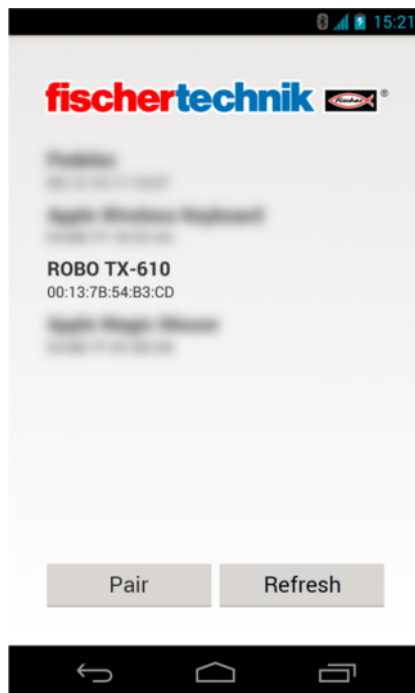


**Figure 8:** QR-Code to download the RoboTXkey App.

Alternatively, the “RoboTXkey.apk” can be available in the folder *Android*. It can be installed manually by using Android tools (adb.exe).

## Pairing devices and using the RoboTXkey App

When the RoboTXkey App is available on an Android device and a corresponding program is running on the ROBO TX Controller, everything is set up properly. Simply select the controller, wait until the connection gets established and start interacting with the controller (see Figure 9).



**Figure 9:** Screenshot of the RoboTXkey App.

If the controller is not listed in the app, the devices need to be paired. To do so, press “Pair” at the lower left corner of the app and select the controller to pair with. The default security code for Bluetooth connections is “1234”.