GARZ FRICKE

Embedded Systems

Garz & Fricke

Embedded Computer Systems

Windows Embedded Compact 7 · User Manual

Windows CE7 10.1r442 · i.MX6Q
Built on October 20, 2017
Manual revision 452

freescale Enabled

Windows Embedded Compact

Zuverlässige Qualität
Made in Germany

Important hints

Thank you very much for purchasing a Garz & Fricke product. Our products are dedicated to professional use and therefore we suppose extended technical knowledge and practice in working with such products.

The information in this manual is subject to technical changes, particularly as a result of continuous product upgrades. Thus this manual only reflects the technical status of the products at the time of printing. Before design-in the device into your or your customer's product, please verify that this document and the therein described specification is the latest revision and matches to the PCB version. We highly recommend contacting our technical sales team priorto any activity of that kind. A good way getting the latest information is to check the release notes of each product and/or service. Please refer to the chapter [► 10 Related documents and online support].

The attached documentation does not entail any guarantee on the part of Garz & Fricke GmbH with respect to technical processes described in the manual or any product characteristics set out in the manual. We do not accept any liability for any printing errors or other inaccuracies in the manual unless it can be proven that we are aware of such errors or inaccuracies or that we are unaware of these as a result of gross negligence and Garz & Fricke has failed to eliminate these errors or inaccuracies for this reason. Garz & Fricke GmbH expressly informs that this manual only contains a general description of technical processes and instructions which may not be applicable in every individual case. In cases of doubt, please contact our technical sales team.

In no event, Garz & Fricke is liable for any direct, indirect, special, incidental or consequential damages arising out of use or resulting from non-compliancy of therein conditions and precautions, even if advised of the possibility of such damages.

Before using a device covered by this document, please carefully read the related hardware manual and the quick guide, which contain important instructions and hints for connectors and setup.

Embedded systems are complex and sensitive electronic products. Please act carefully and ensure that only qualified personnel will handle and use the device at the stage of development. In the event of damage to the device caused by failure to observe the hints in this manual and on the device (especially the safety instructions), Garz & Fricke shall not be required to honour the warranty even during the warranty period and shall be exempted from the statutory accident liability obligation. Attempting to repair or modify the product also voids all warranty claims.

Before contacting the Garz & Fricke support team, please try to help yourself by the means of this manual or any other documentation provided by Garz & Fricke or the related websites. If this does not help at all, please feel free to contact us or our partners as listed below. Our technicians and engineers will be glad to support you. Please note that beyond the support hours included in the Starter Kit, various support packages are available. To keep the pure product cost at a reasonable level, we have to charge support and consulting services per effort.

Shipping address:
Garz & Fricke GmbH
Tempowerkring 2
21079 Hamburg
Germany

Support contact:
Phone +49 (0) 40 / 791 899 - 30
Fax +49 (0) 40 / 791 899 - 39
Email ► support@garz-fricke.com
URL ► www.garz-fricke.com

Contents

1 Introduction

Garz & Fricke systems based on **Freescale i.MX6** can be used with an adapted version of Microsoft Windows Embedded Compact, an operating system for embedded and mobile devices from Microsoft. Windows Embedded Compact – formerly known as Windows CE – comes with different licenses and feature sets and is customized to the Garz & Fricke devices. Usually a Board Support Package (BSP) from a third party is used as base for the port.

A Garz & Fricke device normally comes with a pre-installed Garz & Fricke operating system.

This manual contains information about the usage of the Garz & Fricke Windows Embedded Compact operating system for **i.MX6**, as well as the build process and the integration of custom software components. The Software Development Kit (SDK) can be downloaded from the Garz & Fricke support server:

- http://support.garz-fricke.com/projects/Santaro/WindowsCE7/

It should include all dependencies needed to build Software for the Garz & Fricke device, also containing the Garz & Fricke Application Programming Interfaces (APIs) needed to access the hardware components.

Please note that the OS development at Garz & Fricke is always in progress. Thus, there are new releases of the System at irregular intervals. Due to differences between the various platforms and versions, a separate manual is available for every platform/version. To avoid confusion, the version number of the manual exactly matches the BSP version number.

In addition to this manual, please also refer to the dedicated hardware manuals which can be found on the Garz & Fricke website as well.

Beyond the OS each platform needs its corresponding bootloader like E-Boot, RedBoot, U-Boot or Flash-N-Go-Boot. Usually the sources of the bootloaders are not provided by the Garz & Fricke BSPs. The bootloader is responsible for loading the kernel of the desired operating system into memory and provide the kernel with information it needs to work correctly. In most cases it holds possibilities to configure the target system.

2 Overview

A Garz & Fricke Windows Embedded System generally consists of six basic components:

- Flash-N-Go Bootloader
  A Garz & Fricke bootloader
- Flash-N-Go System
  An alternative linux based os for configuring the device
- Eboot Bootloader
  A second bootloader, should not be used by customers
- Operating System
  The target operating system Windows Embedded Compact 7
- Registry
  The Windows Embedded Compact 7 registry partition
- Device Configuration
  A persistent stored xml file

These software components are usually installed on separate partitions on the backing storage of the embedded system.

2.1 The partition layout

As already stated in chapter [► 2 Overview], the different components of the embedded WinCE system are stored in different partitons of the backing-storage. The backing-storage type of i.MX6 is eMMC. In addition to the partitions for the basic WinCE components there may be some more partitions depending on the system configuration.

The partition layout for the i.MX6 platform is:

Partition	File System	Contents
mmcblk0boot0	none	Flash-N-Go Boot bootloader image
mmcblk0boot1	FAT32	XML configuration parametes (config.xml) and touchscreen configuration (ts.conf)
mmcblk0p1	FAT32	Flash-N-Go System linux kernel image file, bootloader command file (boot-alt.cfg) and Flash-N-Go ramdisk file (root.cpio.gz)
mmcblk0p2	FAT32	Windows Embedded OS image (nk.bnx), the boot.cfg, eboot
mmcblk0p3	TexFAT32	Registry for Windows Embedded Compact
mmcblk0p5	TexFAT32	FlashDisk (free for use)

2.2 Flash-N-Go Boot - the bootloader

There are several bootloaders available for the various platforms. Embedded Systems do not have a PC-like BIOS. In most cases they are started from raw flash memory or an eMMC device. For this purpose, there are certain open source boot loaders available, like RedBoot, U-Boot or Barebox. Furthermore, Garz & Fricke provides its own bootloader called **Flash-N-Go Boot** for its newer platforms (e.g. SANTARO).

i.MX6 uses the bootloader **Flash-N-Go Boot**.

2.3 Flash-N-Go System - the management system

Newer Garz & Fricke devices are shipped with a separate small ramdisk-based Linux system called **Flash-N-Go System** which is installed in parallel to the main operating system. The purpose of Flash-N-Go is to provide the user a comfortable and secure update mechanism for the main operating system components.

Please read the corresponding Flash-N-Go manual for detailed information.

► http://support.garz-fricke.com/projects/Santaro/Flash-N-Go/FNGSystem

3 Operating System information

If you have chosen Microsoft Windows Embedded Compact as operating system, your Garz & Fricke device will be shipped with a licensed copy of the OS. The Starter Kits are always shipped with the license that covers all technologies, even though not all features are installed.

For an explanation of the difference between the different versions, please refer to

- ► Win Emb Compact 2013 AddlTerms 23.05.2013.pdf

3.1 OS release information

Usually, Garz & Fricke devices will be shipped with the latest OS release. In some cases, it might be possible that shortly after shipment a new OS will be released or – in case of larger productions – that a change within a running production is not recommended. We try to keep our existing customers informed about the latest OS revisions and will not ship devices with new OS without prior notice.

Nevertheless it is recommended to check the product download area of our website for the related product, at least before planning and developing a new application. The download area can be found under

- ► http://support.garz-fricke.com/products/Santaro/WindowsCE7/overview.html.

3.2 OS update information

Before updating an OS, it is recommended to contact the technical support of Garz & Fricke. Be aware that all failed modifications might cause a temporary breakdown, which can be repaired only by phone assistance. In worst case, the unit must be returned to Garz & Fricke for maintenance.

For detailed instructions how to update the OS, please refer to the chapter [► 6 Deploying the operating system to the target].

4 Basic Features

This Chapter describes the basic features of the Garz & Fricke Embedded Systems devices. The boot processs will be explained as well as the Touch functionality in more detail and the common interfaces that may be used.

4.1 Bootup

Irrespective of the used operating system (Windows Embedded, Linux or Android) its image file usually is stored in the non volatile memory of the device. This might be a NAND-Flash or an EMMC. Moreover an additional **Flash-N-Go System** image is stored in a separate partition, see chapter [► 2 Overview] for details about the software components. After turning on the device the bootloader **Flash-N-Go Boot** usually starts eboot which in turn loads the taget os image file into the RAM and executes it. Note that loading eboot inbetween is just for architectural reasons. It should not be necessary for customers to touch eboot in any case. Alternatively it is possible to load Flash-N-Go System. You can use the **bootselect** tool to select the desired operating system, see chapter [► 5.2.1 bootselect.exe]. For launching **Flash-N-Go System** on next boot you can set up the following command:

```
\> bootselect alternative
```

The non-headless devices are usually equipped with a display and touch combination. The touches enable a modern user interface and easy operation. There are basically two touch technologies: resitive and capacitive touches.

4.1.1 Resistive Touch

A resitive touch panel works by pressure-sensitive touch sensing. The touch works by measuring the resisantce of two separated layers. The layers therefore need to be flexible and are almost always made of plasics. By measuring the resistance alternating in the horizontal and vertical axes one pressure point can be determined.

Because of that measurenment procedure resistive touchscreens ave usaually only single touch functionality. All resistive touches of the Garz & Fricke devices only support one touch point.

The resitive touch works also best when the touch contact is pointy. Therfore it is often optimized to be used by a stylus. Wider touch contact on the other hand is not as precise.

Another property of the resitive touch is the comparably high pressure that is necessary to trigger the touch. Espceially when the touch point is wider (like a finger tip) high pressure is necessary.

Since the resistive properties differ for each touchscreen it is neccessary to calibrate each touch screen individually. A common calibration for all touch screens will not work.

The touch calibration will be executed on first boot and can be repeated using the **touchc** application executed for example via TELNET, see chapter [► 5.2.7 Touchc.exe]. To start the calibration, you can simply set up the following command:

```
\> touchc
```

4.1.2 Capacitive Touch

A capacitive touch panel works in the capacitance during touch of the touch panel. One capacitive touch technology is the Projected Capacitive Touch (PCT), where a glass layer is used as lense. The PCT is mechanically robust and easy to clean.

The capacitive technology also allows to mesure mutliple touch points at the same time. This enables a multi-touch experience. The different touchsceen controllers allow a various count of simultanous touch points. However, Windows Embedded Compact only allows 2 touch points at the same time.

Using multi touch one can implement gestures with multiple contacs, for example pinch and rotation gestures.

The capacitive touch properties do not vary as much as resitive touches and it is possible to use a common calibration data set for a single type of touch. Garz & Fricke devices therefore stores its touch calibration in an XML-based touch configuration file, that will be used among others to calibrate the touch.

In the current implementation the XML file is read once on boot and the data is mapped into the Registry. As this happens on every boot, the calibration cannot be altered in Windows persitently. To use a different touch calibration, a new XML touch configuration would be necessary.

To acquire a new or updated configuration the corresponding XML file needs to be installed in the **Flash-N-Go System** management system. See the **Flash-N-Go System** manual and the **xconfig import** chapter for more infromation regaring the installation of XML files.

4.1.3 Capacitive USB Touch

Currently there is an exception for the USB-based capacitive touches. Because of the touch driver architecture it is not possible to have a default calibration (from the XML) and a custom calibration. Thus it is neccessary for the customer to calibrate the USB-Touch manually.

The touch calibration will be executed on first boot and can be repeated using the **touchc** application executed for example via TELNET, see [▶ 5.2.7 Touchc.exe].

4.2 Network

Garz & Fricke devices usually come with one Ethernet port that support standard IEEE 802.3u. It also supports the standard TCP/IP connection.

Note that the ip adress in WindowsCE7 will not be shared with **Flash-N-Go System**. Only the MAC address is shared (again using the XML configuration file at bootup).

Thus to be able to use the devices Network capabilities it needs to be configured first. By default the IP settings are as follows:

DHCP	Deactivated
IPv4 Address	192.168.1.1
IPv4 Subnet mask	255.255.255.0
IPv4 Default Gateway	192.168.1.100

An ethernet cable directly connected to a development PC can be used to communicate with the device and to use is's services like FTP and TELNET.

To enable DHCP the graphical settings interface of WindowsCE7 can be used. Another way to enable DHCP is to activate the DHCP feature in the registry and restart the ethernet interface.

The following batch script does that.

```
toolbox reg set HKEY_LOCAL_MACHINE\Comm\ENET1\Parms\TcpIp "EnableDHCP" d 1
savereg -n
ndisconfig adapter rebind ENET1
ping -n 10 127.0.0.1
ipconfig
```

The script can also be executed via the autostart feature.

1. Copy the script above into a .bat file.
2. Create a FAT partition on a SD-Card or USB-Stick and create an autostart folder.
3. Copy the batch file in the autostart folder on the removable device.
4. Insert the removalble drive into the ebmedded device.

The script will be executed and DHCP should be activated afterwards. The retrieved ip adress will be displayed on the serial console. It will also be displayed after selecting the ethernet icon on the button right side of the task bar or via windows settings.

4.3 Display

Except for headless devices, the Garz & Fricke Embedded devices come with a display. Customers have various options to choose from, regarding resolution, size, brightness and other display properties. Currently only LCD

based LVDS displays are supported by the Windows Embedded Compact port. Feel free to contact the sales team at Garz & Fricke for more details and support.

The display that comes with the device needs settings configured in the device to work correctly. These contain the resolution, refresh rate, timing and other features that are necessessary to drive the display. The settings are usually installed during the production test at Garz & Fricke. To install new or updated XML settings the file needs to be installed in the **Flash-N-Go System** management system. See the **Flash-N-Go System** manual and the **xconfig import** chapter for more information regaring the installation of XML files.

Depending on the mechanical assembly it may be neccessary to rotate the display orientation. This can be done with the help of the **disp** tool or programmatically. The rotation parameters will be stored in the Registry and are persistent between consecutive restarts. Be aware that the boot logo is currently not rotated. As a workaround the customer can install a pre-rotated image.

The call of the **disp** tool to change the orientation counter-lockwise by 90 degree via TELNET looks like:

```
\> disp -r 1
```

see [► 5.2.2 Disp.exe] for more information.

to change the display rotation programmatically one can use the **changedisplaysettingsex** funtion of the windows gdi api. see ► https://msdn.microsoft.com/en-us/library/ee505713.aspx for more information.

4.3.1 Backlight and Dimming

As lcd-displays do not emit light directly a backlight is needed to show something on the screen. The backlight can be activated, deactivated and dimmed using pulse-width modulation (pwm). Similar to the display rotation the dimming can be done with a tool (**toolbox**) or programmatically.

The calls of the **toolbox** tool to set the backlight to half the brightness, off and full brightness:

```
\> toolbox backlight 127
\> toolbox backlight 0
\> toolbox backlight 255
```

See [► 5.2.6 toolbox] for more information.

The follwing C-code snippet shows a rough sketch how to change the backlight programmatiacally. First the backlight vaue is set in the registry, then the **BackLightLevelChangeEvent** is triggered.

```
#define REG_PATH              TEXT("ControlPanel\\Backlight")
#define BATT_LEVEL_SUBKEY     TEXT("BattBacklightLevel")
#define AC_LEVEL_SUBKEY       TEXT("ACBacklightLevel")
#define BKL_LEVEL_MIN         (0)     // off
#define BKL_LEVEL_MAX         (255)   // full on
#define EVENTNAME_BACKLIGHTLEVELCHANGEEVENT     L"BackLightLevelChangeEvent"

void setBacklight(DWORD dwExtLevel)
{
        HANDLE hEvent;
        HKEY hKey;

        if(     dwExtLevel < BKL_LEVEL_MIN || dwExtLevel > BKL_LEVEL_MAX)
                return;

        if(ERROR_SUCCESS != RegOpenKeyEx(HKEY_LOCAL_MACHINE, REG_PATH, 0, 0, &hKey))
                return;

        RegSetValueEx(hKey, BATT_LEVEL_SUBKEY, 0, REG_DWORD, (LPBYTE)&dwExtLevel,
            ↪ sizeof(DWORD));
        RegSetValueEx(hKey, AC_LEVEL_SUBKEY, 0, REG_DWORD, (LPBYTE)&dwExtLevel,
            ↪ sizeof(DWORD));
        RegCloseKey(hKey);
```

```
        // Signal backlight driver to update
        hEvent = CreateEvent(NULL,FALSE,FALSE,EVENTNAME_BACKLIGHTLEVELCHANGEEVENT);
        if(hEvent == NULL)
        return;

        SetEvent(hEvent);
        CloseHandle(hEvent);
}
```

4.4 Query version information

An installed operating system and all its software parts provide a Garz & Ficke version information. The Garz & Fricke windows release version information, which is usually the most important version number, can be found in a file `\Windows\OS-Image.ver` on the device. By using the **type** command of the command line the information can be queried. The following snippet shows the version information of release 8.4r391 of config 1.

```
\> type \Windows\OS-Image.ver
OS_MAJOR: 8
OS_MINOR: 4
OS_FIX: 0
OS_SVN: 2:391
OS_CONFIG: 1
Date: 13.10.2016
Time: 10:00
```

The other software parts provide their version information in the `HKEY_LOCAL_MACHINE\Software\` Registry subkeys. Following the querying the version information of the **Flash-N-Go System**, the **Flash-N-Go Boot** bootloader and the **EBoot** bootloader with the help of the toolbox tool.

```
\> toolbox reg show HKLM\Software\FnGSystem
Keys of HKEY_LOCAL_MACHINE\Software\FnGSystem:
Values of HKEY_LOCAL_MACHINE\Software\FnGSystem:
  VersionSVN = 5769 (0x00001689)
  VersionMinor = 0 (0x00000000)
  VersionMajor = 8 (0x00000008)
\> toolbox reg show HKLM\Software\FnGBoot
Keys of HKEY_LOCAL_MACHINE\Software\FnGBoot:
Values of HKEY_LOCAL_MACHINE\Software\FnGBoot:
  VersionMajor = 9 (0x00000009)
  VersionMinor = 0 (0x00000000)
  VersionSVN = 3476 (0x00000d94)
\> toolbox reg show HKLM\Software\EBoot
Keys of HKEY_LOCAL_MACHINE\Software\EBoot:
Values of HKEY_LOCAL_MACHINE\Software\EBoot:
  VersionMajor = 8 (0x00000008)
  VersionMinor = 4 (0x00000004)
  VersionSVN = 391 (0x00000187)
  VersionConfig = 1 (0x00000001)
```

5 System Services and Tools

The standard OS image provided by Garz & Fricke includes several useful services for flexible application handling. Some of them are just run-once services directly after the OS has been started, others are available permanently.

5.1 Services

Services are those programs that run in the backgrouund and provide the functionality on demand.

5.1.1 Autojob

The autojob service is used for initializing, copying and starting of programs. It is a Garz & Fricke service and not a standard Windows Embedded Compact 7 service. The functions provided by this service are called AutoInit, AutoCopy and AutoStart.

AutoInit The AutoInit is executed at the first boot of the device. It executes, similar to the AutoStart function, executables and batch files contained in an "AutoInit" folder of the devices partitions. Via this function the device configuration and calibration may be executed. Production tests make use of this feature to customize the installation for customer specific adaptions.

All subfolders and files within a folder named "autoinit" on an USB stick, SD-Cards or in the persistent eMMC Flash partition (also known as "FlashDisk") will be copied to the root of the device. Afterwards the excetutables and batch files will be executed. Non-existing folders will be created automatically.

AutoInit executes applications in "AutoInit" folders **once**.

AutoCopy The AutoCopy is executed each time after the OS has booted and when any storage medium has been inserted. This service provides a comfortable installation and/or update functionality as well as copy mechanism for specific files that are not included in the OS (e.g. for runtime libraries). Subfolders and files within a folder named "autocopy" on an USB stick, SD-Cards or in the persistent eMMC Flash partition (also known as "FlashDisk") will be copied to the root of the device respectively its equivalent targets. Non-existing folders will be created automatically.

AutoCopy copies files and folders to the root path of the device respectively its subfolders.

The behaviour can be changed in the registry. The key is `HKEY_LOCAL_MACHINE\Software\AutoJob\AutoCopyList`. Possible values are:

Name	Type	Description
DefaultEnable	DWORD	The default setting for all pathes not listed below.
<path_media>	DWORD	1 enables Autocopy for this media path, 0 disables.

Hint: It is also possible to use this mechanism to copy files, e.g. from usb stick to sd card. In this case the subfolder name within the autoinit folder on the usb stick must be named equal to the sd card folder which is usally mounted to the root directory after the sd-card has been plugged in.

AutoStart This service is executed each time after the OS has booted or when a storage medium has been inserted. It checks the "FlashDisk" partition as well as the USB stick and SD-card for an existing autostart folder. All applications and batch files found therein will be executed.

Autostart executes applications in "Autostart" folders. Note that .exe files will be executed before .bat files.

To use this service, create an autostart folder on the storage medium or in the "FlashDisk" partition. Copy your application into this folder.

The behaviour can be changed in the registry. The key is `HKEY_LOCAL_MACHINE\Software\AutoJob\AutoStartList`. Possible values are:

Name	Type	Description
DefaultEnable	DWORD	The default setting for all pathes not listed below.
<path_media>	DWORD	1 enables Autostar for this media path, 0 disables.

Hint: Executables that have been copied to an autostart folder by the autocopy job will not be executed after copying.

Security Hint: Both services allow the external injection of executables via USB-stick or SD-Card. Once, the executables have been copied to the system, they will be started on every reboot. This might be a security leak.

5.1.2 FTP

This service is available after OS boot. It enables a file transfer between the device and a workstation over Ethernet without ActiveSync. A proper network connection is required. The FTP connection via IP address is comfortable for fast file transfer. Note that a password is not required by default. Please refer to chapter [► 9 Securing the device].

5.1.3 Telnet

Telnet gives remote access to the device via a console e.g. to execute applications or utilities directly on the device. Also for this service, a proper network connection is required. After proper connection, the console is available. Note that a password is not required by default. Please refer to chapter [► 9 Securing the device].

5.1.4 Tempmon

Monitors the temperature of the SoC and executes a emergency reset if the temperature is too high. The alarm temperature can be queried and set in the Registry in the `HKLM\drivers\builtin\tempmon\Alarm Temperature` key. The tool may be executed from the command line to query the current temperature. Example:

```
\> tempmonitor
Read cpu temperature 48
```

Example of changing the alarm temperature with **toolbox.exe**. See also chapter [► 5.2.6 toolbox].

```
toolbox reg set HKLM/drivers/builtin/tempmon/"Alarm Temperature" d 99
```

5.2 Utilities

5.2.1 bootselect.exe

This tool offers the possibility to switch the boot target between the flashed operation system (e.g. Linux, Android or Windows Embedded) on the device and the **Flash-N-Go System** management OS. The following code block shows a list of all possible parameters.

```
Usage: bootselect [<bootmode>]

When called without parameters, bootselect reads the
current boot-mode setting from the RTC-register

Options:
    <bootmode>    Writes the given boot-mode into the RTC-register
                  May be one of: initialized regular alternative invalid
```

Note that usually you just need the bootmode **regular** for booting the target os and **alternative** for booting **Flash-N-Go System**.

5.2.2 Disp.exe

In case, the device is not installed in 12 o'clock (0°) view, the screen orientation can be changed. For example, the device is turned into vertical position by 90°clock wise (pls. compare below illustration). In this case, the

screen must be adopted by a 90°turn counter clockwise. `\Windows\disp.exe` Hint: This tool can only be executed by command-line or via Telnet without a GUI dialog box.

```
Command line: disp.exe -r [0=0deg|1=90deg|2=180deg|4=270deg]
```

All angles/rotations are to be understood as counter clockwise and base on the 0°(12 o'clock) default value (not on the current setting).

5.2.3 ntlmadmin.exe

A very simple NT LAN Management implementation. It currently supports adding and deleting users for service authentication.

```
\> ntlmadmin
Tool to manage NTLM users (1.0)
Usage: ntlmadmin <command> [arguments]
Commands:
        adduser <user> <password>
                Adds a user with password.
        deluser <user>
                Deletes a user.
        listusers
                Lists all users.
```

If you want for example ftp access for authenticated users only you need add a user and to enable authentication for ftp. This could be done in the registry as follows:

```
ntlmadmin adduser foo bar
toolbox reg set HKEY_LOCAL_MACHINE\COMM\FTPD "IsEnabled" d 1
toolbox reg set HKEY_LOCAL_MACHINE\COMM\FTPD "UseAuthentication" d 1
toolbox reg set HKEY_LOCAL_MACHINE\COMM\FTPD "AllowAnonymous" d 0
toolbox reg set HKEY_LOCAL_MACHINE\COMM\FTPD "AllowAnonymousUpload" d 0
toolbox reg set HKEY_LOCAL_MACHINE\COMM\FTPD "AllowAnonymousVroots" d 0
savereg -n
services stop FTP0:
services start FTP0:
```

5.2.4 reset.exe

This tool executes a system reset. The reset is executed via the watchdog, thus resetting the device in a hard way. Keep in mind to save all files and flush the registry (see 5.2.5) before before resetting. `\Windows\reset.exe`

5.2.5 SaveReg.exe

All changes performed regarding system settings (backlight, power management, desktop, taskbar etc.) that are written to the Registry will be flushed to the persistent memory. If they have to be kept persistent, the registry must be saved before the next power-cycle. This can be done via GUI or by command line. This tool can also be executed via Telnet (`\Windows\SaveReg.exe`)or called by custom application without a GUI popup.

```
Command line: savereg.exe [-n|-s]
-n: perform saving without GUI dialog box. -s: silent mode, without output
```

5.2.6 toolbox

The toobox.exe is a command line tool and has several useful functions for Windows CE devices:

- kill process
- copy file
- print directory listing
- read and write to registry

- print memory and system information
- list, start and stop drivers
- small performance benchmark
- dump memory to file
- change the backlight

Type toolbox for detailed information.

5.2.7 Touchc.exe

This tool can be used to start the touch calibration manually. The calibration data is written to the Resistry. However for most touches this should not be necessary. The calibration data for most touch types is taken from XML and will be overwritten on next boot.

6 Deploying the operating system to the target

A Garz & Fricke OS image can be installed on the system's internal eMMC flash memory using the Garz & Fricke **Flash-N-Go System**. This is a small RAM-disk-based Linux which is installed on your **i.MX6** in parallel to the regular operating system. This chapter describes how to boot your device into **Flash-N-Go System** and how to use it to install your Windows Embedded Compact 7 image.

6.1 Accessing the target system via serial console

Before updating the target operating system we first need an access to the device. The easiest way to access the target is via the serial console. Simply connect the first RS-232 port of your target system with a COM port of your PC or a USB-to-RS-232 converter using a null modem cable. With the serial connection set up start your favourite terminal program (e.g. minicom or tera term) with the following settings:

- 115200 baud
- 8 data bits
- no parity
- 1 stop bit
- no hardware flow control
- no software flow control

From the very first moment when the target is powered, you should see debug messages in the terminal.

6.2 Booting Flash-N-Go System

There are two ways of booting your device into **Flash-N-Go System**. If the device already has a working OS image installed, you can switch to **Flash-N-Go System** by issuing the following commands on the device console (i.e. via TELNET):

```
> bootselect alternative
> reset
```

The device will reboot and show the **Flash-N-Go System** splash screen on the display. On the serial console (see [► 6.1 Accessing the target system via serial console]), the command prompt should appear:

```
--------------------------------
 Garz & Fricke Flash-N-Go System
--------------------------------

FLASH-N-GO:/
```

The change of the bootmode using the **bootselect** command is permanent, i.e. the next boot of the device will start **Flash-N-Go System** again, until the bootmode is set back to regular operation:

```
FLASH-N-GO:/ bootselect regular
```

See also [► 5.2.1 bootselect.exe] for detailed information. Alternatively, the bootmode can be switched temporarily by pressing down and holding the **bootmode switch** while the power supply is switched on. The location of the **bootmode switch** is shown in [► Figure 1].

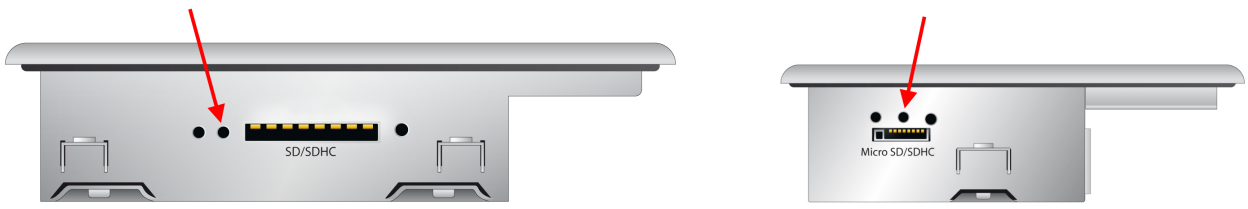

Figure 1: Location of the bootmode switch on the side of the device

This method changes the bootmode only for a single boot. The next boot of the device (without the **bootmode switch** pressed) will boot the regular operating system again.

For more detailed information concerning the Garz & Fricke **Flash-N-Go System** please consult the **Flash-N-Go System Manual**.

6.3 Installing a Windows Embedded Compact 7 image on the device

Garz & Fricke provides a shell script for installing a Windows Embedded Compact 7 image on the device, called **SANTARO_CE7P-10.1r442-0_fng_ce7_install.sh**. The files can be installed either locally on the device (e.g. a USB drive or an SD card) or remotely via TFTP. Regardless which solution you prefer, you will need the following files from the download folder by following the link **Image** of the corresponding os version from the the download area:

► http://support.garz-fricke.com/projects/Santaro/WindowsCE7/

- SANTARO_CE7P-10.1r442-0_fng_ce7_install.sh
- SANTARO_CE7P-10.1r442-0_nk.bnx
- SANTARO_CE7P-10.1r442-0_boot.cfg
- SANTARO_CE7P-10.1r442-0_eboot.nb0
- md5sums.txt

The filenames might differ slightly for the different versions and OS configurations. The file extensions should be equal, though, so it should not be difficult to determine the correct files.

Note: The installation via **SANTARO_CE7P-10.1r442-0_fng_ce7_install.sh** removes any previously installed regular operating system. If the installation fails for some reason, the device will always boot into **Flash-N-Go System** afterwards.

6.3.1 Over the network via TFTP

During development, the most comfortable way of installing the images on the device is by loading them over the network via TFTP. For this purpose it is necessary to install a TFTP Server on your host machine. Copy all files to the TFTP-Root directory. The ethernet connection for **Flash-N-Go System** has to be configured via **sconfig** as described in the corresponding manual:

► http://support.garz-fricke.com/projects/Santaro/Flash-N-Go/overview.html

Note: The network settings in **Flash-N-Go System** are independent of the settings configured in WindowsCE7.

The script can be loaded to the device and executed there via the **Flash-N-Go System** shell. Assuming, that your TFTP host has the IP address 192.168.1.100, type:

```
FLASH-N-GO:/ export TFTP=192.168.1.100; curl tftp://$TFTP/
    ↪ SANTARO_CE7P-10.1r442-0_fng_ce7_install.sh > /tmp/a.sh; sh /tmp/a.sh
```

The above command loads the **SANTARO_CE7P-10.1r442-0_fng_ce7_install.sh** script from your TFTP server to the **/tmp** directory of the **Flash-N-Go System** and executes it from there. During execution of the script, the Windows Embedded Compact 7 image files will be loaded from the TFTP server and written directly to the eMMC flash memory.

The installation procedure will take some minutes. You can observe the output messages of the process on the terminal console. After successful installation the script returns to the Flash-N-Go prompt:

```
Update successful
FLASH-N-GO:/
```

6.3.2 From a local folder using an external storage device

If you do not have a network connection to your device, the **SANTARO_CE7P-10.1r442-0_fng_ce7_install.sh** can be copied to an external storage device, e.g. a USB driver or an SD card, along with the Yocto images. Simply put all files into the same folder and insert the storage device into your i.MX6.

The TFTP environment variable must not be set. Usually the variable is not set, so you do not have to worry about this. If you have tried using TFTP before, though, it probably contains your TFTP server IP address and has to be unset explicitly:

```
FLASH-N-GO:/ sh /mnt/mstick1/SANTARO_CE7P-10.1r442-0_fng_ce7_install.sh
```

The installation procedure will take some minutes. You can observe the output messages of the process on the terminal console. After successful installation the script returns to the Flash-N-Go prompt:

```
Update successful
FLASH-N-GO:/
```

Note: The installation from a local folder requires Flash-N-Go System 4.0 or higher.

6.3.3 Control the installation process using parameters

It is possible to add paramters to the update-script call. This can be used to control the update process, add files and install additional features.

The options available are:

`-a|--AI` Autoinit files: Executeables and batch file which will be executed once after first device boot.
`-b|--BS` Boot script: Specifies the bootscript file (boot.cfg).
`-c|--CleanEboot` Flag which deletes Eboot setting (xml settings will still be used).
`-i|--Image` Boot logo: Specifies the bootlogo file.
`-o|--OS` Winows OS image: The image binary.
`-p|--ParamFile` Parameter file: Each parameter on a separate line.
`-t|--TFTP` Additional file (Format: `<filename>:<target>`): Not applicable for WindowsCE7.

The following line can be used to install a boot logo (that is also located in your TFTP-Server directory):

```
FLASH-N-GO:/ export TFTP=192.168.1.100; curl tftp://$TFTP/
    ↪ SANTARO_CE7P-10.1r442-0_fng_ce7_install.sh > /tmp/a.sh; sh /tmp/a.sh --Image=
    ↪ my-logo.bmp
```

Note: Be aware that the boot logo needs a license, either embedded into logo itself or as general XML-license file.

The parameter file (denoted by the `--ParamFile` flag) provides a way of specifing multiple parameters in one file. This makes a call with many parameters easier to handle.

A parameter file consists of the same parameters line by line. For example, one can write a parameter file that installs a custom boot logo and a custom kernel by writing the following lines into a file `customconfig.txt` located in the TFTP server directory.

```
--Image=my-logo.bmp
--OS=custom-windows-image.bnx
```

Calling the install script with the `-p customconfig.txt` parameter will install the custom files.

7 Garz & Fricke APIs and Frameworks

This chapter gives a short overview of the Garz & Fricke APIs and the Garz & Fricke Device Framework.

7.1 Garz & Fricke Device Framework

The Garz & Fricke Device Framework also called Garz & Fricke .NET Framework serves as an abstraction layer between the native C API calls of the Garz & Fricke devices equipped with the .NET Compact Framework and the object oriented application layer. The framework fulfils .NET designing guidelines. The framework installation package includes in addition to the .NET assemblies a detailled documentation and sample projects including sample code for its classes.

7.1.1 Installing the framework

The framework package can be downloaded here:

- http://support.garz-fricke.com/products/Vincell/Tools/DeviceFramework/

It includes the following files

- Framework ReleaseNotes.rtf: Release notes of the current verion. Please read them carfully before using the framework to ensure that all requirements are fullfilled.
- FrameworkDocumentation.chm: Documentation which describes the framework classes and methods in detail.
- GarzAndFrickeDeviceFramework.msi: The installation binary for Windows Systems.

Uninstall any previeously installed versions of the framework first. It is called **Device Framework** in the **Add or Remove Programs Utility**. Just download the .msi file to a windows work station, execute it and follow the instructions of the installation wizard. Note that you will be asked for a reboot after the installation process has finished.

During the installation of the SDK and/or BSP, on some work stations using VS2008 there might appear error messages such as

```
ToolsMsmCA(Error): IHxFilters filter registration failure:
Err: 0x80040305, Context = pFilters->SetNamespace(
Namespace )
```

and/or

```
ToolsMsmCA(Error): IHxRegisterSession transaction
failure: Err = 0x8004036f,
pRegSession->Transaction()
```

These errors can be mostly observed with Visual Studio 2008 and there is a workaround: Deselect the documentation package during the installation step.

7.1.2 Using the framework

After installing the framework you will find the documentation and some sample projects under a path similar to this:

`C:\Program Files (x86)\Garz & Fricke GmbH\Device Framework.`

Within the subfolder **Samples** you will find some IO scenarios for the different supported interfaces like **CAN** or **RS485** as well as application scenarios. You can just open the corresponding project files with **Visual Studio**. Please read the release notes and the documentation for detailed information.

7.2 Garz & Fricke Software Development Kit

The **Garz & Fricke SDK** allows access to the platform specific interfaces and modules for customized applications. You can download the installation file SANTARO (i.MX6Q) SDK CE7 10.1r442-0.msi from the web page by clicking the **SDK** link of the desired os version:

► http://support.garz-fricke.com/products/Santaro/WindowsCE7/overview.html.

Uninstall any previeously installed versions of the SDK first. After installing the SDK you will find all libraries and header files under a path similar to this:

```
C:\Program Files (x86)\Windows CE Tools\SDKs\Santaro Q CE7 SDK
```

The header files of the **Garz & Fricke** interface and module SDKs includes detailled information of its usage.

7.2.1 can

Please refer to the can api documentation **can_api.h**.

7.2.2 digitalio

Please refer to the digital io api documentation **gfdigitalioapi.h**. the pin assignment is defined in the hardware manual called **handbuch** from the corresponding Santaro download area at
► http://garz-fricke.com/produkte/alle-kategorien/

7.2.3 gfversionapi

Please refer to the digital io api documentation **gfversionapi.h**. note that this api is not complete. most of the functions are not supported anymore.

```
\> gfversionstubtest.exe
gfv_sendversioninfo returns error_not_supported
gfv_creategfversioninfo returns error_not_supported
gfv_destroyversioninfo returns error_not_supported
gfv_svnversionarraytosvnbuildnumber returns error_not_supported
gfv_getcpuid returns error_not_supported
gfv_queryversiondetails returns error_not_supported
gfv_queryhardwarerevision returns error_not_supported
gfv_queryserialnumber returns error_success
gfv_queryserialnumber: szserial: "01800285" lpseriallength: 8
gfv_querysoftwareversion returns error_success
gfv_querysoftwareversion: szmajor: "8 " ulmajorlength: 2
gfv_querysoftwareversion: szminor: "2 " ulminorlength: 2
gfv_querysoftwareversion returns error_success
gfv_querysoftwareversion: szmajor: "8 " ulmajorlength: 2
gfv_querysoftwareversion: szminor: "2 " ulminorlength: 2
```

Please note that the complete version string is stored in a textfile called **os-image.ver** within the windows folder now. You can read it as follows:

```
\> type \windows\os-image.ver
OS_MAJOR: 10
OS_MINOR: 0
OS_FIX: 0
OS_SVN: 2:416
OS_CONFIG: 0
Date: 09.03.2017
Time: 13:54
```

8 Development Workstation

The development environment varies due to individual requirements. The following lineup makes no claim to be complete, but we suppose it to be practicable for most common requirements. Except as noted otherwise, the mentioned parts are not part of the delivery by Garz & Fricke.

8.1 Hardware recommendation

For the minimal hardware requiremenst see also ► https://msdn.microsoft.com/en-us/library/jj200354.aspx.

- x86 or x64 architecture
- 2.4 GHz CPU or better
- Min. 4 GB RAM
- Typ. 100 GB free HDD space
- DirectX 9-capable video card running at 1024 x 768 or higher display resolution
- 10/100 MBit/s Ethernet (directly, or via switch)
- RS-232 serial interface
- USB port

8.2 Software requirements

A typical application development for Windows Embedded Compact 7 (WEC7) requires Microsoft Visual Studio 2008 Professional SP1. For Windows Embedded Compact 2013 (WEC2013) the Microsoft Visual Studio 2013 should be used. Both versions provide compilers for native code written in C/C++ and managed code based on .NET compact framework 3.5 written in C# or VB. compilers. As Windows Embedded Comapct 2013 changed the binary interface, applications compiled for Windows Emebedded Comapct 7 and before are not working without recompiling them for Emebdded Compact 2013. Developing managed C++ applications (CLR) is not supported.

Please note that all Visual Studio Express versions and all Visual Studio 2010 and 2012 versions do not provide compilers for WEC7.

For transferring image files to the target boot loader a TFTP server is recommended. It is also possible to install the image file via USB-Stick or SD-Card.

For building applications with Silverlight Embedded technologies Expression Blend 3 for WEC7 will be necessary too.

Please note that only the compiler from platform builder for WEC7 can generate optimized code for floating point operations.

The installation of the Garz & Fricke SDK will enable you to have access to interfaces that are not supported by the Microsoft standard driver set. It offers a simple and convenient help to set up project environments in your Visual Studio IDE. See also [► 7 Garz & Fricke APIs and Frameworks] for the specific versions. We recommend regularly checks for latest support files on our website.

8.3 Connecting the device via Microsoft ActiveSync (USB)

Formerly the Windows CE devices supported debugging using ActiveSync via USB. This feature is no longer supported on Windows Embedded Compact 2013 and is not supported by Garz & Fricke on WEC7. Please use the Ethernet enabled debuggung feature instead.

8.4 Connecting the unit via TCP/IP (Ethernet)

Using the TCP/IP connection requires some preparation on device side as well as on the workstation. Nevertheless, the integrated FTP service offers simple and fast file transfer over Ethernet.

8.4.1 Device IP-address

Connect the unit to Ethernet. After booting to Windows EC7, please check the current TCP/IP address management by clicking on the network icon on the right side of the task bar. The default setting is the static IP address 192.168.1.1. Use the control panel to change the IP address manually or to enable DHCP mode.

In this example we consider a direct connection to your development PC having the IP 192.168.1.100 and the device with the default IP 192.168.1.1.

8.4.2 Preparing the embedded HMI for TCP/IP debug connection

By default, the incoming debug connection via TCP/IP is deactivated for security reasons. An open debug port allows access to almost all system functions. It is highly recommended to keep the debug ports closed when not in use.

8.5 Setup the development environment

The steps for a proper setup of the development software may vary from case to case.

8.5.1 Setup the TCP/IP connection

The first step is to setup the IP address of the device connection properties. In Visual Studio 2008 for WEC7 choose **Options** from the **Extras** menu. Select **Devices** from **Device tools**, mark **Windows CE-Device** and click on **Properties**.

Open the TCP/IP configuration mask and enter the current IP address of the Garz & Fricke device. Confirm with OK and close all related windows.

8.5.2 Establish the connection

Before creating a new project, we recommend to check the proper connection configuration as per previous chapters.

By default, the devices do not accept debug connections via TCP/IP. To enable this service on Windows Embedded Compact 7, you have to start the connection client ConmanClient2.exe (after every reboot) and CMAccept.exe (before every new connection). Please execute "CMAccept.exe" and "ConmanClient2.exe" via TELNET or by other means (from a tool or the autostart feature).

To enable this service on Windows Embedded Compact 2013, the tools are named "CMAccept3.exe" and "ConmanClient3.exe".

Within a time frame of 3 minutes, the device will accept the access through Visual Studio. On failure, execute at least CMAccept.exe respectively CMAccept3.exe again. If you are not sure, whether the ConmanClient is running, execute it (re-)start it as well.

After proper setup, the connection dialog should confirm the successful connection via **Tools -> Conect to device...** in Visual Studio.

9 Securing the device

The meaning of security for embedded systems is often underestimated. This chapter should sensitize customers to the needs of security and disclose some of the typical security holes. It also provides some tips and hints for the implementation of well chosen security mechanisms. Since we cannot cover the big amount of security issues in this manual we strongly recommend to read further secondary lecture regarding this topic.

Note: For the following list of security risks, no claim of completeness can be made. There may arise other risks or - on the opposite - limitations in the design of your application by following the instructions provided in this chapter.

9.1 Services

The default configuration of a Garz & Fricke device can be described as "developer friendly". This means, all services are available and activated. Depending on the final application, this might be either helpful or a security risk. Once the development has been finalized, we recommend a review of the required services and to disable all services and features which are not used. See

► https://msdn.microsoft.com/en-us/library/ee500226.aspx

and the chapter [► 5 System Services and Tools] how to disable. Special care needs to be taken, e.g. for:

Telnet The chapter [► 5.1.3 Telnet] describes the telnet service. For production devices it should be carefully decided if this feature is needed and how it is secured. At least the password and user suggestions from [► 9.2 User permissions concept] should be implemented.

Note: All services and servers with open ports are listed here [► A.6 Services]. These open ports could pose security risks.

9.2 User permissions concept

Windows Embedded Compact is designed as a single-user system and provides by design no mechanisms like **user persmissions** or **file permissions**. Thus users and applications have permission to change all writable files and the regsitry and may call many functions provided by the opearing system. There is however a destinction between Kernel-Mode and User-Mode drivers and address spaces. See ► https://msdn.microsoft.com/en-us/library/jj659778.aspx for more information.

Therefore it is not always easy for customers to follow the principle of minimal privilege for user rights on the devices. Some network services implement a simple, way to authenticate using a user and password tuple. The API used is a simpel implementation of the NT LAN Manager (NTLM) (see ► https://msdn.microsoft.com/en-us/library/ee498104.aspx).

You can also use the [► 5.2.3 ntlmadmin.exe] Garz & Fricke tool to add and remove users for service authentication.

9.2.1 Root password

Since Garz & Fricke is an OEM manufacturer and we are delivering serial-produced devices to several customers a default user and password tuple would lead to a form of pseudo security. Moreover a default password for all devices is highly vulnerable. For this purpose our devices usually have no user and password set. It is essential that a password will be set by customers before the devices are deployed. Alternatively the services should be disabled by default. These are the minimum security measures to be done.

Adding a root user with a sample password using the Garz & Fircke ntlmadmin tool (see 5.2.3).

```
\> ntlmadmin adduser root myrootpassword
Added user root and password.
\> ntlmadmin listusers
Users:
        root
```

To enable the authentication for TELNET or FTP refer to 9.8.2 and 9.8.1

What does it mean if no user or password is set? Without a user and password attackers might connect to the device via TELNET or FTP and have full system control since the users have all permissions.

9.3 Autojob

Garz & Fricke devices are equipped with an **autostart** and **autoinit** service as described in chapter [▶ 5 System Services and Tools]. As this services may execute anything with root privileges without further checking, this is a possible vulnerability.

Restricting the physical access to the interfaces by the mechanical construction is one way to reduce the risk of attacks. To disable this feature completely, execute:

```
\> toolbox reg del HKLM\Init Launch61
Opening subkey: HKEY_LOCAL_MACHINE\Init:
Deleted value:  Launch61
\> toolbox reg del HKLM\Init Depend61
Opening subkey: HKEY_LOCAL_MACHINE\Init:
Deleted value:  Depend61
```

Note: Updating the device with **Flash-N-Go Update** or any other automatic update tool may not be possible without the autostart feature.

9.4 Flash-N-Go System

Newer Garz & Fricke devices are equipped with a **Flash-N-Go System** as backup OS. Within **Flash-N-Go System** the user has full control of the device's configuration and the partitions on the flash disk respectively eMMC without a password or further authentication.

As described in [▶ 6 Deploying the operating system to the target] booting into **Flash-N-Go System** can be triggered by pressing the **bootmode switch** or with the bootselect tool.

Be aware that the bootselect tool can be executed by any user or application.

The bootmode switch should be secured with restricing physical access by the mechanical construction.

If this is impossible, the access to the backup OS can be disabled with the following command sequence executed in **Flash-N-Go System**

```
FLASH-N-GO:/ mount /dev/mmcblk0p2 /mnt/
FLASH-N-GO:/ cp /mnt/boot.cfg /mnt/boot-alt.cfg
FLASH-N-GO:/ umount /mnt/

FLASH-N-GO:/ mount /dev/mmcblk0p1 /mnt/
FLASH-N-GO:/ mv /mnt/boot-alt.cfg /mnt/boot-alt.cfg.bak
FLASH-N-GO:/ umount /mnt/
```

Note: This change disables the access to the backup OS **Flash-N-Go System** completely. If the normal OS becomes inaccessible for some reason, there is no way for a custumer to fix the device.

Note: Updating the system without the backup OS **Flash-N-Go System** is usually not possible.

9.5 Restrict physical access

Each physical interface like USB, SD-Card or ethernet socket can serve as an entrance gate for hackers. If you limit the number of easily accessible interfaces you in turn decrease the possibility for attackers to connect with the target device. You need less concern about security mechanism for those interfaces which are not accessible or not equipped at all.

9.6 Application security

Application security is often not considered to be a high priority for embedded devices. But it is, of course, essential to take account of identifying risks in embedded applications. Since application development is a very complex subject and it is out of scope for Garz & Fricke developement we will refer to secondary lecture at this point.

9.7 Default configuration (enabled services)

The following services/features are enabled by default.

9.7.1 FTP

A password is not required for the login by default. Files might be deleted, manipulated or replaced.

9.7.2 Telnet

A password is not required. Files might be deleted, manipulated or replaced. Tools may be executed and the device might be reset.

9.7.3 Webserver

The webserver is activated by default and shows an example website. See the Microsoft Documentation ► https://msdn.microsoft.com/en-us/library/ee500332.aspx for more information.

9.7.4 Garz & Fricke management protocol service "G&F Discovery"

This service is used by the Garz & Fricke DeviceControlCenter Software. A password is not required to receive, modify and delete the device configuration.

9.7.5 Garz & Fricke autojob (autocopy and autostart)

Autocopy This service automatically copies files from/to certain folders after an external storage medium such as USB-stick or SD-card has been connected to the device. Existing files will be overwritten.

Autostart This service automatically runs executables in "autostart" folders on external storage media such as USB-stick or SD-card, once it has been connected to the device.

9.7.6 USB Devices

Mouse and keyboard support is enabled.

9.8 How to deactivate single services/features

It is possible to disable single services. Whereas the Garz & Fricke services can be deactivated in the registry of Windows Embedded Compact, for some of the Microsoft standard services the procedure might be more complicated. For example, to disable the USB support for mouse and keyboard, the Microsoft Platform Builder is required and a new ROM image must be created and uploaded to the device.

9.8.1 How to disable or secure FTP

Please follow the Microsoft documentation at ► http://msdn.microsoft.com/en-us/library/ee500827.aspx For authentication options see ► https://msdn.microsoft.com/en-us/library/ms881870.aspx and ► https://msdn.microsoft.com/en-us/library/ms881878.aspx. The users can be added with the ntlmadmin tool 5.2.3.

9.8.2 How to disable or secure Telnet

Please follow the Microsoft documentation at ► http://msdn.microsoft.com/en-us/library/ee501107.aspx For authentication options see ► https://msdn.microsoft.com/en-us/library/ms881870.aspx and ► https://msdn.microsoft.com/en-us/library/ms881878.aspx. The users can be added with the ntlmadmin tool 5.2.3.

9.8.3 How to disable or secure webserver

Please follow the Microsoft documentation at ► http://msdn.microsoft.com/en-us/library/ee500332.aspx For authentication options see ► https://msdn.microsoft.com/en-us/library/dn269481.aspx and ► https://msdn.microsoft.com/en-us/library/ee499476.aspx. The users can be added with the ntlmadmin tool 5.2.3.

9.8.4 How to disable the Garz & Fricke management protocol service "G&F Discovery"

The registry key for activation is stored in `HKEY_LOCAL_MACHINE\Services\Discover` (Type: REG_DWORD)

Name	Type	Enabled by Data (Dec)	Function
Flags	REG_DWORD	0x00000000 (0)	Enabled (default)
		0x00000004 (4)	Disabled

9.8.5 How to configure or disable the Garz & Fricke autojob

Autocopy The registry keys for this service are stored in `HKEY_LOCAL_MACHINE\SOFTWARE\AutoJob\AutoCopyList` By default, the service will scan all media (i.e. Flash drives, USB memory stick, SD card) for "autocopy" folders, for which no separate handling has been configured. To include or exclude single media from the global rule, a separate key must be generated, examples below.

Name	Type	Enabled by Data (Dec)	Function
Default Enable	REG_DWORD	0x00000001 (1)	All media scan enabled (default)
		0x00000000 (0)	All media scan disabled
USBDisk	REG_DWORD	0x00000001 (1)	Perform autocopy from USB-Stick
		0x00000000 (0)	Ignore autocopy on USB-Stick
SD	REG_DWORD	0x00000001 (1)	Perform autocopy from SD card
		0x00000000 (0)	Ignore autocopy on SD card
FlashDisk	REG_DWORD	0x00000001 (1)	Perform autocopy from eMMC FlashDisk
		0x00000000 (0)	Ignore autocopy on eMMC FlashDisk

Hint: Both ways are possible: You can disable the global scan and activate the service for single media, or – vice versa – enable the global search and exclude medias from the autocopy scan.

AutoStart The registry keys for this service are stored in `HKEY_LOCAL_MACHINE\SOFTWARE\AutoJob\AutoStartList` The functionality is exactly the same as explained before for the Autocopy service.

9.8.6 How to secure the device

Please follow the Microsoft documentation at ► http://msdn.microsoft.com/en-us/library/ee498113.aspx

10 Related documents and online support

Support for your Garz & Fricke embedded device is available on the Garz & Fricke website. You may find a list of the documents available, as well as their latest revision and updates for your system.

Hadware manuals and datasheets regarding the SANTARO Products can be found under

- http://support.garz-fricke.com/projects/Santaro/Documentation/

The different documents descibe the SANTARO device in the different hardware configurations. Information regarding the Flash-N-Go architecture can be found under ► GF_Flash-N-Go_Manual-8.0-r5769.pdf. The bootloader Flash-N-Go Boot and the managemant system can be found under Flash-N-Go

- http://support.garz-fricke.com/projects/Santaro/Flash-N-Go/

The different operating systems can be found in the subdirectories of

- http://support.garz-fricke.com/projects/Santaro

For each OS release a release note document is supplied that contains the change notes and test reports as well as known issues. Garz & Fricke is not responsible for the content of the above mentioned third party sources and/or information. We quote these sources just for your convenience.

A Standard Device Configuration

This section describes the default configuration of a Garz & Fricke device. It is considered as a template for further customer configuration, that may also done during the production test. Please request detailed information how to customize the configuration for series production from your contact person at Garz & Fricke GmbH.

A.1 Operating System

OS Type	Windows Embedded Compact 7
OS Release Version	10.1r442-0

A.2 Bootloader

Bootloader Type	Flash-N-Go Boot

A.3 Boot Logo

Boot Logo Type	Standard platform logo
Boot Logo License	No license needed

Note: A custom boot logo always requires a boot logo license!

A.4 Serial diagnostic port

Activated by Default	☑
Serial Port	RS-232 #1
Baud Rate	115200
Data Bits	8
Parity	None
Stop Bits	1
Flow Control	None

A.5 IPv4 Settings

DHCP	Deactivated
IPv4 Address	192.168.1.1
IPv4 Subnet mask	255.255.255.0
IPv4 Default Gateway	192.168.1.100

A.6 Services

Service	Port	Enabled by Default
Telnet Server	23	☑
FTP Server	20	☑
HTTP Server	80	☑
UPnP	5120	☑
HTTPS Server	443	☑
SNMP Server	161	☐
SNTP	123(UDP)	☑
NetBIOS	137(UDP)	☑
	138(UDP)	☑
G&F Discovery	7311(UDP)	☑

A.7 Display

The display orientation of the device is by default landscape. The most connectors of the device are at the bottom side. We consider this to be 0° rotation.

Note: When hardware is rotated clockwise, the display content must be rotated counterclockwise.