



- **GETTING STARTED WITH 1-WIRE DEVICES**

---
- for temperature measurement &

---
- driver/vehicle identification

---
- **APPLICATION NOTE**

---
- **FOR FOX3 & FOX3-3G DEVICES**

---

## Version history:

*This table provides a summary of the document revisions.*

Version	Author	Changes	Modified
1.0.1	F. Beqiri	<ul style="list-style-type: none"><li>- Supported new 1-wire temperature sensor (MAXIM DS18B20)</li><li>- FOX3-LITE and FOX3-3G-LITE do not support 1-Wire interface (accessory port is not present).</li><li>- Added note about unused wires on the cable - see chapter "<a href="#">1-Wire installing considerations</a>"</li></ul>	08/07/2016
1.0.0	F. Beqiri	<ul style="list-style-type: none"><li>- Initial version</li></ul>	05/05/2015

## ***Table of contents***

<b>1</b>	<b>ABOUT THIS DOCUMENT.....</b>	<b>5</b>
1.1	Introduction.....	5
1.2	1-Wire specifications.....	5
1.3	1-Wire installing considerations.....	6
1.4	What are the typical applications?.....	6
1.5	How does the 1-Wire interface work?.....	6
1.6	Installing 1-Wire devices.....	7
1.6.1	Parasite power mode (recommended by FALCOM).....	7
1.6.2	External supply mode.....	7
1.7	Testing your 1-Wire connection.....	8
1.8	Set up automatic 1-Wire alarm reporting.....	9

## Cautions

Information furnished herein by FALCOM is believed to be accurate and reliable. However, no responsibility is assumed for its use. Please read carefully the safety precautions.

If you have any technical questions regarding this document or the product described in it, please contact your vendor.

General information about FALCOM and its range of products are available at the following Internet address: <http://www.falcom.de/>

## Trademarks

Some mentioned products are registered trademarks of their respective companies.

## Copyright

This document is copyrighted by **FALCOM GmbH** with all rights reserved. No part of this documentation may be produced in any form without the prior written permission of **FALCOM GmbH**.

## FALCOM GmbH.

No liability (incl. patent liability) is assumed with respect to the use of the information contained herein.

## Note

Because our products are being continuously improved, specifications and information given in this document are subject to change by FALCOM without notice.

# 1 ABOUT THIS DOCUMENT

This application note provides a general overview of the 1-Wire® technology, its communication concept and explains some of its typical applications. It also gives some practical information on how to interface the 1-Wire **temperature sensors** and **iButtons** to the 1-Wire interface of the FOX3/-3G devices and how to configure and evaluate such external devices. Note that, the FOX3-LITE and FOX3-3G-LITE do not support 1-Wire interface (the 6pin accessory port is not present).

If this application note does not cover all the information you need, please refer to the additional documents "Falcom\_Ordering\_Guide.pdf", "FOX3\_HardwareManual.pdf" and "AVL\_PFAL\_Configuration\_Command\_Set.pdf" available on the FALCOM's website.

## 1.1 Introduction

A 1-Wire bus uses only one wire in addition to ground for signaling and power. One or several 1-Wire sensors can be connected to the bus at the same time. Sensors for driver identification or temperature logging are quite often used on vehicle applications. Thanks to the integration of the 1-Wire interface into the FOX3/-3G devices, system integrators for telematics, fleet management and intelligent transportation, can be more cost- and time-efficient in their work, due to the usage of additional external devices, which are not integrated in the in-vehicle computers. Using the 1-Wire interface allows them to identify vehicles and drivers, record the working time of mobile workers and, at the same time, monitor the temperature of the refrigerated trucks they drive. The shipper is thereby enabled to record the clock in and out of the driver as well as the transportation and delivery state of their temperature sensitive loads.

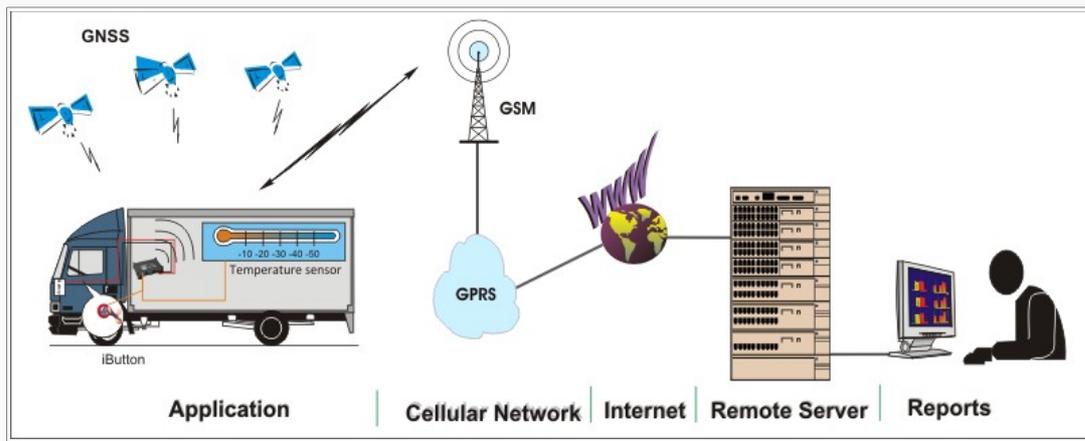


Figure 1: Typical application example using 1-Wire interface.

## 1.2 1-Wire specifications

This section lists the specifications of the 1-Wire interface on the FOX3/-3G:

Parameters	Description
Operating voltage range	+2.9 V to +5V DC
Maximum power output	20 mA
Supported Maximum Number of Slaves	2 x Temperature Sensors 2 x iButtons
Tested Sensor Types	- DS18(B)20 Temperature Sensors in Parasitic Mode using bus topology (from Maxim). - DS1402D-DR8+ Blue Dot Receptor iButton Reader Cable with DS1982 iButton and DS1921 iButton (from Maxim).

Table 1: Specification of 1-Wire interface

### 1.3 1-Wire installing considerations

This section lists some of parameters that need to be considered when interfacing the 1-Wire slave devices to the FOX3/-3G:

Parameters	Characteristics
Operating Temperature	A cold 1-Wire slave device (temperature sensor, iButton) needs more energy. The maximum available current on the FOX3/-3G device for 1-Wire sensors is 20 mA.
Cable & Length	Use cables with a cross-section of at least 0.6mm. For longer distance use a shielded twisted pair cable. The maximum cable length is 6 meters. <b>Hint: To increase the stability of the device you have to cut off all unused wires as short as possible to the connector of the CA71 or CA69 and to prevent the possibility of a short circuit cap them individually and tape. When using your own cable, please use only the wires that are really needed (only 1-Wire and GND wires, and not the complete 6-wires).</b>
Poll cycle	Short time requests slow down the bus communication, so therefore the request cycle should be at 30 seconds and higher.

**Table 2:** Considerations when interfacing 1-Wire device

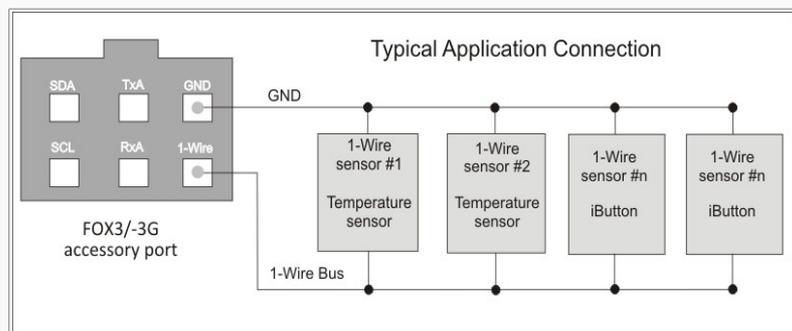
There are 2 types of cables (the CA71 or CA69 cable from FALCOM) that can be used to connect the sensors and iButtons to the FOX3/-3G device. The one you choose will depend on your application, beside the option that the customer is using its own cable. FALCOM does not provide any 1-Wire sensor or iButton. For more details refer to the "Falcom\_OrderingGuide.pdf".

### 1.4 What are the typical applications?

- Identification of both vehicles and drivers
- Authorized driving
- Recording working time of drivers (mobile workers)
- Identification of both private and business rides on vehicles
- Identification of staff management for rescue services
- Monitoring the temperature of the refrigerated trucks
- Monitoring temperature-sensitive loads
- Monitoring the temperature of solar panels
- Monitoring the temperature of batteries i.e. for electrical vehicles

### 1.5 How does the 1-Wire interface work?

A 1-Wire network consists of a master controller which is connected to one or some slave devices. The 1-Wire interface on the FOX3/-3G is a master controller while the i-Button readers and temperature sensors are slave devices. The FOX3/-3G communicates with these slave devices via 1-Wire protocol developed by Dallas Semiconductor, by sending and receiving signals over a single data line plus ground reference. The FOX3/-3G initiates and requests all activities on the 1-Wire bus. Each sensor connected to the FOX3/-3G identifies itself by a unique code identifier stored on-chip ROM (Read Only Memory). This identifier enables the FOX3/-3G to recognize individually each of these sensors without conflict.



**Figure 2:** 1-Wire typical application connection

## 1.6 Installing 1-Wire devices

Since they require only a single data line (DQ) plus ground (GND) reference to communicate with the 1-Wire interfaces, 1-Wire devices are very easy to setup. Most 1-Wire devices do not even have a VDD pin. They draw the current for operation from the data line (DQ). Some other 1-Wire devices have a VDD pin which should be connected either to the ground of the sensor called "parasitic mode" or externally powered called "normal mode". Please note that not all 1-Wire sensors can operate in parasitic mode. Please refer to the manufacturer's datasheet.

### 1.6.1 Parasite power mode (recommended by FALCOM)

In parasite power mode, which is also tested and recommended by FALCOM, the 1-wire sensors requires only one data line (and ground) for communication with FOX3/-3G. In this mode, the data line (DQ) of these sensors must be connected to the 1-Wire bus, the power line (VDD) must be connected to the ground (GND) at the sensor and the ground (GND) of the sensor goes to the ground of the FOX3/-3G device. In addition, the sensors derives power directly from the data line (DQ), eliminating the need for an external power supply.

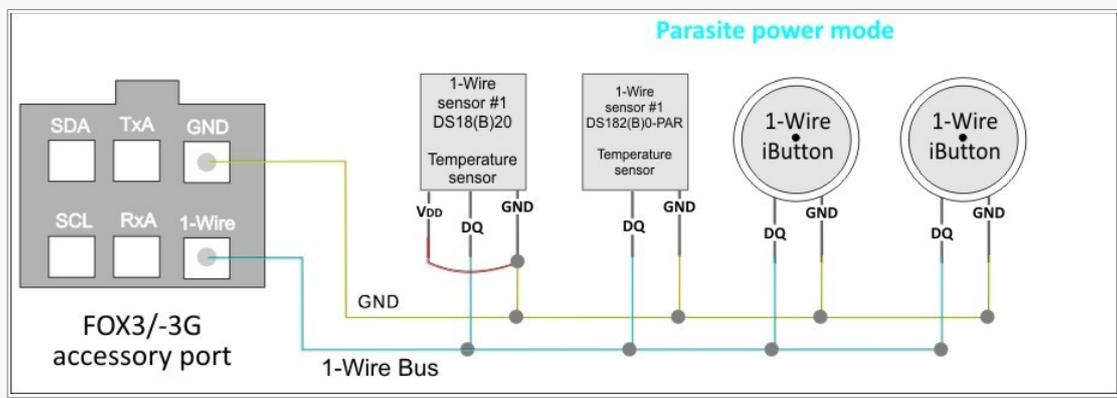


Figure 3: Parasite power mode (1-Wire temperature sensor with VDD pin locally connected to GND).

This current needed to power a 1-wire sensor is usually very small, however may rise up to 1.5 mA when making a temperature conversion and sending the data to the bus. Please note that, the maximum output current into the 1-Wire interface is limited to 20mA.

### 1.6.2 External supply mode

When operating in this mode the sensors requires all three wires of the external sensors. The power line (VDD) derives its power supply +5V externally. The data line and ground of the sensor should be connected to the 1-Wire interface and ground of the FOX3/-3G respectively. **This mode is not tested by FALCOM.**

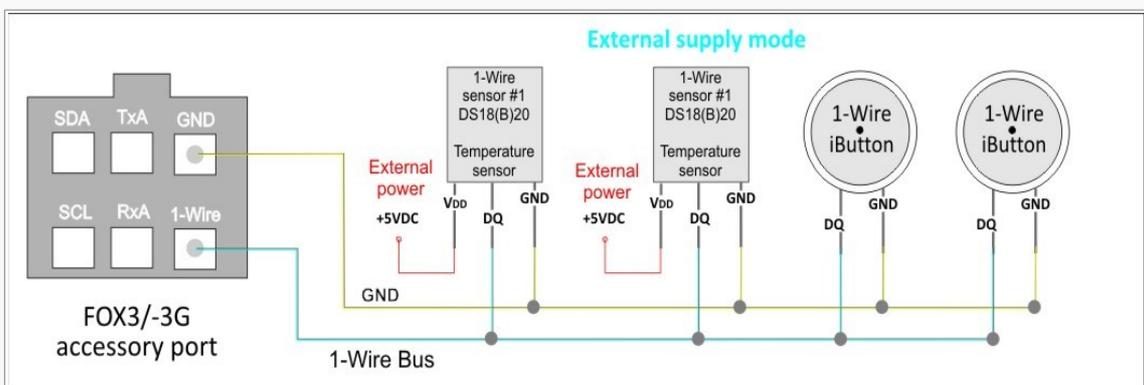


Figure 4: Normal mode (1-Wire temperature sensor with VDD pin connected to external power source).

## 1.7 Testing your 1-Wire connection

After connecting the 1-Wire sensors to the 1-Wire bust, use these commands to test them:

Activating 1-Wire interface:	
Execute	Response
\$PFAL, Sys.1Wire.Enable	\$<SYS.1WIRE.Enable> \$SUCCESS \$<end>
<i>At the same time the following events are generated showing the sensors currently registered:</i>	
\$GPEVENT:SYS.1Wire.eRegister=10c2272e00000E1 \$GPEVENT:SYS.1Wire.eRegister=10c2272e000001E0	
<i>The register and release events for the iButtons are generated once the user connects and releases the iButtons to/from the iButton reader.</i>	
\$GPEVENT:SYS.1Wire.eRegister=2a000006a22b1d09 \$GPEVENT:SYS.1Wire.eRegister=2a000006a22b1126 \$GPEVENT:SYS.1Wire.eRelease=2a000006a22b1d09 \$GPEVENT:SYS.1Wire.eRelease=2a000006a22b1126	
After activation of this interface you can also use the command below to get their IDs:	
\$PFAL, Sys.1Wire.Devices	\$<SYS.1WIRE.Devices> \$devices 10c2272e00000E1, 10c2272e000001E0 \$SUCCESS \$<end:1427883877623>
Reading the temperature data from the connected sensors:	
\$PFAL, Sys.1Wire.Temperature	<Sys.1Wire.Temperature> \$device 10c2272e00000E1 value 20.0`C \$device 10c2272e000001E0 value 21.0`C \$SUCCESS \$<end>
<i>In this example, 20.0`C and 21.0`C are the temperature data reported from the external sensors.</i>	

## 1.8 Set up automatic 1-Wire alarm reporting

The following table shows some application scenarios:

**EXAMPLE 1:** Reporting via TCP the temperature data in (°C) of the specific sensor every 5 minutes. For Fahrenheit or Kelvin applications, a conversion routine must be implemented on the server side.

```
$PFAL,Cnf.Set,AL3=Sys.Device.eStart:Sys.Timer1.Start=cyclic,30000&Sys.1Wire.Enable
$PFAL,Cnf.Set,AL4=Sys.Timer.e1&SYS.1Wire.sAvailable="10c2272e00000E1":TCP.Client.Send,8,"&(1WIRE.Temp=10c2272e00000E1) "
```

**EXAMPLE 2:** Driver identification via iButtons only. To get the iButton's ID, just touch the iButton to the reader and the event generated will have its ID.

```
$PFAL,Cnf.Set,AL5=SYS.1Wire.eRegister="2a000006a22b1d09":TCP.Client.Send,8,"Driver
ID=2a000006a22b1d09"&Sys.Trigger0=low
```

An personal iButton is assigned to each driver and an iButton Reader is installed on the vehicle's dashboard. The IDs of the personal iButtons is entered into the SYS.1Wire.eRegister event and when the driver connects his own personal iButton to the iButton reader, the corresponding event is generated and the ID of this personal iButton is then transmitted to the remote server.

**EXAMPLE 3:** Authorized driving via iButtons (allows ONLY authorized drivers to start the vehicle). To get the iButton's ID, just touch the iButton to the reader and the event generated will have its ID. (Not tested by FALCOM)

```
$PFAL,Cnf.Set,AL5=SYS.1Wire.eRegister=WhiteList:IO5.Set=high&TCP.Client.Send,8,"Veh
icle ID 3456 started"&Sys.Trigger0=low
$PFAL,Cnf.Set,AL6=IO.e8=fedge:IO5.Set=low
```

### To test this configuration (AL5 & AL9), follow the steps below:

Store the iButton IDs, which authorized to have access to this vehicle, into the Whitelist with `$PFAL, Sys.WhiteList.Set<index>="ID"` and connect the IO2 (pin 5) from the main port of the FOX3/-3G through a relay to the vehicle's ignition switch to allow/prevent the vehicle from starting up. When the driver inserts the key into the ignition switch and turns it to the position 3 (START), the engine will only start after connecting one of the authorized iButton ID to the reader. Whenever the driver turns off the vehicle engine (AL6), the relay will deactivate and can be re-activated only by one of the authorized iButton IDs.

**Hint:** As long as the IO2 is low, the relay deactivates the ignition circuit and prevents the vehicle engine from being started. When the IO2 goes high, the relay activates the ignition circuit and the vehicle engine can then be started. If the driver forgets or loses his personal iButton, the command `(PFAL, IO5.Set=high)` should be sent remotely to the FOX3/-3G device to allow the driver to start the vehicle. If you use this configuration, a relay (choose one specifically designed for automobiles) and the IO3 (IGN pin on the FOX3/-3G device) need additionally to be installed to the ignition circuit. The IO3 (IGN pin) is used here (in AL6) to detect if vehicle engine is turned off. The IO3 (IGN pin) can be connected to the position 3 (START) of the ignition circuit that shows 12V when the vehicle engine is started and 0V when turned off. Please refer to your vehicle wiring diagram for specific wire information. The command `Sys.Trigger0=low` is used here for AL9 below (if not used then remove all Triggers and the AL9 too).

**EXAMPLE 4:** Recording the working time of the drivers via iButtons and at the same time the location of the vehicle:

```
$PFAL,Cnf.Set,AL7=SYS.1Wire.eRegister="2a000006a22b1d09":TCP.Client.Send,8," Driver
ID=2a000006a22b1d09 IN: &(Date) &(Time) "&Sys.Trigger0=high
```

```
$PFAL,Cnf.Set,AL8=SYS.SYS.1Wire.eRelease="2a000006a22b1d09":TCP.Client.Send,8,"
Driver ID=2a000006a22b1d09 OUT: &(Date) &(Time) "&Sys.Trigger0=low
```

```
$PFAL,Cnf.Set,AL9=Sys.Trigger.s0=low&IO.e8=redge:IO4.Set=cyclic,1000,3000
```

The iButton reader is installed on the vehicle's dashboard closed to the driver and the driver plugs the personal iButton into the iButton reader to clock in (AL7) and release it to clock out (AL8). Optional (commands in gray), if the driver forgets to plug in the iButton to the reader and the vehicle's ignition is turned on, a buzzer constantly sounds (AL9) as a reminder. If AL9 is used, the IGN wire (pin 3) on FOX3's main port can be connected to the vehicle's ignition and the IO1 (pin 4) to a buzzer, which should additionally be installed. The IO3 (IGN pin) is used here (in AL9) to detect if vehicle engine is turned on. It should show 12V when the vehicle engine is started and 0V when turned off. The power terminal of the buzzer is connected to the IO1 (pin 4) on the FOX3/-3G device and the ground of the buzzer is connected to the one of the grounds of the FOX3/-3G device.