

Optional Package 1WIRE

Paket OPT_OW

V. 1.0.0

für Fli4L - 3.4.0

Karl M. Weckler
news4kmw@web.de
Rockenberg, Mai 2009

Inhaltsverzeichnis

Paket OPT_OW	1
Inhaltsverzeichnis	2
Dokumentation des Paketes OPT_OW	3
1. Einleitung.....	3
1.1. OPT_OW	3
1.2. OW = OneWire = 1-Wire	3
1.2.1. Der 1-Wire Standard.....	3
1.2.2. 1-Wire-Bauteile.....	3
1.2.3. 1-Wire Bus.....	3
1.3. OWFS	4
1.4. Fuse.....	4
2. Lizenz	4
3. Gewährleistungs- und Haftungsausschluss.....	4
4. Installation	5
5. Kompilation, Voraussetzungen	5
6. Konfiguration	5
6.1. OPT_OW	5
6.1.1. owfs	5
6.1.2. owshell	6
6.1.3. owhttpd	6
6.1.4. Rechtevergabe	6
6.1.5. Allgemein.....	6
6.1.6. Bedienung im Browser	6
7. Feedback.....	7
Anhang	8
A I. Pin-, Adernbelegung für TP-Kabel und RJ-45	8
A II. Family Code Referenz (aus MAXIM AppNote 155).....	9
A III. Links zum Thema 1-Wire	10
A IV. Bezugsquellen für 1-Wire Bauteile.....	10
A V. Schaltplanskizzen	10
A VI. Notizen.....	11
A VII. Man-Pages	12

Dokumentation des Paketes OPT_OW

1. Einleitung

1.1. OPT_OW

Dieses Paket installiert das OWFS [1.3.] und bietet so lesenden und schreibenden Zugriff auf einen an den Fli4L angeschlossenen 1-Wire-Bus. Hierzu wird z.B. ein 1-Wire Busmaster (DS 9097U-009, DS 9097U-E25 oder DS 9097U-S09 - Universal 1-Wire COM Port Adapter¹) an eine serielle Schnittstelle des PC gesteckt. An die 1-Wire-seitige RJ-11 Buchse wird dann der 1-Wire Bus angelegt.

1.2. OW = OneWire = 1-Wire

1.2.1. Der 1-Wire Standard

Der **1-Wire**® bzw. One-Wire oder Eindraht-Bus von **MAXIM** (MAXIM hat 2001 Dallas Semiconductor übernommen, bei der die 1-Wire Technologie entwickelt wurde) beschreibt eine serielle Schnittstelle, die mit einer Datenader auskommt, die sowohl als Stromversorgung als auch als Sende- und Empfangsleitung genutzt wird. Gleichwohl ist jedoch eine „Rückleitung“ (GND) erforderlich. Jeder 1-Wire Chip hat eine unikale Identnummer über die er angesprochen wird. So können mehrere (viele) 1-Wire Geräte an einem einzelnen Bus betrieben werden.

1.2.2. 1-Wire-Bauteile

MAXIM bietet eine Vielzahl von **1-Wire-Bauteilen** an, als da sind: Serielle-, USB-, I2C-Adapter, Thermometer, Schalter (bis 8 Kanäle), EEPROMs, Uhren, A/D-Wandler, digitale Potentiometer, u.v.a. Man bekommt eigentlich alles, was man so für die Hausautomation braucht. Eine Übersicht findet sich im Anhang unter A II.

An den Bus können auch **iButton® Bauteile** (NV-RAM, EPROM, EEPROM, Temperatur, Feuchtigkeit, RTC, SHA, div. Logger) angeschlossen werden.

1.2.3. 1-Wire Bus

Der 1-Wire Bus besteht im Prinzip aus zwei verdrehten Leitungen. Meist wird aber für längere Strecken (bis 150 m sollen, unter Beachtung entsprechender Topologie, kein Problem sein) normales Twisted-Pair-Ethernetkabel (Cat 5) genommen.

Zur Belegung der einzelnen Adern gibt es unterschiedliche Ansätze. MAXIM benutzt RJ-11 (6-polige) Buchsen und Stecker und hat einen eigenen Standard kreiert, der jedoch nicht zum 8-poligen RJ-45 Steckzeug passt. Hierüber gibt es aber auch einen **quasi Standard**. Ich persönlich bevorzuge eine Weiterentwicklung², die mit den anderen „Standards“ kompatibel ist.

¹ The DS9097U is a sophisticated RS232 to 1-Wire® adapter, which performs RS232 level conversion and actively generates the 1-Wire communication signals. Together with an adequate TMEX software driver, it enables an IBM®-compatible PC to directly communicate with any 1-Wire device connected to the adapter's RJ-11 port. The DB-9 version reads and writes any non-EPROM 1-Wire device. To write to EPROM devices, the DB-25 version with external power supply is required.

All DS9097U adapters are based on the DS2480B Serial 1-Wire Line Driver chip and therefore are not a plug-in replacement for the DS9097 or DS9097E. The RXD/TXD Polarity Select input of the DS2480B is tied to V_{DD} (non-inverting).

² Anhang AI

Auch zum Thema [Topologie](#) des Busses wird man bei MAXIM fündig, wie überhaupt deren [Webseite](#) alles Nötige bietet, um gut mit 1-Wire zu recht zu kommen.

1.3. OWFS

OWFS steht für „**O**ne **W**ire **F**ile **S**ystem“. Dabei handelt es sich um eine von [Paul H. Alfille](#) et al. entwickelte Software die unter der GPL lizenziert ist. Auf Grundlage einer 1-Wire-Protokoll verständigen Systembibliothek (OWLib) bildet OWFS den 1-Wire Bus wahlweise als Dateisystem (owfs), in Form eines tcp/ip Servers (owserver - owshell), oder eines http Servers (owhttpd) ab. Darüber hinaus bietet das Programm noch weitere Implementierungen (owftpd, owtap, Sprachmodule für capi, perl, tcl, php) die jedoch in der hier vorliegenden Anpassung für Fli4L nicht berücksichtigt wurden. Alles Weitere zu OWFS und viel Interessantes zu 1-Wire findet man auf: <http://owfs.org/> und <http://sourceforge.net/projects/owfs/>.
Eingesetzte Version: OWFS - 2.7p13, vom 05.01.2009.

1.4. Fuse

FUSE steht für „**F**ilesystem in **u**space“. Fuse ermöglicht die Implementierung eines voll funktionsfähigen Dateisystems im Userspace.
Da Fuse *nur* für „owfs“ benötigt wird, wird es *nur* bei dessen Installation als Kernelmodul zugeladen.
Alles weitere zu Fuse findet man auf:
<http://fuse.sourceforge.net/> und <http://sourceforge.net/projects/fuse/>.
Eingesetzte Version: Fuse - 2.7.4, vom 25.07.2008.

2. Lizenz

Dieses Programm ist freie Software. Sie können es unter den Bedingungen der GNU General Public License, wie von der Free Software Foundation herausgegeben, weitergeben und / oder modifizieren, entweder unter Version 2 der Lizenz oder (wenn Sie es wünschen) einer späteren Version.
Eine Kopie der GNU General Public License liegt diesem Programm bei (src/1wire/gpl.txt). Der Text der GNU General Public License ist auch im Internet unter <http://www.gnu.org/licenses/gpl.txt> veröffentlicht.
Eine inoffizielle deutsche Übersetzung findet sich unter <http://www.gnu.de/gpl-ger.html>. Diese Übersetzung soll nur zu einem besseren Verständnis der GPL verhelfen, rechtsverbindlich ist alleine die englischsprachige Version.

3. Gewährleistungs- und Haftungsausschluss

Die Veröffentlichung dieses Programms erfolgt mit dem Willen und der Hoffnung, dass es Ihnen von Nutzen sein wird. Dennoch wird jegliche Gewährleistung - auch die implizite Gewährleistung der Marktreife oder der Eignung für einen bestimmten Zweck - abgelehnt. Details hierzu finden Sie in der GNU General Public License (GPL).
Für Datenverlust, Schäden an Hard- oder Software oder sonstige Schäden übernimmt der Autor **keine Haftung**.

4. Installation

- das Paket ins Fli4I-Verzeichnis entpacken
- /config/ow.txt anpassen [6.]
- Bootmedium neu erstellen oder Remoteupdate

5. Kompilation, Voraussetzungen

Die Binaries und Libs wurden im fbr-3.0.0 unter Verwendung der 2.6.16.56er Kernelquellen erstellt. Für Fuse kommt das beim 2.6er Kernel mitgelieferte Modul „fuse.ko“ zum Einsatz. Das Opt wurde unter Kernel 2.4 noch nicht getestet. Sachdienliche Hinweise an den Autor erbeten.

Die mitgelieferte Fli4I-httpd Applikation basiert auf OWFS (choice 1. in /config/ow.txt). Für die Anzeige im Browser ist der Mini-Webserver (Fli4I-httpd) erforderlich. Weitere Hinweise hierzu unter 6.1.6.

6. Konfiguration

Die folgenden Variablen in der /config/ow.txt sind ggf. zu konfigurieren:

6.1. OPT_OW

Die Standardeinstellung von `OPT_OW='no'`, d.h. das Paket wird nicht installiert.

Mit `OPT_OW='yes'` wird das Paket aktiviert.

Sie können dann unter den drei zur Verfügung stehenden Möglichkeiten wählen, in jeder beliebigen Kombination.

6.1.1. owfs

Die Auswahl `ow_owfs='yes'` erzeugt ein Dateisystem unter dem unter `ow_owfs_path` angegebenen Pfad. Standardeinstellung ist '/mnt/ow'. Mit der Variablen `ow_owfs_dev` legen Sie die Schnittstelle fest, an der der COM-zu-1wire-Adapter angeschlossen wird, /dev/ttyS0 entspricht COM1, /dev/ttyS1 entspricht COM2, usw.

Die im Dateisystem angelegten Verzeichnisse und Dateien sind nach Identnummern³ der Chips geordnet. Über den family code der Bauteile ist eine entsprechende Systematik leicht herzustellen.

Mit z.B.:

- `cat /mnt/ow/10.DEF0A8010800/temperature`
rufen sie die Temperaturmessung eines DS19S20 ab.
- `echo "1" > /mnt/ow/1C.7F6CF7040000/PIO.O`
schaltet den Ausgang 1 eines DS28E04-100 (dual switch) ein.
- `echo "0" > /mnt/ow/1C.7F6CF7040000/PIO.O`
schaltet den Ausgang 1 wieder aus.

Eine weitere Beschreibung gibt es im Anhang „A VII.“ und hier:

<http://owfs.org/index.php?page=owfs>

³ Anhang AII

6.1.2. owshell

Die OW-Shell Programme bestehen u.a. aus einem tcp-server (owserver), der standardmäßig auf Port 4304 (IANA) aktiv ist.

Die Auswahl `ow_owshell='yes'` aktiviert den owserver und installiert die Programme `owdir`, `owread`, `owwrite` und `owget`.

Die Auswahl der Schnittstelle erfolgt wie unter 6.1.1 beschrieben, default ist: `ow_owshell_dev='/dev/ttyS0'`

Mit z.B.:

- `owdir`
wird die Dateistruktur des Busses ausgelesen.
- `owread /1C.7F6CF7040000/PIO.1`
wird der Status von Ausgang 2 des DS28E04 ausgegeben.
- `owwrite /1C.7F6CF7040000/PIO.1 1`
wird Ausgang 2 des DS28E04 auf 1 „on“ geschaltet.

Obige Aufrufe gelten nur in Verbindung mit dem Standardport, ansonsten ist der Port explizit anzugeben (z.B. `-s 4711`). Sonstige Parameter und Hilfe zeigt ein „ow... -h“. Eine weitere Beschreibung gibt es im Anhang „A VII.“ und hier:

<http://owfs.org/index.php?page=owshell>

6.1.3. owhttpd

Der owhttpd ist ein kleiner Webserver, der den 1-Wire Bus listet.

Einstellungen analog 6.1.2.

Hier ist unter `ow_owhttpd_port='8001'` der http-Port 8001 vorgewählt. Es kann aber auch jeder andere, freie Port gewählt werden.

Mit jedem Browser kann dann mit: „Hostname:Port“ der owhttpd abgefragt werden.

Eine weitere Beschreibung gibt es im Anhang „A VII.“ und hier:

<http://owfs.org/index.php?page=owhttpd>

6.1.4. Rechtevergabe

- Wird, falls erforderlich, später implementiert. -

6.1.5. Allgemein

Wenn Sie Daten loggen wollen, können Sie das extern oder auch auf dem Fli4l tun. Im letzteren Fall benötigen Sie dann eine Festplatte bzw. Flashkarte. Näheres dazu unter OPT_HD. Zum Sichern der Dateien sollte man OPT_SARFILE oder OPT_CPMVRMLOG ins Auge fassen. Die Daten lassen sich auch optisch aufbereiten, siehe OP_RRDTOOL. Das Ganze kann man mit EASYCRON auch zeitlich automatisieren.

6.1.6. Bedienung im Browser

Der in Fli4l optional zu installierende httpd (Webserver) bietet die Möglichkeit, eigene Shell/CGI Skript Anwendungen über jeden Browser im Netz zu bedienen. Davon wurde im vorliegenden Falle Gebrauch gemacht.

Im OPT_OW wird eine kleine, rudimentäre Browser Applikation mitgeliefert. Sie wird nur installiert, wenn in `/config/ow.txt` `OW_OWFS="yes"` gesetzt wird, da sie mit „owfs“ zusammen arbeitet. Die Anwendung kann nach eigenen Wünschen und Bedürfnissen an die zu überwachende, resp. zu steuernde Anwendung angepasst werden. Sie kann damit als Grundlage für weitere Eigenentwicklungen verwendet werden. Der

wesentliche Vorteil der Anwendung ist darin zu sehen, dass auch umfangreiche und komplexe Steuerungen unter Verwendung von Shell/CGI Skripten auf bestehender Hardware möglich sind.

Das Skript befindet sich gemäß Norm auf dem Fli4I unter /srv/www/admin/ow.cgi und im Fli4I-Installationsverzeichnis unter fli4I-version\opt\files\srv\www\admin\ow.cgi. Es kann wunschgemäß entweder mittels z.B. „WinSCP“ auf dem Fli4I selbst, oder offline z.B. im Fli4I-Installationsverzeichnis editiert werden.

(n.b.: Über die Auslagerung von 1-Wire-Device spezifischen Funktionen in eine separate Funktionsbibliothek kann zu gegebener Zeit nachgedacht werden.)

7. Feedback

Ich freue mich über jede, auch kurze Rückmeldung, selbst wenn das Paket ohne jegliche Probleme laufen sollte ;-).

Viel Spaß mit 1-Wire wünscht Ihr und Euer

Karl M. Weckler
news4kmw@web.de

Anhang

A I. Pin-, Adernbelegung für TP-Kabel und RJ-45

Der IPS-Entwurf⁴ für die RJ45 Pin-Belegung basiert auf einem Standardisierungsvorschlag, der hier zu finden ist: 1Wire.org.

Die folgende Definition ist eine Erweiterung des genannten Standards und steht nicht im Widerspruch zu diesem.

Farbangaben nach:

EIA/TIA 568A		EIA/TIA 568B
1 Grün/Weiß	Hauptversorgung GND	Orange/Weiß
2 Grün	Hauptversorgung +5V/50mA für 1-Wire Devices	Orange
3 Orange/Weiß	Sekundärer 1-Wire Bus GND	Grün/Weiß
4 Blau	Primärer 1-Wire Bus	Blau
5 Blau/Weiß	Primärer 1-Wire Bus GND	Blau/Weiß
6 Orange	Sekundärer 1-Wire Bus	Grün
7 Braun/Weiß	Hilfsversorgung z.B. +12V/200mA für andere Verbraucher	Braun/Weiß
8 Braun	Hilfsversorgung GND	Braun

Hinweis: Bei langen Busstrecken wirkt sich der ohmsche Widerstand negativ auf die Versorgungsspannungen aus. Insofern ist, soweit möglich, eine anwendungsseitige Stromversorgung zu bevorzugen.

⁴ IPS steht für IP-Symcon, einem Software-Produkt der Fa. Computer- und Steuerungstechnik Steiner in Sarkwitz. <http://www.ip-symcon.de/>

A II. Family Code Referenz (aus MAXIM AppNote 155)

Family Code (hex)	Bauteil / Chip	Beschreibung (Speichergröße in bits, wnaa) 1-Wire net address (registration number) only
01	DS2401, DS2411	
02	(DS1991)*	Multikey iButton, 1152-bit secure memory
04	(DS1994), DS2404	4Kb NV RAM memory and clock, timer, alarms
05	DS2405*	Single addressable switch
06	(DS1993)	4Kb NV RAM memory
08	(DS1992)	1Kb NV RAM memory
09	(DS1982), DS2502	1Kb EPROM memory
0A	(DS1995)	16Kb NV RAM memory
0B	(DS1985), DS2505	16Kb EPROM memory
0C	(DS1996)	64Kb NV RAM memory
0F	(DS1986), DS2506	64Kb EPROM memory
10	(DS1920)	Temperature with alarm trips
12	DS2406, DS2407*	1Kb EPROM memory, 2-channel addressable switch
14	(DS1971), DS2430A*	256-bit EEPROM memory and 64-bit OTP register
1A	(DS1963L)*	4Kb NV RAM memory with write cycle counters
1C	DS28E04-100	4096-bit EEPROM memory, 2-channel addressable switch
1D	DS2423*	4Kb NV RAM memory with external counters
1F	DS2409*	2-channel addressable coupler for sub-netting
20	DS2450	4-channel A/D converter (ADC)
21	(DS1921G), (DS1921H), (DS1921Z)	Thermochron® temperature logger
23	(DS1973), DS2433	4Kb EEPROM memory
24	(DS1904), DS2415	Real-time clock (RTC)
27	DS2417	RTC with interrupt
29	DS2408	8-channel addressable switch
2C	DS2890*	1-channel digital potentiometer
2D	(DS1972), DS2431	1024-bit, 1-Wire EEPROM
37	(DS1977)	Password-protected 32KB (bytes) EEPROM
3A	(DS2413)	2-channel addressable switch
41	(DS1922L), (DS1922T), (DS1923), DS2422	High-capacity Thermochron (temperature) and Hygrochron™ (humidity) loggers
42	DS28EA00	Programmable resolution digital thermometer with sequenced detection and PIO
43	DS28EC20	20Kb 1-Wire EEPROM

Die Liste erhebt keinen Anspruch auf Vollständigkeit

* Diese Chips werden nicht mehr für Neuentwicklungen empfohlen.

A III. Links zum Thema 1-Wire

http://www.maxim-ic.com/auto_info.cfm/

<http://www.1wire.org/>

<http://www.simat.org.uk/>

<http://owfs.org/>

<http://sourceforge.net/projects/owfs/>.

<http://fuse.sourceforge.net/>

<http://sourceforge.net/projects/fuse/>.

A IV. Bezugsquellen für 1-Wire Bauteile

http://www.maxim-ic.com/auto_info.cfm

MAXIM verschickt auch kostenlose Muster (Sample)

<http://www.fuchs-shop.com/>

<http://www.spezial.com/>

<http://www.1-wire.de/>

<http://www.reichelt.de/>

Das 1-Wire Angebot ist eher dürftig, aber sonst ☺

<http://www.homechip.com/catalog/>

<http://www.aagelectronica.com/aag/index.html>

http://www.hobby-boards.com/catalog/main_page.php

<http://www.tme.eu/de/>

A V. Schaltplanskizzen

- folgen nach -

A VI. Notizen

A VII. Man-Pages

NAME

owfs – 1-wire filesystem

SYNOPSIS

owfs *-m mountdir*

DESCRIPTION**owfs**

owfs (1) is the filesystem client of the *OWFS* family of programs. It runs on linux, freebsd and Mac OS X, and requires the *fuse* kernel module and library. (<http://fuse.sourceforge.net>) which is a user-mode filesystem driver.

Essentially, the entire 1-wire bus is mounted to a place in your filesystem. All the 1-wire devices are accessible using standard file operations (read, write, directory listing). The system is safe, no actual files are exposed, these files are virtual. Not all operations are supported. Specifically, file creation, deletion, linking and renaming are not allowed. (You can link from outside to a owfs file, but not the other way around).

SPECIFIC OPTIONS**-m --mountpoint=directory_path**

Path of a directory to mount the 1-wire file system

The mountpoint is required. There is no default.

--allow_other

Shorthand for fuse mount option "-o allow_other" Allows other users to see the fuse (owfs) mount point and file system. Requires a setting in /etc/fuse.conf as well.

--fuse-opt options

Sends options to the fuse-mount process. Options should be quoted, e.g. "

EXAMPLE

owfs -d /dev/ttyS0 -m /mnt/1wire

Bus master on serial port

owfs -F -u -m /mnt/1wire

USB adapter, temperatures reported in Fahrenheit

owfs -s 10.0.1.2:4304 -m /mnt/1wire

Connect to an **owserver (1)** process that was started on another machine at tcp port 4304

SEE ALSO**AVAILABILITY**

<http://www.owfs.org>

AUTHOR

Paul Alfille (paul.alfille@gmail.com)

NAME

owdir owread owwrite owpresent – lightweight **owserver** access

SYNOPSIS

```

owdir -s [host:]port [directory]
owread -s [host:]port filepath
owwrite -s [host:]port filepath value
owget -s [host:]port [directory] | filepath

owdir --autoserver [directory]
owread --autoserver filepath
owwrite --autoserver filepath value
owget --autoserver [directory] | filepath

owdir -f --format f[.i[.[.c] -s [host:]port [directory] [directory2 ...]
owread -C --celsius -K --kelvin -F --fahrenheit -R --rankine [ --hex ] [ --start= offset ] [ --size= bytes ] -s
[host:]port filepath [filepath2 ...]
owwrite -C --celsius -K --kelvin -F --fahrenheit -R --rankine [ --hex ] [ --start= offset ] -s [host:]port
filepath value [filepath2 value2 ...]

owget -f --format f[.i[.[.c] -C --celsius -K --kelvin -F --fahrenheit -R --rankine [ --hex ] [ --start= offset ] [
--size= bytes ] -s [host:]port [directory] | filepath

owdir -V --version
owread -V --version
owwrite -V --version

owdir -h / --help
owread -h / --help
owwrite -h / --help

```

DESCRIPTION**OWSHELL programs**

owdir owread owwrite and **owget** are collectively called the **owshell** programs. They allow lightweight access to an **owserver** (1) for use in command line scripts.

Unlike **owserver** (1) **owhttpd** (1) **owftpd** (1) **owhttpd** (1) there is not persistent connection with the 1-wire bus, no caching and no multithreading. Instead, each program connects to a running **owserver** (1) and performs a quick set of queries.

owserver (1) performs the actual 1-wire connection (to physical 1-wire busses or other **owserver** programs), performs concurrency locking, caching, and error collection.

owshell programs are intended for use in command line scripts. An alternative approach is to mount an **owfs** (1) filesystem and perform direct file lists, reads and writes.

owdir

owdir performs a *directory* listing. With no argument, all the devices on the main 1-wire bus will be listed. Given the name of a 1-wire device, the available properties will be listed. It is the equivalent of

ls directory

in the **owfs** (1) filesystem.

owread

owread obtains for value of a 1-wire device property. e.g. 28.0080BE21AA00/temperature gives the DS18B20 temperature. It is the equivalent of

cat filepath

in the **owfs** (1) filesystem.

owwrite

owwrite performs a change of a property, changing a 1-wire device setting or writing to memory. It is the equivalent of

echo value > filepath

in the **owfs (1)** filesystem.

owget

owget (1) is a convenience program, combining the function of **owdir (1)** and **owread (1)** by first trying to read the argument as a directory, and if that fails as a 1-wire property.

STANDARD OPTIONS**--autoserver**

Find an *owserver* using the Service Discovery protocol. Essentially Apple's Bonjour (aka zeroconf). Only the first *owserver* will be used, and that choice is probably arbitrary.

-s [host:]port

Connect via tcp (network) to an *owserver* process that is connected to a physical 1-wire bus. This allows multiple processes to share the same bus. The *owserver* process can be local or remote.

DATA OPTIONS**--hex**

Hexidecimal mode. For reading data, each byte of character will be displayed as two characters 0-9ABCDEF. Most useful for reading memory locations. No spaces between data.

Writing data in hexidecimal mode just means that the data should be given as one long hexidecimal string.

--start=offset

Read or write memory locations starting at the offset byte rather than the beginning. An offset of 0 means the beginning (and is the default).

--size=bytes

Read up to the specified number of bytes of a memory location.

HELP OPTIONS**-h --help**

Shows basic summary of options.

-V --version

Version of this program and related libraries.

DISPLAY OPTIONS**-f --format f[.i][.c]**

Display format for the 1-wire devices. Each device has a 8 byte address, consisting of:

f family code, 1 byte

i ID number, 6 bytes

c CRC checksum, 1 byte

Possible formats are *f.i* (default, 01.A1B2C3D4E5F6), *fi* *fic* *f.ic* *f.i.c* and *fi.c*

All formats are accepted as input, but the output will be in the specified format.

EXAMPLE

owdir -s 3000 --format fic

Get the device listing (full 16 hex digits, no dots) from the local *owserver* at port 3000

owread -F --autoserver 51.125499A32000/typeK/temperature

Read temperature from the DS2751-based thermocouple on an auto-discovered *owserver* Temperature in fahrenheit.

```
owwrite -s 10.0.1.2:3001 32.000800AD23110/pages/page.1 "Passed"
```

Connect to a OWFS server process (*owserver*) that was started on another machine at tcp port 3001 and write to the memory of a DS2780

SEE ALSO**AVAILABILITY**

<http://www.owfs.org>

AUTHOR

Paul Alfille (palfile@earthlink.net)

NAME

owserver – Backend server (daemon) for 1-wire control

SYNOPSIS

owserver *-p* tcp-port

DESCRIPTION**owserver**

owserver (1) is the backend component of the OWFS 1-wire bus control system. **owserver** (1) arbitrates access to the bus from multiple client processes. The physical bus is usually connected to a serial or USB port, and other processes connect to **owserver** (1) over network sockets (tcp port). Communication can be local or over a network. Secure tunneling can be implemented using standard techniques.

Frontend clients include a filesystem representation: **owfs** (1) , and a webserver: **owhttpd** (1). Direct language bindings are also available, e.g: **owperl** (3). Several instances of each client can be initiated.

Each client can also connect directly to the physical bus, skipping **owserver** (1) but only one client can connect to the physical bus safely. Simultaneous access is prevented by the operating system for USB ports, but unfortunately not serial ports. The safe way to share access to the 1-wire bus is via **owserver** (1) with the clients connecting. Note: **owserver** (1) can connect to another **owserver** (1) process, though the utility of this technique is limited (perhaps as a *readonly* buffer?)

owserver (1) is by default multithreaded. Optional data caching is in the server, not clients, so all the clients gain efficiency.

SPECIFIC OPTIONS**-p**

TCP port or IPaddress:port for *owserver*

Other OWFS programs will access *owserver* via this address. (e.g. *owfs -s IP:port /1wire*)

If no port is specified, the default well-known port (4304 -- assigned by the IANA) will be used.

DEVELOPER OPTIONS**--no_dirall**

Reject DIRALL messages (requests directory as a single message), forcing client to use older DIR method (each element is an individual message)

--no_get

Reject GET messages (lets *owserver* determine if READ or DIRALL is appropriate). Client will fall back to older methods.

--no_persistence

Reject persistence in requests. All transactions will have to be new connections.

--pingcrazy

Interject many "keep-alive" (PING) responses. Usually PING responses are only sent when processing is taking a long time to inform client that *owserver* is still there.

EXAMPLE

owserver -p 3001 -d /dev/ttyS0 runs *owserver* on tcp port 3001 and connects to a physical 1-wire bus on a serial port.

SEE ALSO**AVAILABILITY**

<http://www.owfs.org>

AUTHOR

Paul Alfille (paul.alfille@gmail.com)

NAME

owhttpd – Tiny webserver for 1-wire control

SYNOPSIS

owhttpd *-p* tcp-port

DESCRIPTION**owhttpd**

owhttpd (1) is a small webserver that shows the Dallas/Maxim 1-Wire bus attached to a serial port. The main view shows the devices found, You can then navigate to individual devices, and view/alter their properties.

owhttpd (1) uses the same naming convention as **owfs** (1) , where the URL corresponds to the filename.

The web server is a modified version of chttpd by Greg Olszewski. It serves no files from the disk, only virtual files from the 1-wire bus. Security should therefore be good. Only the 1-wire bus is at risk.

SPECIFIC OPTIONS**-p portnum**

Sets the tcp port the web server runs on. Access with the URL `http://servernameoripaddress:portnum`

If no port is specified, an ephemeral port is selected by the operating system. Use *zeroconf* (*Bonjour*) to discover the assigned port.

EXAMPLE

`owhttpd -p 3001 -d /dev/ttyS0`

Web server runs on tcp port 3001, serial adapter at ttyS0

`owhttpd -p 3001 -s littlehost:4304 --error_level=3`

Web server on port 3001, from *owserver* process on host "littlehost", extensive error messages.

`owhttpd -p 3001 -u -u2 -r`

Read-only web server on port 3001, using two usb adapters.

AVAILABILITY

`http://www.owfs.org`

SEE ALSO**AUTHOR**

Paul Alfille (paul.alfille@gmail.com)

NAME

owfs.conf – owfs programs configuration file

SYNOPSIS

An OWFS configuration file is specified on the command line:

owfs -c config_file [other options]

The file name is arbitrary, there is no default configuration file used.

USAGE

A configuration file can be invoked for any of the OWFS programs (**owfs (1) owhttpd (1) owserver (1) owftpd (1)**) or any of the language bindings (**owperl (1) owcapi (1) owtbl (1) owphp owpython**) to set command line parameters.

SYNTAX

Similar to Unix shell script or perl syntax

Comments

Any # marks the start of a comment
blank lines are ignored

Options

option # some options (like 'foreground') take no values
option = value # other options need a value
option value # '=' can be omitted if whitespace separates
Option # Case is ignored (for options, not values)
opt # non-ambiguous abbreviation allowed
-opt --opt # hyphens ignored

owserver

server: opt = value # only *owserver* effected by this line
! server: opt = value # *owserver* NOT effected by this line

owhttpd

http: opt = value # only *owhttpd* effected by this line
! http: opt = value # *owhttpd* NOT effected by this line

owftpd

ftp: opt = value # only *owftpd* effected by this line
! ftp: opt = value # *owftpd* NOT effected by this line

owfs

owfs: opt = value # only *owfs* effected by this line
! owfs: opt = value # *owfs* NOT effected by this line

Limits

maximum line length of 250 characters
no limit on number of lines

DESCRIPTION**Configuration**

owfs.conf (5) allows a uniform set of command line parameters to be set.

Not all OWFS programs use the same command line options, but the non-relevant ones will be ignored.

Command line and configuration options can mixed. They will be invoked in the order presented. Left to right for the command line. Top to bottom for the configuration file.

Configuration files can call other configuration files. There is an arbitrary depth of 5 levels to prevent infinite loops. More than one configuration file can be specified.

SAMPLE

Here is a sample configuration file with all the possible parameters included.

Sources

device = /dev/ttyS0 # serial port: DS9097U DS9097 ECLO or LINK
device = /dev/i2c-0 # i2c port: DS2482-100 or DS2482-800

```

usb #    USB device: DS9490 PuceBaboon
usb = 2 #    Second DS9490
usb = all # All DS9490s
altUSB # Willy Robison's tweaks
LINK = /dev/ttyS0 #    serial LINK in ascii mode
LINK = [address:]port # LINK-HUB-E (tcp access)
HA7 # HA7Net autodiscovery mode
HA7 = address[:port] # HA7Net at tcp address (port 80)
etherweather = address[:port] # Etherweather device
server = [address:]port # owserver tcp address
FAKE = 10,1B # Random simulated device with family codes (hex)
TESTER = 28,3E # Predictable simulated device with family codes
#
# Sinks
# # owfs specific
mountpoint = filelocation # FUSE mount point
allow_other # Short hand for FUSE mount option "
# # owhttpd owserver owftpd specific
port = [address:]port # tcp out port
#
# Temperature scales
Celsius # default
Fahrenheit
Kelvin
Rankine
#
# Timeouts (all in seconds)
#          cache for values that change on their own
timeout_volatile = value # seconds "volatile" values remain in cache
#          cache for values that change on command
timeout_stable = value # seconds "stable" values remain in cache
#          cache for directory lists (non-alarm)
timeout_directory = value # seconds "directory" values remain in cache
#          cache for 1-wire device location
timeout_presence = value # seconds "device presence" (which bus)
timeout_serial = value # seconds to wait for serial response
timeout_usb = value # seconds to wait for USB response
timeout_network = value # seconds to wait for tcp/ip response
timeout_ftp = value # seconds inactivity before closing ftp session
#
# Process control
configuration = filename # file (like this) of program options
pid_file = filename # file to store PID number
foreground
background # default
readonly # prevent changing 1-wire device contents
write # default
error_print = 0-3 # 0-mixed 1-syslog 2-stderr 3-suppressed
error_level = 0-9 # increasing noise
#
# zeroconf / Bonjour
zero #    turn on zeroconf announcement (default)
nozero #    turn off zeroconf announcement
announce = name # name of announced service (optional)

```

```
autoserver # Add owservers discovered by zeroconf/Bonjour
noautoserver # Don't use zeroconf/Bonjour owservers (default)
#
# tcp persistence
timeout_persistent_low = 600 # minimum time a persistent socket will stay open
timeout_persistent_high = 3600 # max time an idle client socket will stay around
#
# Display
format = f[.i][[.c] # 1-wire address f amily i d code c rc
#
# Cache
cache_size = 1000000 # maximum cache size (in bytes) or 0 for no limit (default 0) #
# Information
# (silly in a configuration file)
version
help
morehelp
```

SEE ALSO**AVAILABILITY**

<http://www.owfs.org>

AUTHOR

Paul Alfille (palfile@earthlink.net)

NAME

DS18S20 – High-Precision 1-Wire Digital Thermometer **DS1920** – iButton version of the thermometer

SYNOPSIS

Thermometer.

10 [.]XXXXXXXXXXXX[XX][/[**die** | **power** | **temperature** | **temphigh** | **templo** | **trim** | **trimblanket** | **trimvalid** |]]

FAMILY CODE

10

SPECIAL PROPERTIES**power**

read-only, yes-no

Is the chip powered externally (=1) or from the parasitically from the data bus (=0)?

temperature

read-only, floating point

Temperature read by the chip at high resolution (~12 bits). Units are selected from the invoking command line. See **owfs(1)** or **owhttpd(1)** for choices. Default is Celsius. Conversion takes 1000 msec.

STANDARD PROPERTIES**DESCRIPTION****DS18S20 DS1920**

The **DS18S20 (3)** is one of several available 1-wire temperature sensors. It has been largely replaced by the **DS18B20 (3)** and **DS1822 (3)** as well as temperature/voltage measurements in the **DS2436 (3)** and **DS2438 (3)**. For truly versatile temperature measurements, see the protean **DS1921 (3) Thermachron (3)**.

ADDRESSING**DATASHEET**

<http://pdfserv.maxim-ic.com/en/ds/DS18S20.pdf>

SEE ALSO**AVAILABILITY**

<http://www.owfs.org>

AUTHOR

Paul Alfille (palfile@earthlink.net)

NAME**DS2401** - Silicon Serial Number**DS1990A** - Serial Number iButton**01** [.]XXXXXXXXXXXX[XX][/[]]**SYNOPSIS**

Unique serial number only.

FAMILY CODE

01

SPECIAL PROPERTIES

None.

STANDARD PROPERTIES**ALARMS**

None.

DESCRIPTION**DS2401 DS1990A**

The **DS2401 (3)** and **DS1990A (3)** are the most basic of 1-wire devices. Their sole property is it's unique address. It can be used for unique identification. Nonetheless, many keylocks, night watchman systems, and tracking systems use this device.

ADDRESSING**DATASHEET**<http://pdfserv.maxim-ic.com/en/ds/DS2401.pdf><http://pdfserv.maxim-ic.com/en/ds/DS1990A-F3-DS1990A-F5.pdf>**SEE ALSO****AVAILABILITY**<http://www.owfs.org>**AUTHOR**

Paul Alfille (palfile@earthlink.net)

NAME**DS2406 DS2407** - Dual Addressable Switch with 1kbit Memory**SYNOPSIS**

Dual Switch, Write-once Memory

12 [.]XXXXXXXXXXXX[XX]/[**channels** | **latch**.[A|B|ALL|BYTE] | **memory** | **pages/page**.[0-3|ALL] | **PIO**.[A|B|ALL|BYTE] | **power** | **sensed**.[A|B|ALL|BYTE] | **set_alarm** | TAI8570/[sibling|temperature|pressure] | T8A/volt.[0-7,ALL]]]

FAMILY CODE

12

SPECIAL PROPERTIES**channels***read-only, unsigned integer*

Is this a 1 or 2 channel switch? The *DS2406* comes in two forms, one has only one *PIO* pin (*PIO.A*). Returns 1 or 2.

latch.A latch.B latch.ALL latch.BYTE*read-write, yes-no*

The activity latch is set to 1 with the first negative or positive edge detected on the associated *PIO* channel. Writing any data will clear latch for all (both)) channels. This is a hardware "feature" of the chip.

ALL references both channels simultaneously, comma separated*BYTE* references both channels simultaneously as a single byte, with channel A in bit 0.**memory***read-write, binary*

128 bytes of non-volatile, write-once data.

pages/page.0 ... pages/page.3 pages/page.ALL*read-write, binary*

Memory organized as 4 pages or 32 bytes. Memory is write-once.

ALL is the aggregate of all 4 pages, sequentially accessed.**PIO.A PIO.B PIO.ALL PIO.BYTE***read-write, yes-no*State of the open-drain output (*PIO*) pin. 0 = non-conducting (off), 1 = conducting (on).

Writing zero will turn off the switch, non-zero will turn on the switch. Reading the *PIO* state will return the switch setting. To determine the actual logic level at the switch, refer to the *sensed* property.

ALL references both channels simultaneously, comma separated.*BYTE* references both channels simultaneously as a single byte, with channel A in bit 0.**power***read-only, yes-no*Is the *DS2406* powered parasitically =0 or separately on the *Vcc* pin =1**sensed.A sensed.B sensed.ALL sensed.BYTE***read-only, yes-no*

Logic level at the *PIO* pin. 0 = ground. 1 = high (~2.4V - 5V). Really makes sense only if the *PIO* state is set to zero (off), else will read zero.

ALL references both channels simultaneously, comma separated.*BYTE* references both channels simultaneously as a single byte, with channel A in bit 0.**set_alarm***read-write, unsigned integer (0-331)*

A number consisting of three digits XYZ, where:

X channel selection
 0 neither
 1 A only
 2 B only

	3 A or B
Y	source selection
	0 undefined
	1 latch
	2 PIO
	3 sensed
Z	polarity selection
	0 low
	1 high

All digits will be truncated to the 0-3 (or 0-1) range. Leading zeroes are optional (and may be problematic for some shells).

Example:

311 Responds on Conditional Search when either latch.A or latch.B (or both) are set to 1.

<100 Never responds to Conditional Search.

TAI8570/

subdirectory

Properties when the *DS2406 (3)* is built into a *TAI8570*.

If the *DS2406 (3)* is not part of a *TAI8570* or is not the controlling switch, attempts to read will result in an error.

TAI8570/pressure

read-only, floating point

Barometric *pressure* in millibar.

TAI8570/sibling

read-only, ascii

Hex address of the *DS2406 (3)* paired with this chip in a *TAI8570*.

TAI8570/temperature

read-only, floating-point

Ambient *temperature* measured by the *TAI8570* in prevailing temperature units (Centigrade is the default).

T8A/volt.[0-7|ALL]

read-only, floating-point

Uses the T8A (by *Embedded Data Systems*) 8 channel voltage convertor. Units in volts, 0 to 5V range. 12 bit resolution.

STANDARD PROPERTIES

ALARMS

Use the *set_alarm* property to set the alarm triggering criteria.

DESCRIPTION

DS2406

The **DS2406 (3)** allows control of other devices, like LEDs and relays. It superceeds the **DS2405** and **DS2407** Alternative switches include the **DS2408** or even **DS2450**

The **DS2407** is practically identical to the *DS2406* except for a strange *hidden* mode. It is supported just like the **DS2406**

TAI8570

The *TAI-8570 Pressure Sensor* is based on a 1-wire composite device by *AAG Electronica*. The *TAI8570* uses 2 *DS2406 (3)* chips, paired as a reader and writer to synthesize 3-wire communication. Only 1 of the *DS2406 (3)* will allow *temperature* or *pressure* readings. It's mate's address can be shown as *sibling*.

The *TAI8570* uses the *Intersema MS5534a* pressure sensor, and stores calibration and temperature compensation values internally.

Design and code examples are available from AAG Electronica <http://aag.com.mx>, specific permission to

use code in a GPL product was given by Mr. Aitor Arrieta of AAG and Dr. Simon Melhuish of OWW.

ADDRESSING**DATASHEET**

<http://pdfserv.maxim-ic.com/en/ds/DS2406.pdf>

<http://pdfserv.maxim-ic.com/en/ds/DS2407.pdf>

[http://www.aagelectronica.com/PDF%20Docs/\(TAI8570\)%20Pressure%20Sensor.pdf](http://www.aagelectronica.com/PDF%20Docs/(TAI8570)%20Pressure%20Sensor.pdf)

<http://www.embeddeddatasystems.com/page/EDS/PROD/IO/T8A>

<http://oww.sourceforge.net/hardware.html#bp>

SEE ALSO**AVAILABILITY**

<http://www.owfs.org>

AUTHOR

Paul Alfille (palfile@earthlink.net)

NAME**DS2408** - 1-Wire 8 Channel Addressable Switch**SYNOPSIS**

8 port switch

29 [.]XXXXXXXXXXXX[XX][/[**latch.[0-7|ALL|BYTE]** | **LCD_M/[clear|home|screen|message]** | **LCD_H/[clear|home|yxscreen|screen|message|onoff]** | **PIO.[0-7|ALL|BYTE]** | **power** | **sensed.[0-7|ALL|BYTE]** | **strobe** | **por** | **set_alarm** |]]

FAMILY CODE

29

SPECIAL PROPERTIES**latch.0 ... latch.7 latch.ALL latch.BYTE***read-write, binary*

The 8 pins (PIO) latch a bit when their state changes, either externally, or through a write to the pin.

Reading the *latch* property indicates that the latch has been set.Writing "true" (non-zero) to ANY *latch* will reset them all. (This is the hardware design).*ALL* is all *latch* states, accessed simultaneously, comma separated.*BYTE* references all channels simultaneously as a single byte. Channel 0 is bit 0.**LCD_H/clear LCD_M/clear***write-only, yes-no*Clears an LCD screen controlled by a *DS2408* using published designs.*clear* will initialize and clear the screen.*LCD_H* uses the Hobby-Boards design of Erik Vickery (see *DATASHEET*)*LCD_M* uses the Maxim public domain design (see *DATASHEET*)**LCD_H/home LCD_M/home***write-only, yes-no*Positions the cursor of an LCD screen controlled by a *DS2408* using published designs. The screen must have been initialized with the *clear* or *message* command first.*home* will set the cursor to the upper left position, but leave the current text intact.*LCD_H* uses the Hobby-Boards design of Erik Vickery (see *DATASHEET*)*LCD_M* uses the Maxim public domain design (see *DATASHEET*)**LCD_H/screen LCD_M/screen***write-only, ascii text*Writes to an LCD screen controlled by a *DS2408* using published designs. The screen must have been initialized with the *clear* or *message* command first.

Text will appear starting at the current location.

LCD_H uses the Hobby-Boards design of Erik Vickery (see *DATASHEET*)*LCD_M* uses the Maxim public domain design (see *DATASHEET*)**LCD_H/yxscreen***write-only, ascii text*Writes to an LCD screen controlled by a *DS2408* using published designs. Like *LCD_H/screen* but starting at a particular screen position.

2 binary bytes

The two first characters of the passed string have the line and row: e.g. "\x02\x04string" perl string writes "string" at line 2 column 4.

ascii 2,12:

Two numbers giving line and row: Separate with a comma and end with a colon e.g. "2,4:string" writes "string" at line 2 column 4.

ascii 12:

Single column number on the (default) first line: End with a colon e.g. "12:string" writes "string" at line 1 column 12.

The positions are 1-based (i.e. the first position is 1,1).

LCD_H/onoff

write-only, unsigned

Sets several screen display functions. The selected choices should be added together.

- 4 Display on
- 2 Cursor on
- 1 Cursor blinking

LCD_H/message LCD_M/message

write-only, ascii text

Writes to an LCD screen controlled by a *DS2408* using published designs.

message combines the actions of *clear*, *home* and *screen* so is the easiest way to display a message.

LCD_H uses the Hobby-Boards design of Erik Vickery (see *DATASHEET*)

LCD_M uses the Maxim public domain design (see *DATASHEET*)

PIO.0 ... PIO.7 PIO.ALL PIO.BYTE

read-write, yes-no

State of the open-drain output (*PIO*) pin. 0 = non-conducting (off), 1 = conducting (on).

Writing zero will turn off the switch, non-zero will turn on the switch. Reading the *PIO* state will return the switch setting. To determine the actual logic level at the switch, refer to the *sensed.0 ... sensed.7 sensed.ALL sensed.BYTE* property.

ALL references all channels simultaneously, comma separated.

BYTE references all channels simultaneously as a single byte. Channel 0 is bit 0.

power

read-only, yes-no

Is the *DS2408* powered parasitically (0) or separately on the Vcc pin (1)?

sensed.0 ... sensed.7 sensed.ALL

read-only, yes-no

Logic level at the *PIO* pin. 0 = ground. 1 = high (~2.4V - 5V). Really makes sense only if the *PIO* state is set to zero (off), else will read zero.

ALL references all channels simultaneously, comma separated.

BYTE references all channels simultaneously as a single byte. Channel 0 is bit 0.

strobe

read-write, yes-no

RSTZ Pin Mode Control. Configures RSTZ as either RST input or STRB output:

- 0 configured as RST input (default)
- 1 configured as STRB output

por

read-write, yes-no

Specifies whether the device has performed power-on reset. This bit can only be cleared to 0 under software control. As long as this bit is 1 the device will always respond to a conditional search.

set_alarm

read-write, integer unsigned (0-333333333)

A number consisting of 9 digits XXXXXXXXXX, where:

- X select source and logical term
 - 0 PIO OR
 - 1 latch OR
 - 2 PIO AND
 - 3 latch AND

Y select channel and polarity
0 Unselected (LOW)
1 Unselected (HIGH)
2 Selected LOW
3 Selected HIGH

All digits will be truncated to the 0-3 range. Leading zeroes are optional. Low-order digit is channel 0.

Example:

100000033

Responds on Conditional Search when latch.1 or latch.0 are set to 1.

222000000

Responds on Conditional Search when sensed.7 and sensed.6 are set to 0.

000000000 (0)

Never responds to Conditional Search.

STANDARD PROPERTIES

ALARMS

Use the *set_alarm* property to set the alarm triggering criteria.

DESCRIPTION

DS2408

The **DS2408 (3)** allows control of other devices, like LEDs and relays. It extends the **DS2406** to 8 channels and includes memory.

Alternative switches include the **DS2406**, **DS2407** and even **DS2450**

ADDRESSING

DATASHEET

<http://pdfserv.maxim-ic.com/en/ds/DS2408.pdf>

http://www.hobby-boards.com/catalog/howto_lcd_driver.php

http://www.maxim-ic.com/appnotes.cfm/appnote_number/3286

SEE ALSO

AVAILABILITY

<http://www.owfs.org>

AUTHOR

Paul Alfille (palfile@earthlink.net)

NAME**DS2413** - Dual Channel Addressable Switch**SYNOPSIS**

Dual Switch

3A [.]XXXXXXXXXXXX[XX]/[**latch.**[A|B|ALL|BYTE] | **PIO.**[A|B|ALL|BYTE] |
sensed.[A|B|ALL|BYTE] |]]

FAMILY CODE

3A

SPECIAL PROPERTIES**latch.A latch.B latch.ALL latch.BYTE***read-only, yes-no*

Poorly documented feature of the chip. Apparently set to 1 routinely, and if different from *PIO* then a physical switch was triggered? *ALL* references both channels simultaneously, comma separated
BYTE references both channels simultaneously as a single byte, with channel A in bit 0.

PIO.A PIO.B PIO.ALL PIO.BYTE*read-write, yes-no*

State of the open-drain output (*PIO*) pin. 0 = non-conducting (off), 1 = conducting (on).

Writing zero will turn off the switch, non-zero will turn on the switch. Reading the *PIO* state will return the switch setting. To determine the actual logic level at the switch, refer to the *sensed* property.

ALL references both channels simultaneously, comma separated.

BYTE references both channels simultaneously as a single byte, with channel A in bit 0.

sensed.A sensed.B sensed.ALL sensed.BYTE*read-only, yes-no*

Logic level at the *PIO* pin. 0 = ground. 1 = high (~2.4V - 5V). Really makes sense only if the *PIO* state is set to zero (off), else will read zero.

ALL references both channels simultaneously, comma separated.

BYTE references both channels simultaneously as a single byte, with channel A in bit 0.

STANDARD PROPERTIES**ALARMS**

Use the *set_alarm* property to set the alarm triggering criteria.

DESCRIPTION**DS2413**

The **DS2413 (3)** allows control of other devices, like LEDs and relays. It differs from the *DS2405* with a cleaner interface and two channels The *DS2413* also has two channels like the *DS2406* and *DS2407* but has no memory, and no alarm. There is also varying types of switch and sensing in the *DS2408*, *DS2409*, *LCD*, *DS276x*, *DS2450*.

Unique among the switches, the *DS2413* can switch higher voltages, up to 28V.

ADDRESSING**DATASHEET**

<http://pdfserv.maxim-ic.com/en/ds/DS2406.pdf>

<http://pdfserv.maxim-ic.com/en/ds/DS2407.pdf>

SEE ALSO**AVAILABILITY**

<http://www.owfs.org>

AUTHOR

Paul Alfille (palfile@earthlink.net)

NAME**DS2431** - EEPROM (1 kBit)**SYNOPSIS**

Erasable programmable read-only memory (EEPROM)

2D [.XXXXXXXXXXXXXX][/[**memory** | **pages/page.0-3**|**ALL**] |]]**FAMILY CODE**

2D DS2431

SPECIAL PROPERTIES**memory***read-write, binary*

128 bytes of EEPROM memory. Initially all bits are set to 1. Memory is non-volatile.

pages/page.0 ... pages/page.3 pages/page.ALL*read-write, yes-no*Memory is split into 4 pages of 32 bytes each. *ALL* is an aggregate of the pages. Each page is accessed sequentially.**STANDARD PROPERTIES****ALARMS**

None.

DESCRIPTION**DS2431**

The **DS2431 (3)** is used for storing memory which should be available even after a reset or power off. It is also switch to erase-only mode (EPROM) although this isn't implemented in OWFS. *OWFS* system handles this automatically.

ADDRESSING**DATASHEET**<http://pdfserv.maxim-ic.com/en/ds/DS2431.pdf>**SEE ALSO****AVAILABILITY**<http://www.owfs.org>**AUTHOR**

Christian Magnusson (mag@mag.cx)

NAME**DS2433** - EEPROM (4 kBit)**SYNOPSIS**

Erasable programmable read-only memory (EEPROM)

23 [.]XXXXXXXXXXXX[XX][/[**memory** | **pages/page.[0-15|ALL]** |]]**FAMILY CODE**

23 DS2433

SPECIAL PROPERTIES**memory***read-write, binary*

512 bytes of memory. Initially all bits are set to 1. Writing zero permanently alters the memory.

pages/page.0 ... pages/page.15 pages/page.ALL*read-write, yes-no*Memory is split into 16 pages of 32 bytes each. *ALL* is an aggregate of the pages. Each page is accessed sequentially.**STANDARD PROPERTIES****ALARMS**

None.

DESCRIPTION**DS2433**The **DS2433 (3)** is used for storing memory which should be available even after a reset or power off. It's main advantage is for audit trails (i.e. a digital purse). *OWFS* system handles this automatically.**ADDRESSING****DATASHEET**<http://pdfserv.maxim-ic.com/en/ds/DS2433.pdf>**SEE ALSO****AVAILABILITY**<http://www.owfs.org>**AUTHOR**

Christian Magnusson (mag@mag.cx)

NAME**DS2450** - Quad A/D Converter**SYNOPSIS**

Voltage * 4 and Memory.

20 [.]XXXXXXXXXXXX[XX]/[**PIO**.[A-D|ALL] | **volt**.[A-D|ALL] | **volt2**.[A-D|ALL]]]

20 [.]XXXXXXXXXXXX[XX]/[**alarm/high**.[A-D|ALL] | **alarm/low**.[A-D|ALL] | **memory** | **pages/page**.[0-3|ALL] | **PIO**.[A-D|ALL] | **power** | **set_alarm/high**.[A-D|ALL] | **set_alarm/low**.[A-D|ALL] | **set_alarm/unset** | **set_alarm/volthigh**.[A-D|ALL] | **set_alarm/volt2high**.[A-D|ALL] | **set_alarm/voltlow**.[A-D|ALL] | **set_alarm/volt2low**.[A-D|ALL] | **volt**.[A-D|ALL] | **volt2**.[A-D|ALL] |]]

FAMILY CODE

20

SPECIAL PROPERTIES**alarm/high.A ... alarm/high.D alarm.high.ALL****alarm/high.A ... alarm/high.D alarm.high.ALL***read-write, binary*

The alarm state of the voltage channel. The alarm state is set one of two ways:

voltage conversion

Whenever the *DS2450* measures a voltage on a channel, that voltage is compared to the high and low limits *set_alarm/volthigh* and/or *set_alarm/voltlow* and if the alarm is enabled *set_alarm/high* and/or *set_alarm/low* the corresponding flag is set in *alarm/high* and/or *alarm/low*

manual set

The flag can be set by a direct write to *alarm/high* or *alarm/low***memory***read-write, binary*

32 bytes of data. Much has special implications. See the datasheet.

pages/page.0 ... pages/page.3 pages/page.ALL*read-write, binary*

Memory is split into 4 pages of 8 bytes each. Mostly for reading and setting device properties. See the datasheet for details.

ALL is an aggregate of the pages. Each page is accessed sequentially.**PIO.A ... PIO.D PIO.ALL***read-write, yes-no*

Pins used for digital control. 1 turns the switch on (conducting). 0 turns the switch off (non-conducting).

Control is specifically enabled. Reading *volt* will turn off this control.*ALL* is an aggregate of the voltages. Readings are made separately.**power***read-write, yes-no*Is the *DS2450* externally powered? (As opposed to parasitically powered from the data line).

The analog A/D will be kept on continuously. And the bus will be released during a conversion allowing other devices to communicate.

Set true (1) only if Vcc powered (not parasitically powered). Unfortunately, the *DS2450* cannot sense it's powered state. This flag must be explicitly written, and thus is a potential source of error if incorrectly set.

It is always safe to leave *power* set to the default 0 (off) state.**set_alarm/high.A ... set_alarm/high.D set_alarm/high.ALL****set_alarm/low.A ... set_alarm/low.D set_alarm/low.ALL***read-write, yes-no*

Enabled status of the voltage threshold. 1 is on. 0 is off.

set_alarm/volthigh.A ... set_alarm/volthigh.D set_alarm/volthigh.ALL
set_alarm/volt2high.A ... set_alarm/volt2high.D set_alarm/volt2high.ALL
set_alarm/voltlow.A ... set_alarm/voltlow.D set_alarm/voltlow.ALL
set_alarm/volt2low.A ... set_alarm/volt2low.D set_alarm/volt2low.ALL

read-write, floating point

The upper or lower limit for the voltage measured before triggering an alarm.

Note that the alarm must be enabled *alarm/high* or *alarm/low* and an actual reading must be requested *volt* for the alarm state to actually be set. The alarm state can be sensed at *alarm/high* and *alarm/low*

set_alarm/unset

read-write, yes-no

Status of the power-on-reset (POR) flag.

The POR is set when the *DS2450* is first powered up, and will match the alarm state until explicitly cleared. (By writing 0 to it).

The purpose of the POR is to alert the user that the chip is not yet fully configured, especially alarm thresholds and enabling.

volt.A ... volt.D volt.ALL

read-only, floating point

Voltage read, 16 bit resolution. Range 0 - 5.10V.

Output (*PIO*) is specifically disabled.

ALL is an aggregate of the voltages. Readings are made separately.

volt2.A ... volt2.D volt2.ALL

read-only, floating point

Voltage read, 16 bit resolution. Range 0 - 2.55V.

Output (*PIO*) is specifically disabled.

ALL is an aggregate of the voltages. Readings are made separately.

STANDARD PROPERTIES

ALARMS

None.

DESCRIPTION

DS2450

The **DS2450 (3)** is a (supposedly) high resolution A/D converter with 4 channels. Actual resolution is reported to be 8 bits. The channels can also function as switches. Voltage sensing (with temperature and current, but sometimes restricted voltage ranges) can also be obtained with the **DS2436** , **DS2438** and **DS276x**

ADDRESSING

DATASHEET

<http://pdfserv.maxim-ic.com/en/ds/DS2450.pdf>

SEE ALSO

AVAILABILITY

<http://www.owfs.org>

AUTHOR

Paul Alfille (palfile@earthlink.net)

NAME

DS28EC20 - EEPROM (20 kBit)

SYNOPSIS

Erasable programmable read-only memory (EEPROM)

43 [.]XXXXXXXXXXXX[XX][/[**memory** | **pages/page.[0-79|ALL]** |]]

FAMILY CODE

23 DS28EC20

SPECIAL PROPERTIES**memory**

read-write, binary

512 bytes of memory. Initially all bits are set to 1. Writing zero permanently alters the memory.

pages/page.0 ... pages/page.79 pages/page.ALL

read-write, yes-no

Memory is split into 80 pages of 32 bytes each. *ALL* is an aggregate of the pages. Each page is accessed sequentially.

STANDARD PROPERTIES**ALARMS**

None.

DESCRIPTION**DS28EC20**

The **DS28EC20 (3)** is used for storing memory which should be available even after a reset or power off. It's main advantage is for audit trails (i.e. a digital purse). *OWFS* system handles this automatically.

ADDRESSING**DATASHEET**

<http://datasheets.maxim-ic.com/en/ds/DS28EC20.pdf>

SEE ALSO**AVAILABILITY**

<http://www.owfs.org>

AUTHOR

Christian Magnusson (mag@mag.cx)