

**Optional Package 1-Wire-Bus**

**Paket OPT\_OW**

**V. 2.0.1**

**für**

**Fli4L - 3.4.0**

von

Klaus der Tiger  
und Karl M.

c/o

Karl M. Weckler  
[news4kmw@web.de](mailto:news4kmw@web.de)  
Rockenberg, Oktober 2010

## Inhaltsverzeichnis

Paket OPT_OW .....	1
Inhaltsverzeichnis .....	2
1. Einleitung .....	4
1.1. OPT_OW .....	4
1.2. Hardware .....	4
1.2.1. Der 1-Wire Standard .....	4
1.2.2. 1-Wire-Bauteile .....	4
1.2.3. 1-Wire-Bus .....	4
1.3. OWFS .....	5
1.4. Fuse .....	5
1.5. libusb .....	5
2. Lizenz .....	5
3. Gewährleistungs- und Haftungsausschluss .....	5
4. Systemvoraussetzungen .....	6
5. Installation .....	6
6. Konfiguration .....	6
6.1. OPT_OW .....	7
6.2. OW_USER_SCRIPT .....	7
6.3. OW_OWFS .....	7
6.3. Sonstige Variable .....	8
7. Bedienung im Browser und auf der Konsole .....	10
7.1. Browser .....	10
7.1.1. Webserver .....	10
7.1.2. Darstellung .....	10
7.2. Konsole .....	10
8. Erweiterte Funktionen .....	11
8.1. Rechtevergabe .....	11
8.2. Bauteilebibliothek .....	11
8.3. OW_USER_SCRIPT .....	11
8.4. RRDTool .....	11
8.4.1. Schnittstelle .....	11
8.4.2. Linienfarben .....	11
8.4.3. Beispielkonfiguration .....	12

9. Feedback.....	12
Anhang .....	13
A I. Pin-, Adernbelegung für TP-Kabel und RJ-45 .....	13
A II. Family Code Referenz (aus Maxim AppNote 155) .....	14
A III. Links zum Thema 1-Wire .....	15
A IV. Bezugsquellen für 1-Wire Bauteile.....	15
A V. Schaltplanskizzen .....	16
Passiver serieller zu 1-Wire Adapter .....	16
Links.....	16
A VI. Man-Pages .....	17
OWFS .....	17
OWSHELL .....	18
OWFS.CONF .....	20
DS18S20.....	23
DS2401 .....	24
DS2406 DS2407 .....	25
DS2408 .....	27
DS2413 .....	30
DS2423 .....	31
DS2433 .....	33
DS2450 .....	34
DS28EC20 .....	36

# 1. Einleitung

## 1.1. OPT\_OW

Dieses Paket installiert das OWFS (siehe Kapitel 1.3.) und bietet so lesenden und schreibenden Zugriff auf einen an den Fli4L kontaktierten 1-Wire Bus. Hierzu wird ein 1-Wire Busmaster an eine serielle Schnittstelle<sup>1</sup> oder einen USB Port<sup>2</sup> des PC angeschlossen. Darüber hinaus unterstützt das Opt auch I<sup>2</sup>C Adapter und die Anbindung an OWServer. Näheres hierzu in den man-pages im Anhang. An die 1-Wire-seitige Buchse des entsprechenden Adapters wird dann der eigentliche 1-Wire Bus angelegt.

## 1.2. Hardware

### 1.2.1. Der 1-Wire Standard

Der 1-Wire ® bzw. One-Wire oder Eindraht-Bus von Maxim (Maxim/Dallas) beschreibt eine serielle Schnittstelle, die mit einer Datenader auskommt und 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 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. Man bekommt eigentlich alles, was man so für die Hausautomation braucht. Eine Übersicht über die wichtigsten Bauteile findet sich im Anhang unter A II.

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

### 1.2.3. 1-Wire-Bus

Der 1-Wire-Bus besteht im Prinzip aus zwei verdrehten Leitungen, wobei unter Beachtung entsprechender Topologien auch längere Strecken bis 150 m kein Problem sein sollten. Häufig wird für die Verkabelung normales Cat.5 Twisted Pair Ethernet-Kabel genommen.

Zur Belegung der einzelnen Adern gibt es unterschiedliche Ansätze. Maxim benutzt 6-polige Modular-Buchsen und -Stecker (RJ-11) und hat einen eigenen Standard kreiert, der jedoch nicht zum 8-poligen RJ-45 Steckzeug passt. Weitere Standards sind im Anhang beschrieben.

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.

---

<sup>1</sup> DS9097U COM Port Adapter.

<sup>2</sup> DS9490R USB Bridge, auch zusammen mit DS1402D-DR8 (Blue Dot™) für iButton. Alle DS9490 Adapter basieren auf dem DS2490 USB-1-Wire-Baustein.

### 1.3. OWFS

OWFS steht für „**One Wire File System**“. Dabei handelt es sich um eine von Paul H. Alfille entwickelte Software die unter der GPL lizenziert ist. Auf Grundlage einer 1-Wire-Protokoll verständigen Systembibliothek (OWLib) bildet OWFS den 1-Wire-Bus als Dateisystem ab. Darüber hinaus bietet das Programm noch weitere Implementierungen, wie owserver, owshell, owhttpd, owftpd, owtap und 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/>.

### 1.4. Fuse

Fuse steht für „**Filesystem in userspace**“. Fuse ermöglicht die Implementierung eines voll funktionsfähigen Dateisystems im Userspace.

Mit der Installation von OPT\_OW wird das mit Fli4L ausgelieferte Fuse automatisch als Kernelmodul beim Systemstart geladen. Alles weitere zu Fuse findet man auf: <http://fuse.sourceforge.net/> und <http://sourceforge.net/projects/fuse/>.

### 1.5. libusb

Bei libusb handelt es sich um eine freie, GPL-lizenzierte USB-Bibliothek. Diese wird benötigt, um über einen USB-Adapter auf den 1-Wire-Bus zuzugreifen.

Alles weitere zu libusb findet man auf:

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

## 2. Lizenz

Dieses Programm ist durch die GNU General Public License, Version 2, Juni 1991, lizenziert und kann unter den angegebenen Bedingungen frei verwendet, vervielfältigt und verändert werden. Der Text der GNU General Public License ist im Internet veröffentlicht unter: <http://www.gnu.org/licenses/gpl.txt>

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 in der Hoffnung, dass es 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 wird **keine Haftung** übernommen.

## 4. Systemvoraussetzungen

Auf Grund der Größe des OPT\_OW benötigt man eine Festplatte bzw. Flashkarte. Näheres dazu unter OPT\_HD. Eine Installation auf FD ist nicht möglich.

Für die Anzeige im Browser ist der im Fli4L-Paket „httpd“ enthaltene Mini-Webserver erforderlich. Weitere Hinweise hierzu unter Kapitel 7.1.

### Zur Beachtung:

Das Opt wurde unter Kernel 2.4 noch nicht getestet.

Die USB Ansteuerung über die W1-Kernelmodule klappt nach Auskunft von Paul Alfille, dem Maintainer von OWFS, noch nicht und wurde im Opt bisher nicht getestet.

Die Verwendung der beiden Programme owshell und owhttpd funktionierte im Testbetrieb auf einigen Hardwareumgebungen bisher nicht zufrieden stellend. Die Autoren bemühen sich in Zusammenarbeit mit Paul Alfille um eine Lösung des Problems. Deshalb wurden die beiden Programme zunächst aus dem Opt herausgenommen. Wer owshell oder owhttpd benötigt, der sollte Version 1.0. benutzen. Allerdings ist damit kein Zugriff über USB-Adapter möglich.

## 5. Installation

Nach dem Entpacken des tar.gz-Archivs in das Fli4L-Verzeichnis ist die Textdatei config/ow.txt zu bearbeiten. Für die Verwendung des Webinterface muss in config/httpd.txt die Variable `OPT_HTTPD='yes'` gesetzt werden (siehe Kapitel 6.3). Wird RRDTool für die Aufzeichnung von Messwerten benötigt, so ist die Konfiguration der Textdatei config/rrdtool.txt erforderlich (siehe Kapitel 8.4).

## 6. Konfiguration

Beispielkonfiguration ohne Kommentare, nähere Erläuterungen weiter unten:

```
OPT_OW='yes'                # install OPT_OW (yes/no)
OW_USER_SCRIPT=''          # e.g. 'usr/local/bin/ow-user-script.sh'

OW_OWFS='yes'              # start owfs (yes/no)
OW_OWFS_DEV='usb'          # usb*, ttyS*, ip:port, etc.
OW_OWFS_GROUP_N='4'        # number of groups
OW_OWFS_GROUP_1_NAME='1-Wire an USB' # name of first group
OW_OWFS_GROUP_1_PORT_N='2' # number of ports of device
OW_OWFS_GROUP_1_PORT_1_ID='81.70D42A000000/ID' # ID of device
OW_OWFS_GROUP_1_PORT_1_ALIAS='ID' # alias of ID
OW_OWFS_GROUP_1_PORT_2_ID='81.70D42A000000/Admin/*' # admin-access
OW_OWFS_GROUP_1_PORT_2_ALIAS='Admin/' # alias of admin

OW_OWFS_GROUP_2_NAME='Heizung'
OW_OWFS_GROUP_2_PORT_N='7'
OW_OWFS_GROUP_2_PORT_1_ID='3A.F6E401000000/PA'
OW_OWFS_GROUP_2_PORT_1_ALIAS='1. Umwälzpumpe'
OW_OWFS_GROUP_2_PORT_2_ID='3A.F6E401000000/PB'
OW_OWFS_GROUP_2_PORT_2_ALIAS='2. Ladepumpe'
OW_OWFS_GROUP_2_PORT_3_ID='10.651BA9010800/temp'
OW_OWFS_GROUP_2_PORT_3_ALIAS='4. Rücklauftemperatur'
OW_OWFS_GROUP_2_PORT_4_ID='10.DEF0A8010800/temp'
OW_OWFS_GROUP_2_PORT_4_ALIAS='3. Vorlauftemperatur'
```

```

OW_OWFS_GROUP_2_PORT_5_ID='3A.F6E401000000/Admin/*'
OW_OWFS_GROUP_2_PORT_5_ALIAS='Admin/Switch-'
OW_OWFS_GROUP_2_PORT_6_ID='10.DEF0A8010800/Admin/*'
OW_OWFS_GROUP_2_PORT_6_ALIAS='Admin/VLT-'
OW_OWFS_GROUP_2_PORT_7_ID='10.651BA9010800/Admin/*'
OW_OWFS_GROUP_2_PORT_7_ALIAS='Admin/RLT-'

OW_OWFS_GROUP_3_NAME='Solaranlage'
OW_OWFS_GROUP_3_PORT_N='3'
OW_OWFS_GROUP_3_PORT_1_ID='1C.7F6CF7040000/P0'
OW_OWFS_GROUP_3_PORT_1_ALIAS='1. Ladepumpe'
OW_OWFS_GROUP_3_PORT_2_ID='1C.7F6CF7040000/P1'
OW_OWFS_GROUP_3_PORT_2_ALIAS='2. Ventil'
OW_OWFS_GROUP_3_PORT_3_ID='1C.7F6CF7040000/Admin/*'
OW_OWFS_GROUP_3_PORT_3_ALIAS='Admin/Switch-'

```

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

### 6.1. OPT\_OW

Die Standardeinstellung von *OPT\_OW*='no', das Paket wird nicht installiert.  
Mit *OPT\_OW*= 'yes' wird das Paket aktiviert.

### 6.2. OW\_USER\_SCRIPT

Hinter dieser Variablen verbirgt sich Pfad und Dateiname einer optionalen Hintergrundsteuerung, mit der zum Beispiel die Heizungsanlage geregelt werden kann.  
Nähere Erläuterungen finden sich in Kapitel 8.3.

### 6.3. OW\_OWFS

OWFS bietet einfachen Zugriff auf den 1-Wire Bus über die Fli4L-Weboberfläche. Mit der Auswahl *OW\_OWFS*='yes' wird mittels Fuse ein Dateisystem unter dem Standardpfad '/var/run/ow' erzeugt. Dort wird der 1-Wire- Bus abgebildet. Die im Dateisystem angelegten Verzeichnisse sind nach den Identnummern (siehe Anhang AII) der Chips geordnet. Über den family code der Bauteile ist eine entsprechende Systematik leicht herzustellen.

Mit der Variablen *OW\_OWFS\_DEV* legt man die PC-Schnittstelle fest, an der der 1-Wire-Adapter angeschlossen wird.

<i>PC-Schnittstelle</i>	<i>Variablenbelegung</i>	<i>Beispiel</i>
seriell	ttyS*	ttyS0 = COM1, ttyS1 = COM2
USB	ttyUSB*	ttyUSB1 = erster USB-Adapter
	usb	usb = erster USB-Adapter
	usb[2-9]	usb3 = dritter USB-Adapter

I <sup>2</sup> C	i2c-[0-9]	i2c-0 = erster I <sup>2</sup> C-Port
Simulation	fake	Für die Verwendung der Modi 'FAKE' und 'TESTER' müssen die Variablen OW_OWFS_FAKE oder OW_OWFS_TESTER auf gültige family codes gesetzt werden, siehe Kapitel 6.3.
	tester	

Die Variable `OW_OWFS_GROUP_N` bestimmt die Anzahl der im Browser angezeigten Gruppen in die zusammen gehörige Ein- und Ausgänge für zum Beispiel die Steuerung einer Solaranlage zusammen gefasst werden und deren Namen mit `OW_OWFS_GROUP_NAME` festgelegt wird.

Mit den beiden untergeordneten Variablen `OW_OWFS_GROUP_%_PORT_%_ID` und `OW_OWFS_GROUP_%_PORT_%_ALIAS` weist man den jeweiligen Aus-, bzw. Eingängen des 1-Wire Bauteils einen Decknamen zu.

Möchte man bestimmte Anzeigen im Webinterface unterdrücken, weil zum Beispiel der Port eines Bauteils nicht belegt wurde oder der Admin-Zweig nach Abschluss der Konfiguration nicht mehr benötigt wird, dann kann man dem Namen ein Rufzeichen (!) voranstellen.

Konfigurationsbeispiel:

```
OW_OWFS_GROUP_%_PORT_%_ID='29.57D305000000/P6'
OW_OWFS_GROUP_%_PORT_%_ALIAS='EA-Modul/!P6'      # Signal unterdrückt
OW_OWFS_GROUP_%_PORT_%_ID='29.57D305000000/Admin/*'
OW_OWFS_GROUP_%_PORT_%_ALIAS='EA-Modul/Admin/!'  # Admin-Pfad komplett
                                                    # abgeschaltet
```

Eine weitergehende Beschreibung zur Konfiguration von OWFS gibt es im Anhang „A VI.“ und hier: <http://owfs.org/index.php?page=owfs>

### 6.3. Sonstige Variable

Die folgenden Variablen können bei Bedarf über die Datei `config/ow.txt` angepasst werden:

Variable	Bedeutung
OW_LOG_DESTINATION	Ziel für Status- und Fehlerausgaben. 0 = mixed (1 und 2) 1 = syslog 2 = stderr 3 = off Standardwert ist '1'.
OW_LOG_LEVEL	Der Log-Level (1-9) bestimmt die Anzahl der Status- und Fehlerausgaben, wobei: 1 = leise und 9 = geschwatzig Standardwert ist '1'.



OW_TEMP_SCALE	Die zur Verfügung stehenden Temperaturskalen. C = "Celsius" F = "Fahrenheit" K = "Kelvin" R = "Rankine" Standardwert ist 'C'.
OW_REFRESH_INTERVAL	Refresh der FLI4L-HTTP-Anwendung in Sekunden. „0“ = kein Refresh. Standardwert ist '10'.
OW_OWFS_FAKE	Aktiviert die zufällige Simulation von 1-Wire-Bauteilen. Es können mehrere Bauteile mit dem family code und durch Leerzeichen getrennt angegeben werden. Die simulierten Zustände sind rein zufällig. Die Option kann nicht gleichzeitig mit dem 'TESTER'-Modus aktiviert werden.
OW_OWFS_TESTER	Aktiviert die systematische Simulation von 1-Wire-Bauteilen. Es können mehrere Bauteile mit dem family code und durch Leerzeichen getrennt angegeben werden. Die simulierten Zustände folgen natürlichen Werten. Die Option kann nicht gleichzeitig mit dem 'FAKE'-Modus aktiviert werden.
OW_OWFS_RUN	Gibt an, ob owfs beim Boot des Routers automatisch gestartet werden soll. Standardwert ist 'yes', wohingegen nach dem Setzen von 'no' die Anwendung manuell gestartet werden muss.
OW_OWFS_READONLY	Legt mit 'yes' fest, dass Bauteilzustände über owfs nur gelesen, aber nicht geschrieben werden können. Standardwert ist 'no'.
OW_OWFS_PATH	Gibt das Wurzelverzeichnis der Fuse-Verzeichnisstruktur an. Standardwert ist '/var/run/ow'. Das gewählte Verzeichnis sollte aus Gründen der Systemleistung unbedingt auf der RAMdisk liegen!
OW_CACHE_SIZE	Dient zur Anpassung der maximalen Größe des cache in [bytes] auf Systemen mit sehr kleiner RAMdisk. Der Standardwert '0' hebt jegliche Limitierung auf.
OW_USER_SCRIPT_INTERVAL	Gibt in Sekunden an, wie lange zwischen zwei Durchläufen des user-script gewartet wird. Der Wert '0' sollte nur verwendet werden, wenn innerhalb des user-script ein 'sleep' ausgeführt wird.
OW_DEVICE_LIB	Legt den absoluten Pfad und Dateinamen der Bauteilbibliothek auf dem Router fest. Durch die Verwendung eines anderen Werts als des Standardwerts '/srv/www/include/ow-device.lib' kann sichergestellt werden, dass bei einer Aktualisierung des Opt die Bibliothek nicht überschrieben wird und so persönliche Änderungen an der Bauteilbibliothek erhalten bleiben.
OW_INVERT_PORT_LEDS	Invertiert den Status der Port-Leds von i/o Ports (latch*, sensed*, PIO*). Standardwert ist 'no'.

## 7. Bedienung im Browser und auf der Konsole

### 7.1. Browser

#### 7.1.1. Webserver

Der in Fli4I optional zu installierende Webserver (opt\_httpd) bietet die Möglichkeit, eigene Shell/CGI Skript Anwendungen über jeden Browser im Netz zu bedienen. Davon wurde im vorliegenden Falle Gebrauch gemacht. Um den Webserver zu nutzen muss *config/httpd.txt* entsprechend konfiguriert werden.

Im OPT\_OW wird eine Browser Applikation mitgeliefert. Sie wird nur installiert, wenn in */config/ow.txt* *OW\_OWFS="yes"* gesetzt wird, da sie mit „owfs“ zusammen arbeitet. Das zugehörige Skript befindet sich gemäß Vorgaben auf dem Fli4I unter */srv/www/admin/ow.cgi* und im Fli4I-Installationsverzeichnis unter *fli4I-version\opt\files\srv\www\admin\ow.cgi*. Der zugehörige Menüpunkt erscheint unter „Opt / 1-Wire-Bus“.

#### 7.1.2. Darstellung

Im Reiter „Status“ werden die am 1-Wire-Bus angeschlossenen Bauteile gemäß der in *config/ow.txt* vorgenommenen Konfiguration gruppenweise in einer Baumstruktur angezeigt. Mittels ‚Klick‘ wird die jeweilige Gruppe geöffnet. Die konfigurierten Werte werden angezeigt. In der Admin-Struktur werden alle in der Bauteilbibliothek (siehe 8.4) für das jeweilige Bauteil definierten Parameter angezeigt. Hinsichtlich der Bedeutung dieser Parameter wird auf die Datenblätter von Maxim und die beiliegenden Man-Pages verwiesen.

Im Reiter „Admin“, der nur im Admin-Modus angezeigt wird, können die gewählten Applikationen ein- und ausgeschaltet werden.

Die angezeigten LEDs signalisieren mit ihren Farben folgende Zustände:

LED grün = inaktiv (Ruhe)

LED rot = aktiv (Betrieb)

LED gelb = inaktiv (Warnung)

Die Schaltknöpfe dienen zum Umschalten der zugeordneten Ports. Das Symbol zeigt zusätzlich den aktuellen Schaltzustand. Hinsichtlich der Berechtigungen siehe 8.1.

### 7.2. Konsole

Die Abfrage und Steuerung der Sensoren und Aktoren ist auch auf der Konsole des Fli4L oder über einen Remotezugriff (WinSCP, Putty o.ä.) möglich.

Mit z.B.:

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

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

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

## 8. Erweiterte Funktionen

### 8.1. Rechtevergabe

In der Fli4L-Webanwendung ist die Vergabe von Benutzerrechten implementiert, siehe hierzu die Erläuterungen in doc/deutsch/pdf/httpd.pdf. Dieser Rechtevergabe bedient sich auch Opt-OW. Zur Nutzung der OW-Rechte können in der Datei config/httpd.txt für den Bereich „ow“ die folgenden Stufen vorgegeben werden:

admin = alle Rechte

exec = Befehle ausführen, Ein- und Ausgänge schalten, Ansicht der Daten

view = Ansicht der Daten

Die Admin-Tabelle, über die owfs und das user-script ein- und ausgeschaltet werden können, wird in den Stufen „exec“ und „view“ nicht angezeigt. Weiterhin werden in diesen Berechtigungsstufen alle Anzeigen, die ein „Admin“ enthalten, ausgeblendet.

### 8.2. Bauteilebibliothek

Auf Grund der Vielzahl der von MAXIM angebotenen 1-Wire Bauteile wurde eine eigene Bauteilebibliothek angelegt. Das entsprechende Library-script liegt auf dem Fli4I unter /srv/www/include/ow-device.lib und im Fli4I-Installationsverzeichnis unter fli4l-version\opt\files\srv\www\include\ow-device.lib. Die Bibliothek enthält bereits einige wichtige Bauteile. Eigene Devices können entsprechend der verwendeten Systematik nachgetragen und über die Fli4I-Newsgroup 'spline.fli4l.opt' den übrigen Fli4L-Nutzern zur Verfügung gestellt werden. Es werden nur in der Bibliothek angelegte Bauteile im Browser angezeigt. Das Libraryscript kann wunschgemäß entweder mittels Programmen wie „WinSCP“ auf dem Fli4I selbst zu Testzwecken oder als dauerhafte Änderung im Fli4I-Installationsverzeichnis editiert werden.

### 8.3. OW\_USER\_SCRIPT

Das Skript findet sich auf dem Fli4L unter /usr/local/bin/ow-user-script.sh und im Fli4I-Installationsverzeichnis unter fli4l-version\opt\files\usr\local\bin\ow-user-script.sh. Es kann nach eigenen Wünschen und Bedürfnissen an die zu überwachenden und zu steuernden Anwendungen angepasst werden. Der Vorteil des Skript ist darin zu sehen, dass auch umfangreiche und komplexe Steuerungen auf bestehender Hardware möglich sind.

### 8.4. RRDTool

#### 8.4.1. Schnittstelle

Die über den 1-Wire-Bus erfassten Daten können mittels Fli4L-Opt „RRDTool“ aufgezeichnet und graphisch aufbereitet werden. Das vorliegende Opt bringt die erforderlichen Schnittstellen bereits mit. Hierzu muss owfs (siehe /config/ow.txt) installiert sein. Bei der Installation von RRDTool sind in der /config/rrdtool.txt die gewünschten Einträge vorzunehmen. Hierbei ist jedoch zu beachten, dass aus technischen Gründen die Formate der Ein- und Ausgangsparameter in den Dateien config/ow.txt und config/rrdtool.txt von einander abweichen.

#### 8.4.2. Linienfarben

Die farbliche Gestaltung der Linien in den Graphen des RRDtool kann man mit der Variablen `OW_RRDTOOL_COLORS` entsprechend anpassen. Es können mehrere

Werte durch Leerzeichen getrennt angegeben werden, wobei jeder Wert aus einem Satz von drei hexadezimalen Zahlenpaaren besteht, die den Farbwert der drei Grundfarben Rot, Grün und Blau angeben. Zum Beispiel steht '00FF00' für Grün und 'FFE000' für einen Gelbton. Die Position des Werts innerhalb der Variablen steht für die Linie im Graphen, daher steht der erste Wert für die erste Linie und so weiter. Die Auswertung erfolgt zyklisch, wodurch nach Verwendung des letzten Werts die nächste Linie wieder in der Farbe des ersten Werts dargestellt wird. Entsprechend werden alle Linien mit derselben Farbe dargestellt, wenn nur eine Farbe angegeben wird.

### 8.4.3. Beispielkonfiguration

```
RRDTOOL_%_SOURCE='owfs'           # name of data source
RRDTOOL_%_LABEL='OWFS'            # optional label setting
                                   # required when using the same
                                   # source more then once
RRDTOOL_%_OPTIONS_N='4'          # number of options
RRDTOOL_%_OPTIONS_1='10.651BA9010800-temperature'
RRDTOOL_%_OPTIONS_1_LABEL='Hzg-V-Temp'
RRDTOOL_%_OPTIONS_2='10.DEF0A8010800-temperature'
RRDTOOL_%_OPTIONS_2_LABEL='Hzg-R-Temp'
RRDTOOL_%_OPTIONS_3='3A.F6E401000000-PIO.A'
RRDTOOL_%_OPTIONS_3_LABEL='L-Pumpe'
RRDTOOL_%_OPTIONS_4='3A.F6E401000000-PIO.B'
RRDTOOL_%_OPTIONS_4_LABEL='U-Pumpe'
```

Will man mehrere Linien in einer Darstellung aufzeichnen, so ist dies durch die Verwendung eines Pluszeichens (+) zwischen den einzelnen Portangaben wie folgt möglich

```
RRDTOOL_%_OPTIONS_1='10.651BA9010800-temperature+10.DEF0A8010800-
temperature+10.652BA9010800-temperature'
RRDTOOL_3_OPTIONS_1_LABEL='Temperaturen'
```

Im vorliegenden Beispiel werden drei Temperaturgraphen in einem Bild dargestellt.

## 9. Feedback

Wir freuen uns über jede, auch kurze Rückmeldung, selbst wenn das Paket ohne jegliche Probleme laufen sollte ;-).

Viel Spaß mit 1-Wire wünschen

Klaus der Tiger      [der.tiger.opt-ow@arcor.de](mailto:der.tiger.opt-ow@arcor.de)  
und  
Karl M. Weckler      [news4kmw@web.de](mailto:news4kmw@web.de)

## Anhang

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

Eine Übersicht über die Belegungsvarianten findet sich hier:

[http://www.1wire.org/media/A\\_Guide\\_to\\_the\\_1WRJ45\\_Standard\\_Draft.zip](http://www.1wire.org/media/A_Guide_to_the_1WRJ45_Standard_Draft.zip)

Wer sich nicht durch alle Varianten quälen möchte, verwendet folgendes universelle und zukunftsfähige Schema für die RJ45 Pin-Belegung.

Farbangaben nach:		
EIA/TIA 568A		EIA/TIA 568B
1 Grün/Weiß	Hauptversorgung GND	Orange/Weiß
2 Grün	Hauptversorgung +5V/50mA für 1-Wire Bauteile	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.

## A II. Family Code Referenz (aus Maxim AppNote 155)

Family Code	Bauteil / Chip	Beschreibung (Speichergröße in bits, wnaa)
(hex) 01	(DS1990A), (DS1990R), DS2401, DS2411	1-Wire net address (registration number) only
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 Hygro- chron™ (humidity) loggers
42	DS28EA00	Programmable resolution digital thermometer with se- quenced 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.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](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/>

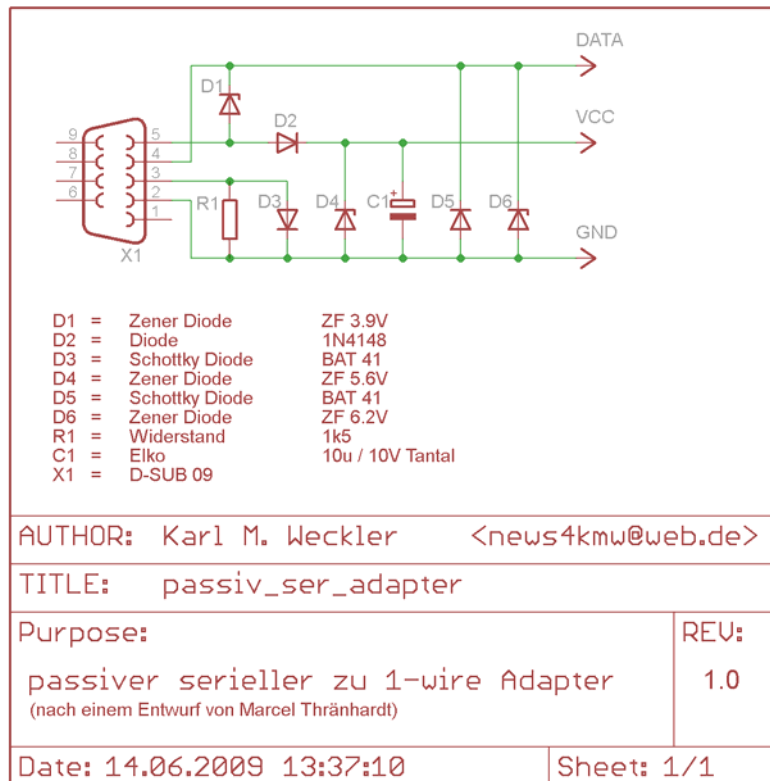
<http://www.aagelectronica.com/aag/index.html>

[http://www.hobby-boards.com/catalog/main\\_page.php](http://www.hobby-boards.com/catalog/main_page.php)

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

## A V. Schaltplanskizzen

### Passiver serieller zu 1-Wire Adapter



### Links

[http://www.hobby-boards.com/catalog/main\\_page.php](http://www.hobby-boards.com/catalog/main_page.php)  
<http://www.rockenberg.net/ow.html>



## A VI. Man-Pages

Die komplette Manual Pages Liste findet sich unter:  
<http://owfs.org/index.php?page=software>

### OWFS

OWFS(1) One-Wire File System OWFS(1)

#### 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](mailto:paul.alfille@gmail.com))

## OWSHELL

### 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 characrters 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* *fic* *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)

## OWFS.CONF

### 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 family 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 ([palfill@earthlink.net](mailto:palfill@earthlink.net))

## DS18S20

### NAME

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

### SYNOPSIS

Thermometer.

**10** [.]XXXXXXXXXXXX[XX][/[ **die** | **power** | **temperature** | **temphigh** | **templow** | **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)

## DS2401

### 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)



## DS2406 DS2407

### NAME

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

### SYNOPSIS

Dual Switch, Write-once Memory

**12** [.XXXXXXXXXXXXXX[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

*I* 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 ([palfille@earthlink.net](mailto:palfille@earthlink.net))

## DS2408

### NAME

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

### SYNOPSIS

8 port switch

**29** [.XXXXXXXXXXXXXX[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.hobby-boards.com/catalog/howto_lcd_driver.php)

[http://www.maxim-ic.com/appnotes.cfm/appnote\\_number/3286](http://www.maxim-ic.com/appnotes.cfm/appnote_number/3286)

**SEE ALSO**

**AVAILABILITY**

<http://www.owfs.org>

**AUTHOR**

Paul Alfille (palfile@earthlink.net)

## DS2413

### 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)

## DS2423

### NAME

**DS2423** - 4kbit 1-Wire RAM with Counter

### Synopsis

RAM and counters.

1D [.]XXXXXXXXXXXX[XX][/[ counters.[A|B|ALL] | memory | pages/page.[0-15|ALL] | pages/count.[0-15|ALL] | address | crc8 | id | locator | r\_address | r\_id | r\_locator | type ]]

Family Code: 1D

### Special Properties

counters.A counters.B counters.ALL

read-only, unsigned integer

Debounced external counter. Associated with RAM page.14 and page.15 Note: counter increments only. It is reset when the chip loses power.

ALL returns the two values, separated by a comma. They are read sequentially.

### memory

read-write, binary

512 bytes of memory.

pages/page.0 ... pages/page.15 pages/page.ALL

read-write, binary

Memory is split into 16 pages of 32 bytes each. Memory is RAM, contents are lost when power is lost. ALL is an aggregate of the pages. Each page is accessed sequentially.

pages/count.0 ... pages/count.15 pages/count.ALL

read-only, unsigned integer

Write access to each page of memory. Actually only page.12 and page.13 have write counters.

page14 and page.15 's counters are associated with the external counters.A and counters.B triggers.

The value 0xFFFFFFFF is returned for pages/count.0 through pages/count.11

ALL is an aggregate of the counters, comma separated. Each page is accessed sequentially.

### Standard Properties

#### address / r\_address - read-only, ascii

The entire 64-bit unique ID. Given as upper case hexadecimal digits (0-9A-F).

address starts with the family code

r\_address is the address in reverse order, which is often used in other applications and labeling.

#### crc8

read-only, ascii

The 8-bit error correction portion. Uses cyclic redundancy check. Computed from the preceeding 56 bits of the unique ID number. Given as upper case hexadecimal digits (0-9A-F).

#### family

read-only, ascii

The 8-bit family code. Unique to each type of device. Given as upper case hexadecimal digits (0-9A-F).

#### id / r\_id - read-only, ascii

The 48-bit middle portion of the unique ID number. Does not include the family code or CRC. Given as upper case hexadecimal digits (0-9A-F).

r\_id is the id in reverse order, which is often used in other applications and labeling.

#### locator / r\_locator - read-only, ascii

Uses an extension of the 1-wire design from iButtonLink company that associated 1-wire physical connections with a unique 1-wire code. If the connection is behind a Link Locator the locator will show a unique 8-byte number (16 character hexadecimal) starting with family code FE.

If no Link Locator is between the device and the master, the locator field will be all FF.

r\_locator is the locator in reverse order.

#### read-only, yes-no

Is the device currently present on the 1-wire bus?

type - read-only, ascii

Part name assigned by Dallas Semi. E.g. DS2401 Alternative packaging (iButton vs chip) will not be distinguished.

Alarms - None.

### Datasheet

<http://pdfserv.maxim-ic.com/en/ds/DS2423.pdf>

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

Erasable programmable read-only memory (EEPROM)

**2D** [.XXXXXXXXXXXXXX[XX]][/[ **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)



## DS2433

### NAME

**DS2433** - EEPROM (4 kBit)

### SYNOPSIS

Erasable programmable read-only memory (EEPROM)

**23** [.XXXXXXXXXXXXXX[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)

## DS2450

### 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.[AD|
ALL] | set_alarm/unset | set_alarm/volthigh.[A-D|ALL] | set_alarm/volt2high.[A-D|ALL] |
set_alarm/voltflow.[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/voltflow.A ... set\_alarm/voltflow.D set\_alarm/voltflow.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 ([palfille@earthlink.net](mailto:palfille@earthlink.net))

## DS28EC20

### 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)