

Netport Webservices

14. Juli 2011

Zusammenfassung

Alle Netportmodule bieten die Möglichkeit eine Userspezifische Webseite zu konfigurieren. Die Webseite kann dazu dienen statische Inhalte zu präsentieren oder auch in Interaktion mit der jeweiligen Hardware des Netportmoduls zu treten. Zur interaktiven Steuerung des Netportmoduls wird HTML in Kombination mit JavaScript verwendet.

Dieses Dokument beschreibt die Schnittstellen, die zur Steuerung der Hardware erforderlich sind.

Inhaltsverzeichnis

1	Umfang der Websteuerung	2
1.1	Die Werksseitige Webseite	2
1.2	Definition einer Benutzerspezifischen Webseite	2
1.3	Grundlegendes zur Kommunikation	3
1.3.1	getRequests	4
1.3.2	setRequests	4
1.4	Konfigurationsvariablen	4
1.4.1	Abfragen des Wertes einer Konfigurationsvariablen	5
1.4.2	Flüchtiges ändern einer Konfigurationsvariablen	6
1.4.3	Persistentes ändern einer Konfigurationsvariable	6
1.5	Flüchtige Variablen	8
1.6	I/O Requests	8
1.6.1	I/O getRequest	8
1.6.2	I/O setRequest	8
A	Java Script Bibliothek	10
A.1	serverRequest(url, callBack)	10
A.2	getVarValue(xml, name) : string	10
B	Vordefinierte Konfigurationsvariablen	11
B.1	PIO84-OCO	11
B.2	PIO48-OCA	14

Tabellenverzeichnis

1	Festlegungen bzgl. Datenrichtungen.	3
2	I/O Requests	8

1 Umfang der Websteuerung

Die Websteuerung umfasst folgendes:

1. Zugriff auf die Werksseitige Webseite
2. Definition einer Benutzerspezifischen Webseite
3. Zugriff auf den Netport aus der Webseite heraus
 - (a) Zugriff auf Konfigurationsvariablen aus der Webseite
 - (b) Zugriff auf Ein- und Ausgänge

1.1 Die Werksseitige Webseite

Hierbei handelt es sich um alle Webseiten die Sie unter “http://IPAddr/**factory/index.html**” erreichen.

Sie können diese Seite nicht löschen und es ist sichergestellt, dass Sie diese Seiten immer erreichen.

Im Auslieferungszustand entspricht factory/index.html der Startseite.

Hinweis

1.2 Definition einer Benutzerspezifischen Webseite

Sie können eine eigene Webseite definieren und im Flash ablegen. Diese eigene Webseite wird dann vom Netport eigenen Webserver gehostet und kann zur Konfiguration der Box, zur Anzeige von Werten oder auch zur Bedienung eingesetzt werden. Die Webseite kann, wie bei der Webentwicklung üblich Komponenten wie:

- Mehrere HTML Dateien (*.html)
- Stylesheets (*.css)
- Bilder (*.png, *...)

enthalten. Einzige Einschränkung ist der begrenzte Speicherplatz von 256kByte. Sobald die eigene Webseite eine Datei mit dem Namen “index.html” enthält wird sie als Startseite automatisch angezeigt, wenn Sie im Browser einfach nur die IP Adresse des Netports eingeben.

Im Auslieferungszustand sind bereits Modulspezifische Demoseiten installiert. Sie können den HTML Quellcode als Beispiel betrachten.

Hinweis

1.3 Grundlegendes zur Kommunikation

Es wird das “Asynchronous JavaScript and XML” (AJAX) Modell verwendet, um parallel zur statischen Webdarstellung im Browser eine- oder mehrere Kommunikationskanäle für die Steuerung zu verwenden. Es gilt folgende Festlegung für die Datenrichtungen:

Begriff	Funktion	Transfer
getRequest	Abfrage eines Zustandes oder Wertes	Netport -> Browser
setRequest	Ändern eines Zustandes	Browser -> Netport

Tabelle 1: Festlegungen bzgl. Datenrichtungen.

1.3.1 getRequests

Um Daten vom Netport abzufragen wird eine Netzwerkresource angefordert, die Daten im XML Format zurückgibt. Die XML Daten können mit Hilfe einfacher JavaScript Kommandos ausgewertet werden, um an die Informationen zu kommen.

1.3.2 setRequests

Daten, die vom Browser zum Netport übertragen werden sollen werden ebenfalls durch den Zugriff auf eine Netzwerkresource gesteuert. Die Daten, sowie die Inhalte werden jedoch im Pfad der Netzwerkresource hinterlegt.

Im folgenden Fall sind sowohl “**name**” als auch “**value**” Bestandteil der URL:

Beispiel:

```
serverRequest ("/dyn/setConfig?name=appCfgSEth0IPAddress&value="+escape("192.168.5.12"), function(xml) { alert("IP Finished"); } );
```

Quelle: ./snippets/setRequest1.js

1.4 Konfigurationsvariablen

Konfigurationsvariablen dienen zur Konfiguration des Netports. Sie werden verwendet um:

- Die IP Adresse zu speichern.
- Namen für Ein- oder Ausgänge zu konfigurieren.
- Voreinstellungen zu machen.
- Einiges mehr.

All Konfigurationsvariablen werden im Flash des Netports persistent gespeichert. Eine Liste gültiger Konfigurationsvariablen kann dem Anhang entnommen werden. Sie unterscheidet sich je nach Netport Modul.

Viele der Konfigurationsvariablen sind bereits über die “Settings” Seite der Werkswebseite einstellbar. Sie können aber auch mit Hilfe der Custom-Webseite angezeigt- oder verändert werden. Es ist sogar ohne weiteres möglich eigene Konfigurationsvariablen zu definieren und zu verwenden.

1.4.1 Abfragen des Wertes einer Konfigurationsvariablen

Algorithm 1 Abfragen einer Konfigurationsvariablen.

```
function showIPAddress ()
{
  serverRequest ("/dyn/getConfig?name=appCfgSEth0IPAddress", function (xml)
  {
    var sText = getVarValue(xml, "value");
    var label=document.getElementById("lblIP");
    var oStateNode = document.createTextNode(sText);
    label.removeChild(label.lastChild);
    label.appendChild(oStateNode);
  } );
}
```

Quelle: ./snippets/getIPAddr.js

Der dazugehörige HTML Code kann z.B. so aussehen:

```
This is the state: <span id='lblTest'>State</span><br>
```

Quelle: ./snippets/lblIP.html

Das erfragen des Inhaltes einer Konfigurationsvariable die nicht existiert resultiert in einem Fehlercode. (Siehe Fehlercodes von serverRequest().)

Hinweis

1.4.2 Flüchtliges ändern einer Konfigurationsvariablen

Um den Wert einer Variablen zu setzen wird zusätzlich der zukünftige Wert in der URL untergebracht. Beim flüchtigen Setzen wird der Wert der Variablen zwar geändert aber noch nicht im Flash gespeichert. Ein Beispiel, welches die IP Adresse ändert kann z.B. so aussehen:

Algorithm 2 Flüchtige Änderung einer Konfigurationsvariablen.

```
serverRequest ("/dyn/setConfig?name=appCfgSEth0IPAddress&value="+escape("192.168.1.12"),  
              function(xml) { alert("Success"); } );
```

Quelle: ./snippets/setIPAddr1.js

Wird der Wert einer Konfigurationsvariable gesetzt die noch nicht existiert, wird sie neu erzeugt.

Hinweis

Zu beachten ist, dass der Aufruf der Werksseitigen “Settings” Webseite alle flüchtigen Daten löscht. Weiterhin führt ein Klick auf “Set Defaults” in der Settings Webseite dazu, dass alle vom User erzeugten Variablen gelöscht werden.

Sollte der Benutzer auf der Einstellungsseite auf “Save Settings” drücken werden alle Variablen abgespeichert. (Auch solche die per JavaScript “flüchtig” angelegt wurden.)

1.4.3 Persistentes ändern einer Konfigurationsvariable

Ändert den Inhalt einer Konfigurationsvariable wie im vorherigen Abschnitt mit dem Unterschied, dass die Änderung persistent im Flash abgelegt wird. Bei neuladen der Konfigurationsdaten werden alle Variablen aus dem Flash gelesen. (Auch solche die vom Benutzer hinzugefügt wurden.)

Algorithm 3 Persistente Änderung einer Konfigurationsvariablen.

```
serverRequest ("/dyn/setConfig?name=appCfgSEth0IPAddress&value="+escape("192.168.1.12")+"&store=yes",  
              function(xml) { alert("Success"); } );
```

Quelle: ./snippets/setIPAddr2.js

Die persistente Änderung einer Konfigurationsvariablen hat einen Flash-Schreibvorgang zur Folge.

Hinweis

1.5 Flüchtige Variablen

Flüchtige Variablen können zum Informationsaustausch zwischen der JavaScript- und der Netport-Script Welt dienen. Im Unterschied zu Konfigurationsvariablen werden flüchtige Variablen niemals abgespeichert. Der Zugriff auf flüchtige Variablen geschieht ähnlich wie der Zugriff auf Konfigurationsvariablen.

1.6 I/O Requests

Ein I/O Request dient zur Steuerung der Hardware oder zur Abfrage von Eingangszuständen. Folgende I/O Requests sind definiert:

Request	Typ	Bemerkung
GPIOxValue	Skalarer Wert (Ganzzahl) { <i>int</i> ₃₂ }	z.B. bei analogen Ein- bzw. Ausgängen
GPIOxState	boolescher Wert { "true", "false" }	bei digitalen Ein- bzw. Ausgängen

Tabelle 2: I/O Requests

1.6.1 I/O getRequest

Dient zum Abfragen eines- oder mehrerer Portzustände. Die Antwort eines solchen Requests sind immer Daten im XML Format, die alle Portstände enthalten. Es folgt ein Beispiel zur Abfrage des Ports **GPIO0Value**.

Algorithm 4 Abfrage eines ADC Wertes vom Port **GPIO0**.

```
function updateStatus ()
{
    serverRequest ("/dyn/getIOState", function (xml)
    {
        var sText = "State update: #" + giNumUpdates + " Value =" + getVarValue(xml, "GPIO0Value");
        var label=document.getElementById("lblGPIO0");
        var oStateNode = document.createTextNode(sText);
        label.removeChild(label.lastChild);
        label.appendChild(oStateNode);
    } );
}
```

Quelle: ./snippets/getGPIO0Value.js

1.6.2 I/O setRequest

Zur Änderung eines Portzustandes wird ein I/O setRequest verwendet. Das folgende Beispiel zeigt, wie der Port GPIO8 auf den Wert einer HTML Checkbox gesetzt werden kann. Dazu muss im HTML Abschnitt eine Checkbox mit der id='cbGPIO8' definiert sein.

Algorithm 5 Ändern von **GPIO8**.

```
function setGPIO8 ()
{
    var x = document.getElementById ("cbGPIO8");
    var s = "/dyn/setIOState?name=GPIO8&state=";
    if (x.checked)
        s += "true";
    else
        s += "false";
    serverRequest (s, function (xml) { /* success */ } );
}
```

Quelle: /snippets/setGPIO8Value.js

Der dazugehörige HTML Abschnitt:

```
<form name="test">
<input type="checkbox" id='cbGPIO8' name="cbGPIO8" checked onchange='setGPIO8 ();' /> GPIO8
</form>
```

Quelle: /snippets/setGPIO8Value.html

A Java Script Bibliothek

A.1 serverRequest(url, callBack)

Führt einen Zugriff auf den Netport durch. Die URL kann dabei Daten enthalten die je nach Anwendung von Bedeutung sind. ("name=GPIO8", "state=true"). Die Funktion callBack wird nach erfolgreichem Datenaustausch aufgerufen.

A.2 getVarValue(xml, name) : string

Evaluiert ein XML Objekt. Rückabewert ist eine Zeichenkette, die dem Inhalt von "name" entspricht. Im Fehlerfalle gibt die Funktion folgendes zurück:

Rückgabewert	Verhalten	Bedeutung
Zeichenkette	Normalfall	Je nach "name"
-1	Fehlerfall	Kein gültiges XML Objekt - Serveraktion gescheitert.
-2	Fehlerfall	Ungültiges XML Objekt. (Serverantwort unbrauchbar.)
-3	Fehlerfall	Ungültiges XML Objekt. (Leere Serverantwort.)
-4	Fehlerfall	Name wurde innerhalb der Antwort nicht gefunden.
-5	Fehlerfall	Datentyp ist falsch. Zeichenkette erwartet aber Server lieferte etwas anderes.

B Vordefinierte Konfigurationsvariablen

B.1 PIO84-OCO

Variable Name	Default Value	Description
appCfgSEth0IPAddress	192.168.1.12	IPv4 address for builtin ethernet port.
appCfgSEth0Gateway	192.168.1.2	IPv4 network gateway for ethernet port.
appCfgSEth0Netmask	255.255.255.0	IPv4 netmask for builtin ethernet port.
appCfgBoEth0UseDHCP	false	When true ethernet will configure automatically using dhcp.
appCfgSUsb0IPAddress	192.168.167.12	IPv4 address for USB ethernet network emulation.
appCfgSUsb0Gateway	192.168.167.2	IPv4 network gateway for USB network emulation.
appCfgSUsb0Netmask	255.255.255.0	IPv4 netmask for builtin builtin usb ethernet emulation.
appCfgBoScriptExecOnReboot	true	When true script is started automatically after reboot.
appCfgBoScriptRestartOnError	false	When true script is started again automatically after error.
appCfgBoSVCEnableRemoteAccess	true	When true two or more boxes can be connected via bridgeConnect within script.
appCfgBoScriptBeepOnError	true	When true box will beep when script fails on error.
appCfgBoSVCEnableTelnet	true	When true telnet is enabled.
appCfgBoSVCEnableRemoteScripts	true	When true the TCP Port for remote scripts is enabled.
appCfgBoEnableBeep	true	When true a the box will do a short beep after reboot.
appCfgBoLoadCustomPages	true	When true the custom pages will be loaded from flash. When false no custom pages are loaded and custom-html directory is not created.
appCfgBoShowCustomPagesAsDefault	false	When true user will see custom page as root page. (Factory pages are still accessible at http://192.168.x.y/factory/index.html .)
appCfgBoDisableBeepOnScriptStop	true	Disable beeper when script fails.
phyCfgSOName0	Output0	Name of output pin in web page.
phyCfgSOName1	Output1	Name of output pin in web page.
phyCfgSOName2	Output2	Name of output pin in web page.
phyCfgSOName3	Output3	Name of output pin in web page.
phyCfgSOName4	Output4	Name of output pin in web page.
Continued on next page...		

Variable Name	Default Value	Description
phyCfgSOName5	Output5	Name of output pin in web page.
phyCfgSOName6	Output6	Name of output pin in web page.
phyCfgSOName7	Output7	Name of output pin in web page.
phyCfgSIName0	Input0	Name of input pin in web page.
phyCfgSIName1	Input1	Name of input pin in web page.
phyCfgSIName2	Input2	Name of input pin in web page.
phyCfgSIName3	Input3	Name of input pin in web page.
phyCfgSO7HighName	High	Name of state when output is high
phyCfgSO3LowName	Low	Name of state when output is low
phyCfgSO2LowName	Low	Name of state when output is low
phyCfgSO1LowName	Low	Name of state when output is low
phyCfgSO0LowName	Low	Name of state when output is low
phyCfgSO7LowName	Low	Name of state when output is low
phyCfgSO6LowName	Low	Name of state when output is low
phyCfgSO5LowName	Low	Name of state when output is low
phyCfgSO4LowName	Low	Name of state when output is low
phyCfgSO0HighName	High	Name of state when output is high
phyCfgSO1HighName	High	Name of state when output is high
phyCfgSO2HighName	High	Name of state when output is high
phyCfgSO3HighName	High	Name of state when output is high
phyCfgSO4HighName	High	Name of state when output is high
phyCfgSO5HighName	High	Name of state when output is high
phyCfgSO6HighName	High	Name of state when output is high
phyCfgSI3LowName	Low	Name of state when input is low
phyCfgSI2LowName	Low	Name of state when input is low
Continued on next page...		

Variable Name	Default Value	Description
phyCfgSI0HighName	High	Name of state when input is high
phyCfgSI1HighName	High	Name of state when input is high
phyCfgSI2HighName	High	Name of state when input is high
phyCfgSI3HighName	High	Name of state when input is high
phyCfgSI1LowName	Low	Name of state when input is low
phyCfgSI0LowName	Low	Name of state when input is low
End of list.		

B.2 PIO48-OCA

Variable Name	Default Value	Description
appCfgSEth0IPAddress	192.168.1.11	IPv4 address for builtin ethernet port.
appCfgSEth0Gateway	192.168.1.2	IPv4 network gateway for ethernet port.
appCfgSEth0Netmask	255.255.255.0	IPv4 netmask for builtin ethernet port.
appCfgBoEth0UseDHCP	false	When true ethernet will configure automatically using dhcp.
appCfgSUsb0IPAddress	192.168.167.12	IPv4 address for USB ethernet network emulation.
appCfgSUsb0Gateway	192.168.167.2	IPv4 network gateway for USB network emulation.
appCfgSUsb0Netmask	255.255.255.0	IPv4 netmask for builtin builtin usb ethernet emulation.
appCfgBoScriptExecOnReboot	true	When true script is started automatically after reboot.
appCfgBoScriptRestartOnError	false	When true script is started again automatically after error.
appCfgBoSVCEnableRemoteAccess	true	When true two or more boxes can be connected via bridgeConnect within script.
appCfgBoScriptBeepOnError	true	When true box will beep when script fails on error.
appCfgBoSVCEnableTelnet	true	When true telnet is enabled.
appCfgBoSVCEnableRemoteScripts	true	When true the TCP Port for remote scripts is enabled.
appCfgBoEnableBeep	true	When true a the box will do a short beep after reboot.
appCfgBoLoadCustomPages	true	When true the custom pages will be loaded from flash. When false no custom pages are loaded and custom-html directory is not created.
appCfgBoShowCustomPagesAsDefault	false	When true user will see custom page as root page. (Factory pages are still accessible at http://192.168.x.y/factory/index.html .)
appCfgBoDisableBeepOnScriptStop	true	Disable beeper when script fails.
phyCfgGPIO5NameH	On	Name of state when high
phyCfgGPIO5NameL	Off	Name of state when low
phyCfgGPIO11NameH	On	Name of state when high
phyCfgGPIO11NameL	Off	Name of state when low
phyCfgGPIO2NameH	On	Name of state when high
phyCfgGPIO2NameL	Off	Name of state when low
Continued on next page...		

Variable Name	Default Value	Description
phyCfgGPIO7NameH	On	Name of state when high
phyCfgGPIO7NameL	Off	Name of state when low
phyCfgGPIO4NameH	On	Name of state when high
phyCfgGPIO4NameL	Off	Name of state when low
phyCfgGPIO10NameH	On	Name of state when high
phyCfgGPIO10NameL	Off	Name of state when low
phyCfgGPIO1NameH	On	Name of state when high
phyCfgGPIO1NameL	Off	Name of state when low
phyCfgGPIO9NameH	On	Name of state when high
phyCfgGPIO9NameL	Off	Name of state when low
phyCfgGPIO6NameH	On	Name of state when high
phyCfgGPIO6NameL	Off	Name of state when low
phyCfgGPIO3NameH	On	Name of state when high
phyCfgGPIO3NameL	Off	Name of state when low
phyCfgGPIO0NameH	On	Name of state when high
phyCfgGPIO0NameL	Off	Name of state when low
phyCfgGPIO8NameH	On	Name of state when high
phyCfgGPIO8NameL	Off	Name of state when low
phyCfgGPIO10Name	GPIO10	Name of I/O line in web interface.
phyCfgGPIO1Name	GPIO1	Name of I/O line in web interface.
phyCfgGPIO3Name	GPIO3	Name of I/O line in web interface.
phyCfgGPIO5Name	GPIO5	Name of I/O line in web interface.
phyCfgGPIO7Name	GPIO7	Name of I/O line in web interface.
phyCfgGPIO9Name	GPIO9	Name of I/O line in web interface.
phyCfgGPIO11Name	GPIO11	Name of I/O line in web interface.
Continued on next page...		

Variable Name	Default Value	Description
phyCfgGPIO0Name	GPIO0	Name of I/O line in web interface.
phyCfgGPIO2Name	GPIO2	Name of I/O line in web interface.
phyCfgGPIO4Name	GPIO4	Name of I/O line in web interface.
phyCfgGPIO6Name	GPIO6	Name of I/O line in web interface.
phyCfgGPIO8Name	GPIO8	Name of I/O line in web interface.
phyCfgGPIO10PWMO	5000	Length of high time for PWM generator in us.
phyCfgGPIO10PWMP	20000	Period of PWM generator in us.
phyCfgGPIO1PWMO	5000	Length of high time for PWM generator in us.
phyCfgGPIO1PWMP	20000	Period of PWM generator in us.
phyCfgGPIO3PWMO	5000	Length of high time for PWM generator in us.
phyCfgGPIO3PWMP	20000	Period of PWM generator in us.
phyCfgGPIO5PWMO	5000	Length of high time for PWM generator in us.
phyCfgGPIO5PWMP	20000	Period of PWM generator in us.
phyCfgGPIO7PWMO	5000	Length of high time for PWM generator in us.
phyCfgGPIO7PWMP	20000	Period of PWM generator in us.
phyCfgGPIO9PWMO	5000	Length of high time for PWM generator in us.
phyCfgGPIO9PWMP	20000	Period of PWM generator in us.
phyCfgGPIO11PWMO	5000	Length of high time for PWM generator in us.
phyCfgGPIO11PWMP	20000	Period of PWM generator in us.
phyCfgGPIO0PWMO	5000	Length of high time for PWM generator in us.
phyCfgGPIO0PWMP	20000	Period of PWM generator in us.
phyCfgGPIO2PWMO	5000	Length of high time for PWM generator in us.
phyCfgGPIO2PWMP	20000	Period of PWM generator in us.
phyCfgGPIO4PWMO	5000	Length of high time for PWM generator in us.
phyCfgGPIO4PWMP	20000	Period of PWM generator in us.
Continued on next page...		

Variable Name	Default Value	Description
phyCfgGPIO6PWMO	5000	Length of high time for PWM generator in us.
phyCfgGPIO6PWMP	20000	Period of PWM generator in us.
phyCfgGPIO8PWMO	5000	Length of high time for PWM generator in us.
phyCfgGPIO8PWMP	20000	Period of PWM generator in us.
phyCfgGPIO4ADCCalibU	V	Text for unit of analog digital channel.
phyCfgGPIO5ADCCalibG	1.0	Gradient for calibration of analog channel.
phyCfgGPIO5ADCCalibO	0.0	Offset for calibration of analog channel.
phyCfgGPIO5ADCCalibU	V	Text for unit of analog digital channel.
phyCfgGPIO6ADCCalibG	1.0	Gradient for calibration of analog channel.
phyCfgGPIO6ADCCalibO	0.0	Offset for calibration of analog channel.
phyCfgGPIO6ADCCalibU	V	Text for unit of analog digital channel.
phyCfgGPIO7ADCCalibG	1.0	Gradient for calibration of analog channel.
phyCfgGPIO7ADCCalibO	0.0	Offset for calibration of analog channel.
phyCfgGPIO7ADCCalibU	V	Text for unit of analog digital channel.
phyCfgGPIO0ADCCalibG	1.0	Gradient for calibration of analog channel.
phyCfgGPIO0ADCCalibO	0.0	Offset for calibration of analog channel.
phyCfgGPIO8ADCCalibG	1.0	Gradient for calibration of analog channel.
phyCfgGPIO0ADCCalibU	V	Text for unit of analog digital channel.
phyCfgGPIO8ADCCalibO	0.0	Offset for calibration of analog channel.
phyCfgGPIO10ADCCalibG	1.0	Gradient for calibration of analog channel.
phyCfgGPIO8ADCCalibU	V	Text for unit of analog digital channel.
phyCfgGPIO10ADCCalibO	0.0	Offset for calibration of analog channel.
phyCfgGPIO10ADCCalibU	V	Text for unit of analog digital channel.
phyCfgGPIO1ADCCalibG	1.0	Gradient for calibration of analog channel.
phyCfgGPIO1ADCCalibO	0.0	Offset for calibration of analog channel.

Continued on next page...

Variable Name	Default Value	Description
phyCfgGPIO9ADCCalibG	1.0	Gradient for calibration of analog channel.
phyCfgGPIO1ADCCalibU	V	Text for unit of analog digital channel.
phyCfgGPIO9ADCCalibO	0.0	Offset for calibration of analog channel.
phyCfgGPIO11ADCCalibG	1.0	Gradient for calibration of analog channel.
phyCfgGPIO9ADCCalibU	V	Text for unit of analog digital channel.
phyCfgGPIO11ADCCalibO	0.0	Offset for calibration of analog channel.
phyCfgGPIO11ADCCalibU	V	Text for unit of analog digital channel.
phyCfgGPIO2ADCCalibG	1.0	Gradient for calibration of analog channel.
phyCfgGPIO2ADCCalibO	0.0	Offset for calibration of analog channel.
phyCfgGPIO2ADCCalibU	V	Text for unit of analog digital channel.
phyCfgGPIO3ADCCalibG	1.0	Gradient for calibration of analog channel.
phyCfgGPIO3ADCCalibO	0.0	Offset for calibration of analog channel.
phyCfgGPIO3ADCCalibU	V	Text for unit of analog digital channel.
phyCfgGPIO4ADCCalibG	1.0	Gradient for calibration of analog channel.
phyCfgGPIO4ADCCalibO	0.0	Offset for calibration of analog channel.
phyCfgSlctCtrlGPIO0V	32	GPIO pin configuration.
phyCfgSlctCtrlGPIO8V	65	GPIO pin configuration.
phyCfgSlctCtrlGPIO1V	32	GPIO pin configuration.
phyCfgSlctCtrlGPIO9V	65	GPIO pin configuration.
phyCfgSlctCtrlGPIO2V	32	GPIO pin configuration.
phyCfgSlctCtrlGPIO3V	32	GPIO pin configuration.
phyCfgSlctCtrlGPIO10V	65	GPIO pin configuration.
phyCfgSlctCtrlGPIO4V	32	GPIO pin configuration.
phyCfgSlctCtrlGPIO11V	65	GPIO pin configuration.
phyCfgSlctCtrlGPIO5V	32	GPIO pin configuration.
Continued on next page...		

Variable Name	Default Value	Description
phyCfgSlctCtrlGPIO6V	32	GPIO pin configuration.
phyCfgSlctCtrlGPIO7V	32	GPIO pin configuration.
End of list.		