User html pages into pCOWeb.
Document date: Jan-14-2005

Introduction
You can create and store your html pages in the writable space of the FLASH memory of pCOWeb.
To easy store the pages the ftp protocol is available.
As the http server of pCOWeb needs html pages to have fixed owner/group/access properties, after the transfer you'll have to launch a special task to fix these properties for all user html pages, alsewhere they will not be retrieved.

Let's start by navigating into pCOWeb tree using ftp.
To explore the directory tree of pCOWeb you can type:

ftp://<pCOWeb IP address or name to be resolved by a DNS>[Enter]

You will be promped for a login. The factory login is:
username: httpadmin
password: fhttpadmin
Note: it means that you will navigate as "httpadmin" user.
The password can be changed by accessing the pCOWeb html configuration pages. The factory password is simply "f" followed by the username. It's a rule for all factory passwords.
The username cannot be changed.

By using a graphical ftp you can see the structure of the pCOWeb tree. That's important as you have to put the html pages in the proper directory.

After the login you will see the content of the folder:

/usr/local/root/flash/http

and you should find:

admin (folder)
index.html (file 17.3kbytes)

The directory:

/usr/local/root/

is the root directory fot http server. It means that the http server will automatically add the path /usr/local/root/ to every page name (eventually containing a path) requested by a client. Look at the "index.html" example:

INDEX.HTML
When you write:

http://<pCOWeb IP address or name to be resolved by a DNS>[Enter]

the http server will automatically retrieve the following page:

(/usr/local/root)/index.html (0.3kbytes)

(the round parenthesys contain the root path for the http server; from now on this path will be omitted).

Note: this page stands inside a ReadOnly zone of the FLASH memory.
If you open this page with a text editor, you'll find inside a redirection to another page:

/http/index.html (17.3 kbytes, the same page we have found by ftp login)

The http server will then search this new page inside the directory:

/http

Actually this directory is a link to the directory flash/http: the http server will then looks into:

/flash/http

Note: the

/flash

zone is a writable space inside the FLASH memory. When you first receive pCOWeb, or after a “Format flashdisk” action, this directory will contain a factory index.html, but you can change it as you want.

> Look into /flash/http/index.html you’ll find some interesting rows.
>
> 1.
> <!--tagparser=”/pcotagfilt”-->
>
> This row must be inserted within the first 10 lines: it tells to http server to call the /pcotagfilt preprocess application (we call it “filter”) that will recognize all the Carel-defined tags and will perform the corresponding actions. If this row is not present the Carel-defined tags will be ignored and, if in your html pages you included some Carel-defined tags to see the values of the pCO variables, you will not able to see the values.
>
> 2.
> At the end of the page you can see several Carel-defined tags: they are useful to retrieve the actual value of the pCO variables. For the syntax and details, see the list of the tags below.

SITE FOR YOUR HTML PAGES
The

/flash/http

directory is the right place to store your html pages and/or directories.
You will have to first make your

/flash/http/index.html

custom page. This page will be automatically retrieved when someone types

http://<pCOWeb IP address or name to be resolved by a DNS>[Enter]

into a browser.

From the

/flash/http/index.html

you will begin to build your html tree.

If you need to build some “cgi” scripts, the right place for those files is:

/flash/usr-cgi/

If you put them in a place other than that, they will not be executed.

CONFIG PAGES
In the index.html page you can see the link:

“Web Administrator reserved area”

This link points to the page:

/config/adminpage.html

When you click on that link, a login will be asked.

The factory login is:
username: admin
password: fadmin

LOGIN FOR THE RETRIEVE OF HTML PAGES
When the http server opens a directory to retrieve a page inside, it first checks for the presence of the special file

.htpasswd

If this file is present inside the directory, it will force the http server to prompt you a login to get the permission to retrieve the content of that directory; the username / password are contained inside the .htpasswd file in a coded format.

Actually the

/config

directory contains the link

.htpasswd

pointing to

/flash/http/admin/.htpasswd

that is a writable file. If you want to change the login you can do it by the html factory configuration pages:

Web Administrator reserved area / Admin utilities / Set directories password / admin

FACTORY CONFIGURATION PAGES
The

/config/adminpage.html

contains several links to other configuration pages.
These pages let you to configure several parameters of pCOWeb. All the factory configuration pages are placed in the directory

/config

This directory is located in a ReadOnly FLASH space.

CONFIGURATION HINTS RELATED TO THE HTML PAGES
The adminpage.html gives some useful configuration features.

View used/free flashdisk space: check for the amount of free writable space inside the FLASH memory.

Set directories password: useful to set/change the logins for the directory of the writable FLASH zone; see the previous "LOGIN FOR THE RETRIEVE OF HTML PAGES" item.
-main directory: the /flash/http directory
-subdirs: you'll find a list of all the subdirs made into /flash/http

Adjust files and directories attributes access: needed to set the properties of the user html and cgi files downloaded into pCOWeb; you must launch this item to make the http server able to open those user files. Every time you create a new page or inadvertertly change the properties of some user files, it is advisable to launch this item.

Flush disk cache: every change in the user files or parameters is temporary written in RAM and later saved in FLASH. This is made to save time for the writing cycles in FLASH. This can lead to data loss or corruption if pCOWeb is powered down before the saving in FLASH. This button lets you to force the save.

Backup / Upload flash images: these two items lets you to Download a exact copy of the flashdisk to your PC and later to Upload again into a pCOWeb. The previous contents of the target pCOWeb flaskdisk will be lost. Note that the flashdisk contains also all the user specified configuration parameters (IP address, for instance).

The Upload item lets you also to upgrade the firmware into pCOWeb by picking-up the proper files from new releases from Carel. The Upload task will automatically understand what type the block is and will write it into the right place.

Format flashdisk: this item will erase ALL the writable FLASH space and will recreate the factory situation inside it. It is intended to recover from corruption of flash file system or to bring the flash disk to the factory situation, but you can also choose to keep the user configuration parameters (not the html pages): in this case the parameters will be first saved in a temporary RAM zone and further moved to the FLASH after reformatting.

Get/Set var: it is a demo page that lets you to set some digital / analog / integer variables.

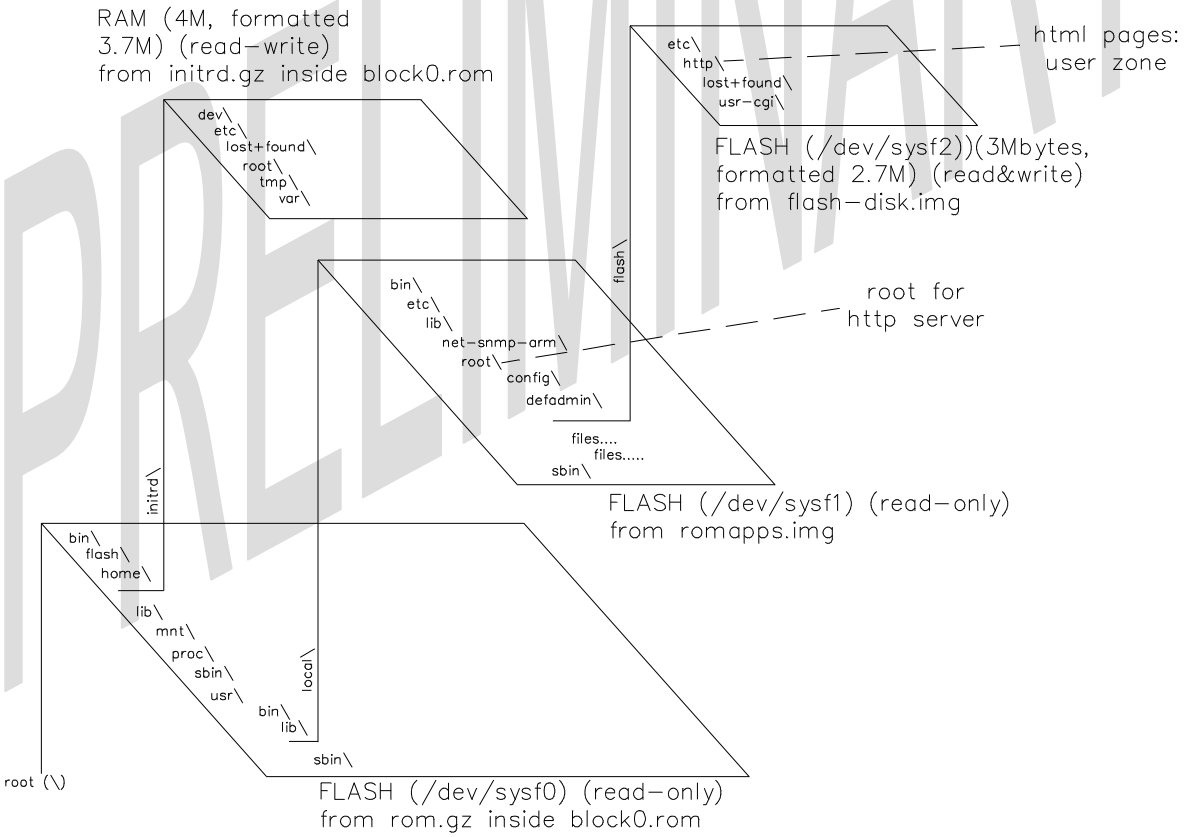

Figura 1 - The file system of pCOWeb

CAREL TAGS

READING

<%setres('string_undefined','string_ok','string_notok')%>

When the tag parser finds this tag, it will substitute it, in the html page to be returned, with one of the three user specified strings.
When a html page has been sent to pCO with some requests to write the values of some pCO variables, the string returned is "string_notok" if the pCOWeb has not received all the confirmations from pCO for the values sent to it; if pCOWeb received all the confirmations, the string returned will be "string_ok"; finally the "string_undefined " will be returned if no confirmations for variations sent to any pCO variable are pending.
To understand the confirmation mechanism it is useful to describe how the variables are managed from pCOWeb and pCO.

How pCOWeb and pCO manage the variables.

pCOWeb holds a table as a mirror for the values of all the variables of pCO. When you try to read a value for some pCO variable, actually pCOWeb shows to you the value read from that table.
pCOWeb continuously update in background the table by asking pCO for the variations.
pCO receives these asks and sends the variations to pCOWeb.
pCO is responsible of keeping track of the variations, as they arise inside it (values read from sensors, or results from calculations, or changes in parameters by the user…).
When you try to change the value of a pCO variable via pCOWeb (e.g. from a html page), pCOWeb will in turn generate a variation issue and will send it to pCO; it will represent a pCO variable variation too. At the same time pCOWeb will keep note of this issue for the particular variable involved and will waits for pCO sending back the variation notification (i.e. a message saying: the value for that variable has changed to…) for that variable. Only after the receive of all the notifications, pCOWeb will resend the html confirmation page to the browser.
It's to be noted that the variation notification is generated into the pCO by the particular user application software running into it; so, if a particular variable is not used by the particular user software application, no notification variation will be sent. When in such situation, after a timeout period pCOWeb anyhow will send back the html confirmation page, but including the "setres" tag in the html page you'll be able to discover if some notifications have not been received.

<%var('#pCO','var_type','var_index')%>

When the tag parser finds this tag, it will substitute it, in the html page to be returned, with the present value of a pCO variable or with a "U" character.
The "U" character is returned if the communication between pCOWeb and pCO is not established (wrong settings in pCO or in pCOWeb or hardware problems).
-#pCO must always be 0; it is provided only for future expansions;
-var_type:
    1=Digital (0 <= value <= 1);
    2=Analog (-3276.8 <= Value <= 3276.7, step 0.1)
    3=Integer (-32768 <= value <= 32767)
-var_index: index of the variable (1 <= index <= 207)

<%getdb('dbfilename','paramkey')%>

When the tag parser finds this tag, it will substitute it, in the html page to be returned, with the present value of a parameter contained in a database file.
pCOWeb holds some configuration parameters in some configuration database text files of the directory:
/usr/local/root/flash/etc/sysconfig
getdb will search 'dbfilename' only in this directory.
Each row in these files has the format:
<paramkey>=<paramvalue>
Note: you can generally find only some parameters into these files; the other will be present after a first user configuration regarding each specific parameter.
The database files are:
commcfg: parameters for the configuration of the serial port used to exchange data with pCO
    parity: parity (Even,Odd,None)
    speed: baud rate (300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 baud)
    data: representation bits (5, 6, 7, 8 bit)

stop: stop bits (1, 2 bit)

ifconfig: network configuration parameters

ip00: the main IP address. Note: if pCOWeb is using the DHCP, you will get the string “DHCP” or “dhcp”, or a empty string “”.

mask00: the subnet mask of ip00

ip0N / mask0N (N: 1, 2, 3): the parameters for the other three network interfaces.

gateway: the gateway IP address

dns1: the first dns IP address

dns2: the last dns IP address

userpwd: users passwords for operative system access

proot: password for “root” user

phttp: password for “http” user

pcarel: password for “carel” user

pguest: password for “guest” user

snmp: snmp system configuration parameters

rwcommunity: the read/write community string

rocommunity: the readonly community string

rwnetwork: the read/write network source host string

ronetwork: the readonly network source host string

syscontact: the syscontact string

sysname: the sysname string

syslocation: the syslocation string

trapcommunity: the community string for system traps

authtrapenable: the enabling flag for the authentification failure trap (0,1)

trapsink_0: the first system traps destination host

trapsink_1: the second system traps destination host

trapsink_2: the third system traps destination host

trapsink_3: the fourth system traps destination host

snmptrap: snmp user defined trap parameters

hostN (N=1, 2, 3, 4): the user defined traps destination hosts

communityN (N=1, 2, 3, 4): the traps community for the destination hosts

enabledN (N=1, 2, 3, 4): enabling flags for the hosts

pcoprotfail_enab: enabling flag for trap for failure of communication with pCO (0, 1)

pcoprotfail_oid: oid describing the trap for failure of communication with pCO

pcoprotfail_ack: # of retries for acknowledge of trap receive for failure of communication with pCO

pcoprotfail_time: time interval between two retries for acknowledge of trap receive for failure of communication with pCO

rN_enabled (N=1, …, 207): enabling flag for trap on digital variable #N change (0, 1)

rN_trigger (N=1, …, 207): trigger mode for trap on digital variable #N (-1=negative, 0=pos&neg, 1=positive)

rN_destH (N=1, …, 207; H=1, .. ,4): enabling flag for sending a trap on digital variable #N to the host #H (0, 1)

rN_trapoid (N=1, …, 207): trap oid for trap on digital variable #N change

rN_ack (N=1, …, 207): # of retries for acknowledge of trap receive on digital variable #N change

rN_time (N=1, …, 207): time interval between two retries for acknowledge of trap receive on digital variable #N change

rN_valoidE (N=1, …, 207; E=1, … ,5): oid of the #E embedded variable value to be send into the trap message

<%checkdbsel(‘dbfilename’,’paramkey’,’paramvalue’)%>

When the tag parser finds this tag, it will substitute it, in the html page to be returned, with the string “selected” or with a null string (“”).

The string “selected” will be used if in the “dbfilename” config file contains the “paramkey” param and this param is set to the value “paramvalue”.

See the previous tag to understand the database files and parameters.

<%rcresult%>

When the tag parser finds this tag, it will substitute it, in the html page to be returned, with the result string coming from the operative system when some special tags have been in advance invoked..

<%env%>

When the tag parser finds this tag, it will substitute it, in the html page to be returned, with several text rows showing informations on http/html environment.

<%title%>
When the tag parser finds this tag, it will substitute it, in the html page to be returned, with the content of the "title" environment variable. This variable must be initialized at the beginning of the same html page in which the <%title%> tag is to be used.

<%macaddr%>
When the tag parser finds this tag, it will substitute it, in the html page to be returned, with the MAC address value used by the operative system to initialize the Ethernet interface.

<%fw_release%>
When the tag parser finds this tag, it will substitute it, in the html page to be returned, with the string representing the firmware release of pCOWeb.

<%factdefcfg%>
When the tag parser finds this tag, it will substitute it, in the html page to be returned, with the values of the factory parameters.
The Factory parameters are used at boot of pCOWeb only when forced by the pushbutton.

<%bootvalues%>
When the tag parser finds this tag, it will substitute it, in the html page to be returned, with the strings "User" or "Pushbutton".
The string "User" will be written when the last boot read the User specified parameters; the string "Factory" will be written when the last boot read the Factory parameters.

WRITE TAGS

setdb('dbfilename','paramkey')
To be used in html forms as field names.

Example:
<input type="text" name="?script:setdb('ifcfg','ip00')" value="<%getdb('ifcfg','ip00')%>">

When the html page parser finds this string, it will attribute the value specified by the user to the parameter specified by the string "paramkey" and contained into the configuration named with the string specified by 'dbfilename'.
pCOWeb holds some configuration parameters in some configuration database text files of the directory:
/usr/local/root/flash/etc/sysconfig
The html parser will search 'dbfilename' only into this directory.
See the previous read tag g*etdb('dbfilename','paramkey')* for the details.
Note: the parameters contained into the files:
-commcfg
-ifconfig
-snmp
are loaded and used into the operative system only at boot of pCOWeb. Changes will therefore have effect only at the next boot.
The parameters contained int the file:
-snmptrap
are read and evaluated whenever the value of a digital variable has changed.

`?script:title('prova titolo')`
Setta variabile globale title nel programma pcotagfilter, variabile globale che può essere riutilizzata in un secondo tempo (nella stessa pagina html) con la tag di lettura `<%title%>`

TITLE
RCCMD
VAR