ACP INTERPRETER

- What is it?

This is a short demo which shows you the functionality of the ACP Interpreter on the SC12 Chip.
The demo contains a single HTML document which includes special syntax to be interpreted by a running executable before the document is shown.
It is an ASP like syntax, that will be replaced by the interpreted text. For that reason, it is possible to create a website which executes different small processes on the DK40 or BC440 which can be shown on the page itself.

- How does it work?

There is a special executable file running on the chip. It should be started right after the system has been booted. You have to add a device parameter, when calling the executable. Either "**dk40**" or "**bc440**", that depends on the device where you run the file. Now you can send an ACP page to the Interpreter through typing following URL syntax:

(for i.e.) : **192.168.0.2/ACP?mypage.acp**

It is important to write the first ACP in capital letters, because it is case sensitive. The name of your page comes right after the "**?**" and is not case sensitive.
Before you can see the ACP page on your browser, the Interpreter will check the whole content for the ACP tags and will interpret the text (commands) between those characters. After interpreting the ACP field, the executable will replace the tags and commands with the interpreted text. There will be no more ACP characters in the finished HTML file. And now the document is ready to be post on the browser.

- What do you need?

- First of all you need a DK40 Starter Kit or a BC440 device.
- You need the ACP Interpreter file stored on the device.
- An autoexec.bat file with an entry to execute the Interpreter after booting.
- Your ACP document
- A browser to watch your site

- Important notes:

- When calling an ACP file, you have to use the complete path starting at the Web Root directory.
- When using the Interpreter, it is recommend to set the Web Root directory to another path than "A:\", because of security reasons (i.e. prevent calling the chip.ini with the interpreter) .
- Don't use more than 300 signs in 1 ACP tag. Otherwise wrong return values could be delivered.

- Explanation of the Syntax

If you want to execute an ACP command you have to write it in special characters. It's like using HTML tags and very simple to do. The syntax looks like as followed: **<% = COMMAND %>**.
You can only use some special functions which can be interpreted. Here is a list of the syntax supported with small examples included.

---

1) Maybe you want to know if pin 5 on the DK40 is activated. You can check it with the Input function **I**. The syntax is very simple. You have to write following line to your html code.

   <% = I5 %>

   This syntax will return a "1" if the pin is active or a "0" if the pin is not active and the number will be shown on your website.
   As you see the command is written between the ACP tags **<% %>**. The "**=**" is very important. Without that sign the tag becomes invalid. The "**I**" means Input and the following number specifies the pin you want to check (0..7) or even the LED1 (L). Here is another short example:

   <% = I0 %> ← will check pin 0
   <% = I7 %> ← will check pin 7
   <% = IL %> ← will check the LED1

   Note: It is important that you write everything in uppercase.

---

2) How to activate a pin or the LED1?
   We have another function and it is called the Activate function. You can use it like the Input function. Instead of an "**I**" you call it with an "**A**". You can activate every pin (0..7) and the LED (L). You can also activate several pins with one command (only pin 0..6). Just separate the pin numbers with a "**,**"

   <% = A0 %> ← activate pin 0
   <% = A7 %> ← activate pin 7
   <% = AL %> ← activate the LED1

   <% = A0,1,2,3 %> ← this line will activate pin 0, 1, 2 and 3

   If you want to activate a pin without deactivating the other pins you have to add a **"+"** before the number(s).

   <% = A0,1,2,3 %> ← this line will activate pin 0, 1, 2 and 3
   <% = A+5,6 %> ← additionally activate pin 5 and 6

---

3) The Deactivate function has the same syntax as the Activate function. You just have to use a "**D**" to call it. Otherwise than the "**A**" function you mustn't use the "**+**" operator. An example is followed.

   <% = D7 %> ← deactivate pin 7
   <% = DL %> ← deactivate the LED1

   <% = D4,5,6 %> ← this will deactivate pin 4, 5 and 6

4) Another easy to use function is the Write function! It writes characters / strings to the document. The easiest way to describe the syntax is to show it on an example. For that reason we will take the “hello world” example.

<% = WRITE “Hello World” %> ← This will write “Hello World” to the document.

Note: The function writes every character between both quotes to the document.

---

5) The Parameter function allows you to receive parameters submitted to the URL. To submit one parameter you have to use the filename with a “**?**” and the parameter contents following. This will be saved in a variable called PAR1. Every further parameter has to be added with a “**&**” in front of it.

URL with one parameter: .....parameterdemo.acp**?**ContentsOf_PAR1

<% = PAR1 %> ← this command will receive the contents (“ContenstOf_PAR1”)

URL with more than one parameter: .....parameterdemo.acp**?**COP1**&**COP2

<% = PAR1 %> ← receive the contents of parameter one (“COP1”)
<% = PAR2 %> ← receive the contents of parameter one (“COP2”)

Note: The maximum of parameters to be sent is 10. If no parameters present, but been checked for, the function will return “0”.

---

6) Now coming to the hardest function, but still easy to handle. It’s the If-Clause function. It compares 2 terms and if they are equal, a third term will be executed. Allowed terms are every function included in the ACP Interpreter, numbers, strings and characters. The last three kind of terms needn’t have quotes, except you use capital letters. In case you have capital letters you must write the whole term in quotes. Look at following examples to make it easier to understand.

<% = A1 %> ← activate pin 1 before checking if it’s on!

<% = IF (I1 = 1) THEN WRITE “pin 1 is active” %>
<% = IF (I1 = 1) ELSE WRITE “pin 1 is not active” %>

The lines above will check if pin1 is equal to “1” (what means the pin is set) and write to the document “pin 1 is active”. The second line does the contrary. Just remember the return value of the functions ”**I**”, ”**A**”, and ”**D**”. If those functions succeed they will return a “1”.
Here is another short example how to compare a submitted parameter. We will send a parameter called “act5” or “ACT5”. The website will check if this is true and set pin 5 on.

URL with the parameter: .....actdemo.acp**?**act5 ← we use a lowercase parameter content

<% = IF (PAR = act5) A5 %>

URL with the parameter: .....actdemo.acp**?**ACT5 ← and now the same thing with uppercase.

<% = IF (PAR = “ACT5”) A5 %>

Note: Use only non ACP functions in the second term. You can also use the if-clause without executing a command at the end.

6.1) Then – Else function: you also have the opportunity to use a "**THEN**" or an "**ELSE**" after an if-clause function has been executed in the document. It will check if the last if was true or false and then execute the command. A small example will show you how it works:

```
<% = IF (I1 = 1) THEN WRITE "pin 1 is active" %>
<% = THEN WRITE "pin 1 is still active" %>
<br> <hr width=100> <br>          ← only some HTML code between
<% = ELSE WRITE "this will be..." %>
<% = ELSE WRITE "...written if pin 1..." %>
<% = ELSE WRITE "...is not active" %>
```

As you can see in the example above, you can also write several non ACP lines between an "IF" – "THEN" - "ELSE" code lines. You also needn't use a "THEN" or an "ELSE" code after an "IF" structure. It is up to the developer!

---

7) There is also a small function that retrieves the Interpreter version and the date of the last update. It's just for information purpose and not a real ACP function dealing with the DK40. The syntax is very simple and goes like this:

<% = VER %>

---

8) If you don't want the Interpreter to show the return result you have the possibility to write a **"!"** to the left side of the "**=**" in the ACP syntax. Here is a short example. It activates the pin 4 and shows the result and additionally activate pin 6 without showing the return value.

<% = A4 %> ← return value = 1
<% != A+6 %> ← return value = empty string!!!

- Explanation of the additional Syntax for the BC440

1) There are some differences between the DK40 and BC440 Input commands. As you can see on the device, there exists 16 input pins separated in two blocks (E0.* and E1.*). Additionally two those two blocks is a third one for the outputs pins (A0.*). If you want to use the Input function you have to do it like this:

   <% = IE0.0 %> ← will check pin E 0.0
   <% = IE1.7 %> ← will check pin E 1.7
   <% = IA0.3 %> ← will check pin A 0.3

   The first character is for the command Input and followed by the block type (E = input block; A = output block) with the specified number. The numbers are signed on the BC440 front panel.

   Additionally you can check the input word of the first input block E0. The syntax looks as followed:

   <% = IWE0 %> ← this will retrieve the input word.

   There is a special LED on the BC440, which can show 3 different colors. To check the LED just use the "**I**" command with a "**L**" for the LED:

   <% = IL %> ← check LED for color.

   The return value can be "**0**" what means that the LED is not active. Value "**1**" to "**3**" means active. There are three colors: (1 = Green, 2 = Red, 3 = Yellow). The return value depends on the color.

   In order to check the value of the "Run-Stop" switch, you can also use the Input function. Just write the "**I**" with a "**S**" followed. It will return a value between 0 and 16 (0 = Stop; 1 - 16 = Run)

   <% = IS %> ← retrieve switch value

   Note: You can use the Input function for all three pin blocks. But the Activation- and Deactivation function will only work with the output pin block.

---

2) Activation of the LED is another exception of the BC440. As mentioned above the LED can show 3 different colors. The can be set with the Activation command "**A**" followed by a "L" plus a number for the color.

   <% = AL1 %> ← set LED to color green
   <% = AL2 %> ← set LED to color red
   <% = AL3 %> ← set LED to color yellow

---

3) Deactivation of the LED is the easiest part of the new functions. Just use the "**D**" command followed by a "**L**".

   <% = DL %> ← deactivate LED