H0420

Programmable MP3 Audio Player for Exhibitions and Kiosk Applications

Firmware Update: Version 1.1 Build 3444

For upgrading to the new firmware, please see page 4 for the procedure.

New functionality

1. File I/O

The H0420 now includes the general purpose "File I/O Support Library" from the "desktop edition" of the PAWN scripting language, enabling reading and writing of text files and binary files on the Compact Flash card. This opens up a entire new category of applications, such as storing configuration settings on the memory card, maintaining usage log file, getting track names from the script, and updating the script itself on a live system.

The updated toolkit comes with a few examples related to file input and output, to illustrate some of the possible uses. There are examples for keeping a log file of the tracks being played (together with date and time stamps), playing files from a play list that conforms to the M3U format, and one that browses through the files on the H0420 and stores volume and balance settings in a configuration file.

2. Sampling support

One of the I/O pins may be configured to "sample" short bursts of activity on the I/O pin. For slow-changing signals, this could already be achieved with the **`@input()`** function and a script that queries the time stamps, but you would get a 1 millisecond resolution at best. The new approach to sampling simplifies the script and gives you high-resolution time stamps.

A typical application of this new feature is the support of a common infra-red remote control to operate the H0420 —adding only an IR-receiver chip and some passive components to the H0420. An example for this utilization comes with the firmware update and it is also described in detail on our web site: http://www.compuphase.com/mp3/remotectrl.htm.

3. Alarm clock functions

A new timer is available for receiving notifications at a specific time and/or a specific date. The **`setalarm()`** function activates the alarm and configures a date (partial or full) and/or a time (also partial or full). When the alarm "goes off", the public function **`@alarm()`** runs. The alarm functions can be used in conjunction with the timer functions.

4. Sound loops

The H0420 supports seamless (gapless and clickless) sound loops, provided that the MP3 files are appropriately prepared. Details on preparing the MP3 files for looping will be published on our web site shortly. If you have a pressing need for sound loops, please contact us.

5. Encrypted MP3 tracks

In some applications, you may wish to protect the audio content from being copied off the Compact Flash card onto notebooks, PCs or retail MP3 players. The H0420 offers protection by allowing you to encrypt the MP3 tracks with a strong encryption algorithm, using a 128-bit key. The key is embedded in the firmware of the H0420, and the firmware is "read-protected" itself. In other words, the key is never transferred to the H0420 and it is *not* accessible from the outside. A customer can request a unique 128-bit key upon ordering the H0420, and optionally enrich this with an additional 64-bit key that the customer chooses him-/herself. Please contact us to register a

unique customer key, or to get more information on this application.

6. Receiving binary data over RS-232

Next to receiving data in UU-encoded format (which was already supported), the H0420 now also provides the public ("event") function **`@receivebyte()`** that receives a binary stream byte-by-byte. Also new is the native function **`receivebyte()`** (without the "@" prefix) that polls for input to be received within a time-out period.

7. PAWN scripting language

The language gained embedded support for state machines (finite "automatons"), including novel and advanced concepts like state variables for state-specific data transfer. Many embedded applications are state machines or contain one or more state machines. Supporting this concept explicitly in the language eases the task of programming the state behaviour.

The PAWN debugger saw improvements in its serial interface ("remote debugging"). It now also allows uploading a compiled script over the RS-232 connection, after which this script is stored on the Compact Flash card in the H0420 MP3 player. During the development cycle, you can therefore leave the Compact Flash card in the H0420, and update the script through the same RS-232 cable that you use for debugging.

8. Miscellaneous new functions

The new **`button()`** function queries the state of any of the switch inputs. This may be convenient if you wish to check the state of a switch while handling a non-button event. Also new is the function **`reset()`**, that causes a hard reset of the H0420 MP3 player. Function **`print()`** existed already, but it has now two extra parameters, that may be of use when implementing functions for horizontally scrolling text.

9. More scripting examples

New examples on logging, play lists, storing configuration settings, and receiving commands from a remote control were already mentioned. Other new examples illustrate using switches and synchronizing digital output (e.g. turning on LEDs or relays) with music. More examples will be published on our web site in the coming months, please http://www.compuphase.com/mp3/ for more information.

Corrections and behavioural changes

1. Event functions (public functions)

- The **`@audiostatus()`** public function could miss a status change if another event also occurred (almost) concurrently. This issue is now fixed.
- The **`@timer()`** function could start iterating at very short intervals, regardless of the interval for which is was set up. This was due to a synchronization error in the standard and high-resolution timers of the device. This issue is now fixed.

2. I/O pins

- On power-on (and reboot), the I/O pins are now initialized as "input" pins. This leaves the pins at a high-impedance state, unless they are explicitly configured. This change was made in order to avoid line level conflicts with external electronics branched on these I/O pins. Scripts that configure the I/O pins properly need no change. Scripts that do not configure the pins, must be adapted in order to run correctly. Note that the default input/output configuration of the I/O pins was not previously documented.
- Configuring the I/O pins could break the initialization of an LCD (which uses 11 of the 16 I/O pins). It would only occur if the LCD was initialized before the I/O pins. This issue was due to a programming error, which has now been fixed.

3. LCD

- Vertical scrolling support has been removed from the console functions (for LCD output), in the interest of supporting larger alphanumerical LCDs.
- See also the section on I/O pins for a correction relating to configuring both the I/O pins and the LCD.

4. Native functions

- The native function **`strmid()`** would swap characters in the destination buffer if this destination buffer was very small (1 cell). This issue is now fixed.
- For compatibility with the general purpose File I/O Library distributed with the "desktop edition" of PAWN, the functions **`exist()`** and **`selectfile()`** are now called to **`fexist()`** and **`fmatch()`** respectively. The functions **`exist()`** and **`selectfile()`** are still provided for backward compatibility.

Upgrading the Firmware

When upgrading firmware, the scripts in the PAWN language should also be recompiled with the latest release of the PAWN compiler. Any "**amx**" file built for earlier versions of the firmware may not function accurately with the new firmware. The new release of the PAWN toolkit is included in each firmware update.

The procedure below describes how to upgrade the firmware using a PC running Microsoft Windows. When you do not run Microsoft Windows, please contact us for an alternative upgrade procedure.

We advise you to read through this procedure before starting the upgrade, so that you have a mental picture of the steps that are involved in the procedure.

1. Install the software

If you have not done so already, downloading the firmware update and then "open" or "run" the file. The firmware update file is a *setup* program that installs the required components. If you received the firmware update on CD-ROM, you can install it directly from the CD-ROM.

2. Connect the H0420 to a PC & remove the Compact Flash card

First unplug the H0420 MP3 player from the 5V power. Then remove the Compact Flash card from the H0420, and connect a standard RS-232 cable (not a null-modem cable) between a PC and the H0420. Plug the power connector back into the H0420 once the serial cable is connected. Do not re-insert the Compact Flash card (this is done in step 5).

If the device was already connected to the PC through the RS-232 cable and the Compact Flash card had already been removed, it is still advised to remove the power plug for a few seconds, so that the device does a full restart.

3. Run the "Firmware Update Tool"

Locate the "Firmware Update Tool" in the *Start Menu* (under *Programs / H0420 MP3 Controller)*, and run it —this tool was installed in step 1. See the screen shot below for the appearance of the Firmware Update Tool.

Make sure that the correct COM port is selected in the update tool.

4. First click "Verify", then click "Update"

Click on the button "Verify" and allow it to complete. This function checks the current firmware version and the device model and reports these to you. If all is well, it will tell you the version number of the firmware that is currently in the H0420, as well as the version number of the latest firmware. If the device already has the latest firmware, this tool will tell you so.

If the "Verify" button found no error, you can click on the button "Update". Updating the firmware may take a minute. Do not abort the program while the firmware update is processing.

After the update has completed, the program will inform you that the device will automatically reset itself after a time-out of a few seconds. If you wish to check that the firmware has indeed been uploaded correctly, you will have to wait at least this number of seconds before clicking on

the button “Verify”.

5. Recompile the PAWN script

Recompile the script and store the resulting file AUTORUN.AMX on the Compact Flash card. Make sure that the script is compiled with the up-to-date PAWN compiler. You can now re-insert the Compact Flash card in the H0420 player.

Trouble shooting — when uploading fails

If the "Verify" button times out and responds with the error message:

Unable to synchronize with the device.
Please check the serial connection.

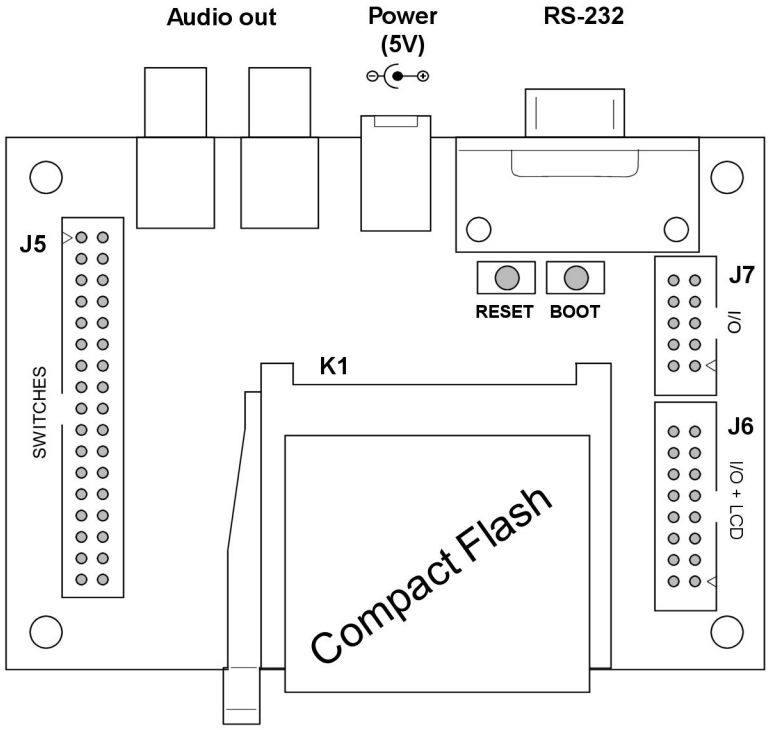
- check the RS-232 cabling
- "power cycle" the H0420 MP3 player (remove the power plug from the device for a few seconds and then re-insert it)
- run the above procedure again.

If this fails again, try to reset the H0420 while the Update Tool is busy verifying the device. That is: click on the button "Verify" and then press the "**RESET**" switch on the H0420. See below for the location of the reset button. You may also want to try to perform the procedure after a reboot of the PC, or from a different PC.

If all fails, a last option that you can try is to set the H0420 in "Boot Loader mode" (see below), and run the procedure again. In this case, the "Verify" button will inform you that it cannot find the "current" version of the firmware in the H0420, but it still allows you to update to the latest revision. Resetting the H0420 to Boot Loader mode is typically necessary when an earlier upload has been aborted or interrupted.

Resetting the H0420 to "Boot Loader mode"

The H0420 has two switches, behind the connector of the RS-232. In order to gain access to these switches, it may be necessary that you open the case in which the H0420 is mounted.



These switches must be pressed and released in the correct order:

- Press "**RESET**"
- Press "**BOOT**"
- Release "**RESET**"
- Release "**BOOT**"

It is advised to "power cycle" the device before resetting it to boot loader mode, and to remove the Compact Flash card.