Operating with the board in the HP Integral PC

The board is designed to have the microcontroller disabled in normal operation. Turn both dipswitches 6 and 7 to the on position. This disables the STM32 microprocessor and disables its access to the memory devices on the board.

On this board, when the LEDs are pointing down and the dipswtich is along the bottom, on means the dipswitch is up and off means the dipswitch is down.

Dipswitches 0 to 5 are set to choose which of the ROM images will appear to be inserted into expansion slots and thus available. It is not a one to one association of switches to ROM images.

This board runs only the internal ROM on the PC, if plugged in, when dipswitches 0 and 1 are both on. To run on the newer Unix System V image, remove the internal ROM and turn both of these switches off.

The Basic ROM is enabled by turning dipswitch 2 on.

Enabling the development (SE) ROM is done by setting both dipswitch 3 and dipswitch 4 off.

Thus the most common settings would be 001000 in dipswitches 0 to 5, on - on - off - on - on - on giving System V with Basic and SE. This is Up Up Down Up Up Up Up UP

One has the ability to instead run on diagnostic ROM images instead of Unix. These are files pulled off of boards kept by a former HP employee at the factory. He labeled the boards A and B which correspond to the two ROM images. The internal ROM must be removed to run either of these images.

Board A will perform a quick CPU test but then stalls waiting for some kind of external boot disk that is not available. Board B performs a more extensive self check and then offers menus of diagnostic tests that one could run.

The Diagnostic rom A can be activated by setting dipswitch 0 off and dipswitch 1 on. The Diagnostic rom B will be activated by setting dipswitch 0 on and dipswitch 1 off.

The board provides the ability to activate up to three additional ROM images, labeled Aux 1, Aux 2 and Aux 3. These allow the user to customize their own ROM or to make their applications nonvolatile.

Aux 1 is selected by setting dipswitch 3 off and dipswitch 4 on. It replaces the SE ROM as it uses the same address range.

Aux 2 and Aux3 are both selected by setting dipswitch 5 on and these are in addition to other ROMs you have activated.

Any pattern not listed above may disable other ROMs and not work properly.

Updating the CPLD (FPGA) chip on the board

If you have to update the Xilinx chip because Mattis Lind provides updated source code for the field programmable device, you will have to remove the board from the HP Integral PC and put it on a workbench.

First you will need to install the Xilinx ISE toolchain if you don't already have it running in your workshop. For newer versions of Windows (10 and above) they provide Oracle VM Virtualbox (VMBox) and the toolchain in a virtual machine running inside VMBox.

This works MUCH better than prior Windows versions of ISE which choked on a kludgy implementation of Linux under Windows to host their software so I highly recommend this version. Download it from https://www.xilinx.com/member/forms/download/xef.html?filename=Xilinx_ISE_14.7_Win10_14.7_VM_0213_1.zip You may have to register with AMD in order to download the software.

The updated code for the field programmable chip will come in a folder that includes the project file as well as the VHDL source. This is what you open in order to compile the new version and to download it onto the board.

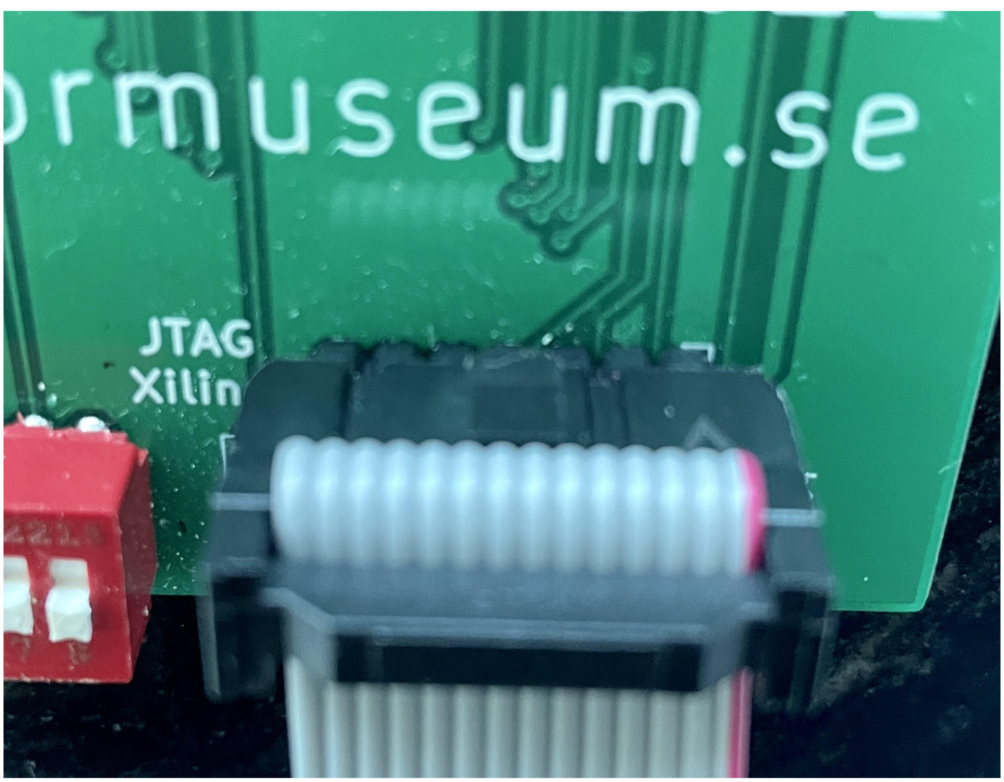
Start VMBox and then double click on the Xilinx virtual machine to start that it. It will start up with a graphical desktop interface having a few icons. Double click on the Project Navigator icon to open the toolchain.

Open the project by selecting Open Project from the File manu and then navigating to the location where you have the new project files installed. Inside that folder it will show the filename *IntegralMemExp.xise* for the project. Select that and let it open.

Under the menu Process you will select Implement Top Module which compiles the code and readies it to load into the field programmable chip on your board.

Install the programmer box by plugging in the USB cable. The ribbon cable from the programmer box is hooked to the 10 pin double row connector on the left of the board labeled JTAG-Xilinx.

Make sure that the connector is oriented with the red wire on the cable at the right as shown in this image:



You will have to supply power to the board with a bench power supply to complete the loading of the new code. Hook the power supply with the positive lead to the small pin labeled 3.3V and the ground lead to the small pin on the board labeled ground. When you turn on the supply (properly adjusted to 3.3V DC) you will see a draw somewhere between 100 and 200ma.

Select the Impact entry from the Tools menu of Project Navigator. This brings up the programming functionality in the toolchain.

Select Launch Wizard from the Edit menu. The action to take is the top one, Configure Devices which will find your programmer device and then scan for the chip.

The tool will ask you if you want to add a configuration file. Choose Yes and navigate to the xxxx.jed file for the project.

You should see a graphical image of the chip show up in the window of the tool. Right click on the icon and select Program to program the device. You should see a positive acknowledgement message. You can also perform the Verify function by right clicking the icon, running it to test the contents of the chip against the updated file you generated on the PC.

Turn off the bench power supply, shut down the Project Navigator and other software, turn off the Xilinx Virtual machine by ACPI Shutdown and quit VMBox. Unplug the programmer box from the PC. Disconnect the ribbon cable from the board. You are done.

Updating the STM32 (microcontroller) chip on the board

If you have to update the microcontroll (STM32) chip because Mattis Lind provides updated source code for the processor, you will have to remove the board from the HP Integral PC and put it on a workbench.

You will need to download and install the STM32Cube IDE software if it is not already available in your workbench environment. Get it from https://www.st.com/en/development-tools/stm32cubeide.html

You also must install the STM32Cube Programmer which loads the software onto the chip. Get it at https://www.st.com/en/development-tools/stm32cubeprog.html

After installation, start the IDE and then select Open Projects from File System from the File menu to open the project from its folder.

In Project manu, select Build Configurations. That opens a submennt where you select Set Active and that gives you two choices. Select Release.

Select Build All from the Project menu to compile the source code. Once it has completed successfully, it is time to install it on the microcontroller on the board. The binary file to load is in the Release folder inside the project folder which you opened to work on this software.

You will have to supply power to the board with a bench power supply to complete the loading of the new code. Hook the power supply with the positive lead to the small pin labeled 3.3V and the ground lead to the small pin on the board labeled ground. When you turn on the supply (properly adjusted to 3.3V DC) you will see a draw somewhere between 100 and 200ma.

Connect the ST-Link V2 programming unit to the double row 20 pin connector on the right side of the board. Make sure the connector has the red wire on the right.

In the STM32 Cube Programmer software, click on the three lines at the top left to open the left menu. Select the second item, Erasing and Programming. Browse to the file path from the left top section to pick the file *HP IPC Memory Expansion.elf*

Dipswitch 6 and dipswitch 7 should be up in order to do this.

On the right side you should see that it has found the ST-Link device hooked to the board via a USB cable to this PC. Check the box to do a verificatio of the image after it is loaded. Push the button marked Program to download the new code to the chip. You should see a message indicating successful completion.

Turn off the bench power supply, remove the ribbon cable, close the IDE on the PC and you are done. Set dipswitches 6 and 7 down to ready the board for normal use inside the HP Integral PC.

Loading ROM images to the board

If you choose to update the ROM images stored in flash chips on the board, you will have to remove the board from the HP Integral PC and put it on a workbench. You might do this to add your own code as AUX 1, 2 or 3.

Set dipswitch 6 off and dipswitch 7 off. This turns on the microcontroller and enables its control over the memory chips on the board.

The USB cable will supply power to the board where the 5V is converted down to 3.3V for the rest of the circuit.

Plug a USB cable into the board and into your PC. You should see a message that a new USB storage disk is connected and that it does not have a valid file system on it. Cancel the offer to format the drive.

Instead, you should start the Oracle VMBox and open a terminal window. Type **lsblk** and write down what entries start with sd - these can be sda, sda1, sdb and others.

In VMBox and choose the USB menu to select the STM32 USB device that is in the menu list. This attaches it to Unix instead of Windows. On the desktop, right click and select Open Terminal.

Type **lsblk** and you should see an addition device starting with sd. On my linux image it was /dev/sdb but it is important to use only the new device name which is the board masquerading as a USB storage/thumb drive, whether that is sdb or another. .

This project provides a ROM folder. Inside it are the images of the various ROMs that are to be programmed into the board. Put updates here if you have a changed ROM to install. Copy this folder (ROM) into the shared file folder that is set up to let both Unix in the VMBox and Windows share files.

In the terminal window, navigate via **cd** commands to get into the ROM folder that VMBox shares from Windows.

There is a Linux script **build.sh** that copies the different ROM images into a single image file called *flash-image.bin* that will be used to load your board. Modify if necessary for your code.

Run that script by typing **./build.sh** in the window. The file is now ready to load. Enter the command

sudo dd if=flash-image.bin of=/dev/sdX bs=1k seek=8192

where you use the name you found instead of sdX. That will copy the file over the USB cable. You will see the LEDs on the board flash in patterns as it receives the file and flashes the chips with that data.

This will program the flash memory on the board with all the ROM images so that you can select and use them via the dips-witches when the board is intalled in the HP Integral PC.

Unplug the USB cable. Shut down VMBox if you used that for a Linux/Unix shell. Turn dip switches 6 and 7 on. Set dipswitches 0 to 5 with the patterns to activate the ROM images you wish to use, usually 001000 (on on off on on on).

Reinsert the board into the HP Integral PC and make use of your new/updated ROM images.