DRAFT VERSION

Introduction

Welcome to "Pioneers Testbench" (PTB). This tutorial will guide you through a first tour and teach the basics that you need to work with the program.

Pioneers Testbench is a Graphical Debugger tool. PTB supports microcontroller's from the Freescale HCS12 and HCS12X families. PTB works with the BDM adapters:

- microlf P10
- GMV P20

The P20-adapter is further compatible with TBDML (Tiny BDM Light). So PTB should work with TBDML as well.

NOTES ABOUT PTB BETA VERSION:
Settings for different derivatives has not been finalized. PTB has been tested mainly for DG256 and XP512 but S12X support is not finalized. Please manually check (and correct if needed) settings for memory regions. Enabling/disabling of icons and menu items are not yet synchronised. Carefully watch the command window before issuing multiple command sequenses.

Installation

Install the software (PTB) BEFORE you connect a P20 BDM-adapter to your computer.
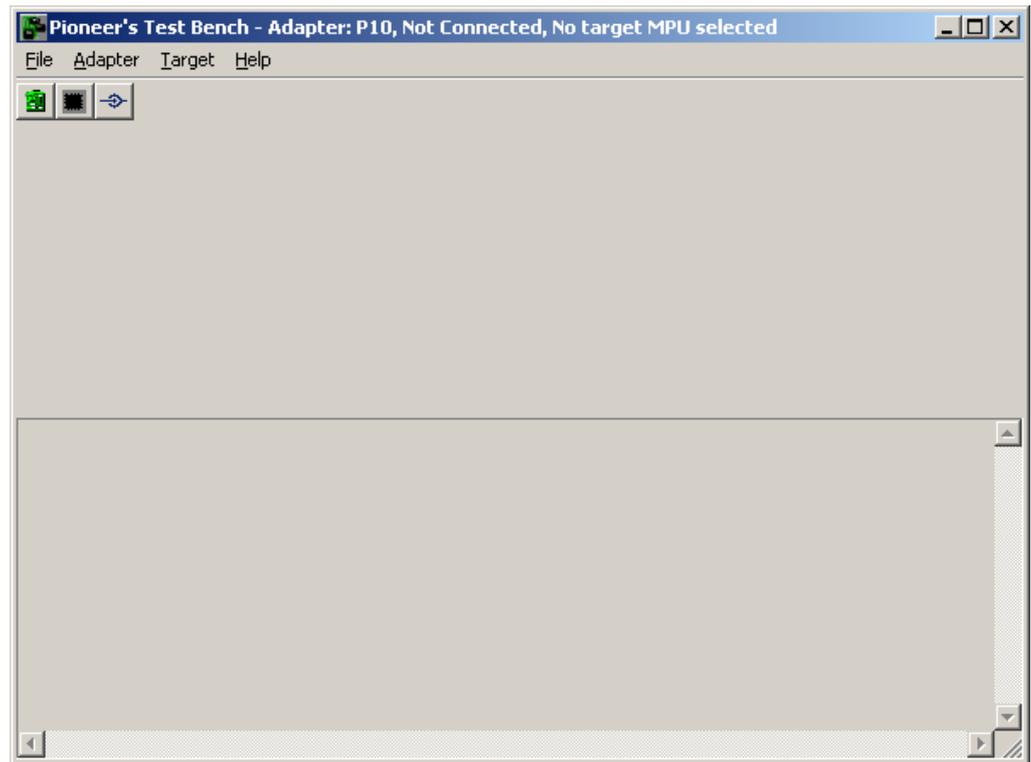The installation includes:

- Pioneers Test Bench application
- Serial drivers for P10.
- USB-drivers for P20
- Documentation files

Connecting the hardware

NOTE: Before you connect P20, to an USB port, carefully read the "Technical Description" of P20, in particular the parts that describes jumper setting for power supply.
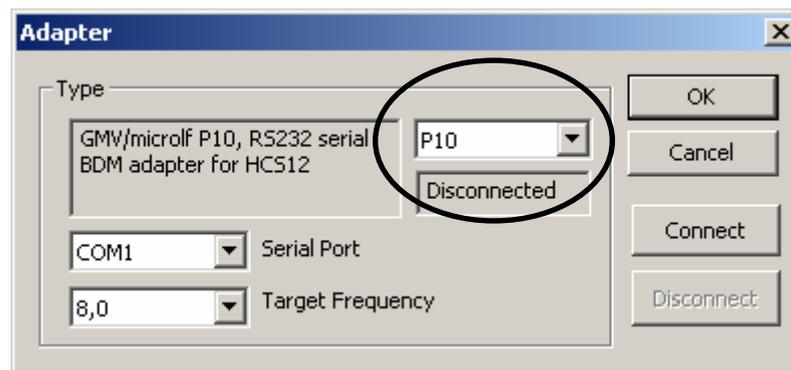
Start PTB

PTB works with two different types of adapters, the default settings is "P10", a serial communication adapter connected to COM1, with a target crystal oscillator 8 MHz. If you use another adapter OR another COM-port OR your target system operates with another frequency, read "Select your adapter" below. If these settings are correct for you, then you should continue reading "Specify your target microcontroller".
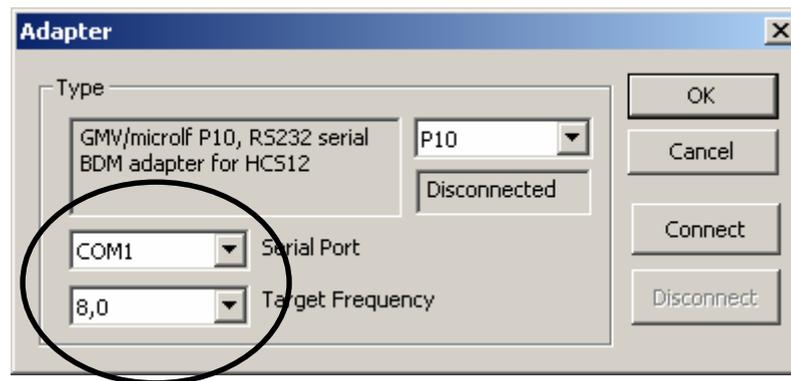
Select your adapter

First thing to do is to specify the hardware adapter you are working with.
Open the "Adapter" Dialog box by clicking the "Select adapter" (green) icon
on the toolbar, or use the menu option "Adapter->select".
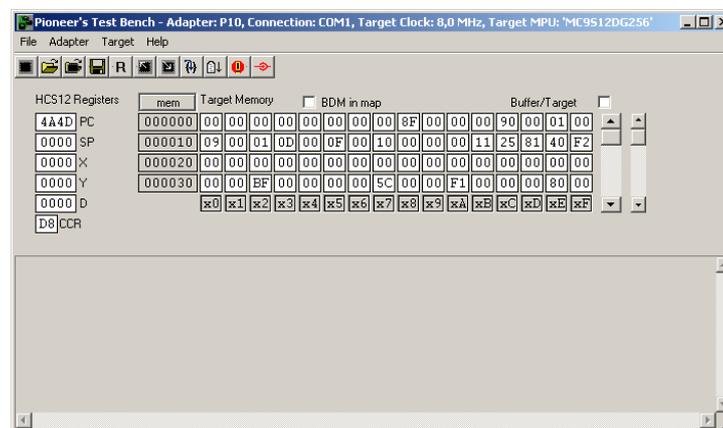

Select the adapter you are using.

P10 adapter

If you are using the P10 adapter you have to specify the serial communication
port (COMx) that the adapter is attatched to. You also have to specify the
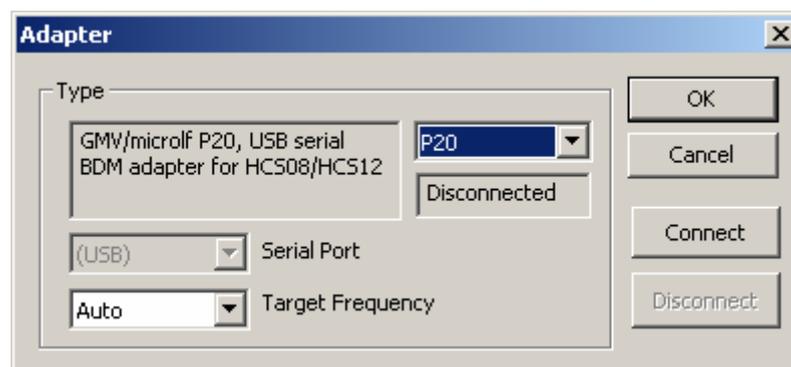crystal frequency of your target system.

Select appropriate settings and click "Connect". PTB now connects to the default target microcontroller and creates the view of the controller.

NOTE: This view might look different depending on your target system. Next step is to specify the target microcontroller. This is essential because different controllers has different memory maps which for example affects the programming function built into PTB.

P20 adapter

If you are using the P20 adapter you have several "automatic" options. For example, the adapter is capable of determining target frequency by communicating with the target. Note however that all target controllers doesn't support this feature (only applies to HCSX12 derivatives). Thus, you might have to specify it. Check the crystal frequency on your target controller board amd choose the closest match from the dropdown list.

Select appropriate settings and click "Connect". PTB now connects to the default target microcontroller and creates the view of the controller.

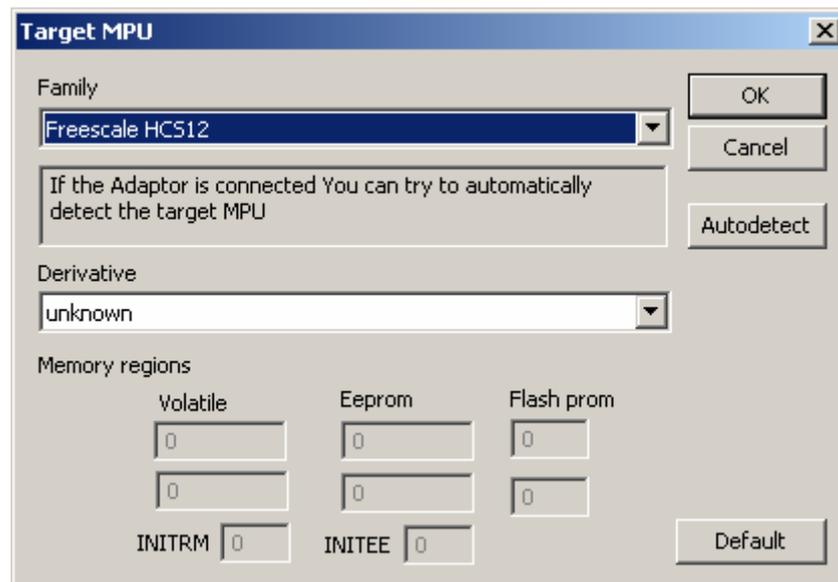# Specify your target microcontroller

Select "Target -> Select Device" from the menu.

This will launch the "Target MPU" dialog.

Select the "family" and appropriate "derivative". You can also let PTB try to detect the MPU by clicking "Autodetect". If this fails ("unknown" is still displayed, then you have to select a device as close as possible to your target. You can do this through the drop-down list.

When you have selected a MPU you can check the different memory regions for this device.

If the default settings doesn't match your needs you can specify the different memory regions affected when you are downloading (programming) the MPU, either from the buffer (described below) or from a file. PTB uses these definitions to select a proper method, e.g, programming the flash memory is slightly different than programming EEPROM etc. Yoy can also exclude memory regions from being updated during a download by unchecking the adjacent control button.

Buffer/Target Memory

PTB provides two memory views. ″Buffer memory″ is only maintained in the client while ″Target memory″ is the actual contents in the connected microcontroller. You can change the memory view by checking the ″Buffer/Target″ checkbox. Unchecked means ″Target memory″ while checked means ″Buffer Memory″.
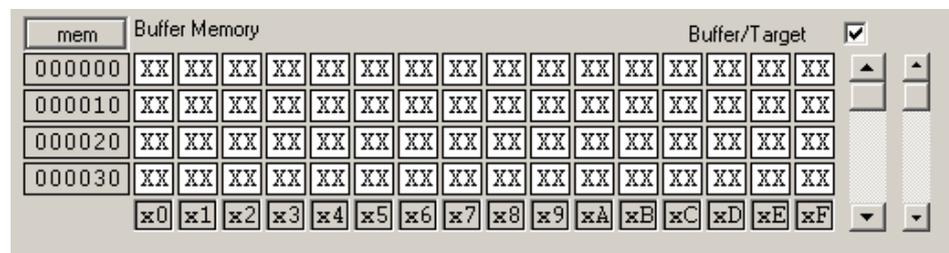
You can edit Buffer Memory.

The target memory can be updated with buffer memory contents (″Program target regions from Buffer″, described below).
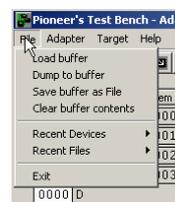
Target memory can be read into the buffer (″Dump to buffer″).

The used (initialised) buffer memory can be saved as a S-records file. (″Save buffer as file″).

Contents of memory regions which are not initialised in the buffer are displayed as ″XX″.

Buffer memory operations are managed from the **File**-menu:

Load Buffer – Load buffer memory from a file. You will be prompted for a filename, currently only Freescale S-records are supported. File extension is ″S19″ by default. Buffer memory becomes a binary image of the S-records.

Dump to Buffer – Target memory is read into the buffer. You must specify the memory regions that should be read:

Specify the memory regions in hexadecimal form (adresses) as pairs of start-end regions. The regions must be separated by a ″;″ character. You can specify up to 16 different regions. If you need to dump more regions just repeat this procedure with new region specifications.
EXAMPLE:
Read target memory C000-CFFF into the buffer. Define the region as:
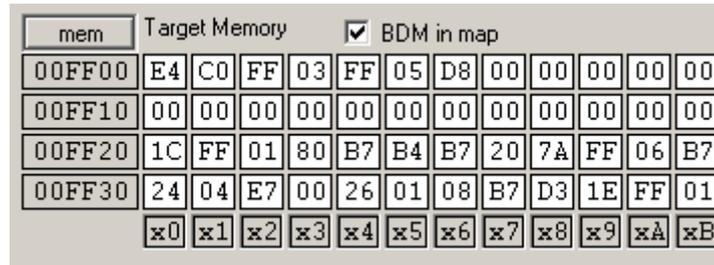0xC000-0xCFFF

Save buffer as File – Save the buffer contents as a S-record file, you will be prompted for a filename. Initialised contents of the buffer will be saved in S-records format.

Clear buffer contents – Erase entire buffer.

Inspecting and changing Target Memory

The Target memory view displays 64 bytes of continous memory.

mem	Target Memory	☐ BDM in map											Buffer/Target ☐
000000	00 00 00 00 00 00 00 00 8F 00 00 00 90 00 01 00												▲ ▲
000010	09 00 01 0D 00 0F 00 10 00 00 00 11 25 81 40 F2												
000020	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00												
000030	3F 00 BF 00 00 00 00 5C 00 00 F1 00 00 00 80 00												
	x0 x1 x2 x3 x4 x5 x6 x7 x8 x9 xA xB xC xD xE xF												▼ ▼

You can easily change the view's base address.

EXAMPLE: Display 64 bytes of memory starting at 0xC000:
Click the "mem" button and edit address

set	Target Mem
00C000	00 00 0
000010	09 00 0
000020	00 00 0

Now Click the same button ("set").

mem	Target Memory	☐ BDM in map											Buffer/Target ☐
00C000	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF												▲ ▲
00C010	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF												
00C020	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF												
00C030	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF												
	x0 x1 x2 x3 x4 x5 x6 x7 x8 x9 xA xB xC xD xE xF												▼ ▼

NOTE: Your target's memory contents might look different.

Use the left (wider) scrollbar to scroll within memory range 0000-FFFF.
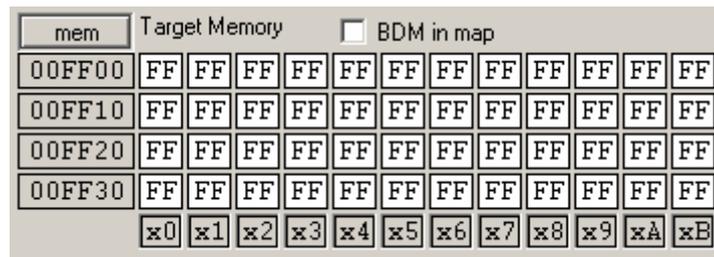Use the right memory scrollbar to scroll ranges above the low memory range.

"BDM in MAP".
The physical memory FF00-FFFF is used for Freescale BDM (Background Debug Mode). Normally, this region is part of ordinary flash memory but the BDM memory is also accesible through dedicated commands. PTB thus allows reading (and writing) from/to the BDM module.
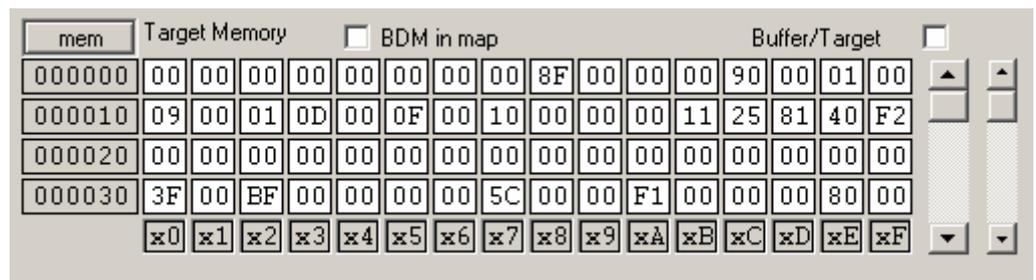
EXAMPLE: Check the "BDM in map" box and set Target Memory base to FF00.
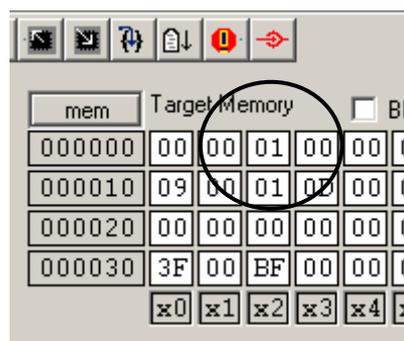

This will display the BDM-registers (FF00-FF0B) and BDM formware ROM. Unchecking "BDM in map" yuilds a view of the FLASH memory, if unprogrammed:
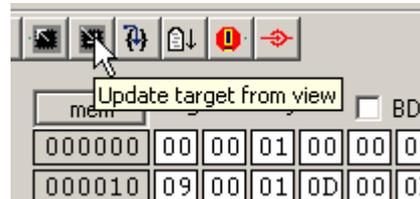

Targets volatile memory and registers can be edited in "Target Memory Mode".

EXAMPLE: Change memory location 0003 to 01:

Now click "Update target from view" to download visible memory contents (memory locations 0000-003F) to the target.

Programming/Download and Erase

There are two different ways of programming target FLASH and/or EEPROM memory.
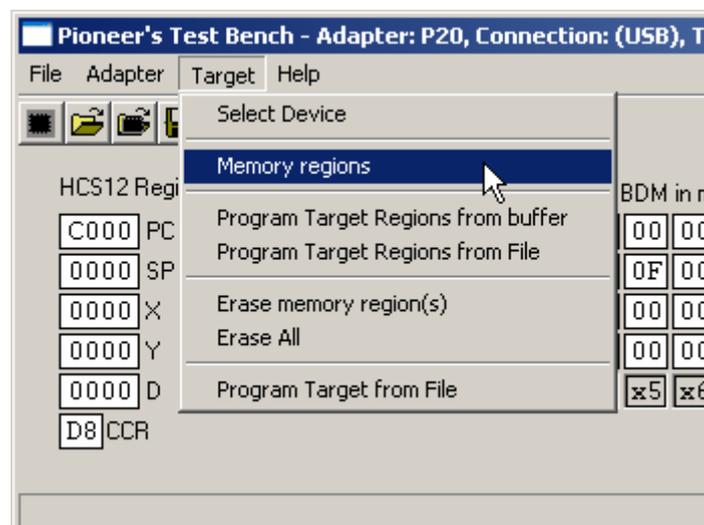- Program memory regions
- Program device from file

In any case, due to the programming algoritms it is very important that the "Selected Device" is correct, otherwise programming may fail. Note that PTB will not distinguish between programming FLASH, EEPROM or simply downloading to volatile memory. The proper algoritm for the selected device will always be used.

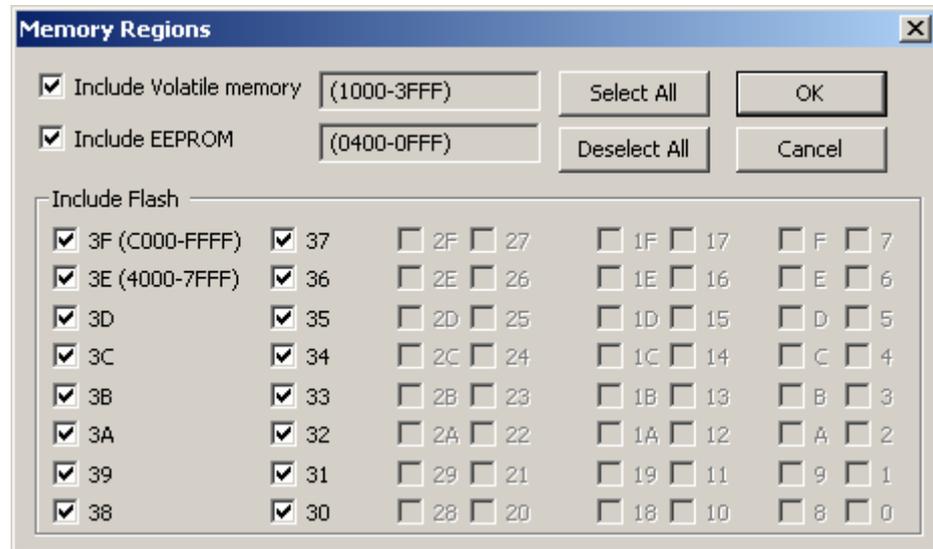There are three different commands that operates on memory regions:
- *Program Target Regions from buffer*
  Active memory regions which are also is initialised in Buffer memory will be transferred to target.
- *Program target regions from file*
  An S-records file is examined. If there are S-records for addresses that match active memory regions, these will be transferred to the target.
- *Erase memory regions*
  The active regions will be erased. In case of volatile memory, this will be filled with zeroes.

EXAMPLE: Activate volatile memory (RAM) and EEPROM memory regions. I.e. specify that subsequent memory region operations (programming/erase) will only affect volatile memory (RAM) and EEPROM memory:
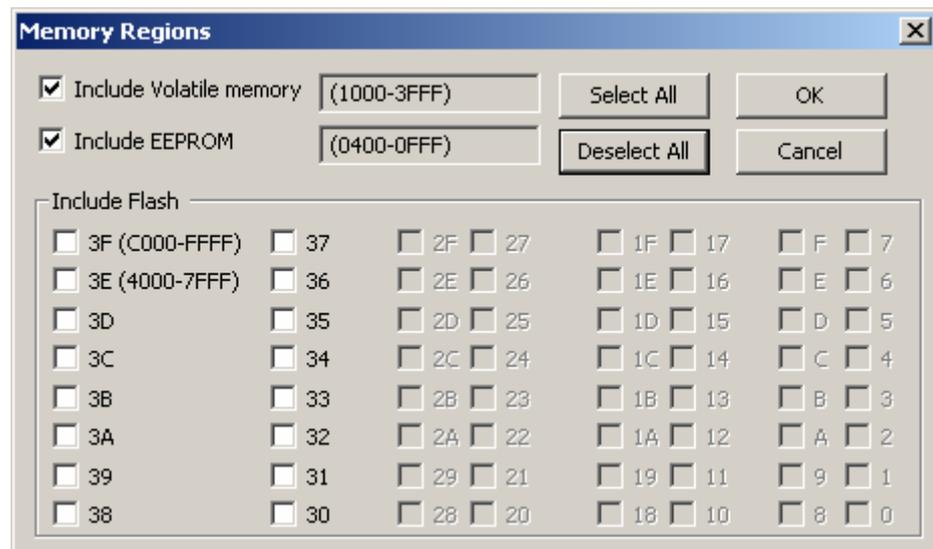Select **Target -> Memory regions**

The ″Memory regions″ dialog is launched:

Click ″Deselect All″ and then check ″Include Volatile memory″ and ″Iinclude EEPROM″:

Now, only the specified regions will be affected by subsequent commands (programming, download and erase) operating on memory regions.
For example, ″Erase Regions″ will erase selected flash blocks, the EEPROM (if selected) and fill the volatile memory with zeroes (if selected).
You can bypass the memory regions check .e.g. when mass erasing the device.
Select **Target -> Erase all**
NOTE: This will erase entire FLASH and EEPROM memories.

Programming device from a file
This command will bypass any memory regions check. Upon this command, PTB will examine an S-records file, and transfer the contents to the target using rules from the "Selected Device".
NOTE: PTB will NOT try to erase memory before programming. You must prepare the device by erasing the regions that should be programmed.
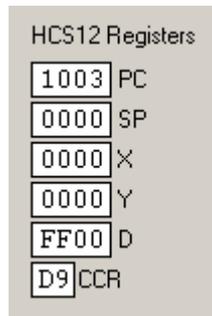
Debugging a program

A typical debug session would normally be initiated by supplying a "Reset target" command. This will force the target into a well known state.
You should then download your program. To start or single step the program:
1. Type your programs start address in the PC Register edit box
2. Click "Update target from view"
Now you can step or run the program. You can stop a running program by clicking the "Halt running program icon".

HCS12 Registers

1003	PC
0000	SP
0000	X
0000	Y
FF00	D
D9	CCR

The Target registers.

Quick Guide

	Select adapter	Select adapter (P10 or P20). This option is available when PTB is disconnected from the target. Most commands within PTB are meanigless unless there is a working connection.
	Select target device	Select target family (HCS12 or HCSX12) and derivative. Although PTB 1.0 can communicate without knowing exactly which derivate is used this affects i.e. programming algoritms and memory regions. Selecting the correct target device is therefore the most safe way to achieve correct operation.
	Connect to target (Blue)	When an adapter has been selected, use this command to start communication with the target. If the connection succeeds you will get a view displaying target registers and (part of) memory contents. If the view looks suspicious, e.g. all register contents are set to 'FF' you should disconnect and check adapter settings. This behaviour might indicate that you are using the wrong crystal frequency. You should also make sure that the device isn't "secured" (see Freescale documentation).
	Disconnect target (Red)	Use this command to disconnect from a connected adapter.
	Reset target	Force target hardware reset.
	Refresh view from target	Reload visible contents (memory and registers) and refresh the target view.
	Update target from view	Download the visible contents (registers and memory) to the target.
	Load buffer from file	An S-records file will be read and it's contents transferred (in binary form) to the buffer. Any previous contents in the buffer will be replaced.
	Dump target memory to buffer	First the "Buffer Memory Regions" dialog will prompt you for the memory regions in the target you want to upload. Then these regions will be read from the target and stored in the buffer. If buffer regions are already used, their contents will be replaced.
	Save buffer to file	Initialised memory regions in the buffer will be saved to an S-record file.
	Step instruction at PC	Send a "step" command to target. A single instruction will be executed, registers and memory view will be refreshed after the instruction
	Run from PC	Send a "go" command to target. Execution will halt either upon execution of a BGND instruction, an exception, or by issuing the "Halt target" command.
	Halt target	Send a "halt" command to the target. If the BDM is communicating properly, execution will halt, then the register and memory view will be updated.

More information

PTB Version 1.0 is scheduled for release during fall 2007.

GMV Internet WEB-site:
   http://www.gbgmv.se

Freescale Internet WEB-site:
   http://www.freescale.com