

Introduction to the uBug12x Monitor

The Adapt9S12X has a serial binary monitor, which requires the use of a client to successfully communicate with it. There are two clients available to you for this purpose; uBug12x and the code warrior debugger. The program uBug12x was written by technological arts (the makers of the adapt9S12X) and provides all the functionality required to upload and run your code. However uBug12x is buggy and does not provide all the functionality that Code Warrior does. It is therefore recommended that you use mainly Code Warrior when loading or debugging your code on the Adapt9S12X. This document is here primarily for those who maybe interested.

Ubug12X Commands

Ubug12X provides communication to the serial binary monitor stored in the 2K write protected portion of the flash within the ADAPT9S12X. The commands provided by uBug12X allow you to modify and view both the registers and the memory of the HS12. It also allows you to load, run and debug your programs.

A list of all available commands and the their purpose is shown in the table below.

Command	Purpose	
RD	Register Display	Register Commands
RM <Register Name> <Data8/16>	Register Modify	
CCR <Data8>	Set CCR Register	
D <Data16>	Set D Accumulator	
PC <Data16>	Set Program Counter	
PP <Data8>	Set P Register	
SP <Data16>	Set Stack Pointer	
X <Data16>	Set X Index Register	
Y <Data16>	Set Y Index Register	Memory Commands
BF <StartAddr> <EndAddr> <Data8>	Fill memory from StartAddr to the EndAddr with the value Data8.	
BFW <StartAddr> <EndAddr> <Data16>	Fill memory from StartAddr to the EndAddr with the value Data16.	
MD <StartAddr> [<EndAddr>]	Display the memory locations from StartAddr to the EndAddr in Byte (8bit) format.	
MDW <StartAddr> [<EndAddr>]	Display the memory locations from StartAddr to the EndAddr in Word (16bit) format.	
MM <Addr> <Data8>	Write the byte Data8 to the memory location Addr.	
MMW <Addr> <Data8>	Write the word Data16 to the memory location Addr.	Flash Commands
FBULK	Bulk Erase the Flash Memory	

Command	Purpose	
<i>FLOAD</i>	<i>Load a Banked SRecord file into Flash</i>	
<i>DEVICE</i>	<i>Get the device name</i>	<i>Information Commands</i>
<i>GO [<StartAddr>]</i>	<i>Start executing the code at StartAddr. If StartAddr is not given then it will start executing from the location referred to by the Program Counter register.</i>	<i>Execution Commands</i>
<i>HALT</i>	<i>Stop the current running program.</i>	
<i>RESET</i>	<i>Reset the Target Processor</i>	
<i>TRACE</i>	<i>Execute a single instruction at the location referred to by the program counter register.</i>	
<i>BR <Address16> <PPage></i>	<i>Set a breakpoint at the specified address Address16 on the page PPage.</i>	
<i>CON <X></i>	<i>Connects to the serial monitor communicating on COM[X], eg. "CON 1" will connect to the serial port COM1.</i>	<i>GUI Commands</i>
<i>DISCON</i>	<i>Disconnect from the serial monitor.</i>	
<i>EXIT</i>	<i>Terminates the GUI.</i>	
<i>HELP</i>	<i>List the commands available.</i>	

Uploading your Code

Once you have built your assembly source into a binary file (?abs), you will want to upload it on to the Adapt9S12X. This can be easily achieved using uBug12X. The first thing you must ensure is that the Adapt9S12X is "ON" and plugged into the serial port. You must also ensure that the switch on the side of the Adapt912 is in the "LOAD" position. You then need to connect the uBug12X client to the serial monitor, by typing "CON 1" at the command line. The command "DISCON" will disconnect.

Once connected you can simply type the command "FLOAD" and select the abs file that code warrior created. Once the code is loaded into flash you can run it by sliding the switch on the side of the Adpat9s12X to the run position and pressing the reset button.