

Application Note 108

Using Adapt912DG128 with the MX1 Wide RAM+ Expansion Card

Introduction

Because MX1 was originally designed for use with the '912B32 (80-pin chip), additional features were designed into MX1, mapped to unused pins on P2. Since the '912DG128 has 112 pins, however, all the H2 pins are already in use, so the corresponding pins on P2 are not usable by MX1. For this reason, only the BB and MIN configurations of MX1 may be used with Adapt912DG128.

To take advantage of external RAM for development purposes, you'll need to use a BDM pod. We recommend our MicroBDM12 pod or Adapt912B32 running in "pod" mode. Both of these are based on Motorola's DBug12. If you're not familiar with DBug12, refer to the DBug12 Reference Guide included on your Starter Package Disk. If you have a BDM pod from a different vendor, refer to the documentation supplied with it for details on equivalent commands for performing the steps outlined below.

Accessing MX1 RAM

When using a BDM pod, you should always start with the target (ie. Adapt912DG128) jumpered for single-chip mode (MODA and MODB jumpers both at 0). Connect your BDM pod, and apply power to the target only. The BDM pod will receive power via the ribbon cable.

Open a terminal program set to 9600 baud, and reset the pod. Specify a target crystal frequency of 8MHz, and STOP the target if it is running. Set the device type with **device da128**. At this point, the target is in Special Single-chip Mode, so the external RAM is not yet accessible. To enable it, and disable internal Flash, you need to modify a few registers: PEAR, MODE, and MISC. Use the **mm** command to modify these registers to the values shown below. Note that the rules for PEAR and MODE are "write any time except the first time" so you'll need to enter the values for these registers twice.

PEAR: 000a => 0c

MODE: 000b => f0

MISC: 0013 => 0c

Now internal Flash is off and external RAM is accessible from 0x4000 to 0xffff and at "holes" between the internal resources (refer to the '912DG128 data sheet for details). To load an s-record into RAM, use **load ;d** (make sure you use the **;d** parameter).

Since the MX1 was designed originally for the 912B32 and 912D60 boards, page emulation of internal paged 128K Flash is not implemented. For this reason, accessible RAM is limited to the unpagged 64K address space. Since the MX1 RAM chip has 128K bytes (arranged as 64Kx16) access to the other block of RAM can be made manually via JB2. This could be useful if you wanted to alternate between two different versions of code you have loaded into RAM for development purposes.

How do I set the jumpers?

JB2 can be used to select between the two banks of 32K words of RAM. It can be done through software, via PS3 (with the jumper in place), or manually, by leaving the jumper open (for upper RAM bank) or tying pin 2 of the JB2 to Ground (for lower RAM bank).

JB1 is to select between software and manual Write Protect. If a jumper is placed between pins 1 and 2 (ie. the position labelled R/W*), RAM will operate normally-- both reads and writes will work. If it is placed between pins 2 and 3 (ie. the position labelled W*), the upper half of the selected RAM bank can be protected via PS2. If you remove the jumper completely, a pullup resistor disables writes to the RAM (ie. the RAM is Write Protected).

Note: If you don't have PS2 and PS3 available (ie. you're using them for the second SCI or something else), you won't be able to take advantage of these features. In that case, you should remove JB2 and put JB1 in the MANUAL position. Of course, you could jumper them to other available port lines if desired.

Code Development

After loading your code you can just type **g xxxx** to execute it (where **xxxx** is the starting address of your code). Or, if you included the reset vector, you can move the MODA/MODB jumpers to expanded wide mode

(ie. both '1') and reset the board to cause the MCU to execute your code from RAM. This is because Flash is disabled when the MCU is reset in expanded modes. Note that the reset state in Expanded Mode has LSTRB and R/W disabled. If your program expects to do any writes to external RAM, you'll need to include an instruction at the beginning of your program to enable these signals (eg. write 0c to PEAR). You'll also need to clear the COPCTL register to disable the COP watchdog feature, or service it regularly (we recommend disabling the COP during program development).

Gotchas...

- clock stretch defaults to maximum, so if you want to optimize the performance of your code, disable clock stretch (the RAM on the MX1 has 15ns access time, so clock stretch is definitely not required)
- if you have the battery-backed RAM option (using DS1210), the power-up delay for RAM Chip Enable can be as much as 250 ms. That means your code will not start executing properly from powerup if the board is jumpered for Expanded Mode. Of course the COP will kick in and reset the MCU a few times before the RAM is fully awake anyway, so your code will start running correctly eventually.
- the fast RAM chip draws a lot of current; if your application is battery-powered this could be a concern, since the battery may drain rapidly

More than memory...

Other functional blocks are included on the Adapt912 MX1 card. Below are descriptions and some tips on using them. If you purchased a minimally-populated card, and plan to add these functions yourself, you should read the following information carefully.

Battery-backup for RAM:

- Important! the DS1210 has a 200ms start-up delay, so your program cannot execute from RAM directly *from a power-on reset*
- it is recommended that you start up in single-chip mode (with code in Flash), switch to expanded mode via the '912DG128 registers, and then jump to your program in RAM (if you want a standalone application without having to manually reset it following power-on)
- if your board doesn't have this chip (DS1210), there should be two jumpers: CE to CEO, and Vcc to VCCO, in order for the RAM to work properly

Other Functions:

- populating any other functions may interfere with the proper operation of your board(s). Consult the schematics and data carefully to determine ramifications before attempting to implement any of these features.

Toll-free: 1-877-963-8996 (USA and Canada) Phone: (416) 963-8996 Fax: (416) 963-9179
sales@technologicalarts.com www.technologicalarts.com

©2001 [Technological Arts](http://www.technologicalarts.com)