5.0 SOURCES

Computer Bulletin Board Systems

•Motorola Freeware BBS
(512) 891-3733 Austin,TX
(619) 279-3907 San Diego, CA

•Dunfield Development Systems BBS: (613) 256-6289

Internet Resources

•Technological Arts: (new product info, files, tips)
http://www.technologicalarts.com

•Motorola Freeware: http://freeware.aus.sps.mot.com/freeweb/

•Karl Lunt (SBASIC): http://www.seanet.com/~karllunt

•Kevin Ross (BDM12): http://www.nwlinkcom/~kevinro

•ImageCraft (ICC12): http://www.imagecraft.com

Publications

Motorola Fax-on-Demand: (602) 244-6609 or 800-774-1848

Motorola Semiconductor Literature Distribution Center
P.O. Box 20912, Phoenix, AZ 85036 1-800-441-2447

• CPU12 Reference Manual (CPU12RM/AD)

1 INTRODUCTION

Thank you...

Thank you for purchasing an Adapt812 Starter Package. You'll find it is a unique and useful tool for working with Motorola's 68HC812A4 microcontroller! Your questions and comments are always welcome. We provide friendly, knowledgeable technical support by telephone, fax, and e-mail to all our customers. As well, we have a comprehensive website with a resource page featuring new information, software, and links to other useful sites on the Internet. See back cover for how to contact Technological Arts.

Purpose of Adapt812...

Adapt812 was designed as an evaluation and application tool for the Motorola MC68HC812A4 microcontroller. It is unique among evaluation boards in that it will easily plug vertically into any standard solderless breadboard, with the supplied adapter. It is a fully functional, standalone implementation of a 68HC812A4 configuration, and can be treated just like a chip in your breadboard. Then it is simply a matter of wiring up the desired application circuits and downloading the appropriate code into the micro to evaluate and develop your application ideas.

Product Configurations

The Adapt812 Starter Package features 1K SRAM and 4K on-chip EEPROM program memory that is directly loadable via your PC serial port for quick and easy programming. A small firmware bootloader makes separate programming hardware unnecessary. A BOOT/RUN switch is polled after reset, facilitating the loading of an s-record file directly to the on-chip memory, via the serial port. Single-chip and expanded memory modes are selectable via two jumpers on the board. In expanded mode, the address and data bus lines are brought out to a second 50-pin

connector.

Operating in single-chip mode, all I/O ports are available for user applications. When operated in expanded-chip mode, additional memory can be added externally; however, some of the I/O ports form the address and data buses, and are thus unavailable for user applications.

Communications

An RS-232-compatible serial interface port (RX & TX only) is built into Adapt812, allowing communication with a PC, or any other device which has an RS-232 serial port. The logic-level RXD and TXD signals from the micro are also brought out to the 50-pin header, for applications such as RS-485 or MIDI. A half-duplex RS-485 serial port is implemented using the second SCI on the 68HC812. This is useful for industrial control applications.

How is Adapt812 different from other evaluation boards ?

Most evaluation and development systems available tend to use up system resources with resident monitors or debuggers. These are fine for learning in a classroom environment, but tend to be too expensive and bulky for embedding into a real application. Also, the prototyping area provided is often limited, and does not lend itself to re-usability. By contrast, we took a modular approach. With the Adapt12 system, all I/O lines and control signals are brought out to two standard 50-pin interface connectors. With several different connector options available, you can use the module in whatever way best suits your needs. With the solderless breadboard adapter, you can treat the module like a big chip, and plug it right into your breadboard. Forget soldering or wire-wrapping-- get started developing your application right away. Your prototyping space is virtually unlimited, using solderless breadboards! When you've got a design working and your ready to make it permanent, modular prototyping accessories are available, which

your s-record file called **myprog.s19** to Adapt812 via COM2, at the DOS prompt, you would enter:

p8s2 myprog

and follow the on-screen instructions. If you experience problems with the DOS batchfile on a Windows95 machine, you may need to edit the batchfile to add "\dev\" to the com port path (eg. copy %1.s19 \dev\com2). As an alternative, you can use the ASCII transfer function of virtually any terminal program. Just set the transfer rate to 1200 baud.

4.2 Using ICC12 for Windows

Before compiling, set up the linker sections with 0x800 for data, 0xf000 for text (code), and stack at 0x0c00. After compling to executable, download the resulting s-record file using the terminal window. Open the terminal window and set communication options for 1200 baud and no character delay. Choose ASCII download, and select the **.s19** file you wish to download. Reset Adapt812 in BOOT mode, and then click OK. When downloading has finished, switch back to RUN mode, and press RESET.

4.3 Using SBASIC

Use the following compiler options to specify the starting addresses for code (EEPROM, usually), variables (RAM), and stack pointer, where **myprog** is your SBASIC program filename:

sbasic myprog.bas /cf000 /v0800 /s0c00 /m6812 >myprog

After successful compilation, run as12 to create an s-record file, as follows:

as12 myprog

Then use any of the downloading methods outlined so far to load the s-record file (**myprog.s19**) into EEPROM.

Modes are selected by jumpers on MODA and MODB (Jumper Block JB1). A slide switch is provided on MODA for the convenience of those using the Adapt812 MX1 Memory Expansion Card. In addition to the jumpers, there are several internal registers which control and define the various possible configurations. For further information on using expanded modes, refer to the 68HC812A4 Technical Summary data sheet.

4 REFERENCE

4.1 How EEPROM is Programmed

Adapt812 uses on-chip 4K EEPROM for program storage. This means you can erase and re-program your code right in-circuit, without the need for special programming boards and UV erasers.

A small (256-byte) bootloader has been installed in the 68HC812A4 EEPROM by Technological Arts. This program runs whenever the chip is powered up or reset in single-chip mode, and looks at the state of PC6 (set via SW2) to decide whether to run a user program or to initiate downloading (BOOT) mode. If the pin is pulled low, the program loads an s-record file via the serial port, and "burns" it into EEPROM. If the pin is open (pulled high by the internal pullup resistor), control passes to the user program. This event is transparent to the user. The only limitations are that EEPROM Block3 ($fe00 to $feff) is not available to the user, and the COP watchdog feature of the chip is disabled. During downloading, the user reset vector is automatically intercepted and stored in a pseudo-reset-vector location established by the bootloader. It is here that the bootloader looks when the chip is reset in RUN mode, passing control to the user's program based on this vector. If the vector has not yet been initialized, the bootloader stops in an infinite loop, flashing LED D1 to indicate an error condition.

To program the on-chip EEPROM from DOS, use **p8s1.bat** (use **p8s2.bat** when using COM2). For example, to download

give you the ability to easily build fully customized, compact applications at low cost. A full range of accesories including backplanes, prototyping cards, memory expansion, and application-specific cards is available (or coming soon).

2 USING ADAPT812 WITH SOLDERLESS BREADBOARDS

The standard Adapt812 Starter Package comes with a 50-pin adapter to allow you to plug the module into a solderless breadboard ("protoboard"). This adapter may be used on either connector (H1 or H2).

CAUTION!

Never insert or remove your module from a "live" breadboard. Make sure the power is OFF !

1) Any breadboard will do; however, you will find that the kind made with a softer, more pliable plastic (such as nylon) will be easier to use and more durable. Avoid excessive removal and insertion of your board, to extend the life of your breadboard.

2) When plugging the module into your breadboard, keep it vertical and press gently but firmly, rocking the module back and forth slightly, until the pins are seated in the sockets. Use the same side-to-side gentle rocking motion, while pulling gently upward, to remove the module.

3) Plug Adapt812 into the middle area of your breadboard strip to allow maximum access on each end to all the signals. If possible, place an additional breadboard section in parallel on each side for easier wiring of your circuits. *(HELPFUL HINT: If you are using the Analog inputs, make sure to wire your analog circuits as close to these pins as possible, to keep noise levels down.)*

4) Choose a convention for wiring your power distribution buses. A logical approach is to make the inside bus logic 5V, and the outside buses GROUND. Never supply external power via J1 if you are supplying 5VDC via the breadboard connector pins. However, always connect the breadboard GROUND to the module GROUND.

5) If you are using voltages other than 5V, make sure to keep these well away from Adapt812 pins and tie-strips, to avoid accidental shorts which may damage the module.

3 TUTORIAL

Note that this manual is not meant to provide an exhaustive study of the 68HC12 family of microcontrollers, but rather to help you get started using the Adapt812 microcontroller board as a learning and application development tool for 68HC812A4, whether you're a beginner or an expert. If you are a beginner, you will benefit from additional material listed in the Reference section of this manual, and links provided on the Resource page of our website (see back cover for URL).

CAUTION!
Never insert or remove your module from a "live" breadboard. Make sure the power is OFF !

3.1 Getting Started

Adapt812 has a demonstration program already programmed into the EEPROM when you receive it. This is a useful program for testing your communications setup and monitoring & controlling the various I/O lines of the micro.

You can power the module in one of two ways:

1) supply power via the external power connector; just connect a DC voltage of 8 Volts or more (maximum 12V) to the

featured SDI pod, costing several hundred dollars, which is beyond the budget of most hobbyists, students, and many engineers Fortunately, there are a couple of low-cost alternatives. One is Adapt912, from Technological Arts. Others include the Motorola 912EVB Evaluation Board, and the BDM12 from Kevin Ross. All of these pods may be used for downloading s-records to EEPROM, and they are all supported by ImageCraft's ICC12 68HC12 C Cross-complier for Windows. See Section 5 for contact information.

3.3 Downloading Your Code to Adapt812

Once you have assembled your code with no errors, you can download the resulting s-record file (*filename*.s19) to Adapt812 using the appropriate DOS batch file provided. Connect the supplied serial cable between connector J4 on your module and COM1 or COM2 of your PC. (You can use a different COM port, but you will need to copy and edit the batchfile to reflect the COM port number you use).

Use **p8s1.bat** (for COM1) or **p8s2.bat** (for COM2). If required, use a text editor to modify these batch files to suit your needs. Note that you must use the baud rate shown in the file. The internal bootloader operates at this baud rate to allow enough time for each byte to be "burned" into the EEPROM. Alternatively, you may use Adapt912 the 912EVB, the BDM12, or ICC12 for Windows software for downloading.

Always reset the board with SW2 in BOOT mode when you are about to download. After downloading, move SW2 back to RUN and press RESET (SW1).

3.4 Adapt812 Expanded Mode

When in expanded mode, the 68HC812A4 uses ports A, B, C, D, F, & G as data bus, address bus, control lines, and chip selects, supporting up to 4 Mbytes code space and 1 Mbytes RAM.

hardware setup (ie. power supply, serial connection, PC software, etc.). It also provides you with an excellent starting point for developing your own program. Rather than starting from scratch, you can make a copy of the demo source file and remove and add features, to transform it into what you need.

Many people approach programming by spending hours or even days writing a program from scratch, then assembling it and downloading it. Then they cross their fingers and reset the board, praying everything will work. About 99% of the time, their hopes are dashed, as the board does something completely different than they expected, or worse– it appears to do nothing! At that point, they either give up, or purchase expensive diagnostic equipment, such as logic analyzers and in-circuit emulators to begin the long hard road of diagnosing and correcting their software and/or hardware mistakes.

A much more sensible– and rewarding– approach is to start with something that works, and then add new features incrementally. The modular design of Adapt812 gives you that starting point– hardware that works, and software that works. Now, if you build on that incrementally, each diagnostic step is small and manageable. And it will probably end up taking a lot less time, and costing a lot less money.

A useful debugging tool for program development is the serial communications interface (SCI). The SCI gives you a window on what's going on inside the microcontroller. Simple diagnostic messages, placed at strategic points in your evolving program, will be invaluable in debugging your software and hardware.

With the HC12, however, Motorola has added an even more powerful feature for debugging and development– the Background Debug Mode (BDM). This feature allows you to examine and modify locations and registers in your system while your program is running or suspended. It is implemented with a single-wire serial protocol, and requires a BDM interface pod to use with a serial port on your computer. Motorola makes a full-

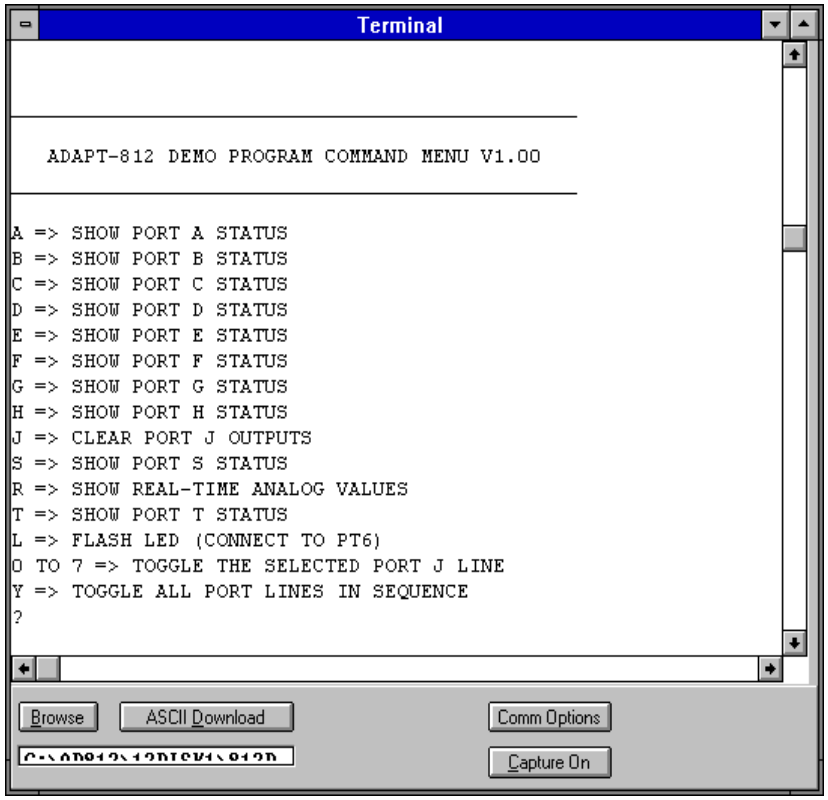

Figure 3.1 - Adapt812 Demo Menu in ICC12 Terminal Window

external power connector J1. Red is positive, and black is negative (ground). ***CAUTION! Make sure you have the polarity correct!***

2) or, supply regulated 5VDC via the appropriate pins on the 50-pin connector (H1). See Appendix A for the module pinout diagram. ***CAUTION! Double-check your connections before applying power!***

To use the demo program, make sure switch SW2 is in the RUN position. Connect the supplied serial cable between Adapt812 and a serial port on your computer. (With some PCs, you will need a 9-pin to 25-pin adapter.) Run any terminal pro-

gram (such as ProCommPlus, or the Windows Terminal program) on your PC. In your terminal program, set the baud rate to 9600, parity to NONE, # DATA BITS = 8, and #STOP BITS = 1. Press Adapt812 RESET button (SW1). LED D1 will blink twice, indicating the demo programming is running. Press the ENTER key on your keyboard. A menu of commands will appear on your computer screen, followed by a command prompt "?" symbol. Each command is activated by a single keystroke. Typing a command not listed will cause the menu to be re-displayed. Figure 3.1 shows the demo program running on ICC11's terminal screen.

Typing the letter name of an I/O port in the demo program returns the state of that port. Try putting switches on some of these input port lines. Connect one side of the switch to the port pin and the other side to ground. Note that external pullup resistors are not required, since most ports have internal pull-up resistors which are enabled out of reset (refer to the 68HC812A4 data sheet for details). In the demo program, PT6 is used as a tone output for a speaker. It is also connected to LED D1, to provide a visual output. You can drive a small piezo speaker directly by hooking one end to PT6 through a 330-Ohm resistor, and the other end to ground. When you press RESET, or type "L" when the demo program is running, you will hear two beeps from the speaker (or the LED will flash twice).

In the demo program, PORTJ is set up as all outputs. Typing a digit between 0 and 7 causes the output state of the corresponding PORTJ line to be toggled (eg. typing 3 causes PJ3 to flip to a high if it was low previously, or a low if it was high previously). This allows you to activate LEDs (when driving LEDs directly from an output port, limit the current to a maximum of 10mA with 330-Ohm current limiting resistors on each LED); or drive relays, solenoids, or motors (with appropriate driver circuits). Typing T forces all PORTJ output lines low. Typing R causes the values of all 8 analog-to-digital converter (AN0-AN7) channels of to be continuously updated on the screen (near Real-

time updates) The display will continue to be updated until a key is pressed. Analog channels (AN0-AN7) can read voltages between 0 and 5 Volts. Try putting a 10K-Ohm (or higher) pot across the VRL and VRH pins (pins 30 and 31), and connect the wiper to an AN input through a 1K-Ohm current limiting resistor; then change the pot setting, monitoring the AN values on the screen. Unused AN channels should be grounded to VRL through a minimum 1K-Ohm resistor. These inputs are not internally protected from electrostatic discharge (ESD) as the other input port lines are. *(HELPFUL HINT: Grounding multiple adjacent analog inputs is easy by plugging a bussed resistor SIP in your breadboard and jumpering the SIP common pin to VRL.)*

3.2 Writing Your First Program

If you are already experienced with the 68HC11 family of microcontrollers, writing 68HC12 programs will not present a challenge. In fact, you can use your existing 68HC11 assembly code and re-assemble it for the 68HC12. There are a couple of things to keep in mind when doing this. The first is assembler syntax. You may need to edit your source file to conform to the syntax and directives requirements of the HC12 assembler you are using. Keep in mind, too, that the register block default location is $0000 and the RAM is at $0800. This means you would initialize the Stack Pointer to $0bff. Also, the HC12 bus speed is a lot higher than the HC11. This will mean changing some initialization values for control registers and revising delay constants if you are doing software timing loops. To explore the new instructions and addressing modes of the HC12, you should refer to the Motorola CPU12 Reference Manual, available from the Motorola Literature Center or in Acrobat format from Motorola's website.

As mentioned in the previous section, a demo program resides in your microcontroller's EEPROM when you receive it. This demo program is written in Freeware AS12 assembler syntax, and is intended to provide you with an easy way to verify your