CoCoIO

Two 57,600 bps RS232 ports with FIFO buffering
Bidirectional parallel port with FIFO buffering

The CoCoIO board represents a major step forward in Color Computer interfaces. Using one of the most advaced UARTs current available, it's preformance rivals, and in some cases surpasses, that of many more 'advanced' machines.

The secret to CoCoIO's surprizing preformance is the on chip 16byte FIFO buffers provided by the National 16553 UART. These buffers reduce the interupt load on CoCo OS9 by a factor of eight. This drastic reduction in interupt service requests not only allows greater preformance from the ports themselves, but leaves much more time for other process.

Release date for the CoCoIO is targeted for this June, with an Introductory price of $130 for board and driver software. A deposit of $30 will ensure early receipt of your CoCoIO.

The demo unit is our original prototype, built to test the feasability of using the National series on the relatively slow CoCo3 buss. This board uses the simplest chip in the series, a single port 16550. As you can see from the figures below, the experiment was relatively successful!

The Tests

The following chart represents a series of tests preformed by Randy Wilson, using the prototype on a modified CoCo3 (6309 and 'native' OS9), and a LAP/M modem. The test consisted of preforming a Y-batch download to RAMdisk from a local unix site. While the download was being preformed, a Basic09 program in another window ran series of math computations. The time required for B09 to complete the test program is given as indication of system load.

	SAcia		s1655x	
	load	thruput	load	thruput
2400	28.3sec	237cps	25.3sec	235cps
9600	73.6sec	938cps	31.4sec	1200cps
57600	not available		32.6sec	5373cps
115200	not available		32.3sec	3000cps

As you can see from the figures, there is no great preformance increase at slower baud rates when only a single port is used. 115200bps seems to be just a bit to fast for the CoCo- flow control remains off much of the time, reducing both thruput and system load. Therefore, this rate is not supported or recommended.

Rick Ulland

The sl655x Driver

The new sl655x driver has been written for the most efficient usage of the National 1655x chip possible on a CoCo under OS9 L2. It supports all functions of the original AciaPak and SACIA, as well as adding new system calls to enhance BBS and hi-speed usage.

IRQ driven on both Tx and Rx, sl655x uses a 180 byte Tx buffer, and an adjustable Rx buffer(256-4096 bytes). The chips built in FIFO registers are fully utilized, allowing bps rates and thruputs previously unnattainable on a Color Computer. Baud rates of 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, and 57600 are supported. 115200bps currently works, but will not be officially supported. Of course RTS/CTS, DTR/DSR, and Xon/Xoff hardshaking are usable.

There will be versions of the driver for 6809, and both B&B and Gale Force native 6309. (The demo driver is pure 6809 code, with all features and editing intact.) There may also be additional versions with some features removed for systems with limited system space.

System Calls

SS.Break - sends a true rs232 line break for .25 seconds

SS.HngUp - shuts off DTR for .5 seconds

SS.CDSig - send signal on DCD state change

SS.CDRel - releases above signal

SS.ComSt - additional DCD and DSR reporting as per SAcia

SS.WrtBf - returns current state of send buffer, including size, fullness, and current Tx flow control state.

SS.TXFRl - Releases all Tx flow control. Useful for unjamming a BBS after DCD was dropped with flow off.

SS.WFlsh - Flushes the Tx buffer and chip fifo. Again, useful for BBS usage, better response to hot keys

SS.Fread - Special Fast read getstt call. Bypasses SCF. Works just like I$Write; no editing preformed.

SS.FWrit - As above, for writing.

(More may be added in future upgrades.)

The demo version of SC is identical to v2.2 except:
It uses the new SS.FRead and SS.FWrit calls.
It does direct screen writes, using 6309 codes.
Minor updates and bug fixs that do not affect performance.

Randy K Wilson 4/28/93 02:13