

PSoC 2002
Circuit Cellar project # 265.
Written in Microsoft Notepad ascii text. 12 July 2002 (or thereabouts).

ABSTRACT.

Project #265 attempts to produce a basic RS232/485 multidrop data acquisition and control system. The current project uses a PC as master station and Cypress CY8C26443 psoc chip for remote station operation. Up to four remotes are possible, each having an addressable ID (0-3).

Each station has the following:

Two ADC and four digital inputs for monitoring.

Two Dac (6 bit) and four digital outputs for control.

Two position jumper selectable chip ID (0-3).

Chip selected led.

Tx/Rx UART 2400 baud, 8 data, one stop bit, parity enabled.

32KHz xtal for PLL of master osc to keep baud rate stable.

5 Volt operation.

PSOC chip connects to RS485 differential line via an 75176 driver, receiver.

Some theory.

On system power up each remote psoc is in Rx mode awaiting addressing and commands (what to do? get input, produce output etc) from a master PC (or maybe an SBC).

To initiate operation a PC comm port (1-4), and software are used. It is essential to have a 16550 UART as the comm port controller because of its FIFO buffers.

As each remote can receive both address/command and plain data a means is needed to differentiate between the two. Here the parity bit comes into play, indicating whether the current incoming byte is address/command or plain data. Nice idea.

On some chips (eg PIC16C74) the parity bit can be directly checked, so all that has to be done is, say, ignore parity checking and simply "force" a 1 bit into the parity slot of the 16550 chip (a MARK) and when a remote receives this an indication is made saying the received data byte is address/command. Similarly when sending plain data a 0 is placed in the parity slot (a SPACE) and this says incoming byte is data. For example say we have a basic system of 4 remotes with up to eight possible commands. A byte could be composed of lower three bits (0-2) for command and bits 3-4 address. Odd parity says this is address command.

Bit 7 6 5 4 3 2 1 0

- - - x x | x x x

| |____command bits 000 - 111
|____address bits 00 - 11
bits 5-7 ignored.

software in the remote chip can detect this and strip the address/ commands for later processing.

Although parity error checking is lost this system works quite well (parity is not that good anyway). It's just that in the psoc chip there is no direct way of checking the parity bit. The chip automatically does a parity check for you, and gives the status in a control register. It still can be used, but you have to deliberately generate a parity error from the PC end, then check in the psoc for a parity error. In this project parity errors are deliberately inserted along with address/command byte, and put back to normal for data. PSOC firmware sorts all this out, and with suitable PC software each remote can turn on/off digital outputs and supply varying DAC along with sending back A/D and digital input status. The psoc is responsible for correct direction operation of the 75176 chip.

Operation and software.

Presently the PC operating software is in plain DOS mode (psoc485.cpp and psoc485.exe) and is a utility for checking operation of the system. Select a comm port, select a remote address and send data. The remote should show led patterns and varying DAC output. A/D and digital input is sent back to PC for basic display. A Windows style GUI control panel could be implemented (it has been done with an older PIC 16C74 system).

And thats about my 500 word limit.

Circuit cellar project # 265.