ESP Host Processor Software "Ecosystem"

Ruby Scripts

Ruby Interpreter

showlog

Console

getty

inetd services

telnetd

sshd

ftpd

Unix command shell

"User" process space

"Kernel" module space

Linux Kernel

USB

Camera Driver

Camera

Filesystem

TCP/IP

NFS

Compact Flash

SLIP or PPP

Network driver

Serial Drivers

Radio Modem

Ethernet

I^2C Gateway

CTD

ISUS

Console

Wireless

Blank Page

ESP Host Processor Hardware: the Central Processor chip

- ARM (Advanced RISC Machine) processor, with on-chip:
  - Ethernet
  - 2 RS-232 serial ports (one for console, other unused)
  - Compact Flash storage (solid-state disk) interface
  - 2 USB (Universal Serial Bus) ports (for camera & wireless)
  - PC/104 parallel bus interface (for 8 additional serial ports)

- RISC = Reduced Instruction Set Computer
  - Does little per machine instruction (ARM, MIPS)
    - Load, Modify & Store are separate operations
  - As opposed to CISC = Complex ... (Intel x86, PDP, VAX)
  - RISC does more with fewer transistors → lower power
  - RISC vs. CISC is a religious argument

ESP Hardware: RS-232 serial

- RS-232 Serial Ports
  - RS-232 = Recommended Standard 232
  - It dates from 1969 !! It's ancient and shows its age.
  - http://en.wikipedia.org/wiki/RS-232
  - Originally specified 25-wires, reduced to 9, minimally 3
  - 3-wire variant ubiquitous in oceanographic equipment
  - Transmit Data(TxD) + Receive Data(RxD) + Ground(GND)
  - Self clocking means clock is embedded in data stream
- 3-wires provide no way for receiver to throttle data
  - receiver may lose data at high bit rates
- ESP requires the 5-wire variant for high rate ports
  - Adds CTS (Clear-to-Send) and RTS (Request-To-Send) signals
- ESP also requires DCD (Data Carrier Detect) for Radio Modems
- Related to RS-422 and RS-485

ESP Hardware: PC/104

- PC/104 board stacking parallel computer bus standard
  - Industry first appeared in AMPRO products in 1987
  - Signals same as those used in the original IBM PC/AT
  - Stacking connectors eliminated need for a backplane
  - Since enhanced to support PCI and PCI-Express
    - But we laggards continue to use the 1987 vintage original
  - Also used by the AUV project at MBARI, et. al.

- PC/104 is an Intel centric standard
  - Many signals came directly off i8088 and i8086 CPU chips
  - Few ARM CPU boards support it
  - Ours supports only its most commonly used pins
  - An FPGA is used to translate PC/104 signals

ESP Hardware: FPGAs and CPLDs

- FPGA = Field Programmable Gate Array
- CPLD = Complex Programmable Logic Device
  - Huge array of logic switches on a chip
  - Organized into generic functional blocks
  - Functional blocks are connected according to data !!
  - These data may be stored permanently in the chip
    - Or read off another storage device when it is reset.
- Massive software packages translate circuit diagrams and, more recently, specialized textual programming languages like VHDL and Verilog, into the required data for the FPGAs.
- These packages often include simulation capabilities
- All the benefits and frustrations of software brought to hardware
- Hardware without soldering, hacks often impossible.
- Nearly killed the discrete logic chip business

ESP Hardware: UARTs aka USARTs

- UART = Universal Asynchronous Receiver/Transmitter
  - Also abbreviated USART
  - A chip that interfaces to RS-232 (or similar) serial ports
  - Not to be confused with USB
  - Better UARTs have large data queues to avoid data loss
    - This is especially important for CPUs running multitasking OS's like Linux, Unix, and Windows.
  - UARTs built-in to ARM chip have only 16-byte FIFO, which proved unable to cope with incoming data at 115k bits/sec during writes to the compact flash disk.
    - Due to long periods where CPU interrupts are disabled
  - So, ESP must use the external PC/104 bus octal UART chip for interfacing with the I2C gateway
    - It has a 64-byte FIFO queue.

ESP Hardware: Universal Serial Bus

- USB = Universal Serial Bus
  - 4-wire, very well standardized interface
  - Separate Data and Clock. No need for handshakes
  - Devices identify themselves (ala MOOS pucks :-)
  - Requires **much** more software than RS-232
  - Provides limited power (up to 500mA @5V)
  - A single host connects to many slaves via a tree of hubs
    - Hubs may be powered or get their power from the bus
  - Slaves cannot communicate with each other.
    - Slaves respond only to host initiated queries
  - Each wire segment may be at most 5m long
    - Max of five hubs between any slave and the host
  - Full speed = 12Mbits/s
  - High speed = 480Mbits/s (~250Mbits/s in practice)

ESP Hardware:
The Motherboard aka 'Sleepy' board

- Large, round circuit board mounted at top of ESP core
  - Contains power supplies and replaceable fuses
  - PC/104 Linux host mounts on top
  - Low-power 8-port UART connects to the PC/104 bus
    - Contextual sensors relay isolated
    - Internal (analytical module) ports are not
- Sleepy itself is just a tiny MSP430 microcontroller on the back
  - Connects to the host ARM processor via RS-232 serial port
  - Does not connect to PC/104 bus at all
  - Is the I^2C Gateway for the host processor
    - Translates RS-232 serial commands <-> I^2C bus
  - Controls power to external devices and the Host itself
- The ESP core may be controlled via an external host computer
  - Just plug it into the port where the internal host would go
  - Only problem is access to the 8-port UART

ESP Hardware: Dwarf Boards

- Contain MSP430 processor to close real-time control loops
  - Networked to each other via a single ribbon cable
    - Cable contains I^2C bus and 3.3V logic power
    - Raw motor power supplied on separate 4-pin Molex
  - One RS-232 serial port for debug console
    - Or passthru to other serial devices (PCR, Elmo, etc.)
  - Two DC brushed servo motor control channels
    - PWM output, quadrature input
  - Daughter boards may be mounted on each for:
    - Four Rotory valve / Eight H-bridge control
    - Analog sensor signal conditioning
    - Dpress board for Deep Water Sampler
  - All run the exact same 'C'-language firmware
    - Except if functioning as an I^2C gateway (ala DWSM)
    - Host configures each according to its I^2C address

ESP Hardware: I^2C Bus

- A tiny Local Area Network for dwarf boards and sensors
  - I^2C = Inter-Integrated Circuit (see http://www.i2c-bus.org)
    - 3-wires = Clock, Data, Ground. Collisions impossible!
    - Extra wires on ESP ribbon cable are for noise reduction
    - We run it at 100 kBits/s, can run 400kb/s (even 3.2Gb/s)
  - Each device on must have unique (typically fixed) address
    - No standard facility for auto configuration like DHCP
    - Dwarf boards have DIP switches to set their address
      - Gateways usually use address 0x20 (32 decimal)
      - Other dwarves use (0x21 – 0x2f)
    - Potential address collisions among manufactures
  - True multi-master network (unlike USB)
    - Dwarves could talk to each other or other I^2C sensors
  - We 'enhance' the standard with our CRC error checking

ESP Host Software: Linux Kernel

- Arbitrates and abstracts access to hardware by 'processes'
  - Most hardware resources can be abstracted and shared
    - Memory, disk, networks, USB
    - Virtual memory, TCP/IP networking
  - Rarely, precesses are allowed direct, exclusive access
    - RS-232 Serial ports, some USB endpoints
  - Starts only one, single process at boot time
    - User processes spawn children to create every other
    - Signals may interrupt (or kill) processes
      - One special signal (-KILL) cannot be ignored
    - Processes live in a protected virtual memory space
  - Modules may be dynamically loaded and removed
    - Drivers provided lowest level access to hardware
    - Also abstractions like net protocols and filesystems
    - Live in an unprotected virtual memory space