Designing a Robust, Low-Power, Embedded System for Data Acquisition and Communications

Dr. Todd Valentic

Center for Geospace Studies
SRI International

Polar Technology Conference
April 16–17, 2009

Introduction

OBuoy - Barrow, Alaska

- Remote, autonomous instruments in the Arctic
- Real-time data and status
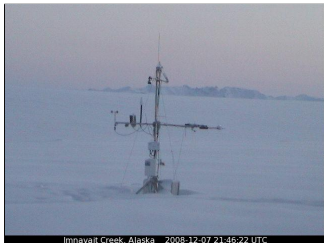- Low-power, battery and renewable energy
- Scale for sensor networks

Designing a Robust, Low-Power, Embedded System for Data Acquisition and Communications

Scope

- Small systems

- Multiple sensors (1-10)

- Multiple sites

- Few MB/day

- Real-time monitoring
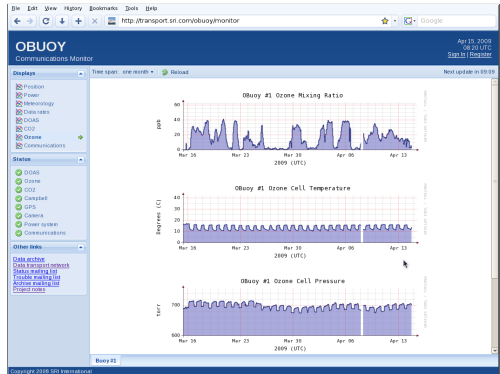
Techniques Scale

Small Systems to Large Systems

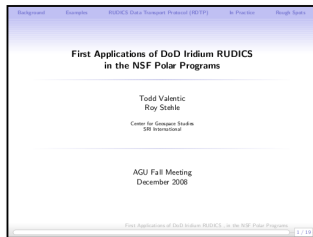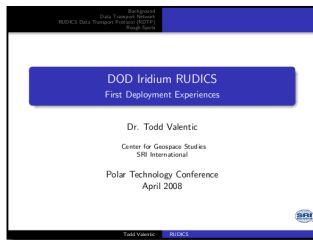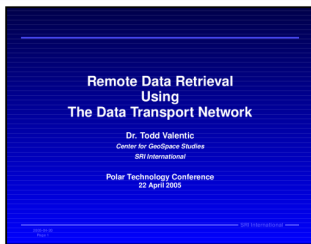Imnavait Creek, Alaska   2008-12-07 21:46:22 UTC
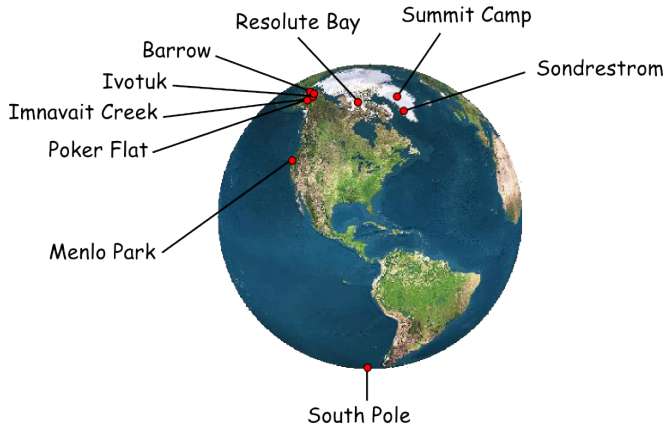

- 10 sensors

- Low power

- 1MB/day

- 4000 transmit elements

- Peak power of 2MW

- 1TB/day

# Past Talks

Deployments

Vendor and Board Selection

Starting Off Right

- Platform for embedded Linux

- Good, open support

- Active mailing lists

- Relevant documentation

- Active in open source community

- Drivers
    - Bad: binary only
    - Good: open source
    - Best: mainline kernel

- OS and distribution agnostic

Example: Technologics TS-7260

- 200MHz ARM9 CPU

- PC/104 expansion bus

- 128MB SDRAM

- 128MB NAND Flash

- Ethernet

- 2 USB ports

- 3 serial ports

- 30 DIO lines

- 2 12-bit ADC

- SD socket

- Watchdog timer, SPI bus
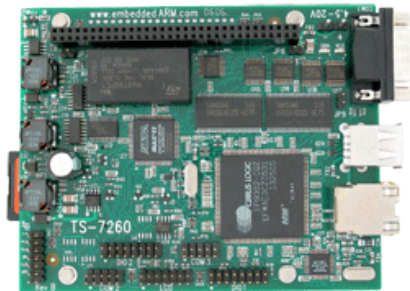
Low Power Operations With The TS-7260

- Ability to turn off USB, Ethernet, UART, LEDs
- Scale CPU clock frequency 14MHz–200MHz

Normal, everything on	5V	220mA	1.1W
Ethernet off	5V	120mA	0.6W
UART off	5V	220mA	1.1W
CPU 166MHz	5V	190mA	0.95W
CPU 42MHz	5V	170mA	0.85W
CPU 14MHz	5V	150mA	0.75W
Ethernet off, CPU 14MHz	5V	50mA	0.25W

Embedded Linux

- Easy to "roll your own"

- Footprint: 1MB base system with kernel, typical 40MB full system

- Busybox - most commands, shell, editor (vi)

- uClibc - lightweight C library

- Buildroot - framework for building everything
  - Set of makefiles
  - Builds a complete cross-compiler toolchain
  - Package selection
  - Add local packages
  - Creates root filesystem

```
buildroot.uclibc.org
     busybox.net
```

File Systems

Resiliency To Power Failure

- Flash-based storage

- Raw NAND chips

- JAFFS/YAFFS/UBIFS

- Avoid SD/CF cards as prime storage

- Minimize writes
    - No atime
    - No journal (ext3)
    - Use ramdisks for logs
    - No swap

Memory Usage

Understanding Memory Pressure

- No swap on flash-based systems

- Free memory is wasted memory

- Complicated accounting
  - Caching layers (slabs)
  - Shared pages

- `/proc` access to kernel internals

Designing a Robust, Low-Power, Embedded System for Data Acquisition and Communications
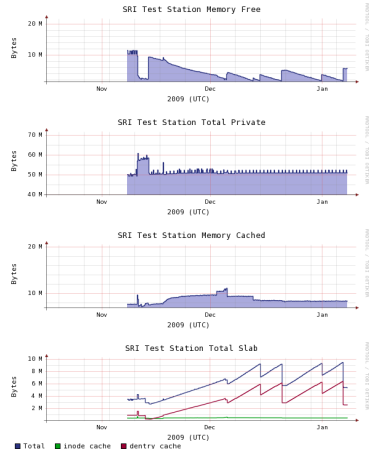
Development Languages

- "Batteries included"

- Highly readable and maintainable

- Broad range of applications
  - System level
  - Shell and text processing
  - Networking and web
  - Visualization
  - Numerics

- Don't have to switch gears between problem domains

- Some tricks needed to get to cross-compile

Revision Control

- Subversion, git, mercurial (DVCS)

- Track buildroot, root filesystems, updates

- Ability to go back to any installed version

- Reliably duplicate deployed systems

- Helps manage large numbers of deployed systems

- Requires discipline

- Use Trac for project management
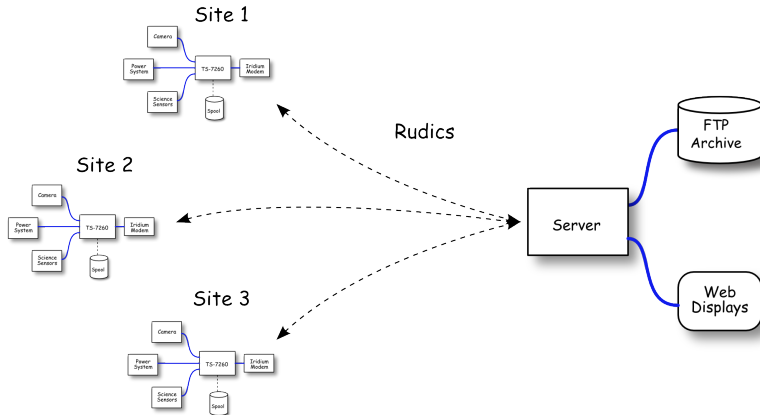 - Wiki
  - Bug tracker
  - Source code browser

`trac.edgewall.org`

Application Design

Data Collection, Transfer and Display
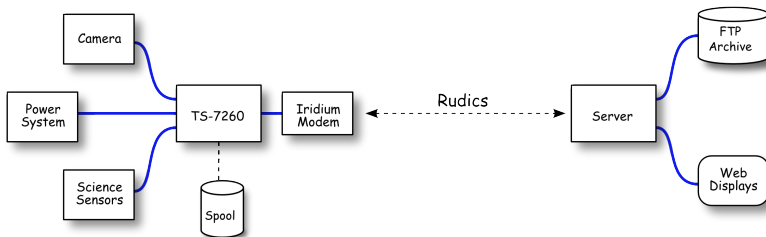
Multiple Sites

Data Transport Network

transport.sri.com/TransportDevel

Application Framework

- Hierarchical organization of programs

- Configuration and log files

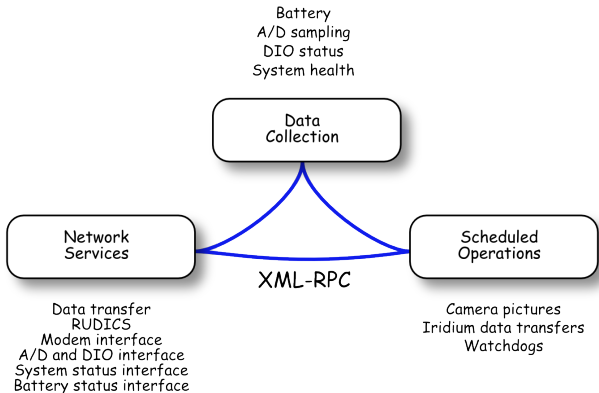- Toolbox of customizable components

Store and Forward

- Handle network outages

- On demand connections
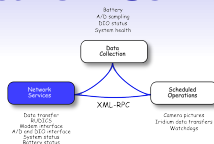
Publish and Subscribe

- News groups

- Separate producers from consumers of data

- Standard access to data streams

Embedded Software Organization

Imnavait Creek Power System Monitor

Battery
A/D sampling
DIO status
System health

Data
Collection

Network
Services

XML-RPC

Scheduled
Operations

Data transfer
RUDICS
Modem interface
A/D and DIO interface
System status interface
Battery status interface

Camera pictures
Iridium data transfers
Watchdogs

Network Services

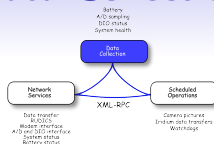
Network API for each sensor

- Uniform interfaces
- Multiple client access
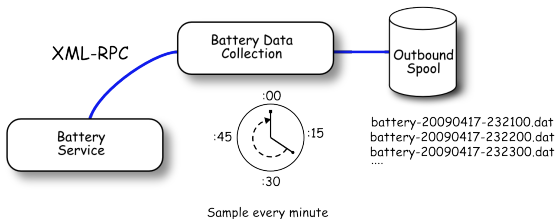- Distributed
- Directory lookup

System status

```
system = directory.lookup('system')
system.status()

{'cpuload': 0.25,
 'mem.free':  120030,
 'mem.cached': 230495,
 ....
}
```

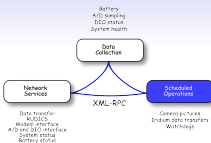# Data Collection

- Independent collection programs
- Different sampling rates
- Output files left in spool directory

Scheduled Operations

- Camera

- Iridium transfers

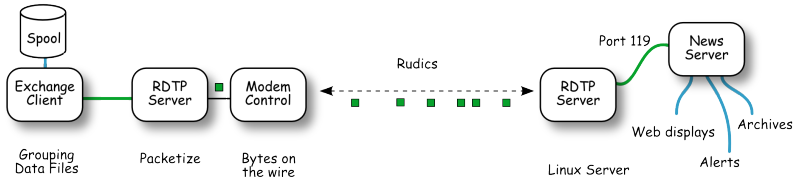- Software updates

- Software watchdog

Example: Iridium Schedule

```
[schedule]

repeat.rate:    2:00:00
repeat.sync:    True
repeat.offset:  00:00:00

repeat.events:  00:00:00 | 00:30:00 | iridium
                00:45:00 | 00:15:00 | update
```

Data Transfer

Iridium RUDICS

- Periodic data transfers (2 hours)
- Files are moved to the outgoing spool
- Small files are combined, large ones split (15KB chunks)
- Call retry if connection dropped
- Transfers are resumed at last 15KB chunk
- Bidirectional
- Over-the-air code updates

Server-Side Processing

File   Edit   View   History   Bookmarks   Tools   Help

← → C ↓ + ✕ | http://transport.sri.com/transport/servermonitor/newsgroups.py

transport.imnavaitcreek.history	0	0	0	9
transport.imnavaitcreek.ic_1991.mean	0	0	1	8
transport.imnavaitcreek.ic_1993.mean	0	0	0	0
transport.imnavaitcreek.powerstation1.camera	0	0	0	0
transport.imnavaitcreek.powerstation1.inbound	0	0	0	0
transport.imnavaitcreek.powerstation1.iridium	0	0	0	0
transport.imnavaitcreek.powerstation1.memory	0	0	0	0
transport.imnavaitcreek.powerstation1.sbc	0	0	0	0
transport.imnavaitcreek.powerstation1.system	0	0	0	0
transport.imnavaitcreek.powerstation1.tsbat3	0	0	0	0
transport.imnavaitcreek.powerstation1.updates	0	0	0	0
transport.imnavaitcreek.powerstation2.camera	0	0	0	0
transport.imnavaitcreek.powerstation2.inbound	0	0	0	0
transport.imnavaitcreek.powerstation2.iridium	0	0	12	123
transport.imnavaitcreek.powerstation2.sbc	0	0	12	123
transport.imnavaitcreek.powerstation2.system	0	0	12	123
transport.imnavaitcreek.powerstation2.tsbat3	0	0	12	123
transport.imnavaitcreek.sbc.camera	0	0	0	0
transport.imnavaitcreek.sbc.inbound	0	0	0	0
transport.imnavaitcreek.sbc.iridium	0	0	0	0
transport.imnavaitcreek.sbc.sbc	0	0	0	0
transport.imnavaitcreek.sbc.system	0	0	0	0
transport.imnavaitcreek.sbc.tsbat3	0	0	0	0
transport.imnavaitcreek.summary	0	0	44	460

Real-Time Web Interfaces
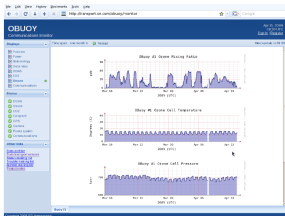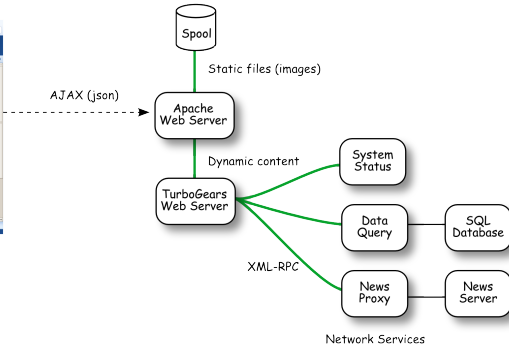
Planning For The Unexpected

Give Yourself Options

- RUDICS dial out
- PPP dial in
    - Request system keep modem on
    - Daily listen window
- SBD heartbeat
- Software watchdogs
    - Reboot based on monitored conditions
    - Example: no successful dialups
- Hardware watchdog
    - Nonresponsive - unconditional reboot

More Information

- Iridium and RUDICS

- Data Transport Network

- Real-time displays

- Copies of presentations

todd.valentic@sri.com
http://transport.sri.com
http://transport.sri.com/rudics