A new Telemetry System for the Automatic Geophysical Observatories using Iridium SBD Messaging

Bob Melville
New Jersey Institute of Technology
The AGOs

- 5 remote sites on the high plateau
- Run year-round
- Wind and solar power
- Iridium comms for data transfer (approx 20 MB/day)
- Need a separate telemetry channel
Existing Telemetry using ARGOS

- Argos ST-5 and ST-20 PTTs (transmitters) running since late 90’s
- Driven by a 8252 micro-processor
- 32-byte data packet
- Transmit at 401.65 MHz
An Alternative using Iridium SBD

- Designed (in something of a rush!) prior to 2010 field season
- Based on 9601 SBD modem
- Two units deployed in Dec 2010
- One failed in early March 2011, one is still running
Main design issues -- general

- Communicate with modem via a serial link
- Modem must be warmer than -30°C for reliable transmissions
- Modem operates in burst-mode; it requires approximately 30 seconds to connect to the network and send a packet
- Transmissions have error-checking and modem receives an ack/nak for each packet
- User receives an e-mail with data packet as an attachment
- Telemetry module runs off of station power when available; if station power drops, module coasts off of an internal rechargeable battery as long as it can
- SBD allows two-way communications; module provides two latching relays whose state can be controlled by a stateside command
Electronic design

- Charging circuit based on TI BQ24450
- Sealed lead-acid, 7Ah at 18 V
- Internal heater with dual set-points – keep it substantially warmer when station power is available
- ADuC843 uP provides eight 12-bit analog channels; use 4 of them for self-monitoring
- Provide additional measurement options for user: 4 analog channels, I2C and two digital channels
- Used bipolar op-amps (33272) for signal conditioning
- Isolate inputs – opto-couplers for digital input, buffers for analog signals
Idea for the TI BQ24450

- User must solve coupled non-linear equations to get resistor values
- Do this with a symbolic solver like Macsyma
- It works great and allows easy experimentation with different values

\[
\text{par}(x,y) := \frac{x \cdot y}{x + y};
\]

\[
\text{solve}([ \text{rc} = 46.4e3, \text{vth} = 26.25, \text{vfloat} = 34.50, \\
\text{vboost} = 36.75, \\
\text{vref} = 2.3, \text{vfloat} = \text{vref} \cdot (\text{ra} + \text{rb} + \text{rc}) / \text{rc}, \\
\text{vboost} = \text{vref} \cdot (\text{ra} + \text{rb} + \text{par}(\text{rc}, \text{rd})) / \text{par}(\text{rc}, \text{rd}), \\
\text{vth} = \text{vref} \cdot (\text{ra} + \text{rb} + \text{par}(\text{rc}, \text{rd})) / (\text{rb} + \text{par}(\text{rc}, \text{rd})), \\
[\text{ra}, \text{rb}, \text{rc}, \text{rd}, \text{vth}, \text{vfloat}, \text{vboost}, \text{vref}]);
\]

ra 354k
rb 17.3k
rc 46.4k
rd 632.4k
The ADuC843

- Uses 8051 instruction set
- 8-channel, 12-bit ADC
- Serial port
- Low-power sleep mode
- Easy to program via serial link
Programming issues

- Code is 2300 lines of assembly
- Does not use interrupt capability
- Assembly language was ideal for manipulating data obtained from various I2C sensors
- Re-use code from www.8052.com
- Randomize transmit times by masking off last three bits from the check-sum of the most recent data packet and adding to transmit interval
- Serial access to both modem and uP via a jumper cable – a simple but extremely effective debugging aid
- uP implements a simple monitor program for debugging via a serial terminal program
- Manufacturer should provide an emulator program!
Mechanical design

- Electronics in a die-cast box; very easy to work with
- Electronics insulated with pink-board inside of a surplus military transit case obtained from Hardigg
- Tested in a home-made cold chamber at -30 C
Sample transmission

- Programmed to send a packet every 6 hours
- Include communications statistics (very useful)
- Include status code with each channel

P4 Engineering Data File: P4_February_28_2011.eng
Time of Session (UTC): Mon Feb 28 14:19:02 2011

======== Analog Channels =========

<table>
<thead>
<tr>
<th>Chan</th>
<th>Hex Value</th>
<th>Engineering Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>7308</td>
<td>26.87  input voltage</td>
</tr>
<tr>
<td>01</td>
<td>0130</td>
<td>-1.00 turbine current</td>
</tr>
<tr>
<td>02</td>
<td>9f08</td>
<td>37.15PV voltage</td>
</tr>
<tr>
<td>03</td>
<td>0635</td>
<td>-1.00 PV current</td>
</tr>
<tr>
<td>04</td>
<td>6268</td>
<td>22.99 battery voltage</td>
</tr>
<tr>
<td>05</td>
<td>fb03</td>
<td>-0.25 battery current</td>
</tr>
<tr>
<td>06</td>
<td>feed</td>
<td>-1.00 vreg current</td>
</tr>
<tr>
<td>07</td>
<td>ea00</td>
<td>-22.00 rack temperature</td>
</tr>
<tr>
<td>08</td>
<td></td>
<td>Invalid</td>
</tr>
<tr>
<td>09</td>
<td>e1c0</td>
<td>-30.25 hut temperature</td>
</tr>
<tr>
<td>10</td>
<td>0d05</td>
<td>30.51 primary voltage</td>
</tr>
<tr>
<td>11</td>
<td>0d0e</td>
<td>22.45 internal battery voltage</td>
</tr>
<tr>
<td>12</td>
<td>066f</td>
<td>17.38 internal battery temperature</td>
</tr>
<tr>
<td>13</td>
<td>0671</td>
<td>17.48 modem temperature</td>
</tr>
<tr>
<td>14</td>
<td>039c</td>
<td>11.28 wind speed</td>
</tr>
<tr>
<td>15</td>
<td>0ff</td>
<td>365.00 wind direction</td>
</tr>
<tr>
<td>16</td>
<td>031f</td>
<td>617.07 barometric pressure</td>
</tr>
<tr>
<td>17</td>
<td>0bbd</td>
<td>3005.00 compact flash progress</td>
</tr>
</tbody>
</table>

==== End of Analog Channels ====

Digital Status A: f
Digital Status B: 0
Comm Attempts: 22
Modem Not Reg Attempts: 00
Poor Signal Reports: 00
Bad Return Code: 00
Successful Comms: 22
Sum of Signal over last 8 Xmissions: 40
Average of Last 8 Signal Strengths: 5.00
Good and bad

• 9601 can draw large transient currents – caused uP to glitch!
• Small puck antenna just as effective as a larger antenna
• Plastic shell connectors are entirely adequate for indoor use; but USB connectors for I2C did not work out
• I2C itself was a disaster – not sure what is wrong
Second generation

- Use 9602 instead of 9601
- Same uP, same code
- Simplify power system – station power and a Lithium primary pack
- Use only linear regulators – excess heat goes for warming electronics
- Electronics package in a Dewar flask
- All analog inputs differential
Acknowledgements

• Special thanks to Gary Ferentchak and Dan Wagster at RPSC for technical and administrative help with the modem
• Dr. Vladimir Papitashvili (NSF) authorized activation of the modems on short notice
• Circuit design help from Dr. Thomas Banwell, Telcordia, and Todd Valentic, SRI
• Pf. Noel Petit (Augsburg college) implemented the state-side data processing
• Andy Stillinger built the cold-chamber in his garage!