Smart Sensor Design and Networking Standards

Dr. Darold Wobschall
Associate Professor Emeritus (EE)
and
President, Esensors Inc.
Goals

- Review electronic sensors and signal conditioners
- Describe smart sensor design
- Discuss sensor networks, both wired and wireless
- Promote the IEEE 1451 sensor network standard
- Comment on wireless sensor future
Sensor Development

past and future

- Most sensor principles known (by physicists) for over 100 years
- Many sensors used industrially for over 60 years
- Computer controls and appetite for data have driven sensor uses, especially Machine-to-Machine (M2M).
- Continuing improvements in manufacturing methods (e.g. MEMS) have made sensors smaller & easier to use
- Advances in electronics (analog, a/d, microcomputers, communications) lower costs and add functionality.
- Smart, digital, networked sensors are the future trend
Sensor Classes

- Basic Sensors
- Smart Sensors
- Networked Sensors
Basic Sensor Block Diagram
Signal Conditioner Example
(Thermistor)

\[-v_T = \left( \frac{V_{\text{ref}}}{R_1} \right) R_T\]
Partial List of
Measured Parameters and Sensor Technologies

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acceleration/Vibration</td>
<td>Resistance</td>
</tr>
<tr>
<td>Level &amp; Leak</td>
<td>Capacitance</td>
</tr>
<tr>
<td>Acoustic/Ultrasound</td>
<td>Inductance &amp; magnetics</td>
</tr>
<tr>
<td>MachineVision</td>
<td>Optical &amp; fiber optic</td>
</tr>
<tr>
<td>Chemical/Gas</td>
<td>Voltage &amp; piezoelectric</td>
</tr>
<tr>
<td>Motion/Velocity/Displacement</td>
<td>Ultrasonic</td>
</tr>
<tr>
<td>Electric/Magnetic</td>
<td>RF/microwave</td>
</tr>
<tr>
<td>Position/Presence/Proximity</td>
<td>Sensors (and sensor industry)</td>
</tr>
<tr>
<td>Flow</td>
<td>are subdivided (fragmented) by:</td>
</tr>
<tr>
<td>Pressure</td>
<td>1. Parameter measured</td>
</tr>
<tr>
<td>Force/Strain/Load/Torque</td>
<td>2. Technology</td>
</tr>
<tr>
<td>Temperature</td>
<td>3. Application area</td>
</tr>
<tr>
<td>Humidity/Moisture</td>
<td></td>
</tr>
</tbody>
</table>

Networked Sensor Seminar
Some Sensor Elements

RTS (temperature)

Photodiodes

Pressure sensors

Force Sensor
**PC Signal Acquisition Plug-in Method**
(older, analog approach)

- A Star configuration: each sensor connected directly to DAQ.
- Results in many wires if multiple sensors
- Calibration done in software for each sensor
- Noise often causes errors
Smart Sensor Block Diagram
Smart (Digital) Sensor Features

- Analog/Digital Converter
  Typically 10-14 bits, usually internal

- Microcontroller (embedded)
  PIC or similar 8-bit micro with appropriate features

- Sensor Identification (serial # etc)

- Calibration information
  Compensates for sensor variations; conversion to engineering units

- Data logging and real-time clock (optional)
Microcontroller Example

Figure 1-1: PIC16F872 Block Diagram

Pin Diagram

DIP, SOIC, SSOP

<table>
<thead>
<tr>
<th>Pin</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MCLR/VPP</td>
</tr>
<tr>
<td>2</td>
<td>RA0/AN0</td>
</tr>
<tr>
<td>3</td>
<td>RA1/AN1</td>
</tr>
<tr>
<td>4</td>
<td>RA2/AN2/VREF+</td>
</tr>
<tr>
<td>5</td>
<td>RA3/AN3/VREF-</td>
</tr>
<tr>
<td>6</td>
<td>RAM10/CKI</td>
</tr>
<tr>
<td>7</td>
<td>RAM1/AN/SS</td>
</tr>
<tr>
<td>8</td>
<td>Vss</td>
</tr>
<tr>
<td>9</td>
<td>OSC1/CLKIN</td>
</tr>
<tr>
<td>10</td>
<td>OSC2/CLKOUT</td>
</tr>
<tr>
<td>11</td>
<td>RC0/T1S0/T1CKI</td>
</tr>
<tr>
<td>12</td>
<td>RC1/T1S1/CCP2</td>
</tr>
<tr>
<td>13</td>
<td>RC2/CCP1</td>
</tr>
<tr>
<td>14</td>
<td>RC3/SQ/SCL</td>
</tr>
<tr>
<td>15</td>
<td>RC4/SDI/SDA</td>
</tr>
<tr>
<td>16</td>
<td>VSS</td>
</tr>
<tr>
<td>17</td>
<td>RC5/SDO</td>
</tr>
<tr>
<td>18</td>
<td>RC7/RXD/T1DT</td>
</tr>
<tr>
<td>19</td>
<td>RC6/T1CK</td>
</tr>
<tr>
<td>20</td>
<td>VDD</td>
</tr>
<tr>
<td>21</td>
<td>RB0/INT</td>
</tr>
<tr>
<td>22</td>
<td>RB1</td>
</tr>
<tr>
<td>23</td>
<td>RB2</td>
</tr>
<tr>
<td>24</td>
<td>RB3/PGM</td>
</tr>
<tr>
<td>25</td>
<td>RB4</td>
</tr>
<tr>
<td>26</td>
<td>RB5</td>
</tr>
<tr>
<td>27</td>
<td>RB6/PGC</td>
</tr>
<tr>
<td>28</td>
<td>RB7/PGD</td>
</tr>
</tbody>
</table>

Flash Program Memory

8-Level Stack (13 dB)

RAM File Registers

14 Program Bus

Instruction Memory

RAM Address 1

Data Bus

Addressing Mode

Port A

Port B

Port C

Port D

Port E

Port F

Timer0

Timer1

Timer2

Data EEPROM

CCP

Serial Port

9-bit A/D
Connection of Non-networked Smart Sensors to Computers

- USB (best for PCs) or RS232 (best for Instruments) Serial Data Lines
- One line and port per sensor (a problem with large systems)
- Data protocol not standardized
Websensor – sensor with Internet Address

- Uses Ethernet as the Network
- Microcontroller has TCP/IP (mini-website) as protocol
- Data can be read anywhere on Internet
- Polling/logging/display by NAGIOS (Lynix) freeware
- A smart sensor but does not have standard interface
Websensor data readout
(typical of digital sensor displays)

<table>
<thead>
<tr>
<th>Networked Sensor Seminar</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Humidity</strong></td>
<td><strong>OK</strong></td>
<td>08-30-2004 11:16:04</td>
<td>00 00 16m 37s</td>
<td>1/3</td>
<td>OK temp: 87.7 F, humidity: 44.5%, illumination: 275.3</td>
<td></td>
</tr>
<tr>
<td><strong>Temperature</strong></td>
<td><strong>OK</strong></td>
<td>08-30-2004 11:16:04</td>
<td>00 00 16m 38s</td>
<td>1/3</td>
<td>OK temp: 87.7 F, humidity: 45.1%, illumination: 275.3</td>
<td></td>
</tr>
<tr>
<td><strong>Humidity</strong></td>
<td><strong>Critical</strong></td>
<td>08-30-2004 11:16:04</td>
<td>00 00 16m 37s</td>
<td>3/3</td>
<td>Critical temp: 87.7 F, humidity: 43.7%, illumination: 275.3</td>
<td></td>
</tr>
<tr>
<td><strong>Temperature</strong></td>
<td><strong>Critical</strong></td>
<td>08-30-2004 11:16:04</td>
<td>00 00 16m 38s</td>
<td>3/3</td>
<td>Critical temp: 87.7 F, humidity: 45.1%, illumination: 275.3</td>
<td></td>
</tr>
<tr>
<td><strong>Humidity</strong></td>
<td><strong>Critical</strong></td>
<td>08-30-2004 11:16:04</td>
<td>00 00 16m 37s</td>
<td>3/3</td>
<td>Critical temp: 87.7 F, humidity: 43.7%, illumination: 275.3</td>
<td></td>
</tr>
<tr>
<td><strong>Temperature</strong></td>
<td><strong>Critical</strong></td>
<td>08-30-2004 11:16:04</td>
<td>00 00 16m 38s</td>
<td>3/3</td>
<td>Critical temp: 87.7 F, humidity: 45.1%, illumination: 275.3</td>
<td></td>
</tr>
</tbody>
</table>

3 Sensors Matching Service Entries Displayed
Sensor/Transducer Networks

- A network connects more than one addressed sensor (or actuator) to a digital wired or wireless network.
- Both network and sensor digital data protocols are needed.
- Standard data networks can be used but are far from optimum.
- Numerous (>100) incompatible sensor networks are currently in use – each speaking a different language.

The Tower of Babel
Networked Sensor Block Diagram
IEEE 1451 – the Universal Transducer Language

- Too many network protocols in common use
- Narrow solutions and borrowed protocols have not worked
- Sensor engineers in the fragmented sensor industry need a simple method of implementation
- How can it be done?
- We need something like USB, except for sensors
- The IEEE 1451 Smart Transducer Protocol standard is the best universal solution
- Supported by NIST, IEEE and many Federal agencies
A review of the
IEEE 1451 Smart Transducer Concept
IEEE 1451 Advantages

- Comprehensive enough to cover nearly all sensors and actuators in use today (not 20/80% approach)
- Many operating modes
  (buffered, no-buffer, grouped sensors, timestamps, timed data, streaming …)
- Extensive units, linearization and calibration options
- Multiple timing and data block size constraints handled.
- Compatible with most wired and wireless sensor buses and networks (point-to-point, mesh, TIM-to-TIM, mixed networks).
- Efficient binary protocol (especially suitable for wireless)
- Standard is 400+ pages for basic part, over 1500 page total
But the Complexity!

- A comprehensive standard is necessarily complex
- There was little adoption of the original IEEE 1451.2 (TII) standard because of its perceived complexity
- Manual preparation of the TEDS is not practical
- A TEDS compiler is needed
- A compliance test procedure is also desirable to prove that the design is correct

Munch –The scream

Networked Sensor Seminar
IEEE 1451.0 (Dot 0) TEDS Format

- Required TEDS  [Memory block with defined format]
  - MetaTEDS
  - Channel TEDS
  - Calibration TEDS (unless SI units)
  - Xdr-name TEDS
  - Phy TEDS
  - Also optional TEDS

- Data Transmission  [specific octet format]
  - TEDS/Status requests
  - Triggering and configuration
  - Sensor read commands and data return
  - Actuator write commands and data sending
TEDS Format

- General format for each TEDS section:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Data Type</th>
<th>Number of Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>--------</td>
<td>----------------------------</td>
<td>-----------</td>
<td>-----------------</td>
</tr>
<tr>
<td>TEDS Length</td>
<td></td>
<td>UInt32</td>
<td>4 bytes</td>
</tr>
<tr>
<td>1 to N</td>
<td>Data Block</td>
<td>Variable</td>
<td>Variable</td>
</tr>
<tr>
<td>--------</td>
<td>Checksum</td>
<td>UInt16</td>
<td>2 bytes</td>
</tr>
</tbody>
</table>

- Binary TEDS Tuple format for each data block: Type-Length-value (TLV)
  
  Example: 01 02 A3 04
  
  Field type is 1, Length is 2 bytes, field value is “A304” hex

- Field example: Meta-TEDS (TEDS # 1)
  
  13: Number of Implemented Transducer Channels (default=1)
TEDS Compiler

- Part of Ph. D. thesis of Wai Liu (Univ. at Buffalo)
TEDS Sections Implemented

- Meta TEDS
- Meta ID TEDS
- Transducer Channel TEDS
- Transducer Channel ID TEDS
- Calibration TEDS
- Calibration ID TEDS
- XdrcName TEDS

Referenced by TEDS section/access code (e.g. #1 for Meta-TEDS)
### SI Based Units

<table>
<thead>
<tr>
<th>Base Quantity</th>
<th>Name</th>
<th>Unit Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>meter</td>
<td>m</td>
</tr>
<tr>
<td>Mass</td>
<td>kilogram</td>
<td>kg</td>
</tr>
<tr>
<td>Time</td>
<td>second</td>
<td>s</td>
</tr>
<tr>
<td>Electric current</td>
<td>ampere</td>
<td>A</td>
</tr>
<tr>
<td>Thermodynamic temperature</td>
<td>Kelvin</td>
<td>K</td>
</tr>
<tr>
<td>Amount of substance</td>
<td>mole</td>
<td>mol</td>
</tr>
<tr>
<td>Luminous intensity</td>
<td>candela</td>
<td>cd</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Data Type</th>
<th>Number of octets</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Physical units interpretation</td>
<td>UInt8</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>(2 \times &lt;\text{exponent of radians}&gt; + 128)</td>
<td>UInt8</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>(2 \times &lt;\text{exponent of steradians}&gt; + 128)</td>
<td>UInt8</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>(2 \times &lt;\text{exponent of meters}&gt; + 128)</td>
<td>UInt8</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>(2 \times &lt;\text{exponent of kilograms}&gt; + 128)</td>
<td>UInt8</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>(2 \times &lt;\text{exponent of seconds}&gt; + 128)</td>
<td>UInt8</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>(2 \times &lt;\text{exponent of amperes}&gt; + 128)</td>
<td>UInt8</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>(2 \times &lt;\text{exponent of kelvins}&gt; + 128)</td>
<td>UInt8</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>(2 \times &lt;\text{exponent of moles}&gt; + 128)</td>
<td>UInt8</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>(2 \times &lt;\text{exponent of candelas}&gt; + 128)</td>
<td>UInt8</td>
<td>1</td>
</tr>
</tbody>
</table>
Dot 0 Command/Response Structure

<table>
<thead>
<tr>
<th>Byte Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Destination Transducer Channel Number (Most significant byte)</td>
</tr>
<tr>
<td>2</td>
<td>Destination Transducer Channel Number (Least significant byte)</td>
</tr>
<tr>
<td>3</td>
<td>Command Class</td>
</tr>
<tr>
<td>4</td>
<td>Command Function</td>
</tr>
<tr>
<td>5</td>
<td>Length (Most significant byte)</td>
</tr>
<tr>
<td>6</td>
<td>Length (Least significant byte)</td>
</tr>
<tr>
<td>7-N</td>
<td>Command dependent bytes</td>
</tr>
</tbody>
</table>

NCAP Command Message Structure

<table>
<thead>
<tr>
<th>Byte Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Success/Fail Flag</td>
</tr>
<tr>
<td>2</td>
<td>Length (Most significant byte)</td>
</tr>
<tr>
<td>3</td>
<td>Length (Least significant byte)</td>
</tr>
<tr>
<td>4-N</td>
<td>Reply dependent bytes</td>
</tr>
</tbody>
</table>

TIM Reply Message Structure
Meta-TEDS Writer Screen
Channel/Calibration TEDS
(for linear sensors)
TEDS Reader
IEEE 451 TIM Compliance Tester

- TIM (Transducer Interface Module) is most complex and done by sensor design engineers (TIM tester can be used by the few NCAP designers)
- Tester verifies compliance of a TIM to IEEE 1451.0 (Dot 0) protocol
- Focus is on TEDS checking and data transfer format
- Physical device compliance not checked (part of other standards, e.g. RS485, Bluetooth)
- Tester uses serial bus (RS232)
- Testing may be done by Internet
Standard TIM/NCAP Configurations

The following 3 slides describe TIM and NCAP configurations for which the TIM tester can be used
Network side (NCAP) options (wired)

- Internet/Ethernet
- PC Readout
- Industrial network

All use Dot 0 protocol
Data Readout Examples
(via Internet)

- Sensor data converted to ASCII for display
- TEDS data is displayed in hexadecimal form
Prototype TIM and NCAP

- NCAP interfaces to Internet via Ethernet
TIM Tester – Data retrieval
Serial Bus Format and Relation to other Networks

- Tester uses RS232 serial bus only but…
- Interfaces to other physical devices (USB, RS485, Bluetooth, Zigbee, ….) available.
- TEDS retrieval is one feature
- Sensor data read (protocol check) for each channel:
  - *Idle mode* – full scale value of sensor reading
    (Checked against TEDS, error flag is not correct)
  - *Operating mode* – actual sensor reading
    (Must be within sensor range)
Example – Wireless Connection

- Wireless modules with RS232 I/O when connected to Dot 2 TIMS are similar to IEEE 1451.5 TIMs (wireless version of IEEE 1451).
- Data format and TEDS are the same (both follow the Dot 0 standard).

Dot 5 TIM built from a Dot 2 TIM and wireless transceiver
IEEE 1451.4 (only) does not use the Dot 0 format TEDS. This is a small, TEDS-only version (no digital data format is specified by the standard).
Transducer Electronic Data Sheet
(Dot 4 TEDS)

- UUID (Universal Unique Identifier)
  Supplied by EEPROM (DS2433) manufacturer (6 bytes)
- Basic TEDS (8 bytes)
  - Model Number (15 bits)
  - Version Letter (5 bits, A-Z)
  - Version Number (6 bits)
  - Manufacturer ID (14 bits)
  - Serial Number (6 bits)
- Manufacturer’s TEDS
  Sensor type and calibration parameters (16 bytes)

Conversion to Dot 0 TEDS possible (but not unique)
Dot 4 TEDS Writer and Reader
(PC Screens)
Harmonization of IEEE 1451 with other sensor standards
Wireless Sensor Networks

- Currently fashionable in academic and VC circles
- Recently available low-cost, high performance RF transceiver chips greatly simplify design
- Improvements in embedded microcomputers and smart sensor design also make wireless sensors much more practical now
- Can be lower cost and easier to use than wired sensors
- Point-to-point and mesh protocols available.
- Specialized sensor networks (e.g. Zigbee) much more efficient for wireless sensors, especially battery operated.
- Lack of standards inhibiting growth of industry.
Wireless Sensor Example

- WS01 wireless temperature sensor
- 433 MHz FSK – range 10 to 100 meters
- 3-volt coin battery

Printed Circuit Board (back)  Photo (front)
RFID with Sensors

- Combines Radio Frequency IDentification with sensors
- Uses standard (ISO/IEC) RFID communication/format with additional sensor memory section
- Sensor format is based in IEEE 1451 protocol
- Typical application is tracking and monitoring perishable shipments (temperature, shock/vibration)
Summary

- Electronic sensors and signal conditioners were reviewed
- Smart sensor design examples given
- Wired and wireless sensor networks were discussed
- The importance of sensor network standards (IEEE 1451) were stressed
- Comment made on networked sensor futures

Contact: designer@eesensors.com