

*Esensors is furnishing this document "as is", without cost, and does not provide any warranty of the item whatsoever, whether express, implied, or statutory, including, but not limited to, any warranty of merchantability or fitness for a particular purpose or any warranty that the contents of the item will be error-free.*

## Preliminary Specifications – Version 1.15

# IEEE 1451 Development Kit Description

Yuan Ma, Avarachan Cherian and Darold Wobschall

Esensors Inc., Amherst NY

May 16, 2013

## Background

The development boards with IEEE 1451 smart transducer standard protocol were intended to provide reference designs for engineers developing TIMs and NCAPs compliant with the IEEE 1451 protocol. The circuits and software are open source and are available for download from our website (and elsewhere). This memo is concerned with the hardware and assumes familiarity with the IEEE 1451 standard and terms. We employ the shorthand terms of “Dot x” to indicate IEEE 1451.x device. A brief review of the IEEE 1451 standard is given in Appendix A and a list of references in Appendix B.

## Overall Description

The system (Fig. 1) consists of an NCAP (gateway) with several ports (wired and wireless) and several different types of TIMs, all compliant with the IEEE 1451 standard (specifically Dot 0, Dot 2, Dot 4 and Dot 5).

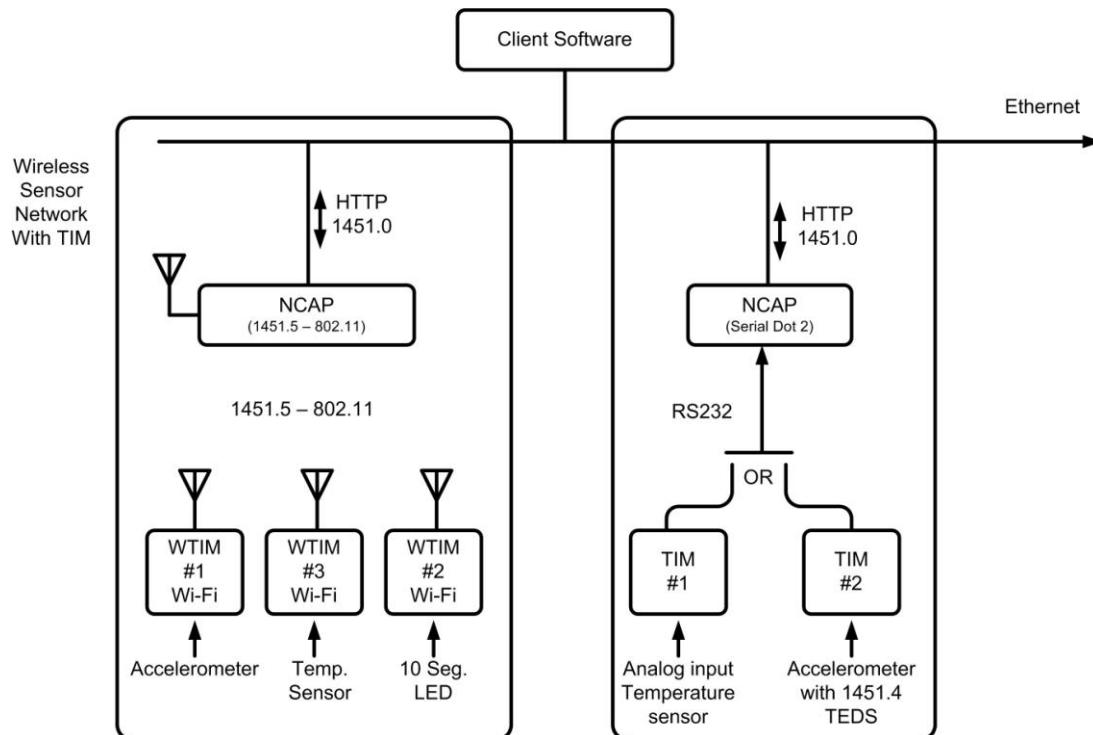
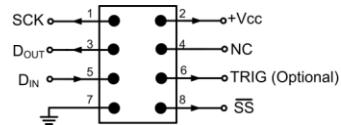


Fig. 1 System Block Diagram

## NCAP

The NCAP is multi-port or multi-functional. The ports are:

- RS232 (Dot 2)
- USB (in future)
- Wi-Fi (Dot 5)
- One 8-pin connector with SPI interface
  - See Dot 2 standard for description



However, only one port is active at a time, in part to simplify the software, which is a goal for the reference design. Note that Dot5 is a local Ad hoc Wi-Fi network, not connected directly to the Internet except through the NCAP. The Internet protocol is the IEEE 145.0 (Dot 0) – HTTP is provided in the NCAP.

The NCAP is available in a case for purchase from Esensors.

## TIM

The types of TIMs provided in this kit are:

- Dot 2 -- RS232
  - With various I/Os and sensor including Dot 4 sensor
- Wi-Fi (Dot 5)
  - With various I/Os and sensors (3 types) including Dot 4 sensor

## **Detailed Description**

### Multiport NCAP Hardware

The NCAP consists of two boards each with a microcontroller (Fig.2). The main board handles the local network or ports while the second handles the Ethernet/Internet interface.

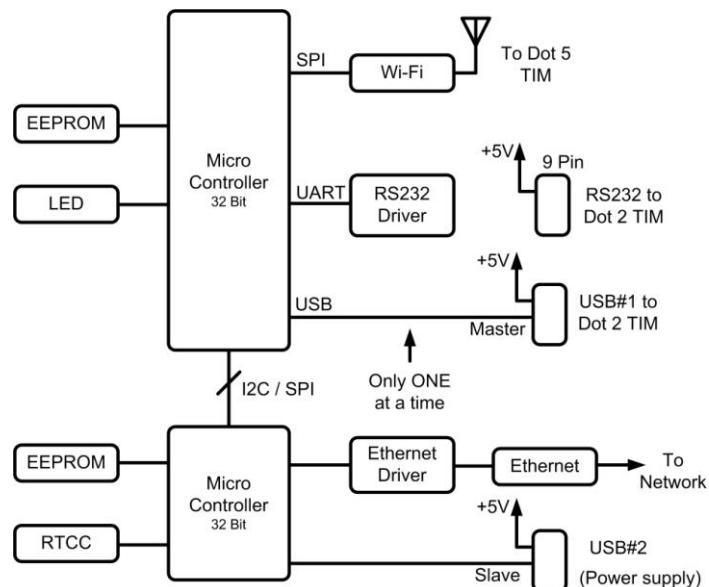


Fig. 2 NCAP Block Diagram

The two boards are stacked one over the other and connected via SPI bus. The connection to the Internet is via Ethernet (not Wi-Fi). Power is provided via 9-12v DC adapter. A photo of the boards is shown in Fig. 3. The circuit is described further in Appendix C.



Fig. 3 NCAP Photo – Ethernet board is on bottom

The Wi-Fi is a local network, as defined by the Dot 5 standard and this communicates with the wireless TIMs (star configuration). The TIMs do not communicate directly with the Internet nor with each other.

The RS232 connector (J1) pinout follows the Dot 2 standard and has +5v power on one pin. The NCAP is connected to the TIM by the RS232 cable.

### NCAP Software

It consists of (a) Main Dot 0 section, (b) three alternative PHY sections (Dot 2, Dot 4, Dot 5), (c) Ethernet (TCP/IP) interface.

### TIM boards

The three types of TIMs (Dot 2, Dot 4, Dot 5) are combined into one pair of boards (Fig. 4). However only the components appropriate to the specific type of TIM are mounted and TIM can communicate with NCAP either via RS-232 or Wi-Fi. The upper board (main) has the sensors and I/O connectors. The lower board (communication) provides the local network or bus communication, along with the TEDS.

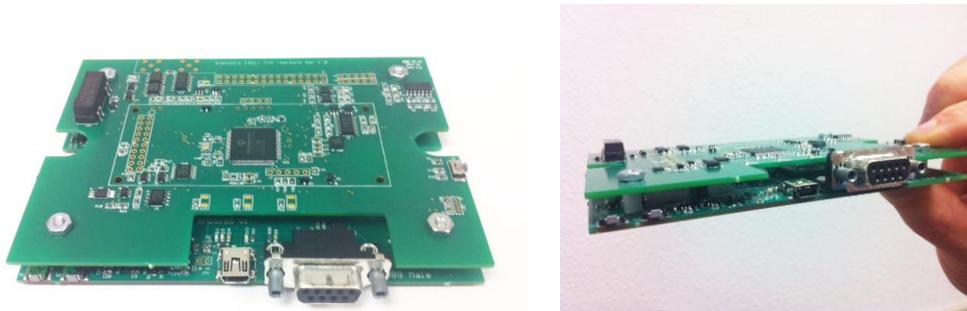


Fig. 4 Photo of Combination TIM boards

These TIMs are relatively complex because they are intended to illustrate multiple types of sensors and features.

## Serial (RS232) Dot 2 TIM

The Dot 2 (serial) TIM has one RS232 port which connects to the NCAP via a cable. It has a 9-pin type D connector with a +5v supply on one pin (see Dot 2 standard). See Fig. 5.

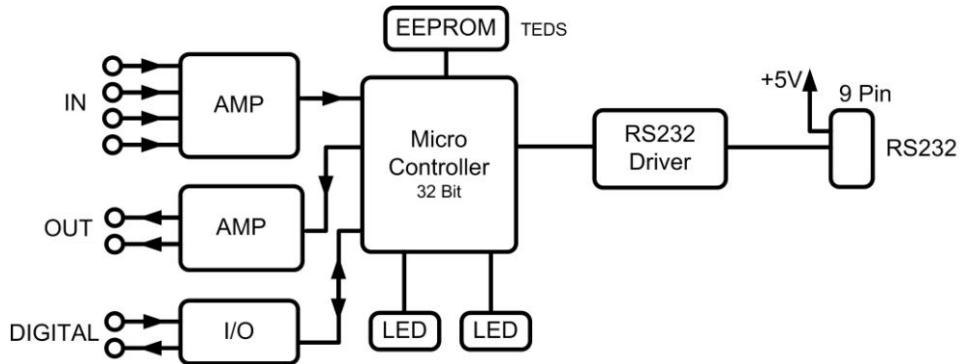


Fig. 5 Block Diagram of Dot 2 TIM

It has these I/O lines on a 30-pin female connector (dual row 15x2, 0.1" centers).

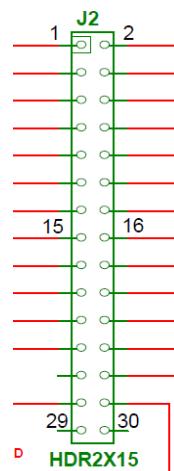
- Four analog inputs, -5 to +5v (switch for 0 to +2.5v)  
With 10-bit a/d
- Two analog outputs (0 to +5v)
- Two 8-bit digital ports
  - One 8-bit input and one 8-bit output

There are 4 LED indicators also (not part of I/O).

The pinouts for the connector J2 are shown in Table 1.

Table 1. Connector J2 Pinouts (Dot 2 and Dot 5)

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	Analog In #2	9	Digital In - 3	17	Digital Out - 3	25	nc
2	Analog In #1	10	Digital In - 2	18	Digital Out - 2	26	Relay NO
3	Analog In #4	11	Digital In - 5	19	Digital Out - 5	27	Ground
4	Analog In #3	12	Digital In - 4	20	Digital Out - 4	28	+5v out
5	Analog Out#2	13	Digital In - 7	21	Digital Out - 7	29	nc
6	Analog Out#1	14	Digital In - 6	22	Digital Out - 6	30	nc
7	Digital In - 1	15	Digital Out - 1	23	Relay Com		
8	Digital In - 0	16	Digital Out - 0	24	Relay NC		



In addition these transducers will be present:

- Temperature sensor (SHT11) with humidity option
- Actuators
  - LCD (16x2 line)
  - Relay (SPDT, dry contact)

The channel numbers assigned to the different inputs and outputs are shown in Table 2.

Table 2.

Dot 2 TIM Data Channel Assignments				
Channel	Signal name	In/Out	Data Type	Note
1	ANx1	Analog Input #1	FP (32)	-5.0V to 5.0 V
2	ANx2	Analog Input #2	FP (32)	-5.0V to 5.0 V
3	Temperature	Input	FP (32)	-40 to 123.8 C, 0.1C
4	Humidity	Input	FP (32)	0 to 100%, 0.5%
5	DI_0	Digital Input #0	UINT8	1: high, 0: low
6	DO_0	Digital Output #0	UIN8	1: high, 0: low
7	Relay	Digital Output	UIN8	1: close, 0: open
8	D2A_1	Analog Output	FP (32)	0V to 5.0 V
9	LCD	Output	Character	Array for 32 characters
10	J7, D1	Dot4 Analog In	FP (32)	
11	Dot4 TEDS	Input	UINT8	32 octets, in Hex format

The TIM is powered (+ 5v) through the 9-pin (RS232) connector.

#### Dot 4 TIM

This is a non-standard TIM which has added features so that it appears compatible with the Dot 0 standard from the Internet (network) viewpoint and with the Dot 4 from the sensor viewpoint. Since the Dot 4 standard is TEDS-only the specification of the a/d needed in addition to generate the digital data for the Internet is added arbitrarily. We use the Dot 0 standard for this function. In effect, this is partly a Dot 0 TIM with the a/d, and partly a sensor with Dot 4 TEDS (Fig. 6). The NCAP treats both Dot4 TEDS and analog output as Dot 0 sensor channels, this is not part of IEEE1451 standard, but NCAP can talk to any Dot4 sensor in this way.

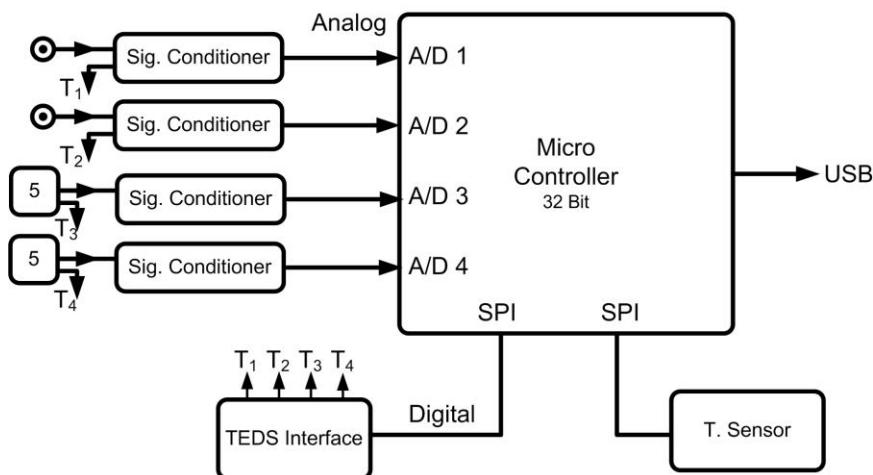


Fig. 6 Block Diagram of Dot 4 TIM

A Dot 4 type sensor is an analog output sensor with a small EEPROM on a 1-wire bus as the TEDS. There are two types of interfaces, one with separate lines for the TEDS and data and another which has the analog and digital on the same wire (2 wires including ground). For the combined analog/digital interface, the analog signal is modulated on a current source for positive voltages (IEPE) and the digital (TEDS) is connected for negative voltages (see Appendix F). This TIM has these ports:

- Class 1 with IEPE (two of these)  
Connector is HD-BNC
- Class 2 with 4-20 mA signal input  
Connector (J5) is 5-pin header with 0.1" spacing
- Class 2 with 0 to 5v signal input  
Connector (J6) is 5-pin header with 0.1" spacing

Table 3 Dot 4 Connector Pinouts

Pin	Signal #1 (J7)	Pin	Signal #2 (J8)	Pin	Signal #3 (J5)	Pin	Signal #4 (J6)
1	IEPE (coax)	1	IEPE (coax)	1	TEDS	1	TEDS
				2	+Vs	2	+Vss
				3	LP+	3	In+
				4	LP-	4	In-
				5	Gnd	5	Gnd

The TIM will have a temperature and humidity sensor (SHT11) also. The channel assignments are given in Table 2.

### Wireless TIM (WTIM – WiFi, Dot 5)

These are WiFi devices (2.4 GHz) which communicate with the NCAP (and from there to the Internet or other WTIMs) and employ the IEEE 1451.5 protocol. See Fig. 7.

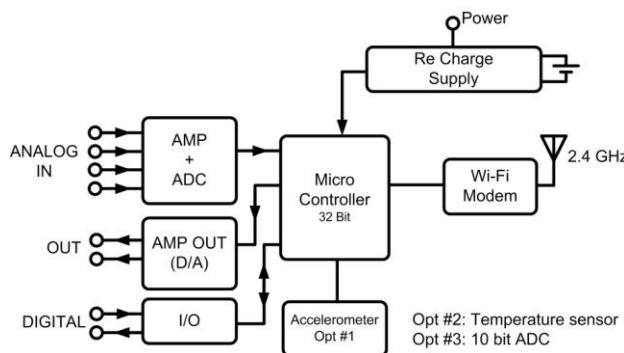


Fig. 7 Block Diagram of Dot 5 TIM

These WTIMs can be battery operated (rechargeable battery) with line operated option (via 5v supply from USB connector or RS-232 connector which also acts as the charger).

The I/O lines are the same for the Serial (Dot 2) TIM (Table 1), except that there is no relay (to conserve battery power), specifically:

- Four analog inputs, -5 to +5v  
With 10-bit a/d
- Two analog outputs (0 to +5v)  
Two 8-bit digital ports (selectable configuration via switch)  
One 8-bit input and one 8-bit output
- Temperature/Humidity sensor
- LCD (16x2 line)

The channel assignments are given in Table 2. Please note Dot4 sensor can be connected to J8 connector on any TIM board(Dot 2 or Dot 5).

## TEDS

The Transducer Electronic Data Sheet (TEDS) is stored in the TIM, here as read-only files (mostly binary). The following TEDS are present:

- META-TEDS
- TransducerChannel TEDS (several)
- User's Transducer Name TEDS (ASCII)
- PHY TEDS

The data in the TEDS is stored with the least significant byte at the highest address (big Endian) following the Dot 0 specification. For example decimal 1000 (hexadecimal 03 E8, two bytes) is stored as 03 E8 which is the human readable format (English). Many computers reverse the order (little endian) and it is the responsibility of the TIM designer to assure that the TEDS data, especially binary, is transmitted in the proper order.

As a development tool, this system provides the way to query, read and write/add raw TEDS via web interface.

- The response of query TEDS is to provide information like TEDS attributes, status, current size, checksum and maximum size.
- The reply of read TEDS is the raw TEDS stored in TIM's non volatile memory output in Hex format.
- User can change or add raw TEDS (in Hex format). TIM device will do sanity checking with the input raw TEDS before storing it to non volatile memory, which includes general format checking (length, checksum), size and if the TEDS type in the raw TEDS match with the input TEDS type. If the specified raw TEDS already exist, according to input Channel Id and TEDS type, TIM will update the old TEDS with the new one after sanity checking, otherwise, it will create a new TEDS.

## Appendix A. Brief Description of IEEE 1451 Smart Transducer Standard

Jay Nemeth-Johannes, Smart Sensor System

Edited by D. Wobschall, Esensors Inc.

### Concept of IEEE 1451 Standard

The IEEE-1451 standard defines hardware and software standards for smart transducers. A 1451 based system contains a pair of components. The Transducer Interface Module (TIM) contains the sensor, signal conditioning, analog to digital conversion, and control logic necessary to acquire a signal and convert it to data. The Network Capable Application Processor (NCAP) collects data from one or more TIMs, applies the correction, calibration and potentially other processing algorithms and passes the data to an application via a LAN connection (Fig. A1).

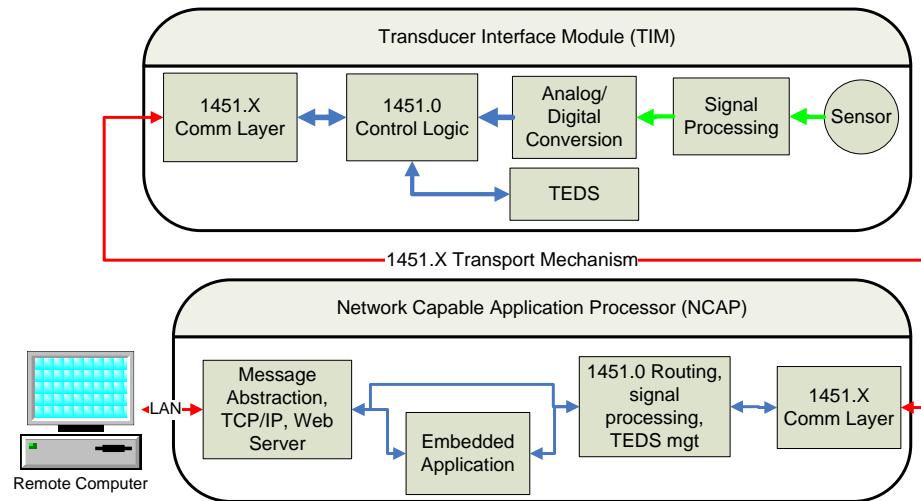


Fig. A1 IEEE-1451 Block Diagram

The TIM also holds a Transducer Electronic Data Sheet (TEDS) that contains the calibration and operating data necessary to create a calibrated result in standard SI units. The TEDS contains additional information necessary to uniquely identify the TIM as well as providing supplemental information to the application. Information is passed between the NCAP and TIM via a common hardware and software interface. The software interface is standardized for the 1451 system, thus simplifying the task of adding new transport interfaces. The hardware interfaces have been defined in the various 1451 documents to including open standard interfaces as RS-232, USB (proposed), CAN (proposed), 802.11, Bluetooth and ZigBee (802.15.4)

The system is built around the NCAP, which manages the TIMs and processes data to be used by the application. When an NCAP is initialized, it searches its interfaces for TIMs and claims the ones it finds. It then transfers a copy of each TIM's TEDS database to a cache area within

the NCAP. When a TIM is asked for a reading, it will acquire the data and generally return it in fundamental A/D counts. The NCAP will then apply the correction data found in the TEDS and convert it to calibrated SI data. The data is then transferred over the external network using http protocol and XML. The reason for this method of transferring from the TIM and processing in the NCAP allows the calibration data to be associated with the sensor itself, but offloads the complex processing to the NCAP. This allows the TIM to be implemented with small microprocessors.

An alternate reference model is shown in Fig. A2.

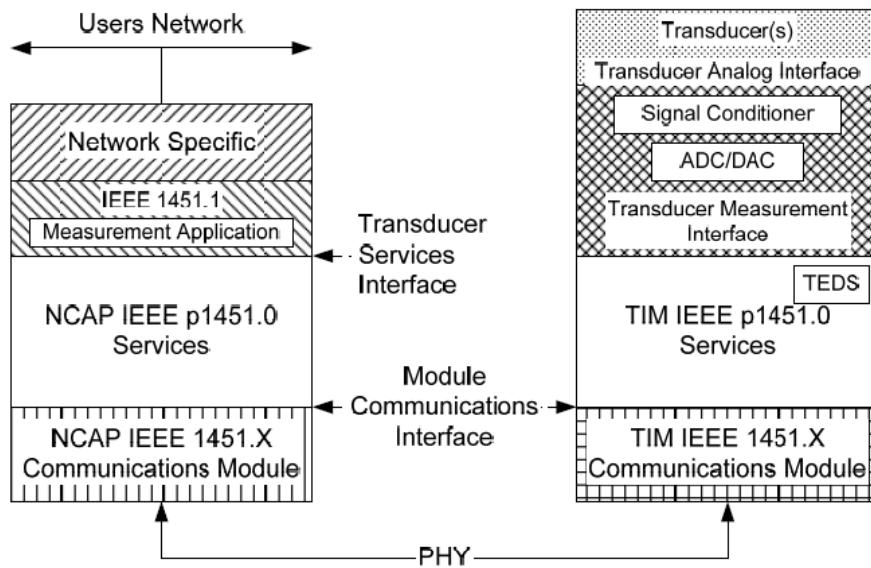


Fig. A2 IEEE 1451 Reference Model showing relation of Dot 0 and PHY sections.

The benefits of this technology are myriad and varied, depending on the configuration chosen. The first, and most apparent, benefit is the open nature of this standard that allows different manufacturers to produce products with the assurance that they will self configure and seamlessly operate with products from other manufacturers (plug and play feature). All 1451 systems benefit from the use of common codes and formats. This means that TIMs can be configured into systems and be recognized and configured by NCAPs without previous knowledge of that TIM. Sensors and actuators can easily be added to or removed from a system. If a wireless sensor is used, no action needs to be taken other than placing the sensor and ensuring that an appropriate wireless NCAP is within range. This capability allows sensors to be placed during testing phases or when a specific testing need is discovered and then easily removed during normal deployment.

A 1451 based system also returns sensor data in calibrated, SI units. This places the correction routines in the sensor where they belong and also reduces the chances for errors to be introduced by unit mismatch. In addition, all operating parameters are queryable, thus allowing the application to certify units, calibration and operating data.

The quality of the measurement can be improved because analog to digital conversion occurs near the signal being measured. This, in turn, eliminates long runs of analog wiring that can introduce noise and crosstalk, as well as eliminating large bundles of wiring. In addition to the noise problem, wire bundles are heavy, prone to failure, and difficult to diagnose and repair.

The status of the different parts of the standard are as follows:

- Dot 0 -- General Format --- approved 2007
- Dot 1 -- NCAP Protocol (optional) – approved 1998, being revised.
- Dot 2 -- Serial, point-to-point – revision done, final approval pending
- Dot 3 -- Multi-drop & timestamp -- approved 2003, but obsolete (no hardware)
- Dot 4 -- Analog & TEDS (approved 2004, TEDS only)
- Dot 5 -- Wireless (Bluetooth, WiFi, maybe Zigbee) -- approved 2007
- Dot 7 --- RFID with sensors -- approved 2010

The basic functions of Dot 0 are to handle the following commands:

- TEDS/Status requests
- Triggering and configuration
- Sensor read commands and data return
- Actuator write commands and data sending

## TEDS

The transducer electronic data sheets required are:

- MetaTEDS
- Channel TEDS
- Calibration TEDS (unless SI units)
- Xdr-name TEDS
- Phy TEDS

These are in specific, compact binary formats. There are many additional, optional TEDS.

## Appendix B. References

1. IEEE 1451 information <<http://ieee1451.nist.gov/>> and grouper.ieee.org/.../Review%20on%20the%20Proposed%20Survey%20of%20Wireless%20Sensor%20Requiremen.ppt
2. Purchase standard (Dot 0 and others):  
[http://webstore.ansi.org/RecordDetail.aspx?sku=IEEE%201451.0-2007&source=google&adgroup=ieee&qclid=CPu547uK\\_rECFQEUKgodCRgA-A](http://webstore.ansi.org/RecordDetail.aspx?sku=IEEE%201451.0-2007&source=google&adgroup=ieee&qclid=CPu547uK_rECFQEUKgodCRgA-A)
3. Dot 4 tutorial: <http://standards.ieee.org/develop/regauth/tut/teds.pdf>
4. IEPE Interface: <http://www.mmf.de/manual/m32mane.pdf>
5. Sensor network lists: Reference: <http://www.smarthomeforum.com/start.shtml>,
6. K. Lee and M. Reichardt “Open Standards for Homeland Security Sensor Networks”, IEEE Instrumentation and Measurement Magazine, Dec. 2005
7. R. Frank “Understanding Smart Sensors,” 2<sup>nd</sup> edition, Artech House (2000)
8. R. Pallas-Areny and J. Webster “Sensors and Signal Conditioning”, John Wiley 2<sup>nd</sup> ed ,2001.
9. S. Y. Yurish, “IEEE 1451 Standard and Frequency Output Sensors: How to Obtain a Broad-Based Industry Adoption?” *Sensors & Transducers Magazine (S&T e-Digest)*, Vol.59, Sept 2005
10. The ZigBee Alliance <http://www.zigbee.org/>
11. D. Wobschall, “Network Sensor Monitoring using the Universal IEEE 1451 Standard”, IEEE I and M, Vol. 11, p 18 (2008)
12. D. Wobschall, “IEEE 1451 -- A Universal Transducer Protocol Standard”, IEEE AutoTestcon (Baltimore, Sept 2007).
13. D. Wobschall “IEEE 1451 Smart Transducer Standard for HVAC Applications”, ASHRAE meeting (Chicago 2009) <http://www.ashrae.org/publications/page/2231>
14. Sensor Web Enablement and OpenGIS SensorWeb (Open Geospatial Consortium):  
<http://www.opengeospatial.org/functional/?page=swe>
15. Open Geospatial Consortium (OGC) website: <http://www.opengeospatial.org/>
16. SensorNet at ORNL: <http://www.sensornet.gov/>
17. SensorML description: <http://vast.uah.edu/SensorML/>
18. OASIS protocol: <http://www.oasis-open.org/specs/index.php>
19. CBRN protocol: <http://www.jpeocbd.osd.mil/guardian.htm>
20. E. Song, K. Lee: “A Unified Web Service for IEEE 1451 Smart Transducers” *IEEE T. Instrumentation and Measurement* 57(8): 1749-1756 (2008)

## Schematics and Software

The full schematics (pdf format) for the NCAP and TIM (3 versions) are provided in separate file on a flash drive or available for download from this site:

<http://www.eesensors.com/media/wysiwyg/pdf/NCAP-Main-Schematic.pdf>

## Appendix C. Circuit Details for NCAP

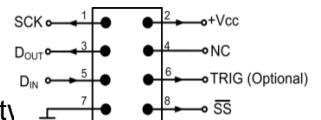
The NCAP consists of two boards each with a microcontroller (Figs. 2 and 3). The main board handles the local network or ports while the second handles the Ethernet interface. The NCAP uses separate microcontrollers for the Ethernet/Internet and the WiFi (and serial) local network functions because both require rapid response and significant computing power. This approach was chosen because combining the functions into a single microcomputer to reduce the hardware cost would increase the program complexity which is undesirable in a demonstration kit. The two boards are stacked one over the other and connected through a 30-pin connector (P4) via SPI. The connection to the Internet is via Ethernet (not WiFi). Power is provided via 9v to 12v wall adaptor.

The WiFi section utilizes a WiFi module (U11) with a built-in antenna. It communicates with the local WTIM network. The Ethernet board is programmed via J5. Power (to both boards) is supplied via J1 which is connected to a 9v to 12v DC wall supply.

The serial section utilizes a RS232 connector (J1) with a pinout following the Dot 2 standard which includes +5v power on one pin. The TIM is connected to the NCAP by a RS232 cable. USB#1 (J3) behaves in software like a second Dot 2 (RS232) port. A second USB port (#2 or J4) is used for reprogramming the board, if necessary. Initially the main board is programmed via J5.

The time on NCAP can be updated via the Internet. Switches S1 and S2 are used for factory reset and updating firmware.

The SPI connector (right) is available in hardware but not implemented in softw



The full schematics (pdf format) are provided in a separate file available for download from this site : <http://www.eesensors.com/media/wysiwyg/pdf/NCAP-Main-Schematic.pdf>

## Appendix D. Flowchart for NCAP

Not yet available

## Appendix E. Circuit Details for Dot 2 (Serial) TIM

The Dot 2 TIM is connected to the NCAP though a 9-pin (RS232) connector which also furnishes the power (+5v).

A detailed block diagram of the Dot 2 TIM is shown in Fig. E1. It has these I/O lines on a *30-pin connector*. The complete diagram is attached to the end of this appendix. The inputs are:

- Four analog inputs, -5 to +5v (switch for 0 to +5v)
- Two analog outputs (0 to +5v)
- Two 8-bit digital ports (One 8-bit input and one 8-bit output)

In addition these transducers will be present:

- Temperature sensor (SHT11)
- LCD (16x1line)
- Relay (dry contact)

In addition there are 4 LEDs

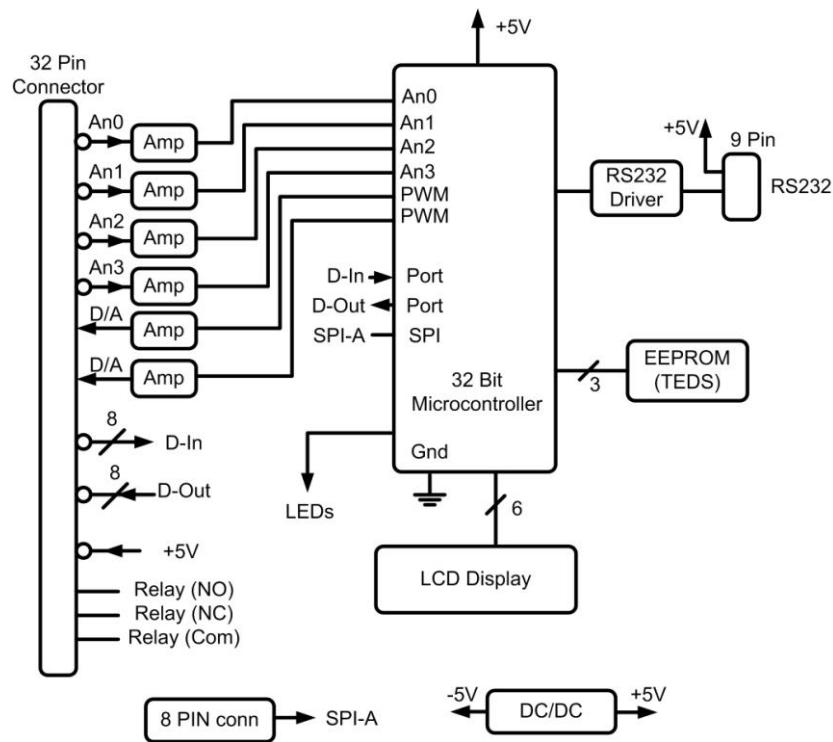


Fig. E1 Dot 2 TIM Detailed Block Diagram

The pinouts for the connector J2 are shown in Table E1 (same as Table 1).

Table E1. Connector J2 Pinouts (Dot 2 and Dot 5)

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	Analog In #2	9	Digital In - 3	17	Digital Out - 3	25	nc
2	Analog In #1	10	Digital In - 2	18	Digital Out - 2	26	Relay NO
3	Analog In #4	11	Digital In - 5	19	Digital Out - 5	27	Ground
4	Analog In #3	12	Digital In - 4	20	Digital Out - 4	28	+5v out
5	Analog Out#2	13	Digital In - 7	21	Digital Out - 7	29	nc
6	Analog Out#1	14	Digital In - 6	22	Digital Out - 6	30	nc
7	Digital In - 1	15	Digital Out - 1	23	Relay Com		
8	Digital In - 0	16	Digital Out - 0	24	Relay NC		

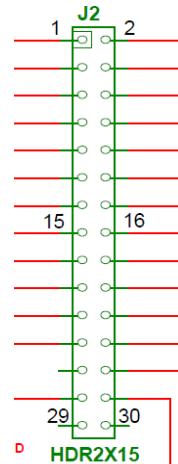


Table E2. Signal List

Signal	# pins	Option	In/Out	Notes
Analog input	4	All	In	A/d
Analog output	2	All	Out	D/a
Digital In and Out	16	Opt A	8in & 8out	Two digital ports
Control switches	2	All	In	Factory reset and updating firmware
LCD	6?	Opt B	Out	2x16 Display
LED (status)	2	Opt B	Out	In addition to power & Communication
Relay	1	Opt B	Out	
Temperature	N/A	All	In	
Humidity	N/A	All	In	
UART	N/A	All	1in, 1out	RS232 – Dot 2 data path

The signal connector (RS232) from the NCAP is shown in Fig. E2

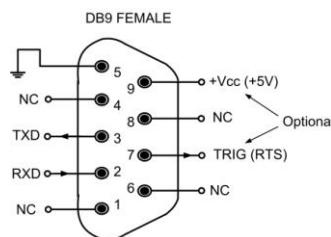


Fig. E2 Connector to NCAP

## Appendix F. Circuit Details for Dot 4 TIM

The Dot 4 standard [3] defines two classes of interfaces (Fig. F1). For the Class 1 sensors, a coax connector is used. For the Class 2 sensors we have selected (4-20 ma and 0-5 v), there will be only 3 analog lines (in addition to the 2 TEDS lines). Note that for Class 1 the analog and digital/TEDS are on the same line but for Class 2 they are separate. The Dot 4 standard does not specify the connector or number of pins (except Class 1, a coax connector).

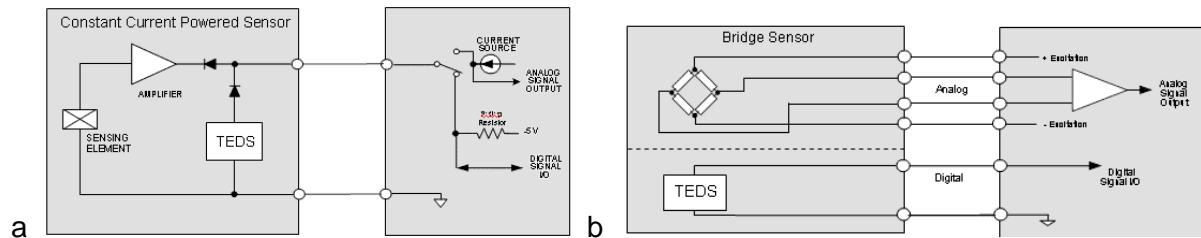


Fig. F1 IEEE 1451.4 Interfaces (a) Class 1 and (b) Class 2

### Block Diagram

A block diagram of the Dot4 TIM is shown in Fig. F2.

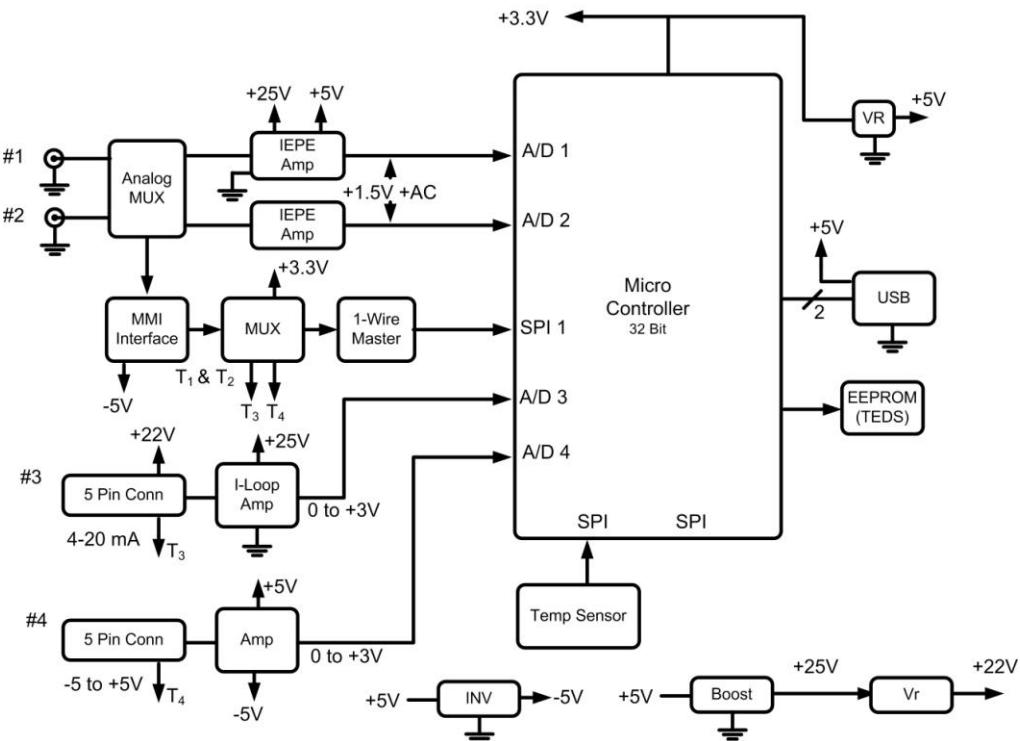


Fig F2 Detailed Block Diagram of Dot 4 TIM

## Class 1 Interface

### Analog Section

The class 1 interface combines an analog line and a 1-wire digital line with a TEDS (Fig. F3a). The analog part utilizes a Integrated Electronic Piezo Electric Interface (IEPE) developed originally by accelerometer (and other) manufacturers ([http://www.mmf.de/iepe\\_standard.htm](http://www.mmf.de/iepe_standard.htm) which is ref [4]). See Fig. F3. It is also known under other brand-names (e.g. ICP).

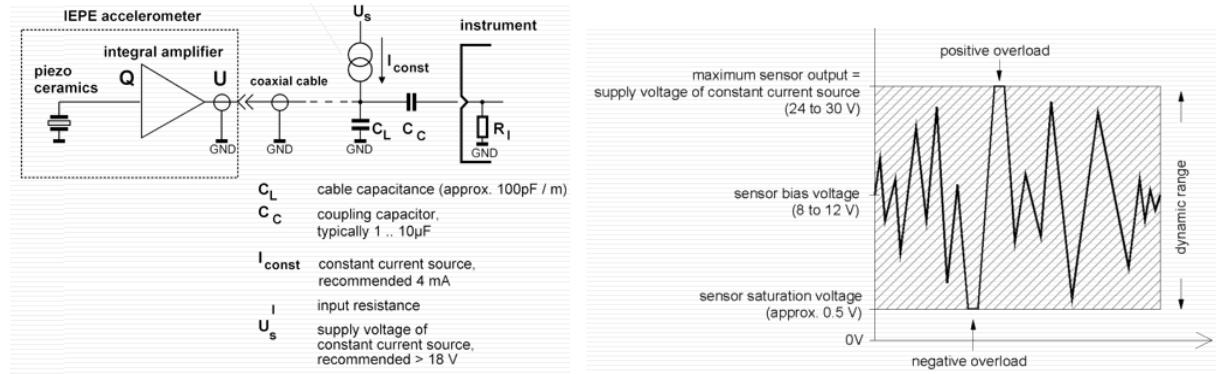


Fig. F3 IEPE Interface

A constant current source is supplied by the signal conditioner (TIM) with an output voltage modulated by the sensor amplifier. The voltage on the wire is between +0.5 and +22 volts (lower than the +24 to +30 v suggested above). Note that the analog signal, including the bias, is always positive, which is switched by the diode. The constant current supply is set at 4.0 mA and the effective load resistor for no signal is about 2kohms. The desired signal is the AC component riding on the bias voltage (6v). The AC signal is coupled through a capacitor to the following amplifier (response 0.3 to 1kHz, min). A detailed circuit is given in Fig. F4.

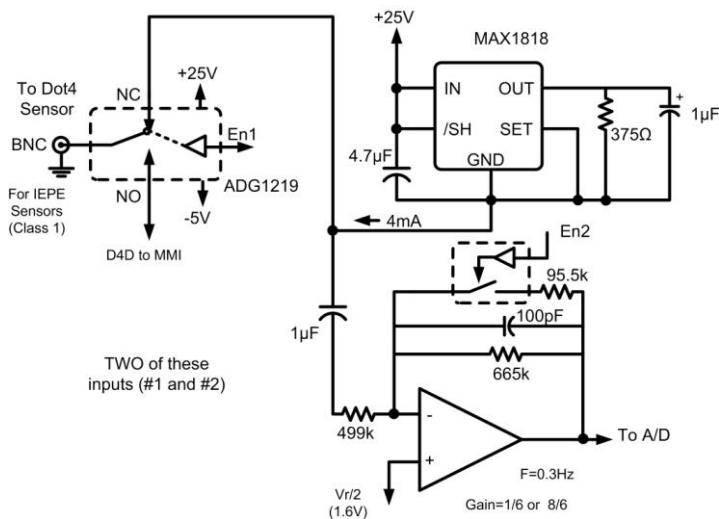


Fig. F4 Circuit Diagram of IEPE Interface

If no sensor is present, the voltage at the connector will go to saturation (about +22 v). This condition is detected by a comparator. When the sensor comes back on line, the TEDS is read again. If the UUID is different, a change in sensor is recorded.

#### *Digital (TEDS interface) Section*

For the Class 1 interface, the TEDS is accessed when the voltage on the wire is negative. Diodes on the sensor side separate the digital (negative) from the analog (positive) signals. The reason for using a 1-wire cable (coax) is that it is much easier to install, especially for small sensors (such as accelerometers), than multi-wire cables. The TEDS is a special EEPROM with a 1-wire interface. Data is transmitted in both directions, depending on timing. The data pin on the EEPROM shifts between input and output (also provides chip power). The EEPROM is standard CMOS but is connected backward at the sensor end (+Vcc to ground). Timing is critical and is best implemented with a special interface “1-wire master”. Converting the bi-directional signal from negative to positive requires a special MMI (mixed mode interface) driver (see [www.maxim-ic.com/app-notes/index.mvp/id/4491](http://www.maxim-ic.com/app-notes/index.mvp/id/4491)) which uses a mosfet pair (and analog switch). This is shown in Fig. F5.

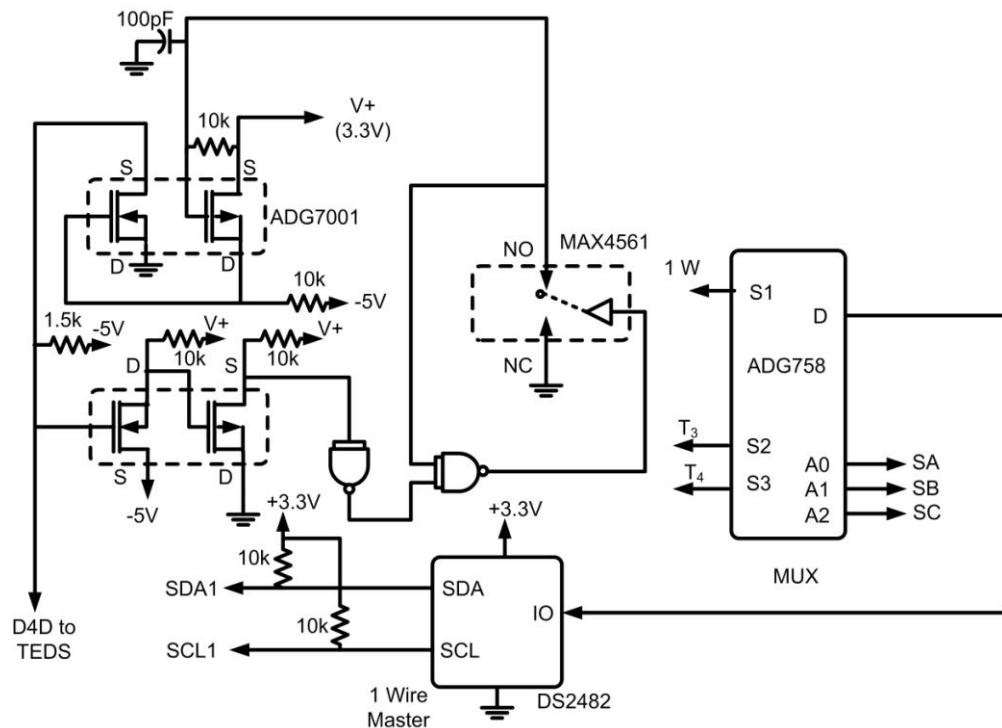


Fig. F5 Circuit Diagram of Dot 4 Class 1 Interface

In effect, the MMI driver converts the backward connected EEPROM in the Dot 4 sensor to one forward connected so that it can be accessed by the 1-wire master chip. The master connects to

a microcomputer via a SPI line. The combination of the two analog switches on inputs #1 and #2 plus the digital mux (#3 and #4) selects one of the four TEDS.

### Class 2 Interface

The class 2 interface simply provides a separate pin for the 1-wire EEPROM (TEDS). It is connected directly to the 1-wire master (no additional interface needed). We have selected two representative inputs (another possibility is a strain sensor). A HD-BNC is the chosen coax connector.

### *4-20 mA Current Loop*

Many industrial sensors have a 4-20 mA current loop (transmitter) as the output. There are two general types: (1) loop powered and (2) sensor excitation voltage external to the sensor. Input #3 is of this type. The +24 volt supply on pin 5 is used for loop powered devices (pins 3 & 4 connected, pins 1 and 2 connected), see Fig. F6

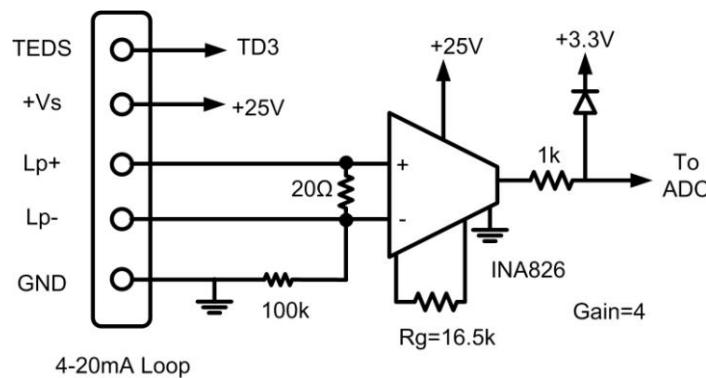


Fig. F6 Circuit Diagram of 4-20 mA Current Loop Interface

### *Signal voltage input*

Many sensors (with built-in signal conditioners) have a voltage as the output. Input #4 is of this type.

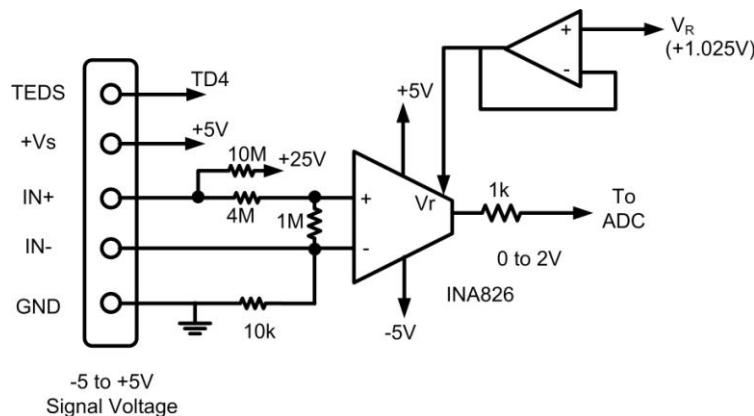


Fig. F7 Circuit Diagram of 0-5v Input Interface

If the input is open (over 10k) , the light pullup resistor brings the input to over +5 v.

#### *D/A Output*

The two analog outputs are the same as for the Dot 2 interface except that the voltage is 0-3 volts because of the battery operation.

#### Power Supply

Power (+5v) is supplied by the USB port (J4). A battery charger is present, although this is not needed for the Dot 4 TIM. Separate 3.3 v regulators are used for the two microcontrollers. A boost switching power supply (U7) generates the +25v (see Fig. F2). A DC/DC supply (U9) provides the -5v.

The power and signals go between boards through redundant connectors J3 and J4.

#### Signal List

Table F1. Signal List for Sensor Microcontroller (U12)

Signal	# pins	In/Out	Notes
Analog inputs	4	In	Class 1 and Class 2
1-Wire master (I2C)	2	Both	For 1-Wire (Dot 4 TEDS)
LCD	4	Out	16x2 alphanumeric (SPI)
Status LEDs	3	Out	

#### TEDS Reading

The reading of the TEDS, which has several sections, is done on power-up and occasionally thereafter, at least if a sensor is removed and replaced. The TEDS data is transferred to a cache TEDS (C-TEDS) on the TIM where it is more easily available. Each TEDS has a unique ID (UUID, 48-bit + a chip family # of 8-bits) which aids in detection of change of sensor. This is in addition to the basic TEDS (64-bits) and a sensor specific section.

The Dot 4 TEDS must be translated to a Dot 0 (Dot 2 TIM) style TEDS which is required by the NCAP. This Dot 0 TEDS is also stored in the cache TEDS and is the TEDS data sent to the NCAP upon request.

The full schematics (pdf format) are provided in a separate file available for download from this site : <http://www.eesensors.com/media/wysiwyg/pdf/NCAP-Main-Schematic.pdf>

## Appendix G. IEEE NCAP Ethernet Hook-up Instructions

The Esensors, Inc. IEEE NCAP Gateway is designed to utilize the Internet to transmit sensor data from a remote acquisition point to a host computer or hosted database.

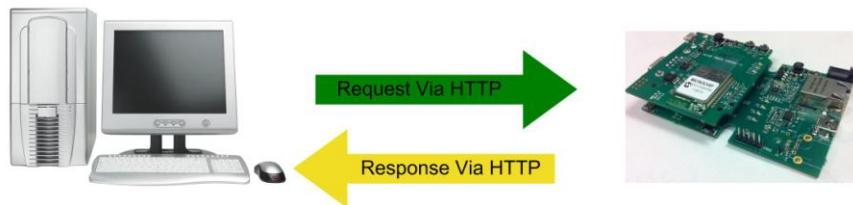


Figure G1 – NCAP Gateway HTTP: Request/Response

Figure G1 (above) shows an NCAP Gateway returning data via a HTTP: response.

The Gateway supports a user configurable *IP address*. This IP address gives it uniqueness on the Ethernet based network which it is attached to. It is through the IP address, that the Gateway is able to take its' place on the Internet, becoming fully accessible to query by the user.

The Gateway consists of a base unit with one Ethernet communication ports. Within the base unit are microcomputers, which handle the Internet protocol, and the communication ports.

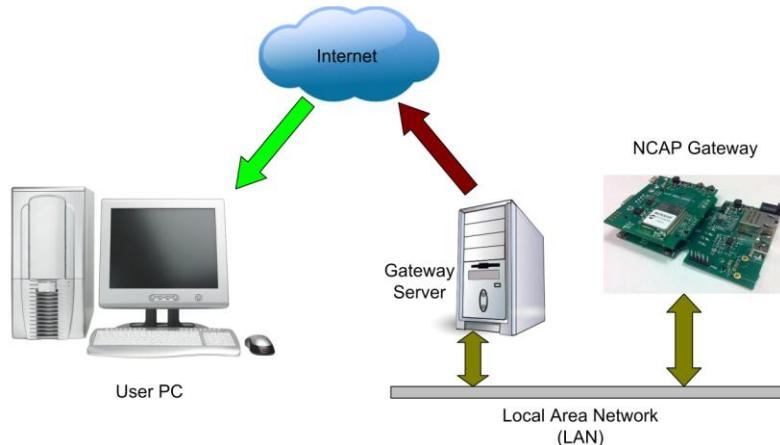


Figure G2 – How NCAP Gateway works.

## IEEE NCAP Gateway Hook-up

To configure and use a NCAP Gateway it must be connected to a PC via an Ethernet cable. Use a straight type Ethernet patch cable.

Figures below show the locations of the connectors for the Ethernet, Wall Power Supply and location of the RESET SWITCH.



Figure G3 – Location of RJ45 and Wall Supply connectors.



Figure G4– Location of RESET SWITCH.

## NCAP Gateway User Interface

Once the **NCAP Gateway** is connected to a PC and powered via the supplied DC Wall Supply, it can be queried by typing the following command into the address line of the browser.

<http://192.168.254.102>

**Note:** The default address for the Gateway is 192.168.254.102. This requires that the PC being used to program the Gateway have its “network interface card” (NIC) set with the same subnet address: 192.168.254.\*\*\*

Once the default IP address is entered, the browser screen should display the “**NCAP Gateway Main Page**” as shown in the figure below.

## IEEE NCAP GATEWAY Network Configuration



Figure G5 – **NCAP Gateway** (Home) Page display

### Network Configuration

The network configuration page is password protected. The default username and password for the device is *admin & password*. The NETWORK CONFIGURATION page enables the user to set up the device IP & host name using the web interface. When the NETWORK CONFIGURATION link is clicked the NETWORK CONFIGURATION page appears; see Figure G6.

From the NETWORK CONFIGURATION screen it is possible to change the various network settings involved in any IP addressable device, port number for the webpage, DNS, gateway, admin ID and password. The device requires an internet connection if you want to set the device time automatically. The device uses TAI time to timestamp sensor data. In case of power loss the device automatically corrects the time upon the Internet connection is available. Once changes are made it is necessary to “left click” on the “Save Config” button to have the changes take effect.

When you change the IP address through the webpage interface, the next page displayed is the “Reboot in Progress” page depicted in Figure G7.

*Note: If you get the message “Error: Connection to NCAP device was lost” then the browser still points to the former IP address. When the new IP address is entered into the address bar the screen will correctly display the NCAP MAIN (Home) Page.*

## IEEE NCAP GATEWAY Network Configuration

Figure G6 – Network Configuration screen.

Figure G7 – “Reboot in Progress” screen

## Basic Troubleshooting Instructions

### 1. Did you change the IP address?

The DHCP server probably assigned the device a new IP address, but your computer's network cache has saved the wrong address. From the command prompt in Windows, enter "nbtstat -R" or "arp -d" to clear old values and try again. Make sure you launch the command prompt as administrator.

### 2. Did you try the IP address?

Try accessing the device directly at the IP address just inputted if DHCP was turned off. (Example: enter "http://192.168.5.23/" into your browser). If this fails, then the IP address you set is not reachable. Accessing device via IP address is more reliable than using host name since some networks may have troubles dealing with host name. If you forgot the IP address and the host name of NCAP Gateway, device, try Step 3 below.

### 3. Use Ethernet Discoverer to Find devices

Microchip's Ethernet Discoverer is a very nice tool to list all devices on the same subnet. Double click *Microchip Ethernet Discoverer.exe*, Click "Discover Devices" button, the IP address, Host name and MAC address of all NCAP devices will be listed. It is also helpful to check NCAP's IP address when it is automatically assigned by DHCP server. *Microchip Ethernet Discoverer.exe* do not need to be installed, it can be copied to and executed from any location.

### 4. Still not working? Device Reset

You can restore the factory default settings by erasing the device's flash ROM.

#### Resetting the NCAP Gateway

- Hold the button by inserting a paper clip into the small hole on the left side panel of the device and then re-power the device.
- The device enters boot loading state which will last for four seconds, both Power LED and Activity LED are extinguished in this state, then they will flash for three times and enter startup state.
- Continue holding the button, the device will enter restoration state, In this state you will notice the Power LED and Activity LED are solid ON in this state.
- You should not release the button until both LEDs are extinguished, which means restoration is done.
- The whole procedure takes about eight to ten seconds. The factory default IP address is "192.168.254.102".

## Appendix H. IEEE 1451 Development Kit Basic Operation

### 1. Dot2 TIM Data channel Assignments

Channel	Signal Name	In/Out	Data Type	J2 Connector	Note
1	ANx1	Analog Input #1	FP(32)	Pin2	-5.0V to 5.0V
2	ANx2	Analog Input #1	FP(32)	Pin1	-5.0V to 5.0V
3	Temperature	Input	FP(32)		-40 to 123.8 C, 0.1C
4	Humidity	Input	FP(32)		0 to 100%, 0.5%
5	DI_0	Digital Input #0	UINT8	Pin8	1: high, 0: low
6	DO_0	Digital Output #0	UINT8	Pin16	1: high, 0: low
7	Relay	Digital Output	UINT8	Pin23(COM), 24(NC), 26(NO)	1: close, 0: open
8	D2A_1	Analog Output	FP(32)	Pin6	0V to 5.0 V
9	LCD	Output	Character		Array for 32 characters
10	J7, D1	Dot4 Analog In	FP (32)		
11	Dot4 TEDS	Input	UINT8		32 octets, in Hex format

### 2. Read Transducer Data

This page is to provide sensor measurement, the *Channel Id* must be sensor type (input), like channel 1, 2, 3, 4 etc. If actuator channel number is provide, the *Transducer Data* field in the output table will be shown as **Not available**, Fig. H1

**IEEE 1451 Development Kit Application**

<a href="#">Overview</a> <a href="#">Status</a> <a href="#">Visualizing Outputs</a> <a href="#">Transducer Discovery</a> <a href="#">Read Transducer Data</a> <a href="#">Write Transducer Data</a> <a href="#">Query TEDS</a> <a href="#">Read Raw TEDS</a> <a href="#">Write Raw TEDS</a> <a href="#">Network Configuration</a>	<h3 style="margin: 0;">Read Data from TIM device</h3> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <b>TIM Id:</b> <input type="text" value="1"/>  <b>Channel Id:</b> <input type="text" value="1"/>  <b>Timeout(seconds):</b> <input type="text" value="10"/>  <b>Sampling Mode(7):</b> <input type="text" value="7"/> <input type="button" value="Read Data"/> </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: left;">TIM Response</th> </tr> <tr> <th>Name</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Error code</td> <td>0</td> </tr> <tr> <td>TIM Id</td> <td>1</td> </tr> <tr> <td>Transducer Channel Id</td> <td>6</td> </tr> <tr> <td>Transducer Data</td> <td>Not available</td> </tr> </tbody> </table>	TIM Response		Name	Value	Error code	0	TIM Id	1	Transducer Channel Id	6	Transducer Data	Not available
TIM Response													
Name	Value												
Error code	0												
TIM Id	1												
Transducer Channel Id	6												
Transducer Data	Not available												

Fig. H1 Read actuator channel

### 3. Write Transducer Data

This page is to set actuator output, the *Channel Id* must be actuator type(output), like channel 6, 7,8,9 etc. If sensor channel number is provide, nothing will be changed.

Example 1: use a jumper wire connect J2 pin 6 (channel 8, analog output) to pin 2 (channel 1, analog Input #1) and set analog output voltage to 3.0V, Fig. H2

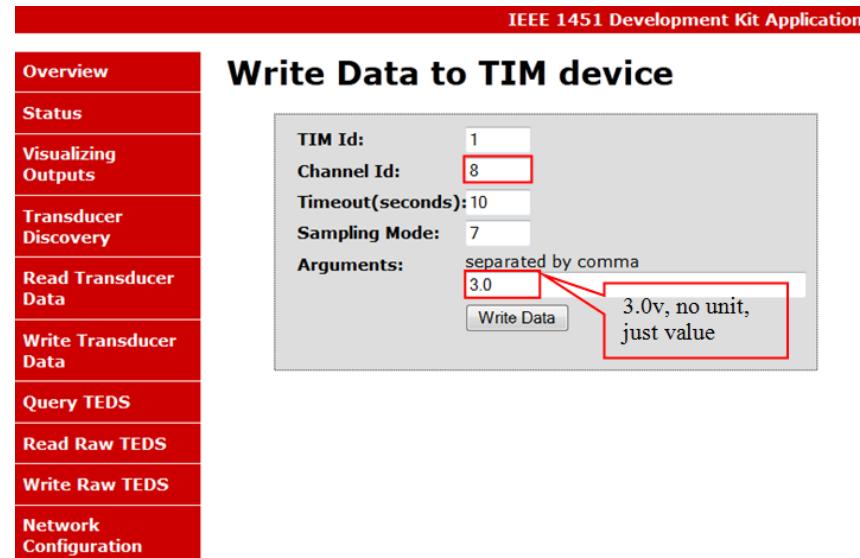


Fig. H2 Write 3.0V to channel 8

Go to "Read Transducer Data" page, check with channel 1's reading, the measurement should be close to 3.0V, Fig. H3. The digital to analog convert circuit on this development kit is based on Pulse-width modulation (PWM), which is not very accurate.

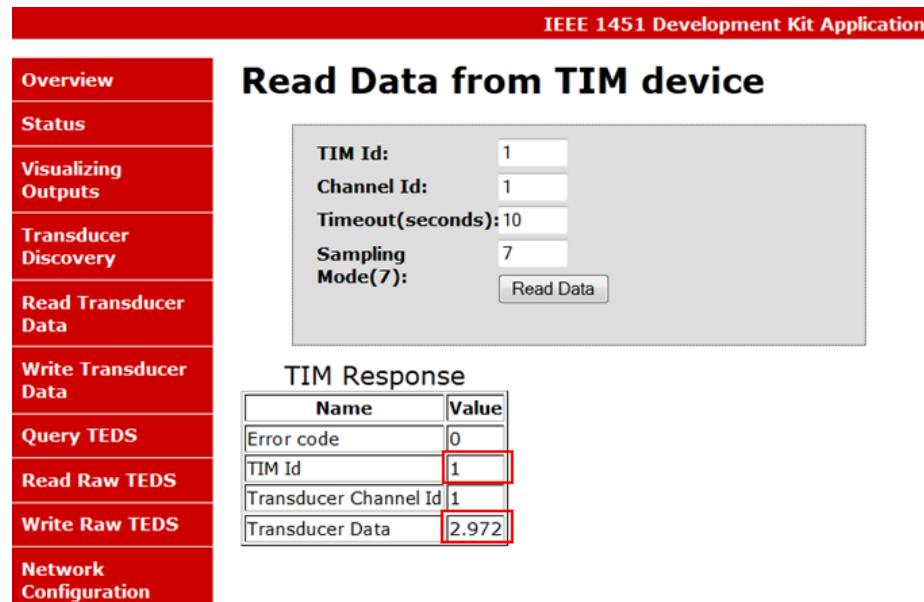


Fig. H3 Read measurement from channel 1

Example 2: write a text message, like "Hello, World!", to channel 9 (LCD)'s *Arguments*, this message will be displayed on TIM device's LCD.

#### 4. Write Raw TEDS

This page is to modify raw TEDS or add a new TEDS. According to IEEE1451.0, each TEDS in the TIM device should be identified by *Channel Id* and *TEDS type*. When the development kit receives inputs from web interface, the TIM device will overwrite the raw TEDS if the *Channel Id* and *TEDS type* associated with this TEDS already exist, otherwise, it will add the input raw TEDS as a new TEDS.

Example: add XdcrName TEDS "ES-TIM04" for channel 4. First, let's read XdcrName TEDS from channel 4. The web interface shows it is not exist, Fig. H4.

Name	Value
Error code	24592
TIM Id	1
Transducer Channel Id	4
TEDS Type	12 , XdcrName
Raw TEDS	

Fig. 4 HRead XdcrName TEDS from channel 4

The raw XdcrName TEDS "ES-TIM04" for channel 4 is "00,00,00,13,03,04,00,0C,01,01,04,01,00,45,53,2D,54,49,4D,30,34,FD,BF", input Channel Id, TEDS type, and raw TED in "Write Raw TEDS" page, Fig. H5, click Write Raw Teds button.

Fig. H5 Write XdcrName TEDS to channel 4

Go back to "Read Raw TEDS" page and read the read XdcrName TEDS from channel 4, we can see the TEDS has been added to the TIM, Fig. H6

Name	Value
Error code	0
TIM Id	1
Transducer Channel Id	4
TEDS Type	12 , XdcrName
Raw TEDS	00, 00, 00, 13, 03, 04, 00, 0C, 01, 01, 04, 01, 00, 45, 53, 2D, 54, 49, 4D, 30, 34, FD, BF

Fig. H6 Read XdcrName TEDS from channel 4

Device will do sanity check for the input from web interface: The raw TEDS must be in Hex format, separated by comma, the length must be 12 to 512 octets. Also the raw TEDS format must fit with general format of TEDS, which is TEDS length followed by data block and checksum, please refer to IEEE1451.0 chapter 8.1. The device will check with TEDS length field, calculate the checksum and compare it with the input checksum field. User can use web interface to calculate the checksum before sending it to device (input raw TEDS and click *Calculate Checksum* button). The *TEDS type* in "Write Raw TEDS" web interface must be same with TEDS access code, class field in TEDS identification header (IEEE1451.0 chapter 8.3), in the raw TEDS, Fig. H7. Device will refuse to modify or add raw TEDS if the sanity check is no passed.

Fig. H7 TEDS type Input must match with TEDS access code in raw TEDS

## 5. Transducer Discovery

To report the available channel from TIM, NCAP reads TIM's MetaTEDS first since MetaTEDS has MaxChan field indicating number of implemented transducer channels , then NCAP reads the each XdcrName TEDS from TIM.

Example: we just add Read XdcrName TEDS for channel 4 in above example, we need to modify the MetaTEDS's MaxChan field from 3 to 4. Otherwise, "Transducer Discovery" will report 3 channels instead of 4, Fig. H8

Name	Value
Error code	0
TIM Id	1
Transducer Channel Ids	1, 2, 3
Transducer Names	1: ES-TIM01, 2: ES-TIM02, 3: ES-TIM03

Fig. H8 "Transducer Discovery" reports 3 channels

To modify the MetaTEDS, read the raw MetaTEDS from TIM first, the *Channel Id* for MetaTEDS should be zero, Fig. H9.

Name	Value
Error code	0
TIM Id	1
Transducer Channel Id	0, TIM
TEDS Type	1, MetaTEDS
Raw TEDS	00, 00, 00, 2A, 03, 04, 00, 01, 01, 01, 04, 0A, 81, C0, F9, 74, 48, 81, F5, 62, 2E, 78, 0A, 04, 3F, 00, 00, 00, 0B, 04, 40, 00, 00, 00, 0C, 04, 43, 88, 93, 33, 0D, 02, 00, 03, F7, FA

Fig. H9 Read MetaTEDS from TIM and copy it

The MaxChan TLV in this raw TEDS is " 0D, 02, 00, 03", change it to " 0D, 02, 00, 04". We don't need to change TEDS length field since no octet had been add or removed. Put the new raw TED in "Write Raw TEDS" page's *Raw TEDS(Hex)* input field, recalculate the checksum by clicking *Calculate Checksum* button then click *Write Raw Teds* button, Fig. H10.

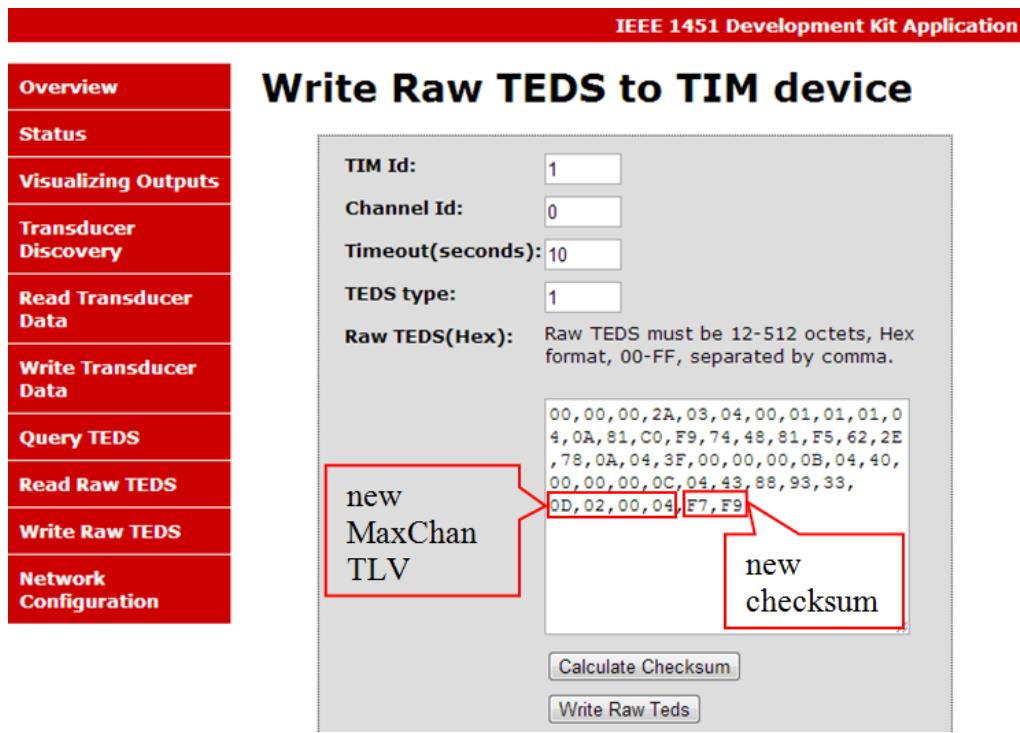


Fig. H10 Write new MetaTEDS to TIM

Next do "Transducer Discovery" again, channel 4 is listed now, Fig. H11

Name	Value
Error code	0
TIM Id	1
Transducer Channel Ids	1, 2, 3, 4
Transducer Names	1: ES-TIM01, 2: ES-TIM02, 3: ES-TIM03, 4: ES-TIM04

Fig. H11 "Transducer Discovery" reports 4 channels

## 6. WTIM Discovery

NCAP device is capable to communicate with 253 WTIM devices (TIM IDs from 2 to 254), See Fig. H12. NCAP maintains a WTIM connection list internally. One can ask NCAP to scan the Wi-Fi network to generate the WTIM list, NCAP will query MetaTEDS for each WTIM Id, mark it as "connected" when it gets a response or "disconnected" when there is no response within 250ms. The scan operation is based on timeout, so it takes up to 70 seconds to scan TIM Id from 2 to 254. The scan time can be reduced by narrowing down the scan range. Whenever NCAP is requested to communicate with WTIM device, NCAP always updates the connection status for the specified WTIM device.

One can also ask NCAP to display (report) WTIM list, save list to non-volatile memory (EEPROM) or erase (clear) list. Please note that for "Report", "Save" and "Clear" operations, NCAP does not communicate with any WTIM device, the operation is for WTIM list cached in the NCAP device.

## Wireless TIM Discovery

The screenshot shows two panels of a web-based configuration interface:

- WTIM Id Scan Range:** This panel contains input fields for "From" (set to 2) and "To" (set to 254). Below the inputs is a note: "Note: WTIM Id must be between 2 and 254. It takes about 70 sec. to scan from 2 to 254". A "Scan" button is located at the bottom.
- WTIM List Cache Operation:** This panel contains three radio buttons: "Report" (selected), "Save", and "Clear". Below the radio buttons is a "WTIM List" button.

Fig. H12 WTIM Discovery

## 7. TIM Type

The NCAP and TIM devices provided support both Dot2 (RS-232) and Dot5 (Wi-Fi) interfaces. NCAP can talk with a TIM either through RS-232 or Wi-Fi. User must specify the interface (Tim type) on NCAP's webpage. For Dot2 (RS-232) interface, the TIM device must be connected to NCAP via RS-232 cable. NCAP will ignore the input value for Tim Id (Dot2 is point-to-point communication, the Tim Id is not necessary). For Dot5 (Wi-Fi) interface, the input for Tim Id must be correct, otherwise, the NCAP will return a non-zero error code.

## 8. Sampling TIM Device

NCAP device is capable to read and buffer a block of sensor data from TIM devices (up to 500 samples). User must configure the settings like TIM Id, Channel Id, samples number, Interval etc before starting the sampling process. This can done in *Sampling TIM Configuration* webpage. NCAP will start sampling as soon as possible if the "Start Time" input is the time before current time, not a future time. Please note the time instance and time interval formats are based on IEEE 1451.0-2007 (Clause 4.9).

It may take some time to collect all sensor data since this depends on the requested number of samples and the sampling interval. Users can always check with the most recent samples by refreshing the *Samples from TIM* webpage, Fig. H13.

## Read Samples from TIM device

Samples from TIM	
Name	Value
TIM Id	WTIM 104
Transducer Channel Id	3
Start Time(TAI secs)	1368484763 , 5/13/2013 6:39:23 PM
Start Time(nanosecs)	335836800
Interval(sec)	0
Interval(nanosec)	500000000
Samples Num	500
Total Samples	7
Transducer Data	001, 1368484763, 231360000, 27.5 002, 1368484763, 733830400, 27.5 003, 1368484764, 233568000, 27.5 004, 1368484764, 732025600, 27.5 005, 1368484765, 234259200, 27.5 006, 1368484765, 732755200, 27.5 007, 1368484766, 233788800, 27.5

Fig. H13 Samples form TIM

### 9. Writing Data block to Tim

NCAP device is capable to write a block of data to actuator channel on TIM device. One can input the settings like TIM Id, Channel Id, samples number, Interval, Arguments etc. This can done in *Writing Data block to Tim* webpage. NCAP will write the data as soon as possible if the "Start Time" input is not a future time. The argument for actuator channel must be separated by comma, the number of input values in "Sample Num" field need not match with the number of comma separated arguments in the "Arguments" field. NCAP device will calculate the samples number based on the "Arguments" field input anyway. For zero input value in the "Interval (Sec)" and "Interval (nanosec)" fields, NCAP will terminate the current writing procedure if there is any.

## Appendix I. Bootloader and Wi-Fi Network Configurations

### 1. Hardware Review

The NCAP hardware consists of two circuit boards. The first one (NCAP-ETH) provides the Ethernet/Internet interface, and the second one (NCAP-MAIN) provides the IEEE1451 Dot0, Dot2 and Dot5 services. The boards are assembled in a stacked fashion, NCAP-MAIN board is stacked on top of NCAP-ETH board. The communication interface between NCAP-MAIN and NCAP-ETH is SPI bus.

The TIM hardware also consists of two circuit boards, TIM-COM board provides IEEE1451 Dot0, Dot2 and Dot5 services while the TIM-SENSOR controls all sensors and actuators. TIM-SENSOR board is stacked on top of the TIM-COM board. The communication interface between TIM-SENSOR and TIM-COM is SPI bus as well.

### 2. Updating the Firmware and Reset State

NCAP-ETH, NCAP-MAIN and TIM-COM boards come with USB bootloader module, which means users can update the firmware for these boards via USB cable without using the in-circuit debugger like ICD3 or PICkit (Fig. I1, I2). TIM-SENSOR board does not have bootloader module, and ICD3 or PICkit is required to update the firmware.

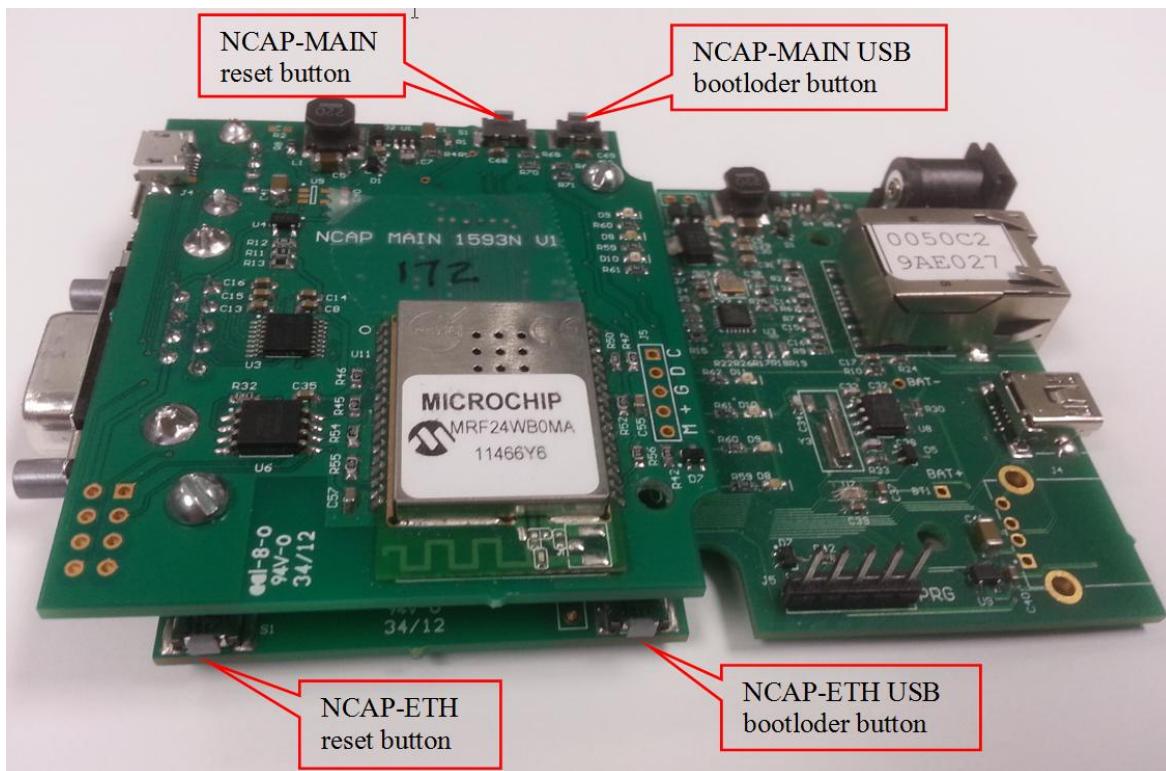


Fig.I1 Reset and Bootloader buttons on NCAP Device

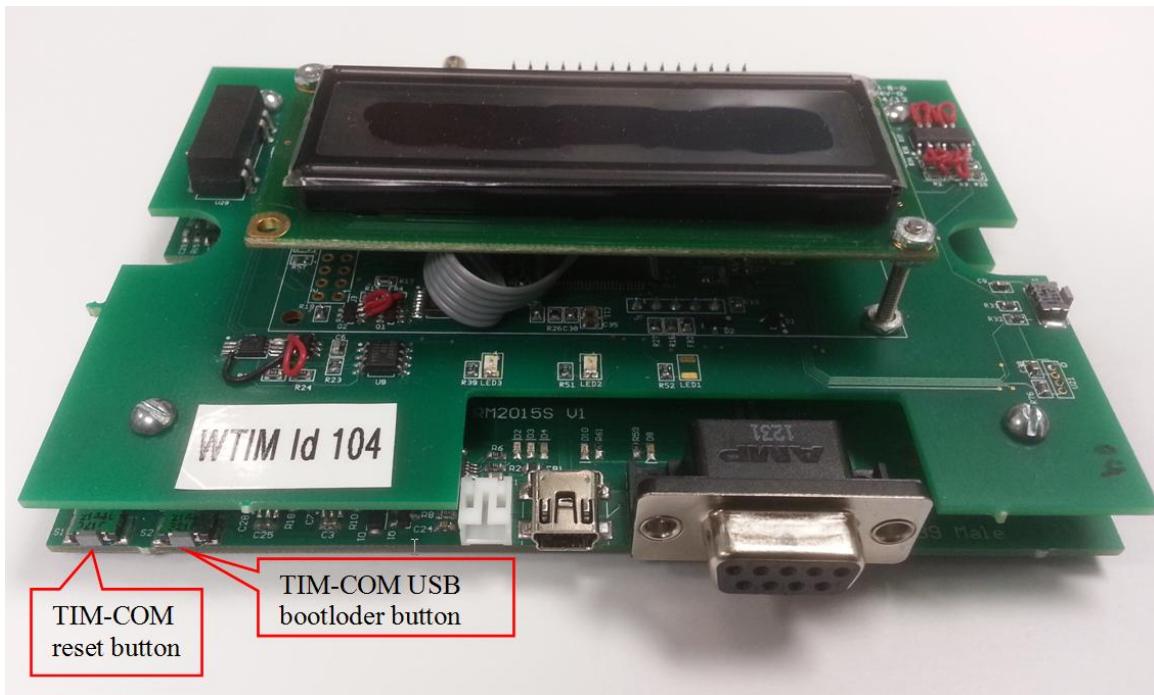


Fig. I2 Reset and Bootloader buttons on TIM Device

*Procedure for updating firmware for NCAP-ETH, NCAP-MAIN or TIM-COM board:*

- Remove the power supply
- Hold the bootloader button and repower the board, bootloader LED will be blinking indicating the board is in bootload mode.

Board Name	Bootloader LED	USB Connector	Type
NCAP-ETH	D10	J3	Mini-B
NCAP-MAIN	D9	J4	Micro-AB
TIM-COM	D9	J4	Mini-B

- Connect the board to a Windows PC with USB cable.
- Run bootloader application " PIC32UBL.exe" on PC. Check the "Enable" checkbox for USB, click "Connect" button, Fig. I3

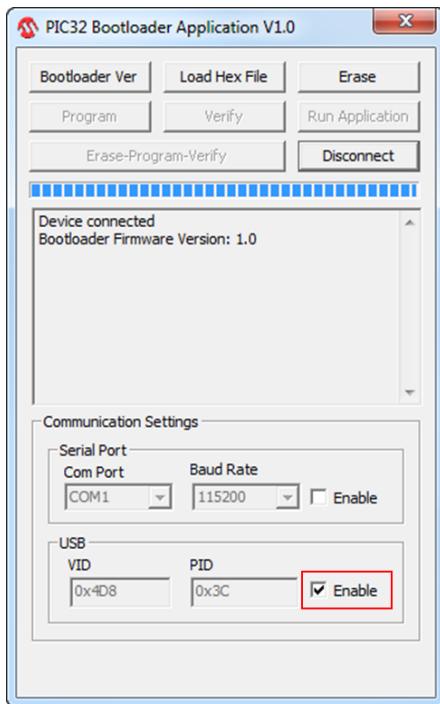


Fig. I3 Bootloader Application on PC

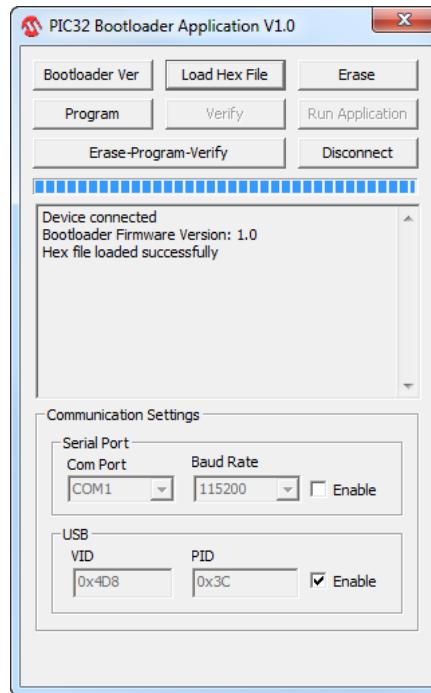


Fig. I4 Load Firmware File

- Click "Load Hex File" button to locate the firmware file on the PC. For easy identification, the firmware file name is named after the board name (NCAP-ETH, NCAP-MAIN or TIM-COM) and extension is "hex". Fig. I4
- Click "Erase-Program-Verify" button to burn the firmware into device
- After the loading is complete, Click "Disconnect" button then remove USB cable from device

It may be desirable to reset the device to factory default state after updating the firmware:

- Repower the device, while holding the reset button PRESSED.
- After power on, both Power LED (green) and Activity LED(amber) will flash for four times in startup state.
- Continue holding the button, the device will enter restoration state, the Power LED and Activity LED are solid ON in this state.
- DO NOT release the button until both LEDs are turned OFF, which means restoration is done.

The whole procedure takes about eight to ten seconds.

The factory default state for NCAP and TIM boards are:

Board Name	IP Address	Host Name	Wi-Fi SSID	Wi-Fi Network Type	Security Type
NCAP-ETH	192.168.254.102	NCAP_BASE_ETH			
NCAP-MAIN	192.168.254.102	NCAP_BASE_WF	NCAP_WF_Setup	Ad hoc	None
TIM-COM	192.168.254.102	TIM_COM_WF	TIM_WiFi_Setup	Ad hoc	None

### 3. Configuring Wi-Fi board Hook-up

To communicate with NCAP-MAIN board, the computer must connect to **NCAP\_WF\_Setup**, which is an ad hoc network hosted by NCAP-MAIN board in factory default state.

First, check with the IP settings of your computer. The option to "Obtain an IP address automatically" should be selected. NCAP-MAIN board comes with a single host DHCP server which can automatically assign IP address to another computer connected to the **NCAP\_WF\_Setup** network. If your computer is using static IP, make sure the IP address is in the same subnet with NCAP-MAIN board and has an IP other than 192.168.254.102.

Next, scan the Wi-Fi networks on the computer. The Ad hoc network **NCAP\_WF\_Setup**, which is established by the NCAP-MAIN board, should show up in the network list, as shown in Fig.5. Connect to **NCAP\_WF\_Setup**. No key is required. Now NCAP-MAIN board is connected and can be pinged.



Fig. I5 Ad hoc networks Hosted by NCAP-MAIN and TIM-COM boards

Connecting to other wireless networks is simple. Click the Network Configuration link on the device's webpage, Fig. I6. Input IP Address, Gateway, Subnet Mask etc, check checkbox Change WLAN Settings before inputting SSID, security Type, security key, if security is required by the desired network, and click the "Submit" button. Please note that WLAN type other than Ad hoc is not acceptable for this application.

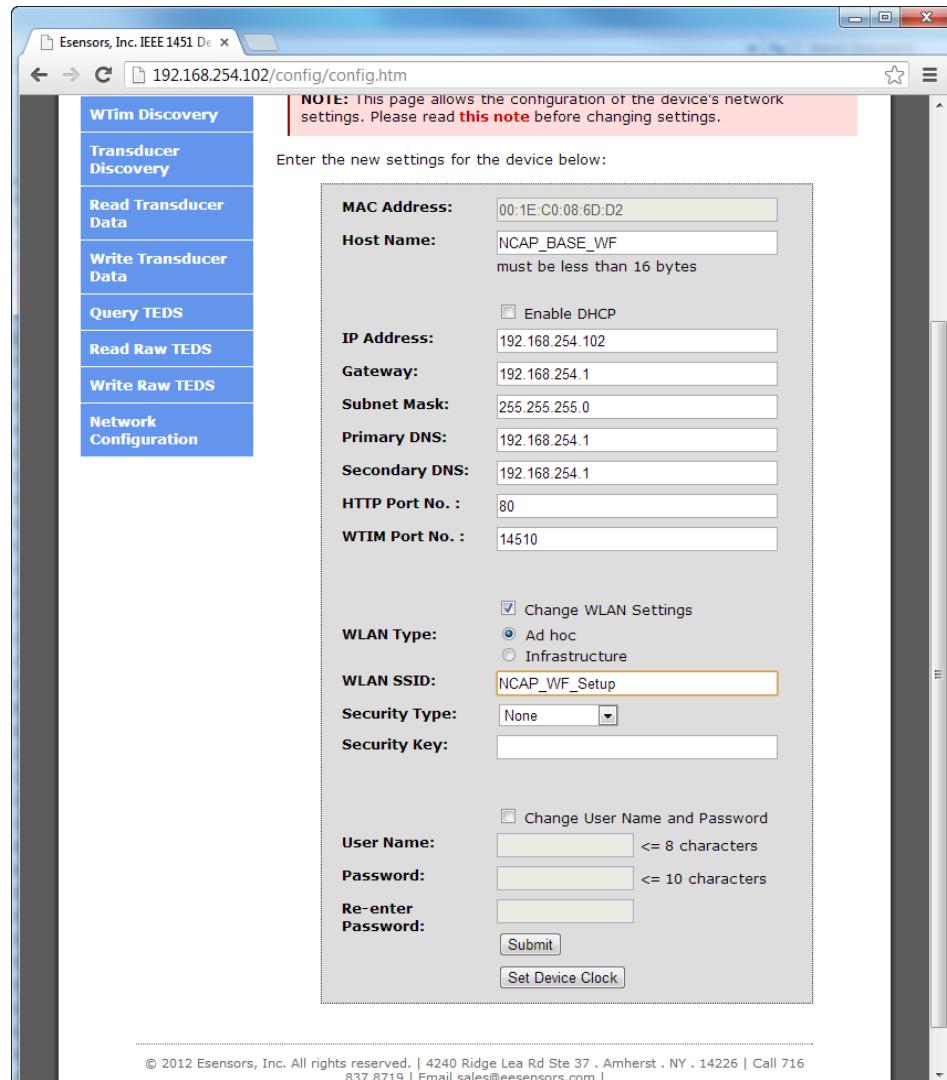


Fig. I6 Network Configuration page for Wi-Fi board

The procedure for connecting computer to TIM-COM board is similar to NCAP-MAIN board, the only difference being the default Wi-Fi network hosted by TIM-COM board is **TIM\_WiFi\_Setup**. The Online LED (green, D9) is OFF when device is connecting to the Wi-Fi network and become solid ON when it is connected.

Note: DO NOT power up multiple TIM-COM boards with default settings at the same time since they have same IP address on the same Wi-Fi network.

#### 4. Uploading Webpage

Input <http://192.168.254.102/mpfsupload> or <http://Device-Host-Name/mpfsupload> in the browser's address bar. One should see, Fig. I7

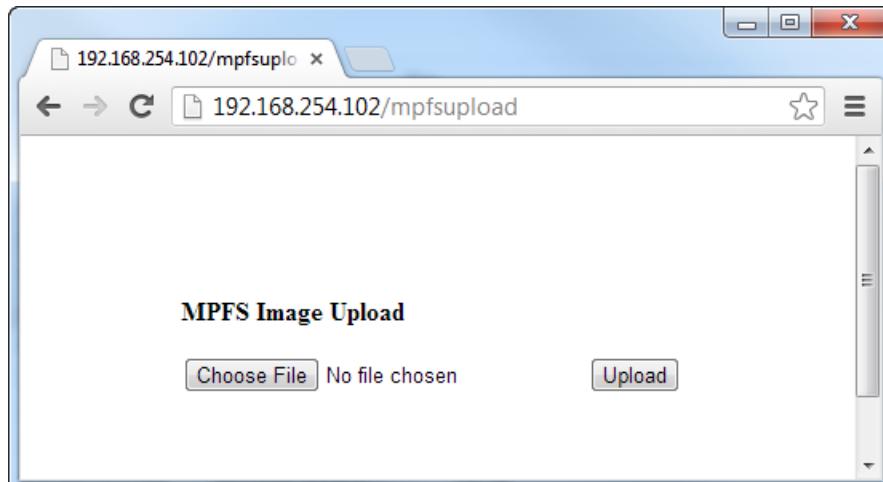


Fig. I7 Upload Webpage

Click the “Choose File” button to locate the webpage file "*IEEE1451\_xxxx\_Webpages.bin*" and then click the “Upload” button to upload the webpage to device.

#### **Important Notes:**

The extension of firmware file is ".hex" and uploaded by Windows application "PIC32UBL.exe". Extension of webpage file is ".bin" and is uploaded by browser via Http. Please don't mix firmware or webpage file to board other than specified, e.g., *NCAP\_Main\_V54202\_bWiFi.hex* and *IEEE1451\_NCAP\_MAIN\_V203\_536\_Webpages05032013.bin* are for NCAP-MAIN board only.

## 5. IEEE1451 Dot5 Wi-Fi Sensor Network

The IEEE1451 Dot5 wireless sensor network is an Ad hoc Wi-Fi network hosted by NCAP-MAIN board. Dot5 communication is established on TCP port 14510 (default). To communicate with NCAP-MAIN board, the WTIM devices must join the same Wi-Fi network, the IP address must be in the same subnet with NCAP-MAIN board and must have unique IP addresses for each device. Also, the WTIM device should have the same WTIM port number as that of the NCAP-MAIN board.

For example, the network settings for NCAP-MAIN board are:

IP Address: 192.168.254.102  
 Gateway: 192.168.254.1  
 Subnet Mask: 255.255.255.0  
 WTIM Port No: 14510  
 WLAN Type: Ad hoc  
 WLAN SSID: NCAP\_WF\_Setup  
 Security Type: None

An example network settings for TIM-COM board (WTIM) can be seen in Fig. I8

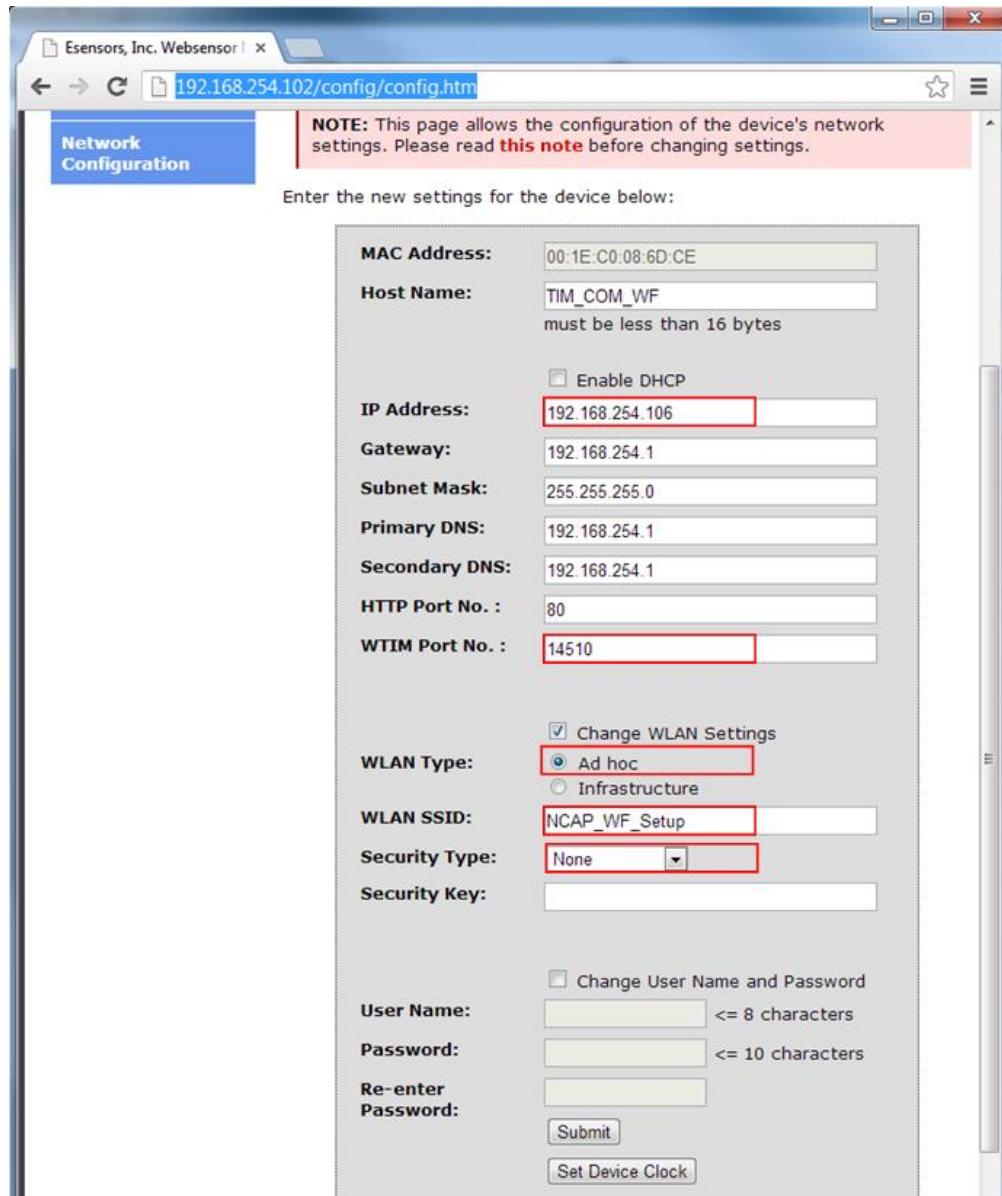


Fig. I8 Example Network Settings for TIM-COM board (WTIM)

Since NCAP-MAIN and WTIM devices are in the same subnet, **NCAP uses the last byte of WTIM IP address as TIM Id**. For example: if WTIM IP address is 192.168.254.106 then its TIM Id will be 106. The TIM Ids of 0, 1 and 255 are reserved. NCAP can communicate with 253 WTIM devices (TIM Ids from 2 to 254). Please DO NOT check "Enable DHCP" for both NCAP-MAIN and TIM-COM boards, use static IP for them instead. NCAP-MAIN board is not capable to assign IP addresses for multiple WTIM devices via DHCP. Your PC can join the Wi-Fi network hosted by NCAP to check with or configure any WTIM or NCAP device.

## 6. Setting Device Time

There are two ways to set the device's clock for NCAP-ETH board. One is to let the device automatically synchronize its clock from Internet every day. There are no accumulated

time errors in this mode. The network configurations like gateway and DNS must be properly set for this feature since device need Internet access.

Another way is to set the device's clock manually. This can be done by clicking the "Set Device Clock" button in *Network Configuration* webpage. Device does not need Internet access in this mode. The disadvantages are: the clock needs to be set whenever device is re-powered since it cannot save time when power is lost. The time may also have accumulated error from longer run times.

Whenever the clock on NCAP-ETH board is updated, it will automatically set the clock on NCAP-MAIN board. NCAP device will put the timestamp for sensor data when it is received from the TIM board. The timestamp format is based on IEEE 1451.0-2007 (Clause 4.9 Time Representation), 32-bits TAI seconds and 32-bits nanoseconds.