DOI: 10.21608/msaeng.2023.291927 VOL. 2 ISSUE. 2 pp.1022-1047

MSA UNIVERSITY ENGINEERING JOURNAL

USANet: an open platform for USB-based Sensors/Actuators Network

Mokhtar A. A. Mohamed[1,a,c], Engy El-Shafeiy[2,b], Ramadan Gad[3,a], and Ahmed ElMahalawy[4,a,d]

[1,a] Comp. Sci. & Eng. Dept., Faculty of Electronic Engineering, Menoufia University, Egypt

[2,b] Comp. Sci. Dept, Faculty of Comp and Artificial Intelligence , University of Sadat City, Egypt.

[1,c] Comp. System Eng. Dept., Faculty of Engineering, MSA University, Egypt.

[4,d] Dept. of Comm.& Comp. Eng., HIE, ElShorouk Academy, ElShorouk, Egypt.

E-mail: [1]mokhtar.mohamed@el-eng.menofia.edu.eg, [2]Engy.Elshafeiy@fcai.usc.edu.eg, [3]ramadangad@yahoo.com [4,*]ahmed.elmahalawy@sha.edu.eg (Corresponding author)

Abstract

The paper presents and proposes USB-based Sensors/Actuators Network (USANet) system. Proposed USANet system is maturely designed according to IEEE 1451 concepts and component-based approach. It consists of five STIMs. These STIMs are Measurement Module of Speed (MMS), Controller Module for Motor (CMM), three Measurement Modules of Distance (MMDs). They are connected to a laptop NCAP host via five USB 2.0 interface cards. Each USB 2.0 interface card is dedicated for only one STIM to exchange messages of sensory data or control commands with NCAP host. USB 2.0 interface card represents TII of such USANet. NCAP host run a program called SCCD. SCCD is the abbreviation of Sensory data Collector/Command Distributor. It is an open source model that is developed by C# programming language under the dot-Net 4.0 framework to provide portability. SCCD is employed to collect sensory data from MMS, and three MMDs. In addition, SCCD sends control commands to CMM. All hardware and software modules of USANet are designed and developed completely in our labs. Two

performance metrics are used to validate and verify the feasibility of such USB-based Networked system. They are Feedback Receiving Time (FRT) and Sensory Data Message Time (SDMT). These time-based metrics are chosen for evaluation because the provision of timely information is crucial in many applications.
Keywords: Sensors/Actuators Network (SANet) ; USB-based SANet (USANet).

1. Introduction

Nowadays, the diversity and availability of embedded processing apparatus such as microcontrollers / digital signal processors (DSPs) / field programmable gate array (FPGA) chips, etc., is facilitating the prevalence of smart transducers (sensors or actuators). Advanced capabilities of communication and data processing in such embedded apparatus enable smart transducers to be employed in a networked environment, often called Sensors / Actuators Network (SANet). SANet, in fact, is used in many application areas which are classified into two broad perspectives according to types of involved transducers. The first perspective is Remote Monitoring Systems (RMS). And the second one is Distributed Measurements and Control Systems (DMCS). RMS, often called sensors network, are found in a broad range of applications in which all involved transducers are only sensors such as meteorological systems [1], health-care diagnosis devices as Electroencephalography (EEG), mammography, or ultrasonography. Joining actuators alongside with sensors lead to outcrop the perspective of DMCS. It includes, but is certainly not limited to, robotics, industrial assembly lines, printing machines, CNC machines, automotive, home automation, and health and welfare devices, etc.

SANet was regulated by the IEEE 1451 smart transducer interface standard [2][3]. The ultimate aims of IEEE 1451 standard are to provide standard guide about means (hardware and software) for achieving smooth transducer-to-network interchangeability and transducer-to-network interoperability. The key components of the IEEE 1451 are the Network Capable Application Processor (NCAP), Smart Transducer Interface Module (STIM), and Transducer Independent Interface (TII) which used for low-level communication between STIM and NCAP. Figure 1 shows the key components of IEEE 1451 Standard.

STIM functionality is performing periodic or on-demand measurement / control functions for a single or array of sensors and actuators. NCAP is a centralized device which comprises of three software and hardware layers; Sensory data Collector/Command Distributor (SCCD), Application Processing Framework (APF), and Network Communication Protocol (NCP). SCCD is a mandatory program to collect sensory data from / send control commands to STIM. APF revolves around transplanting and developing algorithms viz, decision-making and intelligence approaches [4], data fusion management [5][6], GUI-based display tools, and / or security algorithms. Using a standard NCP is imperative to manage high-level communication between heterogeneous types of NCAPs such as EtherCat, Real-Time Ethernet (RTE), wireless technologies, or internet [7].

Integration of such three-layers model in one-chip or one device plagues NCAP's designers. They search about hardware devices that can enclose and harmonize between heterogeneous software approaches of NCAP's three layer model. Also, NCAP device should be able to manage the gradual growth of data during run-time of such software approaches. To the best of our knowledge, low-cost chips (as microcontrollers and DSPs) possess limited resources such as computation time and storage capacity. They are adequate to be employed as processing Unit (PU) of STIM as shown in figure 1. All previous reasons motivate NCAP designers to employ devices such as less-expensive PCs, embedded PCs, or servers to avoid overwhelming storage problem and performance degradation.

Innumerable researches have been and still are proposed to develop various solutions of STIM, TII, and NCAP. Utilized implementation and evaluation frameworks bisect current and predicted researches into two key trends: simulation-based frameworks trend and Instrumentation-based frameworks trend. In some SANet applications (such as Wireless Sensor Network "WSN" and Industrial plant), experimental implementation, verification, validation, and/or evaluation are quite difficult. Reasons behind this difficulty are high cost and lack of instruments that can measure network dynamics and uncertainties. So, some of researchers prioritize using simulation frameworks [8] [9] to implement and validate their proposals.

On the other side, other researchers adopt producing various standard prototypes and solutions because simulation frameworks cannot give real results based on the hardware implementation, real-time hardware analysis, and real components performance and operations. The previous shortcoming of simulation frameworks can lead to catastrophic consequences when the real-hardware solutions are deployed.

The main objective of this paper is presenting and deploying a novel and an open PC-based NCAP. It is capable of managing the collection of sensory information / distribution of control commands to a single or multiple µC-based STIMs. A µC-based TII was designed and employed to provide USB-based and Peer-to-Peer (USB-P2P) communication between such NCAP and each one of its STIMs. Prototypes of µC-based STIMs and µC-based TII are completely in-lab designed firmware. We envision that utilization of USB-P2P technique with in this paper will lead to the prominence of what is called USB-based SANet (USANet) systems. Deployment of large-scale and complex USANet systems necessitates applying a component-based approach during integration process [10]. Component-based approach revolves around designing firmware (i.e. hardware and software) modules that can be developed, and tested separately. They can be seamlessly installed, upgraded, replaced, or moved without affecting the rest of the systems.

The outline of this paper is as the following: Section 2 presents the related work and some of commonly proposed prototypes about SANet. A detailed description, about components of our proposed USANet, is presented in section 3 including firmware solutions of STIMs, NCAP, and TII for such USANet system. The practical experiment setup and performance evaluation of such proposed USANet are shown in section 4. Finally, a conclusion is described in section 5.

2. Related Work

This section is devoted to give short survey about commonly proposed prototypes and frameworks of SANet components. All the following researches will confirm unintentionally that PC is the most suitable hardware for NCAP while µCs are more convenient as the core hardware element for STIMs.

Song et al. [11] presented remote measurement system for force monitoring and control of robot wrist application via internet. STIM was designed using six-axis force/moment sensors, one ADuC812 for sensory data processing and TEDS. NCAP is a PC with a Pentium III – 1GHZ processor and 256 MB SDRAM. The responsibility of NCAP is managing all network communication with a STIM via a CAN bus adaptor (ADLink PCI-7841). TII was designed by using a CAN controller called SJA1000 and a CAN transceiver called 82C250 to construct a 4-wire CAN bus interface which represent low-level communication between STIM and NCAP. Sensory data and other facilities are remotely displayed to users at the same LAN or across internet via a Java-based web application program running on NCAP. Communication from sensor to network is performed using 10 bytes data frame. Song et al. [11] supported using what is called Virtual TEDs [30]. Virtual TEDs terminology means that there is no need to store TEDs data block inside embedded physical memory of STIM. Transferring stored TEDs from STIM to NCAP consumes a considerable time interval at initial communication, which can affect negatively the overall performance. As a result, TEDs data block, which is used in [11], are deployed in a form of separate file stored on the NCAP host, downloadable from the internet. They claimed that this way is very suitable for users to manage and update related information of the sensor.

Chunshan et al. [12] designed NCAP model using ARM processor (S3C2410) and RTOS (Linux OS). This NCAP was supported by Session Initiation Protocol (SIP) [13] to provide connections between it and its SIP-based networked sensors. Chunshan et al. stated and confirmed that the reason for choice SIP is converging any distributed measurement and control system (DMCS) with IP-based information network because SIP is compatible with many protocols such as DHCP, DNS, HTTP, RADIUS, RTP, SDP, TCP, UDP, and some other specially protocols defined by users. Also, three-way handshake messages facilitate on-line register of sensors. In this research, there is no explicit description about design of used STIM(s), and feasibility verification of communication between such NCAP and its STIM(s).

In [1], Jin et al. proposed two-tier meteorological data acquisition system. Involved sensors were divided into special and basic meteorological sensors. Each special sensor(s) is (are) connected to a sub-collector. Sub-collectors represent the first tier of this data acquisition system. Main-collector, in this system, formed the second tier because it collects sensory measurements directly from its basic sensors and also receives collected sensory data from each sub-collector. The microprocessor of main collector and sub-collector employed AT91SAM9263 chip and embedded Linux OS as the core platform of the system software design that depends on the needs of collector type. There is no need to change the wiring connection of the existing sensors, when the system is expanded in case of adding new sensor(s). It is just needed to add new sub-collector and/or sensors to the system and conduct simple software upgrade or configuration. With a reference to the IEEE 1451 standards, sub-collector is considered similar to STIM structure, while main-collector is similar to NCAP.

Wu et al. [14] applied SANet principles to remotely monitor and control the facilities and indicators in henhouses. Only one STIM was used in this paper which consists of temperature sensor, humidity sensor (SHT11), and light sensor (TSL2561). A ZIGBEE networking module (CC2430) represents the TII which responsible of transferring sensory data to NCAP. NCAP comprised of more advanced ARM9 processor (S3C2440) as the core, Linux OS as development platform to realize data collection and transmission through its ZIGBEE module. Linux kernel is transplanted with web server called "Boa" to facilitate processes of monitoring and controlling via internet.

In [15] Wang et al. has introduced the design of a class of typical networked wave maker system based on EtherCat protocols [16]. SANet system in this paper consisted of 10 slaves stations and PC-based master station (NCAP). Each slave station integrated the STIM circuit and EtherCat controller (TII) on one board. dsPIC33FJ256MC710 is employed as the core controller with in STIM to collect sensory data from 16 sensors and sending driving pulses for one servo motor. Also, each slave station has its own EtherCat controller (ET1100) to provide bi-directional communication interface with NCAP which plugged a standard Ethernet network card.

Centralized PC-based remote monitoring system has been developed by Datta et al. [17] to control and monitor multi-three phase induction motors. In such system, each motor is interfaced with a µC-based Dedicate Hardware Unit (DHU) to form the STIM. DHU is developed around AT89C52 microcontroller which is used to extract different motor measurements that incudes input voltage, input stator current, supply frequency, power factor, and motor speed. Each DHU has its unique "MAC Addr" to identify itself individually and explicitly. In PC-based NCAP of this system, Microsoft windows GUI software is developed using Visual Basic (VB) programming language. The salient strength point of NCAP is its ability of on-demand monitoring and controlling each motor individually. Networking purposes is realized by developing bus-based serial connection methodology between RS485 module in each DHU and RS232 port on PC via TTL-to-RS485 converters.

Kumar et al. [18] presented a review of the literature on environmental monitoring systems. The review elucidated brief comparison of more than 20 researches according the hardware and software technologies employed with in proposed prototypes of STIMs, TIIs, and NCAPs. Kumar et al. developed a µC-based STIM module in which the core microcontroller (PIC 18F4550) received information from sensor array viz. humidity (HIH-4000), temperature (LM35CZ), O2 , CO, CO2, SO2-D4, NO2, SO2-BF, OC2, and OC1 via 10 channels. All sensory information is transmitted to a PC-based NCAP via USB 2.0-based TII. In this system, PC-based NCAP was transplanted with an USB driving program that fully developed in LabView 9.0.

3. Modules of Proposed USANet

In this paper, Proposed USANet consists of three broad categories of modules as the following:

1- Modules of STIMs:

STIMs are three types; Measurement Module of Speed (MMS), Measurement Module of Distance (MMD), and Controller Module for Motor (CMM).

2- Module of TII:

This module is represented by an in-lab designed USB interface card.

3- Module of PC-based NCAP:

This module is represented by a PC as the hardware part and a SCCD model as the software program.

3.1. Measurement Module of Speed (MMS):

MMS consists of three firmware parts (i.e. sub-circuits) such as: Shaft encoder sensor, circuit of speed meter (CSM), and circuit of 7-segement display. All these sub-circuits are shown in figure 2. MMS is used to measure and to send periodically speed data od a DC motor to NCAP.

Shaft encoder sensor consists of a pair of infrared emitter and a phototransistor detector. They are mounted in both sides of a spinning disk that has holes cut in it. These holes are distributed uniformly along the broad edge of this spinning disk as shown in figure 2. The spinning disk is mounted on the output rotor shaft of any DC motor or an axle. The job of phototransistor is generating a train of "ON" and "OFF" digital pulses and deliver this pulse train signal to main processing unit of CSM. "ON" pulse refers to that phototransistor receives the light that passed via a hole from the emitter, otherwise, it is referred as "OFF" pulse. "ON-OFF" pulses train is assigned as the disk spins.

PIC 18F452 μC is employed as the main processing unit in circuit of speed meter (CSM). The role of CSM is counting periodically "ON" pulses that received from phototransistor at fixed time periods called Sending Sensory Data Interval (SSDI). After that, CSM displays "ON" pulses count to a user via circuit of 7-segement display and send this "ON" pulses count to its dedicated USB-based 2.0 TII via a message called Sensory Data

Message (SDM), before beginning to count pulses at next SSDI. The format of SDM is shown in figure 3.

3.2. Controller Module for Motor (CMM)

CMM is an in-lab designed firmware as shown in figure 4. L298N dual H-bridge IC [21] and PIC 18F452 μC [22] are employed and integrated to construct CMM. The job of L298N is controlling direction of any DC motor to make it able to rotate in clockwise (CW) direction or counter clockwise (CCW) direction. It has the ability to alter voltage polarity across two voltage connection of any load. L298N is commonly used in motion applications such as robotics.

PIC 18F452 μC is employed to perform many tasks. Firstly, it receives a control message from its dedicated USB-based 2.0 TII with 5 bytes data frame as shown in figure 5. After that, start, command, and end bytes are discarded. Thirdly, it extracts information of New Direction and New Duty Cycle and assign these data to Nd and DCn parameters respectively. Fourthly, it generates a Pulse Width Modulation (PWM) signal which equivalent to the received value in DCn to control the speed of any DC motor. Finally, it sends a feedback message to NCAP via its dedicated USB 2.0 TII.

The feedback message can be classified to three types; positive, warning, or negative feedback. Positive feedback message is a prove that CMM received successfully a new control message and executed new direction and new speed commands on a DC motor. Warning feedback message means that CMM cannot execute the new direction and new speed commands because it may cause failure issues on such DC motor. The situation is completely different when CMM will send a negative feedback message because it means that CMM does not receive any control message over a specific time interval. The main goal of negative feedback message is to refresh connectivity between CMM and NCAP.

3.3. Measurement Module of Distance (MMD)

Measurement Module of Distance (MMD) is an in-lab designed circuit which is mainly used for non-contact and accurate distance measurement of an object [23]. MMD, often called ultrasonic range finder, works according to a naive principle: ultrasonic waves (pulses) are propagated from an ultrasonic peizzo transmitter device with a predefined and specific frequency. Another ultrasonic sensor, called ultrasonic receiver (i.e. detector), try to sense and detect the echo reflected from the first obstacle (object) in front of MMD.

The time consumed between the transmission and the receiving of reflected echo, is assumed to be equal to the double distance of the sensed obstacle. Ultrasonic range finders have a very broad possibility of exploitation in various fields of science and industry such as robotics [24],[25], and measurement of objective vibration [26].

MMD circuit consists of several electronic components which its practical circuit is shown in figure 6. The electronic components are:

- PIC 16F873A µC.
- 40KHZ ultrasonic Piezzo transmitter with its receiver.
- a LC resonator with its freewheeling diode.
- Transistor with high switching frequency.

PIC 16F873A µC, which is developed by Microchip [20], is used to perform two main tasks;

- The first task is related to driving ultrasonic peizzo transmitter by using one of its built-in modules of PWM.
- The second task is related to use the received output signal from ultrasonic peizzo receiver and estimate the accurate distance of the discovered obstacle.

The PWM signal is generated from the RC2 pin of the PIC 16F873A µC with operating frequency equal to 40 KHZ to cope with the operating frequency of the employed peizzo transmitter [27]. First of all, PWM signal is used to generate and send ultrasonic waves for time period of 300 µsec. secondly, one of PIC timers (i.e. Timer1) is enabled and started to measure the consumed time from the moment of finishing transmission of ultrasonic waves until receiving the first echo signal from ultrasonic receiver which is connected to Analog to Digital Converter (ADC) channel 1 of PIC µC.

The ultrasonic receiver (detector) is made to not operate in a constant time (it is often called Protect Sensing Interval "PSI" and which is about 600 µsec) after sending out ultrasonic waves to prevent from the wrong detection which is due to the influence of the transmission waves. After thus, the ultrasonic receiver is allowed to try detecting the echo signal for a specific constant time which is about 32.768 msec. this time period represents Timer1 overflow time and it is called "Maximum Allowed Sensing Interval (MASI)". If the ultrasonic receiver fails in detecting any echo signal along MASI time period, it means that the distance in front of DDM is free from any obstacles for approximately up to 5 meters.

When ultrasonic receiver successes to detect the first echo signal, timer 1 is stopped from counting more. A 16-bit value in its two registers TMR1H and TMR1L is constructed and saved in a buffer. This 16-bit value represent the time elapsed between the transmission of ultrasonic waves and the detection of the first reflected echo signal.

Buffer has the ability of saving measurement samples of timer1 values which are should be averaged to be used with equation 1. Equation 1 is used to estimate the measured distance of the observed obstacle in front of MMD. At last, PIC µC composes its sensory data message (SDM) before estimating the next distance of the next obstacle. SDM is similar in its format to the one in MMS except that it contains information of the measured distance between start and end characters. Similarly to MMS, SDM is transferred to its dedicated USB 2.0 interface card via its built-in serial communication pins.

$$d = 0.0085 * L_T \quad (cm/sec) \quad (1)$$

Where "d" is the distance of ultrasonic waves' trip and "LT" is the last count value that is stored in timer1 at the moment of detecting the first reflected echo signal.

3.4. Universal Serial Bus (USB 2.0) interface card

USB 2.0 interface card is an in-lab designed interface card [28],[29]that employs PIC 18F2550 µC [30] to meet requirement of USB 4-wire communication with PCs. We choose employing USB 2.0 standard as TII because USB 2.0 communication has many benefits, such as no extra power supply required, and ease of plugging and playing such USB interface cards with PCs in peer-to-peer (P2P) communication.

The notable benefit of using USB 2.0 is its high speed as compared to other interfaces because data transfer rate supported by it may be up to 480 Mb/sec [28]. USB interface card is implemented and programmed by using MikroC compiler [31] because it has been supported with an USB-HID library and USB descriptor tools. USB 2.0 interface card is shown in figure 7.

There are many roles of PIC 18F2550 µC that should perform. The main role is building a bi-directional and permanent communication pipe with NCAP by using two important identification values: Vendor ID (VID) and Product ID (PID). VID and PID construct a unique identifier for any USB device [28],[32]. This USB 2.0 pipe is used to exchange Data between two end points (i.e. NCAP and USB interface card) by employing two buffers on each end point; Write Buffer (WB) and Read Buffer (RB).

Another role of PIC 18F2550 µC is routing communication messages from its dedicated STIM to NCAP or vice versa. Routing scenario differs according to type of its connected STIM. In case of sensory-based STIM, PIC 18F2550 µC receives sensory data message (SDM) from its connected STIM and extracts only sensory information by discarding start and end characters. After thus, PIC 18F2550 µC composes an USB-PC message which consists of a 1 byte command code and followed by the extracted sensory information as shown in figure 8. The command code is used to enable NCAP or USB 2.0 interface card how to deal with information which is transferred and followed after such command code. The last step is delivering USB-PC message to Write Buffer of USB interface card to be sent consequently to NCAP.

Routing scenario is quite different in case of actuator-based STIM such as CMM. USB interface card receives a PC-USB message from NCAP via its Read Buffer. format of

PC-USB message for CMM, as an example of actuator-based STIM, is shown in figure 9. If the command code of PC-USB message is 0x60, it depicts that USB 2.0 interface card will construct a control message and send it to its dedicated CMM through its built-in serial communication pins (i.e. TX and RX) of PIC 18F2550 µC as shown previously in figure 8. Another role of USB 2.0 interface card is routing the feedback message that will come back from actuator-based STIM (such as CMM) as a reply for this PC-USB message. Feedback routing methodology is similar to steps of sending SDM.

3.5. Sensory data Collector/Command Distributor (SCCD) Model

SCCD is a software model that developed by C# programming language under the .Net 4.0 framework. Using C# and .Net 4.0 framework will also provide one of the most important and value-added functionality such as portability. Portability of SCCD refers to its ability to be installed with different PC-based NCAPs that run .Net 4.0 framework.

Current and prospective aim of presenting SCCD software model to be the main backbone solution for the USB-based network system. In such network type, SCCD can collect sensory data and measurements from diverse sensors. In addition, the same SCCD model can drive different or similar types of actuators alongside with the network of diverse sensors. General software layers/components of SCCD are shown in figure 10. It consists of two main software layers; the first layer includes all software components that collect sensory data and/or transferring commands from/to sensors and actuators respectively. The first layer is actual implementation of SCCD model. The second layer is devoted to process and manipulate sensory data by using Rules-based algorithms and then prepare suitable control commands for any application. The second layer is called Processing and Decision-making Rules (PDR).

In SCCD layer, developer should provide a point-to-point (P2P) communication link with each one of its connected transducers (i.e. sensors or actuators). This P2P link is achieved by developing a Private GUI (PGUI) windows form. Each PGUI windows form is assigned with a clone of a library called USB Generic HID Communication (USB-GHC) version 2.0.0.0 which is developed by S. Inns et. al [29]. USB-GHC is the actual hidden code layer that manages data transfer between SCCD and each one of its connected transducer devices.

GUI components that used with any PGUI should be designed according to the nature of measurements that each sensor can provide to SCCD or the nature of control commands that any actuator can receive and execute. For example, figure 11 shows the design of PGUI that is devoted to display the received measurements of speed from MMS. Similarly, figure 12 shows similar PGUI design for displaying distance measurements of MMD. Different PGUI design is shown in figure 13 which is developed to send control commands that drive a DC motor.

SCCD software model is maturely designed to enable other developers from adding and designing seamlessly new PGUIs or modifying old ones because each PGUI is just a C# windows form. Each PGUI calls its corresponding method in collector/distributor class to perform extraction process of its sensory measurements or composing process of its control commands. In addition, each PGUI of SCCD sends its sensory data to a Main GUI (MGUI) windows form and to PDR layer. The goal of developing MGUI is to refresh connectivity status between SCCD and PGUI of each connected transducer and to display all sensory data for user of such SCCD.

As stated previously, all measurements from all PGUIs are pooled in PDR layer for processing and manipulating them according to predefined rules to determine proper decisions. PDR converts these decisions to its equivalent control commands. Format of each control command differs according to type of each connected actuator to SCCD system. At last, control command is delivered to its dedicated PGUI to be transferred consequently to its dedicated actuator-based STIM.

3.6. Communication between SCCD and STIMs

This section is devoted to show how SCCD communicates with its connected STIMs. Communication between SCCD and each one of its STIM is divided into two general communication scenarios according the type of connected STIM. They are SCCD to Sensor-based STIM communication scenario and SCCD to Actuator-based STIM communication scenario.

Figure 14 depicts and shows communication scenario between SCCD and sensor-based STIM. Sensor-based STIM sends periodically its sensory data message (SDM) to its dedicated USB interface card over consecutive fixed time intervals, called Sending Sensory Data Interval "SSDI". USB interface card saves the received SDM in short-term buffers. These buffers are updated continuously as soon as receiving a new SDM. SCCD sends periodically a request to the required USB interface card over consecutive time intervals, called Sensory Data Request Interval (SDRI). The required USB interface card replies each request from SCCD by sending a USB-PC message which contains sensory data

Communication scenario between SCCD and actuator-based STIM is quite different. SCCD is responsible for the continuous sending of PC-USB message to the USB2.0 interface card of the required actuator-based STIM over consecutive time intervals, called Sending PC-USB Interval (SPUI). In this communication scenario, USB interface card re-constructs the PC-USB message to produce control message. After thus, it delivers the control message which includes required control instruction to the actuator-based STIM. After applying physically the control instruction, the actuator-based STIM should response to thus by sending an acknowledgement to the SCCD through its dedicated USB interface card. Figure 15 shows the methodology of communication between SCCD and actuator-based STIM.

4. Performance Evaluation of USANet

4.1. Experiment Setup of USANet

In order to verify the feasibility of USANet system, modules of MMS, CMM, three MMDs, and five USB 2.0 interface cards. All these modules are connected to a laptop through two USB hubs (i.e. 4-ports USB hubs). Figure 16 shows how these modules are connected to form a practical test-bed to validate such USANet system. The employed laptop, with Pentium i3 core and 2.20 GHZ microprocessor, and 4 GB RAM, serves as the NCAP host for SCCD program.

In these practical experiments, SCCD deals with collecting distance information from three MMD modules, speed from MMS module, and sending control instructions to CMM. Three MMD modules are referred as Top MMD (TMMD), Right MMD (RMMD), and Left MMD (LMMD) as shown in figure 16. Control instructions, which SCCD sends to CMM, revolves around varying duty cycle continually according to distance value that received from TMMD. Varying duty cycle leads to control speed of DC motor that is connected to CMM. For example, if the measured distance of TMMD is less or equal to 40 cm, duty cycle (which equals to 70) should be sent to stop DC motor. In case of distance between 40 to 60 cm, duty cycle should be any value from 70 to 90, and so on.

Pulses count is used within equation 2 to calculate the speed of the tested DC motor. In equation 2, "v" represents the Velocity (i.e. speed) of DC motor. "TP" is the total number of pulses which SCCD receives from MMS with in SSDI "t". "d" is the diameter of spinning disk that is fixed on the output rotor shaft of tested DC motor. "N" represents the number of holes of spinning disk and also it represents number of generated pulses at only one round of this spinning disk.

$$\mathbf{v} = \frac{3.14 * d * T_P}{N * t} \quad \text{(meters / sec)} \qquad (2)$$

4.2. Experimental Results

This section is devoted to shed light on experimental results, obtained from USANet system. There are some parameters that are determined during these experiments. For example, number of holes " N " will be equal to " 100 " and the diameter of spinning disk " d " = 2.7 mm. the time period of SSDI " t " will be ranged from 10 msec to 100 msec. Multiple runs are performed to obtain averaged result and time period for each run is 300 seconds. In addition, all these experiments are carried out on only one DC motor with device number (DMM-F019A0-F01) from KITASHIBA [34].

Two time-based performance metrics are chosen for evaluation because the provision of timely information is crucial in many applications. These time-based metrics are as following:

- Feedback Receiving Time (FRT): it is related to the sum of time that PC-USB message takes to be built by SCCD and consumed time between sending PC-USB message which includes control instructions to CMM until receiving feedback message from it.
- Sensory Data Message Time (SDMT): it represents elapsed time between sending sensory data request from SCCD until receiving sensory data message SDM.

Figure 17 shows results of FRT at different SPUI intervals for USANet that consists of five STIMs. SPUI intervals are ranged from 5 msec to 100 msec. SPUI represents time interval through which PC-USB message is sent from SCCD to actuator-based STIM (i.e. CMM). SCCD sends PC-USB message periodically over consecutive SPUI intervals to supply CMM STIM with required duty cycle (i.e. speed) and direction over each SPUI interval.

Figure 17 helps in deciding what is the most suitable SPUI interval? This suitable SPUI interval should be set permanently as the timer value of CMM's private GUI in SCCD program before running it. There are two selection scenarios to select the suitable SPUI interval.

- The first scenario related to select SPUI interval that achieves minimum FRT value which is called "Best SPUI interval".

- The second scenario is related to select SPUI interval that causes maximum FRT value, which is called "Worst SPUI interval".

It is observed from figure 17, that the best SPUI interval is 5 msec because it achieves FRT value equals to 23.51 msec, while the worst SPUI interval is 90 msec because its corresponding FRT value equals to 26.73 msec.

The difference between minimum and maximum FRT values equals 3.22 msec. this difference is quite small which persuades us to prefer applying worst SPUI interval which equals to 90 msec. Figure 17 illustrates that it is illogical to use 5 msec as SPUI interval because it means that PC-USB message will be sent to CMM every 5 msec. However thus, the Feedback Message will be received after 23.51 msec after the moment of finishing transmission of its PC-USB message. Through 23.51 msec, SCCD can send four PC-USB messages and will be ready to send the fifth PC-USB message. So, this received Feedback Message replies which one of these four PC-USB messages. The same analysis can be applied for 10, and 20 msec SPUI intervals. The previous analysis confirms our decision to apply 90 msec as the most suitable SPUI interval.

Figure 18 shows results of Speed Message Time (SMT), Right Distance Message Time (RDMT), Left Distance Message Time (LDMT), and Top Distance Message Time (TDMT) at different SDRI intervals. Figure 18 elucidates that worst SDRI interval is also 90 msec because maximum values of SMT, RDMT, LDMT, and TDMT are 18.18, 18.11, 18.12, and 18.24 msec respectively. Also, it is observed that best SDRI interval is 20 msec because SMT, RDMT, LDMT, and TDMT achieve minimum value which equal 13.5, 13.26, 13.39, and 13.64 respectively.

Values of SMT, RDMT, LDMT, and TDMT are tightly close to each other's overall SDRI intervals. Average of them at each SDRI will be called "Sensory Data Message Time (SDMT)". It is found that maximum SDMT is 18.16 msec, while minimum SDMT is 13.45 msec. other average results of SDMTs fluctuates slightly between their minimum and maximum SDMT values. The difference, between such maximum and minimum values, is 4.71 msec. Worst time response cases should be taken into consideration for such USANet systems to protect them from un-normal behavior in real applications.

In USANet system, it is noticed that maximum SDMT response time will be faster than maximum FRT response time. The reason behind thus is which firmware part replies SCCD. It is stated previously in setion 3.6 that TII (i.e. USB 2.0 interface card), of any sensor-based STIM, is responsible of replying to any sensory data request message. The sensory data request message is transmitted from SCCD every SDRI interval. It replies with last saved sensory data.

On the other side, actuator-based STIM has the responsibility to reply SCCD when it sends a PC-USB message. Actuator-based STIM replies with a feedback message. TII, of an actuator-based STIM, is just forwarding node between SCCD (i.e. NCAP) and actuator-based STIM.

5. Conclusion

This paper discusses and presents the development of a new and open platform for USB-based Sensors/Actuators Network (USANet). USANet, as a special research topic of Sensors/Actuators Network (SANet), was regulated by the IEEE 1451 standards. They emphasize that any SANet system should consists of three main components. These components are Smart Transducer Interface Module (STIM), Transducer Independent Interface (TII), and Network Capable Application Processor (NCAP).

In our USANet system, one Measurement Module of Speed (MMS), and three Measurement Modules of Distance (MMDs) are employed as sensor-based STIMs. In addition, Controller Module for Motor (CMM) represents an actuator-based STIM. NCAP of such USANet is a laptop host which is used to run a program called "SCCD" (Sensory data Collector/Command Distributor). SCCD is capable of collecting sensory data from diverse sensors in such USB-based network. Also, SCCD can control and drive different or similar types of actuators alongside with sensors in such USB-based network. TII of our proposed USANet is an in-lab designed USB 2.0 interface card. This USB 2.0 interface card is the reason behind calling our SANet system as "USANet". It is employed to provide and build a P2P and permanent communication link between NCAP and each one of its STIMs.

Feasibility of USANet is verified and evaluated via two performance metrics by multiple practical and experimental runs. These two performance metrics are Feedback Receiving Time (FRT) and Sensory Data Message Time (SDMT). Maximum FRT value equals 26.73 msec and it proves that CMM part of USANet achieves a favorable time-based performance. SDMT represents the average time of all sensory data messages that are received individually and in parallel from MMS and three MMDs by SCCD of USANet system. Also, SDMT value of USANet confirms that four sensor-based STIMs, as a whole, consumes considerable time intervals because each one response to SCCD with sending its sensory data in a period of time (i.e. SDMT) equals18.16 msec.

References

[1] J. Jin, and B. Wang, " Design of Automatic Meteorological Data Acquisition System Based on ARM and CAN Bus", International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), pp. 989 – 992, 2010, Changsha.

[2] NIST, IEEE standards 1451 family, http://ieee1451.nist.gov/

[3] E.Y. Song, and K. Lee, "Understanding IEEE 1451-Networked smart transducer interface standard - What is a smart transducer? ", IEEE Magazine of Instrumentation & Measurement, Vol. 11, issue. 2, pp. 11-17, 2008.

[4] A. H. Elmahalawy, M. A. Ali, H. M. Harb, "Trends and Differences of Applying Intelligence to an Agent", in Journal of Telecommunication and Information Technology (JTIT), Vol. 4, pp. 119-123, 2011.

[5] N. Xiong, and P. Svensson , "Multi- sensor management for information fusion: issues and approaches", Journal of Information Fusion, Vol. 3, issue. 2, pp. 163-186,2002.

[6] T. P. Banerjee, and S. Das ,"Multi- sensor data fusion using support vector machine for motor fault detection", Journal of Information Science, Vol. 217, pp. 96-107, 2012.

[7] A. Flammini, P. Ferrari, D. Marioli, E. Sisinni, and A. Taroni, "Wired and wireless sensor networks for industrial applications ", Microelectronics Journal, Vol. 4, issue. 9, pp. 1322-1336, 2009.

[8] M. Collett, T.J. Esward, P.M. Harris, C.E. Matthews, and I.M. Smith, "Simulating distributed measurement networks in which sensors may be faulty, noisy and interdependent: A software tool for sensor network design, data fusion and uncertainty evaluation", Journal of Measurements, Vol. 46, issue. 8, pp. 2647-2660, 2013.

[9] W. Li, X. Zhang, and H. Li, "Co-simulation platforms for co-design of networked control systems: An overview ", Journal of Control Engineering Practice, Vol. 23, pp. 44-56, 2014.

[10] K. E. Årzén, A. Bicchi, G. Dini, S. Hailes, K. H. Johansson, J. Lygeros, and A. Tzes, "A Component-Based Approach to the Design of Networked Control Systems", European Journal of Control, Vol. 13, pp. 261-279, 2007.

[11] G. Song, A. Song, and W. Huang, "Distributed measurement system based on networked smart sensors with standardized interfaces", Journal of sensors and actuators A: Physical, Vol. 120, issue. 1, pp. 147 – 153, 2005, DOI: 10.1016/j.sna.2004.

[12] S. Chunshan, S. Fei, W. Zhongcheng, L. Jiarong, ""Application of session initiation protocol to networked sensor interfaces", journal of computer standards & interfaces, Vol. 31, issue. 2 pp. 454 – 457, 2009, DOI: 10-1016/j.csi.2008.05.006.

[13] J. Rosenberg, et al., "SIP: Session Initiation Protocol", June 2002 RFC 3261, www.ietf.org.

[14] S. Wu, K. Wu, J. Liang, Z. Li, and P. Yang, "Design of Remote Environment Control system of intelligent network henhouse based on ARM9", Journal of Procedia Engineering, Vol. 15, pp. 1056 – 1060, 2011, DOI: 10.1016/j.proeng.2011.08.195.

[15] L. Wang, M. Guoli, and J. Wang, "Implementing for Networked Motion Control system with large-capacity data acquisition", Journal of Procedia Engineering, Vol. 15, pp. 28 - 32, 2011, DOI: 10.1016/j.proeng.2011.08.007.

[16] http://www.ethercat.org/

[17] J. Datta, S. Chowdhuri, J. Bera, and G. Sarkar, "Remote monitoring of different electrical parameters of multi-machine system using PC", Journal of measurements, Vol. 45, issue. 1, pp. 118 – 125, 2012, DOI: 10.1016/j.measurement.2011.09.009.

[18] A. Kumar, H. Kim, and G. P. Hancke, "Environmental monitoring systems: A Review ", IEEE Sensors Journal, Vol. 13, issue. 4, pp. 1329 – 1339, 2013, DOI: 10.1109/JSEN.2012.2233469.

[19] Avago Technologies, Small Optical Encoder Module (HEDS-9780#A54), http://www.avagotech.com/docs/AV02-1747EN

[20] Microchip Technology Inc., http://www.microchip.com/ TechDoc.aspx? type= datasheet

[21] STMicroelectronics, http://www.st.com/st-webui/static/active/en/resources/ technical/document/datasheet/CD00000240.pdf

[22] Microchip Technology Inc. , http://ww1.microchip.com/downloads/en/ DevicesDoc /39564c.pdf

[23] T. J. Licznerski, J. Jaroński, and D. Kosz, "Ultrasonic system for accurate distance measurement in the air", Journal of Ultrasonics, Vol. 51, issue. 8, pp. 960 – 965, 2011, DOI: 10.1016/j.ultras.2011.05.011.

[24] C. J. Wu, and C. C. Tasi, "Localization of an autonomous mobile robot based on ultrasonic sensory information", Journal of Intelligent and Robotic system, Vol. 30, issue. 3, pp. 267 – 277, 2001, DOI: 10.1023/A:1008154910876

[25] S. Choi, T. Jin, and J. Lee, "Obstacle avoidance algorithms for visual navigation using ultrasonic sensors and a CCD camera", on the 7th international symposium on Artificial Life and Robotics, pp. 132 – 135. 2003, DOI: 10.1007/s10015-002-0232-6.

[26] L. P. Zhang and Q. Yu, "The Ultrasonic Measurement for Object Vibration" on International Conference of Computational Science and its Application (ICCSA), pp. 699 – 707, 2009, DOI: 10.1007/978-3-642-02454-2_53.

[27] Jameco vlauePro company, http://www.jameco.com/Jameco/Products/ProdDS/ 139492.pdf

[28] A. Elsaqaa, A. EL-Ganzoury, K. Mohammed, H. Mahmoud, and A. El-Sobky, "Automatic Traffic Ticket System (ATTS), " Technical Report of Final Year project to Dept. of Computer Engineering & Science, Faculty of Electronic Engineering, Menofia University, 2011.

[29] Simon Inns, http://www.waitingforfriday.com/index.php/ Open_Source

_Framwork_for_USB_Generic_HID_devices_based_on_the_PIC18F_and_Windows.

[30] National Instruments , upgrading your system for virtual TEDs (online), available: http://www.ni.com/

[31] Microchip Technology Inc. , http://ww1.microchip.com/downloads/en/ DevicesDoc /39632c.pdf

[32] MikroElectronica, http://www.mikroe.com/mikroc/pic

[33] J. Axelson, "USB Complete: The Developers Guide, " Lackview research LLC, 4th edition, 2009.

[34] KITASHIBA Electric Co. Ltd., Japan, http://www.kitashiba.co.jp/english/Okawa, Rick (2003), “The model code system in the U.S.”, Saudi Building Code Workshop, Saudi Building Code Committee, Riyadh, Saudi Arabia.

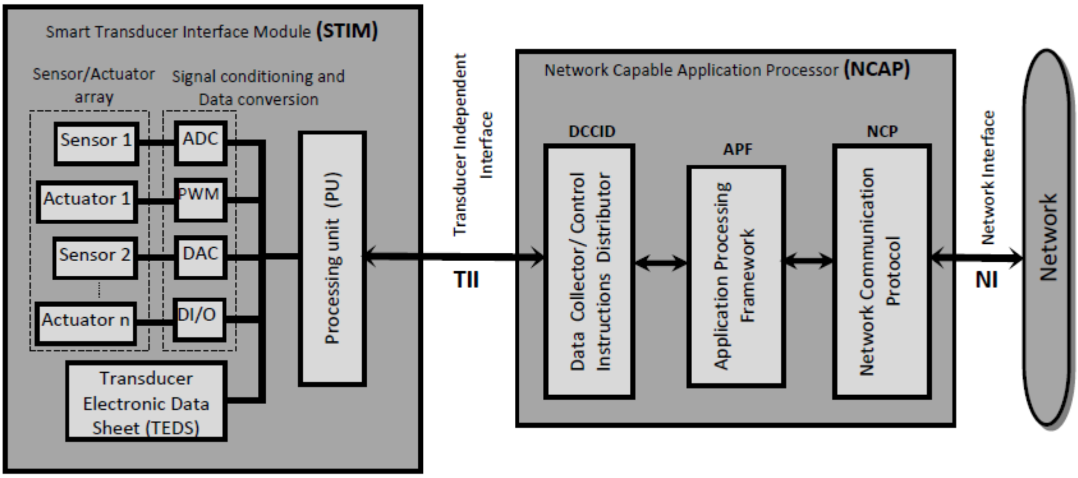

Fig.1 Key components of IEEE 1451 standard

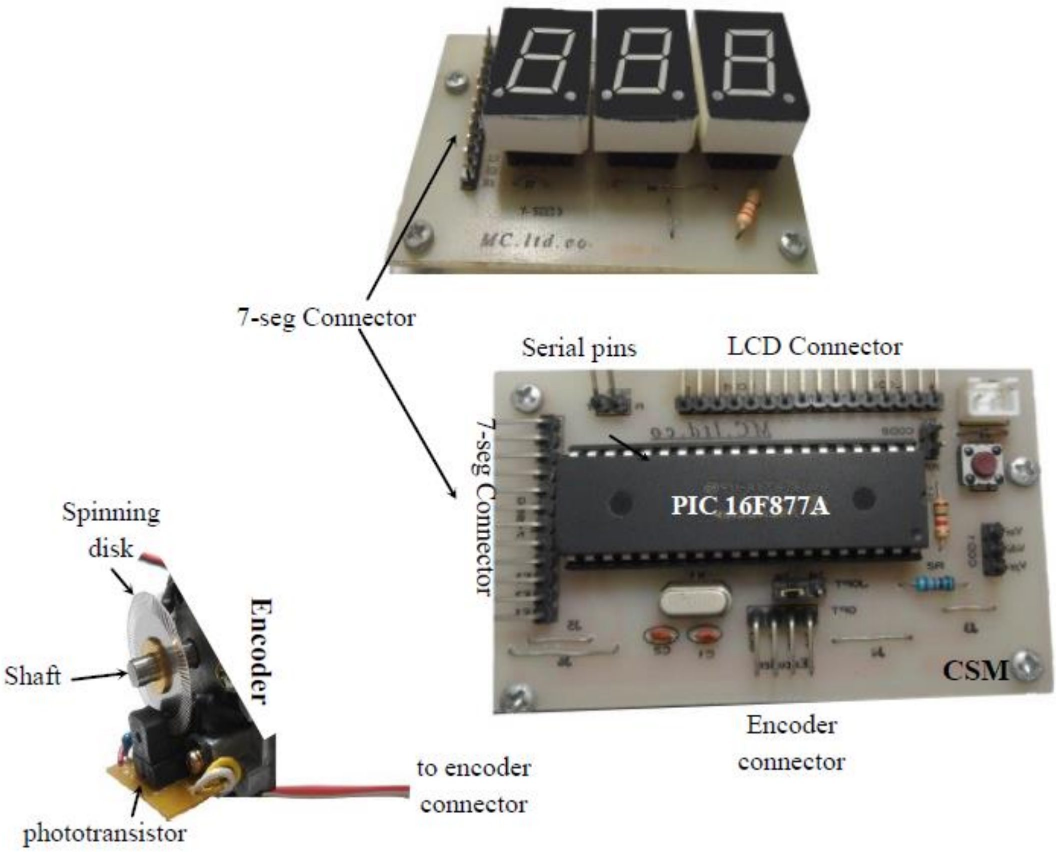

Fig.2 Practical three parts (sub-circuits) of MMS

Start	Sensory Data	End
1 byte	20 byte	1 byte

Fig.3 Format of Sensory Data Message (SDM)

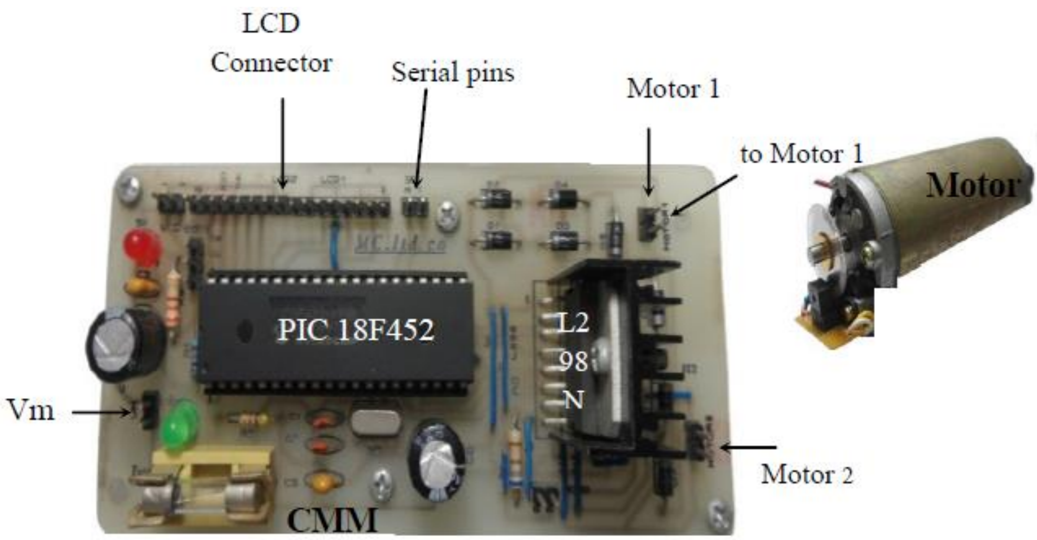

Fig.4 Practical circuit of CMM

Start	Command	New Direction	New Duty Cycle	End
1 byte	1 byte	1 byte	1 byte	1 byte

Fig.5 Format of Control Message

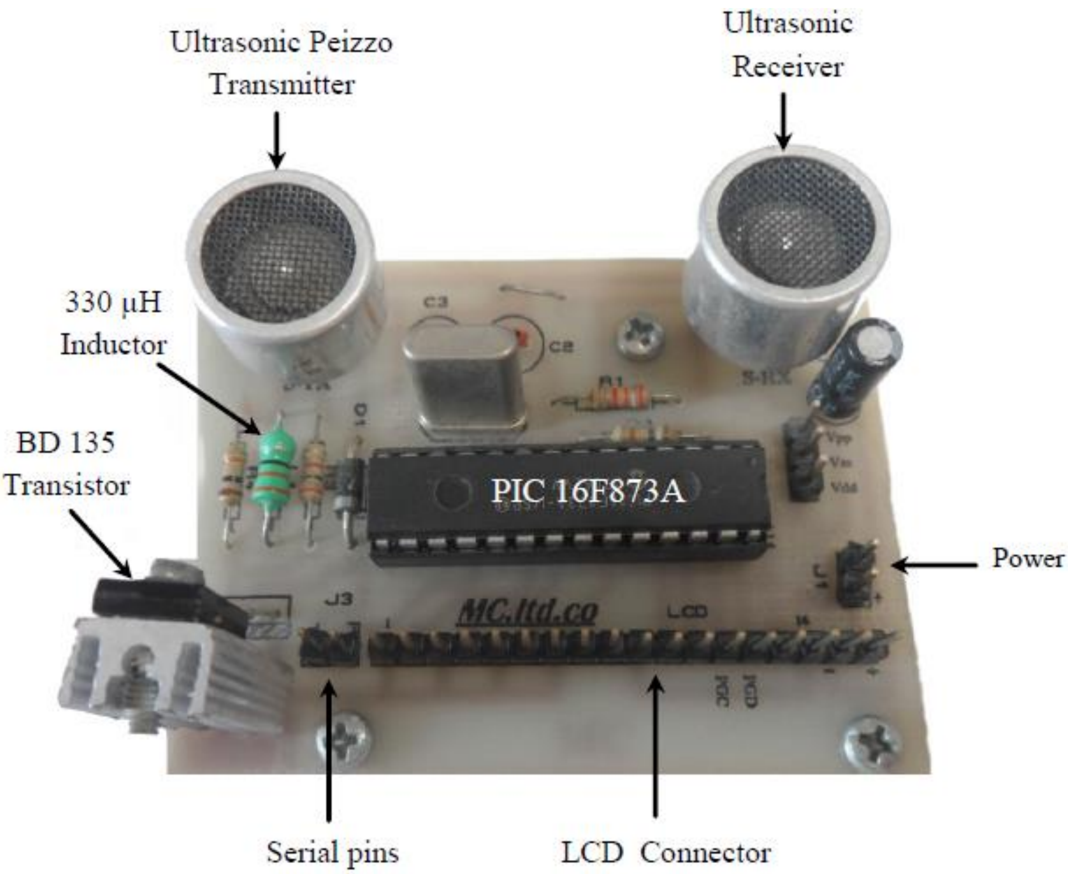

Fig.6 Practical circuit of MMD

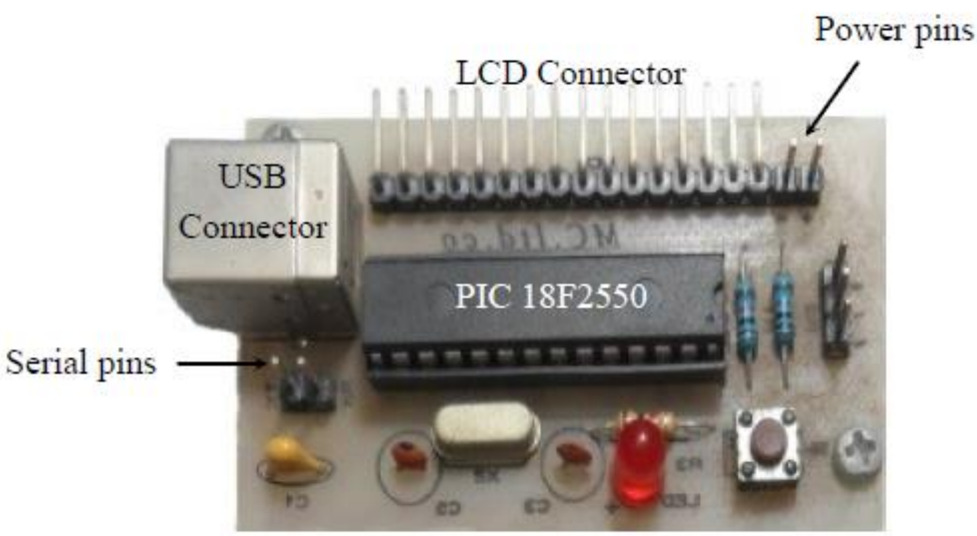

Fig.7 Practical circuit of USB 2.0 interface Card

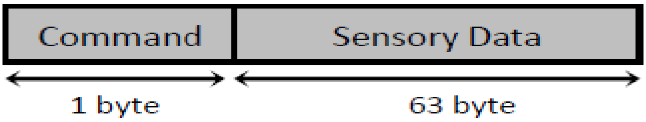

Fig.8 Format of USB-PC message

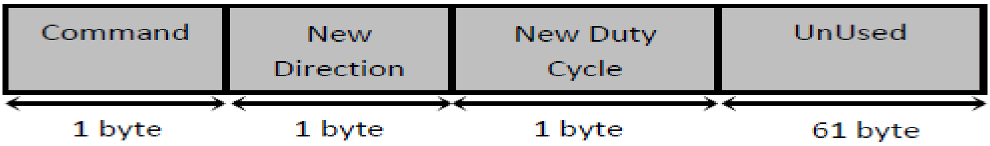

Fig.9 Format of PC-USB message

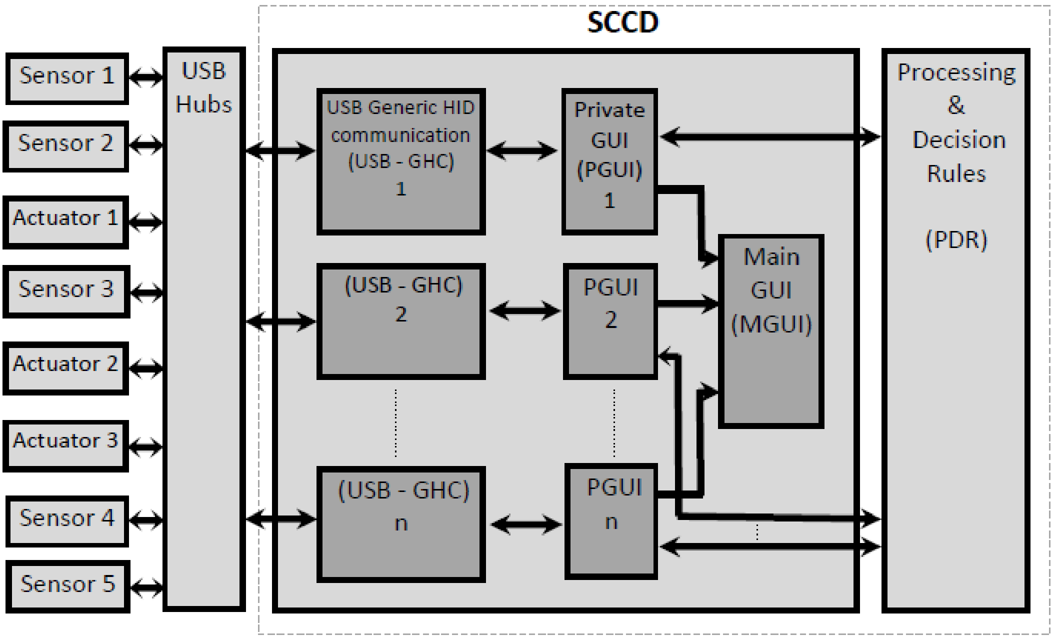

Fig.10 General Software components of SCCD

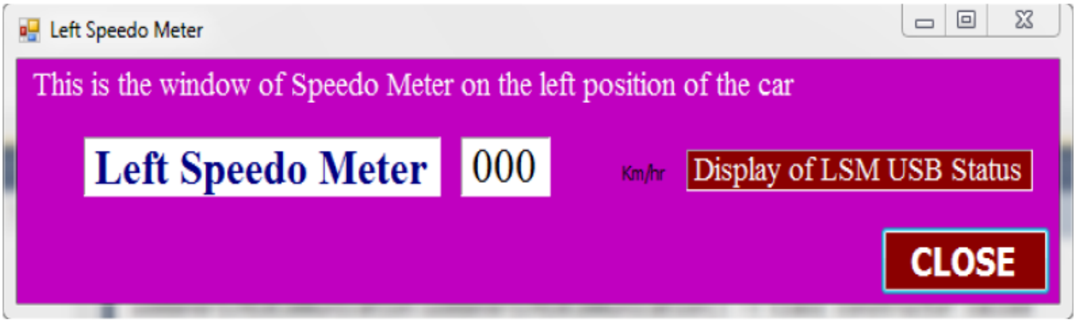

Fig.11 PGUI of MMS transducer device

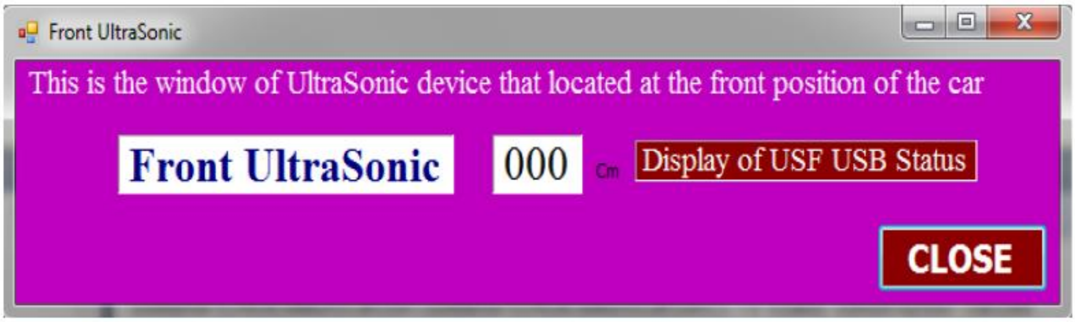

Fig.12 PGUI of MMD transducer device

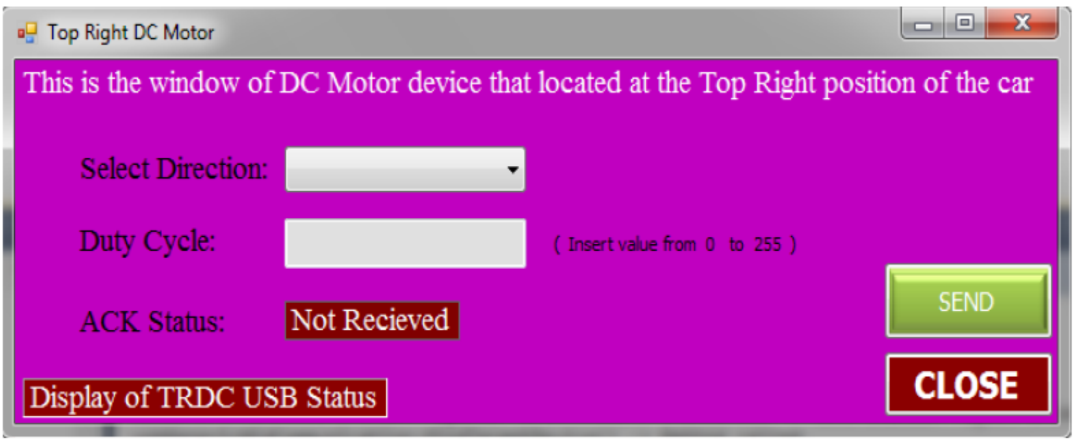

Fig.13 PGUI to control DC motor

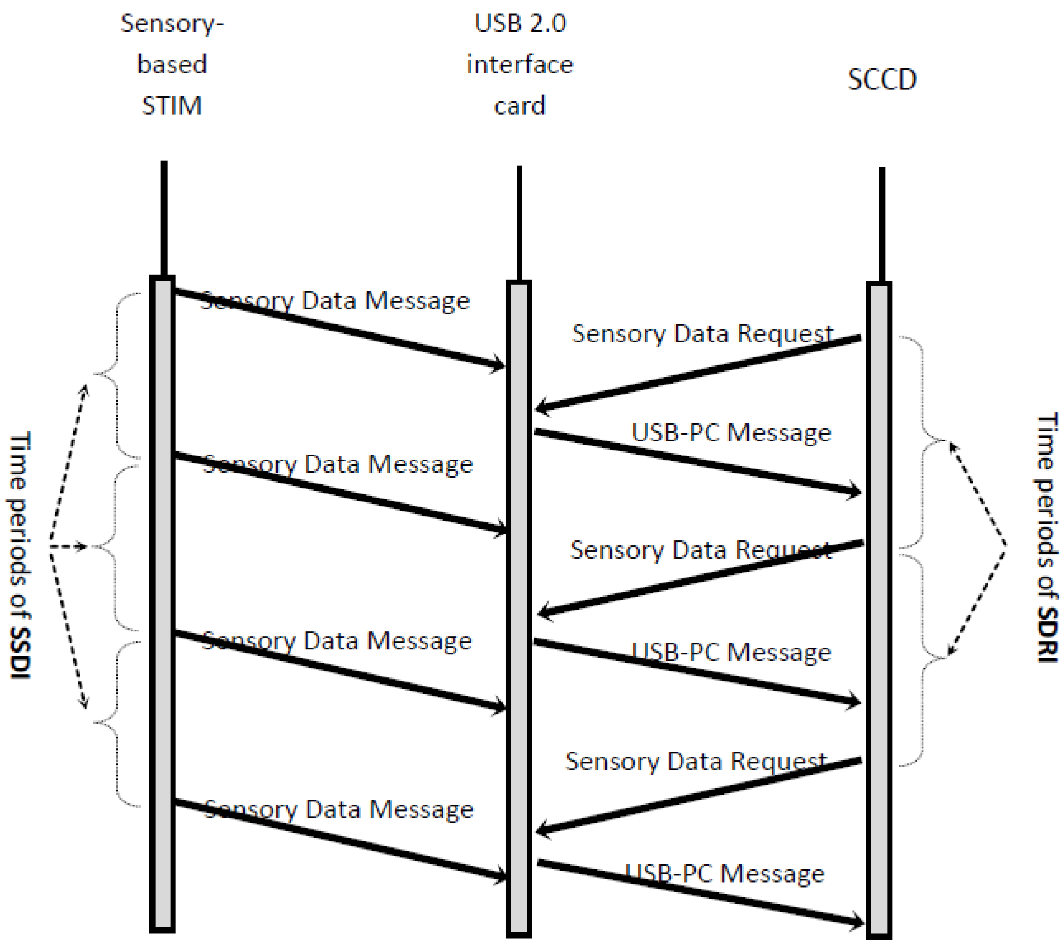

Fig.14 Communication scenario between SCCD and sensor-based STIM

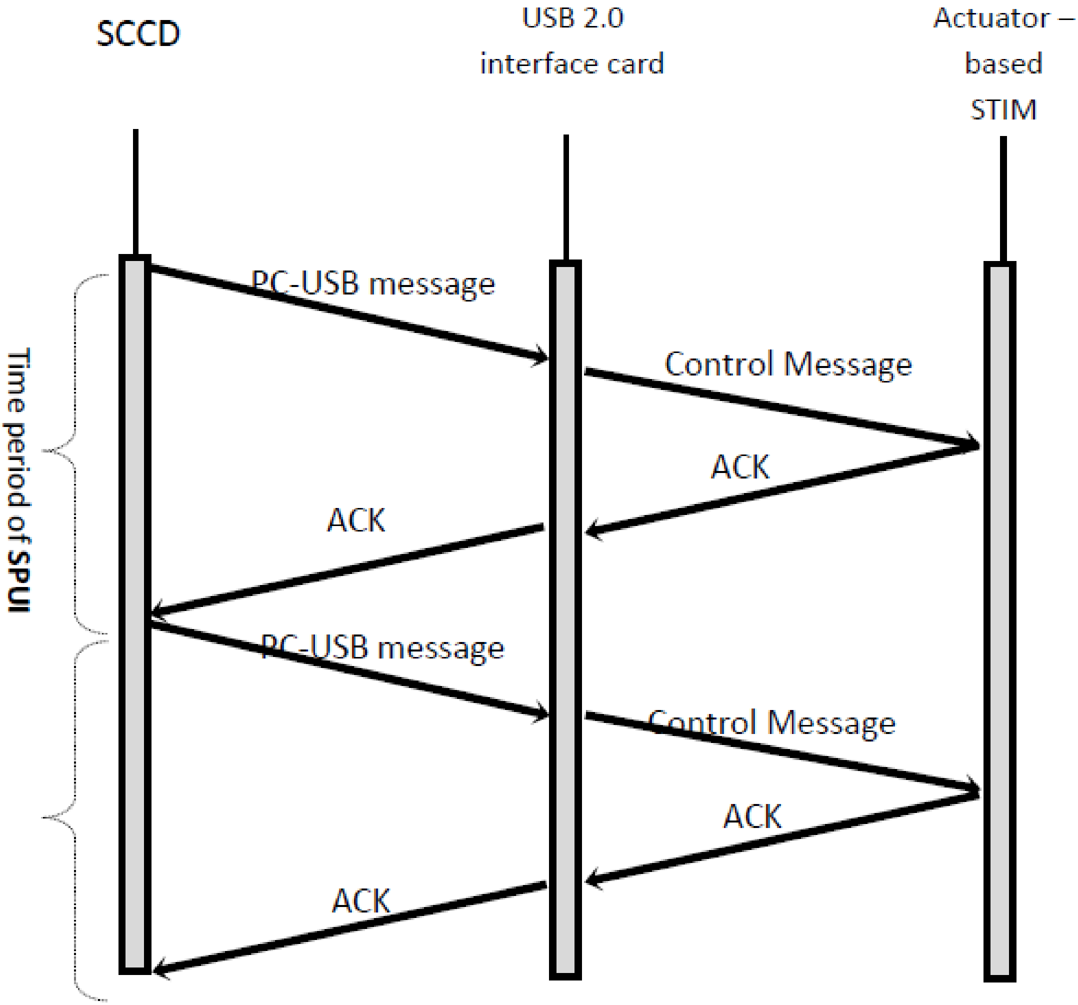

Fig.15 Communication scenario between SCCD and actuator-based STIM

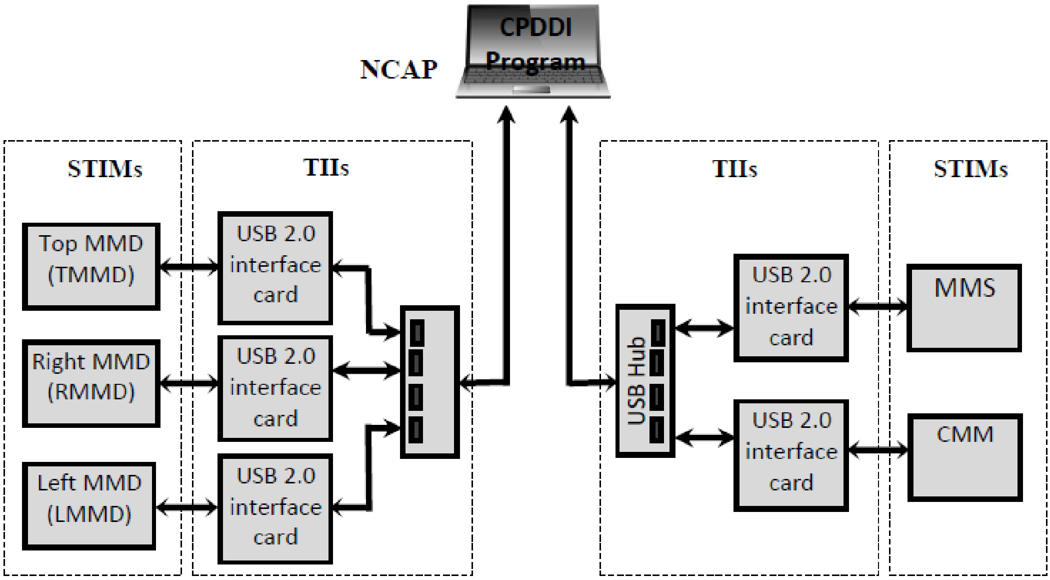

Fig.16 Practical test-bed for USANet system

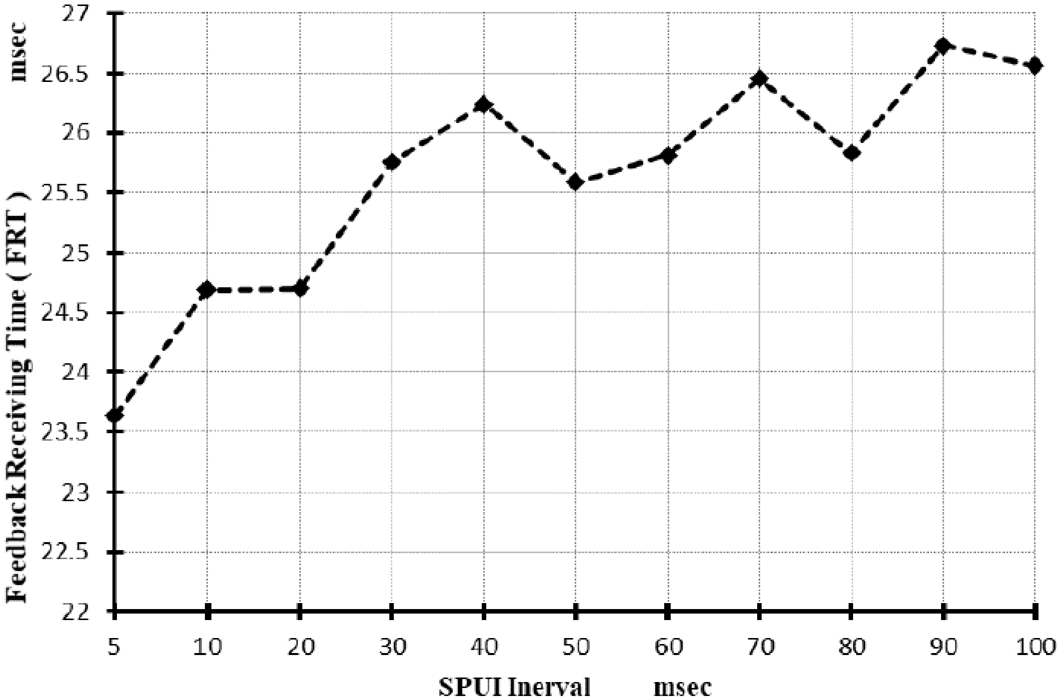

Fig.17 Practical results of Feedback Receiving Time (FRT) for USANet with 5 STIMs at different SPUI intervals

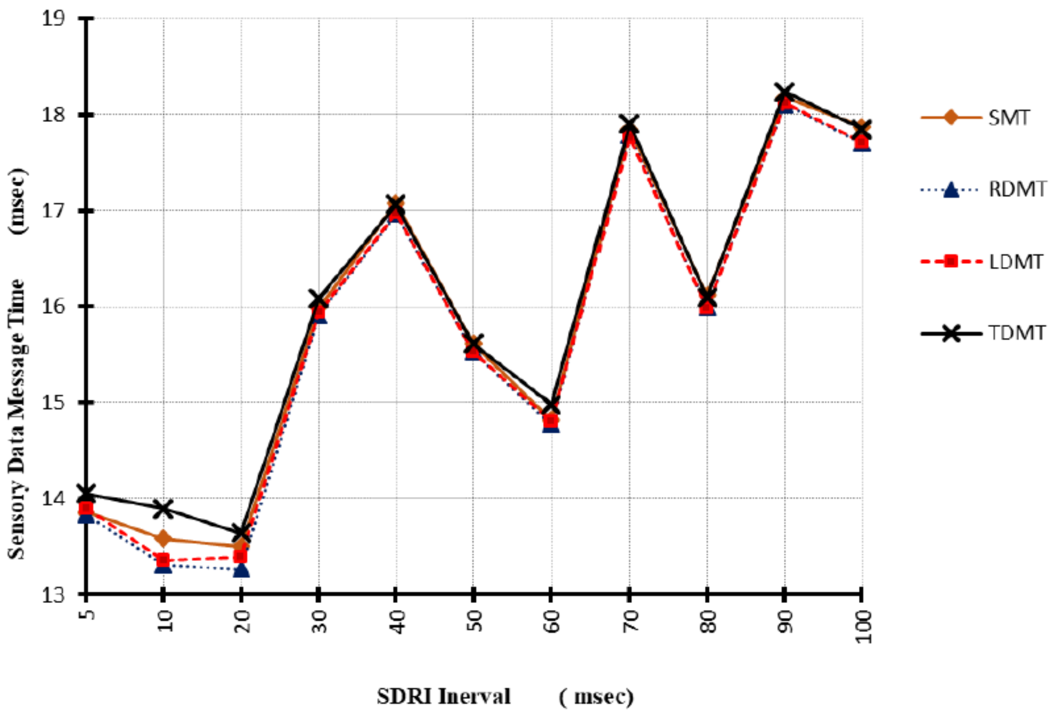

Fig.18 Practical results of SMT, RDMT, LDMT, and TDMT for USANet with 5 STIMs at different SDRI intervals