Performance Characterization of a Wireless Instrumentation Bus

Luigi Ferrigno, *Member, IEEE*, Vincenzo Paciello, *Member, IEEE*, and Antonio Pietrosanto, *Member, IEEE*

***Abstract*—The architecture and operation of a wireless instrumentation bus based on the Bluetooth standard are presented together with a detailed characterization. The bus, via suitable wireless interface boards, allows measurement instruments equipped with standard RS232, IEEE488, universal serial bus, and local area network interface ports to be simply connected within the wireless area without any firmware modification. In addition, a comparison with traditional wired interface buses is carried out in terms of communication reliability and throughput with reference to some basic operating modes. Finally, the results of distance tests and a detailed evaluation of power consumption for each wireless interface module are reported. This will enable measurement system designers to evaluate the convenience of the solution for their application requirements.**

***Index Terms*—Bluetooth (BT), collaborative networks, instrument interface, low power consumption, plug-and-play devices, real-time measurements, smart sensors.**

I. INTRODUCTION

IN A world where all electronic devices are drifting toward wireless connections [1]–[3], only measurement instruments are still forced by interface cables to be near to set up a measurement station. These cables constitute obstacles that reduce safety and free space. Furthermore, changes to the physical configuration of automatic measurement stations cannot be made without the need to interrupt service and update software.

These considerations hold for instruments featured with IEEE488 or RS232 ports [4]–[9], for VXI systems, and for newer instruments featured with universal serial bus (USB) [10] or Firewire ports.

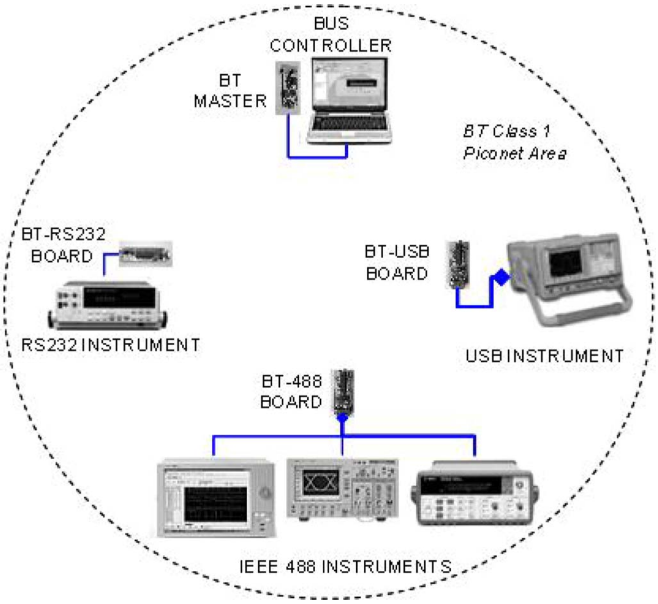

Fig. 1. Example of wireless connection allowed by the proposed wireless interface bus.

To satisfy the need for short-range (10–100-m) low-cost wireless connections for general-purpose measurement instrumentation [11], [12], the authors have designed a wireless instrumentation interface [13], [14] based on the Bluetooth (BT) [15]–[17] communication channel (Fig. 1). This interface system comprises a star bus where a BT master module is plugged into the PC controller and suitable BT slave modules designed to fit either the RS232 or the IEEE488 port of instrumentation. Each standalone instrument, whose RS232 or IEEE488 interface port is plugged into the aforementioned interface module, can connect with seven devices in the Piconet and can exchange data and device commands with a PC hosting a standard BT master module (a commercial BT dongle for a serial or PCMCIA PC port can be used) without any hardware or firmware modifications (see Fig. 2).

The number of applications is increasing where, even if distances between instruments are constrained in the short range (100 m), the controller needs to be elsewhere at a great distance. Moreover, more and more instruments today have a USB port, which often replaces the traditional RS232 port due to its higher maximum bit rate and to the possibility offered by a simple and cheap USB hub of making multiple connections [7], [8]. Hence, a new bus controller of the BT-based interface bus allowing the wireless instrumentation bus to be controlled via remote Ethernet connection was presented in [18], together with a new wireless interface board capable of connecting USB measurement instruments to the BT Piconet (see Fig. 1).

The realized wireless interface devices are economical, easy to install, and instrument independent, because their hardware and software do not depend in any way on the devices they are fitted to. The layout of the implemented wired-to-wireless interfaces is given in Fig. 3.

However, while the advantages characterizing the wireless interface are clear, the same cannot be said about the drawbacks or limitations arising from this solution. Parameters such as

---

Manuscript received March 20, 2009; revised January 2, 2010; accepted January 3, 2010. Date of publication September 23, 2010; date of current version November 10, 2010.

L. Ferrigno is with the Dipartimento di Automazione, Elettromagnetismo, Ingegneria dell'Informazione e Matematica Industriale, University of Cassino, 03043 Cassino, Italy (e-mail ferrigno@unicas.it).

V. Paciello and A. Pietrosanto are with the Department of Information Engineering and Applied Mathematics (DIIIE), University of Salerno, 84084 Fisciano, Italy (e-mail: vpaciello@unisa.it; apietrosanto@unisa.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIM.2010.2047307

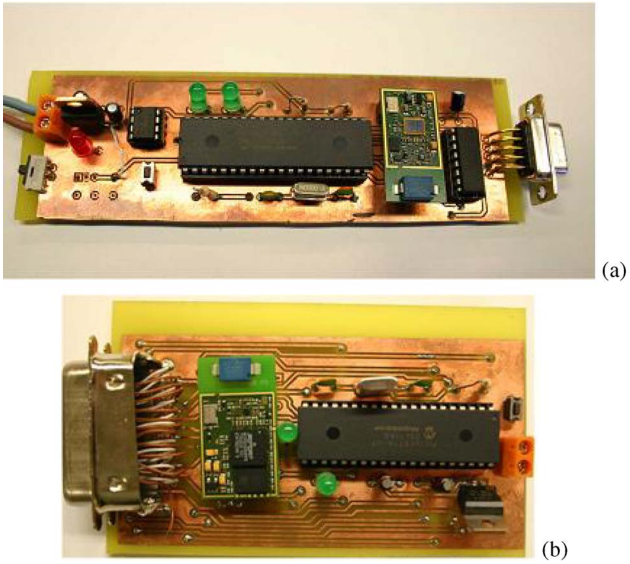

Fig. 2. Photographs of the realized (a) BT–RS232 and (b) BT–IEEE488 interfaces.

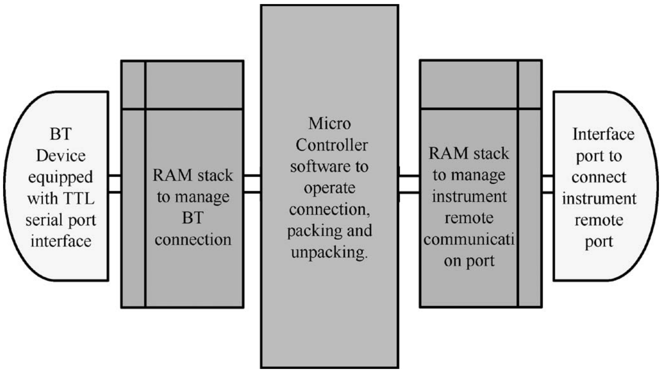

Fig. 3. Block diagram of the architecture of each wired to wireless interface board.

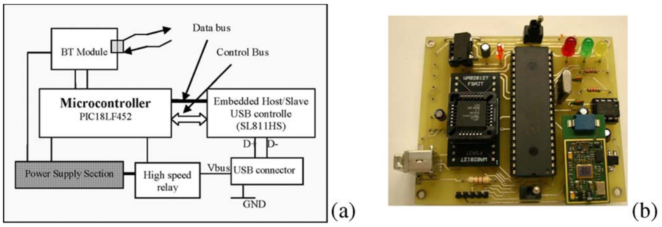

Fig. 4. (a) Block diagram and (b) photograph of the new BT–USB prototype.

the influence of obstacles on the coverage area, the power consumption of the interface module, communication reliability, and throughput need to be accurately evaluated to decide whether the wireless solution is convenient for a particular application or not.

In this paper, after a detailed description of both the BT–USB and BT–Ethernet boards, the complete characterization of the BT-based wireless interface bus is given. A comparison with traditional wired solutions is carried out in terms of communication reliability and throughput with reference to a number of basic operation modes. Furthermore, the results of distance power consumption tests carried out for each wireless interface module are reported with the aim of producing a complete performance characterization of the proposed interface bus.

II. USB INTERFACE MODULE

The USB standard [10], which was designed to allow high-speed connection between dummy peripherals and PCs, has very quickly become widespread. After solving a number hardware and software problems with the introduction of the USB 2.0 standard, most peripherals for connection to PCs are nowadays equipped with USB ports. In addition, with the recent release of the new "USB On-The-Go (OTG)" standard, the USB port has enlarged its frontiers. The previous requirement for a PC to act as a USB host device is no longer the case: USB devices can now be interconnected without the host PC. The USB standard is therefore the undisputed leader thanks to its high speed, reliability, and flexibility. The USB characteristics also match very well with those required by wireless transmission devices. BT USB dongles, Wi-Fi USB dongles, and others are widespread on the market. These devices are designed to be plugged into PCs, handhelds, or phones and allow wireless connections characterized by good throughput and reliability at a very low cost.

In this scenario, the introduction of a wireless BT interface for USB instrumentation that implements the full host USB protocol stack is still innovative since it overcomes some known limitations of the OTG and dongle connections designed to connect only devices produced by the same manufacturer and skipping a very important portion of the host USB handshake.

In particular, the proposed wireless interface for instrumentation: 1) allows hot plug-and-play connections: 2) supplies power to connected USB devices; 3) identifies low-speed or high-speed devices; and 4) assigns channels, pipes, and endpoints as a full-function USB host.

In the following, a description of the interface hardware, software, and functional characteristics is provided.

A photograph of the realized prototype is shown in Fig. 4(b)

1) Hardware of the USB Interface Module: As shown in Fig. 4(a), the realized hardware comprises five fundamental parts.

1) *Embedded host/slave USB controller*: The USB embedded host/slave controller is based on an SL811HS device produced by Cypress. This device can operate as a single USB host or slave under software control with automatic detection of either low- or full-speed devices. It supports both full-speed (12-Mb/s) and low-speed (1.5-Mb/s) USB 1.1 transfer modes. The data transfer to/from the external microcontroller is provided by an 8-bit bidirectional input–output data port. Power consumption can be reduced by a suitable choice of suspend/resume, wake-up, and low-power modes.
2) *Microcontroller*: The microcontroller module is based on a Microchip PIC18LF452 low-power 40-MHz microcontroller, whose main characteristics are 32-kB program memory, 1536-B RAM, one universal synchronous/asynchronous receiver/transmitter (USART) serial port, and the I2C bus.
3) *BT transceiver*: The BT transceiver is based on a class-1 Bluetronics ICM201 device. This is a 1.1 compliant

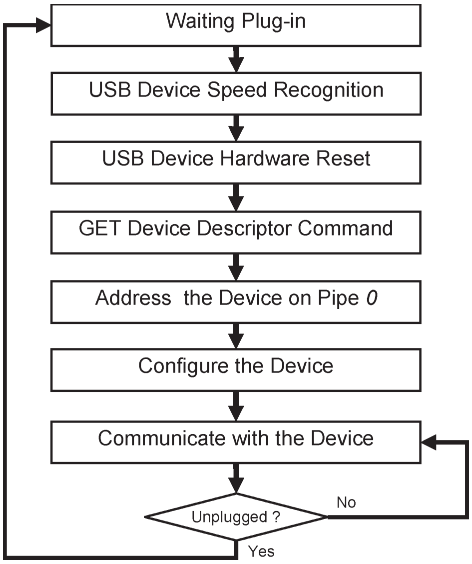

Fig. 5. Block diagram of the BT–USB board software.

class-1 master/slave module with a maximum allowed distance of 300 m. A transistor–transistor logic (TTL)-compatible RS232 communication channel assures communication with the TTL USART bus of the microcontroller module. The selected module requires a 3.3-V power supply and has a power consumption of about 400 mW during the connection phase.

4) *High-speed relay*: A high-speed relay is necessary to turn the USB bus on and off during the recognition and reset phases. This procedure is required by the USB standard.
5) *Power supply section*: The power supply section comprises a 9-V battery package and a number of voltage regulators capable of providing the 5 and 3.3 V required by the USB standard and the BT device.

2) Software: As reported in Fig. 3, the general architecture of each developed interface board operates as a double-layer interface. In particular, the BT–USB interface board manages all the steps required for a slave-side BT wireless connection and operates as a USB host. The required steps for a master–slave BT connection have already been discussed in [11] and [12]; details about the implemented USB host functions are given here. A block diagram of the implemented software is reported in Fig. 5 and can be described as follows. Once the interface module is plugged into the USB port of the instrument, its first step is the recognition of the USB device speed. This step is very important since low-speed and high-speed devices use complementary data coding in the non-return-to-zero inverted code. It is worth remembering that for high-speed devices the baud rate is 11 Mb/s, whereas for low-speed ones, it is 1.5 Mb/s. Once the USB bus rate has been fixed, the USB host must reset the USB interface of the measurement instrument to make it able to accept interface commands on the zero pipe at the zero address (called “control pipe”). To assign a valid address to the connected device and to collect some information that each USB device brings in its device descriptor, the proposed wired-to-wireless USB interface, which acts as a host, uses a suitable command called Get_Device_Descriptor. This command allows the host to know the interface characteristics of the USB device, such as the number of pipes, the number of interfaces, and the type of communication. Then, in a successive addressing phase, the host assigns an address to the USB device, which, hereinafter, will respond only to the new address (“Device addressed”). Finally, the USB interface chooses a possible configuration from the Device_Descriptor and configures the device. After this last phase, the USB interface of the measurement instrument is ready to accept device messages. Starting from this point, the realized prototype operates, as mentioned above, as a double-layer interface. In particular, the device messages coming from the host are unpacked from the USB header and directed to the measurement instrument, while an encapsulation procedure is made for the data transfer on the other side.

III. IMPROVED FEATURES OF THE REALIZED INSTRUMENTATION BUS

The realized instrumentation bus showed two implicit limitations that might restrict its use in many application cases.

1) The PC controller was constrained within the Piconet area.
2) Spatial movements of the PC controller were not possible, since the software realized for the controller had to be implemented on desktop or notebook PCs.

As far as point 1 is concerned, the PC controller cannot be distant from measurement instruments or wireless sensors by more than 10–100 m. These distances could not be enough in applications such as measurements in dangerous or hostile environments, telemetry, measurements on moving objects, and mobile agent networks, which require a remote controller located very far from the Piconet. In these cases, measurement instruments and sensors are located in a low- or medium-coverage operating area, whereas the PC controller, which is placed very far from the wireless sensor network, queries the sensors through an Internet connection. As for point 2, a mobile controller could be very useful to dynamically query more networks. Applications can be found in the environmental monitoring of urban areas, river monitoring, and so on. In these cases, a moving vehicle such as a car or a bus may accommodate a controller that, when located in the Piconet area, can download all the required information.

To satisfy these requirements, the wireless bus controller has suitably been enhanced by setting up both an Ethernet-to-BT controller and a handheld version of the software operating on the PC.

In the following, the hardware and software solutions adopted to realize 1) the remote Ethernet-to-BT controller of the instrumentation bus and 2) a Pocket PC-based handheld version of the instrumentation controller are described in detail.

A. Ethernet-to-BT Controller

The Ethernet-to-BT controller has been realized by interfacing the Bluetronics ICM201 BT master device with an integrated TIBBO EM202 RS232-to-LAN converter (Fig. 6). The EM202 device is a double interface that allows the Ethernet connection of a serial device. It features a 100BaseT Ethernet

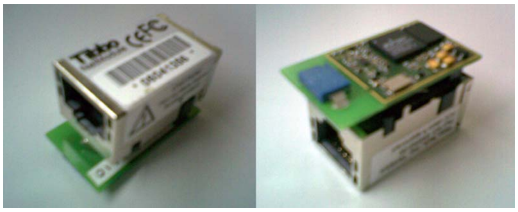

Fig. 6. Photograph of the developed Ethernet-to-BT controller.

port with a built-in RJ45 connector and Ethernet magnetics and a serial TTL port with full-duplex operation and baud rates of up to 115 200 b/s. In particular, as far as the Ethernet port is concerned, the device has a unique medium access control address and a static or Dynamic Host configuration Protocol (DHCP) IP address. The supported network protocols include the User Datagram Protocol (UDP), Transmission Control Protocol (TCP), Address Resolution Protocol, Internet Control Message Protocol (ICMP—PING), and DHCP.

The overall device current consumption is about 230 mA at 5 VDC. With regard to the software/firmware characteristics of the selected device, it is provided with three firmware stacks: the monitor, the application and the netloader. The monitor firmware creates the device operating system and manages the module setup, the network functions, and the device status. The application firmware is written by the user who can create server or client applications. In particular, the authors have implemented a data router device that operates in two stages. In the first stage, all the required network operations are carried out. The main ones concern the assignment of the DHCP address and the selection of the TCP-IP or UDP. In the second stage, the device operates as a double-buffer data router. In this mode, data coming from the serial port are unpacked and divided into specialized "on-the-fly" commands for serial port control, "modem" commands to control network connections, status commands for remote status monitoring, and serial data. The commands are passed to the monitor firmware, while the data are encapsulated by using the selected IP and transmitted over the network. The last firmware present on the device is the netloader, which is used to upgrade the application firmware over the Ethernet.

As far as the master throughput is concerned, no bottleneck is introduced in the device since the EM202 device allows, for its serial port, the same 115 200-b/s rate as the Bluetronics ICM201 BT device, and a very short time is required for the extraction of the data packet from the IP message.

B. *Pocket-PC-Based Handheld Controller*

Born as electronic agendas, handheld devices have nowadays achieved computational and connection power very close to those of PCs. In addition, the presence of both GPS and communication features also makes these devices powerful and economical solutions in the field of wireless sensor networks. Thus, the authors have also implemented a personal digital assistant (PDA) version of the wireless instrumentation bus controller.

As described in [14], the core of the software operating on the PC controller is a kernel daemon called BlueDaemon. This daemon thread is able: 1) to capture the messages that the RS232, IEEE488, and USB-based software exchange with the serial and IEEE488 buses; 2) to create the BT packet; and 3) to redirect these messages to the BT channel. These operations are made without requiring any action from the user. The user, after a configuration phase, sees on his or her PC the RS232 serial and IEEE488 interface ports, as if they are really available on the PC. The daemon is able to manage all the levels of the BT stack, and it can thus also be used for general-purpose software. In particular, the kernel is able to send any BT command and manage the events arriving from the queried module.

The PDA version, which is based on the LabView 7.1 PDA module software environment and the Microsoft Pocket PC driver, replicates the aforementioned thread on a PDA device. An HP IPAQ H3900 handheld device with a 400-MHz core and 64-MB RAM has been considered.

By using the Wi-Fi or GSM modules of the modern PDA handheld devices, it is possible to obtain an extended geographical wireless instrumentation bus with mobile capability.

IV. CHARACTERIZATION OF THE WIRELESS INSTRUMENTATION BUS

The performance of the wireless instrumentation interface bus has been estimated through the following quantities:

1) reliability of the wired-to-wireless interfaces versus obstacles and distance;
2) throughput of each wired-to-wireless interface board when used as a cable replacement connection and comparison with competitive wired solutions;
3) throughput of the whole wired-to-wireless interface bus with reference to a basic measurement station and common measurement operations;power budgeting of the wireless interface bus during the operating cycle.

In the following, the tests carried out are reported.

A. *Reliability of the Wired-to-Wireless Interfaces Versus Obstacles and Distance*

The aim of the test is to evaluate the number of failures in the execution of communication phases: inquiry, connection, and data packet transmission.

As far as the wireless interface is concerned, no difference exists among the BT–RS232, BT–IEEE488, and BT–USB devices. All of them utilize the same BT slave module (Broadcom ICM201) and the same microcontroller (Microchip PIC 18F452L) and implement the same connection software: this means that all exhibit the same communication time and reliability at the BT side. Therefore, a simple Piconet comprising a PC controller equipped with a BT master module and a measurement instrument (Fluke 45 digital multimeter) equipped with a BT–RS232 board has been tested. It should be noted that communication between the BT master and wireless interface boards occurs, as described in [14], by using point-to-point connections for the reverse direction and broadcast messages in the forward one. In a typical scenario, the controller, which is usually a PC, sends interface commands to peripherals (measurement instruments); these perform measurement procedures

TABLE I

INDOOR AND OUTDOOR RELIABILITY RESULTS, EXPRESSED IN THE NUMBER OF FAILURES ON 100 REPEATED TESTS

	Test Conditions								
	Distance [m]								
BT command	*Indoor without obstacles*			*Indoor with two obstacles*			*Outdoor line of sight*		
	1	2	5	10	20	50	20	50	100
Inquiry	2	0	1	1	2	4	1	1	5
Connection	0	0	0	0	0	1	0	0	2
Data send	1	0	0	0	1	2	0	1	2

and return the measured data. The BT wireless interface used for tests allows point-to-point and broadcast messages in the forward (from the PC to the instruments) direction, while only point-to-point messages are allowed for the reverse (from the instruments to the PC) direction. Point-to-point messages are accomplished with acknowledgment messages, and this choice, both for forward and reverse directions, would therefore have allowed good message integrity. Unfortunately, this solution would not have permitted the use of trigger commands that require a broadcast architecture and are very important in instrument interface bus architecture. It was thus preferred to grant the maximum integrity for the measurement data (reverse direction) and to allow broadcast messages from the forward direction (interface commands). The BT standard does not provide any acknowledgment message for broadcast commands, and a possible solution to increment message integrity would be to retransmit the same message several times. This solution was not considered to reduce the overall latency of the controller commands. Thus, given that the weak point in BT communication is represented by broadcast commands, the overall network reliability was tested only for data sent in the forward direction. Tests were carried out both in indoor and outdoor environments, and configurations experimentally reproduce sites with different signal-to-noise ratios. Table I lists some of the reliability results, which are expressed in the number of failures on 100 repeated tests, obtained, respectively, at 1, 2, and 5 m indoors without obstacles; 10, 20, and 50 m indoors all with two walls as obstacles; and 20, 50, and 100 m without obstacles and in line of sight outdoors. For each distance, the reliability of the inquiry, connection, and data sends were analyzed.

As expected, it is possible to note the following.

1) The worst performance is obtained during the inquiry phase. This can easily be understood considering that, during the inquiry phase, the BT slave devices are still not synchronized with the frequency hop sequence created by the Piconet master.
2) Tests at 100 m outdoors show the worst results since the device works very near to its receiver sensitivity limits;
3) The failure percentage in the data sends rarely exceeds the nominal bit error rate of BT broadcast communication (in noisy environments, it is typically equal to 0.1% [18], [19]).

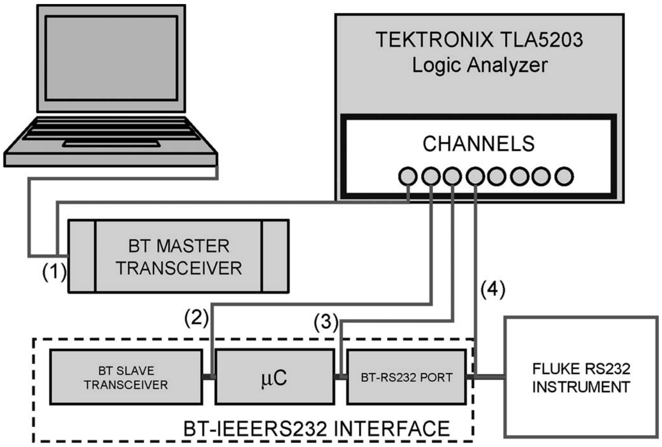

Fig. 7. Block diagram of the measurement station.

In conclusion, the reliability of the BT protocol is assured by the interface modules in all the phases of the master–slave communication. This performance is obtained within the nominal coverage area (100-m radius) in open space. Indoor test results demonstrate that communication between master and slaves is still possible with a low failure percentage in data sends within distances that depend on the number of interposed walls (for example, 50 m and two walls).

B. Throughput of Each Wired-to-Wireless Interface Board When Used as a Cable Replacement Connection

A measurement station based on the Tektronix TLA5203 logic analyzer (2-ps time resolution) was adopted for the experimental characterization of the throughput of each wireless interface module when used as a cable replacement wireless link (Fig. 7).

Three time intervals have been defined and measured.

1) T_{BT} representing the time required to send the considered command from the master BT device to the slave one. This figure of merit is measured between the (2) and (1) points in Fig. 7 and is defined as the amount of time between the departure of the first bit of the transmitted packet at the interface port of the BT master [test point (1) in Fig. 7] and the arrival of the last bit of the received

packet at the interface port of the BT slave [test point (2) in Fig. 7]. This interval comprises delays and latencies of BT master and slave interface ports, latencies in the radio layers of BT devices, and, last, the propagation time over the wireless channel.

2) T_{elab} defined as the time the microcontroller spends to unpack the BT radio packet and to transfer it to the measurement instrument. This figure of merit is estimated between the last bit that arrived at the microcontroller interface port and the first bit sent to the considered instrument interface port. This time is measured between the (3) and (2) test points in Fig. 7.
3) T_{send} defined as the time required to send the measurement packet from the microcontroller to the measurement instrument. This time, which is measured between the (4) and (3) test points in Fig. 7, is defined according to what is stated for the T_{BT} figure of merit. It includes the throughput and the latency characteristics of the realized interface port.

The sum of the three aforementioned times represents the total time (T_{total}) spent to send a command from the controller to the measurement instrument.

Due to the reduced length of the TLA5203 input probes, each test reported in the following was made at short distances between the bus controller and the interface modules (1, 2, and 5 m). In particular, unless otherwise specified, the results for a distance equal to 1 m are considered in the tests. This limited distance does not reduce the validity of the proposed tests because the time of flight of a BT packet is very short if compared to the other times (microcontroller, serial bus bottleneck, reduced BT buffer, etc.) required to send and receive data packets.

As mentioned in the BT specifications, the throughput of a communication system based on the BT wireless connection is strongly dependent on the link type, the packet type, and the state of each module in the Piconet. An accurate analysis would therefore require that a high number of combinations of these variables be experimentally explored. Nevertheless, for the sake of brevity, a unique configuration was used in the tests. The results reported in the following concern with DH5 asynchronous connection link (ACL) packet types.

As for the BT–RS232 module, a number of tests were carried out by connecting it with a FLUKE 45 digital multimeter and varying the dataload and, consequently, the payload, which is sent by the master. The RS232 port of the digital multimeter was set at 9600-b/s baud rate (its maximum available speed), 8 data bits, no parity, and 1 stop bit (9600-8-N-1).

Fig. 8 reports the performance measured versus the payload length in the [1–700] byte range. Some considerations can be outlined.

1) All over the payload range, the elaboration time T_{elab} assumes only two mean values, depending on the payload length. This happens because the microcontroller manages the whole ACL BT packet. In particular, considering a payload of less than 340 B, a single ACL BT packet has to be managed, and a T_{elab} mean value of about 85 ms was measured; when the payload exceeds this threshold, two ACL BT packets are processed, and the T_{elab} mean time gets to 175 ms.

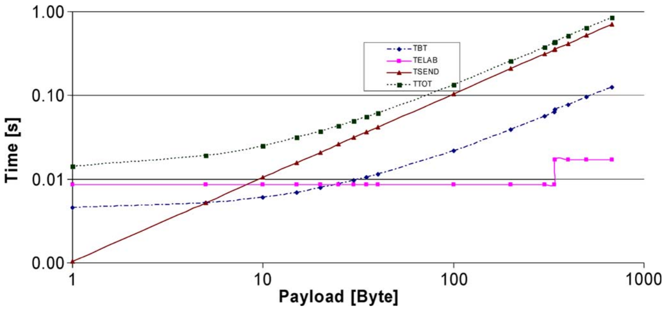

Fig. 8. Performance analysis of the BT–RS232 device versus the payload length. The T_{BT}, T_{ELAB}, T_{SEND}, and T_{TOT} figures of merit are involved.

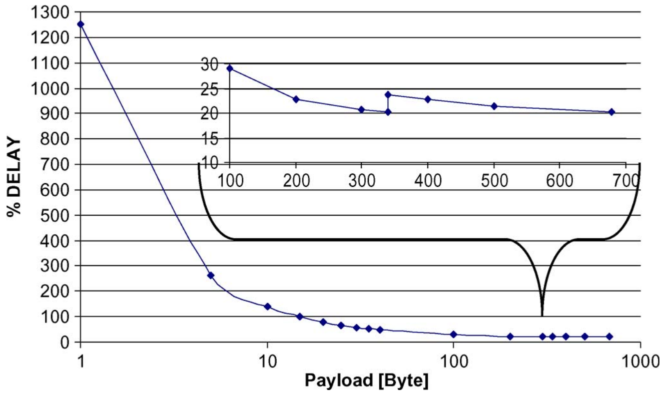

Fig. 9. Percentage delays of the wireless BT–RS232 module versus a wired RS232 connection at 9600 b/s. A zoom in the [100–700] byte range is also highlighted.

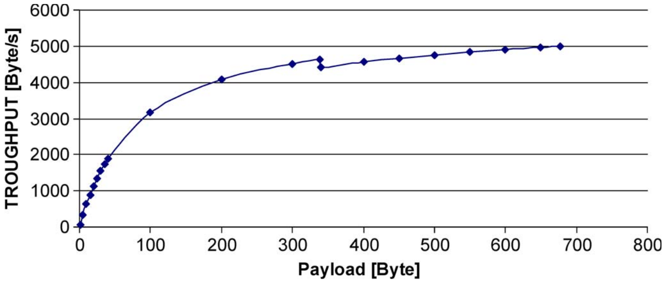

Fig. 10. Performance analysis of the BT–USB device versus the payload length; the throughput figure of merit is considered.

2) T_{elab} is comparable with the other figures of merit up to about 25-B payload. The higher payload is, the less significant T_{elab} is with respect to the other figures of merit.
3) While T_{SEND} is linear versus payload, the same cannot be said for T_{BT} due to its intrinsic latency that becomes negligible only for higher payload values.
4) Considering high values of payload, T_{TOT} becomes very close to T_{SEND} This behavior is better highlighted in Fig. 9 where the percentage delay with respect to a wired 9600-b/s RS232 connection is reported. This delay reaches a minimum of about 20% when a 339-B-long data packet is considered. A successive minimum is reached for a 678-B payload. In a wider range, this behavior would be periodic versus payload since: 1) for

TABLE II
CHARACTERIZATION OF THE THROUGHPUT OF THE REALIZED BT–IEEE488 BOARD

	Interface Commands				Device Command			
	IBSIC	**REN**	**?_!C (Addressing)**		**OHMS**		***TRIGG**	
Payload [Byte]	12	12	15		15		17	
Dataload [Byte]	1	1	4		4		6	
	Total Time	Total Time	Total Time	μc Time	Total Time	μc Time	Total Time	μc Time
Mean Time [ms]	12.49	13.07	12.33	3.338	11.95	3.332	11.81	3.331
$\sigma_{\text{Mean Time}}$ [ms]	0.21	0.16	0.25	0.027	0.14	0.025	0.27	0.023
Dataload Throughput [Byte/s]	80	77	324		334		760	

TABLE III
COMPARISON BETWEEN THE PERFORMANCE OF THE WIRELESS AND WIRED MEASUREMENT STATIONS. THE OVERALL TIME REQUIRED FOR A COMPLETE MEASUREMENT PROCEDURE IS INVOLVED

	Length [bytes]	**Twl [ms]**	**Twd [ms]**	**% (Twl/Twd)*100**
Interface Command	5	0.020	0.006	247.56
Data Arrays	5	0.050	0.006	763.54
	10	0.058	0.012	394.87
	100	0.199	0.117	69.86
	500	0.843	0.585	44.21

each transmission made by the master, the header of the BT packet has a fixed length (14 B for a DH5 ACL connection)—the greater the data packet, the lower the header's relative weight and the better the throughput; and 2) the maximum payload for the ACL DH5 connection is 339 B—one more byte added to the data packet therefore requires a second DH5 BT packet, again reducing the throughput of the wireless bus.

Similar tests have been carried out as for the BT–USB interface connected to an AGILENT 33220A function generator. The same time intervals (figures of merit) defined earlier have been considered, and the overall throughput has been estimated. T_{BT} and T_{ELAB} evolutions are very similar with respect the RS232 case, while T_{SEND} is lower than the aforementioned case thanks to the wired USB connection between the interface board and the measurement instrument. Fig. 10 describes the obtained throughput. It is possible to highlight that, as expected, the wireless connection shows a lower throughput than the wired one. This behavior, which is very far from the 12 Mb/s of the wired USB 1.1 connection, is heavily influenced by the 115 200 kb/s of the serial link of the BT device adopted. Fig. 10 also shows that, in correspondence to the minimum relative percentage delay, the estimated throughput is about 40 kb/s.

It is important to highlight that the choice of adopting the 115 200-kb/s rate for the BT devices is a huge bottleneck when fast connections, as in the USB, are considered. This low data rate worsens the overall throughput of the wireless connection. For example, if a data rate equal to 921 600 kb/s would have been set, a throughput of about 176 kb/s would have been reached. On the other hand, lower data rates allow the use of cheap microcontrollers, thus reducing the cost of the proposed interfaces and assuring the higher reliability of data transmission.

Some differences arose in testing BT–IEEE488 interface modules with respect to the BT–RS232 and BT–USB ones. While RS-232 and USB are general-purpose point-to-point buses sometimes used to connect measurement instruments to a controller, the IEEE488 bus is an asynchronous parallel instrumentation bus. This means that unlike the other two interfaces, the BT–IEEE488 throughput also depends on the handshake lines controlled by the instrument and can change depending on the number and type of connected instruments. A digital multimeter can thus be considered as a typical instrument type, and then, the BT–IEEE488 interface module can be characterized in terms of standard interface commands and device messages. Two common situations have been experimented: 1) the controller sends interface messages to the instrument; and 2) the controller sends device messages to the instrument. The results, in terms of mean time, standard deviation, and throughput, are reported in Table II. In this table, the total time is estimated by considering the total amount of time from the first bit sent by the PC controller to the last bit received from the measurement instrument, while the μc time is the microcontroller time estimated as the time spent from the last bit received from the microcontroller serial port to the first byte sent to the IEEE488 parallel bus. The throughput increases for longer data packets, even if it is quite far from the typical hundreds of kilobytes per second of the wired IEEE488 instruments. A good repeatability is instead estimated.

C. *Throughput of the Wireless Instrumentation Bus*

The throughput of the wireless instrumentation bus was evaluated by comparing the total time it requires to carry out a complete measurement procedure with the time required by an equivalent wired configuration [4], [5]. The measurement procedure was made by a typical sequence of interface and device commands, among which are some queries that were answered by devices with measurement data of different lengths. The measurement station was composed of a PC controller equipped with a BT master module, a BT–IEEE488 interface board connected to a FLUKE 45 digital multimeter, a BT–RS232

TABLE IV
CHARACTERIZATION OF THE REALIZED BT–IEEE488 BOARD IN EXECUTING BROADCAST COMMANDS

	Interface Commands			Device Command	
	IBSIC	**REN**	**?_!C (Addressing)**	**OHMS**	***TRIGG**
Mean Time [ms]	0.46	0.38	0.233	0.267	0.224
$\sigma_{\text{Mean Time}}$ [ms]	0.15	0.13	0.029	0.042	0.034

interface board connected to another Fluke 45, and a BT–USB board connected to an Agilent 33220A function generator. Each interface board has been connected to the TLA 5500 logic analyzer, as reported in Fig. 7, to estimate the time intervals defined in Section IV-B. A summary of the obtained results is given in Table III that reports the overall time necessary to send an interface command to the four instruments (row 1) and to receive data of different lengths (rows 2, 3, 4, and 5) from the measurement instrument for both the wireless (T_{wl}) and wired (T_{wd}) cases. It can be highlighted that the time necessary for the transmission of a significant amount of data could be two or three times greater than the case of wired interfaces.

In addition, to investigate the intrinsic synchronization capability of the bus, the transmission delay in sending broadcast commands from the controller to two devices has been estimated. It has been defined as the absolute time difference between the arrival of the first byte at the first instrument and the arrival of the first byte at the second instrument and has been evaluated considering the mean time and the experimental standard deviation measured in 50 consecutive tests. The test has been organized considering the aforementioned measurement station but involving only two BT–IEEE488 interfaces connected to two Fluke 45 digital multimeters. Table IV reports the obtained results. It is possible to note that the worst case of about 460 μs happens only once. This time represents the intrinsic latency in synchronizing the measurement instruments connected in the wireless bus.

TABLE V
SUMMARY OF THE POWER BUDGETING IN A 1-H OPERATING CYCLE

Interface	*Operating state*	*Required Time [S]*	*Required current [mA]*	*Charge (Current * Time) [A*s]*
BT-RS232	Discoverable	60	88	5.28
	Processing	3526	48	169.248
	Transmission	14	78	1.092
BT-488	Discoverable	60	80	4.8
	Processing	3526	40	141.04
	Transmission	14	70	0.98
BT-USB	Discoverable	60	101	6.06
	Processing	3526	61	215.086
	Transmission	14	91	1.274

D. Power Budgeting During the Operating Cycle

Among the measurement devices that one could want to connect to the wireless instrumentation bus, one might also find battery-supplied devices such as smart sensors or portable instruments. There are even cases where choosing a wireless connection for a measurement device means that it is difficult to reach it with any wiring and that it is not convenient to frequently remove it to change or recharge batteries. In all these cases, the wireless interface module must also be battery supplied, and so, energy management becomes a topic of great interest. Since the implemented interface boards can be also be supplied with battery packages, a power budgeting analysis was made to estimate board lifetime for a given battery type.

As for power budgeting, the realized interfaces manifest different types of behaviors due to their hardware differences. The architecture of each wireless interface board can be grouped into three parts: 1) the microcontroller; 2) the transceiver, and 3) the hardware required for the connection to the instrument measurement port. In particular, points 1 and 2 *share* parts common to each interface board, whereas point 3 changes for the wired bus considered. The microcontroller is made by using a Microchip PIC16LF452 low-power microcontroller and has a maximum current lower than 1.6 mA at 5 V during normal operation and less than 0.1 uA during standby. The transceiver consists of a Broadcom ICM101 BT master/slave transmission device that adsorbs about 80 mA at 3.3 V during the transmission phase. For point 3, three different solutions are possible: 1) a TTL-to-RS232 Maxim MAX232 transceiver (8 mA at 5 V) for the BT–RS232 board; 2) a Cypress SL811HS (21 mA at 5 V) USB host for the BT–USB board; and 3) a set of 16 digital lines for the BT–IEEE488 board.

Each board can operate in three stages: 1) discoverable; 2) processing; and 3) communication. These stages are detailed in the following.

1) In the discoverable stage, each board makes itself visible to the Piconet master waiting for a master–slave connection. In this stage, the microcontroller and the interface hardware are on, while the transceiver is in discoverable mode. This mode represents the highest mean current consumption mode for the BT transceiver. Current consumption in this stage is about 88 mA for the BT–RS232 board, 80 mA for the BT–IEEE488 board, and 101 mA for the BT–USB board, respectively. The time that the boards remain in this state depends on the master inquiry.
2) In this state, the microcontroller operates in run mode, the interface hardware is on, and the transceiver is also powered up. Current consumptions are about 48, 40, and 61 mA, respectively, for BT–RS232, BT–IEEE488, and BT–USB, respectively.
3) In the communication state, all the devices are on, and the transceiver is in transmission state. The maximum available transmission rate is 115 200 b/s, while the total current consumption is 78, 70, and 91 mA for the BT–RS232, BT–IEEE488, and BT–USB interfaces, respectively.

Since the time that each board remains in a given state is user and application dependent, only a statistical lifetime can be estimated.

For example, the results reported in Table V are obtained considering a discoverable time equal to 1 min and an operating cycle expecting one connection and data equal to 100 kB to be processed and transmitted for each hour.

Looking at the total charge reported in Table V, it is possible to highlight that the throughput improvement obtained by the BT–USB interface board implies more charge to achieve the imposed operating cycle. Considering a typical battery package comprising a set of Duracell AA batteries with 2700-mAh capacitance, the estimated lifetimes are 15, 18, and 12 h for the BT–RS232, BT–IEEE488, and BT–USB interface boards, respectively.

V. CONCLUSION

The performance of a wireless interface bus, which is enhanced through the introduction of a further interface module (BT–USB) and through extension to the geographic network, has been completely characterized. The wireless interface allows RS232, IEEE488, and USB instruments to participate in the same BT Piconet controlled by a PC equipped with a commercial BT master dongle. The results of tests carried out on general-purpose instrumentation (digital multimeters, oscilloscopes, and function generators) featured with the proposed wireless interface have shown that instruments can be up to 100 m away from the controller in open space (50 m with obstacles indoors). Moreover, while a significant reduction of throughput must be taken into account, particularly when compared to some wired one-to-one connections (USB), if more instruments have to be interfaced or moving instruments have to be controlled, the overcharge becomes less evident, and the advantages of a flexible, fast-to-set-up, and dynamic connection increase.

REFERENCES

[1] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2001, pp. 2033–2036.

[2] K.-S. Tang, K.-F. Man, and S. Kwong, "Wireless communication network design in IC factory," *IEEE Trans. Ind. Electron.*, vol. 48, no. 2, pp. 452–459, Apr. 2001.

[3] M. Dunbar, "Plug-and-play sensors in wireless networks," *IEEE Instrum. Meas. Mag.*, vol. 4, no. 1, pp. 19–23, Mar. 2001.

[4] P. Arpaia, A. Baccigalupi, and A. Pietrosanto, "Performance optimization of VXI-based measurement stations," *IEEE Trans. Instrum. Meas.*, vol. 44, no. 3, pp. 828–831, Jun. 1995.

[5] L. Angrisani and A. Pietrosanto, "Performance comparison of IEEE-488 and 1155 based measurement stations," in *Atti del VII IMEKO TC-4*, Praga, Czech Republic, Sep. 1995, pp. 634–638.

[6] M. Gooding, "VXI plug'n play," in *Proc. AUTOTESTCON—IEEE Syst. Readines Tech. Conf.*, Aug. 1998, pp. 448–451.

[7] F. Alegria and H. G. Ramos, "A remote controlled automated measurement system," in *Proc. IEEE IMTC*, Ottawa, ON, Canada, May 1997, pp. 1186–1189.

[8] D. Aviles and O. Gutierrez, "Automatization of the multifunctions laboratory," in *Proc. Conf. Precision Electromagn. Meas. Dig.*, 1998, pp. 199–200.

[9] S. V. Wunnava and P. Hoo, "Remote instrumentation access and control (RIAC) through Internetworking," in *Proc. IEEE Southeastcon*, Mar. 1999, pp. 116–121.

[10] *The USB Specification Documents in Adobe pdf.* [Online]. Available: http://www.usb.org/developers/docs

[11] U. Bilstrup and P. A. Wiberg, "Bluetooth in industrial environment," in *Proc. IEEE Int. Workshop Factory Commun. Syst.*, Sep. 2000, pp. 239–246.

[12] D. Famolari and P. Agrawal, "Architecture and performance of an embedded IP Bluetooth personal area network," in *Proc. IEEE Int. Conf. Personal Wireless Commun.*, Dec. 2000, pp. 75–79.

[13] L. Ferrigno, V. Paciello, and A. Pietrosanto, "A Bluetooth-based proposal of instrument wireless interface," *IEEE Trans. Instrum. Meas.*, vol. 54, no. 1, pp. 163–170, Feb. 2005.

[14] L. Ferrigno, A. Pietrosanto, and A. Celano, "Enhancement of a Bluetooth-based instrument wireless interface," in *Proc. XVII IMEO World Congr.*, Dubrovnik, Croazia, Jun. 2003, vol. TC4, pp. 521–526.

[15] *Standard IEEE 802.15.1 Documents in Adobe pdf.* [Online]. Available: http://ieeexplore.ieee.org/iel5/9980/32053/01490827.pdf

[16] *Specification of the Bluetooth System Documents in Adobe pdf.* [Online]. Available: http://www.bluetooth.com/English/Technology/Building/Pages/Specification.aspx

[17] J. C. Haartsen, "The Bluetooth radio system," *IEEE Pers. Commun.*, vol. 7, no. 1, pp. 28–36, Feb. 2000.

[18] L. Ferrigno, V. Paciello, and A. Pietrosanto, "Extending a short-range wireless instrumentation bus to a geographic network," in *Proc. XXIII IEEE IMTC*, Sorrento, Italy, Apr. 24–27, 2006, pp. 1268–1273.

[19] A. El-Hoiydi, "Interference between Bluetooth networks—Upper bound on the packet error rate," *IEEE Commun. Lett.*, vol. 5, no. 6, pp. 245–247, Jun. 2001.

[20] N. Golmie, R. E. Van Dyck, and A. Soltanian, "Interference of Bluetooth and IEEE 802.11: Simulation modeling and performance evaluation," in *Proc. ACM Int. Workshop Modeling, Anal., Simul. Wireless Mobile Syst.*, Rome, Italy, Jul. 2001, pp. 11–18.

Luigi Ferrigno (M'04) received the M.S. degree in electronic engineering from the University of Salerno, Salerno, Italy, and the Ph.D. degree in electrical engineering from the University of Napoli, Napoli, Italy.

He is currently an Aggregate Professor of electrical and electronic measurements and the Chief of the Metrological Laboratory with the Dipartimento di Automazione, Elettromagnetismo, Ingegneria dell'Informazione e Matematica Industriale, University of Cassino, Cassino, Italy. His current research interests include the realization and characterization of sensors and wireless sensor networks and the realization of the measurement system for nondestructive testing and characterization of RF digital apparatuses.

Vincenzo Paciello (M'08) was born in Salerno, Italy, in 1977. He received the M.S. degree in electronic engineering and the Ph.D. degree in information engineering from the University of Salerno, Salerno, in 2002 and 2006, respectively.

Since 2008, he has been an Assistant Professor of electrical and electronic measurements at the University of Salerno. His current research interests include wireless sensor networks, instrument interfaces, and digital signal processing for advanced instrumentation.

Antonio Pietrosanto (M'99) was born in Naples, Italy, in 1961. He received the M.S. and Ph.D. degrees in electrical engineering from the University of Naples, Naples, in 1986 and 1990, respectively.

In 1991, he was an Assistant Professor of electrical measurements with the Department of Information Engineering and Applied Mathematics, University of Salerno, Salerno, Italy, where he became an Associate Professor of electrical measurements in 1999 and has been a Full Professor of electrical and electronic measurements since 2001. His scientific interests are primarily focused on sensor realization and characterization, wireless instrument interface, digital signal and image processing, and instrument fault detection and isolation.