Collecting data from a sensor network in a single-board computer

F Casciati¹, S Casciati², Z-C. Chen³, L Faravelli¹, M Vece¹

¹ DICAR, University of Pavia, Via Ferrata 3, 27100 Pavia, Italy
² DICAr, School of Architecture, University of Catania at Siracusa, P.za Federico di Svevia, 96100 Siracusa, Italy
³ Institute of Micro/Nano Devices and Solar Cell, College of Physics and Information Engineering, Fuzhou University, QiShan Campus, Fuzhou 350108, China

E-mail: miche.vece@gmail.com

Abstract. The EU-FP7 project SPARTACUS, currently in progress, sees the international cooperation of several partners toward the design and implementation of a satellite based asset tracking for supporting emergency management in crisis operations. Due to the emergency environment, one has to rely on a low power consumption wireless communication. Therefore, the communication hardware and software must be designed to match requirements which can only be foreseen at the level of more or less likely scenarios. The latter aspect suggests a deep use of a simulator (instead of a real network of sensors) to cover extreme situations. The former power consumption remark suggests the use of a minimal computer (Raspberry Pi) as data collector. In this paper, the results of a broad simulation campaign are reported in order to investigate the accuracy of the received data and the global power consumption for each of the considered scenarios.

1. Introduction

The European Union FP7 project named “Satellite Based Asset Tracking for Supporting Emergency Management in Crisis Operations (Spartacus)” is currently in progress. It is devoted to the implementation of a local low power wireless communication network. The Spartacus project aims to develop Galileo-ready satellite-based applications in order to: (i) track, trace and localize critical transport assets especially in case of major failure of existing networks, (ii) track the flow of relief goods from the sending side to the receiving ones and (iii) support coordination and ensure the safety of first responders in disaster management operations [1]. The aspect of this project, of interest for the authors, is to develop a mobile communication infrastructure on site by integrating terrestrial wireless access technologies with satellite backhauling ones in which the low power local communication network is required to connect GNSS based tracking units and a centralized collecting unit. Indeed, since many GNSS based tracking units are expected to be placed in any considered transportation vehicle, the conventional cabling method suffers problems due to inflexible installation and negative impact on the vehicle structure [2-4]. Moreover, the local wireless communication network should be low power in order to operate long enough to support the first 24 hours after an emergency and it should have encryption function to ensure a high security. This paper details the way to achieve a link to the positioning data transfer, with a NMEA protocol, from a GNSS sensor to a credit-card sized...
computer (Raspberry Pi) used as data collector of a reliable and independent wireless communication network.

2. State of the art
At present, most investigation efforts aim at improving the current framework related to the exploitation of the potential offered by the satellite network. Indeed, the coordination of operations in emergencies, based on precise positioning and timing system, can find help in the transfer of information coming from a network of units reading the Global Navigation Satellite System (GNSS). Commercially, most GNSS signal generators and record/playback systems are designed specifically for high volume production test applications for devices that use commercial GPS, SBAS, GLONASS, Galileo and other GNSS receivers [5]. The quality of these commercial systems continues also to increase with the addition of new GNSS signals and features [6], but despite all the flexibility provided by these commercial systems, there is still a need for full access to GNSS signal simulation environments for some GNSS developers to generate their own customized simulation scenarios [7].

![Figure 1. Focus on satellite navigation system orbits](image)

As well known in literature [8-10], global positioning systems (GPS) offer a valid solution because this technology covers a large area and it is suitable for the monitoring of constructions requiring accuracy of centimetres and which are characterized by a large own period. Indeed, the main feature of GPS units is to rely on internet for transferring the local information to a recording nucleus, however in any disaster areas, it is quite likely that the existing terrestrial communication network and power grid could be cut off after the catastrophic event. Therefore, a reliable asset tracking system requires being independent of those terrestrial networks, especially in terms of communication.

3. System architecture
The developed system architecture consists of an inertial sensor with internal GNSS receiver and a ZigBee wireless data transmit device both connected to a hardware interface, as shown in Figure 2.
In order to satisfy all the requirements of the project, selected devices are a sensor provided by SBG Systems (Ellipse N) [11], a credit-card sized computer (Raspberry Pi Model B) [13] and GBAN ZigBee Wireless Data Transmission Devices.

4. The output code

These NMEA-0183 messages [12] allow external devices to use selected data collected or computed by the GNSS receiver. All messages conformed to the above format begin with $ and end with a carriage return and a line feed. Data fields follow comma delimiters and are variable in length; null fields still follow comma delimiters, although they do not contain information, and an asterisk delimiter with a checksum value follow the last field of data. The checksum is the 8-bit exclusive of all characters in the message, including the commas between fields, but not including the $ and asterisk delimiters. The hexadecimal result is converted to two ASCII characters and the most significant character appears first [12]. The NMEA messages handled by the sensor to input aiding data are the follow:

- GGA, for position and altitude;
- RMC, for horizontal, velocity and course;
- HDT, for true heading;
- ZDA, for UTC time data.

In the case under study, the required messages (RMC and GGA) are achieved using the configuration settings of the sensor within the proprietary software. An example of a RMC message string is $GPRMC,123519,A,4807.038,N,01131.000,E,022.4,084.4,230394,003.1,W*6A and in Table 1 the message fields are reported in detail.
Table 1. GPRMC message fields.

| Field | Meaning |
|-------|---------|
| 0     | Message ID $GPRMC |
| 1     | UTC of position fix |
| 2     | Status A=active or V=void |
| 3     | Latitude |
| 4     | Longitude |
| 5     | Speed over the ground in knots |
| 6     | Track angle in degrees (True) |
| 7     | Date |
| 8     | Magnetic variation in degrees |
| 9     | The checksum data, always begins with * |

When NMEA-0183 output is enabled, a subset of NMEA-0183 messages (version 3.01) can be sent to external instruments and equipment connected to the receiver serial ports. The output data transmission is provided by an USB 2.0 interface, which is compatible with the data collector (Raspberry Pi Model B) input ports.

5. The data collector

The Raspberry Pi is a credit card-sized single-board computer developed in the UK by the Raspberry Pi Foundation [13] with the intention of promoting the teaching of basic computer science in schools. The Raspberry Pi is based on the Broadcom BCM2835 system on a chip (SoC), which includes an ARM1176JZF-S 700 MHz processor, VideoCore IV GPU and was originally shipped with 256 megabytes of RAM, later upgraded to 512 MB. The system has Secure Digital (SD) or Micro SD (Model A+ and B+) sockets for boot media and persistent storage [13]. Figure 4 shows the chosen model with a view of its components: a micro USB power supply, an SD card slot, an RCA video and audio jack output, two USB 2.0 and Ethernet ports.
The Foundation also provides Debian and Arch Linux ARM distributions for download and several tools are available for Python as the main programming language, with support for BBC BASIC (via the RISC OS image or the Brandy Basic clone for Linux), C, C++, Java, Perl and Ruby. The RasPi can support USB mouse and keyboard and for USB devices drawing more than 100 mA a powered USB hub is strongly recommended. The most relevant pins for the project, in order to process the data are the GPIO pins, are shown in Figure 5.

**Figure 3.** Raspberry Pi Model B (elaborated from [13])

The header provides 17 Pins that can be configured as inputs and outputs. By default they are all configured as inputs except GPIO 14 and 15.

**Figure 4.** RasPi Model B GPIO layout

The wireless data transmission devices

The ZigBee standard network is a low-cost, low-power, wireless mesh network. ZigBee operates in the industrial, scientific and medical (ISM) license-free radio bands: 868 MHz in Europe, 915 MHz in the USA and Australia and 2.4 GHz in most jurisdictions worldwide. Data transmission rates vary from 20 kilobits/second in the 868 MHz frequency band to 250 kilobits/second in the 2.4 GHz frequency band. ZigBee networks are secured by 128 bit symmetric encryption keys. Its transmission distances range from 10 to 100 meters line-of-sight, depending on power output and environmental characteristics. In order to have a longer range, an external radio power amplifier can be adopted with
ZigBee compliant transceiver. There are many ZigBee compliant transceivers and protocol stacks offered by different companies, which can implement the ZigBee network. The chosen device for the project is the GBAN GB-RFTO (Figure 6), based on CC2530 SoC.

**Figure 5. GBAN GB-RFTO ZigBee device**

GB-RFTO [14] series ZigBee wireless data transmit device is a reinforced ZigBee module, integrated with an RF transceiver and a microprocessor based on ZigBee standard protocol. It can achieve long distance communication, strong anti-jamming capability, flexible networking. It can also realize a point-to-multipoint and multipoint-to-multipoint transfer of data. Its equipment can support the transmission of the serial port RS232, USB, Wi-Fi and NET data. The layout is reported in Figure 7.

**Figure 6. GBAN GB-RFTO ZigBee layout (elaborated from [14])**

The CC2530 [15] is a true System on Chip (SoC) solution for IEEE 802.15.4, ZigBee and RF4CE applications. It enables robust network nodes to be built with very low total bill-of-material costs. The CC2530 combines the excellent performance of a leading RF transceiver with an industry-standard enhanced 8051 MCU, in-system programmable flash memory, 8-KB RAM. The CC2530 has various operating modes, making it highly suited for systems where ultralow power consumption is required. Short transition times between operating modes further ensure low energy consumption. Combined with the industry-leading and golden-unit-status ZigBee protocol stack (Z-Stack™) from Texas Instruments, the CC2530F256 provides a robust and complete ZigBee solution. An integrated development environment (IDE) is needed to program and compile the firmware code for the 8051 microcontroller core integrated in the CC2530 transceiver. The adopted and compatible IDE software is IAR Embedded Workbench 8051, provided by Texas Instruments [15].
It is also possible to program the boards with the Texas Instruments Flash Programmer [16] as an alternative to IAR.

A ZigBee protocol stack is needed, which is provided free by Texas Instruments and is named Z-Stack™ [17]. It is a complete protocol stack and application development solution, conforming to ZigBee Alliance standards. The downloaded Z-Stack installation package contains all of the documentation and software required to install, configure, and develop applications using Z-Stack. The package employs a Microsoft Windows-based installation application, which guides the installation process. Texas Instrument also provides two other software tools for free to facilitate the debug and testing of ZigBee transceiver and wireless network: SmartRF Studio 7 [18] and Packet Sniffer [19]. SmartRF Studio 7 is a PC application that can be used in combination with CC2530. It runs on Windows XP, Windows Vista and Windows 7 (32 and 64 bit) and uses USB (or parallel port for legacy boards) to communicate with the evaluation board (EB) which has an evaluation module (EM) with the RF chip mounted. The program provides an easy-to-operate PC interface to all of the chip’s radio configuration registers, and it is very helpful for functional testing and for finding the appropriate radio settings. SmartRF Studio 7 can also be used without any hardware, but then only to generate, edit and export radio register values. The main feature of SmartRF Studio 7 is its control panel for direct access to the RF-IC’s chip registers and radio-related features. It gives an overview of the many device specific features and provides full read and write access to the chip’s radio registers [18]. The Device Control Panel expert mode screenshot is reported in Figure 7.

SmartRF Studio 7 communicates with the Evaluation Board over the USB interface via a library called CEBAL (Chipcon Evaluation Board Access Layer). This is a SW library developed to interface the USB driver and firmware running on the evaluation board, containing all the functions required to read/write data over the SPI interface between the USB MCU and the Transceiver or the debug
interface in case of a System-on-Chip. The SmartRF Packet Sniffer is a PC software application used to display and store RF packets captured with a listening RF HW node [19].

The Packet Sniffer filters and decodes packets and displays them in a convenient way, with options for filtering and storage to a binary file format. The supported protocols can be seen in the Launch window when starting the packet sniffer and for IEEE 802.15.4 devices, the required channel must be selected.

7. Conclusions

This contribution drives the reader across the selection path followed by the author to achieve a preliminary solution for a low-power wireless communication network. The task was carried out within the European FP7 Spartacus project. Mainly one adopts the ZigBee-based technology, i.e., CC2530 SoC transceiver and Z-Stack. More or less likely scenarios are addressed through the detailed study of communications protocols and reception systems to implement several functions, such as saving collected data that allow (one to set up) the data post-processing.

References
[1] F. Casciati, S. Casciati, Z-C. Chen, L. Faravelli, M. Vece. Development of a Reliable Wireless GNSS Sensor Network. EWSHM 2014, 7th European Workshop on Structural Health Monitoring, Nantes, France, 8-11 July 2014.
[2] F. Casciati, L-J. Wu. Wireless links for global positioning system receivers. Smart Structures and Systems, vol.10, n.1, pp 1-14, 2012.
[3] S. Casciati, L. Faravelli, Z-C. Chen. Energy harvesting and power management of wireless sensors for structural control applications in civil engineering. Smart Structures and Systems, vol.10, n. 3, pp 299-312, 2012.
[4] S. Casciati, Z-C. Chen. A multi-channel wireless connection system for structural health monitoring applications. Structural Control & Health Monitoring, vol. 18, n. 5, pp 588-600, 2011.
[5] A. Brown, R. Tredway, R. Taylor. GPS Signal Simulation using Open Source GPS Receiver Platform. Proceedings of the 21st Virginia Tech Symposium on Wireless Personal Communications, Blacksburg, Virginia, June 2011.
[6] 2013 Simulator Buyers Guide. GPS World, May 1, 2013.
[7] F. Casciati, C. Fuggini. Engineering Vibration Monitoring by GPS: Long Duration Records. Earthquake Engineering and Engineering Vibration, vol. 8, no. 3, pp. 459-467, 2009.
[8] F. Casciati, C. Fuggini. Monitoring a steel building using GPS sensors. Smart Structures and Systems, 7(5):349–363, 2011.
[9] L-J. Wu, F. Casciati. Local positioning systems versus structural monitoring: a review. Structural Control and Health Monitoring, DOI: 10.1002/stc.1643, 2014.
[10] Alison Brown and Neil Gerein. Advanced GPS Hybrid Simulator Architecture.
[11] Ellipse AHRS & INS - High Performance, Miniature Inertial Sensors User Manual. Document ELLIPSEUM.1. Revision 1 - Oct 7, 2014.
[12] http://www.trimble.com/OEM_ReceiverHelp/V4.44/en/NMEA-0183messages_MessageOverview.html
[13] http://en.wikipedia.org/wiki/Raspberry_Pi
[14] http://www.gban.cn/en/product_show.asp?action=p297&id=43
[15] http://www.ti.com/product/cc2530
[16] CC2530 Software Examples User’s Guide. Copyright © 2009, Texas Instruments Incorporated.
[17] http://vast.ucce.uga.edu/files/projects/traumagps_files/docs/chipcon/Z-Stack.pdf
[18] SmartRF Studio 7 Tutorial Product. Copyright © 2011, Texas Instruments Incorporated
[19] SmartRF™ Packet Sniffer User Manual. Copyright © 2010, Texas Instruments Incorporated