Creating a New Scalable, Low Cost and High Performance M2M Platform with Multiple Communication Interfaces Support

Bruno Cendón, Javier Valiño and Juan Rico
Tecnologías Servicios Telemáticos y Sistemas S.A.,
39011 Santander, Spain,
Email: {bcendon, jvalino, jrico}@tst-sistemas.es

Abstract— **The actors playing in current telecommunications world have clearly evolved from a human exclusive perspective towards a segregated view in which the communications between machines start to lead a significant role. This special profile of communications is gathered under the umbrella of what is called *machine-to-machine* communications, most widely known as M2M. Initially conceived as remote monitoring and/or actuator networks, the devices conceived to serve M2M purposes are in constant evolution, with higher requirements in their processing power and capabilities in order to support improved protocols, and applications. These market demands are not always followed as many of the existing devices are based on outdated hardware and software architectures or have been created to cover less demanding deployments.**

In this paper we describe the conception, development and deployment of a versatile M2M platform with different communication interfaces (ZigBee, cellular, Wi-Fi, Ethernet,…) based on state-of-the-art low cost but high capabilities 32 bits processors with a supporting API over an open source RTOS.

I. Introduction

The diversification in the different usages of the Internet is one of the reasons of its exponential growth these last years. Originally conceived from an exclusive user point of view, the arrival of new communication technologies and the significant reduction in the prices of the devices enabling access to the net has enlarged the potential applications coverage. Looking back just a few years ago, the purchase of a device that enabled the access to the Internet was estimated in the order of the thousand of EUR. Nowadays, there is a plethora of devices providing this access, and minimal capabilities for programming, for prices even lower than 50 EUR. The consequences of this so called “democratization” of the access to the Internet have allowed the creation of multiple new usages and applications that were unthinkable just a few years ago. The main beneficiaries of this drastic reduction of costs have been the electronic devices that have seen improved its capabilities by means of having the possibility to access the Internet. This improvement has allowed the users to “talk” with any kind of electronic device, and has even enabled the possibility of having these machines “speaking” between them. This last has been such a successful application that is now denominated by the dedicated term “machine-to-machine communications” and its acronym M2M. This category includes all devices, topologies, technologies and devices that enable an exchange of information data between machines and its proper processing.

Current developments in the M2M world are demanding richer features in the devices for a lower price to allow its inclusion in new applications. Concepts like the integration with WSN (Wireless Sensor Networks) or the support of applications related with the IoT (Internet of Things) are showing that current devices are becoming outdated. New developments and ideas are needed to have smarter, less power consuming and cheaper devices.

TST is a young high-tech SME born as a spin-off of one of the leading teams in the Telecommunications school in the University of Cantabria, Spain. The focus of the company has been initially put in R&D tasks in order to create a cost sensitive, above the current market offering and smart M2M platform highly scalable in connectivity (sensors, actuators, I/O interfaces, storage, smart cards…) and communication technologies, going from the support of low power WPAN technologies like ZigBee up to wired technologies like PLC.

This platform, internally called *TSmarT*, has been implemented in two different devices, an open API over the devices firmware and a MCV application middleware.

II. Novelties of the Proposed Platform

After the experience of using several commercial solutions made for creating WSN and M2M topologies, TST made the decision of developing its own platform. The main goal was to include on it the features which were not available on other already developed systems, so as to make a real approach to smart environments and provide high scalability at a reasonable price. These new features are summarized below.

A. Use of brand new processors

Most of low cost and low powered processors on sale so far were based on 8-bit solutions with low speed and limited memory resources. New families of processors developed for this kind of deployments, with high capabilities and low costs, has been lately introduced into the market landscape. After a thoughtful analysis of the market offer, the choice between new 32 bits microcontrollers retained the attention for the platform. Finally it was the STM32 [1] processor family developed by ST Microelectronics [2], based on the ARM Cortex M3 [3] core the one the one chosen to develop the platform, due to its advantages talking about scalability, pinout, processor's speed and memory configuration.

B. Using a true Real Time Operating System

Software developments for commercial *motes* used on WSN deployments are based on Operative Systems (OS) created for this type of low capabilities' devices. TinyOS [4], Contiki [5] or LiteOS [6] are all of them great approaches to enable basic OS features, derived from big development efforts. Any monolithic development will not include these advantages. Nevertheless, there is a limitation in the capabilities of these reduced OSs based on its own conception for being used on low capabilities' devices. By using these new ARM Cortex-M3 based processors, it is possible to use OSs offering better features as eCos [7], µC/OS-II [8] or FreeRTOS [9]. They are more powerful, and also able to perform a better exploitation of the microprocessor capabilities offering real time. After an exhaustive study of all the options available in market, the OS chosen for the platform is the open development FreeRTOS.

C. Radio Modules with scalable mesh network's protocols

There are many WSN solutions using proprietary RF front-ends and communication stacks. Despite the fact that this design is versatile, its scalability is compromised, and it is really difficult to develop new designs based on it. In order to avoid this situation, after a deep market analysis, Digi's XBee modules have been selected as the radio transceivers. These modules allow using different protocol stacks (ZigBee [10], ZigBee PRO, IEEE 802.15.4 [11] and DigiMesh$_{TM}$ [12]), different ISM frequency bands (2.4 GHz and 868/900 MHz), different antenna's plug-in mode (PCB integrated, wire integrated, UF.L connector and RP-SMA connector) and different emission power ranges, when selecting among its two versions (XBee 1mW and XBee PRO 10mW at 2,4GHz). Moreover, Digi offers a Wi-Fi module compatible with its XBee module in terms of pinout. Both devices support two modes, the first one using AT commands and transparent data transmission, and the second one on API mode.

D. Provide full system scalability

There are several differences and commonalities between WSN and M2M deployments, making difficult to distinguish when to use one or the other. WSN is often related to low cost, low power and limited capabilities developments, with short radio coverage and some sensors attached. On the other hand, M2M is associated with remote sensors and actuators integrated on mid/low capabilities devices, IP connectivity and, sometimes, other communication interfaces. Given this description, a so-called M2M device can be regarded as a WSN gateway. The proposed platform aims to clarify these commonalities and differences by offering a scalable system, made up of several devices with different capabilities, sharing the same hardware and software basis. This commonality will be reflected in the use of a STM32 microcontroller with an ARM Cortex-M3 core, FreeRTOS OS and the same hardware abstraction API.

Based on the concepts previously depicted, it is possible to develop a platform with scalable, low-cost and high-capabilities devices. This way, a device created for M2M applications with capabilities to work in WSN and IoT developments, where all the devices shall fulfill minimum capabilities in terms of memory and processing.

III. Platform Description

Given the scalability and RTOS capability showed, the best way to describe the platform is from hardware and software's point of view. The first one will define the minimum capabilities, footprint and cost. On the other hand, the second one will provide the tools, based on common software, which will enable working independently of eventual hardware variations.

A. Hardware

As previously mentioned, the platform basis consists on a microcontroller with an ARM Cortex-M3 core, which offers processing and managing features for a number of surrounding devices. Figure 1 shows a schematic view of the architecture, presenting the most relevant platform's possibilities. Depending on the application envisaged, the features integrated may vary, thus affecting the overall cost of the final platform.

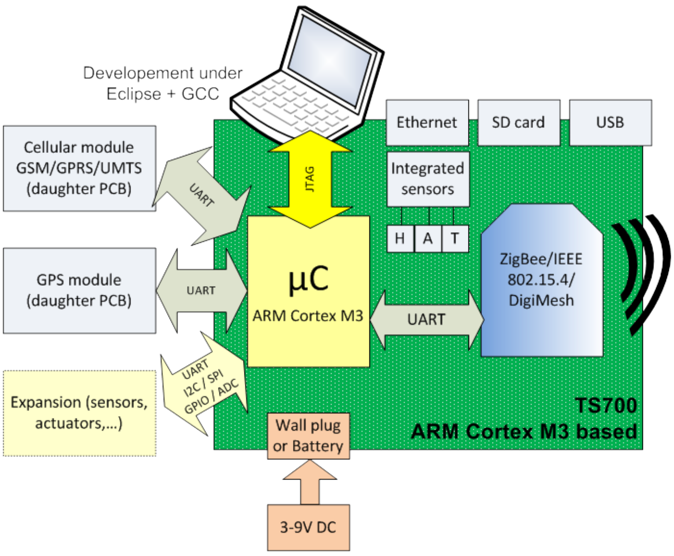

Figure 1. Generic Hardware architecture.

As a result of an exhaustive state of the art research, the solution chosen for the microprocessor is the one provided by ST Microelectronics, STM32. The decision was based on the chip's features, pinout and both static and dynamic memory size. Among the whole family of chips, STM32F1 [13] series will be used, providing different sub-families with differences between them in terms of connectivity, memory sizes and packaging:

- Value line STM32F100xx: 24MHz CPU with home appliance's managing options.

- Access line STM32F101xx: 36MHz CPU with up to 1MB flash.
- USB access line STM32F102xx: 48MHz CPU with USB FS interface.
- Performance line STM32F103xx: 72MHz CPU with up to 1MB Flash with USB and CAN bus.
- Connectivity line STM32F105/107xx: 72MHz CPU with MAC Ethernet, CAN bus and USB 2.0 OTG.

A number of scalable communications peripheral devices will be pluggable on the platform and managed by the microprocessor:

- Low-power mesh networks communications through Digi's XBee module.
- Cellular communications with a 2G/3G PCB modem, SAGEM's HiLO currently used.
- Wired communications through Ethernet port, taking advantage of STM32F107 features.
- Other kind of communications through Wi-Fi compatible XBee modules, PLC modems or direct data transmission through UART, I2C, SPI, etc…

There will be a considerable number of extension ports enabled: UARTs, I2C, SPI, GPIOs, CAN bus and USB FS/OTG port (depending on the processor chosen). Many other peripheral devices will be initially on in successive releases supported such SD memory card, SPI Flash memory, LCD interface/controller, onboard sensors (humidity, temperature and 3-axis accelerometer), expansion industrial sensor board, parking sensor, NFC card reader, GPS, etc.

Device must be powered in the range of 3.3-5 VDC, converting internally this level to 3.3 VDC (with some exceptions, as SAGEM's HiLo, 3.7 VDC). Power source electronics will be dimensioned taking into consideration whether to use batteries or not, the basis for making that decision will be the application due to its critical role on platforms cost.

B. Software

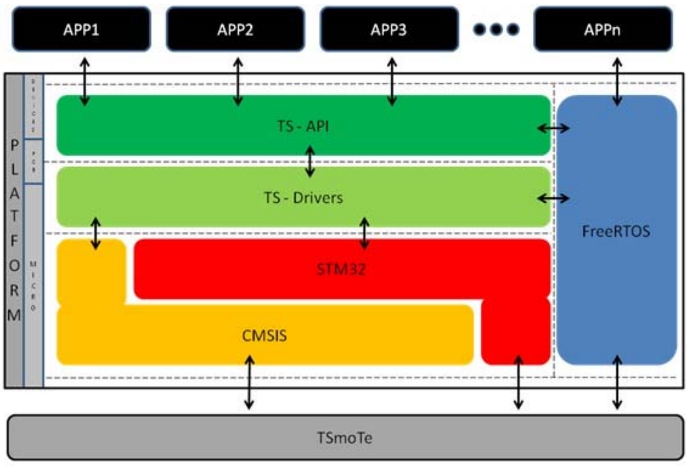

Figure 2. Software architecture.

Among the platform's strengths, its modular and configurable software implementation is one of the most remarkable. Due to this versatile conception of different devices, several software modules have been created so as to configure the microcontroller and peripheral devices. With this idea of simple and easy programming, an API is provided in order to manage all the features enabled by the device with no need to understand the very low software or OS level. Figure 2 summarizes this software architecture:

- **CMSIS**: *Cortex Microcontroller Software Interface Standard* [14], generic hardware abstraction layer for Cortex-M processors (regardless manufacturer). Provides a common base for all Cortex-M microcontrollers, making software's portability easier. It brings few capabilities, but lets the programmer avoid dealing directly with records.

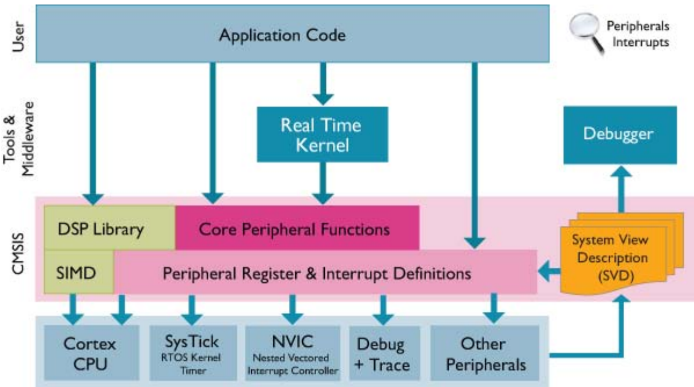

Figure 3. Cortex Microcontroller Software Interface Standard

- **STM32 Library**: As CMSIS presents quite basic abstraction features, ST Microelectronics provides an additional library in order to enable packaging in several microcontroller's atomic operations. It is widely used on the platform, as it is versatile and powerful.
- **FreeRTOS**: Popular real time OS for embedded devices, with specific implementation for ARM Cortex-M3 family chips. It is delivered under GPL license, allowing its use on proprietary code when the kernel is no modified. This GPL license is intended to encourage the industry about using this particular OS. Commercial version is available as well, named OpenRTOS. In order to reduce its footprint to the most, it is based on standard ANSI C, enabling the creation of understandable code, something difficult in other reduced low footprint OS such as TinyOS or Contiki. The scheduler con be configured either on cooperative or appropriative modes, supporting co-routines.
- **TS-Drivers**: API designed to offer the programmer an easy way to use peripheral devices and microprocessor's ports. For instance, this API includes libraries to manage UART, SPI, I2C, or analog and digital I/O ports.
- **TS-API**: Built on top of TS-Drivers layer, this API enables the use of common peripheral devices. As an example, with this API it is possible to manage the

XBee module, the 2G/3G modem, the GPS module, the SD card, a NFC device or the Ethernet port.

- **Configuration files**: The platform architecture will allow reconfiguring hardware for the intended use. The changes in configuration could be applied to different levels, at microprocessor level (pinout, PLL and clock tree, port assignment, etc.), PCB level (PCB pinout, PCB port assignment, etc.) and device level (initial configuration of peripheral devices).
- **Application**: All devices implementing these software architecture will be able to perform multitask, real time applications, with co-routines and interruptions support. Among the default system applications, some tasks as system initialization or TCP/IP stack (based on µIP) will be already programmed.

Finally, it is important to point out that all the toolchain needed in order to make the complete development can be performed with free software tools. It is possible to do so by using Eclipse as development platform and GCC (Codesourcery release) as compiler. That implies no extra costs, apart from the JTAG used, required when developing application based on this hardware platform.

IV. Implemented Devices

The platform's first generation is made up of two different devices. The idea is to respect the node-repeater-gateway architecture, developing two different modules with a hardware and software common basis, but with different capabilities and prices. For the node and repeater functions, a "node and repeater" device is presented, with optimized costs and enhanced capabilities in comparison to available commercial solutions. The "gateway" is the other device conceived, with IP gateway features and enhanced connectivity features.

A. Node and Repeater

Figure 4. Node and repeater device.

From a commercial point of view, this device aims to be a flexible and powerful wireless communication platform, with optimized costs and equipped with multiple communication interfaces in order to manage sensors or actuators. In addition to these potential external sensors, it has three built-in sensors onboard: temperature, humidity and 3-axis accelerometer. Furthermore, a GPS-enable daughterboard (integrating *Fastrax IT500* module) is available, enabling it to perform mobile tracking applications, for instance fleet management. Other expansion interfaces are available as well, including UART, I2C, SPI and CAN bus, apart from analog and digital I/O interfaces.

Finally, an USB socket, a SD card slot, a SPI Flash memory, a Wi-Fi module (Digi's XBee Wi-Fi), a 2G/3G modems (SAGEM's HiLo), an IEEE802.15.14/ZigBee/DigiMesh module and a RFID/NFC interface (NXP's PN532) are available to be integrated. Envisaged applications cover not only remote sensing, processing and data transmission from sensors connected to the device, but also remote controlled applications supporting remote management of relays and/or actuators.

B. Gateway

Figure 5. Gateway device.

Once depicted all features available at the node/repeater device, the gateway is introduced as a powerful communication platform, integrating multiple interfaces, both wireless and wired ones. That makes it the most appropriate way to enable the outside communications from a network of nodes.

The added features consist on an Ethernet socket and the availability of more I/O ports/interfaces, due to the fact that a bigger microprocessor is used. Furthermore, it usually will implement higher memory features.

In order to manage this Ethernet socket, the software running here will include TCP/UDP/IP, HTTP, ICMP and DHCP support, by using µIP library [15].

V. Success Stories

Potential applications for this new platform are huge, but, due to its novelty, few real tests have been performed. Nevertheless, some devices are successfully being used on commercial and R&D projects:

A. Lighting and Signalling Harbour's control system

The system comprises nodes with connected relays acting on street lights and railway semaphores and Ethernet-connected gateways as concentrators. Control and management software, based on PHP/MySQL web developments and Google Maps, is provided as well. In addition, a set of mobile remote controls based on the smaller TSmarT node are developed so as to act directly on the network.

Figure 6. Device deployed inside an antenna case on Bilbao's harbor controlling railway signaling over a dedicated WPAN.

B. Hospital inventory management system

This time the nodes are independent and NFC enabled, controlling medicament's inventory. They are able to transmit data using ZigBee, GPRS, Wi-Fi and/or Ethernet, depending on the device placement.

The system is controlled with a PHP/MySQL web developed software. A Java resident program is deployed as well in order to monitor devices, sending alarms in case of connectivity issues.

C. Testbed for R&D Projects

Given the scalability of the platform introduced, it has been proposed as an experimentation platform to different R&D projects. Initially using the commercial version of the device, adaptations to the implementation are being done to address specific requirements in the different projects addressed. Among them, we can point out the following ones:

- EXALTED (EXpanding LTE for Devices) [16] FP7 project aims to integrate M2M communications on the actual LTE environment. TST is actively involved on the capillary network's design, using the devices as traffic aggregators, enabling context-aware routing protocols and allowing OMADM remote management.
- BUTLER FP7 project's objective is creating a Smart Life system based on context. Devices are introduced here as the basis for the smart cities cluster, on which TST will act as FP7 SmartSantander [17] project interface.
- SICRA [18] from the Spanish AVANZA program is a joint effort between industry, university and public authorities to design an improved emergency management system form avalanches using advanced GNSS techniques and ad-hoc WPAN deployments between injured and rescue teams . Devices will be used in the demonstrator given their scalability and processing capabilities.

VI. CONCLUSIONS

The integration objectives are ambitious, and its goals are quite complex. The strategic development plan has been aligned in order to fit into the commercial and R&D projects on which the company is involved.

The platform evolution will be conditioned not only by these requirements, but also by the improvements of implemented components, such as open source solutions' enhancements and novelties, and potential new hardware versions (for example, using either STM32F2 or STM32W families).

The same way, this platform will be the basis of a wide range of devices, the family can be easily expanded by adding more/less powerful or cheap modules.

From the application point of view, the developments shall be focused on SmartLife and the offered connectivity shall be enhanced, either by adding IPv6 support or by improving the web services implemented.

VII. ACKNOWLEDGEMENT

Part of the work presented has been performed in the framework of the ICT project ICT-5-258512 EXALTED, which is partly funded by the European Union. The authors would like to acknowledge the contributions of their colleagues. The Community is not liable for any use that may be made of any of the information contained therein.

VIII. REFERENCES

[1] ST Microelectronics. Website STM32 http://www.st.com/internet/mcu/class/1734.jsp

[2] ST Microelectronics. Website http://www.st.com

[3] ARM. Cortex M3 processors. Website http://www.arm.com/products/processors/cortex-m/cortex-m3.php

[4] TinyOS. Project Website http://www.tinyos.net/

[5] Contiki OS. Project Website http://www.sics.se/contiki/

[6] LiteOS. Project Website http://www.liteos.net/

[7] eCos. Project Website http://ecos.sourceware.org/

[8] µC/OS-II. Website http://www.micrium.com/page/products/rtos/os-ii

[9] FreeRTOS. Project Website. http://www.freertos.org.

[10] Zigbee Alliance. Website http://www.zigbee.org/

[11] IEEE 802.15.4 Standard. Website http://www.ieee802.org/15/pub/TG4.html

[12] DigiMesh networking protocol. Website http://www.digi.com/technology/digimesh/

[13] ST Microelectronics. "STM32 - 32-bit ARM® Cortex™-M MCUs". Presentación de la familia de procesadores STM32. http://www.st.com/internet/com/SALES_AND_MARKETING_RESOURCES/MARKETING_PRESENTATIONS/PRODUCT_PRESENTATION/stm32_marketing_pres.pdf

[14] CMSIS - Cortex Microcontroller Software Interface Standard. Website http://www.arm.com/products/processors/cortex-m/cortex-microcontroller-software-interface-standard.php

[15] µIP library. Website http://www.pjort.com/micro-ip/

[16] EXALTED (EXpanding LTE for Devices). Project website. http://www.ict-exalted.eu/

[17] SmartSantander. Project Website. http://www.smartsantander.eu/

[18] SICRA. Project Website http://www.sicra.es