Adaptive STM32F4 Microcontrollers
Case of Flexible Smart Meters[*]

Aymen Jaouadi^{1,2}, Olfa Mosbahi^{3}, Mohamed Khalgui^{3} and Ahmed Toujani^{2}

^{1}*Cynapsys Company, France-Germany*

^{2}*FST, University of Tunis, El Manar, Tunisia*

^{3}*LISI Laboratory, INSAT Institute, University of Carthage, Tunis, Tunisia*

Keywords: Microcontroller, STM32F4, Reconfiguration, Smart Meter, Smart Grid, Software Agent, Modeling and Verification, Simulation.

Abstract: The research paper deals in theoretical level with flexible and adaptive microcontrollers following the well-known industrial technology STM32F4. It is applied in the practical level to a Smart Meter SM which is developed at Cynapsys for future generations of Smart Grids. A reconfiguration scenario in theory is assumed to be any flexible operation allowing the addition-removal-update of OS tasks in order to adapt the micro-controller to its environment according to user requirements. It is assumed in practice to be any addition-removal-update of new services to-from SM such as the energy consumption, the remote information reading and power shutdown, the stabilization of the delivered power, the management of new power provider offers, the sale of energy and finally the peak consumption management. We propose an agent-based architecture for a STM32F4 device where a hierarchical software agent is defined to control the environment evolution before applying local reconfigurations for a required flexibility of the microcontroller. We model the agent by using nested timed automates, and design the whole architecture to manage all possible reconfiguration forms. The agent-based architecture is totally implemented and applied to SM, and a simulator X-SM is developed for the evaluation of this paper's contribution.

1 INTRODUCTION

Microcontrollers, STM32 F4 are special-purpose computer systems designed to perform one or few dedicated functions often with real-time computing constraints in order to control physical processes in the real world. The requirements in the development of microcontrollers are increasingly growing in term of flexibility and agility (Vyatkin et al., 2005) (Pratl et al., 2007). The most promising directions to address these issues is the reconfiguration of one of the most used microcontrollers in the world which is the STM32 F4 to be autonomous as a new challenge, STM32 F4 represents 45% of the microcontroller market, and provides users a number of annexes integrated component, it allows the addition of devices, such as the management of LCD displays. This functionality refers to the process of modifying the software as well as hardware structure and behavior of the system during its execution. Being reconfigurable is important for reacting fast to sudden and unpredictable requirement changes or perturbations with minimum cost and risk. Several interesting academic and industrial research works have been made in recent years to develop reconfigurable control systems (Gehin and Staroswiecki, 2008).We distinguish in these works two reconfiguration policies: static and dynamic reconfigurations such that static reconfigurations are applied off-line to apply changes before the system could starts (Angelov et al., 2005), whereas dynamic reconfigurations are applied dynamically at run-time. Two cases exist in the last policy: manual reconfigurations applied by users (Rooker et al., 2007) and automatic reconfigurations applied by Agents (Al-Safi and Vyatkin, 2007)(Theiss et al., 2009). We are interested in software reconfigurations, and assume that all the

[*]This research work is a collaboration between LISI Laboratory (INSAT Institute) at University of Carthage and the German-French Company Cynapsys in Tunisia. We thank Ing. Haythem Tebourbi Technical Director and Ing. Souhail Kchaou Director of Research and Development for long fruitful discussions and stable financial and technical supports.

Jaouadi A., Mosbahi O., Khalgui M. and Toujani A..
Adaptive STM32F4 Microcontrollers - Case of Flexible Smart Meters.
DOI: 10.5220/0004975903640374
In *Proceedings of the 3rd International Conference on Smart Grids and Green IT Systems* (IEEHSC-2014), pages 364-374
ISBN: 978-989-758-025-3

reconfigurations are applied automatically by an internal module of the microcontroller that should be kept autonomous as much as possible. In the case of a manual reconfiguration, we assume that the reconfiguration agent interacts with user to execute there requests. This controller is assumed to be a superset of OS software tasks such that only a particular subset is loaded at run-time to implement it under user constraints. A reconfiguration will be any automatic operation allowing the addition, removal and update of tasks at run-time. We propose a new module named Embedded Control Module ECM to handle software reconfigurations of the microcontroller. This module is a superset of sub-modules that manage all possible reconfiguration scenarios such as the addition, removal or update of tasks at run-time (Zhang and Wang, 2010) (Pfeffer and Ungerer, 2004) (Chauhan et al., 2009) (Otero et al., 2006). We propose a UML-based model for this ECM which plays a key function inside the autonomous microcontroller. We apply this contribution by assuming the case of electric meters following the STM32 technology of microcontrollers. We consider in particular the smart meter developed at the German-French Company Cynapsys located in Tunisia. This meter supports the classic task dealing with the computation of the energy consumption. Nevertheless, the company plans to add new functionalities that allow these meters to be autonomous and smart. The remote shutdown of the electric power when the bill is not paid, the stabilization of energy, the remote reading of data or display of news, the management of peaks, the management of renewable energy or the remote negotiation between the provider and the clients, will be new original services to be added to the microcontroller of the planned Smart Meter in order to enhance the performance of the devices and optimize the energy management. All these original services to be considered as OS tasks will be handled by the ECM that adapts the device according to user requirements by adding or removing or updating tasks at run-time. The planned smart meter will be a new challenge for Cynapsys in Tunisia to guarantee services with high qualities and with low-costs. We specify these services as well as the ECM according to the model checker by using the tool UPPAAL which is used to simulate the new meter (Alur and Dill, 1994) (Bengtsson et al., 1996). We propose also the implementation of the STM32 smart meter and show the experimentations that we did at Cynapsys to evaluate the whole contribution. The organisation of this paper is as follows. The next section analyzes the Background and a detailed description of the Reconfigurable Embedded Systems, the STM32 Microcontrollers and Current Smart Meters. Section 3 is dedicated to the description of the Smart Meter at Cynapsys Company and describe new challenges and solutions. Section 4, proposes reconfigurable STM32 microcontrollers where we formalize, model and design their behaviors followed by a detailed modeling of our system modules by using UML Diagrams. Section 5 proposes the new original Smart Meter at Cynapsys where we detail their verification, design and implementation. Finally Section 6 concludes this work. Our approach is original by implementing new services in the Smart Meter to manage production and the optimisation of the consumption at Real-Time.

2 BACKGROUND

In this section, we present an overview on reconfigurable embedded systems, the STM32 microcontrollers family and the current used smart meters.

2.1 Reconfigurable Embedded Systems

An embedded system is reconfigurable if it changes its software or hardware behavior at run-time according to user requirements. The software reconfiguration is any operation allowing the addition, removal or update of software tasks that implement the system to encode corresponding functions. The hardware reconfiguration is assumed to be any operation allowing the addition, removal or update of hardware components according to user requirements. An addition or removal can be of memory, of data-event inputs-outputs, or of a new network for communication. The update of hardware components can be the modification of the processor speed. The constant growth of the complexity afferent and necessary to the management of embedded software systems makes reconfiguration autonomy increasingly important. The challenges include both the model design level and the environment level of runtime support. According to(Boukhannoufa, 2012), real-time systems can be large, distributed, and have a dynamic environment. This requires the introduction of various modes of operation and reliability techniques to ensure its operation and maintainability. Moreover, these dynamic changes of architecture and behavior have a negative impact on the temporal characteristics of systems that require a special study on the ability of adaptive behaviors to ensure the hard real-time constraints imposed to the systems. These adaptive behaviors amplify the complexity of developing real-time systems. According to (Wang et al., 2010), the new generations of embedded control systems are addressing new criteria such as flexibility and agility. To enhance their operations, the embedded control systems should be changed when random disturbances happen or when improvements of the performance should be applied and propose a dynamic low power reconfigurations of real-time embedded control systems that should respect hard real-time constraints and perform modifications of the periods and deadlines, modification of the worst case execution times (WCETs), and finally the removal of some tasks to minimize the energy consumption in order to adjust the behavior of the processor. According to (Kramer and Magee, 1985), the authors propose a reconfiguration model based on the dynamic incremental modification and extensions. In this work, the required properties are determined by languages and their execution environment. In (Thramboulidis, 2004b) (Thramboulidis, 2004a), the authors propose a new vision for the reconfiguration of distributed control applications. The main objective is to bridge their works on software engineering (*eg.* UML) to IEC 61499 (J.H.Christensen et al., 2005). In (Zhang et al., 2013) the authors present a reconfiguration architecture for real-time distributed control systems. Specifically, the low-level control of the physical components which is shown with the handling of real-time requirements. The main contribution of this work is to define a reconfiguration request that interacts with the application under reconfigurations through three interfaces. More specifically, the modification interface provides its role based on a set of key reconfiguration services identified in a previous work (Zoitl et al., 2010). Our contribution is to provide an original approach for the automatic and manual agent-based reconfiguration model for STM32 F4 microcontrollers implementing original services for low power management.

2.2 STM32 Microcontrollers

A microcontroller is an integrated circuit that brings together the essential elements of a computer: CPU, memory (ROM for program RAM for data), peripheral units and input-output interfaces. Microcontrollers are characterized by a higher degree of integration, lower power consumption, lower operating speed, and reduced cost compared with versatile microprocessors used in personal computers. Microcontrollers are widely used in embedded systems such as auto engine systems, remote controls, home appliances and mobile phones. The STM32 technology of 32bit Flash microcontrollers based on the ARM Cortex processor is proposed to provide a 32bit product range that combines high performance, real-time capabilities, digital signal processing, and low-power, low-voltage operation, while maintaining full integration and ease of development. The different STM32 microcontrollers, based on an industrial standardization with a large choice of tools and software, make this family of STM32 the ideal technology for small projects and for entire platform decisions. To apply our reconfiguration approach, we opted for the choice of a STM32 card type STM32 F4 Discovery which is based on STM32F407 Cortex M4 controller with 1MO Flash and 192KO RAM. The STM32F407 family is based on the high-performance ARM Cortex-M4 32-bit RISC core operating at a frequency of up to 168 MHz. The Cortex-M4 core features a floating point unit (FPU) single precision which supports all ARM single precision data-processing instructions and data types. It also implements a full set of instructions and a memory protection unit (MPU) which enhances application security. This card uses power from the USB bus or from an external 5V power supply, a 3-axis accelerometer LIS302DL, two buttons and a micro USB connector (STMicroelectronics, 2013). The use of STM32 F4 cards is the first approach in the field of reconfigurable microcontrollers, in terms of originality, our scientific paper is characterized by the application of different types of architectural, scheduling and data reconfiguration in order to add new services at runtime to the operation of the Smart Meter developed at Cynapsys.

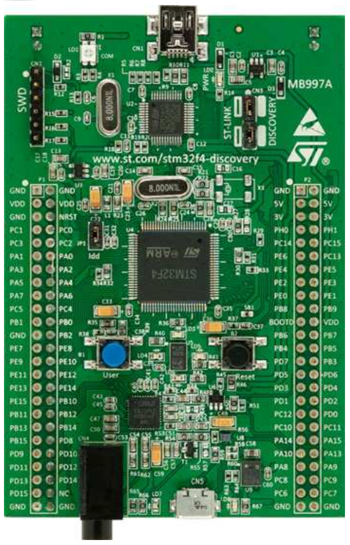


Figure 1: STM32 F4 Discovery Card.

2.3 Overview on Current Smart Meters

According to (S.Depuru et al., 2011), a smart meter is an advanced energy meter that measures the consumption of electrical energy, provides additional information compared to a conventional energy meter.

A smart meter is a counter that automatically registers and communicates data on electricity consumption, it is divided into two main categories: traditional meters (the old electromechanical meters) measure only the total electricity consumption and Smart meters measure the amount of electricity consumed and the time of consumption, they automatically transmit data to providers by using a wireless communication technology (Ahmad, 2011) (Hoenkamp and Huitema, 2012). The price of electricity with these data may vary during the day, giving houses a new way to manage the costs of their consumptions. You could, for example, choose to reduce your consumption during the most expensive time (peak or normal) and partially use electricity during off-peak hours, during which it will be cheaper. Investigations conducted in Tunisia have for the moment appear incomprehensible bill increases in number. These Meters are intelligent and used to regulate the energy delivery in peak of consumption periods, for example the French Electricity Distribution Network ERDF has with the Smart remote Meter Linky, the ability to change their rates from one minute to another, according to the consumption of international prices electricity (ERDF, 2009). In Cynapsys, we propose a new strategy to add the original proposed services to the old developed Smart Meter which was a remote control example of a DC motor. This last represents one of the end devices that can be used at home or a building. The communication with the Energy Counter is established by means of wireless communication protocol ZigBee. The Energy counter module allows providing information about the average of the energy consumption of all the elements connected to it. In addition it permits to get the real-time consumption of the home or building via ZigBee protocol. Today the deployment of Smart Meters should allow a better understanding of the positions of power consumption and thus save money. Monitoring real-time also opens the door to energy billing in real time, but also to the differentiated based billing electricity demand (peak load). Into this research we will enhance the behavior of the electric meter by the services described above in order to make it more intelligent. Intelligence brought many advantages for the customer and energy companies. Starting with a bill which can be calculated on the basis of real consumption, interventions carried out remotely (without constraint of appointment) including stabilization of the varying voltage and regulating consumption by micro-cuts in periods of high consumption known to peak loads. Consume less power and better is the goal of our study. These optimizations will be applied to the electric meter developed by Cynapsys. The intelligence provided to the meter is by adding new services that provide an immediate benefit to the counter in real-time. Smart meters equipped consumers should not pay an estimate bill but they consider real power consumptions. This new generation of meters can display power consumption and remote meter reading. Thus the consumer can also save money by better monitoring its power consumption. The use of smart meter certainly makes life easier for users, thanks to its new services such as the automatic control of electrical devices at home and the regulation of supply and demand through remote micro-cuts in order to not exceed a fixed consumption threshold. Our paper consists of two novel approaches, the first is a theoretical approach to enhance the electric meters with new original services, and in the second we propose a practical approach by proposing the X-SM Simulation tool.

3 ADAPTIVE SMART METERS OF CYNAPSYS COMPANY: NEW CHALLENGES AND SOLUTIONS

At the German-French Company Cynapsys, the current research paper is applied to a well-developed STM32-based electric meter for the remote control of home end devices. The meter encodes a main counter module ECM to provide the real-time energy consumption of any connected device to it by using a Zigbee-based protocol. In addition to this classic function, we propose to enrich the meter with original services such as the remote shutdown of power, the remote reading of the energy consumption from the meter to the provider, the smart stabilization of power by using renewable energy that we assume available at home, the display of the energy consumption for users at home, the smart management of promotions to be offered from the provider by relatively activating or deactivating home end devices when the price of power (per hour) is low or high, the sale of energy to the provider when the available renewable energy is enough, the smart peak management by deactivating home end devices during peak times, and finally the management of news to be sent from the provider about the network status in the next weeks or months. These optimizations will be applied to the electric meter developed by Cynapsys. Note that all these services will not be loaded together on memory in order to not overload the memory by the applications and the tasks that will not be executed. Thus, the system preserves all its resources to treat most substantial services. Except the main service dealing with counting, we assume that each another will be loaded on memory when needed at run-time. Reconfiguration in our approach keeps the system available, increase operating efficiency and especially without remarkable performance degradation. We propose a model of dynamic automatic reconfiguration, or possibly manual to be executed through a decision module that handles these cases in real-time. The current paper deals with the theoretical level of the STM32 Microcontroller to be assumed flexible and autonomous. Each service is assumed to be implemented by an OS task. The system is then implemented by a superset of tasks such that only one subset implements it at a particular time after a well-defined reconfiguration scenario. The paper deals in the application level with this Smart Meter of Cynapsys to be enriched with these new services. Each addition-removal of a service on the smart meter corresponds to a reconfiguration scenario. The services and the tasks in our approach are executed according to the priority to be assigned and based on the importance of the operations as the remote power shutdown that must be executed upon his arrival. We denote in the following by:

- T_1: the main software task that measures the energy consumption by reading directly and not by an estimation of consumption. This task has the highest priority,
- T_2: the software task allowing the remote shutdown. The meter we conceive is remotely programmable and equipped with a remote switching device called ”AMM” (Advanced Meter Management). This service is crucial in the case of non-payment of bills,
- T_3: the software task allowing the remote reading from the smart meter to the power provider and remote users,
- T_4: the software task allowing the stabilization of the delivered power from the provider by using if possible (if the input load is decreased at run-time) the renewable energy that we assume available at home,
- T_5: the software task allowing the display of information on the meter which is equipped with a digital displayer,
- T_6: the software task allowing the management of promotions to be offered from the provider, this task informs users by SMS and emails about these offers,
- T_7: the software task allowing the sale of energy to the provider when the available local energy is enough, this task informs users by SMS and emails,
- T_8: the software task allowing the management of peaks by using useful information to be sent by SMS and emails from the provider,
- T_9: the software task allowing the management of all news to be sent from the provider about the network status. This task informs users by SMS and emails.

The proposed tasks are considered as a new generation of energy production and consumption. They represent a solid medium for effective management of the power across the electric Smart Meter which is the entrance of the smart grid.

4 THEORETICAL CONTRIBUTION: RECONFIGURATIONS OF STM32F4 MICROCONTROLLERS

We aim in this section to dynamically reconfigure a STM32 microcontroller which is assumed to be implemented by a set of independent OS tasks. The goal is to adapt its behavior at run-time to its environment according to well-defined user requirements. A reconfiguration scenario is assumed to be any dynamic operation allowing the addition, removal or update of tasks to-from the microcontroller. We propose an Embedded Reconfiguration Module to be denoted by ERM that controls the evolution of the microcontroller's behavior and considers also user requirements to apply run-time reconfigurations. The ERM is assumed to be encoded in three hierarchical software levels: (a) Architecture Level (to be denoted by AL), (b) Scheduling Level (to be denoted by SL), and (c) Data Level (to be denoted by DL). We define in AL, all the possible software architectures that can implement the STM32 microcontroller at run-time. An architecture in AL is a set of OS tasks that perform control activities. A reconfiguration scenario can change the software architecture of the microcontroller by adding or also removing OS tasks. For each architecture in AL, we need to define an execution model of the corresponding tasks. A scheduling is then defined in SL to affect a priority to each task. For each architecture and for each scheduling of the corresponding tasks, we define also in DL all the possible corresponding values of data to be handled at run-time. Thanks to this hierarchical structure, the ERM can handle all possible reconfiguration scenarios of a STM32 microcontroller, our approach is original because no previous work did the same hierarchical architecture on STM32F4 to handle reconfiguration scenarios.

4.1 Formalization of Reconfigurable STM32 Microcontrollers

Let *Sys* be the STM32 microcontroller that can be reconfigured at run-time to adapt its behavior to its environment. We denote by Γ_{Sys} the big set of all the possible tasks involved in the different implementations of the system *Sys*, which is implemented at any particular time *t* by a subset ξ_{Sys} which represents the set of tasks involved in a particular implementation $\xi_{Sys} \subseteq \Gamma_{Sys}$. We model the architectural level *AL* of *ERM* by a finite state machine S_{AL} such that each state of S_{AL} corresponds to a particular implementation at architectural level.

$S_{AL} = (\Gamma_{Sys}, O, \delta)$, such that,

- O is a set of n states in S_{AL} ($O = \{S^i_{AL}/i \in 1..n\}$),
- δ is a state-transition function $\Gamma_{Sys} XO \rightarrow \Gamma_{Sys} XO$

A reconfiguration scenario $R^{i,j}_{AL}$ is a transition from a state S^i_{AL} corresponding to particular subset of tasks ξ^i_{Sys} to a state S^j_{AL} corresponding to particular subset of tasks ξ^j_{Sys}. In this scenario, we assume run-time operations allowing addition-removal of tasks to adapt the system *Sys* to its environment. For each state S^i_{AL}, we define in the second hierarchical level(Scheduling Level *SL*) a particular state machine to be denoted by S_{SL}. Each state in S^i_{SL} defines a particular scheduling of the subset of tasks ξSys. This scheduling affects a priority to each task in order to get a deterministic execution model of the microcontroller *Sys*. We denote by $\Psi(\xi_{Sys})$ the set of all possible execution models of tasks of ξ_{Sys} at the Scheduling Level.

$S_{SL} = (\Psi(\xi_{Sys}), P, \beta)$, such that,

- P is a set of m scheduling states in S_{SL} ($P = \{S^i_{SL}/i \in 1..m\}$)
- β is a state-transition function $\Psi(\xi^i_{Sys})XP \rightarrow \Psi(\xi^i_{Sys})XP$

A reconfiguration scenario $R^{i,j}_{SL}$ at Architectural Level AL, is a transition from a state S^i_{SL} to another state S^j_{SL} of S_{SL}. The reconfiguration of the microcontroller *Sys* in the third hierarchical level *DL* can be the update of data. We define for each state S^i_{AL} of S_{AL} and also for each state S^j_{SL} of S_{SL} a new state machine S_{DL} where each state corresponds to new values to be affected to data of tasks belonging to ξ_{Sys} under the scheduling S^i_{SL}. Let $\Upsilon(\xi_{Sys})$ be the set of all possible values of data for the tasks of ξ_{Sys} under the scheduling S^j_{SL}.

$S_{DL} = (\Upsilon(\xi_{Sys}), Q, \beta)$, such that,

- Q is a set of l data states in S_{DL} ($Q = \{S^i_{DL}/i \in 1..l\}$)
- β is a state-transition function $\Upsilon(\xi^i_{Sys})XQ \rightarrow \Upsilon(\xi^i_{Sys})XQ$

A reconfiguration scenario $R^{k,h}_{DL}$ is a transition to change the values of data from a state S^k_{DL} to another state S^h_{DL} of S_{DL}. We denote finally by $Config_{i,j,k}$ a configuration of the STM32 microcontroller *Sys* to be implemented by the subset of tasks ξ^i_{Sys} under a well-defined scheduling corresponding to a state S^j_{SL} in S_{SL} with particular values of data defined in a state S^k_{DL} in S_{DL}. A reconfiguration scenario to be denoted by $Reconfig^{u,v,w}_{i,j,k}$ of *Sys* is defined then as any run-time operation allowing the adaptation of the microcontroller from a configuration $Config_{i,j,k}$ to a new one $Config_{u,v,w}$. The originality of our approach is distinguished by a specific formalization of the scenarios as possible reconfigurations and the exclusivity of the application of our original ideas to the Electric Smart Meter.

4.2 UML-based Design of Reconfigurable STM32 Micro Controllers

To implement the different run-time reconfiguration scenarios of the STM32 microcontroller *Sys*, we propose in Figure 2 a UML class diagram that designs both the Embedded Control Module ECM and also the different OS tasks belonging to Γ_{Sys}. Our class diagram consists of a ECM class that plays a very important role in our reconfiguration approach. This class manages all the interactions in the system, it is related to the classes AL, SL, DL which respectively represent the architectural reconfiguration level, scheduling reconfiguration level and data reconfiguration level. We have also four other classes: Provider Interface, User Interface, Equipment Interface and Task. The Task class represents all the tasks included in the different implementations of our system. The user interface class represents all the customers of our system. The Equipment Interface class, includes all devices connected to the counter. Finally, the Provider Interface class manages the relationship with the electricity supplier.

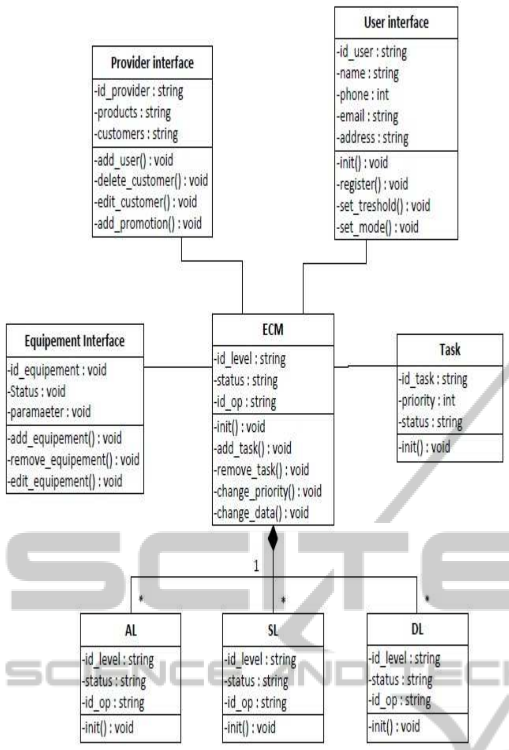


Figure 2: UML Class Diagram of an Adaptive STM32 Microcontroller.

5 PRACTICAL CONTRIBUTION: CASE OF NEW SMART METERS AT CYNAPSYS

In this section we present the practical contribution of our approach.

5.1 Timed Automata Models

We describe in Figure 3 the different state machines encoding the ECM of the STM32-based Smart Meter. The state machine encoding the Architectural Level is composed of four states:

- Counting the energy consumption
- Regulation of the energy
- Promotion negotiation
- Remote shutdown of the electricity

The first state Counting corresponds to the system's implementation $\xi^1_{Sys} = \{T_1, T_3, T_5, T_9\}$ when the main task T_1 is active as well as the tasks T_3, T_5, T_9 that allow respectively the remote reading, the display and the management of news. The second state Regulation corresponds to the system's possible implementation $\xi^2_{Sys} = \{T_1, T_4, T_5, T_9\}$ when the main task T_1 is also active as well as the tasks T_4, T_5, T_9 that allow respectively the stabilization, the display and the management of news. The third state Negotiation corresponds to the system's possible implementation $\xi^3_{Sys} = \{T_1, T_5, T_6, T_7, T_8, T_9\}$ when the main task T_1 is active as well as the tasks T_5, T_6, T_7, T_8, T_9 that allow respectively the display, the management of promotions, the sale of energy, the management of peaks, and finally the management of news. We describe also in the scheduling level the different possible execution models of tasks encoding each state in the architectural level. The scheduling *Sched*1 of the state Counting executes the main task of counting T_1 each time it executes T_3, T_5 and T_9. The precision of counting is high in this case. The scheduling *Sched*2 of the same state Counting reduces the precision by executing T_1 two times in a trace of execution. The precision of counting is low in *Sched*3 that executes T_1 only one time. We describe in Data Level the periodicity of each trace of execution in the second level. In this case, *Sched*1, *Sched*2 and *Sched*3 are respectively executed periodically each $100ms$, $300ms$ and $500ms$.

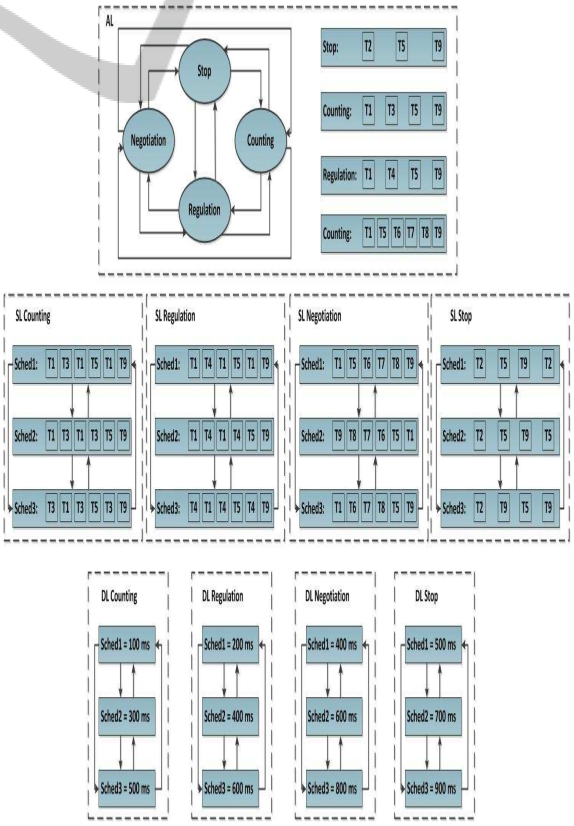


Figure 3: State Machines of the Embedded Control Module that handle the reconfigurations of the STM32-based Smart Meter.

The design of reactive systems must comply with logical correctness: ”the system does what it is supposed to do” and timeliness: ”the system has to satisfy a set of temporal constraints”. The smart meter should respect the notion of time such as if a power interruption is requested it should execute this task immediately, then it can reactivate the electricity due to the existence of functional constraints such the payment of bills or the passage of a period of high consumption. We model and simulate the different state machines of ECM as well as the different assumed tasks by using the well-known model checker UPPAAL in order to check the behavior of the adaptive smart meter after each reconfiguration scenario. According to (Palshikar, 2004), model checking is the most successful approach that’s emerged for verifying requirements. We describe in Figure 4 the model of the second task T_2 allowing the remote shutdown of the smart meter according to the formalism Timed Automata. The meter we conceive is remotely programmable and equipped with a remote switching device called ”AMM” (Advanced Meter Management). This service is crucial in the case of non-payment of bills, the deployment of a team to ensure power cuts today becomes useless.

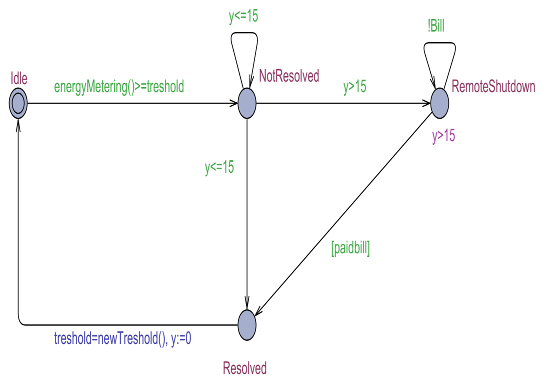


Figure 4: Model of T_2.

The voltage variation is defined in relation to other disturbances that can affect the electric network. It is generally defined as a decrease followed by an increase (sometimes the opposite, but less frequently) of the electric intensity. We describe in Figure 5 the model of the task T_4 allowing the stabilization of the consumed energy. The operation of stabilization is based on the use of the stored energy derived from green renewable sources. In the case of voltage fluctuation the battery is checked in the state BatteryCheck and the compensation process begins.

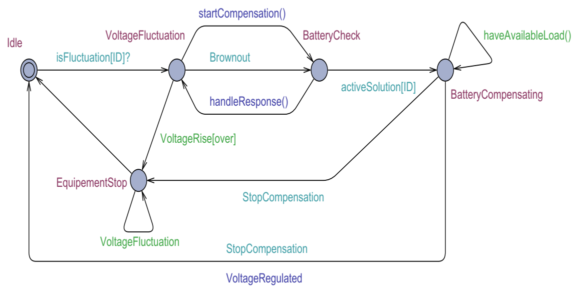


Figure 5: Model of T_4.

5.2 Implementation of the X-SM Simulation Tool

The simulation is a mandatory step for the final deployment of any system to be represented by a model. In our research work, a simulator is developed for the simulation of all these services that encode the electric meter of our company Cynapsys. We developed an environment X-SM for the simulation of a STM32F4 based smart meter. Our tool is essentially a program that allows the counting (a)the total power consumption of all devices that are connected to the electric meter, (b) the execution of any remote shutdown of the electrical energy in the case of exceeding the threshold of consumption, (c) the management of promotional offers proposed by producers,(d) and finally the stabilization of the energy variation. The main interface of our tool presents the different services offered by the Smart Meter and the various devices connected to it. To test the microcontroller, we assume three types of devices: a lamp, a refrigerator and a boiler with their characteristics and parameters such as the device name, the power, the device state (on or off), the priority, the shutdown condition, the output number (identifier) and the period of use.

The power consumption counting service, oper-

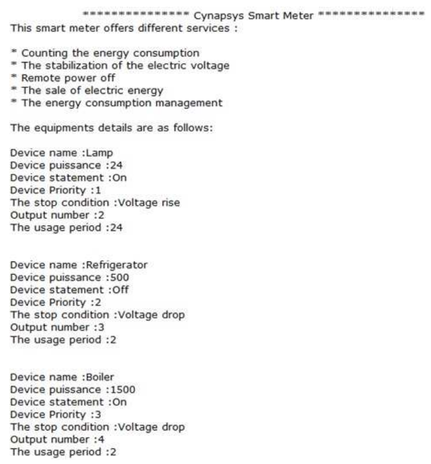
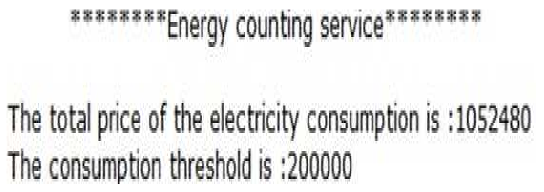

```
               =============== Cynapsys Smart Meter ===============
This smart meter offers different services :

* Counting the energy consumption
* The stabilization of the electric voltage
* Remote power off
* The sale of electric energy
* The energy consumption management

The equipments details are as follows:

Device name :Lamp
Device puissance :24
Device statement :On
Device Priority :1
The stop condition :Voltage rise
Output number :2
The usage period :24


Device name :Refrigerator
Device puissance :500
Device statement :Off
Device Priority :2
The stop condition :Voltage drop
Output number :3
The usage period :2


Device name :Boiler
Device puissance :1500
Device statement :On
Device Priority :3
The stop condition :Voltage drop
Output number :4
The usage period :2
```

Figure 6: Tool Main Interface.

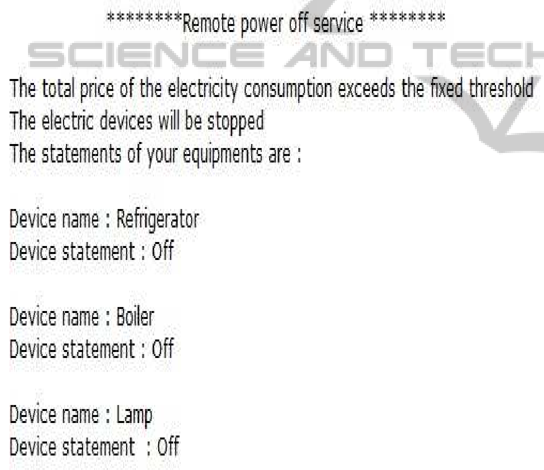
ates in real-time and provides the equivalent amount of the consumed quantity. It stills compared to the fixed consumption threshold as shown in the figure below.



```
********Energy counting service********

The total price of the electricity consumption is :1052480
The consumption threshold is :200000
```

Figure 7: Energy Counting Service Interface.

The remote shutdown service interface informs the user by SMS and email, that the consumption threshold is exceeded. It performs the shutdown operation and changes the state of the device from On to Off. The user is informed in parallel about the electricity shutdown so that he can pay his unpaid bills, or rationalize his energy consumption.

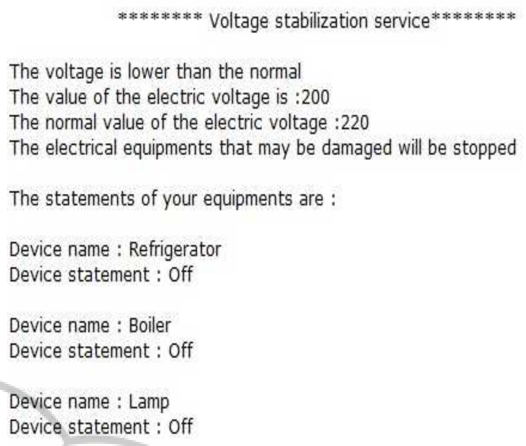


```
********Remote power off service ********

The total price of the electricity consumption exceeds the fixed threshold
The electric devices will be stopped
The statements of your equipments are :

Device name : Refrigerator
Device statement : Off

Device name : Boiler
Device statement : Off

Device name : Lamp
Device statement  : Off
```

Figure 8: Remote Shutdown Service Interface.

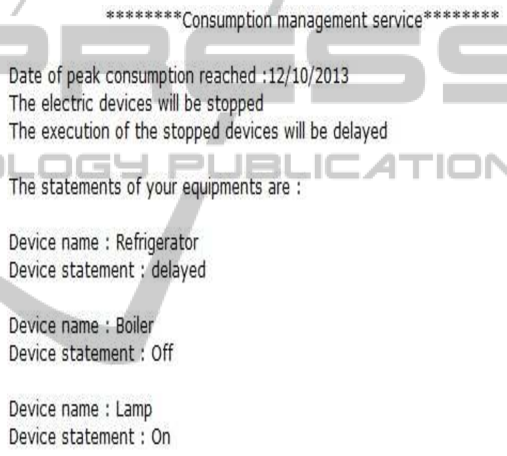
The voltage stabilization is a capital service, it helps to maintain the electrical equipment in the house against voltage variations that can damage them. The interface of the tool detects a variation and identifies its type, whether it is an increase or decrease in voltage. It informs the user by SMS and email that a shutdown must be performed. After the Shutdown the system pass to the compensation solution by using the amount of energy stored in the batteries produced from supposed renewable sources at home.

Below the management peak consumption service interface proposed by our environment X-SM, a peak energy consumption is generated by a strong demand for electricity and an imbalance in supply and demand. The smart meter allows smooth management of these peak periods through an anticipation mechanism.



```
******** Voltage stabilization service********

The voltage is lower than the normal
The value of the electric voltage is :200
The normal value of the electric voltage :220
The electrical equipments that may be damaged will be stopped

The statements of your equipments are :

Device name : Refrigerator
Device statement : Off

Device name : Boiler
Device statement : Off

Device name : Lamp
Device statement : Off
```

Figure 9: Voltage Stabilization Service Interface.



```
********Consumption management service********

Date of peak consumption reached :12/10/2013
The electric devices will be stopped
The execution of the stopped devices will be delayed

The statements of your equipments are :

Device name : Refrigerator
Device statement : delayed

Device name : Boiler
Device statement : Off

Device name : Lamp
Device statement : On
```

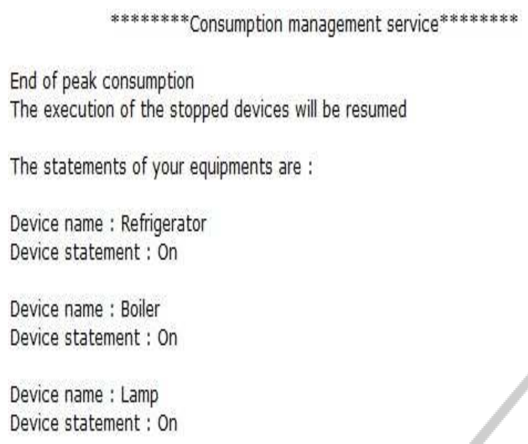
Figure 10: Peak Consumption Management Service Interface.

After crossing the period of peak consumption, the user regains his normal operating mode. The stopped devices will resume there normal executions. The system of anticipation / reaction offers an innovative solution that provides a huge gains and preserve the equipment in houses, which can be damaged by sudden and unexpected cuts. Below the end of the peak consumption interface.

5.3 Experimentation

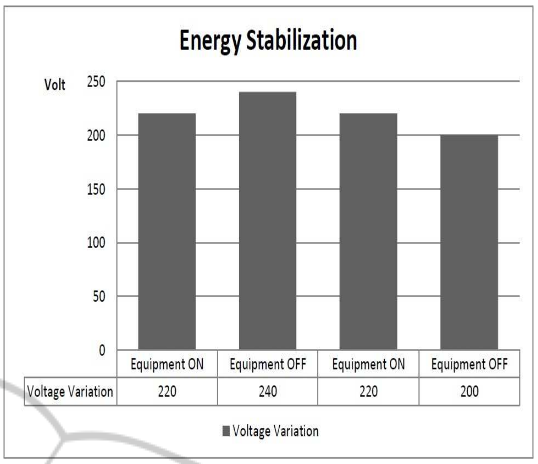
The figure 12 shows a comparison between the results obtained before and after the contribution of our proposal. We assume that the daily consumption of electricity in regular time is estimated at 4 KWH. During peak consumption this value should not exceed 2 KWH per day. The reconfiguration applied to our electric meter, gives rise to direct cuts that should not exceed the fixed amount of electricity.

After connecting the STM32 F4 card to our X-



```
********Consumption management service********

End of peak consumption
The execution of the stopped devices will be resumed

The statements of your equipments are :

Device name : Refrigerator
Device statement : On

Device name : Boiler
Device statement : On

Device name : Lamp
Device statement : On
```

Figure 11: End of Peak Consumption Service Interface.



	Equipment ON	Equipment OFF	Equipment ON	Equipment OFF
Voltage Variation	220	240	220	200

Figure 13: Energy Stabilisation Histogram.

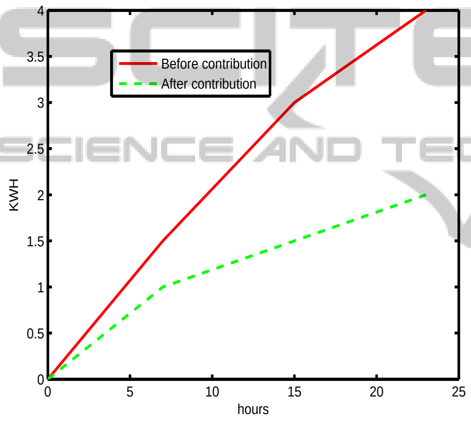


Figure 12: Peak Consumption Management Service.

SM simulator, the data acquisition is performed via a communication and information collecting protocol. Our simulation environment shows the reaction of the electric smart meter face to the detection of voltage variation. The reconfiguration process enables the service of voltage stabilization via the compensation through renewable production sources and electrical equipment shutdown, so they are not damaged.

6 CONCLUSIONS

This paper deals with an agent-based dynamic reconfiguration of a microcontroller solution. Our approach is started by a complete study of the actual electric meter in the field of energy development and applied to the Cynapsys developed smart one. The reconfiguration agent allows the embedded systems to change its software or hardware behavior at runtime according to user requirements or environmental changes. We propose in this work an embedded reconfiguration of the microcontrollers following the STM32 F4 technology the well-used in the world. To manage the reconfiguration process we propose an agent-based architecture, composed of the architectural level, Scheduling level, and data level. We present a detailed formalization, verification of nested state machine models and a software design version which is based on UML Class Diagram before the development of complete simulation tool applied to the Smart Meter developed at the German-French company Cynapsys. The proposed approach is original and is distinguished from the related works in this field, the product will now proceed to the production and marketing by the company Cynapsys. we will focus more in the future on the communication between electricity providers and smart meters, we will give more importance to the electricity metering based on the real consumption of the consumer, deploy this prototype in a smart city and finally the massive industrialization of this innovative product.

REFERENCES

Ahmad, S. (2011). Smart metering and home automation solutions for the next decade. In *International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*, pages 200–204.

Al-Safi, Y. and Vyatkin, V. (2007). An ontology-based reconfiguration agent for intelligent mechatronic systems. In *Third International Conference on Industrial Applications of Holonic and Multi-Agent Systems*.

Alur, R. and Dill, D. L. (1994). A theory of timed automata. In *Theoretical computer science*, pages 183–235.

Angelov, C., Sierszecki, K., and Marian, N. (2005). Design models for reusable and reconfigurable state machines. In *International Federation for Information Processing*, pages 04–17.

Bengtsson, J., Larsen, K., Larsson, F., Pettersson, P., and Yi, W. (1996). Uppaal: a tool suite for automatic verification of real-time systems. In *Hybrid Systems III*, pages 232–243.

Boukhannoufa, M. (2012). Adaptabilit et reconfiguration des systmes temps-rel embarqus. In *UNIVERSITE PARIS-SUD*.

Chauhan, A., Rajawat, A., and Patel, R. (2009). Reconfiguration of fpga for domain specific applications using embedded system approach. In *International Conference on Signal Processing Systems*, pages 438–442.

ERDF (2009). Linky documentation. In *Linky, le compteur nouvelle gnration*.

Gehin, A.-L. and Staroswiecki, M. (2008). Reconfiguration analysis using generic component models. In *IEEE Transactions on Systems, Machine and Cybernetics*, pages 152–163.

Hoenkamp, R. A. and Huitema, G. B. (2012). Good standards for smart meters. In *9th International Conference on the European Energy Market (EEM)*, pages 1–6.

J.H.Christensen, Vyatkin, V., Strasser, T., Valentini, A., and Zoitl, A. (2005). Theiec 61499functionblockstandard: Overviewof the secondedition. In *International Electrotechnical Commission*.

Kramer, J. and Magee, J. (1985). Dynamic configuration for distributed systems. In *IEEE Transactions on Software Engineering*, pages 424–436.

Otero, J., Wagner, F., and Carro, L. (2006). Reconfiguration of embedded java applications, parallel and distributed processing symposium. In *IPDPS, 20th International*.

Palshikar, G. K. (2004). An introduction to model checking. In *Tata Research Development and Design Centre*.

Pfeffer, M. and Ungerer, T. (2004). Dynamic real-time reconfiguration on a multithreaded java-microcontroller. In *Seventh IEEE International Symposium on Object-Oriented Real-Time Distributed Computing*, pages 86–92.

Pratl, G., Dietrich, D., Hancke, G., and Penzhorn, W. (2007). A new model for autonomous, networked control systems. In *IEEE Transactions on Industrial Informatics*.

Rooker, M., Sunder, C., Strasser, T., Zoitl, A., Hummer, O., and Ebenhofer, G. (2007). Zero downtime reconfiguration of distributed automation systems. In *Third International Conference on Industrial Applications of Holonic and Multi-Agent Systems*.

S.Depuru, Wang, L., Devabhaktuni, V., and Gudi, N. (2011). Smart meters for power grid challenges issues, advantages and states. In *Power Systems Conference and Exposition (PSCE), IEEE/PES*, pages 1–7.

STMicroelectronics (2013). Datasheet. In *STM32F4 Datasheet*.

Theiss, S., Vasyutynskyy, V., and Kabitzsch, K. (2009). Software agents in industry: A customized framework in theory and praxis. In *IEEE Transactions on Industrial Informatics*.

Thramboulidis, K. (2004a). Model integrated mechatronics: An architecture for the model driven development of mechatronic systems. In *2nd IEEE International Conference on Mechatronics*, pages 497–502.

Thramboulidis, K. (2004b). Using uml in control and automation: A model driven approach. In *2nd IEEE International Conference on Industrial Informatics*.

Vyatkin, V., Christensen, J., and Lastra, J. (2005). An open, object-oriented knowledge economy for intelligent distributed automation. In *IEEE Transactions on Industrial Informatics*, pages 04–17.

Wang, X., Khalgui, M., Li, Z., and Mosbahi, O. (2010). Automatic low-power reconfigurations of real-time embedded control systems. In *Technical Reprot, Systems Control and Automation Group School of Electro-Mechancial Engineering Xidian University*.

Zhang, J., Khalgui, M., Li, Z., and Mosbahi, O. (2013). R-tnces: A novel formalism for reconfigurable discrete event control systems. In *IEEE Transactions On Systems, Man, And Cybernetics, Part A: Systems And Humans*, pages 757–772.

Zhang, L. and Wang, Z. (2010). Design of embedded control system based on arm9 microcontroller. In *International Conference on Electrical and Control Engineering*, pages 3579–3582.

Zoitl, A., Lepuschitz, W., Merdan, M., and Vallee, M. (2010). A real-time reconfiguration infrastructure for distributed embedded control systems. In *IEEE International Conference Emerging Technologies and Factory Automation*.