Programming paradigm of a Microcontroller with Hardware Scheduler Engine and Independent Pipeline Registers – A software approach

Authors,
ph.d. eng. **Lucian ANDRIEȘ**
prof.dr.eng. **Vasile Gheorghiță GĂITAN**
ph.d. eng. **Elena-Eugenia Moisuc (Ciobanu)**

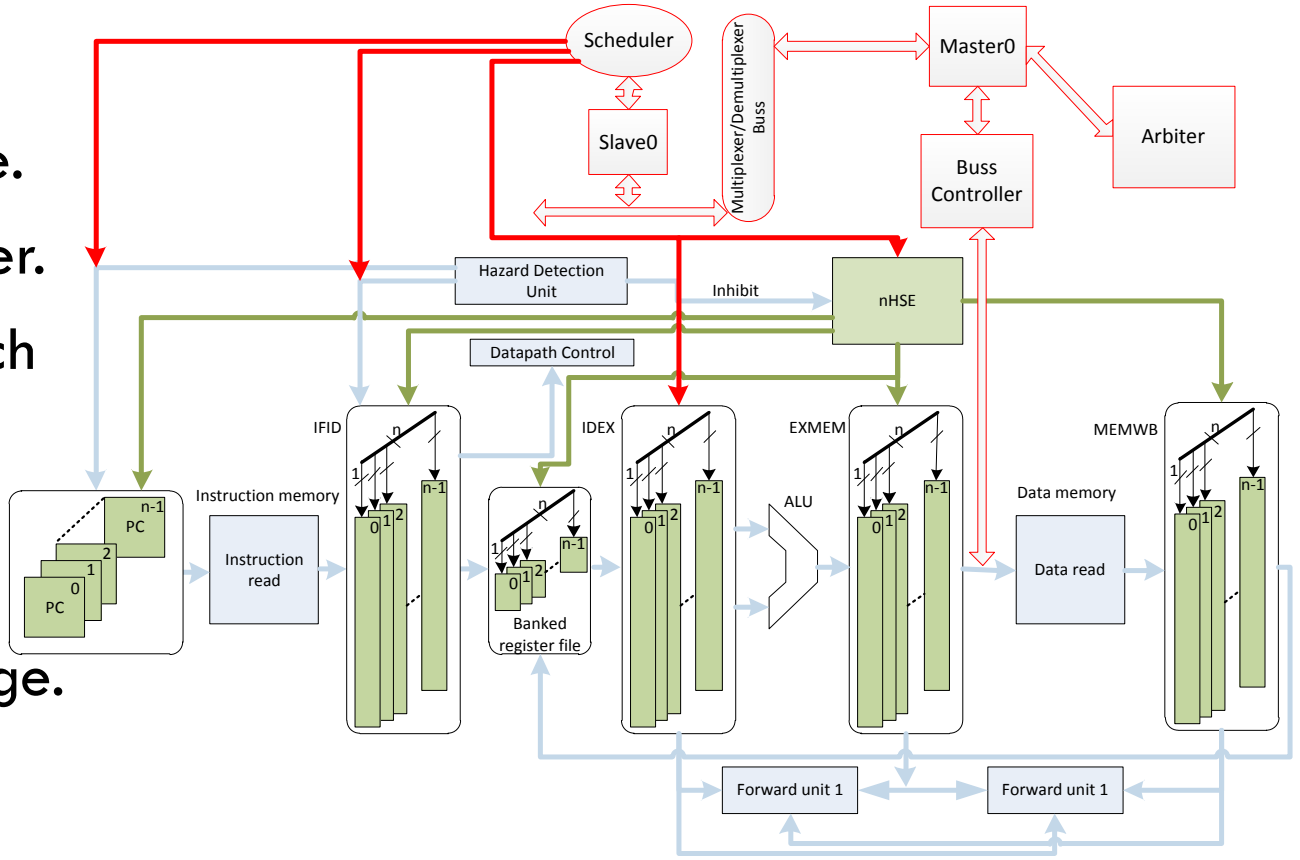
Cheile Gradistei 2015

Contents

1. Overview

The nMPRA architecture.

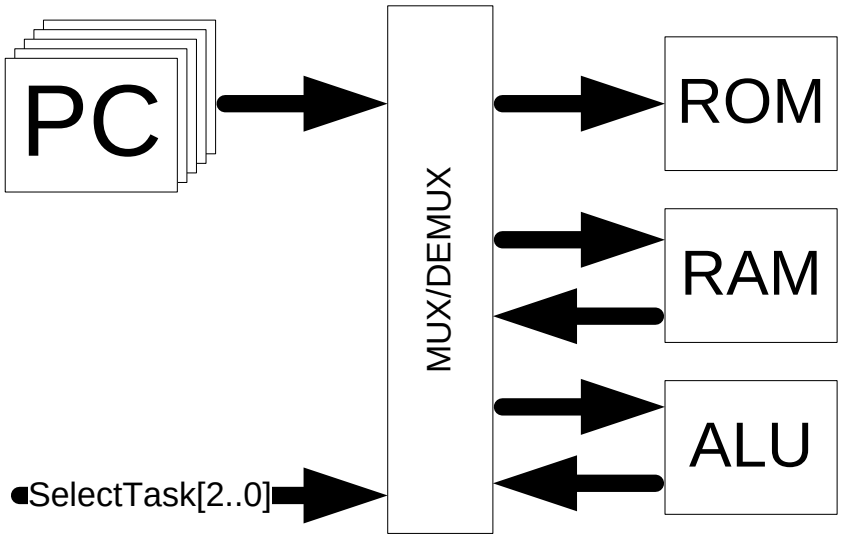
- PC – program counter.
- IFID – Instruction Fetch Instruction Decode stage.
- ID/EX – Instruction decode–execute stage.
- EX/MEM – execute-memory stage.
- MEM/WB – memory-write back stage.



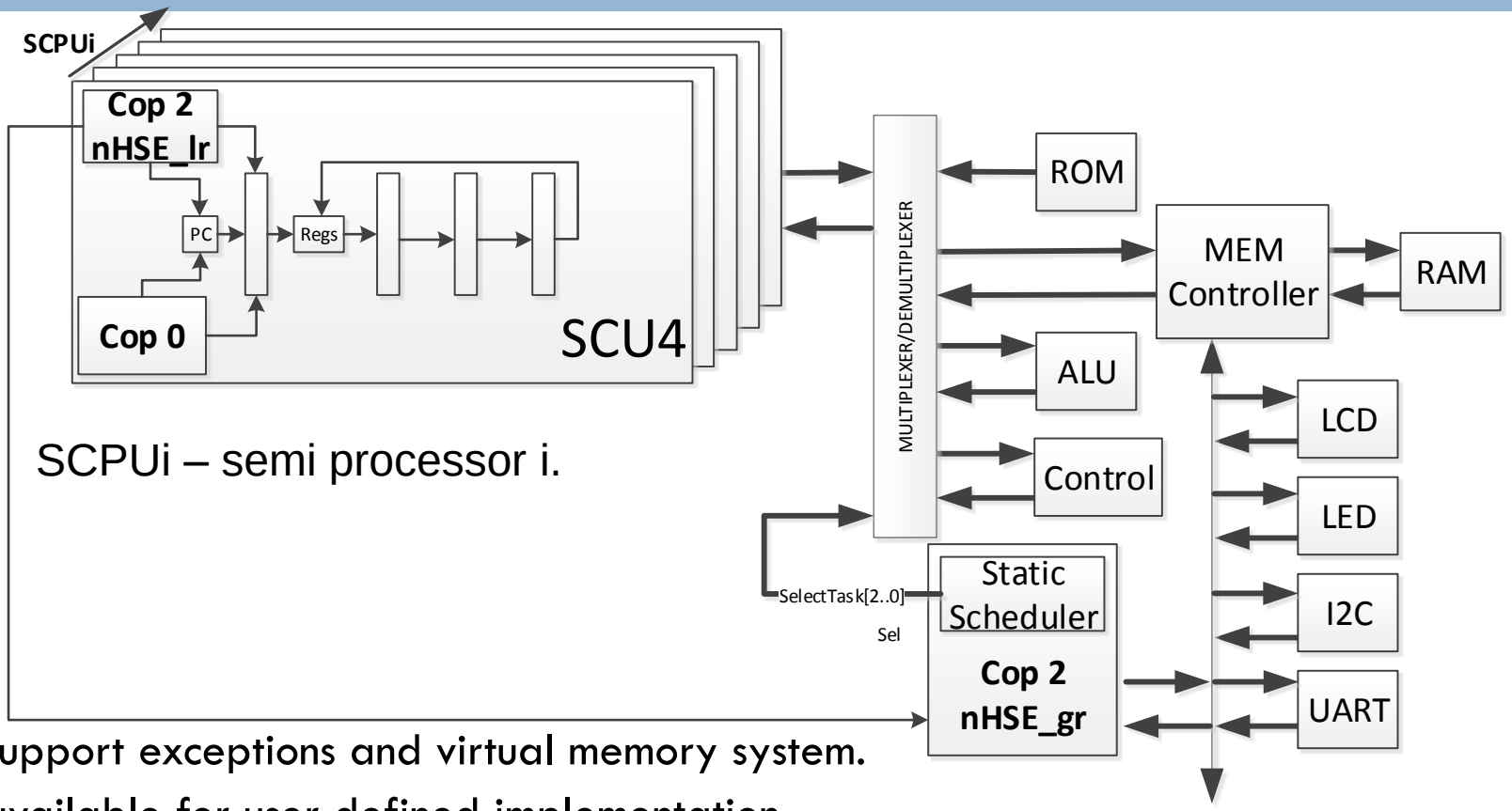
1. Overview

The simplified nMPRA architecture.

- The RAM, ALU and ROM are shared between multiple resources using only one big multiplexer/demultiplexer.
- Each task will share only RAM, ALU, ROM. The register files and the program counter will be multiplied. This approach will ensure that the data will be remain valid after every task switch.

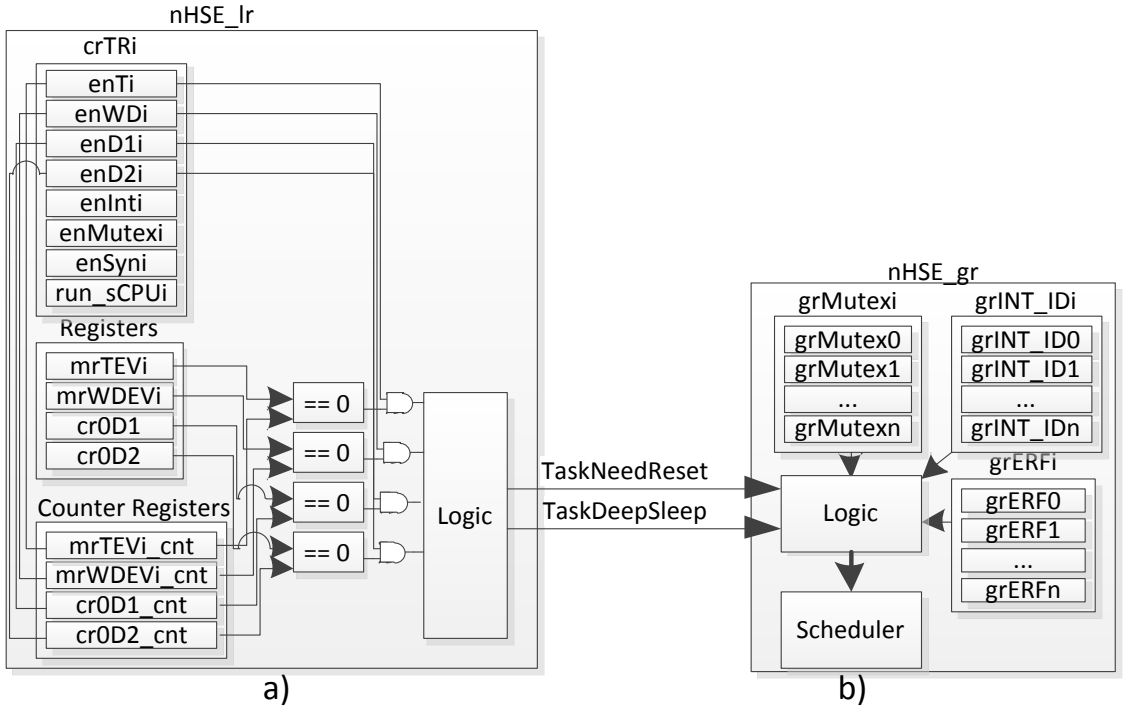


1. Overview



- COP0: support exceptions and virtual memory system.
- COP2: available for user defined implementation.
- nMPRA (multiple pipeline registers architecture for n tasks): offer support for hardware synchronization between tasks and peripherals.
- nHSE (hardware scheduler engine for n tasks): offer support for static and dynamic hardware scheduler for n tasks.

1. Overview

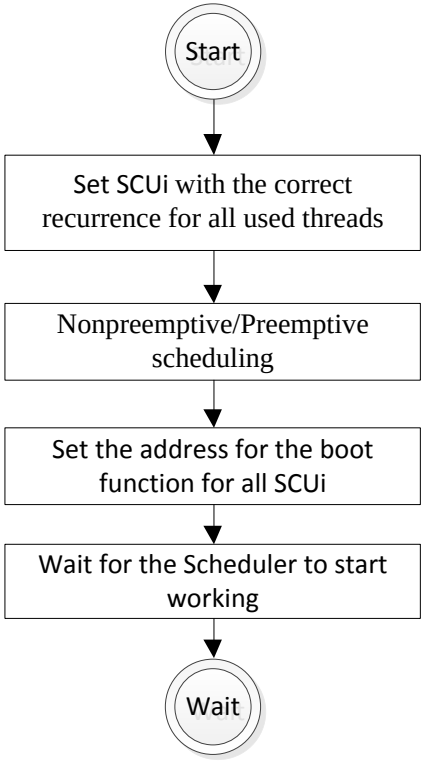


- *Alarms:* mrD1EVi, mrD2EVi.
- *Watch dog timer: mrWDEVi.*
- *Timers:* mrTEVi.

2. Programming paradigms

a) SCPUi initialization

- Each SCPUi must initialize its own registers and COP2 module correctly for a proper operation because these hardware resources are not shared.
- After RESET only SCPU0 is active and must be used to initialize the others SCPUi's.
- The first task that will start to execute code will be the task with less priority, in our case task1, followed by the task with higher priority.



2. Programming paradigms

b) Using shared variables between tasks

- Using global variables that are shared between SCPUi, the compiler might consider that the code generation can be optimize (internal implementation of the compiler) to use local variables that can be stored in registers.
- This case may appear when a variable is initializing globally and read in each task.
- *The optimize variable will be visible only in one task.*
- *In order to use shared variables between tasks we must force to variables to be stored in RAM.*

2. Programming paradigms

b) Using shared variables between tasks

- When only one variable is used, inside a function, the register $8 (t0) is used to store the value.

C language	Generated assembler
1. uint32_t localVariable = 0;	1. move t0,a0 2. addiu t0,t0,0

2. Programming paradigms

c) *Interfacing with COP2 (nHSE_lr)*

- The MIPS32 architecture has provided a means by which can be upgraded to do custom computation. In this case we discuss about the COP2 that has some standard assembler instruction that can be used to interface with:
  - mtc2 – move word to COP2
  - mfc2 – move word from COP2

2. Programming paradigms

c) Interfacing with COP2 (nHSE_Ir)

- Writing to nHSE_Ir registers

C language

```
1.    //force variable _reg to be placed in
2.    //register $8
3.    volatile register
4.    unsigned long __reg asm ("$8");
5.
6.    inline void crTR_SET(unsigned long val){
7.        //val will be assigned to reg $8
8.        __reg = (unsigned long)(val);
9.        asm("addiu  $8,$8, 0\n\t");
10.       //load reg $8 value to reg $0(crTR)
11.       asm("mtc2 $8, $0");
12.   }
```

Generated assembler

```
1.    00000e14 <crTR_SET>:
2.    e14: 00804021     move t0,a0
3.    e18: 25080000     addiu t0,t0,0
4.    e1c: 03e00008     jr ra
5.    e20: 48880000    mtc2 t0,$0
```

2. Programming paradigms

c) Interfacing with COP2 (nHSE_Ir)

- Reading from nHSE_Ir registers

C language

```
1.    //force variable _reg to be placed in
2.    //register $8
3.    volatile register
4.    unsigned long __reg asm ("$8");
5.
6.    inline unsigned long crTR_GET(void){
7.       //load reg $0(crTR) value into _reg
8.       //variable
9.        asm("mfc2 $8, $0");
10.       return __reg;
11.   }
```

Generated assembler

```
1.    00000e24 <crTR_GET>:
2.    e24: 48080000    mfc2 t0,$0
3.    e28: 03e00008    jr ra
1.    e2c: 01001021    move v0,t0
```

3. Conclusions

- In this paper we created the development environment for the nHSE architecture that is functional and can be used to create applications. The drawback of this architecture is a more restrictive environment that the one with a single core, because the gcc compiler will generate code only for single core microcontrollers
- Also with this architecture we can create more secure and hard to crack applications, when it comes to third party, because no SCPUi can alter the internal state of another SCPUi.

4. References

- Zhikao Ren, Coll. of Inf., Qingdao Univ. of Sci. & Technol., Qingdao, China, Chen Ye, Guozu Liu, “Application and Research of C Language Programming Examination System Based on B/S”, ISBN 978-1-4244-8627-4.
- Wu, Z., Comput. Lab., Cambridge Univ., UK, "Making C++ a distributed programming language", ISBN 0-8186-4430-3 http://en.wikipedia.org/wiki/GNU_toolchain.
- Gaitan, V.G.; Gaitan, N.C.; Ungurean, I., "CPU Architecture Based on a Hardware Scheduler and Independent Pipeline Registers," Very Large Scale Integration (VLSI) Systems, IEEE Transactions on , vol.PP, no.99, pp.1,1, ISSN : 1063-8210, doi: 10.1109/TVLSI.2014.2346542.
- Andries, L.; Gaitan, G., "Dual priority scheduling algorithm used in the nMPRA microcontrollers," System Theory, Control and Computing (ICSTCC), 2014 18th International Conference , vol., no., pp.43,47, 17-19 Oct. 2014, doi: 10.1109/ICSTCC.2014.6982388.
- Gaitan, N.C.; Gaitan, V.G.; Moisuc, E.-E.C., "Improving interrupt handling in the nMPRA," Development and Application Systems (DAS), 2014 International Conference on , vol., no., pp.11,15, 15-17 May 2014 doi: 10.1109/DAAS.2014.6842419.
- Gaitan, N.C.; Andries, L., "Using Dual Priority scheduling to improve the resource utilization in the nMPRA microcontrollers," Development and Application Systems (DAS), 2014 International Conference on , vol., no., pp.73,78, 15-17 May 2014, doi: 10.1109/DAAS.2014.6842431.
- Moisuc, E.-E.C.; Larionescu, A.-B.; Gaitan, V.G., "Hardware event treating in nMPRA," Development and Application Systems (DAS), 2014 International Conference on , vol., no., pp.66,69, 15-17 May 2014 doi: 10.1109/DAAS.2014.6842429.
- Moisuc, E.-E.C.; Larionescu, A.-B.; Ungurean, I., "Hardware event handling in the hardware real-time operating systems," System Theory, Control and Computing (ICSTCC), 2014 18th International Conference, vol., no., pp.54,58, 17-19 Oct. 2014, doi:10.1109/ICSTCC.2014.6982390.


Tingki
Komapsumnida
Shukuria
atu
Paldies
Tashakkur
gozaimashita
Maake
bolzin
Ekhmet
Mehrbani
Arigato
You
Dankscheen
Yaqhanyelay
Efcharisto
suksama
Merci
Gracias
Shukria
Thank
Biyan
Grazie
Juspaxar

Submission number	138
Authors or proposers	**Andries*, Lucian (Stefan cel Mare University of Suceava) (59273)** **Galtan, Vasile Gheorghita (Stefan cel Mare University of Suceava) (58655)** **Moisuc (Ciobanu), Elena-Eugenia (Stefan cel Mare University of Suceava) (58709)**
Title	**Programming Paradigm of a Microcontroller with Hardware Scheduler Engine and Independent Pipeline Registers – a Software Approach**
URL for supplementary information	
Type of submission	Contributed paper
Received on	June 4, 2015
Code	
Keywords	Embedded Systems
Abstract at initial submission	In computer science, for embedded exists only two types of microcontrollers that can be used to develop a working system. You can use a single core or a multicore which is much faster but will not be the equivalent of a single core multiple with the numbers of cores, because a small part from the power will be used for inter process communications. Our approach is a little bit different because we use a single core CPU that have a number of finite task that act like different CPU's. In this new architecture there is no need for inter process communication because the processor is a single core and the hardware tasks use the same resources as the others. The peripherals of this architecture will improve interrupt latencies and task switching times, what makes this microcontroller the best choice when it comes in interrupt response time.
Attachments	Copyright Transfer Form (attachment to final submission). Status: Received
Profile	Contributed Papers
Status	**Final version received**
Date of latest decision or action	August 5, 2015
Abstract	In computer science, for embedded field only two types of microcontrollers exists, that can be used to develop a working system. You can use a single core or a multicore which is much faster but will not be the equivalent of a single core multiple with the numbers of cores, because a small part from the power will be used for inter process communications. Our approach is a little bit different because we use a single core CPU that have a number of finite task that act like different CPU's. In this new architecture there is no need for inter process communication because the processor is a single core and the hardware tasks use the same resources as others. The peripherals of this architecture will improve interrupt latencies and task switching times, what makes this microcontroller the best choice when it comes in interrupt response time.
Number of pages in final manuscript	6
Copyright transferred	
Session	Embedded Systems (Regular Session)
Schedule code	ThA5.6
Scheduled time of presentation	Thursday October 15, 2015 10:50–12:50 Room 5 12:30–12:50 The session, schedule code and scheduled time of presentation, if present, are tentative. Please refer to the final program

pp. 693-698

Postolache, Mihai Gheorghe Asachi Tech. Univ. of Iasi

CAN (Controller Area Network) and TIA-485 are two of the most used standards in fieldbus systems. While CAN ISO IS-11898 includes complete data link layer specifications on top of its physical layer, TIA-485 only addresses the physical layer of the 7-layer OSI model. Other communication parameters like speed, format, and data transmission protocol are not specified by RS-485 in order to provide interoperability of similar devices from different manufacturers. After a brief introduction and comparison of the two communication protocols, the paper investigates to what extent the CANopen specification - an application level protocol developed primarily for CAN networks - should be used in TIA-485-based fieldbus networks in order to provide the user an application programming interface (API) independent of the physical layer. Selected and customized CANopen services have been implemented and tested on a network of microcontroller-based stations equipped with CAN ISO 11898 and TIA-485 communication interfaces.

12:10-12:30 ThA5.5

Future House Automation, pp. 699-704

Florea, Adrian Lucian Blaga Univ. of Sibiu
Bancioiu, Iosif Lucian Blaga Univ. of Sibiu

In this paper, we propose the future house automation, a PLC-based embedded system that aims reducing the house energy consumption by optimizing the entire hardware assembly and software algorithms. The project started from the idea of designing a self-controlled house, to increase user's comfort in his daily environment, reducing the cost and optimizing the energy consumption. Our embedded application represents a green solution into a growing number of environmentally aware consumers, very suitable for the market of energy-efficient control systems. We provide a cheap solution for developing by everyone its own automation system control house. Therefore, our project contributes for helping the elderly, which represents another social challenge with global character.

12:30-12:50 ThA5.6

Programming Paradigm of a Microcontroller with Hardware Scheduler Engine and Independent Pipeline Registers – a Software Approach, pp. 705-710

Andries, Lucian Stefan Cel Mare Univ. of Suceava
Gaitan, Vasile Gheorghita Stefan Cel Mare Univ. of Suceava
Moisuc (Ciobanu), Elena-Eugenia Stefan Cel Mare Univ. of Suceava

In computer science, for embedded field only two types of microcontrollers exists, that can be used to develop a working system. You can use a single core or a multicore which is much faster but will not be the equivalent of a single core multiple with the numbers of cores, because a small part from the power will be used for inter process communications. Our approach is a little bit different because we use a single core CPU that have a number of finite task that act like different CPU's. In this new architecture there is no need for inter process communication because the processor is a single core and the hardware tasks use the same resources as others. The peripherals of this architecture will improve interrupt latencies and task switching times, what makes this microcontroller the best choice when it comes in interrupt response time.

Chair: Petre, Emil, Emil	Univ. of Craiova
Co-Chair: Postolache, Mihai	Gheorghe Asachi Tech. Univ. of Iasi

10:50-11:10 ThA5.1

Specialized Device – Slave – for Monitoring the Antihail Systems, pp. 673-678.

Sulea-Iorgulescu, Constantin	Univ. of Craiova
Petre, Emil, Emil	Univ. of Craiova

11:10-11:30 ThA5.2

HVAC Control Methods - a Review, pp. 679-686.

Belic, Filip	TEO-Belisce
Hocenski, Zeljko	Univ. of Osijek
Sliskovic, Drazen	Univ. of Osijek

11:30-11:50 ThA5.3

Smart Home System Based on Comparative Analysis among AODV and DSDV Protocols in MANET, pp. 687-692.

MOHAMAD, OMAR ABDULWAHABE	Univ. Pol. of Bucharest
Hameed, Rasha Talal	Univ. Pol. of Bucharest
Ţăpuş, Nicolae	Univ. Pol. of Bucharest

11:50-12:10 ThA5.4

CANopen Implementation Using BCP Protocols Over TIA-485 in Networked Embedded Systems, pp. 693-698.

Postolache, Mihai	Gheorghe Asachi Tech. Univ. of Iasi

12:10-12:30 ThA5.5

Future House Automation, pp. 699-704.

Florea, Adrian	Lucian Blaga Univ. of Sibiu
Bancioiu, Iosif	Lucian Blaga Univ. of Sibiu

12:30-12:50 ThA5.6

Programming Paradigm of a Microcontroller with Hardware Scheduler Engine and Independent Pipeline Registers – a Software Approach, pp. 705-710.

Andries, Lucian	Stefan Cel Mare Univ. of Suceava
Gaitan, Vasile Gheorghita	Stefan Cel Mare Univ. of Suceava
Moisuc (Ciobanu), Elena-Eugenia	Stefan Cel Mare Univ. of Suceava

Technical Program for Friday October 16, 2015

FrA1 Room 1
Control Applications (Regular Session)

Chair: Precup, Radu-Emil	Pol. Univ. of Timisoara
Co-Chair: Paraschiv, Nicolae	Petroleum - Gas Univ. of Ploiesti

10:50-11:10 FrA1.1

On the Development of an Intelligent System for Particulate Matter Air Pollution Monitoring, Analysis and Forecasting in Urban Regions, pp. 711-716.

Oprea, Mihaela	Petroleum - Gas Univ. of Ploiesti
Ianache, Cornel	Petroleum-Gas Univ. of Ploiesti
Mihalache, Sanda Florentina	Petroleum-Gas Univ. of Ploiesti
Dragomir, Elia Georgiana	Petroleum - Gas Univ. of Ploiesti
Dunea, Daniel	VALAHIA Univ. of Targoviste
Iordache, Stefania	VALAHIA Univ. of Targoviste
Savu, Tom	Pol. Univ. of Bucharest

11:10-11:30 FrA1.2

Proceedings of
2015 19th International Conference on System Theory, Control and Computing (ICSTCC)

Joint conference SINTES 19, SACCS 15, SIMSIS 19

October 14 - 16, 2015

CheileGradistei - Fundata Resort, ROMANIA

Editors:

Sergiu Caraman

Marian Barbu

Răzvan Şolea

IEEE Catalog Number: CFP1536P-USB
ISBN: 978-1-4799-8480-0