Time- and angle-triggered real-time kernel

Damien CHABROL, Didier ROUX
KRONO-SAFE
Orsay, France
firstname.lastname@krono-safe.com

Vincent DAVID, Mathieu JAN, Moha AIT HMID
CEA LIST LaSTRE
Gif-sur-Yvette, France
firstname.lastname@cea.fr

Patrice OUDIN
Gilles ZEPPA
DELPHI DIESEL SYSTEMS
Blois, France
firstname.lastname@delphi.com

***Abstract* — Powertrain controllers are automotive applications that bring real-time constraints on software treatments based on the angular position of tooth in the motor. Theses constraints depend on the engine speed and can be as short as 100µsec. A time-triggered approach provides a predictable and reproducible execution of real-time systems but cannot cope with so tight constraints and does not allow to directly specifying the temporal behavior of the system based on angles. The contribution of this work is to present how the ability of the PharOS technology to combine several time domains (time and angle triggered) allows designing and executing powertrain controllers in a deterministic way on multi-core architectures. To this end, we present a prototype of a subset of a powertrain controller from Delphi based on PharOS.**

***Index Terms* — time-triggered, angle-triggered, automotive powertrain controller.**

I. INTRODUCTION

In the automotive market, the Euro emission norm aims at reducing pollutants from new wheeled vehicles whereas upcoming automotive standard aims to improve confidence in the embedded product. In this work, we focus on controllers used in diesel engines for the real time adjustments of parameters to help powertrain systems to operate efficiently, reliably and economically. The temporal behavior of such systems shall be mastered in order to reduce their development costs as well as to reach the level of confidence required by automotive safety standards. A predictable and reproducible execution has to be ensured on multi-core architectures that are targeted for next generation of powertrain controllers. New solutions must be found to meet these needs as current technologies are unable to solve these problems.

Powertrain controllers use different time domains (time and angle) in order to activate tasks. In [1][2], we present the PharOS technology and its support of a single time domain for the execution of hard-real time systems in a deterministic way as well as the ability to execute functions with very short reactivity constraints. The contribution of this paper is to show how the combination, in a deterministic way, of several time domains provided by PharOS is an answer to the needs of powertrain controllers. The adequacy of the execution model of PharOS and the expressive power of the need-oriented programming model have been verified by CEA LIST and Krono-Safe by developing a first prototype over a PowerPC MPC551x evaluation board.

978-3-9815370-0-0/DATE13/©2013 EDAA

It is based on the analysis of a subset of powertrain controllers from DELPHI and integrates one injector management with static and dynamic tooth synchronization.

II. RELATED WORK

A main difference between classical Real-Time Operating Systems (RTOS) and PharOS lies in the programming models used. Most RTOS are indeed designed for a particular API, like POSIX, OSEK or ARINC 653. Using PharOS, no APIs or system calls are available. They are instead introduced by the PharOS off-line tool chain to ensure the fulfillment of the specified temporal behavior.

TTPOS [5] is an RTOS for hard real-time systems that implement an execution model based on a single time domain, physical time. Besides, critical part of the scheduling is left under the programmer's responsibility. ARINC 653 compliant RTOSes, such as Integrity [4], provide a spatial and temporal isolation of partitions (a set of tasks), while PharOS provides such isolation properties at a finer granularity (task). Again, only a single time domain exists based on physical time.

The closest work to ours is xGiotto [6]. The activation of task can be based on asynchronous events (e.g. not only using physical time). However, deadlines must be specified using physical time. Besides, there is always a delay between an event and the activation of tasks, as the system must first wait that previous instances of tasks finishes. Several time domains cannot therefore be combined.

III. PHAROS TECHNOLOGY

A PharOS application is viewed as a static set of communicating real-time tasks that interact in order to achieve their functionalities. A task is a set of jobs, which have precedence relationships expressed through deadlines based on a time scale. A deadline defines the latest date by which it must be finished as well as the earliest starting date of the next job. Each job has therefore a temporal window of execution which is automatically deduced from the temporal requirements specified with a semi-formal language. Note that the temporal width associated to each job can be different. Using PharOS all possible temporal behaviors are statically described. However, at run-time the followed execution path and therefore the associated temporal constraints can be chosen based on the values of variables. The rhythm of a task may be multiple, periodic or not, regular or not.

A task executes either in a Time-Triggered (TT) domain, in which temporal constraints are defined using physical clocks, or in eXternal-Triggered (xT) domain, in which temporal constraints are defined using a clock based on an external event to the system. Each domain therefore defines its own time scale. An xT domain allows to associate temporal constraints to

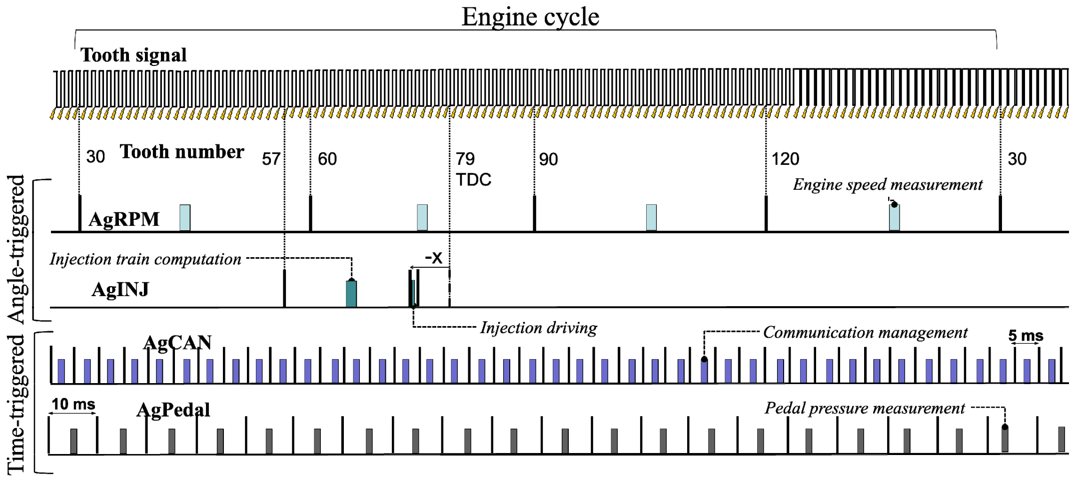

Figure 1. Engine controller temporal diagram

functional code that are linked to the occurrence of external events, whose behavior change from time to time. Release date and deadline cannot therefore be defined using physical clocks in a TT domain as only intervals of when these external events can occur are known. An expected temporal behavior for a source interrupt of a domain must be specified in order to ensure that the sizing demonstration can still be made and to detect any deviation from the expected behavior at run-time. The specification of the expected temporal behavior can also be made for a given set of ranges of values for an application domain variable. For each range, the worst-case and the best-case for the occurrence of the source interrupt must be specified. Finally, communication between domains is supported but due to space constraints we do not provide more details.

In addition, at each time occurrence of the source interrupt of an xT domain a hook function can be executed. The goal is to allow application designers to implement a functional processing that is specific to a given application. It is up to the applicative function to analyze the event, to validate or not this event and potentially to interact with its environment by for instance sending a message in response to data that were received by the peripheral that generates the source interrupt. The applicative function can also freeze the execution of the domain in order to wait for some conditions to occur. This functionality can be used to synchronize the beginning of the execution or suspend the execution of a domain on some event, defined by the application domain.

IV. POWERTRAIN CONTROLLERS WITH PHAROS

The diesel engine management system mixes angle-triggered and time-triggered processing. Angle-triggered functions are run according to the engine speed given by tooth events while time-triggered functions are periodic. In such system, accuracy and timeliness are two key features to fulfill in order to ensure efficient and economic engine difficult to control and demonstrate witch current technologies. Thanks to PharOS technology, these requirements are ensured by construction through a seamless process. Indeed, mixed TT & xT domains help to describe software architecture with temporal and logical constraints that are directly transposed without distortion of the upstream needs.

The injection management is compounded of several processing that are executed on a predefined teeth number (static) or this number can be dynamically computed (dynamic) in order to compute the injection sequence according to the torque demand, the injection timing demand and the fuel quantity. According to engine speed, tooth occurrence interval is then ranged between 200 microseconds (5000 rpm) and 4 milliseconds (400 rpm). As illustrated in Figure 1, from engine speed computed by AgRPM and pedal pressure measured by AgPedal, AgINJ computes the injection sequence then drives the injector near from the Top Dead Center (TDC) tooth. In this final step, AgINJ is executed on a dynamic tooth configured from an ahead mapping such as x teeth before TDC according to the actual engine speed. Communication among domains is possible such as illustrated by AgPedal and AgINJ.

The communication management (AgCAN) is translated in one periodic task triggered at 5 milliseconds ensuring by construction that no frame is lost and communication protocol is implemented in an efficient and accurate way.

V. CONCLUSION

In the automotive field, new solutions must be found to design and execute powertrain controllers on multi-core architectures with a predictable and reproducible behavior. We have presented how the execution model of PharOS that combines several time domains can be used to successfully fulfill requirements of such type of application, through the development of a subset of a powertrain controller from DELPHI using the PharOS technology.

Thanks to these relevant preliminary results, Delphi aims to continue the evaluation of the PharOS technology but in a whole controller over a representative powertrain multicore microcontroller. In particular, we plan to demonstrate the use of several ranges of speed for an engine, a similar problem to the one described in [7] (multi-mode support) and not only for the schedulability analysis but also at run time. We also plan to demonstrate the use of different temporal specifications for a same task in order to support the loss of domains.

The PharOS technology is today industrialized by KRONO-SAFE in its KRON-OS product and will be available on off the shelf at end of 2013. Note that an execution model that combines several time domains is also required by other domain fields, such as medium voltage protection relays [3].

REFERENCES

[1] D. Chabrol and C. Aussagues and V. David. A spatial and temporal partitioning approach for dependable automotive systems, IEEE Conf. on Emerging Technologies Factory Automation (ETFA 2009). Toulouse, France. Sept. 2009.

[2] M. Lemerre, E. Ohayon, D. Chabrol, M. Jan and M. Jacques. Method and Tools for Mixed-Criticality Real-Time Applications within PharOS. Proc. of the 14th IEEE Intl Symp. on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops (ISORCW'11). Newport Beach, CA, USA. March 2011.

[3] M. Jan and V. David and J. Lalande and M. Pitel. Usage of the safety-oriented real-time OASIS approach to build deterministic protection relays. IEEE Fifth International Symposium on Industrial Embedded Systems (SIES 2010). Pages 128-135. University of Trento, Italy, July 2010.

[4] Green Hills software inc. Integrity. http://www.ghs.com/products/rtos/integrity.html

[5] TTech. TTPOS. http://www.tttech.com/?id=233

[6] Arkadeb Ghosal and Thomas A. Henzinger and Christoph M. Kirsch and Marco A. A. Sanvido. Event-driven programming with logical execution times. Proc. of HSCC 2004, Lecture Notes in Computer Science.

[7] Mircea Negrean, Rolf Ernst, and Simon Schliecker, "Mastering Timing Challenges for the Design of Multi-Mode Applications on Multi-Core Real-Time Embedded Systems" in 6th International Congress on Embedded Real-Time Software and Systems (ERTS), (Toulouse, France), February 2012.