

Sistemas de Tiempo Real Distribuidos Robots y Microcontroladores

Fernando Romero, Mariano Méndez , Diego Encinas,
Santiago Medina, Armando De Giusti¹, Fernando G. Tinetti²

Instituto de Investigación en Informática LIDI (III-LIDI)
Facultad de Informática – Universidad Nacional de La Plata

{fromero, mmendez, dencinas, smedina, degiusti, fernando}@lidi.info.unlp.edu.ar

Resumen

El objetivo de esta línea de investigación es el estudio y desarrollo de sistemas de software que poseen restricciones temporales, como son los Sistemas de Tiempo Real (STR), atendiendo en especial los aspectos relacionados con planificadores y comunicaciones, realizando implementaciones de robots y en general sistemas de adquisición y control utilizando microcontroladores.

Contexto

Esta línea de Investigación forma parte del proyecto "Arquitecturas multiprocesador en HPC: software de base, métricas y aplicaciones" (11/F018) del Instituto de Investigación en Informática LIDI acreditado por la UNLP.

Palabras Claves: Tiempo Real, Sistemas Embebidos, Comunicaciones, Red de Sensores, Robots, Microcontroladores.

1. Introducción

Los Sistemas de Tiempo Real (STR) son sistemas que interactúan con su entorno

físico y deben satisfacer restricciones de tiempo de respuesta explícitas [14] [7] [15] [8] [23]. El requerimiento de satisfacer los plazos llega a ser tanto o más importante que la corrección lógica de la respuesta, que puede incluso relajarse en función de cumplir los plazos. Son en su mayoría Sistemas Embebidos, en cuanto controlan un sistema, por lo general físico, adquiriendo información de las variables del sistema controlado a través de sensores, detectores y realizando el control a través de distintos tipos de actuadores. Muchos de estos sistemas son móviles, con lo que el aspecto consumo se debe tener en cuenta. Esto requiere el uso de hardware con bajo consumo energético, tales como los microcontroladores. En los microcontroladores se tiene una menor capacidad de cómputo, y a la vez una sofisticada sección de entrada y salida. Asimismo, los SOTR (Sistemas Operativos de Tiempo Real) que utilizan están optimizados para lograr baja latencia en la elaboración de las respuestas frente a señales externas, por lo general a través de interrupciones de hardware. Parte de las aplicaciones actuales de estos sistemas pertenecen al

¹ Investigador CONICET

² Investigador CIC Prov. de Buenos Aires.

campo de la robótica, que combina el control con cierto grado de inteligencia [5] [14], creando sistemas que llevan a cabo tareas que ni las máquinas comunes ni los seres humanos pueden realizar.

La satisfacción de los requerimientos temporales de los plazos de estos sistemas implica una planificación rigurosa de la ejecución de tareas. Este es el punto central del diseño de aplicaciones de tiempo real. Es objetivo de esta línea de investigación abordar este tema, desde el punto de vista teórico y experimental. Esta planificación puede ser realizada en forma manual, durante el diseño del software, o utilizar herramientas automáticas para la misma, conocidas como planificadores, mayormente formando parte de los SOTR. El grado de complejidad de estos planificadores será acorde al grado de complejidad del sistema. Para este estudio se dispone de diversas plataformas de hardware y software para la construcción de sistemas reales, de diferente grado de complejidad, desde sistemas basados en microcontroladores simples sin SOTR, como Arduino hasta sistemas basados en PC, pasando por microcontroladores sofisticados capaces de soportar SOTR basados en Linux, tales como los Freescale [32] y CIAA [27]. Los sistemas más simples implican abordar el estudio del nuevo paradigma llamado “Internet of Things” (IoT) [22] [21] [3], en el cual cosas u objetos (sensores, actuadores, tabletas, teléfonos celulares, etc.) que se conectan en un esquema de identificación única son capaces de interactuar unos con otros y cooperar.

En las diferentes implementaciones que se llevan a cabo en esta línea de investigación se trabaja de manera experimental con mini-robots, plataformas de cómputo (PC, Arduino, Intel Galileo, Freescale, CIAA, etc.), con diferentes sistemas

operativos (Linux RT-Preempt, Freertos, MQX, OSEK-OS, etc.) [20] [10] y sistemas de comunicación basados en los protocolos I2C, CANBUS [29] [30] [31] [32] [33] y MODBUS [28].

Tratándose en algunos casos de sistemas distribuidos, se experimentan sistemas de microcontroladores [11] interconectados con PC, en forma cableada e inalámbrica (radiofrecuencia, Bluetooth, WiFi). Esto implica el estudio de protocolos de comunicación diseñados para este tipo de aplicaciones, tales como CANBUS y MODBUS. Con respecto a los micro-robots, se utilizan algunos de diseño y fabricación propia como también los sofisticados robots Khepera [24] [25] [26]. Una característica de los SOTR es tener un alto grado de fiabilidad, por lo que el estudio de la detección y control de condiciones de falla es un tema importante [2] [17].

2. Líneas de Investigación y Desarrollo

Se plantean como temas de estudio:

- Verificación y validación del hardware, donde pueden encontrarse detalles o resultados de simulaciones que impliquen modificaciones [9] [10].
- Construcción de red de sensores en la plataforma de microcontroladores interactuando con una PC, utilizando MODBUS y CANBUS como protocolo de comunicaciones [1] [18].
- Diseño y construcción de vehículos no tripulados y sistema de navegación [12].
- Estudio de diferentes plataformas y combinaciones de las mismas.
- Odometría con robots móviles, en particular con los robots Khepera [6] [16] [4].

3. Resultados Esperados y Objetivos

De acuerdo con las tareas desarrolladas y a desarrollar, los resultados se enfocarán en varias direcciones relacionadas con los sistemas de tiempo real:

- Construcción y uso de robots tipo vehículo autónomo para el estudio de Odometría.
- Medición de consumo de diferentes tipos de microcontroladores bajo distintas condiciones de uso.
- Estudio de distintas plataformas de hardware: Arduino, Intel Galileo, CIAA, FreeScale, Rasperry Pi, entre otras.
- Construcción y estudio de dos redes de Sensores, empleando rf para las comunicaciones en una y los protocolos CANBUS y MODBUS en otras.

4. Formación de Recursos Humanos

En base a estos temas se están desarrollando trabajos de varios alumnos encuadrados en el marco de la Convocatoria a Proyectos de Desarrollo e Innovación de la Fac. de Informática de la UNLP como también de PPS (Práctica Profesional Supervisada) con la que concluyen sus estudios los alumnos de Ingeniería en Computación.

5. Referencias

[1] Akyildiz, Ian F., Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. "Wireless sensor networks: a survey." *Computer networks* 38, no. 4 (2002): 393-422.

[2] Andersen, B. L. "Method of detecting systemic fault conditions in an intelligent electronic device." U.S. Patent 6,434,715, issued August 13, 2002.

[3] Atzori, L., A. Iera, G. Morabito. "The internet of things: A survey." *Computer networks* 54, no. 15 (2010): 2787-2805.

[4] Azizi, F., N. Houshangi. "Mobile robot position determination using data from gyro and odometry." In *Electrical and Computer Engineering*, 2004. Canadian Conference on, vol. 2, pp. 719-722. IEEE, 2004.

[5] Bekey, George A. *Robotics: state of the art and future challenges*. Imperial College Press, 2008.

[6] Borenstein, Johann, Liqiang Feng, "Gyrodometry: A new method for combining data from gyros and odometry in mobile robots." In *Robotics and Automation*, 1996. Proceedings., 1996 IEEE International Conference on, vol. 1, pp. 423-428. IEEE, 1996.

[7] Burns, A, A, Wellings. "Real-Time Systems and Programming Languages: Ada, Real-Time Java and C/Real-Time POSIX", Addison-Wesley Educational Publishers Inc., 2009.

[8] Buttazzo, G. C., "Hard RealTime Computing Systems", Third edition, Springer, 2011.

[9] Eickhoff, J., *Simulating Spacecraft Systems*, Springer, 2009.

[10] "FreeRTOS - market leading RTOS (real time operating system) for embedded systems supporting 34 microcontroller architectures". <http://www.freertos.org/>.

[11] Jenkins, T., I. Bogost. "Designing for the internet of things: prototyping material interactions." In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*, pp. 731-740. ACM, 2014.

[12] Jones, J. L., A. M. Flynn, Bruce A. Seiger. *Mobile robots: inspiration to*

implementation. Vol. 2. Wellesley MA: AK peters, 1999.

[13] Kopetz. H., "Real-Time Systems, Design Principles for Distributed Embedded Applications". Second Edition. Springer. 2011.

[14] Krishna, C. Mani. Real-Time Systems. John Wiley & Sons, Inc., 1999.

[15] Liu, J. W. S. Liu, "Real Time Systems", Integre Technical Publishing Co., Inc., 2000

[16] Rekleitis, I. M., G. Dudek, E. E. Milios. "Multi-robot exploration of an unknown environment, efficiently reducing the odometry error." In International Joint Conference on Artificial Intelligence, vol. 15, pp. 1340-1345. Lawrence Erlbaum Associates Ltd, 1997.

[17] Rohani, A., H. R. Zarandi. "An analysis of fault effects and propagations in AVR microcontroller ATmega103 (L)." In Availability, Reliability and Security, 2009. ARES'09. International Conference on, pp. 166-172. IEEE, 2009.

[18] Savvides, A., M. B. Srivastava. "A distributed computation platform for wireless embedded sensing." In Computer Design: VLSI in Computers and Processors, 2002. Proceedings. 2002 IEEE International Conference on, pp. 220-225. IEEE, 2002.

[19] Silberschatz, A., P. B. Galvin, G. Gagne, Operating System Concepts, 8th Edition, ISBN : 978-0-470-12872-5, Wiley, 2009.

[20] The Cheddar project: A free real time scheduling analyzer, <http://beru.univbrest.fr/~singhoff/cheddar/>

[21] Weber, Rolf H., and Romana Weber.

Internet of Things. New York: Springer, 2010.

[22] Xia, Feng, L. T. Yang, L. Wang, and Alexey Vinel. "Internet of things." International Journal of Communication Systems 25, no. 9 (2012): 1101.

[23] PHILLIP A. LAPLANTE, SEPPO J. OVASKA. REAL-TIME SYSTEMS DESIGN AND ANALYSIS Tools for the Practitioner Fourth Edition. A JOHN WILEY & SONS, INC., PUBLICATION. IEEE PRESS. 2012.

[24] <http://ftp.k-team.com/khepera/documentation/Kh2UserManual.pdf>

[25] <http://ftp.k-team.com/khepera/documentation/Kh2ProgrammingManual.pdf>

[26] <http://ftp.k-team.com/khepera/documentation/Kh2IRAN.pdf>

[27] <http://www.proyecto-ciaa.com.ar/devwiki/doku.php?id=desarrollo:edu-ciaa:edu-ciaa-nxp>

[28] <http://www.tolaemon.com/docs/modbus.htm>

[29] http://www.bosch-semiconductors.de/en/ubk_semiconductors/ip_modules_3/produkttablelle_ip_modules/can_literature_1/can_literature.html

[30] <http://www.can-cia.de/can-knowledge/can/can-fd/>

[31] http://www.iso.org/iso/catalogue_detail.htm?csnumber=59165

[32] http://www.freescale.com/files/microcontrollers/doc/app_note/AN1798.pdf

[33] http://www.can-cia.org/W_Scanner_Auto_OBD2_OBD1_2x2_E_OBD_20_PIN.html