Hardware and software architecture for a Rover robot

Arquitectura hardware y software para un robot Rover

Juan Camilo Gómez, Ing. [1,2] **& Nelson David Muñoz M. Sc.**[2]
1. Universidad de Antioquia, Medellín Colombia2. ICARO Grupo de Investigación (Instrumentacion, Control Automatico y Robotica).
Politecnico Colombiano, Jaime Isaza Cadavid, Medellín, Colombia
juancgomez@elpoli.edu.co, ndmunoz@elpoli.edu.co

Recibido para revisión 26 de junio de 2011, aceptado 18 de octubre de 2011, versión final 9 de diciembre de 2011

***Resumen* — Se presentan las principales características de la arquitectura hardware-software para el control del robot móvil RTT. Este vehiculo es un prototipo de investigación tipo rover con una suspensión Roker-Bogie de seis ruedas, que opera en terrenos irregulares, en condiciones arbitrarias. Todo el sistema está diseñado por un paradigma escalable y ofrece alto grado de modularidad de hardware y software, para permitir su fácil reconfiguración y adaptación a nuevas tareas. La arquitectura hardware, tiene un computador dedicado basado en S.O. Linux, donde todas las funciones del vehículo se controlan mediante sistemas de adquisición de datos modulares y controladores de hardware de interfaces de sensores y actuadores. La arquitectura de software utiliza una colección de varios subprocesos asincrónicos concurrentes utilizando el *middleware* JDE-Robot. Los esquemas proporcionan la capa de abstracción de hardware *(hardware abstraction layer) HAL*, comunicaciones TCP / IP y componentes para comportamientos inteligentes. Tambien, se usa un servidor SSH para comunicar la estación remota para monitoreo y programación todas las funciones del vehículo. La arquitectura ha sido probada obteniendo buenos resultados, controlando el desempeño del robot y la gestión de procedimientos de captura, procesamiento y entrega de toda la información sensorial.**

***Palabras Clave*— Robótica, Robot Móvil, *Rover*, Arquitectura *Software*, Arquitectura de *Hardware*, JDERobot.**

***Abstract*— This paper presented the main characteristics of hardware-software architecture for controlling the autonomous mobile robot RTT. This vehicle is a rover-type research prototype with a six-wheel Rocker-Bogie suspension, which operates in rough terrain under arbitrary conditions. The whole system is designed by a scalable paradigm, and provides high degree of hardware-software modularity, to allow easy reconfiguration and adaption for new tasks. The hardware architecture, has a dedicated computer based in OS Linux, where all the functions of the vehicle are controlled using modular data acquisition systems and hardware drivers to interface sensors and actuators. The software architecture uses a collection of several concurrent asynchronous threads using the JDE-Robot middleware. Schemas provides hardware abstraction layer HAL, TCP/IP communications and components for intelligent behaviors. Also, a SSH server is used to communicate the remote station to monitor and programming all vehicle functions. The architecture has been tested obtaining good results, controlling the robot performance and managing the procedure of capturing, processing, and delivering all sensory information.**

***Keywords*— Robotics, Mobile Robot, Rover, Software Architecture, Hardware Architecture, JDERobot.**

I. INTRODUCTION

Outdoor robotics, is gaining increasing interest due to its great potential for application in agricultural work, exploration, search and rescue in natural disasters, surveillance and explosives deactivation.

In order to achieve these tasks, mobile robots must deal with complex environments, consisting of different types of natural terrain and structural problems.

Outdoor robotics must have a basic structure that allows it to move in rough terrain. Also, a robust observation of this kind of environments needs a high diversity of sensors and actuators.

To handle this diversity and manage the procedure of capturing, processing, and delivering all sensory information acquired, a Hardware-Software (H/S) architecture is needed [1]. It should be added that this class of robots deal with concurrent embedded real-time performance and intelligent algorithms at the same time. The development of a generic robotic architecture to complex robotics systems has mainly two challenges: the complex level of the robotic systems; and the hardware diversity in which robots are built. To deal with these problems, a mobile robot in these conditions must have a distributed hardware and software architecture [2] [3].

Currently, there are relevant developments among the scientists and engineers for hardware and software architecture for mobile robots. In [4], authors proposed a distributed H/S architecture using two centralized embedded microprocessors and a CAN network to communicate sensors. Software architecture uses two mechanisms that permit parallel execution of software modules. One of them, use blackboards to permit interaction between software modules. The other mechanism, is a logic representation of the CAN bus called procCAN. The mobile robot exhibits a good real-time behavior, where the response time and the temporal predictability of the system are important. A similar architecture is also presented in [5].

The Intelligent Robotics Group at NASA Ames Research Center [6], developed its hardware architecture through a synergistic mix of off-the-shelf and custom designed components, allowing eased transplantation into a wide variety of mobile robot platforms. The system was implemented on the K9 rover, then integrated into the K10 series of human-robot collaboration research robots showing good results.

Also in [7], the authors build a distributed architecture, designed by a modular and layered structure, for the AMOR outdoor mobile robot. This architecture aims to manage different robots and provide a distributed communication system which allows the robots communicate among themselves and with a control base station.

All this developments, shows that it is possible to build a robust and flexible autonomous robotic system well suited for outdoor operation in rough terrain using a distributed H/S architecture.

This paper, presents the development of H/S architecture of a rover mobile robot for navigation on rough terrain designed by a scalable and distributed paradigm, to allow easy reconfiguration and adaption, providing integrated components for distributed communication and control to increase the system support. The architecture was designed and implemented in a modular fashion, both in hardware and software, with electronic devices for sensing and power management [8], [9], and a middleware that implements the model of schema-based cognitive architecture JDE using concurrent asynchronous threads [18].

II. RTT MOBILE ROBOT OVERVIEW

The mobile robot RTT (Fig 1) is a rover-type research prototype with Rocker-Bogie suspension [10][11], six-wheel independent, grouped in two sets of three wheels on each side of the vehicle and secured by means of an articulated structure. The rear wheels are linked to the robot's body by a rigid arm known as Rocker, and the same is fixed by means of a pivot, a second arm known as the Bogie, which holds the middle and front wheels. A differential mechanism that connects the two rocker arms to the body of the vehicle, keeps the RTT balanced even when the trains are at different heights. Overall, the differential system further the articulated structure, seeks to ensure that all six wheels are always kept in contact with the floor, allowing a permanent traction.

III. Hardware Architecture

A. Sensorial System

1. Proprioceptive Sensors: The RTT has sensors to inform several aspects of the state in which the robot is, these are:

- Current sensors (indirect torque measure)
- Orientation sensors
- Battery level sensors.
- Encoders - Wheels Angular speed
- Attitude Sensors

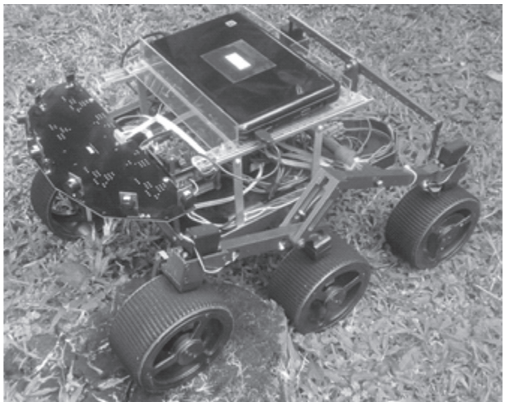


Fig. 1. Rover-type research prototype mobile robot

2.Exteroceptive sensor: Supported by a SRF02 ultrasonic sensor array [12], located taking into account the reflective characteristics typical of an ultrasound wave and considering the maximum detection focus that includes the angular width and the minimum and maximum distance detected. Similarly, it is considered the effects of crosstalk noise referred to the crossing of signals between ultrasonic sensors. The configuration used in the RTT is composed of 7 ultrasonic sensors located as shown in Fig 2. Each sensor uses an I2C communication system. The distance measurements are managed through a USB interface, by a dedicated computer with a data acquisition system custom designed, using a 16-bit embedded microcontroller.

3.Inertial sensor: Three CruizCore® R1001E fully self-contained MEMS digital gyroscopes for measuring heading angles were used.

Each uses a digital UART-USB board as communication interface with the dedicated computer, and are located in each axes of the robot's body frame. The R1001E has 50Hz bandwidth and precisely measures angular rates up to ± 100 °/sec, it can also measure rates up to ± 150 °/sec with lesser accuracy.

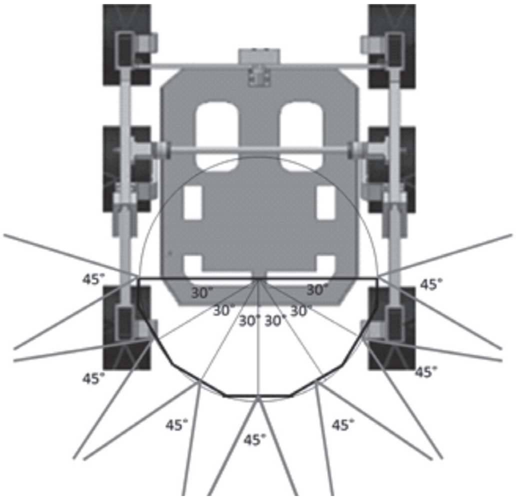


Fig. 2. Distribution of ultrasound sensors. (Circular panorama and detection focus).

There is also a low cost Freescale® MMA7260® 3-axis accelerometer as second source of redundant information for tilt estimation and an electronic compass for alternative estimate of the robot pose.

This sensors, allow estimate the robot attitude. The attitude is usually expressed in terms of three special angles known as "Euler angles". The angles are φ, θ, and ψ, which are usually referred to as roll (sometimes also called "bank angle"), pitch (also called "elevation"), and yaw (also called "heading" or "azimuth") respectively. Rates of rotation of the body frame relative to the navigation frame can be expressed in terms of the derivatives of the Euler angles, [13][14]. The inertial measurement unit (IMU) is shown in Fig 3.

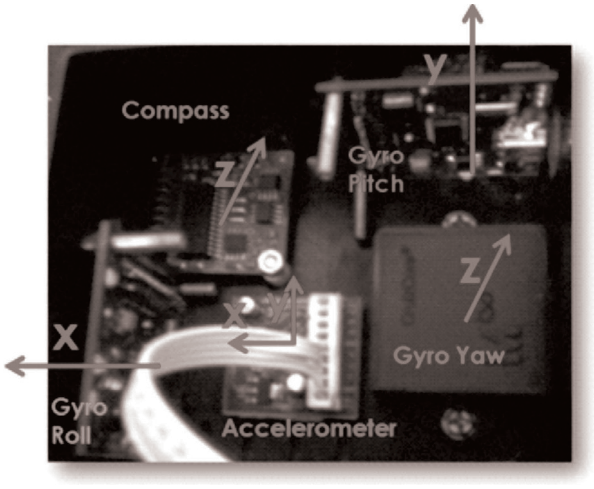


Fig. 3. RTT Inertial measurement unit (IMU)

B. *Hardware Control System*

The Hardware architecture is shown on Fig 4.

The control unit has an onboard-compact dedicated computer, with an Intel ® Atom ™ N270 1.60 GHz, with sufficient processing speed to perform autonomous navigation algorithms and control systems. Sensor data and control signals are handled by the computer through USB, using two data acquisition interfaces (DAQ).

The primary DAQ interface board (DAQ-1) is formed by a digital signal controller (DSC) dsPIC33FJ256MC710 [15], which is responsible for sending control signals to each of the traction motors through low-level PID controllers, providing stability and ability to function in a variety of terrain. In addition has the function of collecting and processing the signals from the proprioceptive sensors.

The second DAQ interface board (DAQ-2) uses a PIC18F4455 microcontroller [15], which aims to receive information from a set of exteroceptive sensors that provide data and inertial environment and state of the robot respectively. An obstacle detection system based on ultrasonic sensors was used, and the use of positioning techniques to correct odometric estimation errors using the IMU.

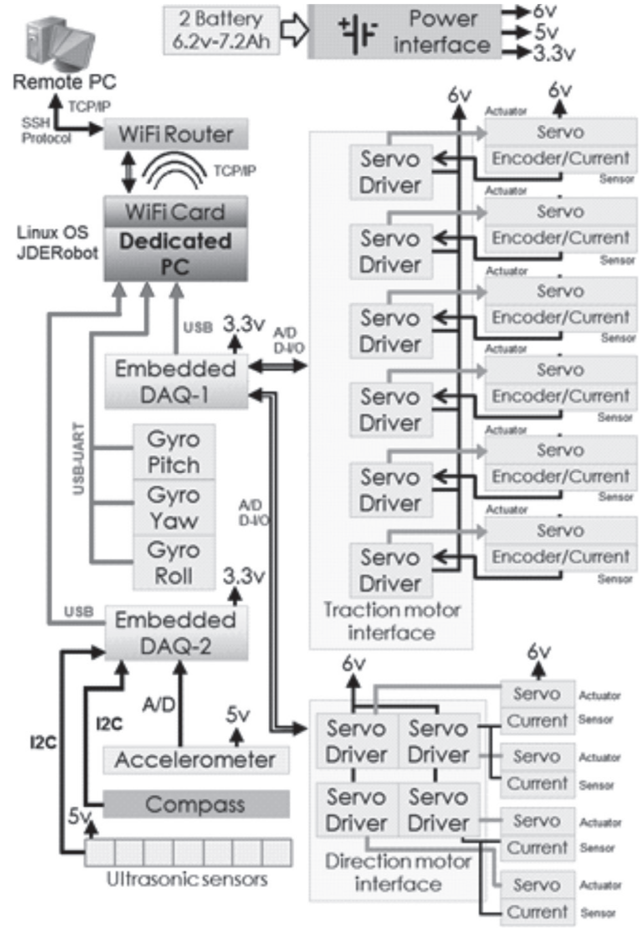


Fig. 4. RTT Hardware architecture

Communication interfaces with the central processing system are established via the USB port, while other peripherals for sensor managing are performed using different communication buses such as I2C, SPI and RS232.

Additionally, it profits from computer's wireless card for making a call via WiFi, for monitoring, teleoperation and platform programming. The WiFi allows communication range up to 200 meters and the possibility of extending this distance using routers in repeater configuration [16].

IV. SOFTWARE ARCHITECTURE

The software architecture is implemented in the dedicated computer mentioned above, which has a Linux Ubuntu 8.04 LTS "Hardy Heron" with kernel 2.6.21. The software layer is developed on the middleware "JDE-Robot" [17][18], which is a development software for robotic applications and bases its operation on JDE cognitive architecture (Jerarquía Dinámica de Esquemas or Dynamic Schemas Hierarchy) [19][20]. JDE-Robot is free and uses C/C++ programming language. The software architecture is shown in Fig 5.

JDE-Robot provides a programming environment where the robot control program is made up of a collection of several concurrent asynchronous threads named "schemas" that can be organized hierarchically. Each schema has the ability to run at a controlled rate. Under these specifications, our software architecture runs in real -time [19].

These schemas are loaded using a configuration file on JDE-Robot execution, where there are defined schema behavior executable as driver, service or application schema [18].

JDE-Robot also provides the ability to use shared variables that can be used in any scheme, this provides dynamism and modularity between different applications developed, greatly facilitating the programming and debugging.

A. Hardware Driver Schemas

JDE-robot was used to simplify the access to robotic hardware, from the control program using several custom drivers to support the RTT hardware. On this way, getting sensor measurements is as simple as reading a local variable, and ordering motor commands as easy as writing an actuator variable. The platform, updates those sensor variables with fresh readings and implements such actuator variables. All of them together, set a shared variable API for the robot programming. The robotic application reads and writes such variables to unfold its behavior. This configuration provides the Hardware Abstraction Layer (HAL) [21] between the hardware and software architecture. As shown in Fig 5, for each hardware interface (DAQ-1, DAQ-2 and gyroscopes sensors), a scheme (thread) were assigned.

Additionally, a scheme that is responsible for wireless communication with the sockets was implemented, they were programmed to transmit schemes data based on the TCP/IP protocol. A socket is simply a "communication channel" between two programs running on remote computers [22]. This Driver-Scheme, behaves like a Client-Server, so the onboard computer acts as the RTT server, which is constantly on listening mode, waiting until a client (remote computer) makes a connection request. When the server receives the request, establishes a bidirectional communication with the client. The client computer contains a schema, which provides the user with a graphical interface for monitoring and tele-operation of the prototype (see Fig 6).

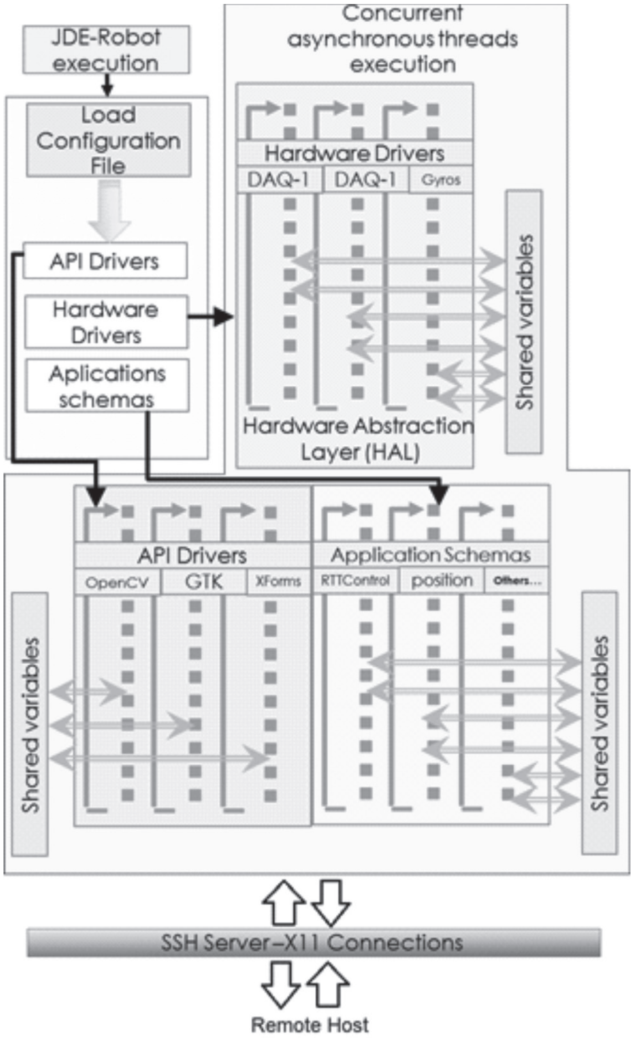
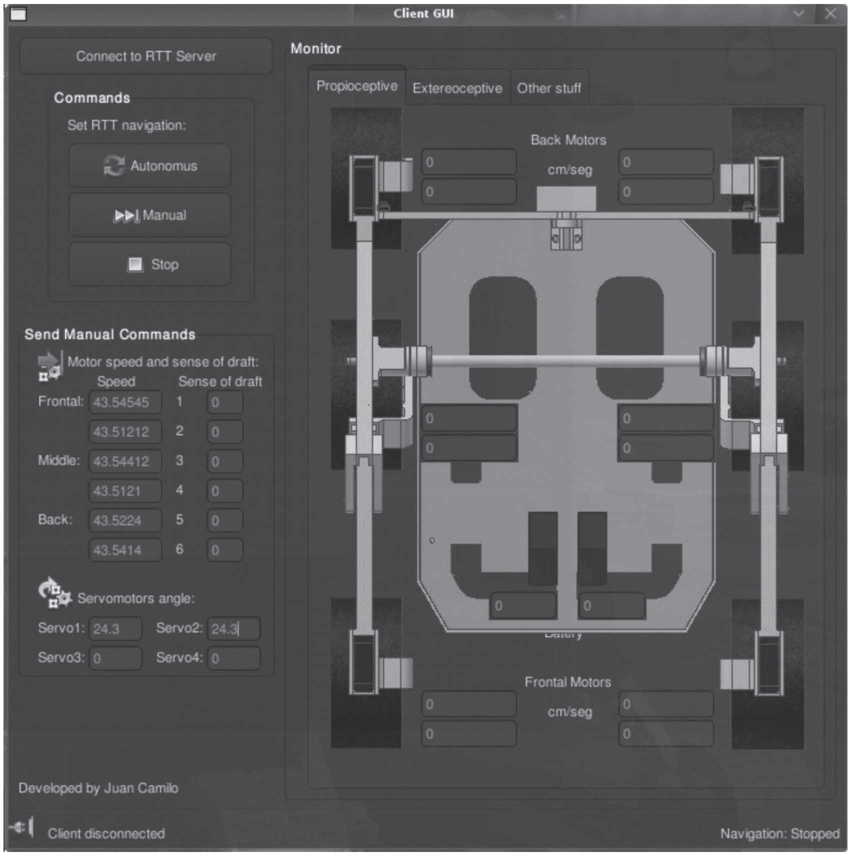


Fig. 5. RTT Software architecture with JDE-Robot

B. Control Schemas

Two schemes were developed for robot control and navigation of the RTT. The first one (Position scheme) is responsible for estimating the odometry, based on the Ackerman kinematic model [23], which describes in a more approximate way the robot kinematic behaviour. The second is the control thread (RTTcontrol scheme), where the robotic application is programmed (navigation algorithms), which also has a graphical user interface (Fig 7) to review the status of the perception and action variables of the robot. Two schemes were used to perform these tasks, due to the complexity and high computational cost of a single scheme, which may decrease the performance of the robot.



Set RTT navigation:

Motor speed and sense of draft:

	Speed		Sense of draft
Frontal:	43.54545	1	0
	43.51212	2	0
Middle:	43.54412	3	0
	43.5121	4	0
Back:	43.5224	5	0
	43.5414	6	0

Servomotors angle:

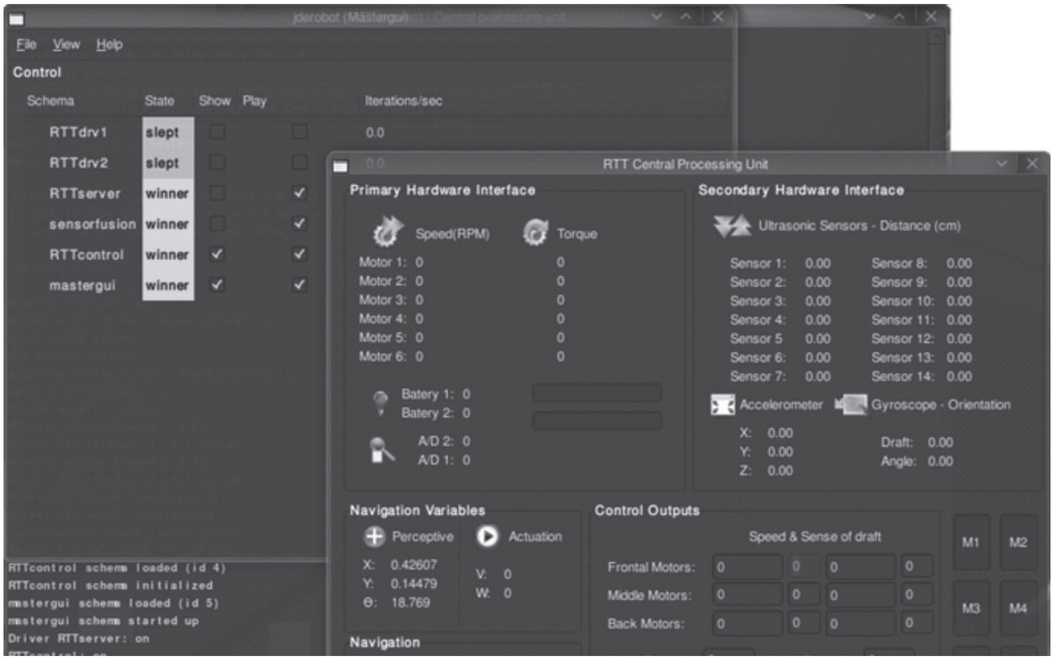
Servo1: 24.3 Servo2: 24.3
Servo3: 0 Servo4: 0

Developed by Juan Camilo

Client disconnected

Navigation: Stopped

Fig. 6. Graphical user interface for monitoring and teleoperation via Wifi.



Schema	State	Show	Play	Iterations/sec
RTTdrv1	slept	☐	☐	0.0
RTTdrv2	slept	☐	☐	0.0
RTTserver	winner	☐	✓	
sensorfusion	winner	☐	✓	
RTTcontrol	winner	✓	✓	
mastergui	winner	✓	✓	

```
RTTcontrol schema loaded (id 4)
RTTcontrol schema initialized
mastergui schema loaded (id 5)
mastergui schema started up
Driver RTTserver: on
```

	Speed(RPM)	Torque
Motor 1:	0	0
Motor 2:	0	0
Motor 3:	0	0
Motor 4:	0	0
Motor 5:	0	0
Motor 6:	0	0

Batery 1: 0
Batery 2: 0
A/D 2: 0
A/D 1: 0

Sensor 1:	0.00	Sensor 8:	0.00
Sensor 2:	0.00	Sensor 9:	0.00
Sensor 3:	0.00	Sensor 10:	0.00
Sensor 4:	0.00	Sensor 11:	0.00
Sensor 5	0.00	Sensor 12:	0.00
Sensor 6:	0.00	Sensor 13:	0.00
Sensor 7:	0.00	Sensor 14:	0.00

X: 0.00
Y: 0.00
Z: 0.00

Draft: 0.00
Angle: 0.00

X: 0.42607
Y: 0.14479
Θ: 18.769

V: 0
W: 0

Frontal Motors:	0	0	0	0
Middle Motors:	0	0	0	0
Back Motors:	0	0	0	0

Fig. 7. Graphical user interface for monitoring and teleoperation via Wifi.

C. *Platform Programming*

A wireless link speeds up the programming, compiling and debugging of algorithms for navigation and control. This was accomplished with the informatic protocol SSH (Secure SHell) and used to access remote machines via a network, allowing completely manage on-board computer through a command interpreter, or tracking the X11 graphical environment for Linux, from any platform (Windows, Linux, Mac, etc.) Fig 8. SSH works similar to telnet: the main difference is that SSH uses encryption to ensure that the information that travels over the media, goes on a non-legible and no third person can find the username and password of the connection over the entire session. [24].

V. CONCLUSIONS

The use of JDE robot software development platform, facilitated the implementation and integration of hardware and software components of RTT, appropriating some schemes of the platform and developing some for the particular system not supplied by JDE-Robot.

The mobile platform has a control architecture equipped with a functional processing that allows to have both internal and external communication, monitoring data from the sensors and drive the actuators. It also allows for future expansion in its modules. The sensory system integrates proprioceptive and exteroceptive sensors and provides comprehensive information of the state of the RTT, which integrated with the control architecture facilitates navigability.

The hardware abstraction layer (HAL) implemented in the drivers, allows simplified access to devices, bringing benefits in terms of the ability to debug robotic applications.

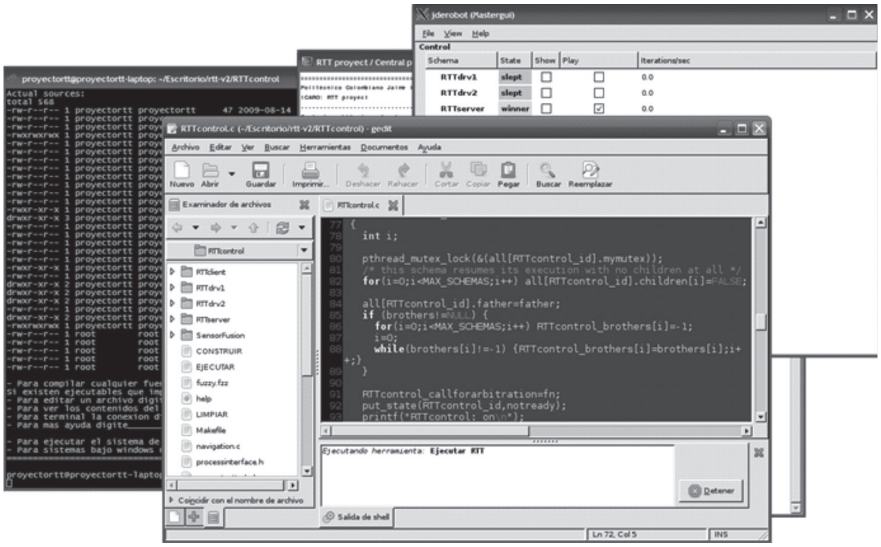
VI. FUTURE WORK

The RTT odometry is based on an approximate kinematic model, however, is of great interest to study and implement a more accurate model, that describes the behavior of the robot, so it achieves an accurate estimate of the position.

Another future work, is to adapt the robot for autonomous navigation using local and global navigation methods through

JDE-Robot software plataform.

It is proposed to integrate an artificial vision system, with the aim to increase the capacity of sensing and perception of the robot.



Schema	State	Show	Play	Iterations/sec
RTTdrv1	slept	☐	☐	0.0
RTTdrv2	slept	☐	☐	0.0
RTTserver	winner	☐	☑	0.0

```
Actual sources:
total 568
-rw-r--r-- 1 proyectortt proyectortt     47 2009-08-14
-rw-r--r-- 1 proyectortt proy
-rw-r--r-- 1 proyectortt proy
-rwxrwxrwx 1 proyectortt proy
-rw-r--r-- 1 proyectortt proy
-rw-r--r-- 1 proyectortt proy
-rw-r--r-- 1 proyectortt proy
-rw-r--r-- 1 proyectortt proy
-rw-r--r-- 1 proyectortt proy
-rw-r--r-- 1 proyectortt proy
-rw-r--r-- 1 proyectortt proy
-rw-r--r-- 1 proyectortt proy
-rwxr-xr-x 1 proyectortt proy
drwxr-xr-x 3 proyectortt proy
-rw-r--r-- 1 proyectortt proy
-rw-r--r-- 1 proyectortt proy
-rw-r--r-- 1 proyectortt proy
-rw-r--r-- 1 proyectortt proy
-rw-r--r-- 1 proyectortt proy
-rwxr-xr-x 1 proyectortt proy
-rw-r--r-- 1 proyectortt proy
drwxr-xr-x 2 proyectortt proy
drwxr-xr-x 2 proyectortt proy
drwxr-xr-x 2 proyectortt proy
-rw-r--r-- 1 proyectortt proy
drwxr-xr-x 2 proyectortt proy
-rwxrwxrwx 1 proyectortt proy
-rw-r--r-- 1 root        root
-rw-r--r-- 1 root        root
-rw-r--r-- 1 root        root
-rw-r--r-- 1 root        root
-rw-r--r-- 1 root        root

- Para compilar cualquier fue
Si existen ejecutables que im
- Para editar un archivo digi
- Para ver los contenidos del
- Para terminal la conexion d
- Para mas ayuda digite

- Para ejecutar el sistema de
- Para sistemas bajo windows
================================

proyectortt@proyectortt-lapto
```

```
{
  int i;

  pthread_mutex_lock(&(all[RTTcontrol_id].mymutex));
  /* this schema resumes its execution with no children at all */
  for(i=0;i<MAX_SCHEMAS;i++) all[RTTcontrol_id].children[i]=FALSE;

  all[RTTcontrol_id].father=father;
  if (brothers!=NULL) {
    for(i=0;i<MAX_SCHEMAS;i++) RTTcontrol_brothers[i]=-1;
    i=0;
    while(brothers[i]!=-1) {RTTcontrol_brothers[i]=brothers[i];i+
+;}
  }

  RTTcontrol_callforarbitration=fn;
  put_state(RTTcontrol_id,notready);
  printf("RTTcontrol: on\n");
```

```
Ejecutando herramienta: Ejecutar RTT
```

Fig 8. Remote programming from Windows through SSH, and X11

ACKNOWLEDGMENT

This work was partially funded by a research agreement between the Colombian Polytechnic Jaime Isaza Cadavid and the University of Antioquia, project number 2061080185 developed by the research groups GIMEL from UdeA, and Icaro from Colombian Polytechnic Jaime Isaza Cadavid. The authors acknowledge the advice provided by the researcher Lauro Ojeda from the mobile robotics lab at the University of Michigan – EEUU, and Professor Jose Maria Cañas Plaza from the Rey Juan Carlos University, Madrid - Spain. www.jderobot.org

REFERENCES

[4] Gutemberg S. Santiago and Adelardo A. D. Medeiros,A Proposed Hardware and Software Architecture for a Robotic System, 1st

Conferencia Brasileira de Sistemas Embarcados Críticos , CBSEC, Brasil, 2011.

[5] Ricardo S. Britto, Andre M. Santana, Anderson A. S. Souza, Adelardo A. D. Medeiros, Pablo J. Alsina,A Distributed Hardware-Software architecture for control an autonomous mobile robot, In Proceedings of the Second European Conference on Arti- cial Life, Brussels, 1993.

[6] Brooks, R.; , "A hardware retargetable distributed layered architecture for mobile robot control," *Robotics and Automation. Proceedings. 1987 IEEE International Conference on* , vol.4, no., pp. 106- 110, Mar 1987

[7] Camargo, R.F.; Chatila, R.; Alami, R.; , «Hardware and software architecture for execution control of an autonomous mobile robot,» *Industrial Electronics, Control, Instrumentation, and Automation, 1992. Power Electronics and Motion Control., Proceedings of the 1992 International Conference on* , vol., no., pp.818-825 vol.2, 9-13 Nov 2008.

[8] Garcia, O.; Solaque, L.; Aviles, O.; Nino, P.; , "Hardware and software architecture of a mobile robot with anthropomorphic arm," *ANDESCON, 2010 IEEE* , vol., no., pp.1-6, 15-17 Sept. 2010.

[9] Park, E.; Kobayashi, L.; Lee, S.Y.; , "Extensible Hardware Architecture for Mobile Robots," *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on* , vol., no., pp. 3084- 3089, 18-22 April 2005.

[10] Kuhnert, K.-D.; , "Software architecture of the Autonomous Mobile Outdoor Robot AMOR," *Intelligent Vehicles Symposium, 2008 IEEE* , vol., no., pp.889-894, 4-6 June 2008

[11] Arroyave, Maribel; Mazo, Diana L;Cano, Juan B.;Arias, Mauricio ;Vergara, Jairo M.;Londoño O. Nelson; Muñoz Nelson C., Sierra, Carlos I. RTT: Robot para navegacion en terrenos irregulares, Scientia Et Technica ISSN 0122-1701, Vol. XII, Núm. 32, december, pp. 79-84, 2006..

[12] J. B. Cano and others. Sistema de control y comando de un robot móvil para terrenos irregulares, En: Revista Politécnica, Año 4, No 6, Enero-Junio de 2008. Politécnico Colombiano Jaime Isaza Cadavid. ISSN 1900-2351. Available on: http://www.politecnicojic.edu.co/medios_de_comunicacion/revista_politecnica.html

[13] M. Thianwiboon, V. Sangveraphunsiri, R. Chancharoe. Rocker-Bogie Suspension Performance. Robotics and Automations Laboratory, Chulalongkorn University, Bangkok, Thailand 10330. 2001.

[14] Suojun Li; Haibo Gao; Zongquan Deng; , "Mobility performance evaluation of lunar rover and optimization of rocker-bogie suspension parameters," *Systems and Control in Aerospace and Astronautics, 2008. ISSCAA 2008. 2nd International Symposium on* , vol., no., pp.1-6, 10-12 Dec. 2008

[15] J. A. Echeverri A. S. Montoya C. Implementación de un módulo de sensórica para un robot de terrenos irregulares. Instrument & Control Engineering PFC. Politécnico Colombiano Jaime Isaza Cadavid. Medellin, 2009.

[16] Boney I.A et'al, Advantages of the modified Euler angles in the design and control of PKMs, Proceeding of the 3rd Chemnitz Parallel Kinematics Seminar/ 2002 Parallel Kinematic Machines International Conference, Chemnitz, Germany, April 23--25, pp. 171-188, 2002

[17] Ojeda, L.; Borenstein, J.; "FLEXnav: fuzzy logic expert rule-based position estimation for mobile robots on rugged terrain," *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on* , vol.1, no., pp. 317- 322 vol.1, 2002

[18] MICROCHIP TECHNOLOGY INC, dsPIC33FJ256MCXXX & PIC18F4455/4550/ Data Sheet, 2003. Available on: www.microchip.com.

[19] TELEPIEZA.COM, Dispositivos Inalámbricos WIFI para una Red LAN (Punto de Acceso y Punto de Extensión), Disponible en: www.telepieza.com/wordpress/2008/05/14/dispositivos-inalambricos-wifi-para-una-red-lan-punto-de-acceso-y-punto-de-extension/

[20] JDE-ROBOT. Robotics Research Group, Universidad Rey Juan Carlos. Available on: www.jde.gsyc.es

[21] Jdec: arquitectura basada en esquemas para aplicaciones roboticas J. M. Cañas, V. Gómez, P. Barrera and V. Matellán. Proceedings of RoboCity2030 1st Workshop on Arquitecturas de Control para Robots, Madrid, 12 febrero (2007)

[22] Jde+: an open-source schema-based framework for robotic applications J. M. Cañas, D. Lobato and P. Barrera. Proceedings of Second International Workshop on Software Development and Integration in Robotics SDIR 2007 affiliated with ICRA-2007, Rome, abril, 2007

[23] J. M CAÑAS. Jerarquía dinámica de esquemas para la generación de comportamiento autónomo, Telecomunications PhD Tesis. Universidad Politécnica de Madrid, Madrid, 2003.

[24] TANENBAUM, A.S, Structured Computer Organization, 5ta Edition, Prentice Hall, Upper Saddle River, New Jork, ISBN 0-13-148521-0. 1979.

[25] Xue, X.; Yang, S.X.; Meng, M.Q.-H.;, "Remote sensing and teleoperation of a mobile robot via the Internet," Information Acquisition, 2005 IEEE International Conference on , vol., no., pp. 6 pp., 27 June-3 July 2005.

[26] Aníbal Ollero Baturone, Robótica; manipuladores y robots móviles, 1ra Edition, Marcombo, 2002

[27] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Transport Layer Protocol" , RFC 4253, January 2006.

[28] A. M. Lazaro, J. Del Rio Fernandez. LabView 7.1 Graphic Programming for instrument control. Ed. Thomson, 2005.

UNIVERSIDAD
NACIONAL
DE COLOMBIA
SEDE MEDELLÍN

Cisco | Networking Academy®
Mind Wide Open™

Cisco Networking Academy es un programa ampliamente conocido de e-doing que enseña a los estudiantes las habilidades tecnológicas de Internet en una economía global. El programa proporciona contenido basado en la Web, pruebas en línea, seguimiento del desempeño de los estudiantes, laboratorios con equipos reales y con simuladores, soporte y entrenamiento por parte de los instructores, así como preparación para las certificaciones estándares de la industria.

Oferta de cursos

- ✓ Mantenimiento de PC: IT Essentials
- ✓ Redes básicas: Cisco Certified Network Associate
- ✓ Redes avanzadas: Cisco Certified Network Professional
- ✓ Seguridad en routers: CCNA Security
- ✓ Voz sobre IP
- ✓ Asterisk básico

Programación 2011
Ciclo 48: Inicia 17 de enero. Finaliza 12 de marzo
Ciclo 49: Inicia 22 de marzo. Finaliza 23 de mayo
Ciclo 50: Inicia 30 de mayo. Finaliza 29 de julio
Ciclo 51: Inicia 8 de agosto. Finaliza 3 de octubre
Ciclo 52: Inicia 10 de octubre. Finaliza 12 de diciembre

Consulte los horarios de cada nivel a través de nuestros canales informativos al pie de página

Además...

- ✓ Alquiler de laboratorios virtuales para auto-estudio o cursos empresariales
- ✓ Presentación de exámenes de certificación para multiples areas bajo el Centro Pearson VUE
- ✓ Cursos exclusivos para su empresa
- ✓ Pregunte por nuestros descuentos

CATC - Academia Regional - Academia Local
Universidad Nacional de Colombia sede Medellín
Calle 65 78-28 Bloque M1 Oficina 101. Facultad de Minas
Teléfono: +57 4 4255268 Fax: +57 4 2341002 E-mail: **catc@unal.edu.co** Web: **http://cnap.unalmed.edu.co**
Facebook: **fb.me/catcunal** Twitter: **@catcunal** Buzz: **google.com/profiles/catcunal**
Medellín, Colombia

CISCO.
Networking Academy