

Control de dirección para un robot con estructura Ackerman Steering usando plataforma NIOS II

José Hernández⁽¹⁾, John Rivera⁽²⁾, Ronald Ponguillo⁽³⁾
Facultad de Ingeniería en Electricidad y Computación⁽¹⁾⁽²⁾⁽³⁾
Escuela Superior Politécnica del Litoral (ESPOL)⁽¹⁾⁽²⁾⁽³⁾
Campus Gustavo Galindo, Km 30.5 Vía Perimetral, Apartado 09-01-5863. Guayaquil, Ecuador⁽¹⁾⁽²⁾⁽³⁾
jjhernan@espol.edu.ec⁽¹⁾, jariver@espol.edu.ec⁽²⁾, rponguil@espol.edu.ec⁽³⁾

Resumen

En este trabajo se muestra el diseño e implementación de un robot con estructura Ackerman Steering usando plataforma NIOS II. Para telecontrolar vía bluetooth la dirección del robot se ha desarrollado una interfaz gráfica a través del software AppInventor que está disponible para dispositivos móviles que operan con el sistema operativo Android.

Para la etapa de acoplamiento y control de la estructura con mecanismo Ackerman Steering utilizamos la tarjeta de desarrollo DE0-Nano Cyclone IV de Altera, en el que se ha diseñado la arquitectura de un mini-computador basado en el microprocesador NIOS II.

Este trabajo "Control de dirección para un robot con estructura Ackerman Steering usando plataforma NIOS II" se ha desarrollado con las nuevas tecnologías como son las FPGA de Altera debido a su versatilidad, excelente rendimiento y bajo costo.

Palabras claves: AppInventor, Estructura Ackerman Steering, FPGA, NIOS II.

Abstract

In this paper to show the design and implementation of a robot with Ackerman Steering structure using NIOS II processor. For telecontrol via bluetooth the direction of the robot has developed a graphical interface through AppInventor software that is available for mobile devices running the Android operating system.

For the coupling step and controlling the structure with Ackerman Steering mechanism use the NIOS II Cyclone IV FPGA form Altera, in which the architecture was designed for a mini-computer based on the NIOS II microprocessor.

This work "Steering control structure for a robot with Ackerman Steering using NIOS II platform" has developed witch the news technologies as the FPGA from Altera due to its versatility, excellent performance and low cost.

Keywords: AppInventor, Ackerman Steering Structure, FPGA, NIOS II.

1. Introducción

Actualmente las nuevas tecnologías como son los microprocesadores embebidos basados en dispositivos FPGA brinda una plataforma confiable por su fiabilidad, alta capacidad de procesamiento, versatilidad y excelente rendimiento.

El desarrollo del proyecto se centra en controlar una estructura con mecanismo Ackerman Steering mediante un Sistema Embebido. Además fue necesario elegir una plataforma de desarrollo adecuada para la interfaz entre el usuario y el prototipo de robot móvil. Dada la reciente aparición del sistema operativo Android y su rápida

adaptación, se optó por explorar su kit de desarrollo y las posibilidades que ofrece. Además, fue un factor importante para tomar esta decisión el hecho de que sea un sistema operativo abierto, lo que permite al desarrollador llevar a cabo sus ideas y poder usarlas de primera mano en su dispositivo, así como ofrecerlas a la comunidad Android.

2. Objetivos

2.1 Objetivo Principal

El principal objetivo es desarrollar un sistema embebido basado en el microprocesador NIOS II que permita ejercer el

control sobre la dirección de un robot móvil terrestre con estructura Ackerman Steering mediante bluetooth.

2.2 Objetivos Específicos

- Desarrollar una aplicación basada en el Sistema Operativo Android utilizando el software AppInventor para el telecontrol del robot vía bluetooth.
- Control de un prototipo robótico con estructura Ackerman Steering.
- Diseñar la arquitectura de la mini-computadora basado en el microprocesador NIOS II.
- Desarrollo de un conjunto de algoritmos en NIOS II IDE para la manipulación de sensores infrarrojos de distancia mediante el módulo ADC, motores mediante el módulo PWM y comunicación serial UART.

2.3 Identificación del problema

Hoy en día existen prototipos robóticos que sin duda alguna son muy prometedores para la ayuda del hombre, actualmente en nuestro país el desarrollo de robots está en auge por esta razón hemos decidido colaborar con nuestro conocimiento para desarrollar el control de dirección de un robot con estructura Ackerman Steering usando plataforma NIOS II.

Junto a una secuencia de investigaciones podremos observar su comportamiento para poder dejar encaminado el desarrollo de este robot para futuras mejoras e investigaciones, incorporando nuevas y avanzadas tecnologías.

2.4 Metodología

La elección del tipo de estructura es un aspecto importante para la construcción de un robot móvil, en nuestro caso se utilizó la estructura Ackerman Steering la misma que hace referencia a la forma en que se encuentran distribuidos los principales elementos que componen dicho sistema mecánico como son la plataforma, motores y ruedas, para lo cual adquirimos un modelo a escala de dicho sistema permitiéndonos utilizar la estructura sin ningún problema.

En la etapa de acoplamiento del móvil se incluye la etapa de control cuyos algoritmos son diseñados en NIOS II IDE.

La etapa de control está ligada a los periféricos que son los controladores, motores DC, sensores infrarrojos y dispositivo bluetooth. Se optó por un control asíncrono mediante bluetooth por su menor tasa de interferencia, además de permitir comunicar el robot con un dispositivo móvil. Por la acogida que tiene actualmente Android se ha decidido crear una aplicación con el fin de tele controlar al prototipo de robot.

3. Marco Teórico

3.1 Tarjeta DE0-Nano

La DE0-Nano es una tarjeta de desarrollo basada en la familia lógica programable FPGA Cyclone IV, que es ideal para el aprendizaje de la lógica digital y diseño con microprocesadores embebidos.

Una de las ventajas es su peso y tamaño que permite desarrollar una amplia variedad de proyectos de diseños portables y móviles. Así como también la capacidad de ser reconfigurado por el usuario de acuerdo a sus requerimientos.



Figura 1. Tarjeta DE0-Nano de Altera. [8]

La tarjeta de desarrollo DE0-Nano posee las siguientes características. [8]

Características	Descripción
FPGA	Cyclone IV EP4C22F17C6N con 153 pines I/O
Interfaces E/S	USB-Blaster para la configuración FPGA 2 Cabecera de expansión de 40 pines 1 Cabecera de expansión de 26 pines
Memoria	SDRAM de 32MB, EEPROM de 2Kbits con comunicación de I2C.
Sensor	Acelerómetro de 3 ejes con una resolución de 13 bits.
Switches y LEDs	2 Switches 8 LEDs verdes 4 interruptores DIP
Convertidor A/D	8 canales y 12 bits de resolución
Clock	50 MHz clock

Tabla I. Características de la tarjeta DE0-Nano de Altera.

3.2 NIOS II

NIOS II es un microprocesador de 32 bits de propósito general, basado en una arquitectura tipo Harvard, dado que usa buses separados para instrucciones y datos cuyas principales características son: [1]

- Tamaño de palabra asignado de 32 bits.
- Juego de instrucciones RISC de 32 bits.
- 32 registros de propósito general de 32 bits(r0-r31)
- 6 registros de control de 32 bits(ctf0 – ctf5)
- Instrucciones dedicadas para multiplicaciones de 64 y 128 bits.
- Acceso a variedad de periféricos integrados e interfaces para manejo de memorias y periféricos.

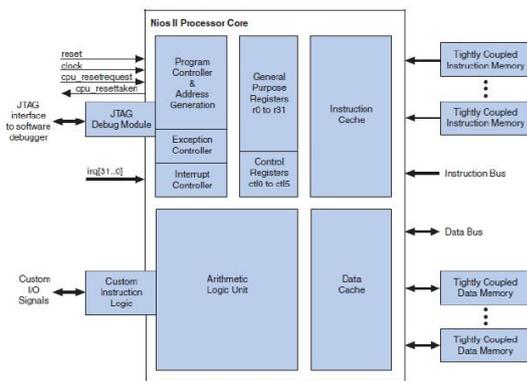


Figura 2. Diagrama de Bloques de la Arquitectura NIOS II.

3.3 Plataforma Android

Android es una plataforma de desarrollo para dispositivos móviles que permite el manejo de aplicaciones como:

- Bluetooth
- Wi-fi
- Cámara
- GPS
- Acelerómetro

Posee un entorno de desarrollo muy elaborado mediante un SDK disponible de forma gratuita.

3.4 Herramienta de desarrollo AppInventor

El software que se utilizó tiene características propias para desarrollar sistemas móviles suscritos por Android.

El kit de desarrollo (SDK) nos permite diseñar los requerimientos de la aplicación como son los controlares, adquisición de datos o segmentos de la programación orientados al enlace inalámbrico. Además se utilizó la herramienta de desarrollo (AppInventor de Google), por la facilidad de manejo de algoritmos a través de bloques interactivos.

La aplicación o interfaz de usuario fue cargada en un dispositivo móvil con el software requerido (Android Versión 2.3.1).



Figura 3. Interfaz del desarrollador.

3.5 Dispositivo HC06

Es un módulo con tecnología Bluetooth que sirve para el intercambio de datos de forma inalámbrica. En la tabla II se muestra algunas características del dispositivo.

Características HC06	
Protocolos soportados	Serial UART, USB, SPI
Frecuencia de trabajo	2.4 Hz
Rango de trabajo	3.1 V-4.2V
Corriente en comunicación	8mA
Bluetooth clase	1
Alcance máximo	100 metros
Temperatura soportada	-25° C + 75 ° C

Tabla II. Características del dispositivo Bluetooth HC06.

3.6 Dispositivos GP2D12

Es un dispositivo medidor de distancia que procesa señales infrarrojas mediante un transductor que permite obtener salida de voltaje analógico.



Figura 4. Sensor infrarrojo GP2D12.

3.7 Dispositivo LM298

El integrado permite el control de dos pequeños motores de corriente continua, consta con un bloque de alimentación, dos bloques de señales de control y dos bloques para la conexión de los motores.

El voltaje de control es una referencia que utiliza el circuito integrado, es necesario conectarlo directamente al voltaje del dispositivo que emite las señales de control.

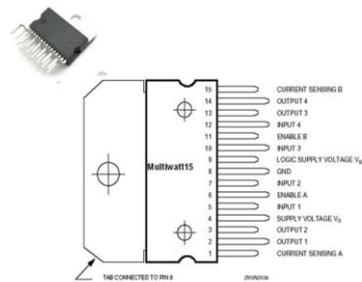


Figura 5. Circuito Integrado LM298.

4. Mecanismo Robótico

4.1 Estructura general de un robot móvil

El diseño de un robot móvil es basado en el comportamiento del ser humano como se muestra en la Figura 6.

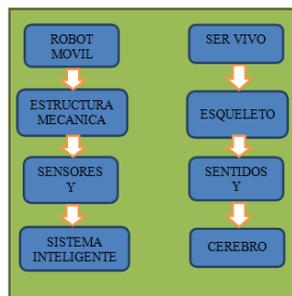


Figura 6. Comparación entre sistemas vivos y robóticos.

4.2 Grados de libertad y tipos de rueda

Los grados de libertad de un robot, así como los tipos de ruedas son aspectos que interviene en el proceso de control y análisis de movimiento del robot.

4.2.1 Grado de libertad (GDL)

Es cada uno de los movimientos de desplazamiento y rotación que puede realizar el robot.

- Un cuerpo que se mueve en dos dimensiones tiene 3 GDL (una rotación y 2 traslaciones).
- Un cuerpo que se mueve en tres dimensiones tiene 6 GDL (3 rotaciones y 3 traslaciones).

4.2.2 Tipos de ruedas

Las ruedas son unos de los elementos importantes que permite la movilidad en un robot y se clasifican como:

- Rueda fija.
- Orientación centrada.

- Orientación descentrada
- Rueda sueca

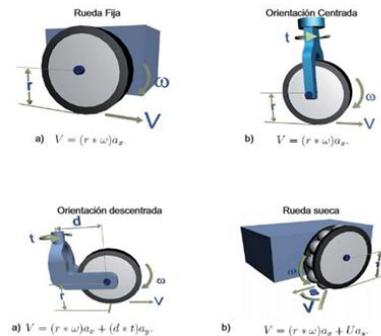


Figura 7. Tipo de ruedas.

4.3 Configuración Ackerman

Es usado en la industria de automóviles ya que posee dos ruedas traseras de tracción y dos ruedas delanteras para la dirección. Esta configuración está creada para evitar el derrape de las ruedas, haciendo que la rueda delantera interior posea un ángulo θ_i ligeramente mayor que el ángulo de la rueda exterior θ_o cuando el sistema se encuentra girando. Esto puede verse en el modelo cinemático mostrado en le figura 8. Donde:

L = separación lateral entre las ruedas.

D =separación longitudinal entre las ruedas.

ω =es la velocidad angular del robot.

V = velocidad lineal del robot.

θ =ángulo de rotación del robot.

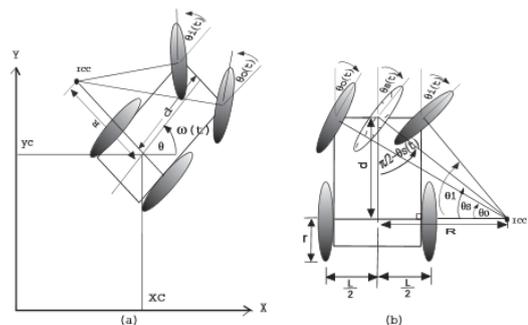


Figura 8. Modelo cinemático Ackerman.

La configuración Ackerman constituye un buen sistema de tracción aplicado inclusive para terrenos inclinados.

5. Diseño e Implementación

5.1 Diseño del Software

Para el diseño de la minicomputadora se ha utilizado la herramienta Qsys en la que se han agregado componentes

tales como: PWM, ADC y UART para la interacción entre la tarjeta y los dispositivos externos, por medio de un algoritmo creado en NIOS II IDE.

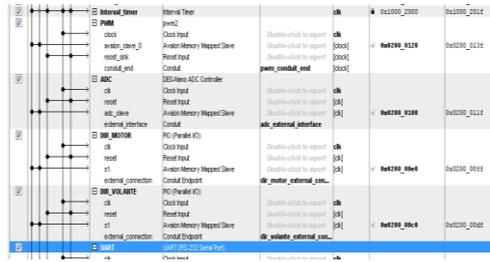


Figura 9. Diseño del sistema en Qsys.

5.2 Código del programa principal

5.2.1 Programación modo Manual

Se muestra de forma resumida parte del código principal, así como la definición de librerías, prototipos de funciones y variables globales.

```

/* librerías */
#include "system.h"
#include <stdio.h>
#include "altera_up_avalon_parallel_port.h"
#include "altera_up_avalon_de0_nano_adc.h"
#include "Sk_Pwms.h"
...

/* prototipos de las funciones */
void control(int * motor,int sentido, int
vel_motor, int * direccion, int giro, int
vel_direccion);
void delay(int tiempo);
...

/* variables globales */
unsigned int sentido =
volatile int * motor = (int*)DIR_MOTOR_BASE;
char modo='1'
int derecha;
unsigned int data=0;
...

/* programa principal */
int main(void)
{
alt_up_parallel_port_write_data(led, 2);
control(motor,0,vel_motor,direccion,0,0);
}
...

```

5.2.2 Programación modo Semi-Autónomo

```

/* librerías */
#include "system.h"
#include <stdio.h>
#include "altera_up_avalon_parallel_port.h"
#include "altera_up_avalon_de0_nano_adc.h"
#include "Sk_Pwms.h"
...

/* prototipos de las funciones */
void equilibrio(int * motor,int vel_motor, int
* direccion, int vel_direccion, int
dist_frente, int adelante, int data3);
void delay(int tiempo);
...

/* programa principal */
int main(void)
{
alt_up_parallel_port_dev * led;
alt_up_de0_nano_adc_dev * adc;
int canal0,canal1,canal2,canal3,k;
FILE * fp
fp= fopen ("/dev/UART","r+");
getc(fp);
if (canal==3)
{ while (inc<19)
{ dist_der=(data3-pow(2,29)-pow(2,30))/16;
acum=acum+dist_der;
}
}
}

```

5.3 Programación en AppInventor

En la plataforma AppInventor se desarrolló la aplicación para controlar el robot. En el menú principal se define los bloques de algoritmo que permite conectar e iniciar la comunicación directa vía bluetooth.

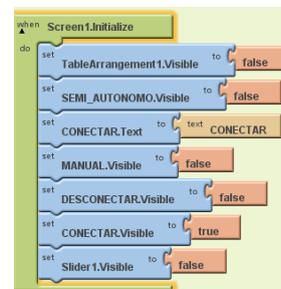


Figura 10. Presentación inicial.

5.3.1 Representación de datos para el enlace de control

Los datos enviados para telecontrolar el robot con estructura Ackerman fueron escogidos pensando en futuras mejoras, los caracteres se ordenan para una cómoda manipulación de envío de comandos desde el teclado del dispositivo móvil.

Estos caracteres son recibidos por el módulo bluetooth y son interpretados por el robot.

Dirección	Carácter de control	Dato formato hexadecimal	Dato formato binario
Adelante	W	0x57	01010111
Atrás	X	0x58	01011000
Giro a la derecha	D	0x44	01000100
Giro a la izquierda	A	0x41	01000001
Adelante e izquierda	Q	0x51	01010001
Adelante y derecha	E	0x45	01000101
Atrás e izquierda	Z	0x5A	01011010
Atrás y derecha	C	0x43	01000011
Incremento de cobertura del sensor	4	0x34	00110100
Decremento de cobertura del sensor	3	0x33	00110011

Tabla III. Datos de control de dirección del robot.

5.4 Interfaz Android y tarjeta DE0-Nano

Para controlar el robot vía bluetooth se utilizó el protocolo de comunicación UART, debido a que el dispositivo móvil transmite los datos por este medio.

DE0-Nano		Bluetooth	
Rx	Pin_J14	DTx	Pin 2
Tx	Pin_J13	DRx	Pin3

Tabla IV. Interconexión DE0-Nano y el módulo Bluetooth.

6. Resultados

6.1 Análisis de velocidades con respecto al PWM

En la Tabla V se muestra el incremento de la velocidad del robot móvil con respecto al módulo PWM. La velocidad máxima depende del suministro de corriente de las baterías y un PWM al 100%, dado que la masa total del robot es de 1.2Kg y un peso de 11.76 N.

PWM (%)	Velocidad (m/s)	Distancia (m)	Tiempo (seg)
100	0.74	2	2.67
90	0.63	2	2.91
80	0.62	2	3.22
70	0.54	2	3.67
60	0.48	2	4.1
50	0.39	2	5.02
40	0.37	2	5.03

Tabla V. Velocidad del móvil.

6.2 Datos experimentales de los sensores

En la Tabla VI se observa la comparación de los valores de voltajes experimentales y teóricos, con respecto a las distancias dadas por el sensor GP2D12.

Distancia (m)	Valor teórico de Voltaje[V]	Valor experimental [V]	% Error
30	0.91	0.90	0.5
25	1.10	1.15	2.2
20	1.27	1.40	4.8
15	1.60	1.70	3.0
10	2.09	2.45	7.9

Tabla VI. Medidas obtenidas del sensor GP2D12.

En la Figura 11 se observa la relación de voltaje experimental vs distancia.

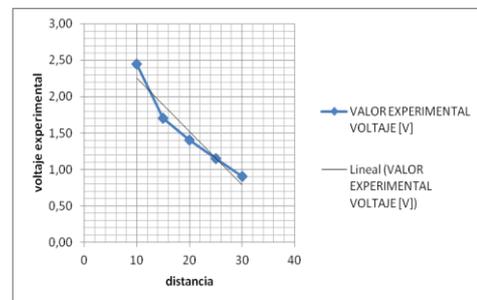


Figura 11. Relación voltaje experimental vs distancia

6.3 Consumo de corriente total

Dispositivo	Cantidad	Consumo mínimo [mA]	Consumo máximo[mA]
DE0-Nano	1	10	20
L298N	1	12	36
Sensor	3	90	120
Bluetooth	1	8	40
Total		120	216

Tabla VII. Consumo de corriente de los dispositivos.

7. Conclusiones

1. La minicomputadora fue consolidada mediante la interconexión de hardware y periféricos externos, por medio de las interfaces agregadas en Qsys.
2. El consumo de corriente de los dispositivos afectan directamente a la funcionalidad de los sensores de distancia infrarrojos.

3. AppInventor disminuye el tiempo de diseño de aplicaciones Android y la curva de aprendizaje en programación de dispositivos móviles.
4. El enlace inalámbrico entre el dispositivo Android y el robot puede ser interrumpido por bajo voltaje en las baterías.

8. Referencias

[1]. Altera-Corporation-DE0-Nano,
[http://www.altera.com/education/univ/materials/boards/de0-Nano/unv_de0-nano-board.html](http://www.altera.com/education/univ/materials/boards/de0-nano/unv_de0-nano-board.html). (Consultado el 10 de junio del 2013).

[2]. Altera Corporation - University Program “Quartus II Simulation QSIM” (consultado el 10 de junio del 2013).

[3]. Corporation – “Embedded Design”, January 2011
http://www.altera.com/literature/hb/nios2/edh_ed_handbook.pdf, (consultado el 10 de junio del 2013).

[4]. LaVahid, F. & Givargis, T. “Embedded System Design”. John Wiley & Sons, Inc. 2002. Disponible en: <http://dsp-book.narod.ru/ESDUA.pdf>, (consultado el 10 de junio del 2013).

[5]. Wang, C. “A Survey of Embedded Operating System”. 2005.
Disponible:<http://cseweb.ucsd.edu/classes/fa01/cse221/proje cts/group2.pdf>, (consultado el 2 de Julio del 2013).

[6]. David A. Perez A. - Centro de Investigación en Comunicación y Redes (CICORE), Caracas, Octubre 2009, “Sistemas Embebidos y Sistemas Operativos Embebidos” ND 2009-03, (consultado el 2 de Julio del 2013).

[7]. Rodríguez Araujo Jorge - “Estudio del Microprocesador Nios II”. Disponible en: <http://es.scribd.com/doc/28358833/Estudio-del-microprocesador-Nios-II>. Publicado Marzo 2010, (consultado 6 de agosto del 2013).

[8]. Altera Corporation, “NIOS II Software”, Disponible en http://www.altera.com/literature/hh/nios2/n2sw_nii5v2.pdf, (consultado 6 de agosto del 2013).