

Competencia de Robótica R2-D2 2014

Categoría: Velocista

Nombre del Robot: Velociraptor V1

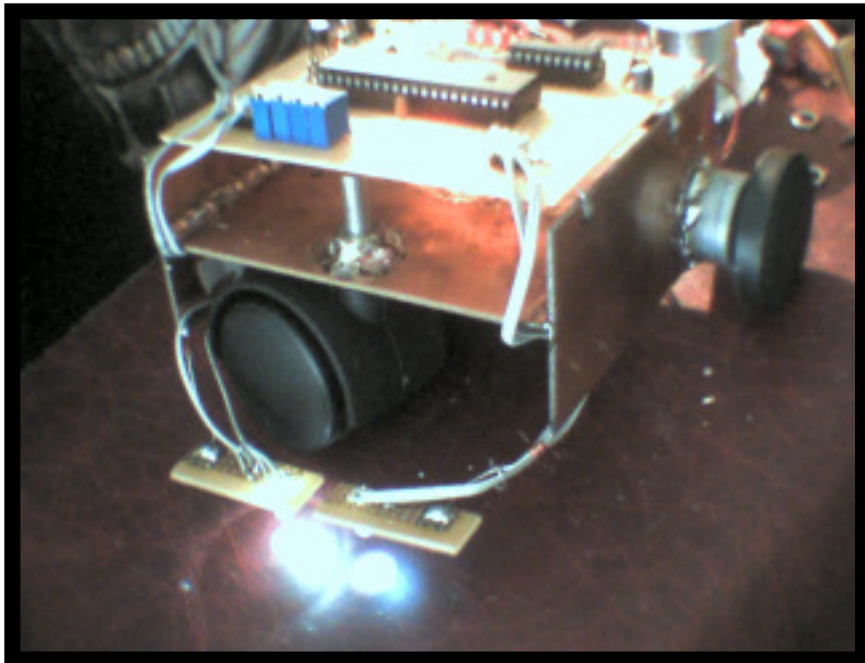
Institución: RTC_Argentina (Centro Regional de Entrenamiento
de Microchip en Argentina)

Participantes:

Andrés Raúl Bruno Saravia

Nicolas Ezequiel Bruno Saravia

Federico Andrés Bruno Saravia



Introducción:

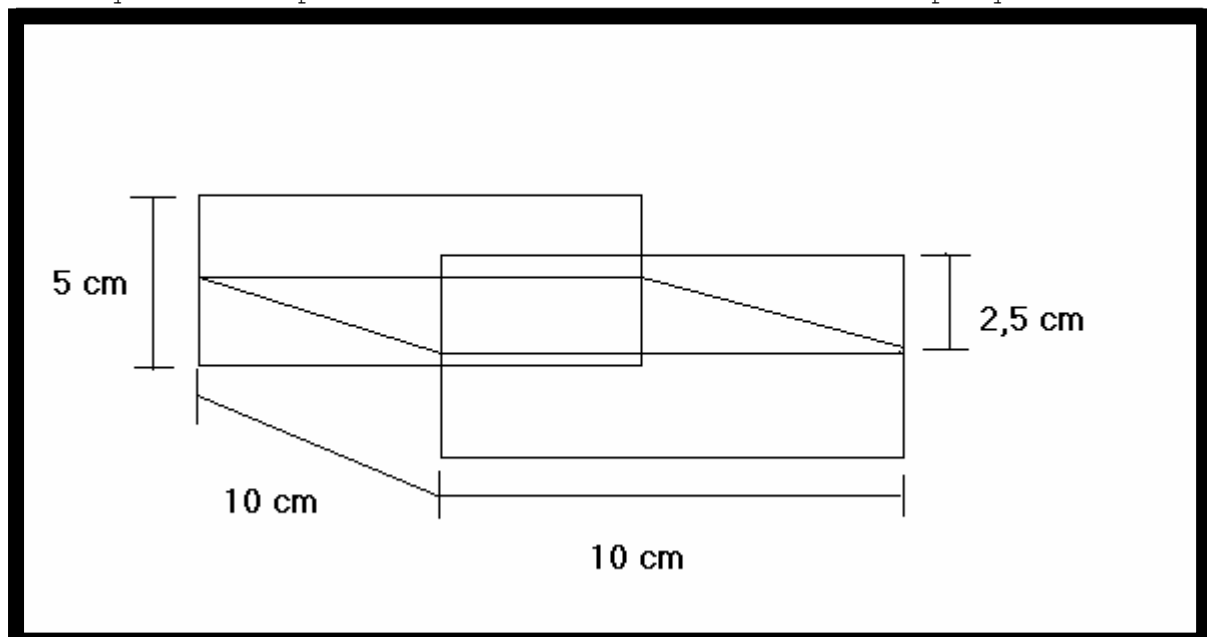
El Robot **Velociraptor V1** es un robot seguidor de líneas, su tracción esta realizada por 2 motores individuales y una rueda "gira loca" pasiva. Sus sensores son ópticos y están realizados con 2 LDRs y 2 LEDs blancos. La CPU esta realizada con un microcontrolador PIC de la Linea media mejorada con una velocidad de procesamiento de 4MIPS.

Objetivos:

Se ha buscado realizar el robot de la forma mas sencilla posible usando para la tracción motores lineales usados normalmente en el desplazamiento de las bandejas de las lectoras de CD y que se consiguen fácilmente en los comercios del gremio electrónico. En Nuestro caso los motores se extrajeron de 2 lectores de CD. Tanto las ruedas de tracción como el gira loco, se obtuvieron en la ferretería y son rudas para sillas. Las baterías que impulsan al Robot se obtuvieron de viejos celulares en desuso. La CPU se construyó a partir de un microcontrolador PIC de muy bajo costo AR\$42,21 en comercio local. En la construcción del esqueleto se usaron placas vírgenes doble faz FR4 que se tenían de recortes. El robot es perfectamente repetible, sencillo en su mecánica y posible de armar en cualquier parte

Mecánica:

El esqueleto del Velociraptor V1 esta realizado con plaquetas doble faz FR4 (fibra de vidrio-epoxi) de 10 x 5 cm. Esto permite primero tener una estructura rígida y de muy poco peso, en segundo lugar permite usar un soldador de 60 W y estaño, elementos convencionales en electrónica, para construir el robot y crear una estructura rígida, perfectamente reparable en caso de averías. No tienen partes unidas con pegamento, TODO ESTA SOLDADO hasta los motores. El esqueleto simplemente es una H realizada con las plaquetas:



El robot se alimenta con un pack de 3 baterías de litio de celulares colocadas en serie de 3.7V cada una, las cuales estando completamente cargadas (Tensión de flote adquieren 4.2V cada una). Las baterías son de 1500mAh, 2800mAh y 900mAh. (esto es porque se obtuvieron de distintos celulares). La tensión del PACK bajo carga es de 11V.

Para llegar a la carga completa necesitan de 12hs de carga continua.

Los motores se obtuvieron de 2 bandejas de CD de la misma marca, son motores que controlan el desplazamiento lineal de la bandeja de CD, son de una 500 RPM, y tienen un consumo con el eje cargado con el ROBOT de 120 mA por motor

Los sensores estan realizados con 2 Células Foto resistivas (LDRs) de 5mm de diámetro y 2 LEDs de alto brillo 10000 mlc de 15° color blanco.

Electrónica:

El circuito es muy sencillo usa como CPU un Microcontrolador PIC de la linea Media mejorada, preparados para trabajar en Lenguaje C (tienen 4 veces mas memoria que los viejos PIC16F, y su reloj interno funciona cuatro veces mas rápido, esto es necesario porque el código C genera un HEX 4 veces mas grande que en ASSEMBLER.

Se usa el oscilador interno. El PIC seleccionado es el PIC16F1519. Para leer los sensores se usa el ADC que esta operando a 2MHz de frecuencia de clock con un tiempo de conversión de de 5.5us en 10 bits.

Se usan 6 canales del ADC, con 2 se leen los LDRs que se usan como sensores, y 4 se usan para leer los presets que permiten ajustar el umbral de detección de cada sensor, y el ancho de pulso del disparo de cada motor. De esta forma la operación del robot es totalmente modificable sin tener que alterar el algoritmo del programa.

La fuente de alimentación para el microcontrolador esta realizada con un 7805. Un inductor de 680uHy a la entrada del regulador con un capacitor de 100nF disco evitar que el ruido eléctrico generado por el colector de los motores, alcance al regulador. Tanto la entrada como la salida del regulador tienen capacitares electrolíticos para mejorar la respuesta de riple y filtros de alta frecuencia realizados con capacitares de 100nF disco.

Para mejorar la respuesta instantánea de la fuente al consumo del MCU y su inmunidad al ruido, se colocaron 2 capacitores en paralelo, un electrolítico y otro cerámico de 100nf junto a los pines de alimentación del micro.

Para mejorar la respuesta instantánea de los motores en paralelo con los mismos se colocó el mismo conjunto de capacitares mencionado anteriormente.

Para comandar los motores se usó un ULN2803, el cual también es usado para excitar los LEDS auxiliares que indican la operación del ROBOT.

Programación:

El lenguaje usado para la implementación del programa es el C, el compilador es el de CCS (esto permite migrar el código a cualquier otro microcontrolador de forma casi automática, ya que el compilador hace las adaptaciones según el microcontrolador seleccionado).

El algoritmo es muy sencillo, consiste en leer los sensores y establecer si nos encontramos por debajo o no del nivel de umbral definido por un preset, y en función de esto impulsar los motores con un pulso cuyo ancho se determina por un preset.

Para leer los sensores y los presets se hacen 3 lecturas de cada canal y se calcula el promedio, de esta forma se integran las desviaciones entre lecturas. Esto permite tener mas precisión en la determinación de los pasos del algoritmo.

El umbral de los sensores esta determinado por el nivel de tensión colocado con presets multivuelta en las entradas analógicas.

El ancho del pulso que excita los motores también es determinado por el valor de tensión fijada por 2 presets.

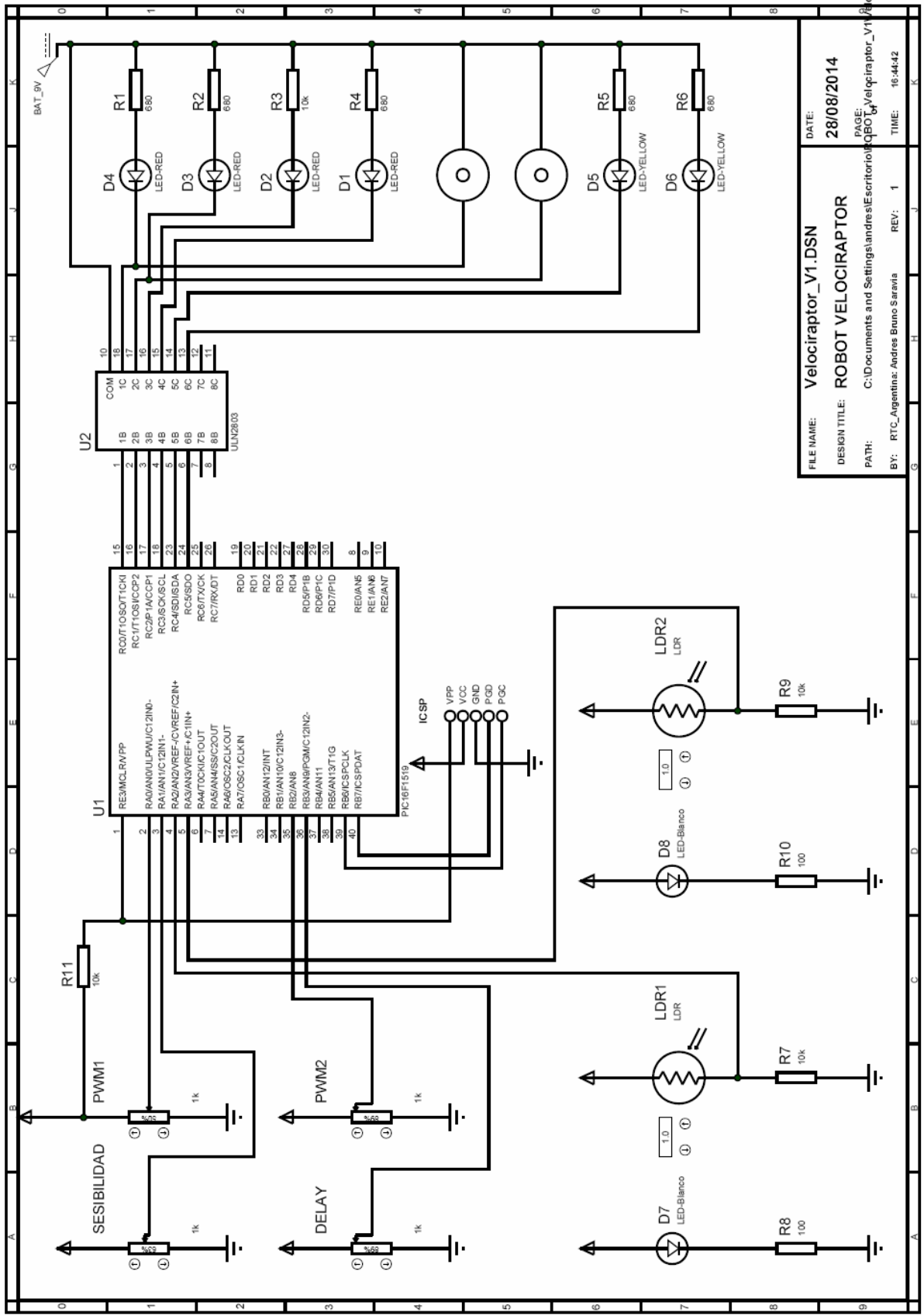
Para programar el microcontrolador se usa un PICKIT2 Clone de fabricación nacional. (fabricados en el RTC_Argentina)

Conclusiones:

El ROBOT en su etapa actual esta en pleno ensayo, una de las posibles mejoras es trabajar con los motores por PWM usando los módulos CCP1 y CCP2 en modo single, de hecho las salidas para motores fueron puestas en esos puertos con la finalidad de implementar dicha innovación.

Dicha modulación se alteraría a partir del cálculo de la cantidad de veces por milisegundo que el ROBOT se sale de la linea para calcular la velocidad que debe imprimirse por medio de la modulación PWM

ANEXO: Circuito eléctrico:



FILE NAME:	Velociraptor_V1.DSN
DESIGN TITLE:	ROBOT VELOCIRAPTOR
PATH:	C:\Documents and Settings\andres\Escritorio\Proyectos\Velociraptor_V1.dsn
BY:	RTC_Argentina; Andres Bruno Saravia
REV:	1
DATE:	28/08/2014
PAGE:	1
TIME:	16:44:42

```

#include "16F1519.h"
#device adc=10
#use delay(clock=16000000)

#fuses INTRC_IO,NOWDT
#fuses PUT,NOMCLR,NOPROTECT,NOBROWNOUT
#fuses NOCLKOUT,NOIESO
#fuses NOIESO,NOFCMEN,NOWRT,NOVCAP
#fuses STVREN,NODEBUG
#fuses NOLVP

//definiciones generales
#define MOTOR_LEFT PIN_C1
#define MOTOR_RIGHT PIN_C2
#define SAMPLES 30
//variables del sistema
unsigned long buffer_sample_analog[SAMPLES],adc0,adc1,adc2,adc3,adc8,adc9;
float volt0,volt1,volt2,volt3,volt8,volt9;
//funciones prototipo
void ConfigSystem(void); //Configura el sistema
void ReadAnalogS(void); //lee las entradas analogicas
unsigned long Prom(void); //calcula el promedio
void OutputControl(void); //controla las salidas
float ConverVolt(unsigned long valor); //convierte a volt
//función principal
void main(void)
{
    ConfigSystem();
    while(1)//Loop Multi Task Coperative
    {
        ReadAnalogS();
        Prom();
        OutputControl();
    }
}

//funciones auxiliares
void ConfigSystem(void)//Configura el sistema
{
    setup_oscillator(OSC_16MHZ|OSC_INTRC);
    setup_adc_ports(sAN0|sAN1|sAN2|sAN3|sAN8|sAN9);
    setup_adc(ADC_CLOCK_DIV_2);
    output_low(MOTOR_LEFT);
    output_low(MOTOR_RIGHT);
}

void ReadAnalogS(void)//lee las entradas analogicas
{
    unsigned char i=0;
    set_adc_channel(0);
    for(i=0;i>SAMPLES;i++)
    {

```

```

        buffer_sample_analog[i]=read_adc();
    }
    adc0=Prom();
    volt0=ConverVolt (adc0);

    set_adc_channel(1);
    for (i=0;i>SAMPLES;i++)
    {
        buffer_sample_analog[i]=read_adc();
    }
    adc1=Prom();
    volt1=ConverVolt (adc1);

    set_adc_channel(2);
    for (i=0;i>SAMPLES;i++)
    {
        buffer_sample_analog[i]=read_adc();
    }
    adc2=Prom();
    volt2=ConverVolt (adc2);

    set_adc_channel(3);
    for (i=0;i>SAMPLES;i++)
    {
        buffer_sample_analog[i]=read_adc();
    }
    adc3=Prom();
    volt3=ConverVolt (adc3);

    set_adc_channel(8);
    for (i=0;i>SAMPLES;i++)
    {
        buffer_sample_analog[i]=read_adc();
    }
    adc8=Prom();
    volt8=ConverVolt (adc8);

    set_adc_channel(9);
    for (i=0;i>SAMPLES;i++)
    {
        buffer_sample_analog[i]=read_adc();
    }
    adc9=Prom();
    volt9=ConverVolt (adc9);
}

unsigned long Prom(void)//calcula el promedio
{
    unsigned char i;
    unsigned long prom;
    for (i=0;i<SAMPLES;i++)
    {
        prom=prom+buffer_sample_analog[i];
    }
}

```

```

        prom=prom/SAMPLES;
    }
    return prom;
}

void OutputControl(void)//controla las salidas
{
    if(volt2<volt8)
    {
        output_high(MOTOR_LEFT);
        output_low(MOTOR_RIGHT);
        delay_us(adc0);
        output_low(MOTOR_LEFT);
    }
    if(volt3<volt9)
    {
        output_low(MOTOR_LEFT);
        output_high(MOTOR_RIGHT);
        delay_us(adc1);
        output_low(MOTOR_RIGHT);
    }
}

float ConverVolt(unsigned long valor)
{
    float tension;
    tension=(float)((valor*5.0)/1024.0);
    return tension;
}

```