

Implementación de un Microcontrolador PIC 16F628A de la marca Microchip, por medio de un controlador RSD-Micro 800; para manipular un carro prototipo mediante el sistema RS232, por medio de un teclado de PC

Implementation of a Microchip PIC 16F628A Microcontroller, by means of a RSD-Micro 800 controller; to manipulate a prototype car through the RS232 system, by means of a PC keyboard

Daniel Castellanos Hernández^a, Himbad Herrera Barajas^b, Adriana Zepeda Barragán^c y Julissa Geovanna Barajas Pacheco^d

Resumen / Abstract

México se ha caracterizado por ser un país tradicionalmente consumidor y no (generador de tecnología), tratando de modificar esta visión se ha trabajado en el Instituto Tecnológico de Ocotlán por generar e Implementar Interfaces a través de Microcontroladores PIC. Esto nos permite ser generadores de proyectos de automatización para la industria de la región, en el presente manipularemos un prototipo "carro pequeño", por medio de del teclado de una PC, modificando el control remoto original del prototipo, y adaptándolo a un controlador RSD-Micro 800, (Realizado por la misma Institución), se realizó la programación por medio del LabVIEW (Laboratory Virtual Engineering Workbench). (Instruments, 2012). Este resultado de manipulación aparentemente pequeño nos permite

a. El Ing. Castellanos Hernández tiene una Licenciatura en Ingeniería Electromecánica. Académico que labora como Profesor Investigador del Tecnológico Nacional de México, campus Tecnológico de Ocotlán en el Estado de Jalisco. Perteneció a la Academia de Ingeniería Electromecánica del Instituto Tecnológico de Ocotlán. Correo electrónico: daniel.ch@ocotlan.tecnm.mx.

b. El Doctor Herrera Barajas es Licenciado en Informática con grado de Maestría en Computación Aplicada con mención Base de Datos y con Doctorado en Ciencias de la Educación. Se desempeña como docente del Tecnológico Nacional de México, campus Tecnológico de Ocotlán, en el Estado de Jalisco. Perteneció a la Academia de Ingeniería en Sistemas Computacionales y actualmente es parte del staff académico que atienden la carrera de Ingeniería en Gestión Empresarial. Labora también como docente en la Universidad Interamericana para el Desarrollo, UNID, campus Ocotlán en el Estado de Jalisco. Correo electrónico: himbad.hb@ocotlan.tecnm.mx.

c. La Dra. Alfaro Castellanos es licenciada en Informática con Maestría en Tecnologías de Información y Doctorado en Educación. Profesora de Tiempo Completo del Departamento de Ciencias Básicas del Centro Universitario de la Ciénega de la Universidad de Guadalajara. Correo electrónico: kleophe.alfaro@academicos.udg.mx.

d. La Profesora Barajas Pacheco labora como personal docente en el Instituto Tecnológico Nacional de México en la sede de Ocotlán, Jalisco. Contadora Pública con Maestría en Educación. Correo electrónico: jgbarajasp@yahoo.com.mx.

incursionar en la Automatización de máquinas más grandes como Rauter, manipulación de motores, controlar la iluminación de edificios, sistemas de riego, entre otro más, todo ello controlado por medio de la PC.

Palabras clave: Tecnología Diseño Innovación.

Mexico has been characterized as a consumer, traditional country (technology generator), trying to change this vision has worked at the Institute of Technology Ocotlán to generate and implement interfaces dare PIC Micro controllers. This allows us to be generators of automation projects for the industry in the region, this handle a prototype "small car", through a PC keyboard, changing the original remote control of the prototype, and adapting it to RSD-Interface micro-800, will be held programming and compilation by means of LabVIEW (Laboratory Virtual Engineering Workbench). This apparent result of handling small mind allows us to venture into the larger machines Automation and Rauter, engine handling, control the lighting of buildings, irrigation systems, among other more, all controlled by the PC.

Key words: Technology, Design, Programming.

Key words: Technology, Design, Programming.

Introducción

Hoy en día en el Instituto Tecnológicos de Ocotlán se está desarrollando tecnología propia, como son controladores RSD-MICRO 200B con 6 entradas y 4 salidas con oscilador de 4 de Mhz. Externo (Tx) y RSD-MICRO 800 con 10 entradas y 6 salidas con el oscilador de 4 de Mhz Interno (IntRC I/O). Los cuales nos permiten implementar la manipulación de un carro prototipo pequeño a través de un Teclado de una PC, mediante la programación embebida del Microcontrolador *RSD-MICRO 800*, utilizando el software de programación LabVIEW (Laboratory Virtual Engineering Workbench), para control y monitorización del controlador por medio del protocolo de comunicación RS232. A través de él podemos administrar algunos recursos del PC, que nos permitirán controlar el prototipo, también se realizará una adaptación al control remoto y conectado a la interface *RSD-MICRO 800*, se logrará el control del prototipo y cumplir con los objetivos: Empezar a incursionar en la programación embebida, desarrollar investigación y generar prototipos y proyectos de interés académico e industrial.

MICROCONTROLADORES

¿Qué es un Microcontrolador?

De acuerdo a lo descrito por (Castellanos Hernandez, 2010), Es un circuito integrado programable que contiene todos los componentes de un computador. Se emplea para controlar el funcionamiento de una tarea determinada y, debido a su reducido tamaño, suele ir incorporado en el propio dispositivo al que gobierna. Esta última característica es la que le confiere la denominación de «controlador incrustado» (*embedded controller*).

De acuerdo a (Vesga Ferreira, 2008), El Microcontrolador es un dispositivo dedicado. En su memoria sólo reside un programa destinado a controlar una aplicación determinada; sus líneas de entrada/salida soportan la conexión de sensores y dispositivos de control que permitan efectuar el proceso deseado.

Según menciona en su manual del primer tomo (Castellanos Hernadez, 2010), Un Microcontrolador es un computador completo, aunque de limitadas prestaciones, que está contenido en el chip de un circuito integrado y se destina a gobernar una sola tarea. A continuación de muestra en la figura 1, Estructura de un sistema abierto basado en un microprocesador y en la figura 2, el Microcontrolador en un sistema cerrado.

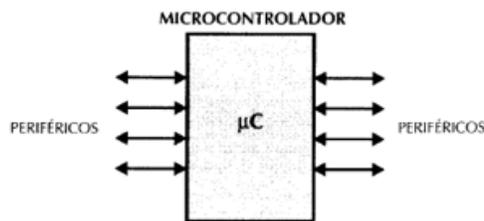


Fig. 1, Estructura de un sistema abierto basado en un microprocesador, (Castellanos Hernadez, 2010).

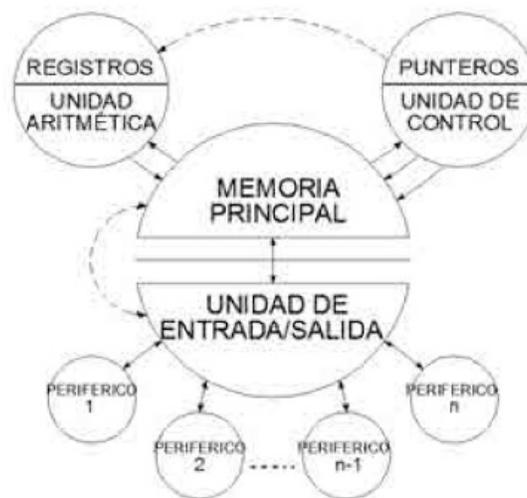


Fig. 2, Microcontrolador en un sistema cerrado, (Castellanos Hernadez, 2010)

ARQUITECTURA INTERNA

Un Microcontrolador posee todos los componentes de un computador, pero con unas características fijas que no pueden alterarse.

Las partes principales de un Microcontrolador son:

1. *Procesador:*
2. *Memoria volátil y no volátil para contener el programa.*
3. *Memoria de lectura y escritura para guardar los datos.*
4. *Líneas de EIS para los controladores de periféricos:*
 - a) *Comunicación paralelo*
 - b) *Comunicación serie*
 - c) *Diversas puertas de comunicación (bus I²C, USB, etc.)*
5. *Recursos auxiliares:*
 - a) *Circuito de reloj*
 - b) *Temporizadores*
 - c) *Perro Guardián («watchdog»)*
 - d) *Conversores AD y DA*
 - e) *Comparadores analógicos*
 - f) *Protección ante fallos de la alimentación.*
 - g) *Estado de reposo o de bajo consumo.*

A continuación, se pasa revista a las características más representativas de cada uno de los componentes del Microcontrolador.

La necesidad de conseguir elevados rendimientos en el procesamiento de las instrucciones ha desembocado en el empleo generalizado de procesadores de arquitectura Harvard frente a los tradicionales que seguían la arquitectura de von Neumann. Esta última se caracterizaba porque la UCP (Unidad Central de Proceso) se conectaba con una memoria única, donde coexistían datos e instrucciones, a través de un sistema de buses véase Figura 3, la Arquitectura de Van Neuman.

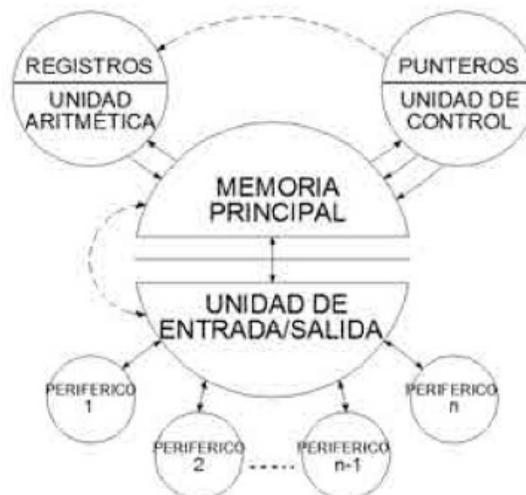


Fig. 3, Arquitectura de Van Neuman, (Martinez Amador, 2016)

En la arquitectura Harvard son independientes la memoria de instrucciones y la memoria de datos y cada una dispone de su propio sistema de buses para el acceso. Esta dualidad, además de propiciar el paralelismo, permite la adecuación del tamaño de las palabras y los buses a los requerimientos específicos de las instrucciones y de los datos. También la capacidad de cada memoria es diferente figura 4. Arquitectura de Harvard.

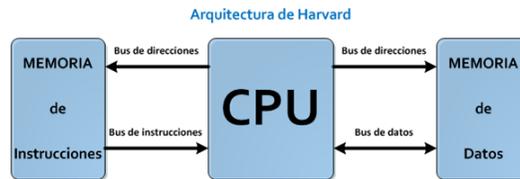


Fig. 4, Arquitectura de Harvard, (Eletro tools, 2021)

El procesador de los modernos Microcontroladores responde a la arquitectura RISC (Computadores de Juego de Instrucciones Reducido), que se identifica por poseer un repertorio de instrucciones máquina pequeño y simple, de forma que la mayor parte de las instrucciones se ejecuta en un ciclo de instrucción.

Otra aportación frecuente que aumenta el rendimiento del computador es el fomento del paralelismo implícito, que consiste en la segmentación del procesador (pipe-line), descomponiéndolo en etapas para poder procesar una instrucción diferente en cada una de ellas y trabajar con varias a la vez.

El alto rendimiento y elevada velocidad que alcanzan los modernos procesadores, como el que poseen los Microcontroladores PIC, se debe a la conjunción de tres técnicas:

- Arquitectura Harvard
- Computador tipo RISC
- Segmentación

El Microcontrolador que utilizaremos es el PIC 16f628A para nuestro proyecto el cual veremos a continuación. Figura 5.

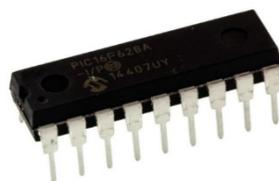


Fig. 5, Microcontrolador 16F628A, (Geek Factory, 2023)

Notas: Este Microcontrolador está adaptado a la siguiente interface misma que fue diseñada por el Ingeniero Daniel Castellanos Hernández, docente del Instituto Tecnológico de Ocotlán, Sus características son las que se pueden observar en la figura 6, Características del microcontrolador.

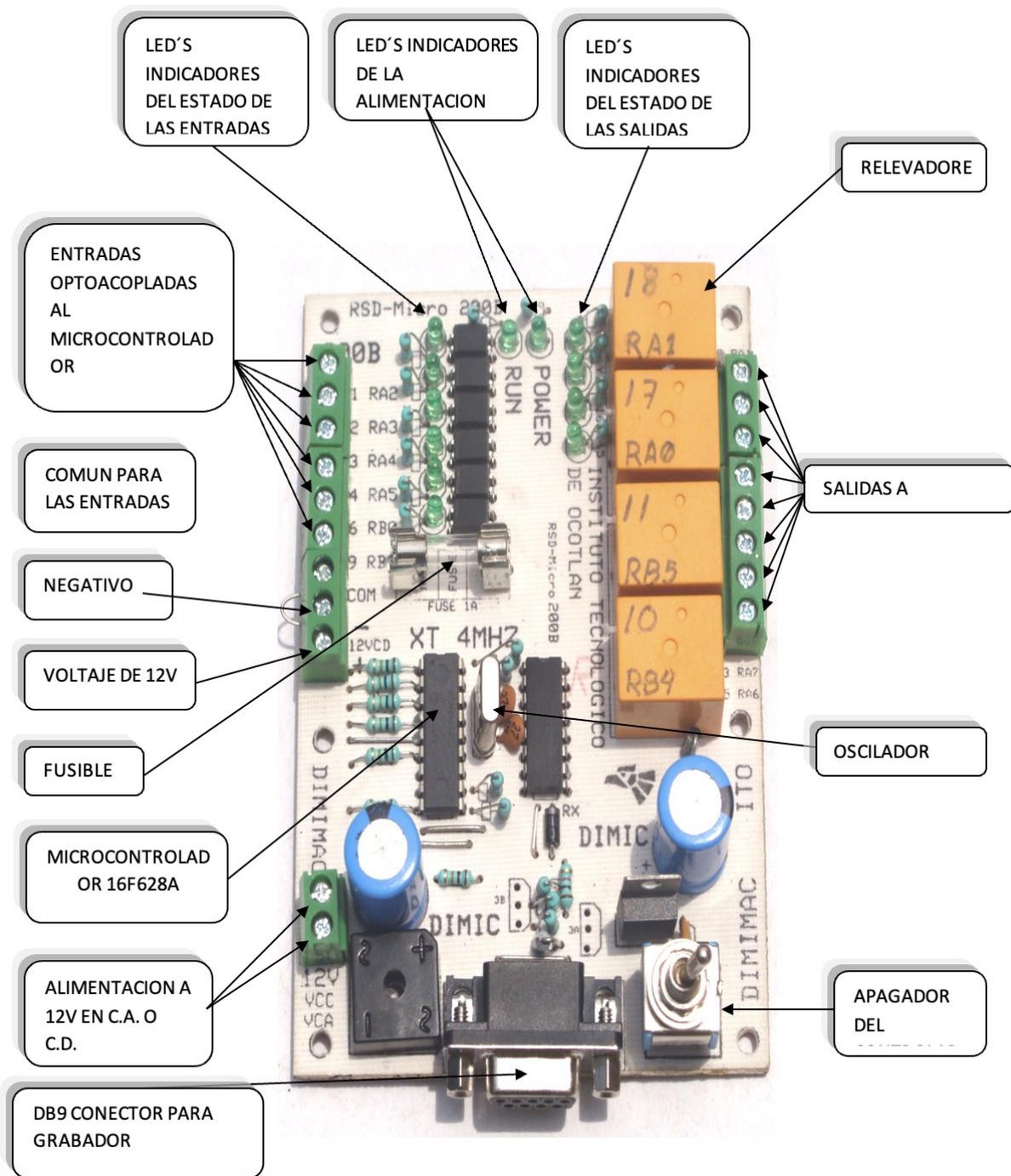


Figura 6, Características del microcontrolador, (Castellanos Hernandez, 2010)

Conexión de voltaje para las entradas, de forma interna y externa

La conexión externa se utiliza cuando tenemos sensores mayores de 1Amper, para no dañar el controlador. Como se muestra en la figura 7, conexión externa.

Sensores o apagadores

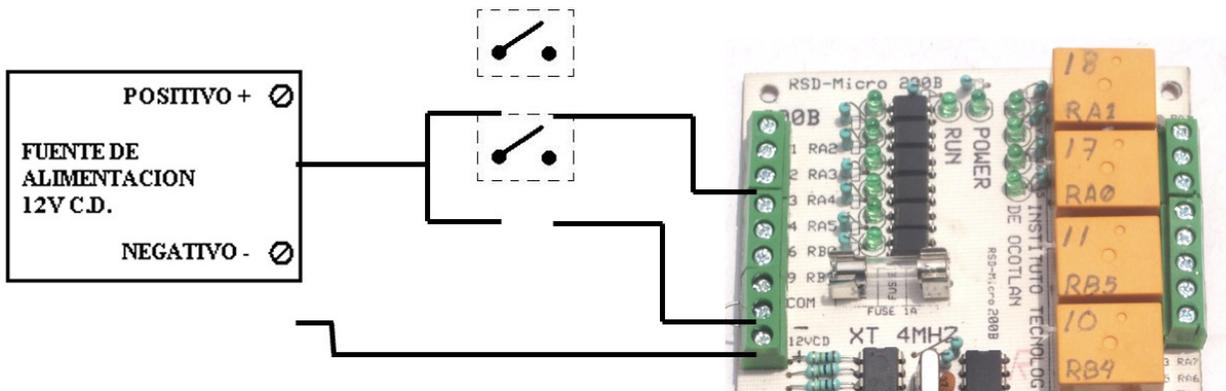


Fig. 7, Conexión externa, (Castellanos Hernandez, 2010)

CONEXIÓN INTERNA

La conexión interna se utiliza cuando tenemos sensores menores de 1Amper tomando el voltaje del controlador, como se muestra en la figura 8, conexión interna.

Sensores o apagadores

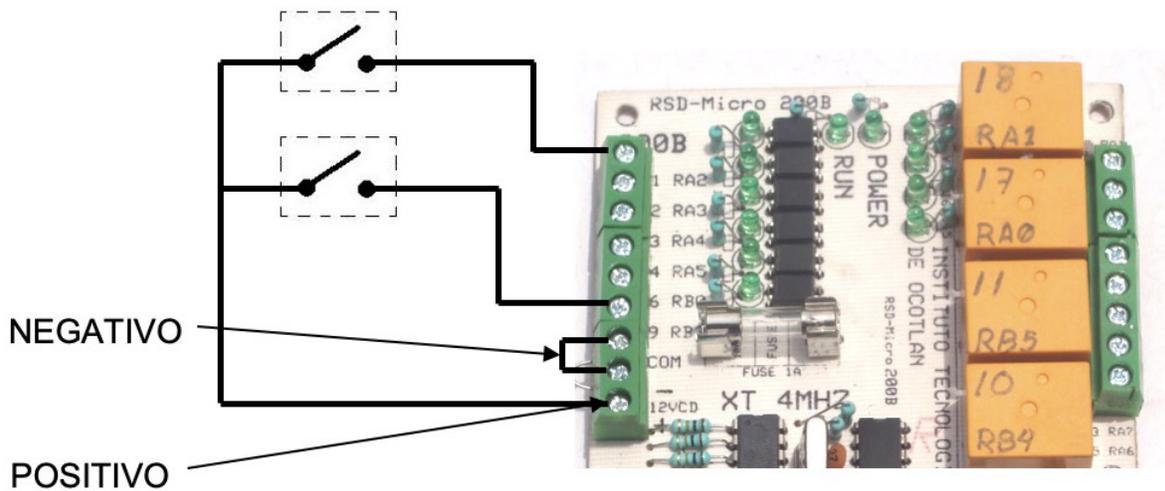


Fig. 8, conexión interna, (Castellanos Hernandez, 2010)

En la figura 9, se muestra la interface de Microcontroladores (controlador de la serie RSD-Micro 100 al 900.) Para grabar la programación en el Microcontrolador se utilizará la siguiente:

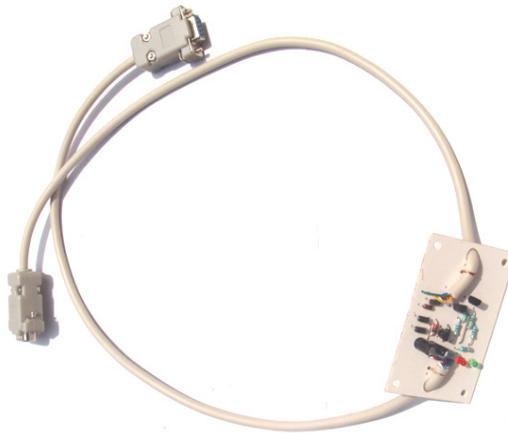


Fig. 9, interface de Microcontrolador para grabar la programación, (Castellanos Hernandez, 2010).

NOTA: este cable se fabricó exclusivo para este controlador por lo cual no se debe conectar a ningún controlador o PLC de otra marca por que se dañarían, y fue diseñada por el Ing. Daniel Castellanos Hernández, docente del Instituto Tecnológico de Ocotlán.

Para la implementación del prototipo tomaremos en cuenta las entradas y salidas de la interface RSD-Micro 800 con 10 entradas y 6 Salidas, que se muestran en la figura 10 y 11.

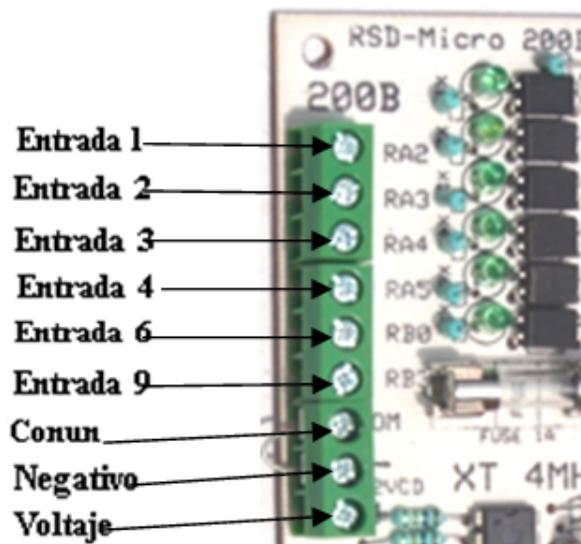


Fig. 10, Entradas, interface RSD-Micro 800, (Castellanos Hernandez, 2010)

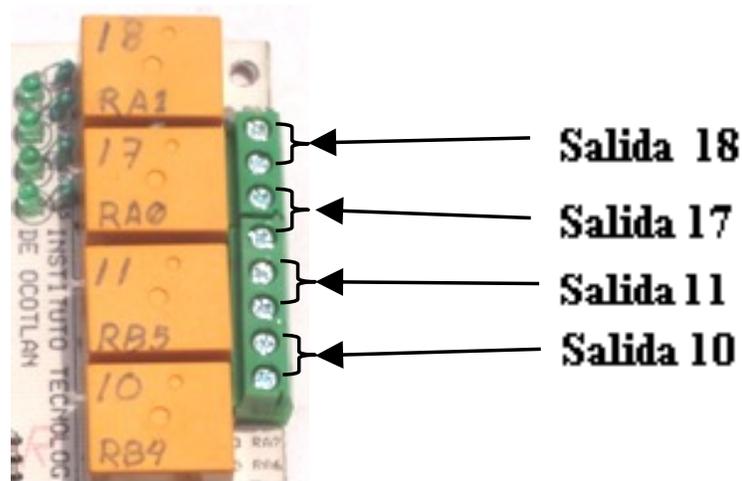


Fig. 11, Salidas, interface RSD-Micro 800, (Castellanos Hernandez, 2010)

Para la programación del Microcontrolador utilizaremos el Software LabVIEW, Según (Lajara Vizcaino, 2001), es una herramienta para la creación de sistemas SCADA, que viene de las siglas de “Supervisory Control And Data Acquisition”, es decir: adquisición de datos y control de supervisión. Se trata de una aplicación software especialmente diseñada para funcionar sobre ordenadores en el control de producción, proporcionando comunicación con los dispositivos de campo (controladores autónomos, autómatas programables, etc.) y controlando el proceso de forma automática desde la pantalla del ordenador.

La programación para controlar el prototipo es la siguiente en LabVIEW, se muestra en la figura 12.

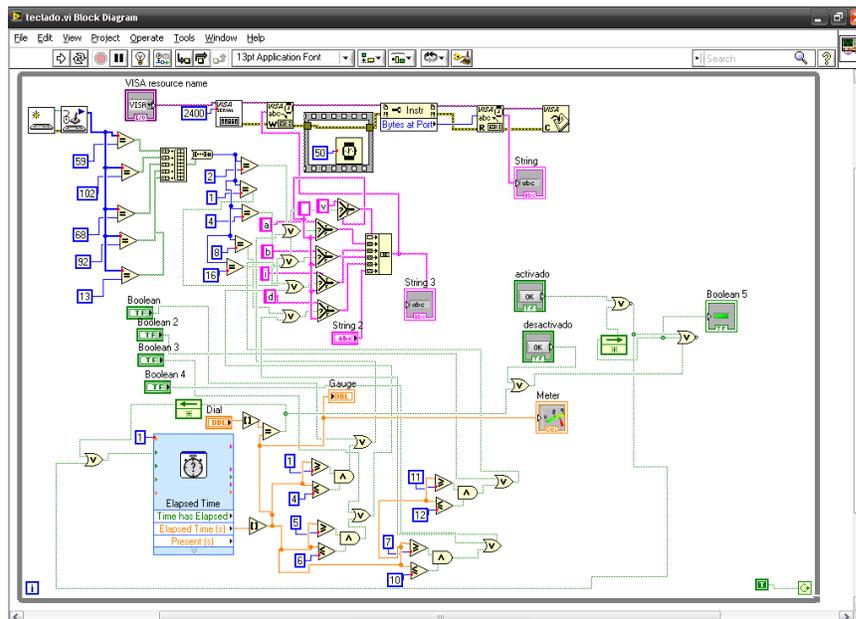


Fig. 12, La programación para controlar el prototipo en LabVIEW

Descripción de componentes se muestran en la tabla 1.

Tabla 1, Componentes

Componente	Nombre	Componente	Nombre
	Initialize KeyBoard.vi		VISA Clase
	Arquitecture Input Data.vi		String
	Visa Resour name		Equal
	Integer de 32 Bits		Build Array
	Visa Configure Serial Port		Boolean Array to Number
	Visa Write		Select
	Flat Sequence Structure (Wait(ms))		Concatenate Strings
	Property Node		Meter
	FeedBack Node		Not or
	Apagador Boolean		Round Nearest
	Elapsed Time		Greater or Equal
	Boolean 4		And
	Gauge		Gauge

Sustentó Matemático Booleano se muestra en la tabla 2. 1er. Paso. Interface del teclado a la configuración del puerto USB a Serial.

Tabla 2, Sustento Matemático Booleano, primer paso

	Visa Resour name	Visa Configure Serial Port	Visa Write	Flat Sequence Structure y Wait	Property Node	Visa Read	String	VISA Closed
Entrada	Dirección de puerto	Velocidad del puerto 2400 Bps. Com1 (03f8-03ff) v,a,b,i,d	COM2(usb) No error Cadena de datos (v,a,b,i,d)	No error	Toma pin (ver anexo A) Pagina 85	COM2 Bites al port	Visualización de datos de entrada.	Cierra Proceso de VISA
Salida	Com1 (03f8-03ff)	COM2(usb) No error	COM2 No error	Retención de tiempo a 50 Milesegundos.	COM2	No error	Combo de visualización	0

El Sustentó matemático Booleano 2do. Paso. Se muestra en la tabla 3.

Tabla 3, Sustentó matemático Booleano 2do. Paso.

	Teclado	Justic	Igualador	Build Array	Boolean Array to Numero	Igualador	Select	Concatenate String	String
Entrada	0	Valor numérico de cada tecla.	Valor numérico de cada tecla.	Valor Booleano	False ó True	1,2,4,8,16, 32,64	F,T,?	v,a,b,i,d	v,a,b,i,d
Salida	0	59,102,68, 92,13	False ó True	False ó True	1,2,4,8,16, 32,64	False ó True	variables v,a,b,i,d	v,a,b,i,d	Panel frontal visual v,a,b,i,d

El Sustentó matemático Booleano 3er. Paso, se muestra en la tabla 4.

Tabla 4, Sustentó matemático Booleano 3er. Paso

	Elapsed Time	Round To Nearest	Greater or Equal	And
Entrada	Bolean Reset	Contador	Valor Absoluto Compara con la segunda entrada si es mayor o igual	Cuando las dos entradas están activadas
Salida	Contador	Valores Enteros	Valor Booleano	Valor Booleano Amnada la activacion en al 1er. Proceso.

En la figura 13, se muestra la interface Gráfica donde se controla el prototipo.

Esta interface está realizada con LabView (Laboratory Virtual Engineering Workbench), por la cual podemos controlar el PROTOTIPO, Utilizando botones que para el movimiento de izquierda, derecha, arriba y abajo, utilizando un indicador (curvo) y uno de caratula, en los cuales verificamos el tiempo transcurrido del Elapsed Time, hasta su reinicio, contamos también con un Dial en este programa el tiempo de duración del proceso, cuanta también con 4 indicadores de estado, visualizando el puerto de salida, COM1(Serial), COM2(USB), para este proyecto utilizamos un convertidor de USB a Serial, funcionando con ambos puertos.

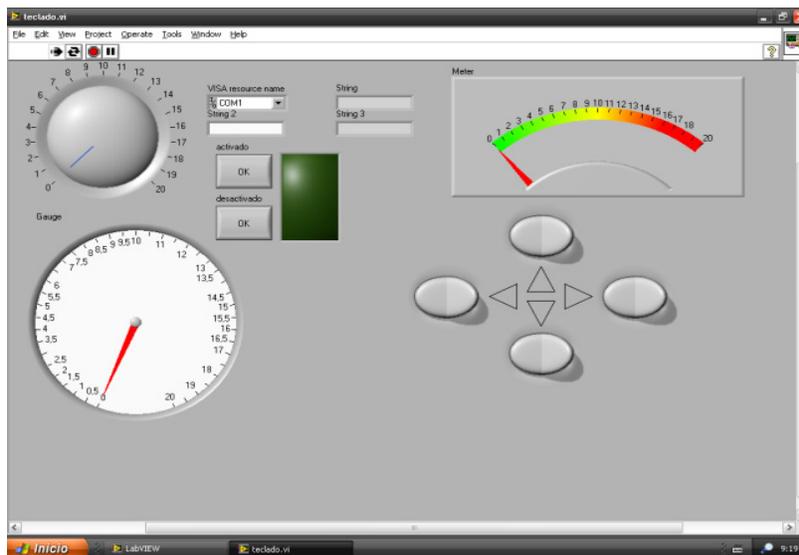


Fig. 13, Tabla 5, la interface Gráfica donde se controla el prototipo

Las conexiones de la Interface por el puerto se ven claramente en la fotografía.

En la figura 14, podemos observar que en el conector USB, que está conectado el convertidor USB a Serial (RS232), de este se conecta al convertidor de Serial al TTL (5V.), a su vez está conectado al controlador y el controlador está conectado al control Remoto del prototipo.

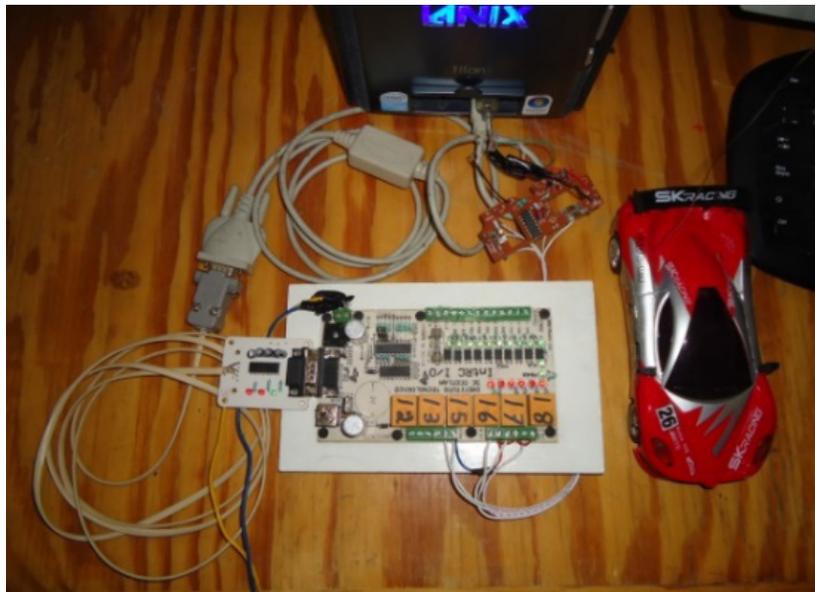


Fig. 14, podemos observar que en el conector USB, que está conectado el convertidor USB a Serial (RS232)

En la figura 15, podemos observar que en el conector USB, que está conectado el convertidor USB a Serial (RS232), de este se conecta al convertidor de Serial al TTL (5V.), a su vez está conectado al controlador y el controlador está conectado al control Remoto del prototipo.



Figura 15, Podemos observar que en el conector USB, que está conectado el convertidor USB a Serial (RS232), de este se conecta al convertidor de Serial al TTL (5V.).

En las figuras 16 y 17 respectivamente, podemos observar la tarjeta del control remoto del prototipo, en este se soldaron los cables a las Entradas (Derecha, Izquierda, Adelante, Atrás), y la alimentación está conectada al puerto USB del CPU.

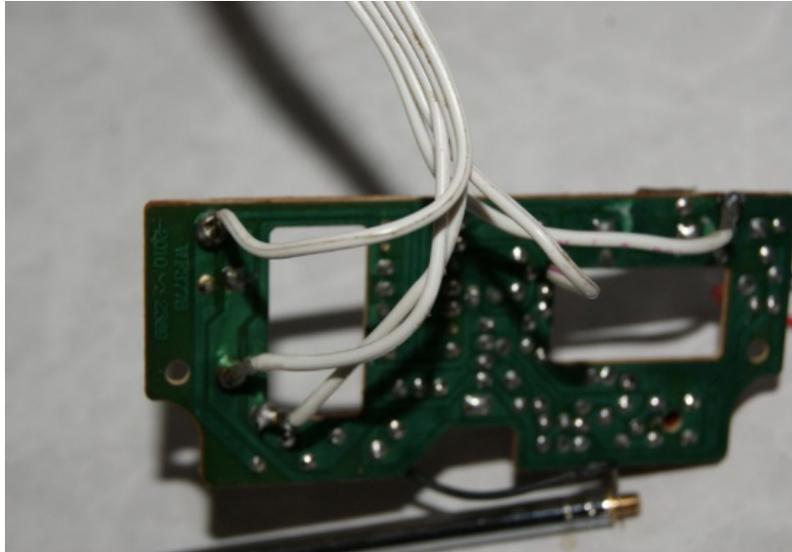


Figura 16, Podemos observar la tarjeta del control remoto del prototipo, en este se soldaron los cables a las Entradas (Derecha, Izquierda, Adelante, Atrás), reverso.



Figura 17, Podemos observar la tarjeta del control remoto del prototipo, en este se soldaron los cables a las Entradas (Derecha, Izquierda, Adelante, Atrás) de frente.

Análisis del desplazamiento del prototipo:

El movimiento: es el cambio de la posición en función del tiempo.

De acuerdo al análisis de: (Sokolovsky, 2002), Supongamos que tenemos un cronómetro para medir “ese tiempo”, a cada instante podemos designarlo con una letra, usualmente suele utilizarse la letra t. El instante en que comenzamos a medir es el instante cero, así que podemos designarlo como t_0 (te sub-cero); y asimismo se puede indicar en el subíndice el instante en el que móvil se encuentra. Por ejemplo: si transcurren 5 segundos podemos indicarlo como t_5 .

Si tomamos dos instantes cualesquiera, la diferencia entre ambos nos indicará el tiempo transcurrido entre ambos instantes:

$$\Delta t = t - t_i \text{ (el subíndice } i \text{ indica que es el instante inicial del intervalo).}$$

Este símbolo Δ (diferencial) es un elemento matemático que se utiliza para indicar la resta, “diferencia” entre dos valores de una variable y está representado por la letra griega mayúscula delta.

Si el movimiento es horizontal podemos considerar al piso como si fuera el eje de las abscisas (eje x), de esa manera cada posición se designará con la letra x. La posición correspondiente al instante cero (t_0) se designa, entonces, como x_0 , mientras que la posición correspondiente a t la denominaremos x. De manera similar a lo que hicimos con el tiempo, la diferencia entre dos posiciones cualesquiera nos permite calcular el desplazamiento lineal que hemos hecho existente entre ellas: $\Delta x = x - x_i$.

Movimiento Rectilíneo Uniforme (MRU)

El movimiento más sencillo es el movimiento en línea recta del prototipo (lógicamente denominado rectilíneo) Como todo movimiento puede describirse por el espacio que se recorre en unidad de tiempo, supongamos que recorreremos siempre la misma cantidad de espacio por cada unidad de tiempo. Imaginemos que por cada segundo recorreremos dos metros. En el primer segundo recorreremos dos metros, al segundo habremos hecho cuatro, al tercero seis y así sucesivamente...

Para facilitar aún más nuestro estudio imaginemos que partimos de la posición cero en el instante cero. Ubiquemos nuestra suposición en una tabla 5.

Instante (t)	0	1	2	3	4	5	6	7	8	9	10
Posición (x)	0	2	4	6	8	10	12	14	16	18	20

Tabla 5, posición 0, (Sokolovsky, 2002)

El espacio y el tiempo matemáticamente son directamente proporcionales, eso implica que si dividimos cada posición por el instante en que se encuentra nos dará un valor constante.

$$\frac{2}{1} = 2 \quad \frac{4}{2} = 2 \quad \frac{6}{3} = 2 \quad \frac{8}{4} = 2 \quad \frac{10}{5} = 2 \quad \frac{12}{6} = 2$$

$$\frac{14}{7} = 2 \quad \frac{16}{8} = 2 \quad \frac{18}{9} = 2 \quad \frac{20}{10} = 2$$

Físicamente ese valor constante, la razón entre el espacio recorrido y el tiempo transcurrido, se denomina velocidad $v = \frac{\Delta x}{\Delta t}$

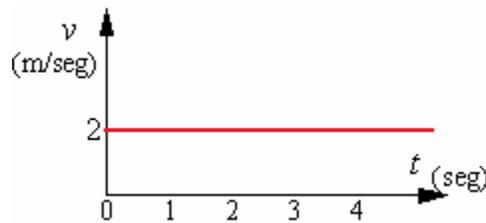


Figura 18, Ejemplo de velocidad, (Sokolovsky, 2002)

Así que la velocidad en este tipo de movimiento es constante, como se ve en la figura 18 de velocidad en función del tiempo (v(t)) donde está representada la velocidad.

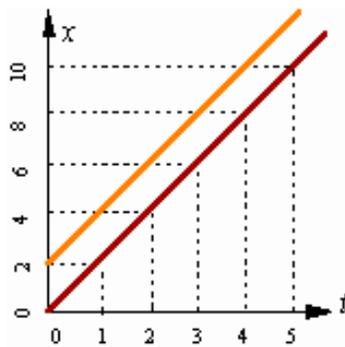


Figura 19, Posición de cada Instante, (Sokolovsky, 2002)

Si llevamos a un gráfico la posición a cada instante que está indicada en la tabla, veremos que encontramos una recta. Si observamos detenidamente la figura 19, podemos darnos cuenta de que la posición a cada instante se puede calcular multiplicando ese instante (t) por la velocidad (v), de esa manera tenemos que:

$$x = v \cdot t$$

No tiene por qué partirse de cero, así que las distintas posiciones pueden determinarse sumando la posición de donde partimos, posición inicial (x₀), y lo que se avanza (Δt.v).

Supongamos que partimos de la posición 2, la $x_0 = 2$ m, como la velocidad es 2m/seg. sumemos 2 m a la posición anterior:

Tabla 6, Instantes y Posición, (Sokolovsky, 2002)

Instante (t)	0	1	2	3	4	5	6	7	8	9	10
Posición (x)	2	4	6	8	10	12	14	16	18	20	22

Es interesante destacar que obtenemos una recta cuya pendiente es la velocidad (2) y la ordenada al origen es la posición inicial (2): matemáticamente la ecuación obtenida es: $x = 2Dt + 2$. (utilizo las variables indicadas en la tabla de instantes y posición).

De esa manera la ecuación del espacio en función del tiempo que a partir de ahora la llamaremos ecuación horaria, la escribiremos: $x = x_0 + v \cdot \Delta t$

Magnitudes vectoriales y escalares: Los números son entes abstractos que por sí solos no representan nada. Esa es su mayor virtud, pues podemos asignarle el significado que queramos. Un simple tres, según la ocasión, puede ser una cantidad de dinero, una mala nota, lo que sea ... Todo lo que podemos medir puede ser representado por un número. Todo lo medible se llamará, entonces, magnitud. Y las magnitudes pueden dividirse en dos subgrupos: escalares y vectoriales.

Supongamos que estamos mirando los coches que transitan por una avenida recta, todos los autos tendrán la misma dirección (la calle)

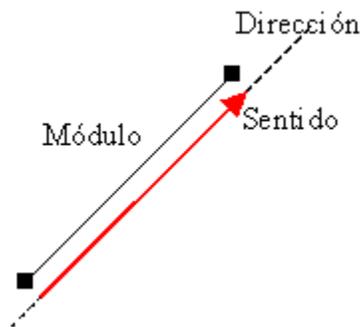


Figura 20, Módulo, Dirección y Sentido

pero no tienen que ir hacia un mismo lado, pueden poseer distinto sentido.

Es importante en un movimiento indicar la dirección (recta a la que pertenece) y el sentido en que se mueve. En matemática existe un elemento que indica sentido y dirección además del módulo (cantidad de velocidad) es el vector. A toda variable que puede ser representada por un vector la llamaremos “*magnitud vectorial*”.

Lo que nos indica la lógica es utilizar el vector para indicar la velocidad de un auto. La velocidad es una magnitud vectorial y su módulo señala su parte escalar, la cantidad que representa. Se

indica encerrando al vector entre dos líneas: $|v|$. El módulo siempre es un valor positivo.

Por supuesto que encontramos magnitudes que no pueden ser representadas por un vector, ejemplo: el tiempo. Las variables de las que sólo podemos indicar su cantidad se denominan magnitudes escalares. Para entender mejor su diferencia expliquemos un ejemplo típico:

Diferencia entre espacio recorrido y desplazamiento:

Estuvimos hablando de posiciones

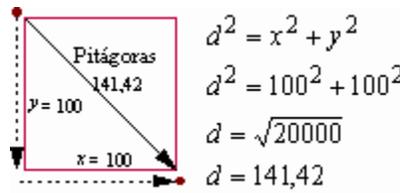


Fig. 21, Aplican teorema de Pitágoras

(x), espacio (Δx) y, aunque no lo nombramos, de desplazamiento. Pero estas tres palabras tienen distinto significado en física. Supongamos que te encuentras en una esquina, ésta será tu posición inicial y para facilitar las cosas desde allí empezaremos a contar por lo que $x_0 = 0$ m. Ahora caminas dos cuadras sobre la misma manzana. El espacio recorrido será de 200 m, ya que cada cuadra tiene 100 m, pero el desplazamiento, la línea recta que une ambas posiciones, si aplicamos teorema de Pitágoras ver figura 21 será de 141,42 m. Es más, si das la vuelta manzana, el espacio recorrido ha de ser de 400 m. pero el desplazamiento nulo.

El desplazamiento es un vector, el espacio recorrido una magnitud escalar, sólo un número.

Conclusiones

Podemos concluir que este prototipo nos da la pauta para empezar a desarrollar investigación, en el campo de electrónica, Robótica y Mecatrónica, creando nuestras propias interfaces para diferentes proyectos en el área de Electromecánica y Sistemas Computacionales, causando gran impacto en los estudiantes de ambas carreras en la generación de proyectos interdisciplinarios, cabe mencionar que este proyecto se realizó gracias a los cursos de Capacitación Docente antes impartidos en el Tecnológico de Ocotlán, por el Ingeniero Daniel Castellanos Hernández titulados:

- Programación de Microcontroladores PIC'S.
- Programación Avanzada de Microcontroladores.
- Automatización Industrial con Microcontroladores.
- Control Distribuido o Scada con Labview y Microcontroladores.

Esto nos da la pauta de trabajar en forma Interdisciplinaria con mira de crecimiento en tecnología propia, para el desarrollo de múltiples proyectos para la industria de la región.

Referencias Bibliográficas

- Castellanos Hernandez, Daniel (01 de 01 de 2010). Programación de Microcontroladores PIC'S. Ocotlán, Jalisco, México.
- Eletro tools. (2021). Eletro tools. Obtenido de Eletro tools: <https://www.electrontools.com/Home/WP/>
- Geek Factory. (2023). Geek Factory. Obtenido de Geek Factory: <https://www.geekfactory.mx/tienda/componentes/microcontroladores/pic16f628a-microcontrolador-pic-8-bits-microchip/>
- Instruments, N. (1 de Febrero de 2012). LabVIEW System Desing Software. Recuperado el 9 de Febrero de 2012, de <http://www.ni.com>
- Lajara Vizcaino, J. R. (2001). LabVIEW. Mexico: Alfaomega.
- Martinez Amador, H. (2016). Arquitectura de Computadoras, basado en competencias para nivel medio superior. Ciudad de Mexico: Exodo.
- Sokolovsky, S. (3 de Enero de 2002). Soko.com.ar. Recuperado el 3 de 01 de 2012, de http://soko.com.ar/Fisica/caida_libre.htm
- Vesga Ferreira, J. C. (2008). Microcontroladores Motorola - Freescale. México D.F.: Alfaomega.

Implementación de un Microcontrolador PIC 16F628A de la marca Microchip, por medio de un controlador RSD-Micro 800; para manipular un carro prototipo mediante el sistema RS232, por medio de un teclado de PC Ingeniería Química

P. 129-148