

Algoritmo de Comunicación Visual Basic-Microcontrolador (ARDUINO) para el control de un servomotor

María del Carmen Vargas García

Universidad Tecnológica de Tijuana

carmen.vargas@uttijuana.edu.mx

José Felipe Castañeda Ortíz

Universidad Tecnológica de Tijuana

felipe.castaneda@uttijuana.edu.mx

Jesús León Cames

Universidad Tecnológica de Tijuana

jesus.cames@uttijuana.edu.mx

Resumen

El movimiento de una banda transportadora al igual que el de un brazo robótico, requieren de un dispositivo que responda a una señal de control aplicada que genere un movimiento (Acedo Sánchez, 2003) , dentro de la gama de dispositivos existe uno que se conoce como servomotor. El contar con un algoritmo de comunicación para el control de un servomotor puede auxiliar en la capacitación de estudiantes de ingeniería, en la universidad tecnológica de Tijuana, en la carrera de ingeniería en mecatrónica en el cuatrimestre mayo- agosto 2013, el 90% de los estudiantes había tenido contacto o realizado proyectos con un servomotor, el 35% de los estudiantes que controlaron un servomotor se encontraron con problemas para hacer que dicho servomotor funcionara correctamente, como consecuencia tuvieron que dedicar más de 8 horas para investigar sobre la comunicación y funcionamiento (Hernandez Zentella, 2013). El tiempo de demora de este proceso no es muy productivo, quitando tiempo para el control y automatización de los proyectos.

El saber controlar un servomotor es de gran utilidad para la formación profesional del estudiante, y el contar con un algoritmo de comunicación entre Visual Basic –Arduino por medio de la elaboración de una aplicación para el control de un servomotor es una manera de enfocar al estudiante a la realidad industrial y darle otra alternativa para solucionar problemas.

Palabras clave: servomotor ,microcontrolador, visual basic, control

Keywords: servomotor, microcontroller, Visual Basic, algorithm, control

Introducción

Antecedentes del proyecto

La Universidad Tecnológica de Tijuana (UTT) es una universidad pública descentralizada del gobierno del estado de Baja California, que forma parte del subsistema nacional de universidades tecnológicas. Inicio sus actividades el 2 de septiembre de 1998, representando una alternativa de estudios para los egresados de escuelas del nivel medio superior, cuenta con un modelo educativo que facilita su incorporación del alumno al ámbito laboral, con clases 70% prácticas y 30% teóricas, las modalidades de estudio permiten formar técnicos superiores universitarios (TSU), e ingenieros en diferentes ramas tecnológicas, entre las que se encuentra mecatrónica. (Universidad Tecnológica de Tijuana, 2014)

La carrera de mecatrónica, permite desarrollar en los alumnos una competencia en particular que es: *“desarrollar y conservar sistemas automatizados y de control, utilizando tecnología adecuada, de acuerdo a normas, especificaciones técnicas y de seguridad para mejorar y mantener los procesos productivos”* (Sánchez , Caballero , Morales, Barajas , & Real , 2013). Al tener un modelo setenta-treinta permite que los alumnos realicen proyectos con prototipos, en los cuales incorporan el software y el

hardware, algunos de estos incluyen la utilización de servomotores, esto implica conocer el funcionamiento, las conexiones, la programación. En el periodo mayo-agosto 2013 se tomó una muestra de noventa y cinco alumnos, a los cuales se les aplicó una encuesta para identificar necesidades con respecto al control de los servomotores por un lenguaje de programación visual, se observó que el 90% de los encuestados conocen lo que es un servomotor. El 31 % de los alumnos realizaron el control físico con la placa Arduino, solo el 10% ha usado una interfaz en software para este tema. Aunque la mayoría de los alumnos conoce un servomotor, consideran que la parte de la conexión y programación de servomotores es en donde se invierte un tiempo de entre tres a ocho horas de trabajo de investigación. Este motivo ha permitido pensar en una alternativa didáctica para ayudar a los alumnos en el proceso de investigación y puesta en marcha de prácticas que involucren la utilización de un servomotor. ver figura 1, 2, 3. Llegando a lograr el desarrollo de un algoritmo de comunicación entre Visual Basic y la placa Arduino, por medio de una aplicación de software visual para el control de un servomotor industrial. (Hernandez Zentella, 2013).

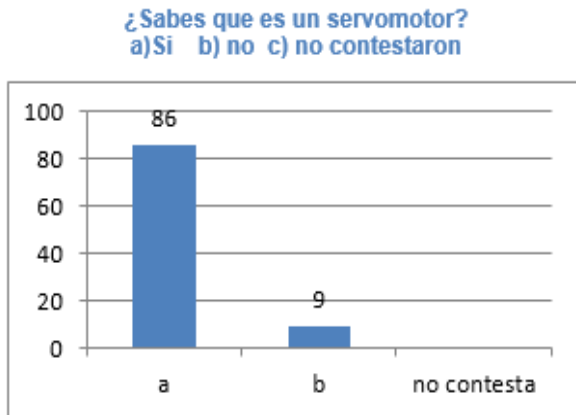


Figura 1 muestra que de noventa y cinco (100%) alumnos que fueron encuestados, ochenta y seis (90%) de ellos conocen un servomotor.



Figura 2 muestra que cincuenta y ocho (61%) alumnos ha realizado prácticas con un servomotor.

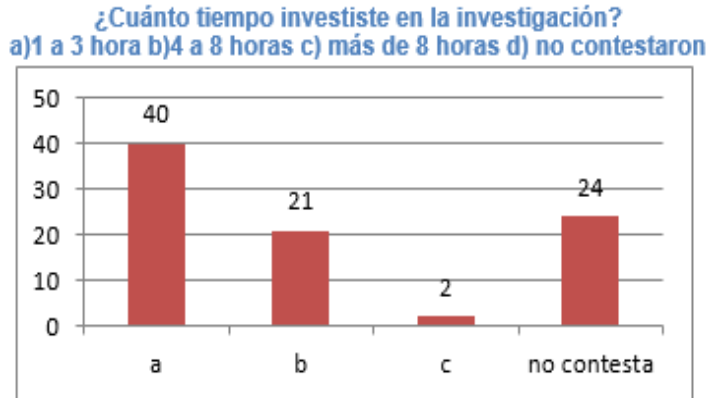


Figura 3 muestra que sesenta y un (64%) de los alumnos ha invertido de 3 a 8 horas en investigación sobre este tema.

El saber controlar un servomotor es de gran utilidad para la formación profesional del estudiante, contar con un algoritmo de comunicación entre Visual Basic–Arduino es una manera de enfocar al estudiante a la realidad industrial y darle otra alternativa para solucionar problemas de automatización.

Para poder describir los pasos (algoritmo) de solución al problema de comunicación Visual Basic – Arduino (Eslava Muñoz, 2012) se realizó un prototipo compuesto de hardware y software ver figura 4.

El componente más importante de hardware que utiliza el prototipo es el servomotor Industrial, este responde a una señal de control aplicada y genera un movimiento (Acedo Sánchez, 2003). El prototipo propuesto utiliza un servomotor industrial MITSUBISHI MR-J2S-40A, el cual cuenta con seis funcionalidades: modo velocidad, modo posición, modo par, modo posición y velocidad, modo velocidad y par, modo par y posición. El modo velocidad ayudara a desarrollar el algoritmo de comunicación, la finalidad es seleccionar tres velocidades distintas y lograr que el motor gire a la velocidad seleccionada.

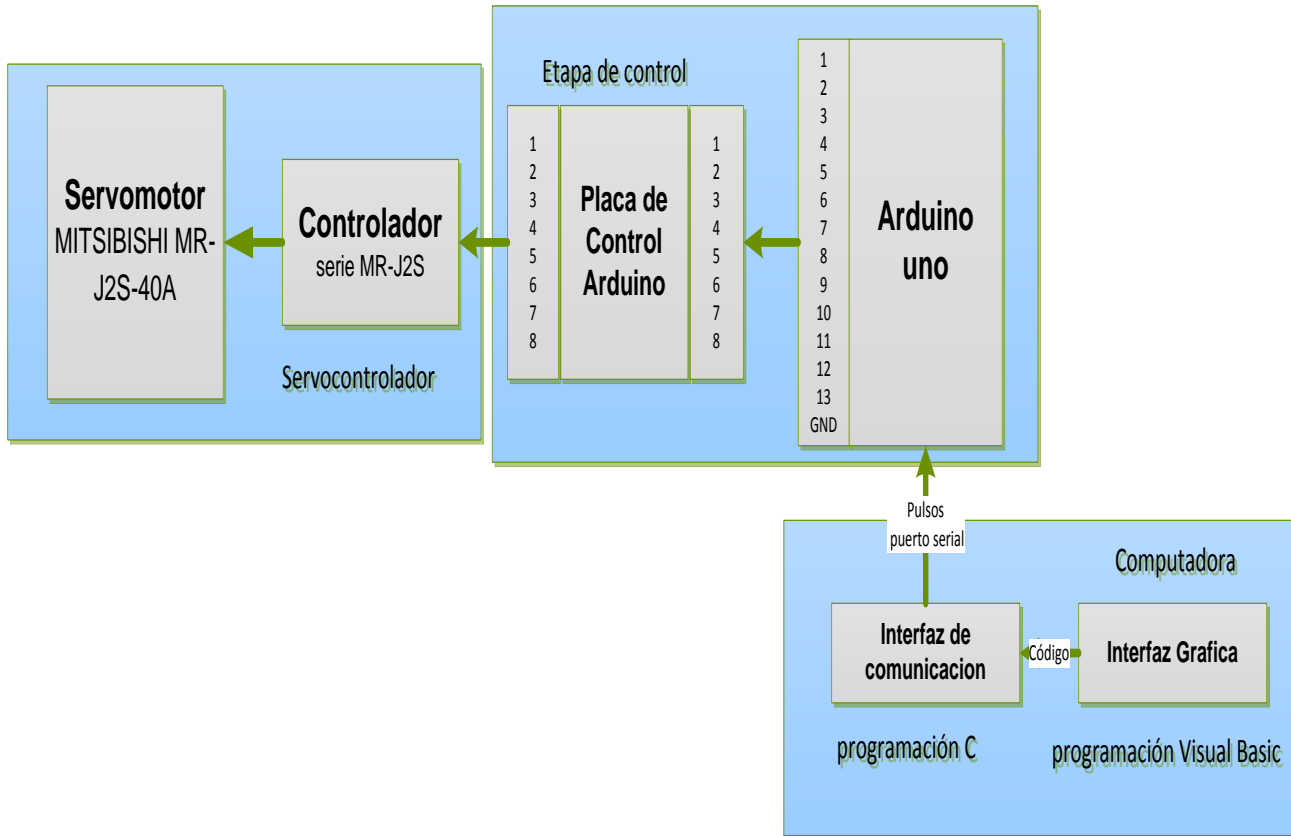


Figura 1 Diagrama Esquemático que sirve como base para desarrollar el algoritmo de comunicación Visual Basic –Arduino

El servomotor necesita de la configuración de un controlador (driver) este es el dispositivo intermediario de la parte lógica y física del control de un servomotor (Solbes Monzo, 2013), este prototipo incorpora el controlador con conmutación Sinusoidal / Brushless AC serie MR-J2S.

Para la parte de la comunicación se utiliza una placa basada en un microcontrolador de plataforma de hardware abierto llamada Arduino, la cual a través de un programa hecho en lenguaje de programación C y una comunicación serial, envía señales digitales y/o analógicas (Zavala, 2014). Las señales son enviadas a través de los catorce pines con los que cuenta la placa, y se configuran por medio de software (Arduino, 2014). Estas señales serán interpretadas por el controlador del servomotor y producirá un

cambio en la función de velocidad. El dispositivo que utiliza el prototipo es un Arduino uno con un microprocesador Atmega328 y con conexión serial USB¹.

La programación se lleva a cabo con la herramienta el Visual Studio que es un entorno de desarrollo integrado para el sistema operativo Windows (Studio, 2014), en este se manejan varios lenguajes de programación entre ellos Visual Basic 2010, este lenguaje utiliza un método que permite crear interfaces graficas de usuario (GUI) que son una combinación de imágenes y/o textos, que definen las funciones de control de un sistema a través de software. (Rodríguez Penin, 2007).

La configuración del controlador del servomotor en modo velocidad, combinada con la aplicación gráfica, programada en Visual Basic, sirve para controlar el servomotor en tres diferentes velocidades 500 revoluciones por minuto (rpm), 2000rpm, 4000 rpm estas velocidades se tomaron al azar solo para comprobar el funcionamiento.

Los siguientes apartados describen los algoritmos propuestos para la comunicación Visual Basic – Arduino.

Análisis de entradas y salidas del servomotor

Se manejan tres tipos de entradas y/o salidas una de ellas son las que se configuran en el controlador del servomotor, las otras son las que involucra la placa Arduino y las últimas son aquellas que se programan dentro del Visual Basic² y el software de la placa.

Para la configuración de las entradas y/o salidas del controlador, se necesita definir una de las seis funcionalidades con las que cuenta, destacando que el modo velocidad es el que servirá de base para el

¹Por su siglas en inglés/español: Universal Serial Bus/bus universal en serie BUS

² En este proyecto las llamaran: entradas y/o salidas lógicas.

algoritmo de comunicación. El controlador necesita ser configurado para establecer este modo³. Los parámetros que se configuran son: la velocidad máxima, la velocidad mínima, así como si se desea realizar una aceleración o deceleración fija. Esta configuración se hace a través del panel de configuración del controlador.

La configuración de las entradas y/o salidas de la placa Arduino necesitan de una interfaz vía serial, que requiere de la instalación de un controlador de software, el cual ya viene incluido en la placa Arduino. El controlador se instala una vez que se conecta la placa Arduino a la entrada de la USB de la computadora, la instalación es de manera automática, con ella se asigna un puerto serial de comunicación⁴. También es necesario un software para programar el microcontrolador de la placa Arduino en el se realiza un programa con el fin de controlar las entrada y salidas a manejar.

Para la comunicación lógica con Visual Basic se declaran variables dentro del software que actúan como entradas y/o salidas lógicas.

Como parte del proceso didáctico, que se pretende con este trabajo se desarrollaron algoritmos. La finalidad consiste en llevar paso a paso al alumno, en la configurar las entradas y /o salidas descritas anteriormente, estos algoritmos se documentaron en un manual de usuario.

Para poder realizar la configuración de las entradas y/o salidas de la placa Arduino requiere seguir el algoritmo de la figura 5

³ La velocidad que maneja cada servomotor depende del fabricante. El servomotor MITSUBISHI MR-J2S-40A maneja rangos que van de 0 a 3000 rpm.

⁴ Para ver que puerto fue asignado usamos la siguiente dirección: Panel de control\Todos los elementos de Panel de control\Sistema \ administrador de dispositivos.

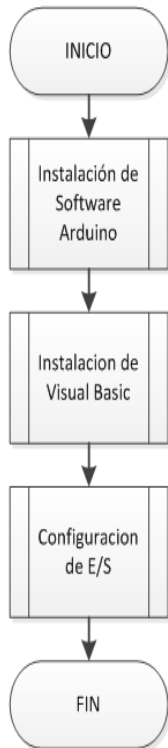


Figura 2 pasos para la configuración de entradas y salidas

Instalación del software Arduino

1. Descargar de la página de Arduino el instalador⁵
2. Se siguen los pasos que marca el asistente de instalación; no es necesario configurar algún parámetro. El resultado de este paso es un acceso directo en el escritorio.
3. Al dar doble clic al icono se abrirá la pantalla que permitirá introducir el código en lenguaje C, que controlara las entradas y salidas.

⁵ <http://www.arduino.cc/>

Instalación del software Visual Basic

4. Se ejecuta el instalador del Visual Studio⁶
5. Se configuran las características de los pasos que el asistente de instalación marca.
6. Una vez finalizada la instalación se podrá utilizar el software de Visual Basic

El proceso de comunicación se define en el algoritmo de la figura 6, y las instrucciones que ejecutan la comunicación se aprecian en la tabla1 y 2

Tabla 1 Código para comunicación Visual Basic (Microsoft, 2014)

Función/propiedad	Descripción
Open() Close ()	Abre o cierra el puerto de comunicación
PortName	Obtiene o establece el puerto de comunicaciones, incluidos por lo menos todos los puertos COM disponibles.
DiscardInBuffer()	Descarta los datos del buffer de recepción del controlador serie.
Write()	Escribe datos en el búfer de salida del puerto serie.
ReadExisting()	Lee todos los bytes inmediatamente disponibles, basándose en la codificación, en la secuencia y el búfer de entrada del objeto SerialPort.

⁶ <http://www.visualstudio.com/products/visual-studio-online-overview-vs>

Tabla 2 código para la comunicación en el software de la placa Arduino (Evans, 2011)

pinMode(pin, mode)	Sirve para configurar el modo de trabajo de un PIN pudiendo ser INPUT (entrada) u OUTPUT (salida).
Serial.Read()	Lee o captura un byte (un carácter) desde el puerto serie
Serial.available()	Devuelve un entero con el número de bytes disponibles para leer desde el buffer serie, o 0 si no hay ninguno
digitalWrite(pin, value)	Envía al 'pin' definido previamente como OUTPUT el valor HIGH o LOW (poniendo en 1 o 0 la salida)

Algoritmo de configuración de entradas y/o salidas Visual Basic-Arduino

1. En visual Basic, se inserta el objeto *SerialPort* el cual se configura para realizar la comunicación.
2. Se configuran la propiedad *PortName* en el objeto *SerialPort* asignándole el puerto de comunicación COM+ número de puerto⁷.
3. Por medio de código en lenguaje Visual Basic se cargan las variables de comunicación y se habilitan el puerto de entrada y/o salida.
4. Se envía por el puerto la señal.
5. En el software de Arduino se configuran las entradas y/o salidas

En el apartado de comunicación Visual Basic – Arduino se desarrolla la explicación de la comunicación.

Desarrollo de la etapa de control

En el software de la placa del Arduino se realiza un programa con el fin de controlar la entrada y/o salidas, para poder verificar que estas instrucciones funcionen, se realizó una tarjeta con un bloque de relevadores de cinco volts para activación de las entradas necesarias para el movimiento del servomotor

⁷ El número de puerto se asigna cuando se instala el software de programación de la placa Arduino

a este proceso se nombrara etapa de control. El objetivo de esta etapa es generar un medio seguro, ya que separa la etapa de control de la etapa de potencia. El algoritmo que se describe a continuación permite realizar la conexión entre la placa Arduino y la placa de control.

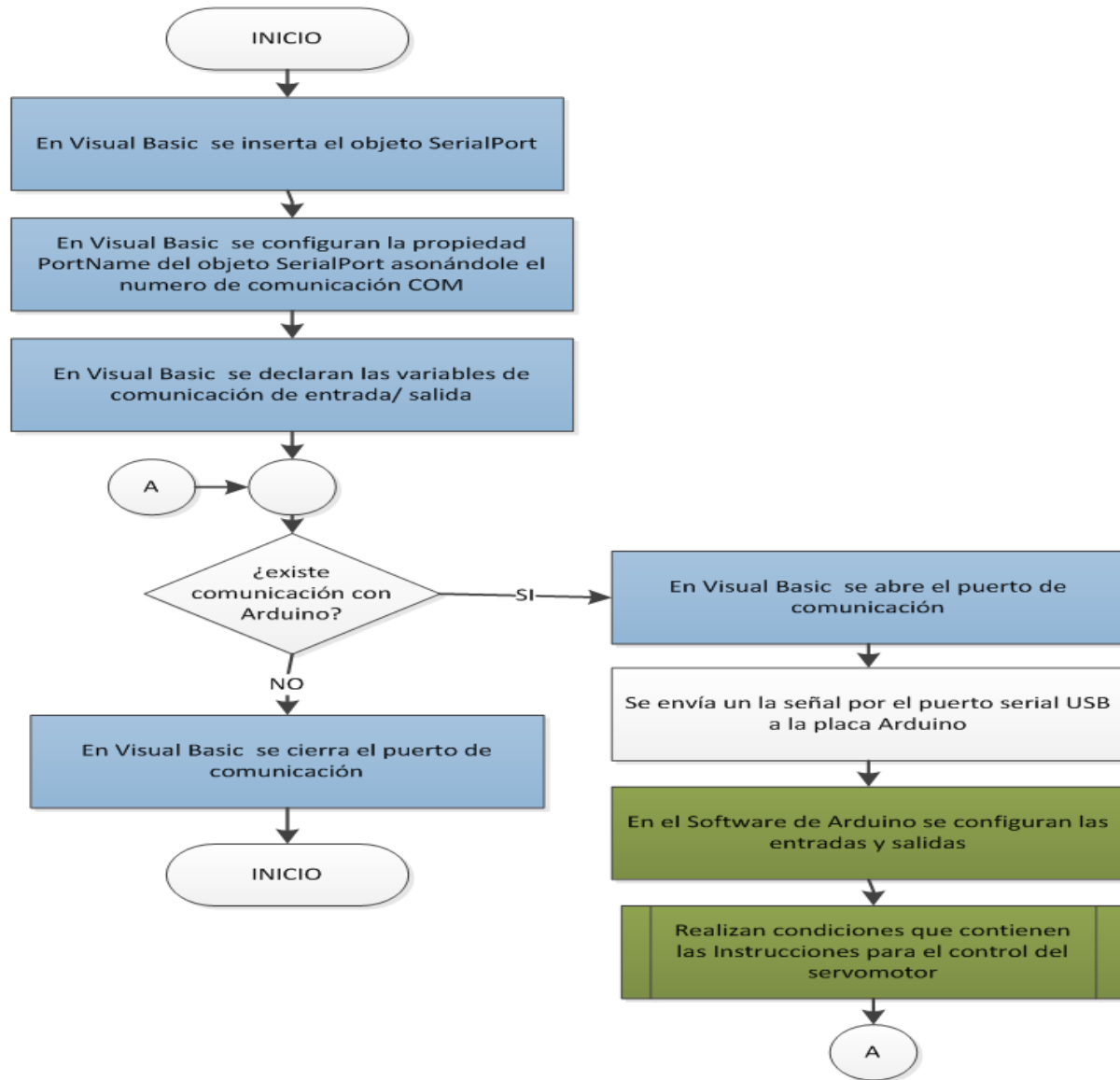


Figura 3 Algoritmo para el proceso de comunicación Visual Basic- Arduino; el color azul en el diagrama representa las operaciones realizadas en Visual Basic, el color verde en el diagrama representa las operaciones realizadas en el software de la placa Arduino

Algoritmo para la configuración de la etapa de control

Con ayuda del diagrama de bloque de la figura 7 se realizan las conexiones correspondientes.

1. Activan la función en el driver Servo ON. (Encender el Servomotor). Arduino pin 8 al placa de control pin 1
2. Activan la función en el driver Velocidad (Activa la velocidad dos en el servomotor). Arduino pin 9 al placa de control pin 2
3. Activan la función en el driver Forward. (Activa el Servomotor hacia Adelante). Arduino pin 10 al placa de control pin 3
4. Activan la función en el driver Reverse. (Activa el Servomotor hacia Atrás). Arduino pin 11 al placa de control pin 4
5. Activan la función en el driver Velocidad 1. (Activa la velocidad uno en el Servomotor). Arduino pin 12 al placa de control pin 5
6. Activan la función en el driver Dirección en el modo Posición. (Activa la Reversa en el Servomotor). Arduino pin 13 al placa de control pin 6
7. Activan la función en el driver Entrada de los pulsos en el modo Posición. (Activa los pulsos para el modo Posición en el Servomotor). Arduino pin 6 al placa de control pin 7
8. Conectar la tierra de la placa de control para que los optoacopladores realicen su función. Arduino pin GND al placa de control pin 8

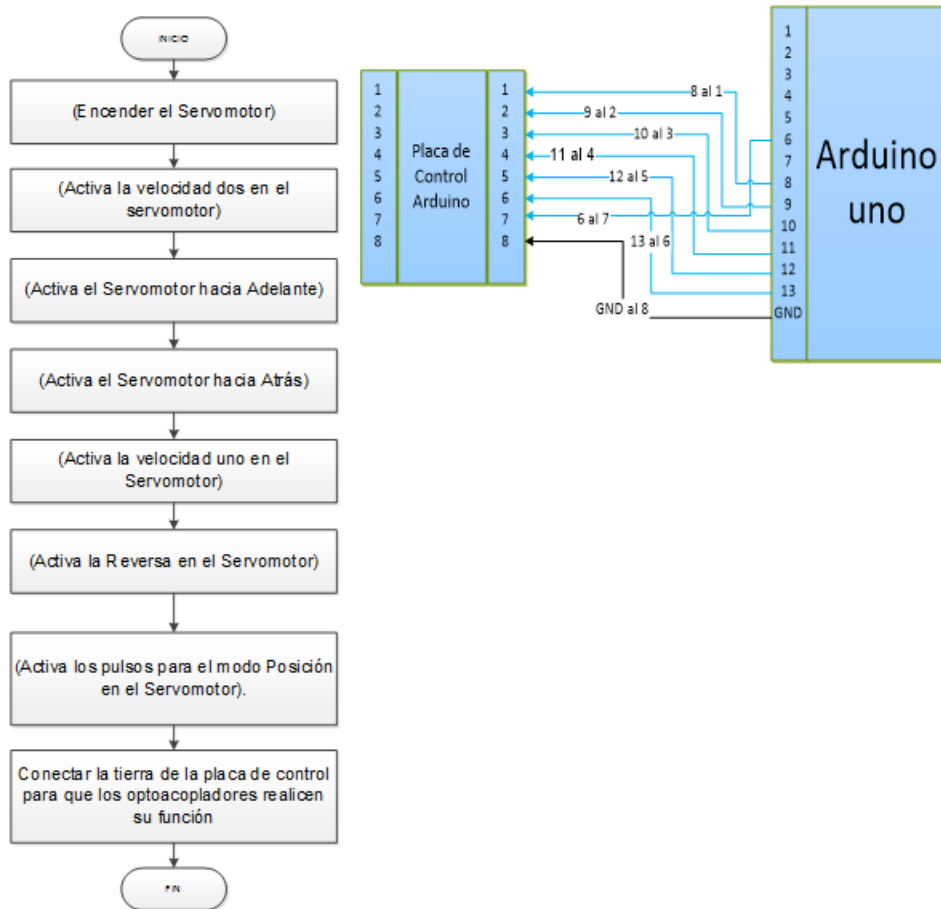


Figura 4 algoritmo y diagrama de bloques de la conexión de la etapa de control con la placa Arduino

Para definir la conexión entre la placa de control y el controlador del servomotor, se tiene que configurar le modo que se desee controlar, como ya se mencionó antes la configuración se hará en modo velocidad.

Algoritmo de configuración del modo velocidad

Con ayuda del diagrama de bloque de la figura 8 se realizan las conexiones correspondientes.

1. Conectar de la salida de la placa de control (pin 1, 2, 3,4), al conector CN1B (clemas: 5, 7, 8,9)
2. Conexión de salidas de la placa de control (5) al conector (CN1A: 8) del controlador.
3. Conexión del sensor limite izquierda Pin NC al controlador (CN1B: 17)
4. Conexión del sensor limite derecha Pin NC al controlador (CN1B: 16)
5. Conexión del paro de emergencia NC al controlador (CN1B: 10)

La implementación de los algoritmos, permitió realizavr la comunicación entre las interfaces lógicas y físicas, el siguiente apartado complementa el proceso que se ha estado describiendo.

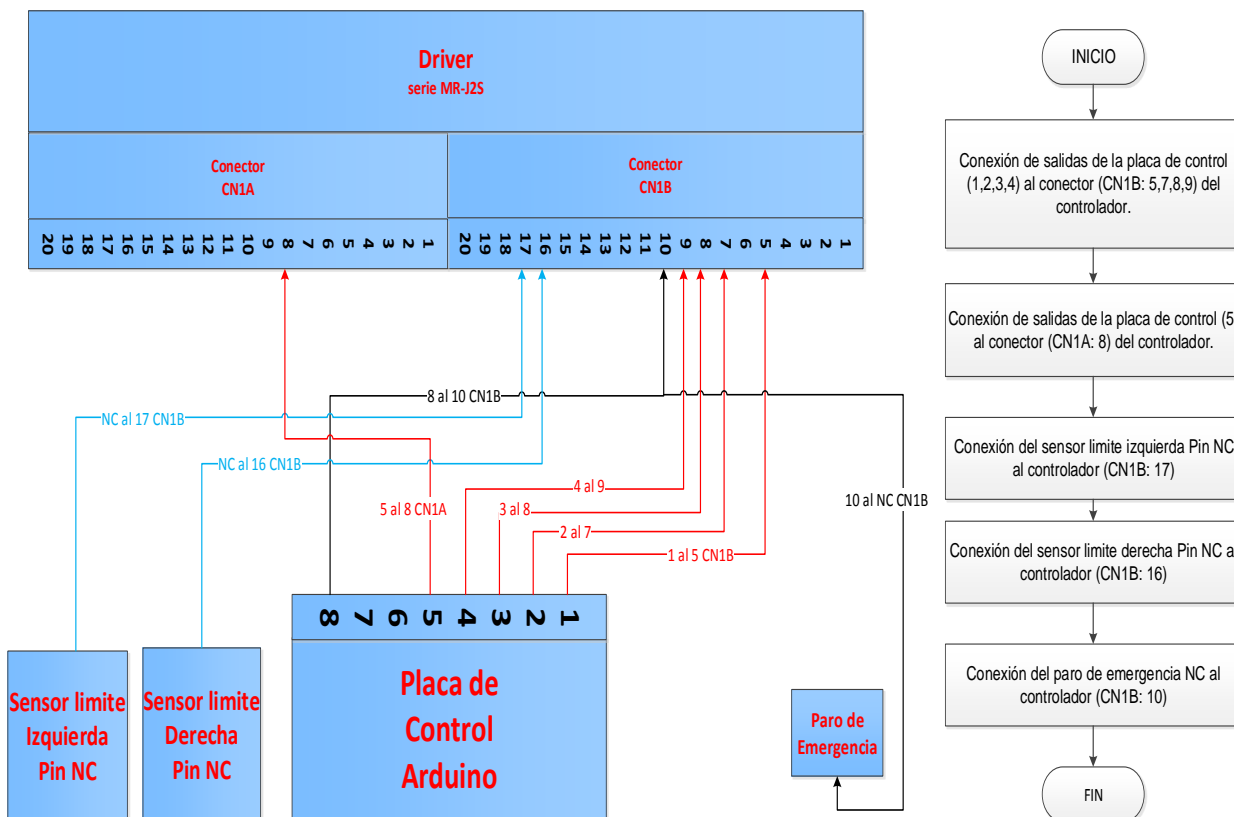


Figura 5 diagrama de bloques y algoritmo de la configuración del modo velocidad

Comunicación Visual Basic- Arduino

Parte del proceso de configuración de entradas y/o salidas figura 6 se encuentra el proceso de comunicación figura 9, en el cual se describen los pasos para lograr el proceso de comunicación.

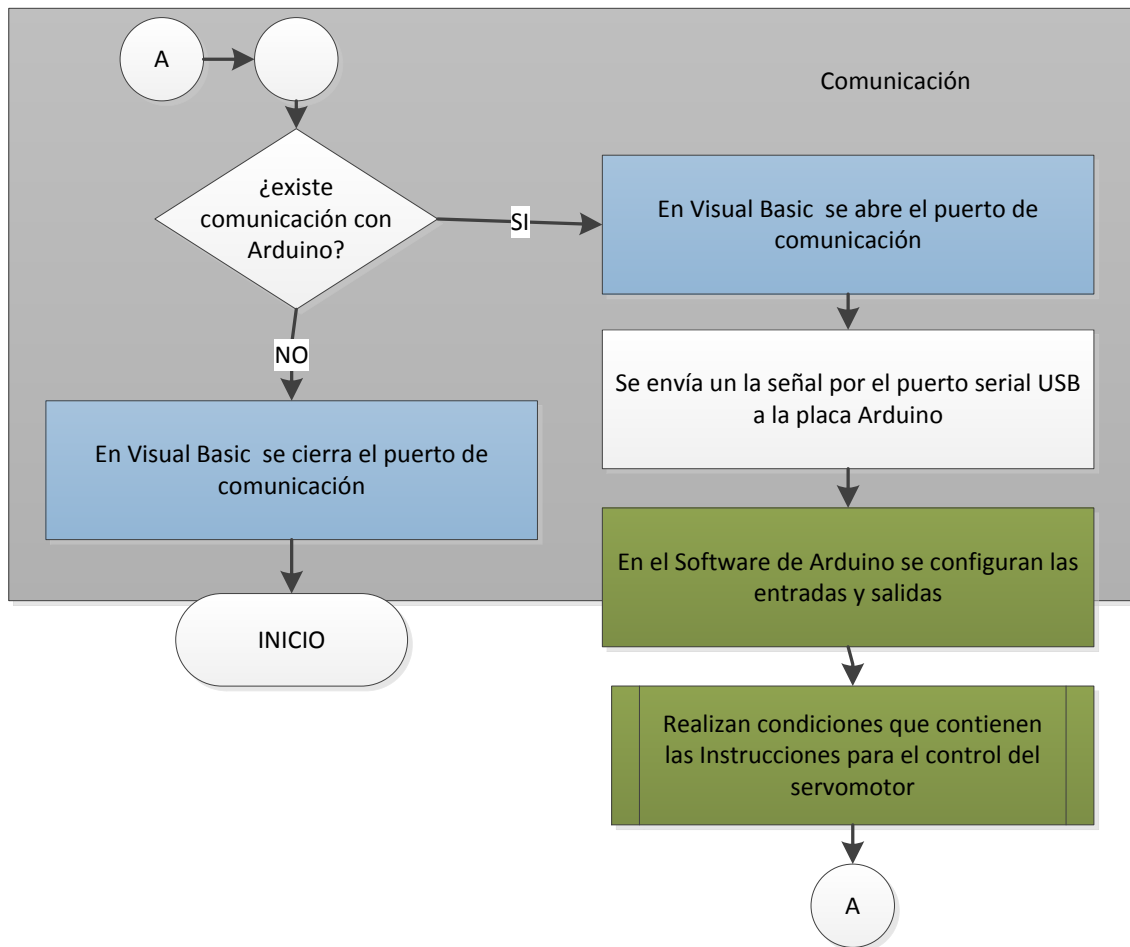


Figura 6 Algoritmo de comunicación Visual Basic- Arduino

Algoritmo de comunicación Visual Basic- Arduino

1. Si existe una señal de Visual a Arduino entonces
 - a. Se abre el puerto de comunicación
 - b. Se envía la señal
 - c. Se configuran en el software de la placa del Arduino las variables de entrada y/o salida.
 - d. se generan las condiciones que permitirán la ejecución de instrucciones que realizara el servomotor.
 - e. se regresa al paso 9
2. si no hubiera comunicación entonces
 - a. se cierra el puerto de comunicación

Para realizar el proceso de comunicación se tiene que desarrollar una aplicación de software, y en ella insertar el código de los comandos que permiten las conexiones lógicas. Los detalles de esta aplicación se explicaran el apartado aplicación de software.

Aplicación de software

Para poder probar la funcionalidad de los algoritmos se diseñó una aplicación de software, con la herramienta Visual Studio 2010 en el lenguaje de programación Visual Basic 2010 que consta de dos pantallas, la primera que se observa en la figura 10, muestra la conexión del puerto serial. La interfaz guía al usuario sobre los pasos que tiene que seguir para realizar la conexión, los cuales consisten en determinar el tipo de conexión, el puerto de comunicación y posteriormente del modo de funcionalidad del servomotor.

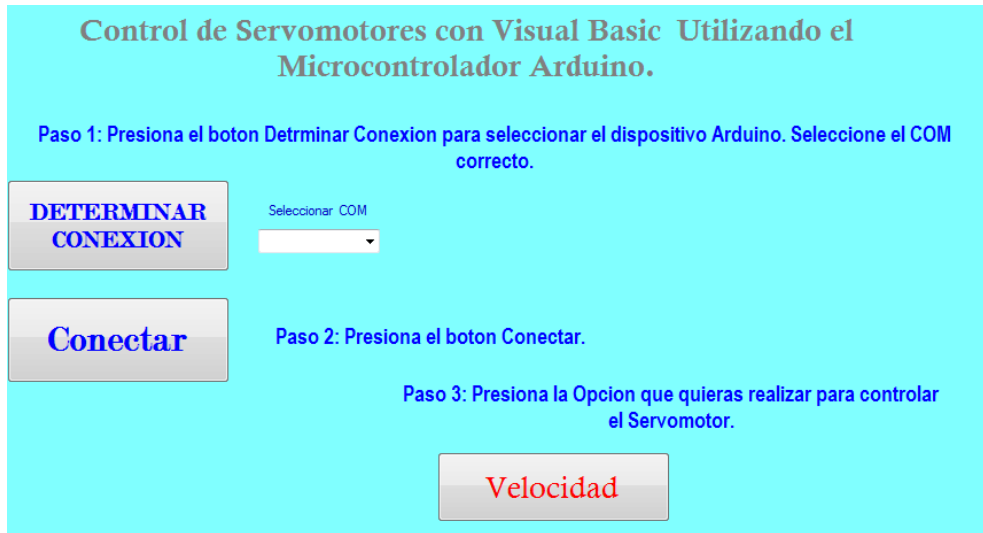


Figura 7 pantalla principal de la aplicación

La segunda pantalla figura 11, permite seleccionar tres velocidades diferentes así como el tipo de desplazamiento del motor adelante, detener (STOP) y atrás. En esta pantalla también se podrá detener el servomotor, además de poder regresar a la pantalla principal.



Figura 8 pantalla para el control de Velocidad.

Como continuación de este trabajo se pretende configurar los demás modo de funcionalidad con los que cuenta el servomotor. Además de agregar un diseño ergonómico a las interfaces.

Pruebas

Para las pruebas se desarrolló un prototipo físico figura 12, que consiste en un servomotor con un par de sensores ubicados en las orillas de la mesa son los límites de la puesta en marcha. Cuenta también con un paro de emergencia. En la parte eléctrica se observó un contacto trifásico para el control de encendido del controlador, también una fuente de veinticuatro volts para activación de los relevadores. Para los cables de entrada y salidas del controlador se utilizaron cuarenta clemas, veinte para el conector CN1A y veinte para el conector CN1B, esto con el fin de poder conectar y activar las entradas necesarias para el control del servomotor.

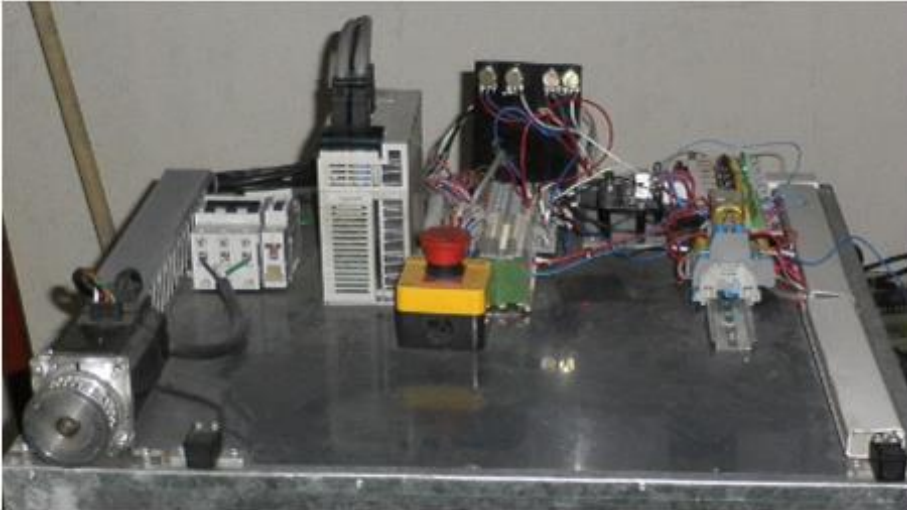


Figura 9 prototipo físico

Con este prototipo y las conexiones descritas en los apartados anteriores se logró verificar el funcionamiento de los algoritmos.

Por otra parte, la información obtenida de esta investigación, sirvió de apoyo para las materias que se imparten en la carrera de mecatrónica en la UTT, en particular la materia de programación visual, en donde se les pidió a los alumnos que siguieran los pasos descritos por los algoritmos, he implementaran la comunicación con una aplicación diferente. El tiempo que invirtieron en las conexiones de comunicación disminuyó, se produjeron menos errores relacionados con este tema, este proyecto sirvió para generar nuevas propuestas de comunicación.

Bibliografía

Acedo Sánchez, J. (2003). servomotor. En *Control Avanzado de Procesos* (pág. 91). Díaz de Santos.
Arduino. (2014). *Arduino*. Recuperado el 2 de marzo de 2014, de Arduino: <http://arduino.cc/en/Main/arduinoBoardUno>

- Eslava Muñoz, V. J. (2012). Concepto de Algoritmo. En V. J. Eslava Muñoz, *Aprendiendo a programar paso a paso con C.* (págs. 26-27).
- Evans, B. (2011). *arduino*. Ardumania.
- Hernandez Zentella, R. (2013). Encuesta para proyecto de estadias . Tijuana.
- Microsoft. (2014). *developer Network, Microsoft* . Recuperado el 5 de mar de 2014, de developer Network, Microsoft : <http://msdn.microsoft.com/dn308572#fbid=OSGpTxKq4X8>
- Rodríguez Penin, A. (2007). Interfaces Graficas . En A. Rodríguez Penin, *Sistema SCADA* (pág. 51). Barcelona, España: Marcombo.
- Sánchez , S. J., Caballero , B. A., Morales, M. K., Barajas , C. R., & Real , V. K. (2013). PERFILES DE LAS CARRERAS OFRECIDAS EN LA UNIVERSIDAD TECNOLÓGICA DE TIJUANA . En S. J. Sánchez, B. A. Caballero, M. K. Morales, C. R. Barajas, & V. K. Real, *MANUAL CURSO PROPEDEÚTICO 2013 PARA ALUMNOS DE NUEVO INGRESO A NIVEL TÉCNICO SUPERIOR UNIVERSITARIO* (págs. 10-11). Tijuana: secretaria academica.
- Solbes Monzo, I. R. (2013). Servomotores. En R. Solbes Monzo, *Automatismos Industriales. Conceptos y procedimientos* (pág. 225). Valencia: Nau Libres.
- Studio, V. (2014). *Corporacion Visual Studio*. Recuperado el 4 de mar de 2014, de Corporacion Visual Studio: <http://www.visualstudio.com/explore/features-overview-vs>
- Universidad Tecnologica de Tijuana. (12 de feb de 2014). *Portal-Universidad Tecnológica de Tijuana*. Obtenido de Portal-Universidad Tecnológica de Tijuana: <http://portal.uttijuana.edu.mx/Portal/Historia.aspx>
- Zavala, G. (2014). Arduino como Plataforma de open hardware . En G. Zavala, *Robotica* (pág. 31). USERSHOP.