

Arduino Uno, retos y oportunidades

Lara Urzua Martin Alejandro, Buelna Benítez Andrés, Quintero Rosas Verónica y Diaz-Ramírez Arnoldo

RESUMEN—Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. En este artículo se presenta una de la gran cantidad de aplicaciones que se pueden implementar con arduino, haciendo notar que es fácil de usar y permite mucha flexibilidad al momento de utilizar el hardware y agregar nuevos componentes. Esta implementación consiste en controlar un brazo robótico por medio de Arduino, permitiendo realizar los movimientos de este por medio de secuencias predefinidas y por medio del teclado de una computadora.

PALABRAS CLAVE—Arduino UNO, Microcontrolador, brazo robótico.

I. INTRODUCCIÓN

Con el propósito de hacer notar a distintos lectores sobre las facilidades que ofrece Arduino al momento de desarrollar o implementar un aplicación con microcontroladores y los beneficios que se ponen a su disposición gracias a las herramientas con las que se cuenta, desde su propio software para desarrollar los programas del controlador hasta la gran variedad de tutoriales y ejemplos con los que se facilita el aprendizaje y la utilización de toda funcionalidad que pueda ofrecer, además se cuenta con una gran comunidad de usuarios en la Web siempre dispuestos a ofrecer su conocimiento para resolver cualquier problema que se pueda presentar. Para efectos de este artículo se toma como ejemplo la implementación de un control que permite manejar un brazo robótico con Arduino Uno, por medio del teclado de una computadora o secuencias predefinidas.

El resto del documento está organizado de la siguiente manera. En la Sección II se presentan brevemente las especificaciones básicas de la placa Arduino Uno. En la Sección III se presenta la arquitectura del sistema desarrollado. En la Sección IV se discuten algunas de las herramientas disponibles utilizadas para implementar la aplicación del brazo robótico. Los detalles y sugerencias de la implementación de prototipos se muestran en la Sección V. Las conclusiones y el trabajo futuro se presentan en la Sección VI.

II. ESPECIFICACIONES BÁSICAS DE ARDUINO UNO Cabe mencionar el funcionamiento del circuito integrado

Arduino Uno como se menciona en la guía de usuario [1] es una plataforma de prototipos electrónica de código

Martin Alejandro Lara Urzua es estudiante de Ingeniería en Sistemas Computacionales del Instituto Tecnológico de Mexicali, Baja California, e-mail: mlaraurzua@itmexicali.edu.mx..

Andrés Buelna Benítez es estudiante de Ingeniería en Sistemas Computacionales del Instituto Tecnológico de Mexicali, Baja California, e-mail: andres.buelna.b@gmail.com.

Verónica Quintero Rosas es profesora del Departamento de Sistemas y Computación del Instituto Tecnológico de Mexicali, Baja California, e-mail: veroquintero@yandex.ru.

Arnoldo Díaz Ramírez es profesor-investigador del Departamento de Sistemas y Computación del Instituto Tecnológico de Mexicali, Baja California, e-mail: adiaz@itmexicali.edu.mx.

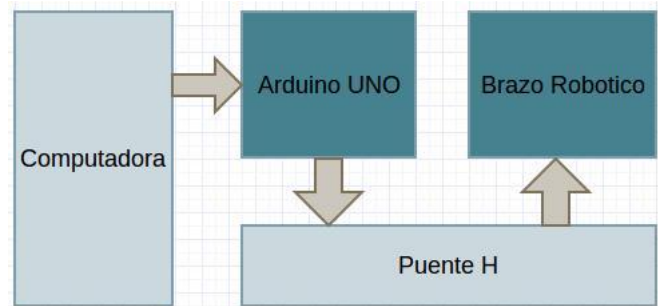


Figura 1. Arquitectura del sistema

abierto basado en hardware y software flexibles y fáciles de usar, pensado para artistas, diseñadores y para cualquiera interesado en crear objetos o entornos interactivos. Es una placa electrónica basada en el microcontrolador Atmega328. Cuenta con 14 pines digitales de entrada/salida, 6 entradas analógicas, un resonador cerámico de 16 MHz, una conexión USB, un conector de alimentación, un header ICSP, y un botón de reinicio. Cuenta con todo lo necesario para utilizar el microcontrolador simplemente conectándolo a una computadora por medio de un cable USB, o alimentarla con un adaptador de corriente AC a DC.

III. ARQUITECTURA DEL SISTEMA

Como puede observarse en la Fig. 1, la arquitectura de esta implementación se compone de dos componentes principales, como son: el brazo robótico y la placa Arduino Uno que controla al robot. En la imagen se muestra que la placa Arduino, al recibir una señal por parte de la computadora se comunica con el robot por medio de un puente H o inversor de polaridad que gracias a esta funcionalidad permite mover los distintos motores que utiliza el robot en sus dos posibles direcciones, en este caso se presentan solo dos motores pero el brazo robótico utilizado durante este proyecto contiene cinco motores que permiten realizar todos sus movimientos.

Contiene 8 pines por lado como se muestra en la Fig. 2, de los cuales 4 pines son digitales (2,7, 10, 15) y se utilizan para controlar la dirección de los motores con un high (1) o un low (0). Los pines enable (1,9) admiten como entrada una

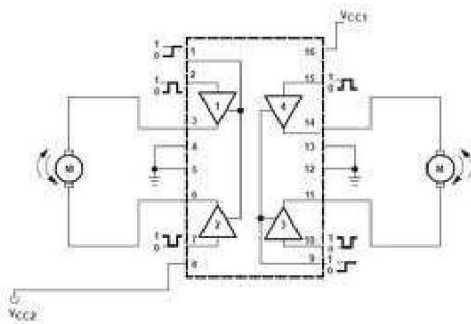


Figura 2. Circuito integrado inversor de polaridad o puente H

```
const int enPin = 5; // PWM se conecta al pin 1 del puente-H
const int gripper = 7; // Entrada 2 del puente-H
const int elbow = 4; // Entrada 7 del puente-H

void setup() //Inicializa y prepara los valores
{
  Serial.begin(9600); //Abre el puerto serie a 9600 bps
  pinMode(gripper, OUTPUT); //Configura gripper como salida
  pinMode(elbow, OUTPUT); //Configura elbow como salida
  Serial.println("Speed (0-9) or + - to set direction");
}
```

Figura 3. Ejemplo Código parte 1

señal PWM (por sus siglas en inglés pulse-width modulation), se utiliza para controlar la velocidad de los motores con la técnica de modulación de ancho de pulso. Los motores van conectados entre uno de los pines 3, 6, 11, o 14. Así que cuando un pulso (0 o 1) es enviado por los pines digitales ira en sentido o en contra a las manecillas del reloj, por ejemplo un pulso 1 al pin 2 y un 0 al pin 7 dará un giro hacia la derecha (clockwise) y si invertimos los pulsos, enviando un 0 y un 1 a los pines 2 y 7 respectivamente nuestro giro será en reversa (anticlockwise).

IV. TECNOLOGÍAS Y HERRAMIENTAS

En esta sección se presentan brevemente distintas tecnologías y herramientas, que se utilizaron para la implementación del sistema, como son: el lenguaje de programación Arduino para codificar el algoritmo que permite controlar los movimientos del hardware, el entorno de código abierto Arduino que facilita el escribir el código y cargarlo en la placa por medio del puerto de E/S, un brazo robótico modelo OWI-007 que se estará controlando por medio de Arduino.

IV-A. Lenguaje de programación Arduino

El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino, este lenguaje se encuentra basado en Wiring que es un framework de programación libre para microcontroladores, permite escribir software multi-plataforma para controlar dispositivos conectados a una amplia variedad de tarjetas microcontroladoras para crear toda clase de objetos o espacios interactivos o experiencias físicas [3].

Como se explica en su página oficial [4], los programas hechos con Arduino se dividen en tres partes principales: estructura, valores (variables y constantes), y funciones. El

```
void loop() //La función loop se ejecuta consecutivamente
{
  char ch;
  if (Serial.available()) //Si el puerto serie esta disponible
  {
    ch = Serial.read(); //variable tipo char desde puerto serial
    if (ch >= '0' && ch <= '9') //Si ch es un numero
    {
      int speed = map(ch, '0', '9', 0, 255);
      analogWrite(enPin, speed); //0 a 0 '9' a 255
      Serial.println(speed);
    }
    else
    {
      switch(ch)
      {
        case '+':
          digitalWrite(gripper,HIGH);
          break;
        case '-':
          digitalWrite(gripper,LOW);
          break;
        case 'w':
          digitalWrite(elbow,HIGH);
          break;
        case 's':
          digitalWrite(elbow,LOW);
          break;
      }
    }
  }
}
```

Figura 4. Ejemplo Código parte 2

Lenguaje de programación Arduino se basa en C/C++. Cuenta con funciones ya conocidas además de los tipos de datos estándar, agrega nuevas funciones para controlar las entradas y salidas, ya sea en formato analógico o digital. Para demostrar lo sencillo que puede ser un programa desarrollado con este lenguaje, en la Fig. 3 se puede ver como se declaran las constantes y se inicializan en la función setup utilizadas para hacer un manejo básico del brazo robótico, permitiendo controlar el codo y la pinza de este. En la Fig. 4 se puede ver como quedaría la función loop para controlar el robot, con las teclas W y S para subir y bajar el codo y las teclas + y - para abrir y cerrar la pinza.

IV-B. Entorno de código abierto Arduino

En la Fig. 5 se muestra el entorno gráfico que ofrece el IDE de Arduino para facilitar la codificación de los programas y cargarlos en la placa. El entorno de código abierto Arduino hace fácil escribir código y cargarlo a la placa E/S. Funciona en Windows, Mac OS X y Linux [1]. Está escrito en Java y basado en Processing, avr-gcc y otros programas también de código abierto. Processing como se menciona en el libro [5] es un lenguaje de programación y el entorno de desarrollo para las comunidades artísticas de los medios. Está creado para enseñar los fundamentos de la programación informática en el contexto de artes los medios y para servir como apoyo hacia los estudiantes, permitiendo el desarrollo ágil de prototipos.

IV-C. Brazo robótico OWI-007

El modelo utilizado es un OWI-007 el cual como se explica en el manual de usuario [6] es un brazo robótico de entrenamiento para enseñar los principios básicos de la robótica.

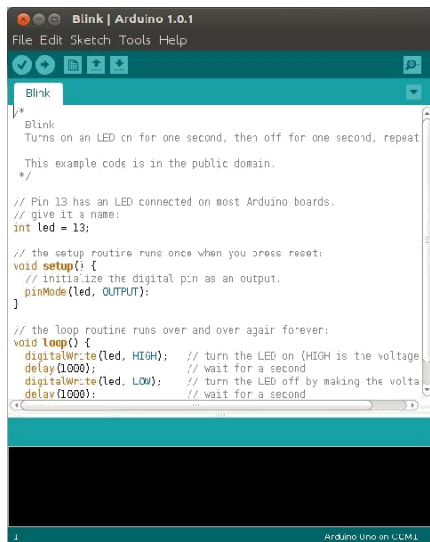


Figura 5. Entorno grafico que ofrece el IDE de Arduino

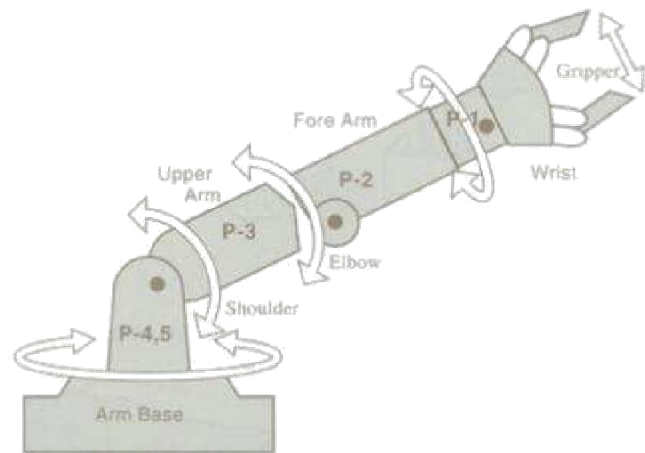


Figura 6. Hardware del brazo robotico, modelo OWI-007 ilustrando los posibles movimientos que puede realizar.

Se puede manipular mediante sus 5 pines de entrada cada uno correspondientes a sus puntos de movilidad los cuales son: la base, hombro, codo, muñeca y mano o pinza, cada uno de estos puntos cuenta con un motor el cual permite moverse en dos sentidos. La interfaz de comunicación con el robot es un conector con 8 pines los cuales tienen la siguiente configuración:

1. Pin 1: Tierra.
2. Pin 2: Base, permite moverse al sentido de las manecillas del reloj y en contra (360 grados).
3. Pin 3: Hombro, permite la movilidad hacia arriba y hacia abajo (con libertad de 120 grados).
4. Pin 4: Codo, libertad de movimiento de 135 grados.
5. Pin 6: Muñeca, permite una rotacion de la muñeca de hasta 340 grados.
6. Pin 7: Pinza, abre y cierra la pinza del brazo robotico.
7. Pin 8: Voltaje.

En la Fig. 6 se observa como es el hardware del brazo a controlar.

V. IMPLEMENTACIÓN DEL PROTOTIPO

Antes de empezar a conectar cables entre los dispositivos, o empezar a desarrollar el programa que permitirá controlar el robot o dispositivo que se quiera utilizar, es necesario identificar como funciona internamente el dispositivo y de que manera convendría utilizar Arduino para obtener un mejor desempeño, como por ejemplo si se utilizaran salidas analógicas o digitales.

Una vez familiarizado con el desarrollo para Arduino y definido como se llevara a cabo el manejo del dispositivo se procede a programar las secuencias predefinidas así como la funcionalidad para controlar el robot por medio del teclado, en la Fig. 7 se puede ver el circuito basico de la conexión de arduino al puente H y a los distintos motores que utiliza el robot para moverse.

Debido que al desarrollar una aplicación que permita realizar movimientos o controlar algún dispositivo por medio

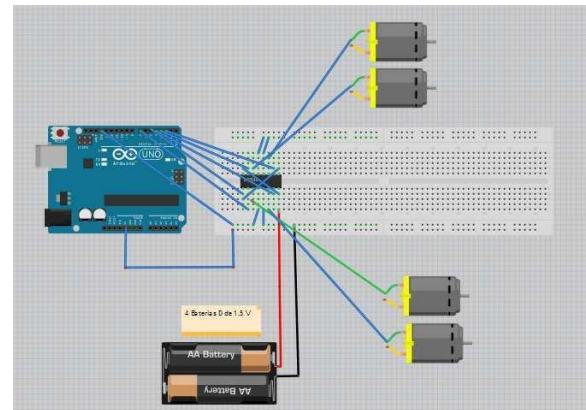


Figura 7. Circuito basico de la conexión entre arduino y los motores a controlar del brazo robotico

del teclado nos permite tener conocimiento y familiarizarnos con que tipo de señales se deben enviar a tramos de Arduino para que realice cierta función, se puede implementar la utilización de un modulo de Bluetooth para Arduino el cual nos permitiría enviar las señales de forma inalámbrica y evitaría las problemáticas que se podrían presentar al tener que estar conectado siempre por medio de un cable a la placa para enviar las señales de las acciones del robot.

Además haciendo uso de las tecnologías emergentes se puede llevar a cabo la implementación de tecnologías de detección de movimiento y procesamiento de imágenes como Kinect que por medio de processing nos permitiría controlar el brazo robótico no solo de forma inalámbrica con la ayuda de Bluetooth si no que por medio de simples señas con las manos que serian procesadas e interpretadas por la computadora para enviar la señal correspondiente a la placa y realizar el movimiento específico sin necesidad de utilizar el teclado.

VI. CONCLUSIONES Y TRABAJO FUTURO

El desarrollo de proyectos que se dediquen a la creación de sistemas electrónicos o bien modificar el funcionamiento de estos, utilizados en el entorno en que se desarrolla el estudiante

puede tener un gran impacto en beneficio no solo de estos mismos, y por ende de los alumnos, sino también en beneficio de la tecnología donde se puede crear desde cero cualquier sistema mediante el cual se ayude a las personas de un área determinada, realizando tareas de forma más sencilla de forma que la interacción con los usuarios sea más adaptable y menos invasiva.

Con el uso de esta tecnología (Arduino), se construyó un prototipo de la aplicación propuesta, en el artículo se presentaron detalles de la implementación.

Como trabajo futuro, se pretende integrar a la aplicación propuesta el manejo a través de bluetooth utilizando una aplicación desarrollada en el sistema operativo Android, por lo que al integrarlo con la aplicación propuesta en el presente artículo, permitirá llevar un mejor manejo desde cualquier dispositivo móvil que tenga el OS de Android, además con la ayuda de la tecnología kinect y processing se pretende reproducir los movimientos que hace una persona con su brazo haciendo que el robot se mueva de forma similar a este.

REFERENCIAS

- [1] R. E. Herrador, "Guía de usuario de arduino," 11 2009.
- [2] E. C. Datasheet, "Funcionamiento de circuitos integrados y microcontroladores. alldatasheet.com <http://www.st.com>."
- [3] J. A. P. W. C. M. R. W. D. d. L. R. A. Lina Antoláñez, (Project Leader), "Wiring challenges,"
- [4] <http://www.arduino.cc/>, "Página oficial de la plataforma libre arduino."
- [5] B. F. Casey Reas, Processing: programming for the media arts. Springer-Verlag, 2006.
- [6] Robotic arm trainer model: OWI-007. Division of Omnitron Electronic, OWIROBOTS.