IMPLEMENTACIÓN DE UN CONTROLADOR DE TORRETA DE HERRAMIENTAS PARA UN TORNO CNC EMCO PC TURN 120

Alan Roberto Martinez Vazquez, David Elias Martínez Navarro, Atali del Carmen Liahud Salas, Iván Josué Juárez Díaz.

Maestría en Educación, Maestría en Ingeniería Industrial y Medio Ambiente, Maestría en Desarrollo Educativo, Ingeniería Ambiental.

Centro de Bachillerato Tecnológico Industrial y de Servicios No. 78.

roberto.mava96@gmail.com
david.martinez@itspozarica.edu.mx
atalita86@gmail.com
ivanjjd@hotmail.com

Referencia de este artículo [1].

RESUMEN

En el presente artículo se describe el proceso de implementación de un dispositivo para controlar el giro y posicion de una torreta portaherramientas basado en la plataforma Arduino para el funcionamiento de un equipo de Torno CNC (Control Numérico Computarizado) marca EMCO modelo PC TURN 120, dicho controlador tiene la función de mover la torreta portaherramientas de 8 posiciones para los diferentes tipos de mecanizado del equipo, a través de código G y además de detectar cada una de dichas posiciones con ayuda de un encoder rotativo. Este proyecto se implementó en el Centro de Bachillerato Tecnologico Industrial y de Servicios No. 78 de la ciudad de Poza Rica de Hidalgo, Veracruz, donde se encuentran estas máquinas para el uso de estudiantes de la especialidad de mecánica industrial mismos que no tenían la capacidad de llevar a cabo sus prácticas debido a que los equipos se encontraban fuera de servicio ocasionado por la falla de la torreta antes mencionada.

Palabras clave: Torno, Arduino, CNC, encoder rotativo, Código G.

MSC 2020: 35R11

1. Introducción

Dentro del CBTIS No. 78 de la ciudad de Poza Rica de Hidalgo, Veracruz; se cuenta con dos tornos de control numérico marca EMCO PC TURN 120 que son utilizados para realizar prácticas e impartir materias de la especialidad de mecánica industrial. Dichos equipos presentan fallas en la torreta portaherramientas producto de la falta de mantenimiento y recursos económicos para su reparación, el daño presentado se debe a que la tarjeta controladora de la torreta se encontraba quemada en ambos equipos.



Figura 1. Torno CNC EMCO PC TURN 120

En el presente documento se presenta una solución económica y funcional para la reparación de los tornos, implementando un dispositivo de control basada en la plataforma Arduino, que mediante un puente H energiza un motor que da movimiento a la torreta y al mismo tiempo es retroalimentada por un encoder rotativo que le indica la posición en la que se encuentra cada herramienta en un momento preciso.

2. Funcionamiento

Actualmente la máquina trabaja con una tarjeta principal que funciona bajo el software Mach3, de acuerdo con el fabricante, cuenta con 4 salidas que pueden ser configuradas por el usuario. Para el uso de la torreta portaherramientas se habilitó una de estas salidas para enviar una señal digital intermitente (S0) que posteriormente será interpretada por un controlador Arduino y así poder girar la torreta en la posición requerida.

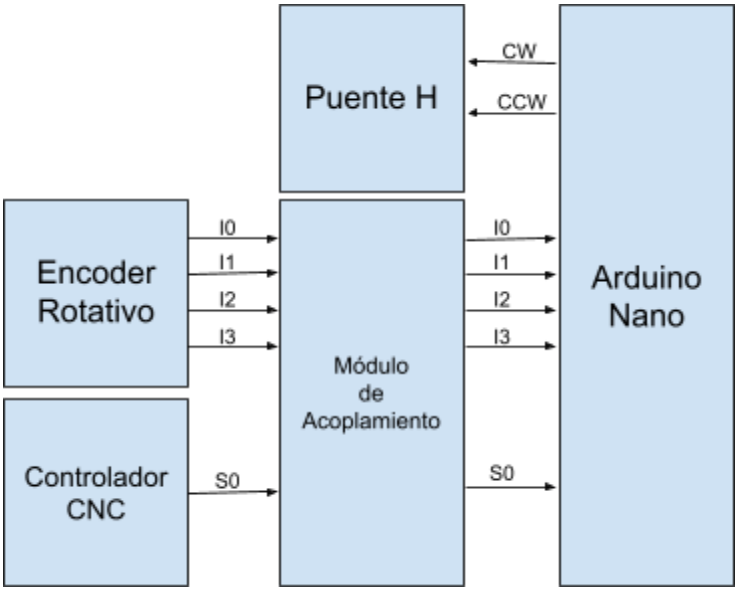


Figura 2. Diagrama de bloques de las señales y dispositivos.

En la figura 2 se observa un módulo de acoplamiento, este módulo se colocó con la intención de proteger el controlador arduino ya que la capacidad en sus entradas digitales es de 5VDC, mientras que las señales del controlador CNC como del encoder rotativo funcionan a 24VDC.

Dicho encoder cuenta con 4 señales que cambian en función de la posición de la torreta, a continuación se presenta una tabla en la que se muestran dichas posiciones así como el estado de cada señal del encoder en ese instante.

Posición de la herramienta		1	2	3	4	5	6	7	8
Encoder Rotativo	I0	1	1	1	1	1	0	0	0
	I1	1	1	1	0	0	0	1	1
	I2	1	0	0	0	1	1	1	1
	I3	0	0	1	1	1	1	1	0

Tabla 1. Posiciones y estados del encoder.

Estas señales son enviadas al módulo de acoplamiento que sirve para aislar los 24VDC que salen del encoder y convertirlas en señales digitales de 5VDC que pueda recibir la tarjeta Arduino. De igual manera la señal del controlador CNC es también enviada al módulo de acoplamiento, esta señal envía pulsos intermitentes de acuerdo a la posición que se requiera, es decir, si el software requiere que la torreta cambie a la posición número 3, se enviaran 3 pulsos de 50ms.

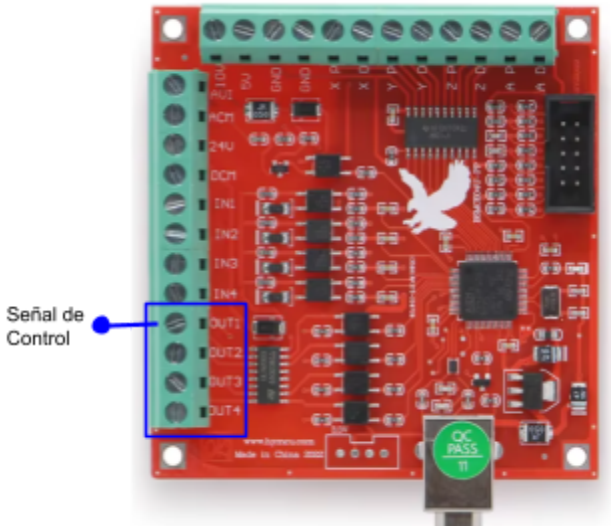


Figura 3. Tarjeta principal del torno CNC.

Una vez que la tarjeta CNC envía la señal de control a la tarjeta Arduino, este interpreta el número de pulsos que recibe, determina la posición que se requiere y envía un pulso al puente H quien arrancará el motor de la torreta haciéndola girar. Cuando el encoder rotativo alcanza la posición deseada, envía el número de 4 bits que se muestra en la tabla 1, haciendo que arduino intérprete que se ha alcanzado la posición indicada y así detener el motor. Posteriormente el controlador arduino volverá a mandar un pulso

pero esta vez para hacer girar la torreta en sentido contrario, ya que es necesario que este elemento retroceda para que un pasador de forma mecánica se active y fije la posición de manera precisa.

3. Diseño del controlador

3.1. Hardware

Los dispositivos fueron seleccionados en base a algunas especificaciones de la máquina cnc, para mover el motor de la torreta el cual consume una corriente de 1.2A a 24V de corriente continua se utilizó un puente H modelo L298N que tiene la capacidad de soportar hasta 2A y un voltaje de 5 a 35VDC.
Como módulo de acoplamiento se empleó una tarjeta DST-1RBP-N que cuenta con 8 canales optoacoplados que pueden recibir señales de 24VDC y convertirla en una salida de 5VDC.
Además se tomó la decisión de utilizar la plataforma Arduino Nano como controlador debido a que cuenta con los puertos suficientes para el proyecto, sin mencionar su bajo costo y fácil acceso.

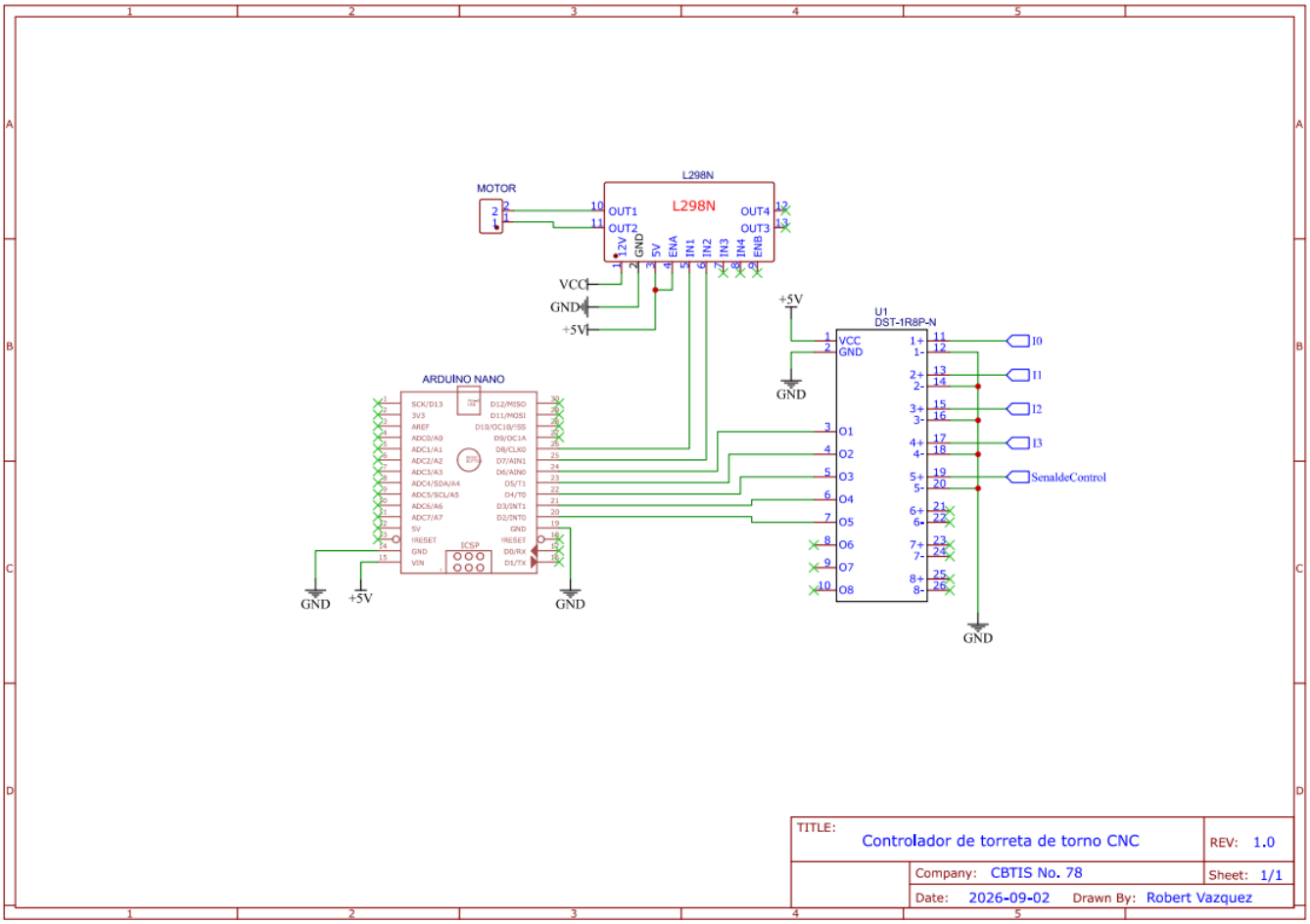


Figura 4. Esquemático electrónico del controlador.

3.2. Software

El código del controlador fue estructurado en el mismo IDE de Arduino, iniciando por declarar las variables necesarias para el proyecto las cuales fueron las 4 señales del encoder (I0, I1, I2, I3), la señal de control que viene de la tarjeta de la máquina CNC (Señal de Control) y las 2 salidas que controlan la dirección del motor de la torreta teniendo un total de 7 puertos, 6 entradas y 1 salida.

```
const byte pinPulsador = 2;  // Entrada de pulsos
const byte pinMotor = 7;     // Salida motor (lógica inversa)
const byte pininversor = 8;
const byte s1 = 3;
const byte s2 = 4;
const byte s3 = 5;
const byte s4 = 6;
```

Figura 5. Declaración de variables.

Parte importante del código es una función que lee la posición de la torreta, en la cual se definieron los lugares que ocupa cada herramienta de la misma con las respectivas señales del encoder como se ejemplifica en la tabla 1 y retornando un valor numérico por cada una de dichas posiciones.

```
// Función que retorna la posición actual según sensores s1..s4 (1-8)
int leerPosicion() {
  int v1 = digitalRead(s1);
  int v2 = digitalRead(s2);
  int v3 = digitalRead(s3);
  int v4 = digitalRead(s4);

  // Definir posición según combinación de sensores
  if      (v1==1 && v2==1 && v3==1 && v4==0) return 1;
  else if (v1==1 && v2==1 && v3==0 && v4==0) return 2;
  else if (v1==1 && v2==1 && v3==0 && v4==1) return 3;
  else if (v1==1 && v2==0 && v3==0 && v4==1) return 4;
  else if (v1==1 && v2==0 && v3==1 && v4==1) return 5;
  else if (v1==0 && v2==0 && v3==1 && v4==1) return 6;
  else if (v1==0 && v2==1 && v3==1 && v4==1) return 7;
  else if (v1==0 && v2==1 && v3==1 && v4==0) return 8;

  return 1; // Siempre retorna una posición válida
}
```

Figura 6. Función para leer la posición del encoder.

A continuación se muestra parte de la lógica de funcionamiento en donde Arduino envía una señal para que el motor arranque y empieza a buscar su posición objetivo, una vez que la posición es alcanzada se vuelve a enviar una señal para que el motor se detenga.

```
void loop() {
  int posActual = leerPosicion();  // Leer posición actual

  // Activar motor solo si ya pasaron X ms desde el último pulso
  if (!motorEncendido && contador > 0 && (millis() - lastPulseTime >= tiempoEsperaPulsos)) {
    if (posActual != posicionObjetivo) {
      digitalWrite(pinMotor, HIGH);   // Encender motor
      digitalWrite(pininversor, LOW);
      motorEncendido = true;
      Serial.print("Motor encendido, objetivo posición: ");
      Serial.println(posicionObjetivo);
    } else {
      // Ya está en la posición, reiniciar contador
      contador = 0;
      Serial.println("Ya en posición objetivo, motor no arranca");
    }
  }
```

Figura 7. Lógica de encendido y búsqueda de la posición de la torreta.

4. Conclusiones

Con el empleo de este controlador se logró hacer funcionar de nuevo los tornos CNC que se encontraban fuera de servicio, sin ninguna falla durante las pruebas realizadas. La implementación de este proyecto fue de suma importancia para el plantel donde se llevó a cabo, debido a que las máquinas de control numérico que ahí se encuentran son utilizadas por alumnos de nivel medio superior para llevar a cabo prácticas en las cuales desarrollan habilidades propias de la especialidad de mecánica industrial. Además de promover el espíritu científico en los jóvenes que se interesan en áreas relacionadas con la ingeniería despertando el interés y mejorando el aprendizaje de los mismos.

Referencias

BIGWING. (2011, Junio 01). *8 position lathe toolchanger*. ArtSoft. https://www.machsupport.com/forum/index.php?topic=18606.0

EMCO. (1996). *Machine description EMCO PC TURN 120*. Hallein, Austria.

STMicroelectronics. (2000, Febrero 01). *L298N Datasheet (PDF)*. ALLDATASHEET.ES. https://www.alldatasheet.es/datasheet-pdf/pdf/22440/STMICROELECTRONICS/L298N.html