

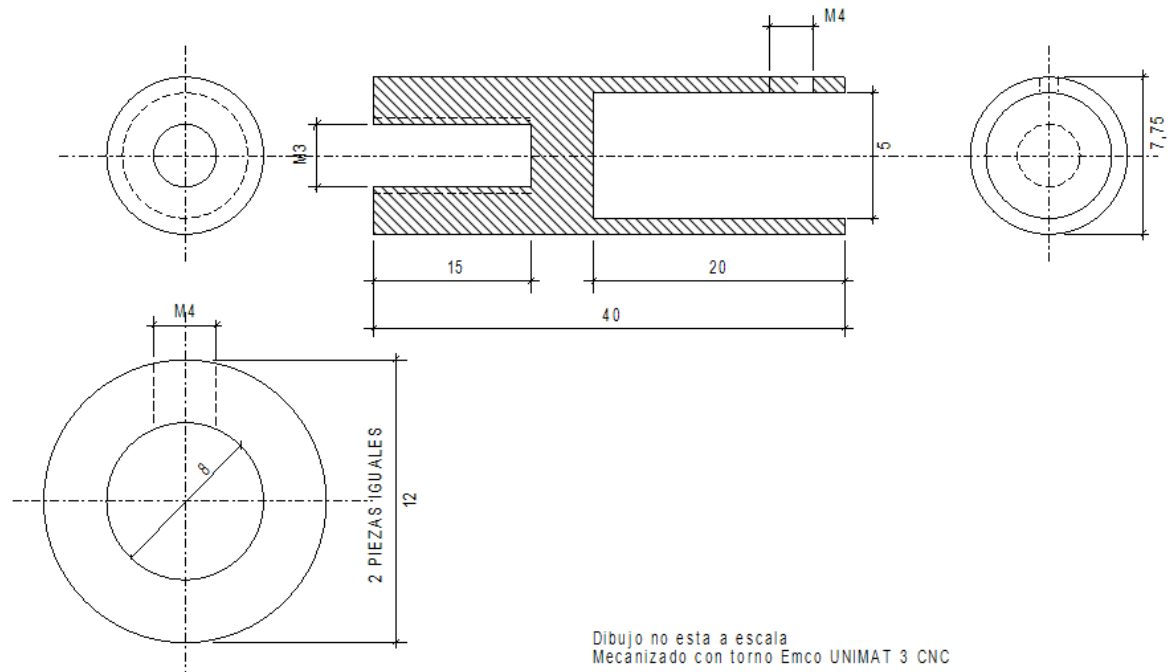
## MOTORIZACION DE UN PUENTE GIRATORIA PECO H0, CON UN MOTOR PASO A PASO NEMA 17 DE DOBLE EJE PROCEDENTE DE UNA IMPRESORA

### a) SUSTITUCION DEL EJE DE PLASTICO DEL PUENTE POR UNO DE BRONCE O LATON

Se procedió a cortar el eje de plástico a ras de la base del puente

Se rellena el agujero con resina epoxi e introducimos hasta el fondo un trozo de varilla roscada de M 3, vigilando que quede bien perpendicular.

Se construye en el torno el nuevo eje que se adapte al eje del motor Nema y las fijaciones correspondientes según el croquis siguiente.



Montamos y verificamos suavidad en el giro.



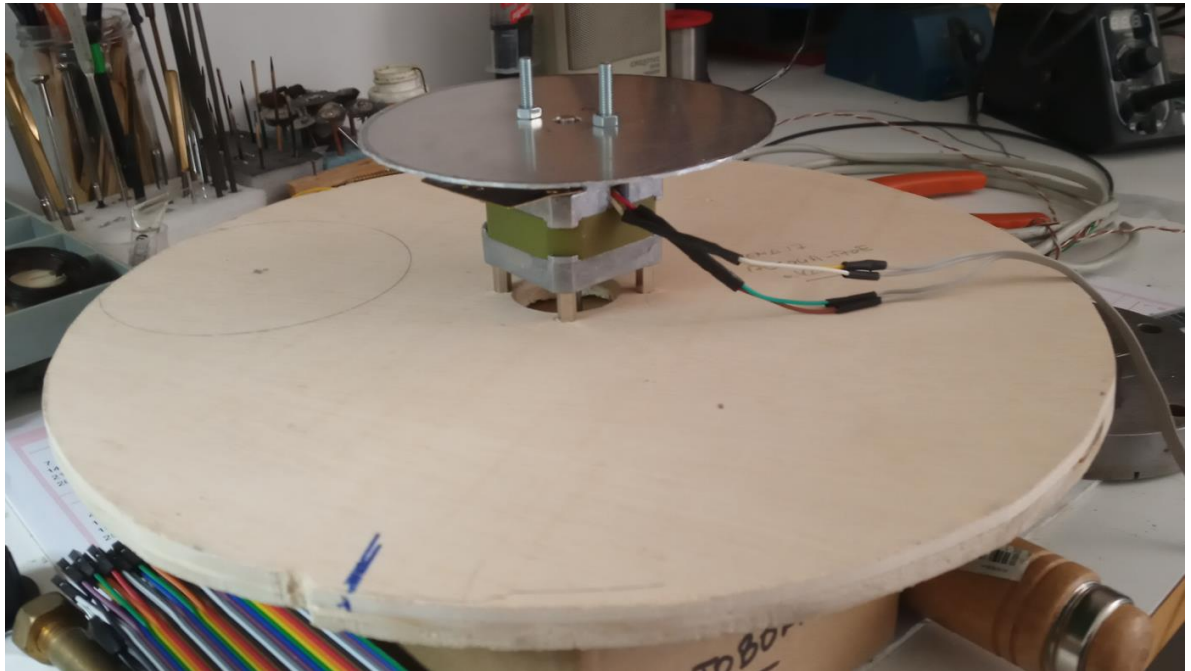
## b) FIJACION MOTOR Y SENSORES

Cortamos un disco de madera, yo tenía contrachapado de 9 mm, para que encaje en el interior de la parte trasera del foso del puente, taladramos el centro a un diámetro de 40 mm. y fijamos, bien centrado, el motor con unas torretas de 20 mm. entre el disco y el motor.

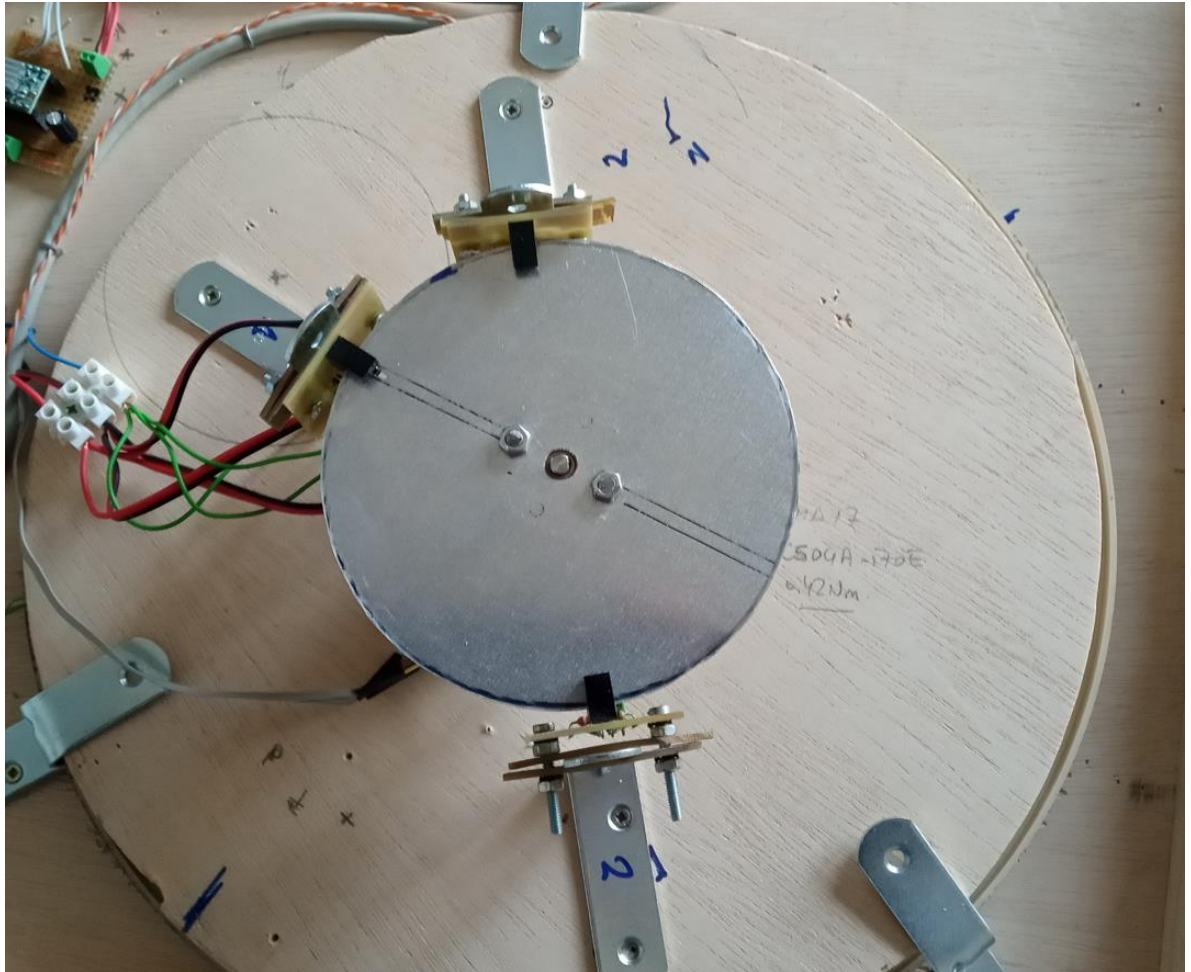
Corto de una chapa de aluminio de 1 mm de grosor, un disco de 40 mm. de diámetro centrado fijo una pieza de Mecano con el centro retaladrado a 5 mm. para el eje del motor. Este disco girará dentro de la ranura del fotointerruptor TCST1103

Como que mi puente giratorio solo tiene una entrada y una salida, es decir tres posiciones, en lugar de efectuar las ranuras para que la fotocélula detectara el momento de parar, he preferido colocar tres detectores, uno por cada posición de parada más fáciles de ajustar y hacer una sola ranura.

## MOTOR Y DISCO ALUMINIO



## LOS 3 SENSORES MONTADOS



### c) ELECTRÓNICA

#### 1) MATERIAL EMPLEADO

Motor Paso a Paso bipolar NEMA 17 doble eje

Arduino Nano

Stepper Driver A4988

Fotointerruptor de ranura TCST1103

2 Condensador electrolítico 220 mF 64V

Puente rectificador 5 A.

Resistencias 10K 1/4W

Resistencias 220R 1/4W

Tiras de pines hembra

Bornas terminales a tornillo paso 2,54

Placa C.I. para prototipos

Placa C.I. diseño para Fotointerruptor



```

#include <AccelStepper.h>
#include <MultiStepper.h>

// Define conexiones del motor y tipo de interfazestepper motor.
#define dirPin 3
#define stepPin 4
#define motorInterfaceType 1

// Define ENABLE, PULSADOR y CELULA
const int PULSADOR = 6;
const int ENABLE = 8;
const int CELULA = 9; // Pulsador arranque y celulas de paro
int val;
int val1;

// Crear instancia para tipo motore con AccelStepper
AccelStepper stepper = AccelStepper(motorInterfaceType, stepPin, dirPin);

void setup() {

  // Definir pines arduino
  pinMode (ENABLE, OUTPUT);
  pinMode (PULSADOR, INPUT);
  pinMode (CELULA, INPUT_PULLUP);
  stepper.setMaxSpeed(1000);

}
void loop() {
  val = digitalRead(PULSADOR);
  val1 = digitalRead (CELULA);

  if ((val == HIGH) && (val1 == LOW)) {
    digitalWrite (ENABLE, LOW);
    stepper.setSpeed(600); //Velocidad de rotacion
    stepper.runSpeed();}

  else {
    AccelStepper (stop); // Paro avance del motor con bobinas activadas para
      // bloquear la posición. Este tiempo se ajusta para poder
      // efectuar la maniobra de colocar la locomotora en su sitio
      // sin que el puente se mueva.
    delay (1000); // Tiempo antes de desconectar bobinas
    digitalWrite(ENABLE, HIGH); // desconecta las bobinas del motor

  }
}

```