

CLASIFICADOR DE CAJAS



Realizado por:
Angel Lopez Gonzalez
Alvaro Velasco Ortiz-Villajos



**Universidad
Europea de Madrid**

LAUREATE INTERNATIONAL UNIVERSITIES

AUTOMATISMOS Y CONTROL

PROFESOR: RAMIRO DIEZ ZAERA



LAUREATE INTERNATIONAL UNIVERSITIES

Universidad Europea de Madrid

LAUREATE INTERNATIONAL UNIVERSITIES

PROFESOR: RAMIRO DIEZ ZAERA

AUTOMATISMOS Y CONTROL

LAUREATE INTERNATIONAL UNIVERSITIES

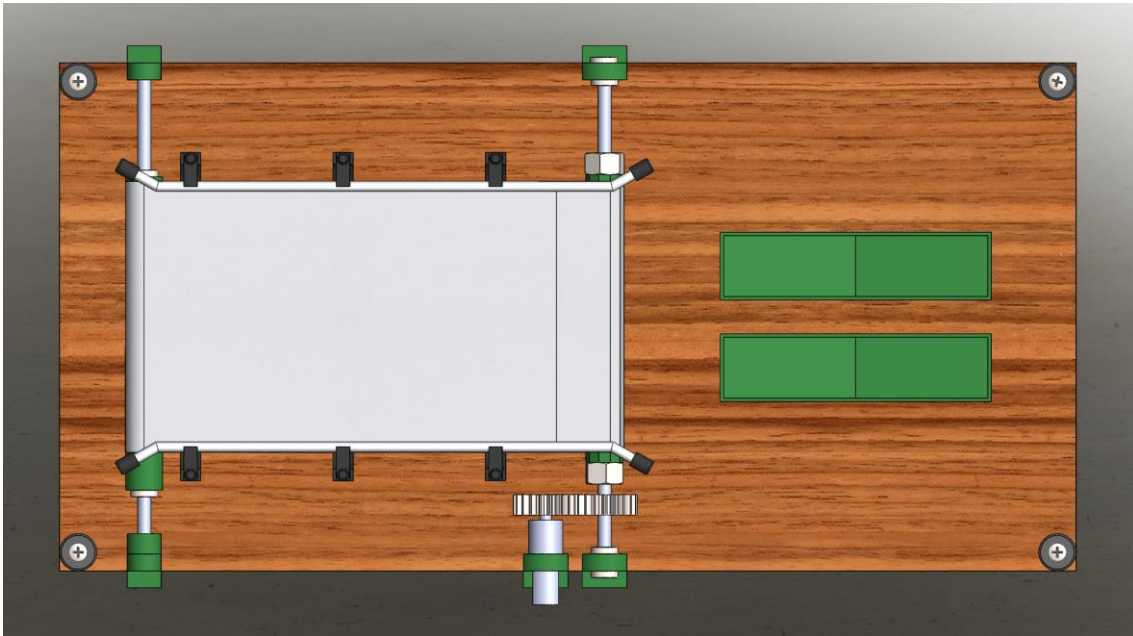
ÍNDICE

ÍNDICE.....	2
INTRODUCCIÓN	3
CÁLCULOS INICIALES	4
ELECCIÓN DE SENSORES.....	4
SELECCIÓN DE SENSORES.....	5
PROGRAMACIÓN	6
PROGRAMACIÓN ARDUINO C	7
PROGRAMACIÓN KOP TIA PORTAL	12
CONCLUSIONES	14
ELEMENTOS UTILIZADOS EN EL PROTOTIPO Y DATASHEETS	15
ELEMENTOS EQUIVALENTES Y DATASHEETS	15



INTRODUCCIÓN

Se crea un escenario industrial en el que se dispone de un transportador por el cual entran paquetes de dos tamaños, estos paquetes pueden entrar en una posición aleatoria y además conviven 2 tamaños. Las cajas se deben de recoger y clasificar para destinarlos en su cubeta correspondiente tal y como se muestra en la figura.



Se tienen en cuenta las dimensiones del transportador, así como su velocidad, para confeccionar un sistema automatizado sobre la parte existente. También se tiene en cuenta la ubicación de las cubetas para la clasificación de las cajas.

Los parámetros a partir de los que se realiza la automatización son:

- Ancho de transporte 160mm.
- Longitud de trayecto 300mm.
- Velocidad de giro 25rpm.
- Diámetro de la zona de tracción 18mm.
- Dimensiones de cajas:
 - 9mm³.
 - 16mm³.
- Ubicación de las cubetas:
 - 50mm.
 - 110mm.



CÁLCULOS INICIALES

La velocidad de la cinta se halla a través de la velocidad del motor de tracción, el diámetro del elemento tractor y la longitud del transportador:

$$2 \cdot \pi \cdot r = 2 \cdot \pi \cdot 9mm = 56,54mm$$

$$\frac{56,54mm \cdot 25rpm}{60s} = 23,56mm \text{ por Segundo}$$

$$\frac{300mm}{23,56mm / seg} = 12,73seg$$

Obtenemos una velocidad de trayecto total de 12 segundos aprox. En este trayecto se debe implementar un sistema que sea capaz de detectar el tamaño y posición de las cajas con una antelación suficiente como para que el actuador disponga de tiempo suficiente para realizar el trabajo de transporte.

ELECCIÓN DE SENSORES

Para la localización de las cajas, se barajan distintos sensores:

- Ultrasonidos: el emisor emite una onda ultrasónica que, si encuentra un objeto, la onda rebota y es capturada por el módulo receptor. De esta manera midiendo el tiempo que la onda tarda en ir y reflejarse el sensor calcula la distancia a la que se encuentra el objeto. Con este tipo de sensores, podríamos realizar una detección de presencia del objeto, pero no tendríamos información fiable de la posición y tamaño del objeto. Habría que utilizar una cantidad elevada de sensores para mejorar la detección.
- Capacitivo: sucede lo mismo que en el caso anterior. Si ubicamos una barrera de sensores, podríamos obtener la posición con baja precisión y no obtendríamos una información fiable del tamaño de caja, por lo que no es una aplicación adecuada.
- Cámara VGA: con una cámara VGA o de resolución superior, se podría realizar una localización de los envases y su tamaño de forma muy precisa, contabilizando píxeles y su ubicación en la matriz. El elevado grado de dificultad en la programación y su coste dejan esta opción como último recurso.
- Óptico: Los sensores ópticos de barrera láser crean barreras luminosas que pueden dar la información que se precisa para esta aplicación. Con una única fotocélula podemos detectar el tamaño de los paquetes midiendo el desplazamiento del transportador entre flanco descendente y flanco ascendente. Ubicando un segundo sensor en ángulo con el



primero, podremos obtener la posición en el eje x dependiendo del tiempo transcurrido entre la primera fotocélula y la segunda.

Para el transporte de cajas, se barajan distintos actuadores:

- Brazo servo-controlado: una vez localizada la caja se puede utilizar un brazo servo-controlado para recoger el objeto y depositarlo en su correspondiente cubeta. Es una buena opción, aunque requiere de 4 servomotores y altos conocimientos de programación, por lo que se mantienen como posible recurso.
- Switch servo-mecánico: se puede utilizar un switch mecánico acoplado a un servomotor, a modo de palanca, para el reenvío final del paquete, pero este sistema puede dañar el producto, además dependiendo de la posición puede no ser efectivo y provocar atascos.
- Servo-Tolva: se idea un sistema de posicionamiento sobre un carro que obtiene el desplazamiento a través de un motor paso a paso. Sobre el carro se dispone un servomotor de $\pm 90^\circ$ de libertad, acoplado a la tolva, para la transferencia desde la cinta hasta las cubetas correspondientes. Este sistema permite utilizar como referencia los pasos del motor paso a paso, aunque requiere de un sensor final de carrera como referencia de punto cero.

SELECCIÓN DE SENSORES

Para el clasificador de cajas DT-BOX Controller, se decide utilizar el sistema de fotocélulas en ángulo y el actuador Servo-Tolva por ser los sensores y actuadores más económicos.

La construcción de las barreras fotoeléctricas laser, se componen de un emisor de diodo laser de 650nm y un receptor LDR junto con una resistencia de 2Kohm como divisor de tensión. La señal facilitada por este sistema es analógica de 0-5V que, a través del convertor analógico-digital de 10 bit de la placa utilizada, entrega valores 0-1023 que sirven para regular la sensibilidad de la barrera.

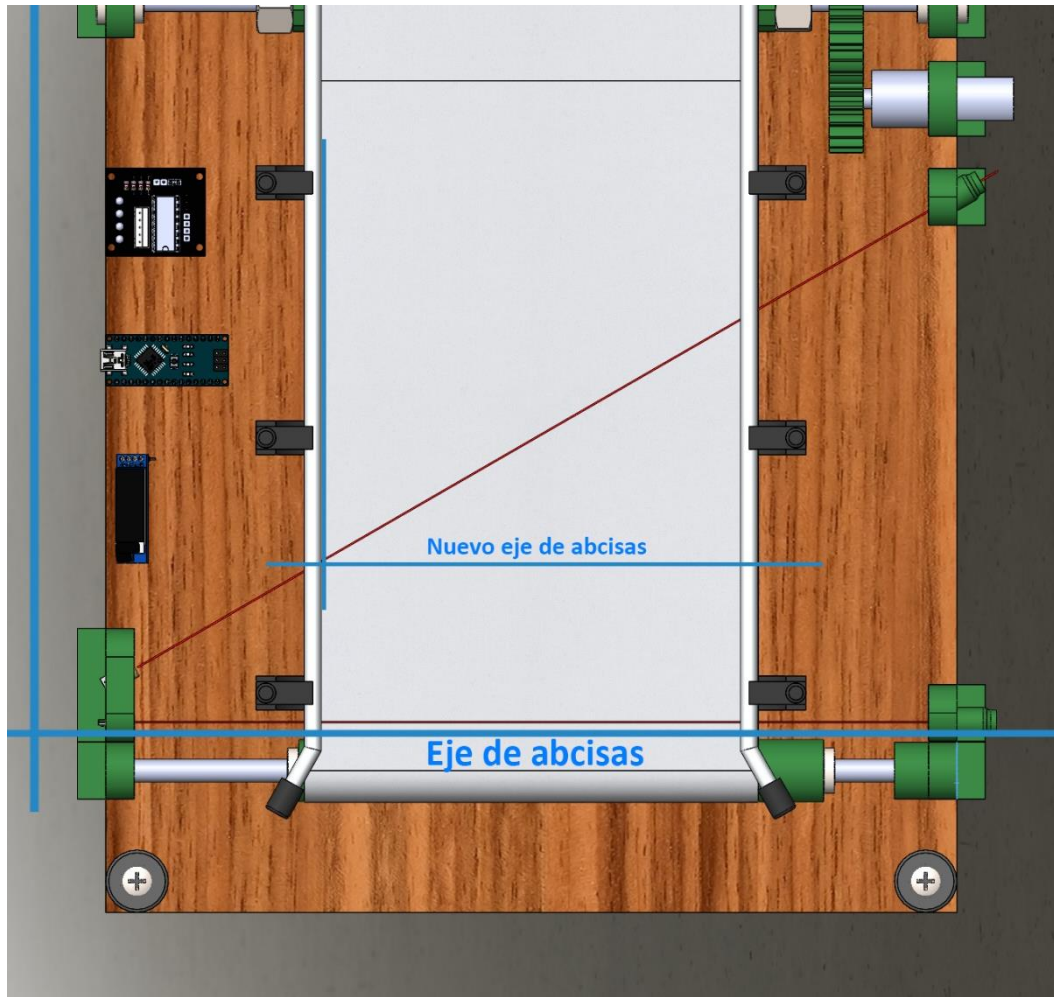
Al ser un prototipo, se utilizan elementos de muy bajo coste, pero se denota su baja calidad y la falta de Datasheets o en caso de haberla, de poca información. Por ello se realiza un anexo al final de este documento indicando los elementos equivalentes de nivel industrial y sus correspondientes fichas técnicas.

El sistema propuesto con los sensores y actuadores seleccionados tiene la capacidad de localizar el objeto en cualquier punto de la entrada de la cinta y de detectar el tamaño de los dos tipos de objetos en un periodo inferior a 12 segundos con tiempo suficiente como para que el actuador haga su función. La velocidad máxima del stepper motor es de 15 rpm lo que nos da una velocidad de giro de 4 segundos por vuelta. Teniendo en cuenta este valor se configura un sistema de poleas en el que la polea tractora tenga un perímetro superior a la distancia a recorrer, para que este sea recorrido en un periodo inferior a 4 segundos. Esto deja 8 segundos para realizar la detección y clasificación, que se ve reducido a 6 segundos ya que hay que dejar una zona de entrada de cajas hasta que se comienza con la detección.



PROGRAMACIÓN

Para la programación se realiza la linealización del sistema. Al disponer de la barrera óptica montada en el borde del bastidor, se configuran las barreras con un ángulo de 30 grados para tener cierta precisión y margen suficiente para el movimiento de la servo-tolva.



Para la programación del algoritmo, utilizamos la ecuación de la recta $F(x) = m \cdot x + n$ en donde n es la corrección de la posición sobre el nuevo eje de abscisas que en este caso son 56mm que dividido por 23,56mm/s corresponde a 2382 milisegundos. Con esto conseguimos que m sea la pendiente de la recta y para el cálculo de la pendiente, consideramos que entre el punto mínimo y el punto máximo va desde 0% hasta 100% para luego aplicar al número de pulsos del trayecto del stepper motor. La distancia máxima desde el nuevo eje de abscisas es de 80mm, pero como hay que tener en cuenta las dimensiones de las cajas, se consideran 70mm que divididos por 23,56mm/s obtenemos un tiempo máximo de 2971 milisegundos. La ecuación del sistema utilizando los cálculos anteriores será:

$$F(x) = \frac{100x}{2971} - 2382$$



Esta función nos dará un valor entre el 0 y el 100% que lo multiplicaremos por el número de pulsos correspondientes al trayecto total de la tolva y el resultado será la cantidad de pulsos a desplazarse para recoger la caja. Después con la referencia fija en las posiciones de las cubetas se llevará la tolva hasta la posición correspondiente y se procederá al giro de la tolva a través del servomotor. Por último, para finalizar el proceso, la tolva volverá hasta su posición inicial marcado por un final de carrera y podrá dar comienzo un nuevo ciclo de detección.

La programación se ha realizado en pseudo C (Arduino) y se ha realizado un código alternativo en KOP TIA PORTAL que se adjunta a continuación.

PROGRAMACIÓN ARDUINO C

```
#include <VarSpeedServo.h>
#include <Time.h>
#include <U8g2lib.h>
#ifdef U8X8_HAVE_HW_SPI
#include<SPI.h>
#endif
#ifdef U8X8_HAVE_HW_I2C
#include<Wire.h>
#endif
U8G2_SSD1306_128X32_UNIVISION_F_HW_I2C u8g2(U8G2_R0);

const unsigned char myBitmap [] U8X8_PROGMEM = {
// Logotipo uem, 128x32px formato XBMP

0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0x3f, 0xfc, 0xff, 0xff, 0xff, 0xfa, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0x2f, 0xf8, 0xff, 0xff, 0x3f, 0xc0, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x0f, 0xfc, 0xff,
0xff, 0x0f, 0xa0, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0x0b, 0xf8, 0xff, 0xff, 0x0b, 0x80, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0x01, 0xf8, 0xff, 0xff, 0x07, 0x00, 0xff,
0xdf, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x01, 0xf8, 0xdf,
0xff, 0x03, 0x00, 0xff, 0x03, 0xf8, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0x00, 0xf8, 0x87, 0xff, 0x01, 0x80, 0x7f, 0x01, 0xe8, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0xfc, 0x0f, 0xff, 0x01, 0x00, 0x7f,
0x00, 0xc0, 0xff, 0xff, 0xe9, 0xff, 0xff, 0xff, 0x00, 0xf8, 0x03,
0xfe, 0x00, 0x80, 0x0f, 0x00, 0x80, 0xff, 0x1f, 0x00, 0xff, 0xff, 0xff,
0xff, 0x03, 0xfc, 0x03, 0xff, 0x00, 0xc0, 0x1f, 0x00, 0x00, 0xfe, 0x3f,
0x00, 0xfc, 0xff, 0xff, 0x0f, 0xf8, 0x01, 0xfe, 0x00, 0xa0, 0x03,
0x00, 0x00, 0xfd, 0x03, 0x00, 0xfc, 0xff, 0xff, 0xff, 0x0f, 0xfc, 0x01,
0xff, 0x00, 0xe0, 0x03, 0xd0, 0x00, 0xf8, 0x07, 0x00, 0xf0, 0xff, 0xff,
0xff, 0x1f, 0xf8, 0x00, 0xfe, 0x00, 0x70, 0x00, 0xfc, 0x03, 0x68, 0x01,
0x00, 0xe0, 0xff, 0xff, 0x0f, 0xf8, 0x00, 0xbe, 0x00, 0xf8, 0x00,
0xfd, 0x07, 0x00, 0x00, 0x00, 0x80, 0xff, 0xff, 0xff, 0x1f, 0x38, 0x00,
0x54, 0x00, 0x16, 0x00, 0xff, 0x1f, 0x00, 0x00, 0x1f, 0x80, 0xff, 0xff,
0xff, 0x1f, 0x30, 0x00, 0x00, 0x00, 0x06, 0xc0, 0xff, 0x3f, 0x00, 0x40,
0x7f, 0x00, 0xfc, 0xff, 0xff, 0x1f, 0x00, 0x00, 0x00, 0x00, 0x00, 0xe0,
0xff, 0x3f, 0x00, 0xa0, 0xff, 0x00, 0xfa, 0xff, 0xff, 0x3f, 0x00, 0x00,
0x00, 0x00, 0x00, 0xe8, 0xff, 0xff, 0x00, 0xe0, 0xff, 0x03, 0xf0, 0xff,
0xff, 0x3f, 0x00, 0x10, 0x00, 0x00, 0x00, 0xf8, 0xff, 0xff, 0x03, 0xfc,
```




```

void setup() {

    u8g2.begin();
    Serial.begin(9600);

    //Declarar pines como salida Stepper
    pinMode(motorPin1, OUTPUT);
    pinMode(motorPin2, OUTPUT);
    pinMode(motorPin3, OUTPUT);
    pinMode(motorPin4, OUTPUT);

    //Declarar pines de entrada LDRs
    pinMode(LDR1, INPUT);
    pinMode(LDR2, INPUT);

    //Declarar pines de entrada Endstop
    pinMode(endstop, INPUT);

    //Pin servo
    Tolva.attach(9);

    //Puesta a cero Tolva
    volcado ();
    Tolva_Actuadores ();

}

void loop() {

    datoldr1 = analogRead(LDR1); //Control LDR1
    if (datoldr1 < 500)
    {
        tiempo1 = millis();
        while (datoldr1 < 500) //Bucle detección flanco UP
        {
            datoldr1 = analogRead(LDR1);
        }
        tiempo2 = millis();
    }

    diferencia1 = tiempo2 - tiempo1; //Diferencia flancos DOWN-UP LDR1
    if (diferencia1 < 400) //Clasificación de cajas
    {
        tipocaja = 0;
    }
    else
    {
        tipocaja = 1;
    }

    datoldr2 = analogRead(LDR2); //Control LDR1
    if (datoldr2 < 500)
    {
        tiempo3 = millis();
        Actuadores = 1; //Habilitacion de actuadores
        if (tipocaja==0)
        {
            u8g2.clearBuffer(); //Limpieza de memoria interna
            u8g2.setFont(u8g2_font_logisoso20_tr); //Fuente texto
            u8g2.drawStr(8,29,"Small Box"); //Texto
            u8g2.sendBuffer(); //Transferencia memoria interna-display
        }
    }
}

```



```

}
else if (tipocaja ==1)
{
u8g2.clearBuffer(); //Limpieza de memoria interna
u8g2.setFont(u8g2_font_logisoso20_tr); //Fuente texto
u8g2.drawStr(8,29,"Big Box"); //Texto
u8g2.sendBuffer(); //Transferencia memoria interna-display
}
}

diferencia2 = tiempo3 - tiempo1; //Diferencia de tiempo LDR1-LDR2
datoprevio = diferencia2 - 2200; //Valor numérico tiempo en zona de distancia mínima

if (datoprevio < 0) datoprevio = 0; //Parche límite mínimo
if (datoprevio > 2971) datoprevio = 2971; //Parche límite máximo stepper
porcentaje = (datoprevio*100)/2971; //Distancia entre ldr2 minimo y ldr2 máximo

posicion = (1100 - (11*porcentaje)); //Pulsos del stepper recorrido total cinta 1100
if (posicion <= 0) posicion = 1; //Parche límite mínimo stepper

//Bucle movimiento stepper
while (counter < posicion && Actuadores == 1)
{
clockwise();
delayMicroseconds(motorSpeed);
counter ++;
esperacaja = 1; //Deshabilitacion stepper y tiempo espera
}
if (esperacaja == 1)
{
stepperlow (); //Tiempo de espera recogida
delay (6000);
esperacaja = 0;
}
if (tipocaja == 0 && Actuadores == 1)
{

retorno = (posicion - 300); //Posición caja pequeña
if (retorno < 0)
{
while (retorno < 0)
{
clockwise ();
delayMicroseconds(motorSpeed);
retorno ++;
}
stepperlow ();
}
else if (retorno > 0)
{
while (retorno > 0)
{
counterclockwise ();
delayMicroseconds(motorSpeed);
retorno --;
}
stepperlow ();
}
}
if (tipocaja == 1 && Actuadores == 1)
{
retorno = (posicion - 800); //Posición caja grande
if (retorno < 0)
{
while (retorno < 0)

```



```

    {
        clockwise ();
        delayMicroseconds(motorSpeed);
        retorno ++;
    }
    stepperlow ();
}
else if (retorno > 0)
{
    while (retorno > 0)
    {
        counterclockwise ();
        delayMicroseconds(motorSpeed);
        retorno --;
    }
    stepperlow ();
}
}

if (Actuadores == 1)
{
    //Procedimiento de volcado y puesta a cero
    volcado();
    Tolva_Actuadores();
    Actuadores = 0;
    counter = 0;
    retorno = 0;
}
}

void Tolva_Actuadores()
{
    //Proceso de puesta a cero Tolva
    while (digitalRead (endstop) == HIGH)
    {
        counterclockwise();
        delayMicroseconds(motorSpeed);

    }
    stepperlow ();
    Actuadores = 0;
    posicion = 0;
    u8g2.clearBuffer();
    u8g2.setBitmapMode(0);
    u8g2.firstPage();
    u8g2.drawXBMP( 0, 0, 128, 32, myBitmap);
    u8g2.nextPage() ;
}

void volcado()
{
    //Proceso de volcado
    Tolva.write(120,255,true);
    delay (1000);
    Tolva.write(60,50,true);
    delay (1000);
}

void clockwise()
{
    stepCounter++;
    if (stepCounter >= numSteps) stepCounter = 0;
    setOutput(stepCounter);
}

void counterclockwise()

```



```

{
  stepCounter--;
  if (stepCounter < 0) stepCounter = numSteps - 1;
  setOutput(stepCounter);
}

void setOutput(int step)
{
  digitalWrite(motorPin1, bitRead(stepsLookup[step], 0));
  digitalWrite(motorPin2, bitRead(stepsLookup[step], 1));
  digitalWrite(motorPin3, bitRead(stepsLookup[step], 2));
  digitalWrite(motorPin4, bitRead(stepsLookup[step], 3));
}

void stepperlow ()
{
  //Proceso de desactivación de las bobinas del stepper para evitar sobrecalentamiento
  digitalWrite(motorPin1, LOW);
  digitalWrite(motorPin2, LOW);
  digitalWrite(motorPin3, LOW);
  digitalWrite(motorPin4, LOW);
}

```

PROGRAMACIÓN KOP TIA PORTAL

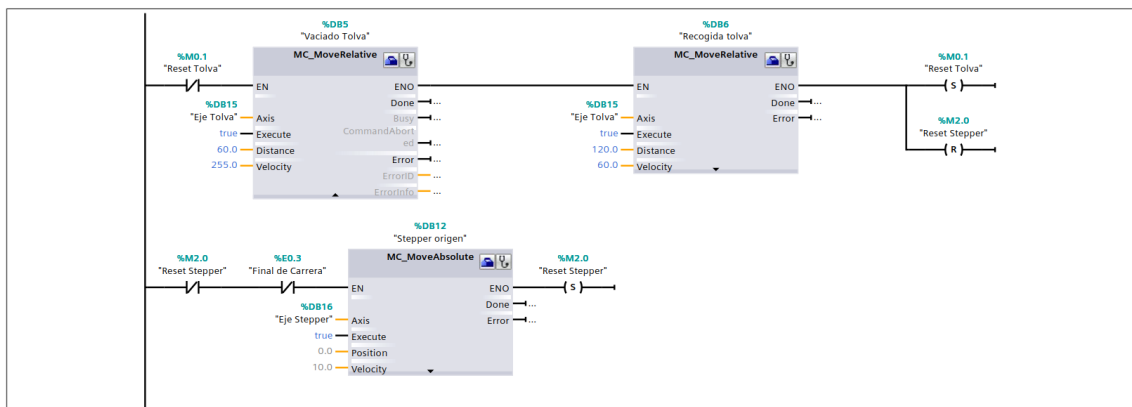
DT-BOX CONTROLLER / PLC_1 [CPU 1211C AC/DC/Rly] / Bloques de programa

Main [OB1]

Main Propiedades							
General							
Nombre	Main	Número	1	Tipo	OB	Idioma	KOP
Numeración	automática						
Información							
Título	"Main Program Sweep (Cycle)"	Autor		Comentario	DT-BOX CONTROLLER	Familia	
Versión	0.1	ID personalizada					

Main			
Nombre	Tipo de datos	Valor predet.	Comentario
▼ Input			
Initial_Call	Bool		Initial call of this OB
Remanence	Bool		=True, if remanent data are available
Temp			
Constant			

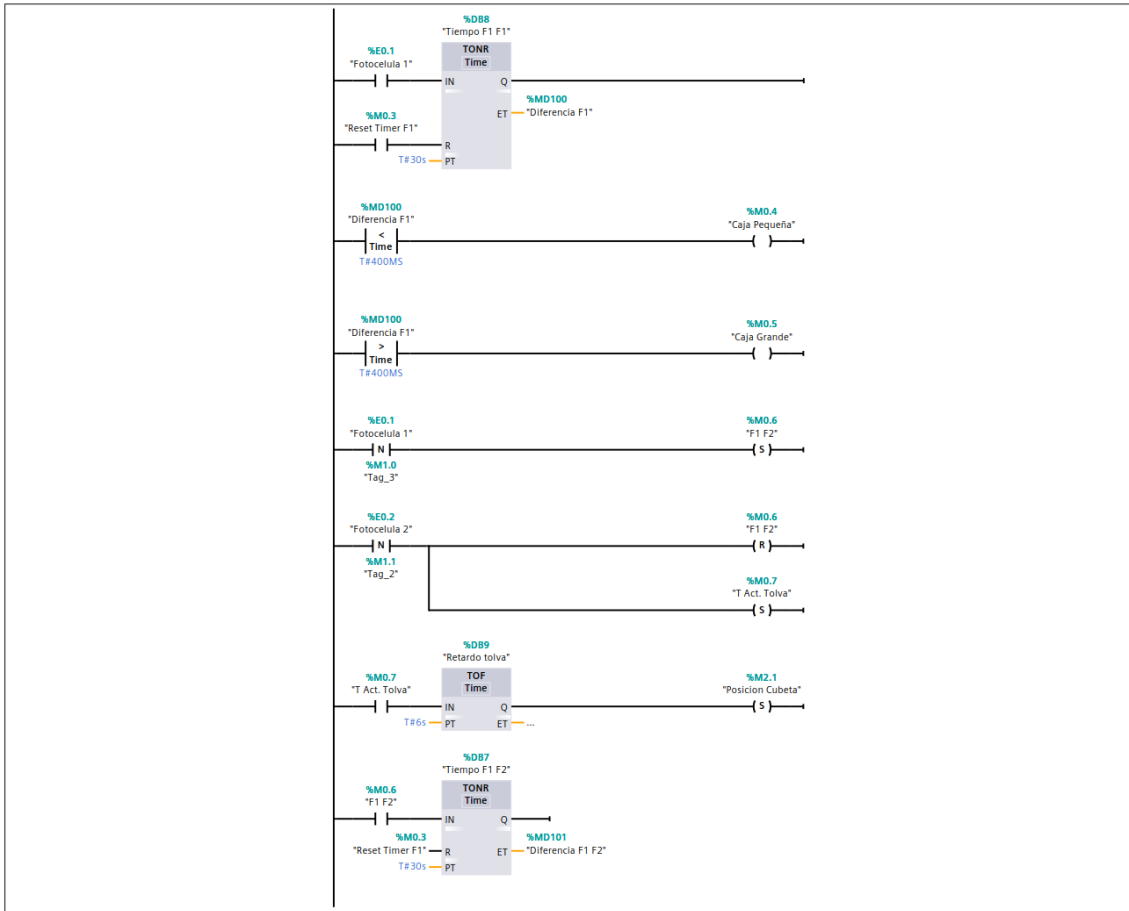
Segmento 1: Inicio



Símbolo	Dirección	Tipo	Comentario
"Final de Carrera"	%E 0.3	Bool	
"Reset Stepper"	%M2.0	Bool	
"Reset Tolva"	%M0.1	Bool	

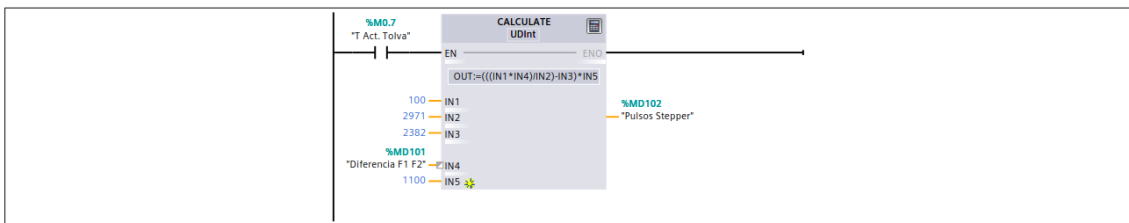


Segmento 2: Deteccion caja, tiempo de medición, movimiento stepper y tolva



Símbolo	Dirección	Tipo	Comentario
"Caja Grande"	%M0.5	Bool	
"Caja Pequeña"	%M0.4	Bool	
"Diferencia F1 F2"	%MD101	Time	
"Diferencia F1"	%MD100	Time	
"F1 F2"	%M0.6	Bool	
"Fotocelula 1"	%E0.1	Bool	
"Fotocelula 2"	%E0.2	Bool	
"Posicion Cubeta"	%M2.1	Bool	
"Reset Timer F1"	%M0.3	Bool	
"T Act. Tolva"	%M0.7	Bool	
"Tag_2"	%M1.1	Bool	
"Tag_3"	%M1.0	Bool	

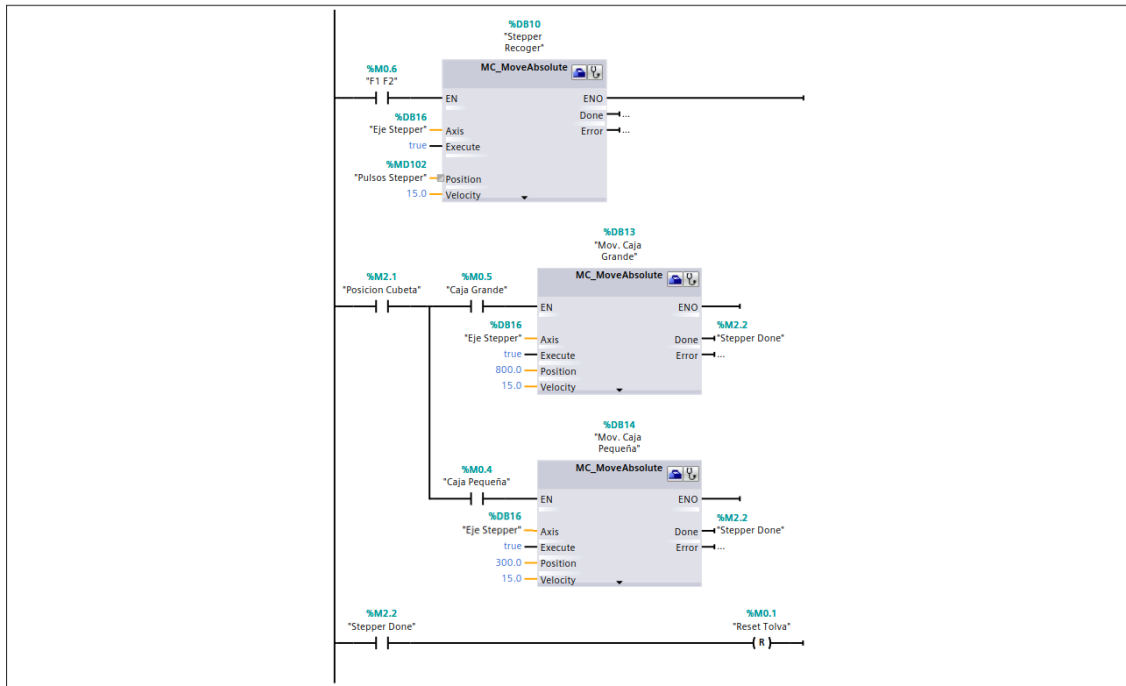
Segmento 3: Calculo Posición Caja



Símbolo	Dirección	Tipo	Comentario
"Diferencia F1 F2"	%MD101	Time	
"Pulsos Stepper"	%MD102	UDInt	
"T Act. Tolva"	%M0.7	Bool	



Segmento 4: Movimiento Stepper



Símbolo	Dirección	Tipo	Comentario
"Caja Grande"	%M0.5	Bool	
"Caja Pequeña"	%M0.4	Bool	
"F1 F2"	%M0.6	Bool	
"Posicion Cubeta"	%M2.1	Bool	
"Pulsos Stepper"	%MD102	UDInt	
"Reset Tolva"	%M0.1	Bool	
"Stepper Done"	%M2.2	Bool	

CONCLUSIONES

Tras comprobar los cálculos previos, se ha realizado una maqueta del sistema y se han introducido los valores calculados, siendo bastante precisos, aunque ha sido necesaria una corrección del parámetro para el cambio de eje de abscisas de 2382 a 2200 para aumentar la precisión con los 2 tipos de cajas. El dispositivo tiene un acierto del 95% y debido a la sencillez del sistema es fácilmente escalable a un sistema de mayores dimensiones. Se considera que se ha dado una buena solución técnica a la gestión de las cajas con la aleatoriedad de su entrada tanto en posición como en ángulo.

Los siguientes elementos del prototipo se catalogan como elementos de desgaste estándar:

- Rodamientos del sistema.
- Engranajes.
- Cinta de transporte.
- Cuerda de tracción Servo-Tolva.
- Emisores de luz laser



ELEMENTOS UTILIZADOS EN EL PROTOTIPO Y DATASHEETS

- Diodo Laser 660nm.
<http://www.mantech.co.za/datasheets/products/913731-R0.pdf>
- LDR.
<https://www.electan.com/datasheets/cebek/CE-C2795.pdf>
- Endstop
https://elecfrakes.com/estore/download/EF03002-mechanical_endstop.pdf
- Mini Servo 9g.
http://www.ee.ic.ac.uk/pcheung/teaching/de1_ee/stores/sg90_datasheet.pdf
- Stepper motor 28byj-48
<http://robocraft.ru/files/datasheet/28BYJ-48.pdf>
- DC Motor 25rpm.
<https://www.pololu.com/product/1589/specs>
- Arduino Nano.
https://www.mouser.com/pdfdocs/Gravitech_Arduino_Nano3_0.pdf
- Pantalla OLED 0,91”
<https://www.vishay.com/docs/37894/oled128o032dlpp3n00000.pdf>
- Fuente de alimentación 12v 30A.
<http://www.wlgled.com/estaticos/media/producto-pdf/8PS120360DCU.pdf>

ELEMENTOS EQUIVALENTES Y DATASHEETS

Recomendados para un sistema de tamaño real con un transportador de 2x4 metros.

- Sustituyendo diodo laser y LDR, Fococélula Sick reflex WL12-3P2431.
https://cdn.sick.com/media/pdf/0/10/310/dataSheet_WL12-3P2431_1041436_es.pdf
- Sustituyendo Endstop, sensor inductivo Pepper Fluch NBB0 6-3M22-E2.
https://files.pepperl-fuchs.com/webcat/navi/productInfo/edb/204254_spa.pdf?v=20190215083548
- Sustituyendo Mini Servo 9g, Servomotor Siemens 1FK7042-5AF21-1PA3 más Driver de potencia Sinamics S110 Profinet.
<https://docs-emea.rs-online.com/webdocs/127e/0900766b8127e237.pdf>



<https://docs-emea.rs-online.com/webdocs/13eb/0900766b813eb060.pdf>

- Sustituyendo StepperMotor, Motor paso a paso, 1.8°, Bobinado Bipolar, tipo Híbrido, 4 Cables, Par 1.0Nm, 24 VDC.

<https://docs-emea.rs-online.com/webdocs/132e/0900766b8132eb16.pdf>

- Sustituyendo DC Motor, Motor AC con caja reductora, 230 V a.c., 400 V a.c., 3 Fases, Inducción, 28 rpm, 180 W más variador de frecuencia 0,25 kW, 0 → 650Hz, 4,5 A, 230 V ac, IP20.

<https://docs-emea.rs-online.com/webdocs/068a/0900766b8068a7aa.pdf>

<https://docs-emea.rs-online.com/webdocs/1647/0900766b81647add.pdf>

- Sustituyendo Arduino Nano, CPU para PLC Siemens S7-1200 6ES7211-1BE40-0XB0

<https://docs-emea.rs-online.com/webdocs/1639/0900766b81639d60.pdf>

- Sustituyendo pantalla OLED, Pantalla táctil HMI, Siemens, 4,3 pulg., TFT, Color, 480 x 272pixels, 141 x 116 x 33 mm.

<https://docs-emea.rs-online.com/webdocs/13a9/0900766b813a9d2a.pdf>

