

**Microcontroladores y electrónica de potencia**

Proyecto global integrador  
**Desvío de cajas mediante barrera automática**

Ferran Martinez   
Legajo Nº 12601  
Ingeniería en mecatrónica

Índice

[Introducción 3](#_Toc173632846)

[Esquema tecnológico 4](#_Toc173632847)

[Detalle de módulos 6](#_Toc173632848)

[Microcontrolador 6](#_Toc173632849)

[Motor 1 6](#_Toc173632850)

[Motor 2 7](#_Toc173632851)

[Sensor 1 y Sensor 2 8](#_Toc173632852)

[LCD 9](#_Toc173632853)

[CP2104 TTL to USB 9](#_Toc173632854)

[Leds 10](#_Toc173632855)

[Funcionamiento general 11](#_Toc173632856)

[1 Inicio del sistema 11](#_Toc173632857)

[2 Detección de una caja y desviación 12](#_Toc173632858)

[3 Caja procesada correctamente 13](#_Toc173632859)

[4 Detección de otra caja 14](#_Toc173632860)

[5 No detección de una caja que fue desviada 14](#_Toc173632861)

[Programación 15](#_Toc173632862)

[Configuración del clock 15](#_Toc173632863)

[Configuración de los pines 16](#_Toc173632864)

[Configuración de los timers 18](#_Toc173632865)

[TIM1 18](#_Toc173632866)

[TIM2 18](#_Toc173632867)

[TIM3 18](#_Toc173632868)

[Interrupciones y prioridad de interrupción 19](#_Toc173632869)

[Programa 20](#_Toc173632870)

[Variables globales 20](#_Toc173632871)

[Funcion menú para la selección de opciones 22](#_Toc173632872)

[Inicializacion del sistema 24](#_Toc173632873)

[Captura y proceso de las entradas del usuario 24](#_Toc173632874)

[Interrupción de los sensores 27](#_Toc173632875)

[Movimiento del motor 1 29](#_Toc173632876)

[Mensajes por LCD 34](#_Toc173632877)

[Etapas de montaje 35](#_Toc173632878)

[Conclusiones 38](#_Toc173632879)

[Referencias 39](#_Toc173632880)

# Introducción

En el presente proyecto se realiza el desarrollo del control de cajas que circulan en una cinta transportadora y el desvío a otro canal u otra cinta en caso de ser necesario.

La idea para el desarrollo del proyecto reside en imitar los sistemas de clasificación de maletas o valijas que hay en el interior del aeropuerto (véase la figura 1), donde se identifica la maleta y se deriva al avión que le corresponda.



Figura Nº 1: Sistema de direccionamiento de equipajes de un aeropuerto.

El objetivo del proyecto es identificar correctamente dos tipos de cajas: las negras y las blancas. Para lograr esto, se utilizará un sensor que será capaz de diferenciar entre ambos tipos de cajas.

Cuando el sensor detecte una caja blanca, la compuerta o barrera se activará para desviar esta caja del camino original. Mientras tanto, las cajas negras continuarán su curso normal.

# Esquema tecnológico

Para el correcto funcionamiento del sistema, se plantea la siguiente trama lógica que debe realizar el microcontrolador emplazado.

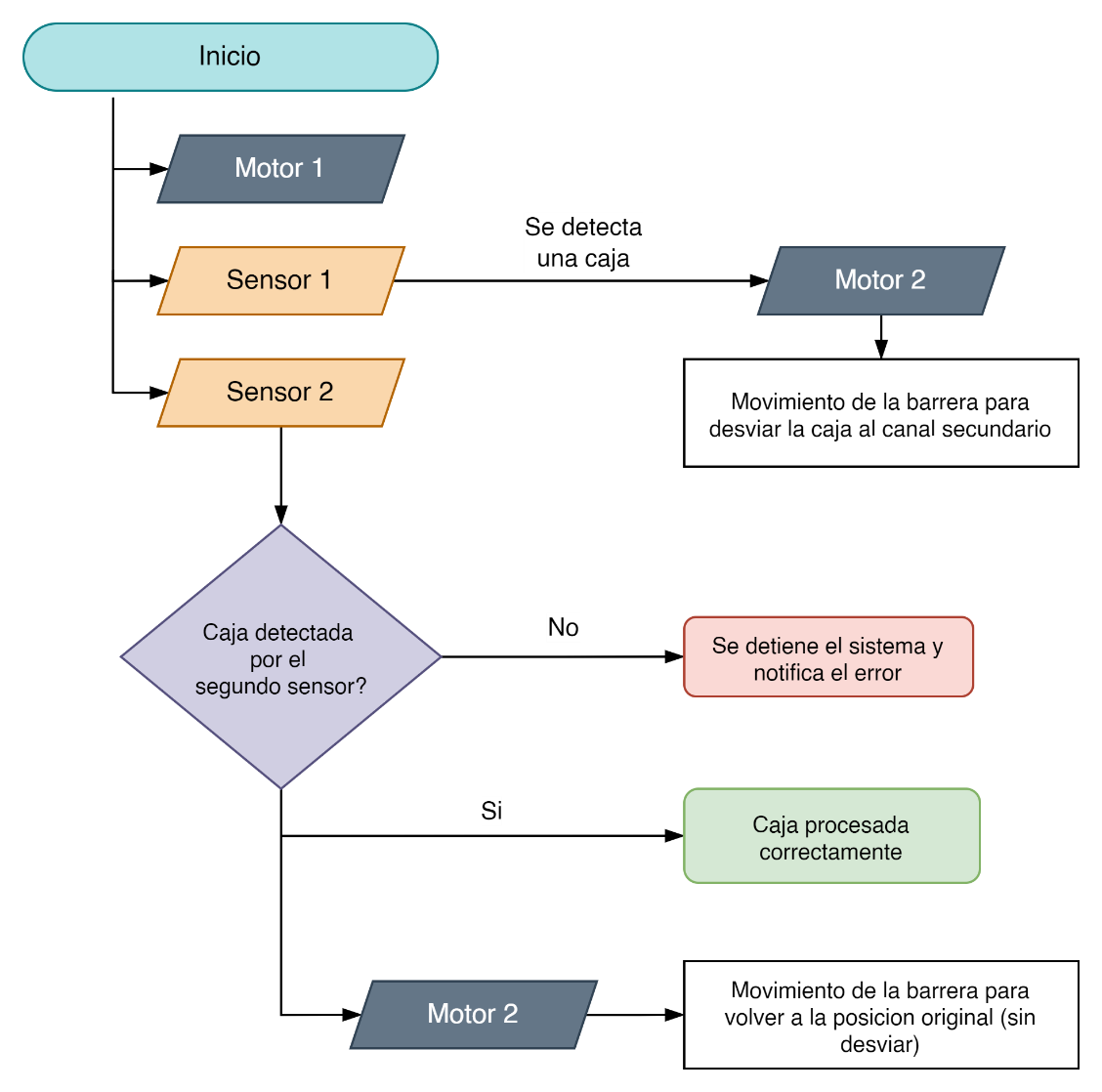


Diagrama Nº 1

El funcionamiento propuesto con el diagrama Nº 1 se desarrolla de la siguiente forma

1. Al iniciar se pone al sistema en marcha, lo que activa al motor 1 poniendo en movimiento a las cajas en el canal principal. Al iniciar, el sistema tendrá disposición de las señales de los dos sensores colocados a lo largo del canal .
2. Una vez se ha iniciado el sistema, las cajas empiezan a circular mediante la cinta transportadora, y por lo tanto en algún momento se detectará a través del primer sensor una caja blanca.
3. Al detectarse una caja con el primer sensor, se activa el segundo motor, que será el encargado de posicionar la barrera o compuerta de tal forma que la caja blanca tope y sea desviada a un segundo canal.
4. Finalmente, para saber si el proceso del desvío se realizó de forma correcta, un segundo sensor debe detectar que la caja blanca fue desviada correctamente.
5. En caso de que el segundo sensor no detecte una caja, el sistema se detiene, frenando al motor encargado de la cinta transportadora y deshabilitando las señales de los sensores.

Para lograr el cometido propuesto en el diagrama Nº 1, se opta por escoger el siguiente esquema de componentes

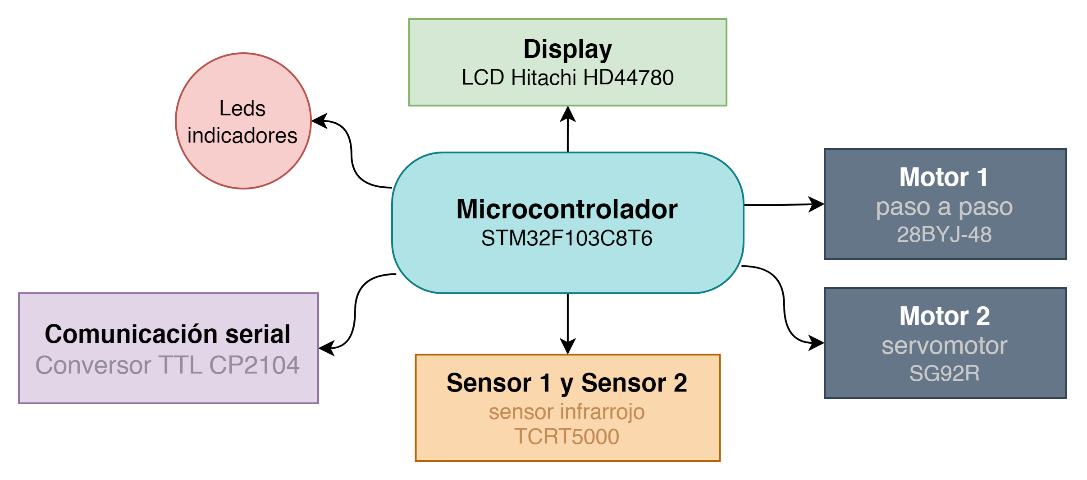


Diagrama Nº 2

# Detalle de módulos

A continuación, se detallará la elección de los componentes (véase el diagrama Nº 2) utilizados para el desarrollo del proyecto.

## Microcontrolador

Se selecciono el microcontrolador STM32F103C8T6 para la gestión del sistema. Está basado en una arquitectura ARM córtex M3 de 32 bits operando en 72MHz de frecuencia.

Dicho microcontrolador viene acoplado en la placa de desarrollo conocida como bluepill, facilitando el acceso a los pines y alimentación del microcontrolador.

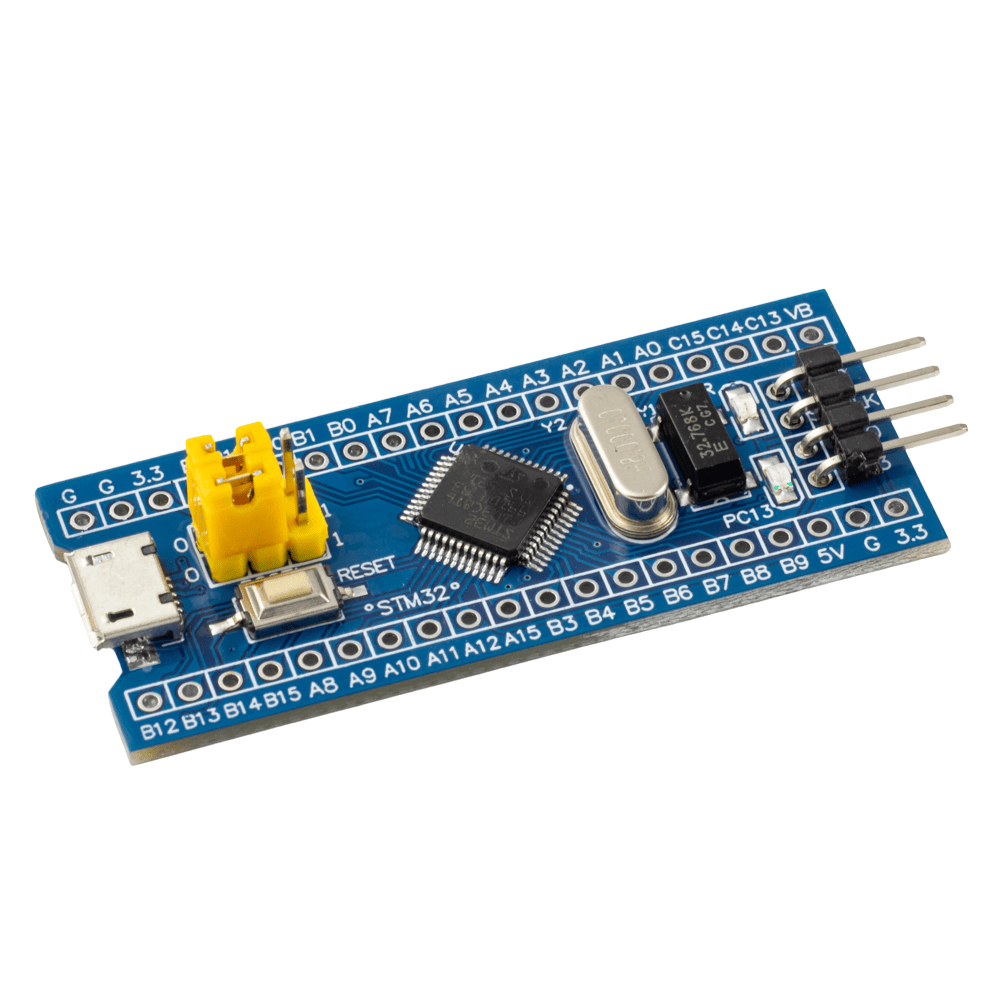


Figura Nº 2: Placa de desarrollo bluepill con el microcontrolador STM32F103C8T6 incorporado

Otras características importantes del microcontrolador

* Memoria flash de 64kB y 20kB de memoria RAM
* Interfaz de comunicación USART, SPI e I2C
* 3 timers de 16 bits y un timer de 32 bits (advanced timer)
* Opera con tensión de 2V a 3.6V
* 26 pines de entrada y salida

Con estas características el microcontrolador es suficientemente apto para controlar el funcionamiento del sistema de clasificación de cajas.

## Motor 1

Para el motor encargado de accionar la cinta transportadora se ha seleccionado un motor paso a paso unipolar 28BYJ-48 y para manejar el motor se usa el integrado TBD62003APG, que nos permite alimentar cada una de las bobinas del motor según la lógica deseada.

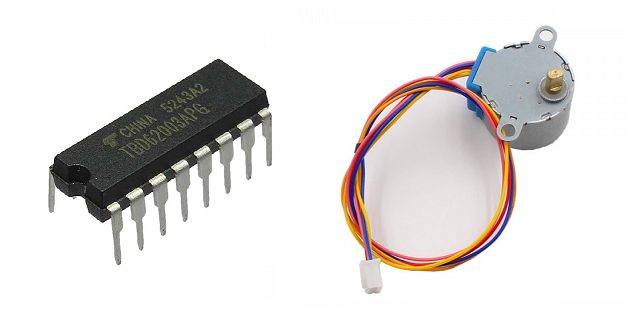


Figura Nº 3

El motor ofrece las siguientes características

|  |  |  |
| --- | --- | --- |
| Tensión de alimentación | 5 | V |
| Cantidad de fases | 4 |  |
| Resistencia por bobina | 50 | Ω |
| Reducción | 1/64 | (5.265º/64) |
| Torque | 34.3 | mN.m |

Tabla Nº 1

La selección de un motor paso a paso junto con al integrado es suficiente para el manejo de la cinta transportadora, ya que podemos controlar la velocidad del motor variando la frecuencia con la que se energizan las bobinas y además contar la cantidad de pulsos que requiere que una caja se traslade desde el primer sensor hasta el segundo sensor.

## Motor 2

Para el movimiento de la compuerta o barrera se consideraron dos opciones, incluir un servomotor o un motor paso a paso. Se decidió seleccionar el servomotor ya que utiliza menos pines que el paso a paso (considerando que se contaba con un motor paso a paso unipolar) y es más sencillo de programar su accionamiento.



Figura Nº 4: Servomotor Tower Pro SG92R con un rango de 180º

Para el proyecto se usa el servomotor ilustrado en la figura Nº 4, cuyas especificaciones se detallan en la siguiente tabla:

|  |  |  |
| --- | --- | --- |
| Tensión de alimentación | 4.8 | V |
| Periodo | 20 | ms |
| Ancho de pulso | 1 ~ 2.5 | ms |
| Rango de rotación | 180 | º |
| Velocidad de giro | 0.08 | s/60º |

Tabla Nº 2

La barrera que desviará las cajas se encontrara en todo momento en posición de no desviar, es decir que se encontrará en paralelo con la dirección de las cajas. Cuando se detecta una caja esta barrera se colocará gracias al servomotor en una posición de 45º. Por lo tanto, no existe ningún problema al usar el servomotor SG92R para el desarrollo del proyecto.

## Sensor 1 y Sensor 2

Para detectar las cajas se usa un sensor infrarrojo tanto para detectar la caja a desviar como para confirmar que la caja fue desviada correctamente. En este caso se usa el sensor TCRT5000, el cual es un módulo seguidor de línea, donde da un nivel de alto lógico hasta que detecta un obstáculo, a excepción de que dicho obstáculo sea de color negro, en tal caso no lo detecta y la salida del sensor seguirá siendo de un alto lógico.



Figura Nº 5 : Módulo TCRT500 seguidor de línea

Además, el sensor cuenta con un potenciómetro que le permite ajustar el rango de detección, permitiendo que se coloque a un costado de la cinta transportadora y se pueda ajustar a las características de dicha cinta.

## LCD

Para la visualización de datos de interés como puede ser la cantidad de cajas procesadas o en caso de haber un error, se selecciona un LCD simple como es el Hitachi HDD44780 de 16x2 (16 columnas y dos filas). Se selecciono este elemento porque es sencillo de programar y para el caso propuesto cumple los requisitos mínimos para mostrar la información del sistema.



Figura Nº 6 : LCD Hitachi HDD44780 16x2 de pantalla azul

## CP2104 TTL to USB

El conversor serie va a permitir que nos comuniquemos con el microcontrolador desde nuestro computador o PC a través de un canal serie. Este módulo cuenta con un USB y 6 pines, los cuales se comunicarán con la UART del microcontrolador.

Imagen que contiene par

Descripción generada automáticamente

Figura Nº 7: Convertidor serie con un módulo USB 2.0

## Leds

Considerando que puede surgir un error al desviar las cajas, es necesario no solamente contar con el LCD para dar el aviso, si no que debería haber otros indicadores visuales más sencillos de ver, en este caso, el indicador visual será un led para señalar que se está desviando una caja y otro led intermitente para notificar que una caja no fue desviada.



Figura Nº 8 : Leds de colores

# Funcionamiento general

Para abordar el desarrollo del proyecto que hace que el funcionamiento del sistema sea lo más completo posible, se ha dividido el mismo en las siguientes partes

1. Inicio y configuración del sistema por comandos en consola.
2. Detección de una caja y su posterior desviación mediante la barrera.
3. Detección de que la caja fue desviada correctamente.
4. Detección de otra caja.
5. No detección de que la caja fue desviada.

De esta forma se espera un mejor entendimiento del programa desarrollado en c que se explicará en la sección posterior.

## 1 Inicio del sistema

El sistema se encuentra inicialmente detenido, es decir que el motor 1 encargado de la cinta transportadora se encuentra inmovilizado y los sensores estarán inhabilitados. Inicialmente contamos con un menú para configurar el funcionamiento.

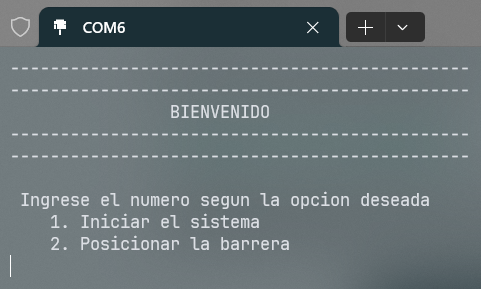


Figura Nº 9: Menú principal cuando el sistema está detenido.

En dicho menú podemos iniciar el sistema o, en caso de que haya elementos sobre la cinta, dar una posición inicial como puede ser la posición para desviar cajas.

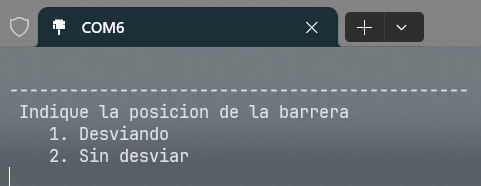


Figura Nº 10: Opciones para el cambio de posición de la barrera desde la consola de comandos del PC.

Al iniciar el sistema se abre otro menú en el cual indicaremos la velocidad deseada de la cinta.

Texto

Descripción generada automáticamente

Figura Nº 11: Menú para la selección inicial de la velocidad.

En el menú se observa una unidad que no es de velocidad, si no su inversa, y esto es así para ayudar a entender al usuario cuanto tiempo tardará una caja en recorrer un metro, por lo que según sea la distancia entre sensores, será más fácil conocer el tiempo que tardará en completarse el proceso.

Una vez se ha seleccionado una velocidad, se enciende el motor 1 y la cinta empezará a trasladar las respectivas cajas y los sensores se encontrarán habilitados para la detección de cajas.

Cuando el sistema se encuentra en marcha, por consola contamos con el siguiente menú.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Figura Nº 12: Menú del sistema en marcha.

En dicho menú podemos modificar la velocidad de la cinta o en caso de ser necesario, detener completamente el sistema, lo que hará que el motor 1 deje de funcionar, se deshabiliten los sensores y nuevamente el usuario se encuentre en el estado inicial.

## 2 Detección de una caja y desviación

Una vez se ha iniciado el sistema, los sensores ya se encuentran habilitados para la detección de cajas. Como se comentó en la sección de detalle de módulos, los sensores utilizados no detectan cajas de color negro y a nuestro favor estas cajas no detectadas no serán desviadas y no se iniciará ningún proceso para desviarlas.

Al detectarse otra caja que no sea de color negro, se inicia el proceso para desviar la misma a un canal secundario. La barrera se encuentra en una posición intermedia entre los dos sensores (véase la figura Nº13), por lo que al detectar una caja la barrera no cambia de posición automáticamente, si no que espera una cierta cantidad de tiempo antes de hacerlo.

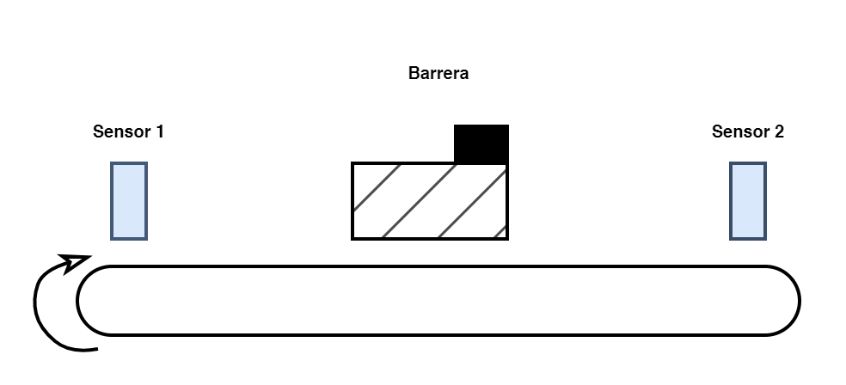


Figura Nº 13: Disposición de los sensores y la barrera sobre la cinta transportadora.

Este retraso entre la detección de la caja y el desvío de esta permite que si hay alguna caja negra en camino ésta no sea desviada por equivocación. Por lo tanto, el retraso permitirá a dicha caja negra pasar por la barrera sin ser desviada y luego la barrera se activará para desviar a la caja detectada por el primer sensor.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Figura Nº 14: Mensaje por puerto serie de la detección de la caja.

## 3 Caja procesada correctamente

En este punto, el sistema se encuentra en proceso y el motor 2 se encuentra en posición de desviar, entonces solo queda que el segundo sensor detecte la caja que ha sido desviada.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Figura Nº 15: Mensaje por puerto serie de la detección de la caja en el segundo sensor.

Durante el tiempo que la caja se encuentre entre el primer y el segundo sensor se activará un led para indicar que el proceso de desviación se está llevando a cabo. Durante este tiempo el sensor debe detectar a la caja y en dicho caso se aumentará en uno el contador de cajas procesadas y se informará a través del LCD.

Finalmente, una vez se procesó la caja se procede a modificar la posición de la barrera a su posición inicial, es decir, sin desviar.

## 4 Detección de otra caja

Puede ser que en el lapso de 5 segundos que dura el proceso, el primer sensor detecte otra caja a desviar. En este caso se reinicia el proceso, pero manteniendo la posición de la barrera para desviar. De esta forma evitamos accionar múltiples veces el motor 2 cuando se detecten dos cajas consecutivas.

## 5 No detección de una caja que fue desviada

Puede suceder que se haya detectado una caja y que la misma deba ser desviada, pero resulte que el segundo sensor no la detecte. Por lo tanto, podría considerarse una falla de alguno de los sensores o del motor 2 que mueve la barrera para el desvío de las cajas.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Figura Nº 16: Mensaje de error acerca del desvío fallido de la caja.

En este caso se detiene el sistema, esto considera el frenado del motor 1, un aviso de error al usuario y el encendido intermitente de un led de alerta.

Para volver a iniciar el sistema una vez se haya detectado la falla o problema que inicio el error, se vuelve a ingresar al menú de inicio. Al iniciar nuevamente se apagará el led de alerta y se iniciará el sistema como se consideró en el primer punto.

# Programación

Al realizar el desarrollo con un microcontrolador del modelo STM32, el proveedor del modelo ofrece un software para su desarrollo, dicho software es el STM32CubeIDE, un entorno de desarrollo para la configuración, programación y debug de proyectos en microcontroladores STM32.

Imagen que contiene Logotipo

Descripción generada automáticamente

Figura Nº 17: Logo del STM32CubeIDE

Este entorno de desarrollo nos facilita el desarrollo proyecto ofreciendo

* Selección del microcontrolador con el que se trabajará, lo que prepara las librerías necesarias y genera los archivos necesarios para su funcionamiento.
* Fácil asignación de los pines del microcontrolador a través de una imagen del microcontrolador mismo.
* Rápida configuración de timers y otras funciones para los pines como lectura, conectividad y analógicos.

## Configuración del clock

La configuración más importante para el microcontrolador es la frecuencia de clock, esta frecuencia determina la rapidez con la que el microcontrolador trabajará y será la base para configurar los timers.

Al trabajar con la placa de desarrollo Bluepill el clock viene de un cristal externo, por lo tanto, esto debe indicarse desde el entorno de desarrollo.

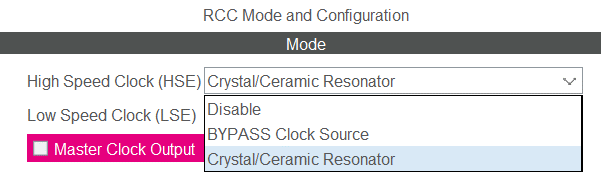


Figura Nº 18: configuración del clock externo.

Una vez configurado el clock externo, ahora se puede modificar la frecuencia a la que trabajará el microcontrolador. Recordemos que la frecuencia máxima es de 72MHz y para este proyecto se ha decidido trabajar con esta misma frecuencia.

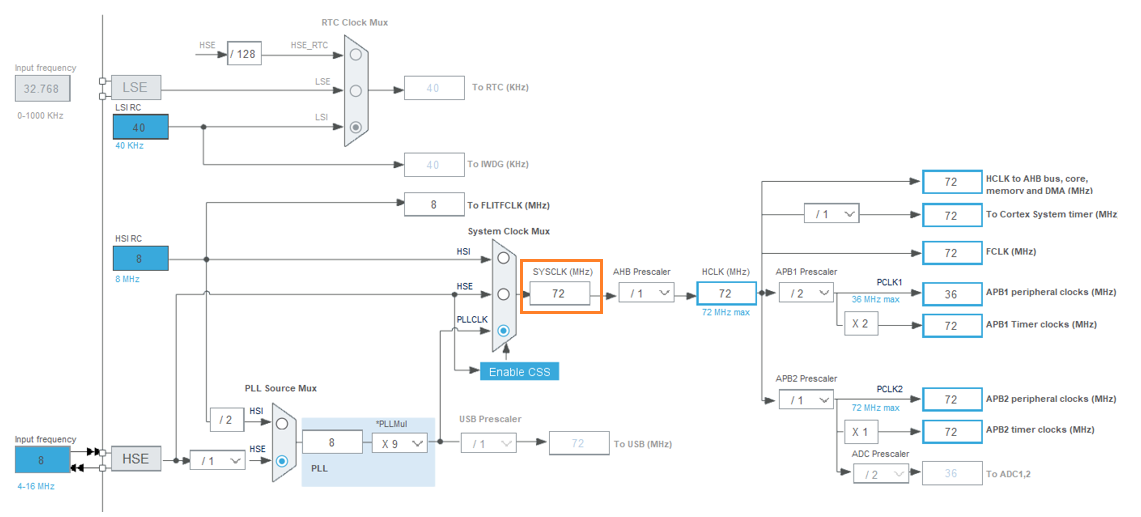


Figura Nº 19: Vista del clock configuration del entorno. Se remarca la selección de 72 MHz para el SYSCLK.

Esta configuración va a ocupar dos pines en nuestro esquema de pines de la siguiente sección. Estos pines serán **RCC\_OSC\_IN** y **RCC\_OSC\_OUT**.

## Configuración de los pines

Para este proyecto, la disposición de los pines será la siguiente

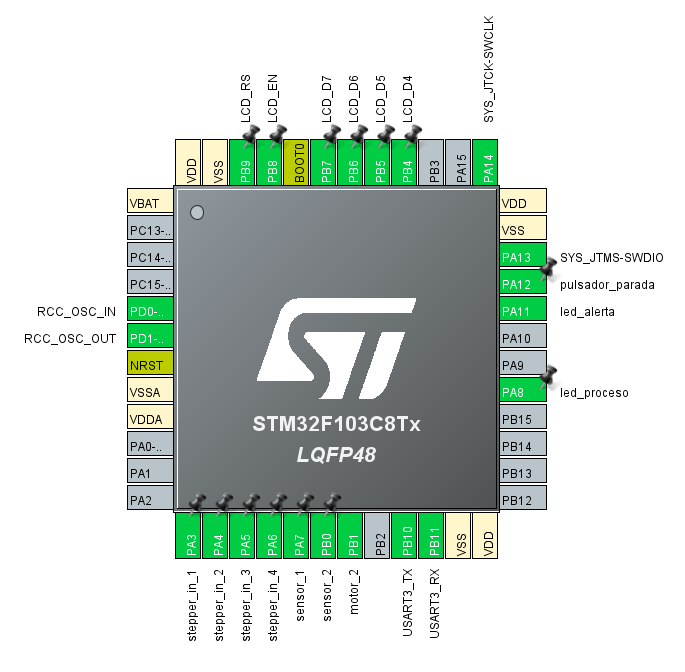


Figura Nº 20: Distribución de pines para el microcontrolador seleccionado

En total se ocupan 18 pines y el detalle de cada uno se explica en la siguiente tabla

|  |  |  |
| --- | --- | --- |
| **PIN** | **Detalle** | **Configurado como** |
| **A3** | Salida 1 del motor 1 | GPIO |
| **A4** | Salida 2 del motor 1 | GPIO |
| **A5** | Salida 3 del motor 1 | GPIO |
| **A6** | Salida 4 del motor 1 | GPIO |
| **PB1** | Salida al PWM del motor 2 | PWM |
| **PB0** | Entrada de la señal del sensor 2 | GP EXTI input |
| **A7** | Entrada de la señal del sensor 1 | GP EXTI input |
| **A12** | Entrada de la señal del pulsador de parada | GP EXTI input |
| **A11** | Led de alerta | PWM |
| **A8** | Led de proceso | GPIO |
| **B4** | Conexión a la terminal D4 del LCD | GP Output |
| **B5** | Conexión a la terminal D5 del LCD | GP Output |
| **B6** | Conexión a la terminal D6 del LCD | GP Output |
| **B7** | Conexión a la terminal D7 del LCD | GP Output |
| **B8** | Conexión a la terminal EN del LCD | GP Output |
| **B9** | Conexión a la terminal RS del LCD | GP Output |
| **B10** | Pin TX para la UART | USART |
| **B11** | Pin RX para la UART | USART |

Tabla Nº 3

Donde:

|  |  |  |
| --- | --- | --- |
|  | **GP EXTI input** | General purpose input with external interruption |
|  | **GP Output** | General purpose output |
|  | **USART** | Universal Sync/Async Receiver/Transmitter |
|  |  |  |

## Configuración de los timers

Se configuraron tres de los cuatro timers disponibles en el controlador, donde se ocupan para las siguientes funcionalidades

|  |
| --- |
| **TIM1**: Temporización de 1s para led de alerta |
| **TIM2**: Control del motor 1 |
| **TIM3**: Control del motor 2 |

Cada timer cuenta a su vez con cuatro canales (channels), permitiendo que un pin determinado tenga una configuración de salida basada en la temporización del propio timer.

### TIM1

Usado para encender y apagar el led de alerta cuando una caja no es detectada. Dicho led se enciende y apaga en un segundo.

|  |  |  |  |
| --- | --- | --- | --- |
| **Clock source** | Internal clock |  | **CHANNEL 4** |
| **PSC** | 36000 |  | PWM generation mode 1 |
| **ARR** | 2000 |  | Pulse: 1000 |
| **Frecuencia** | 1 Hz |  |  |

### TIM2

Usado para controlar la frecuencia a la que se emiten los pulsos al motor paso a paso (motor 1). Recordemos que cada bobina debe ser energizada una cierta cantidad de tiempo para luego energizar la siguiente y seguir así según el patrón de paso designado.

|  |  |
| --- | --- |
| **Clock source** | Internal clock |
| **PSC** | 3000 |
| **ARR** | 240 |
| **Frecuencia** | 100 Hz |

Para este timer no se usan canales de salida.

### TIM3

Usado para controlar la posición del servomotor. Se deben tener en cuenta las características del motor provistas en la tabla Nº1.

|  |  |  |  |
| --- | --- | --- | --- |
| **Clock source** | Internal clock |  | **CHANNEL 4** |
| **PSC** | 22 |  | PWM generation mode 1 |
| **ARR** | 65454 |  | Pulse: 0 |
| **Frecuencia** | 50 Hz |  |  |

Estos valores de PSC y ARR se seleccionaron al momento de comprobar las posiciones del servomotor.

## Interrupciones y prioridad de interrupción

Los timers mencionados anteriormente se usarán para diferentes procesos que se comentarán en la sección de programa. Dichos procesos deben ejecutarse cuando el timer se desborda, es decir que el contador del timer alcanza el valor de ARR. Para tomar acción una vez esto ocurre se deben habilitar las interrupciones por desborde de timer.

Otra interrupción que debe habilitarse son las generadas por los sensores, cuyos pines fueron configurados como EXTI, pero igualmente deben habilitarse las interrupciones cuando se lea un nivel de bajo.

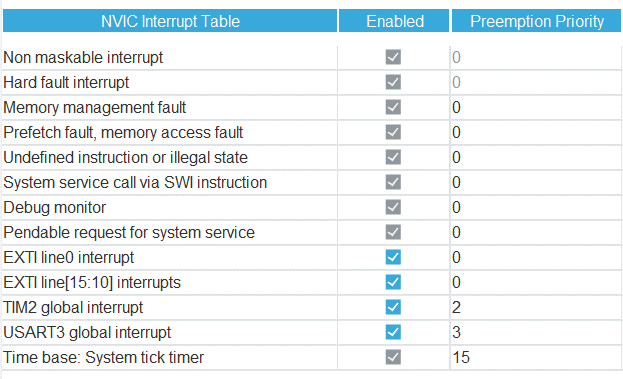


Tabla Nº 4: Tabla de interrupciones ordenadas por prioridad

En la tabla Nº4 se encuentran las interrupciones habilitadas en el microcontrolador. En la tercera columna se encuentra el nivel de prioridad de cada una de ellas, siendo 0 una prioridad alta y 15 la prioridad más baja.

Las interrupciones se identifican para cada elemento como

|  |  |
| --- | --- |
| **EXTI line0** | Sensor 2 |
| **EXTI line[10:15]** | Sensor 1 |
|  | Pulsador de parada |
| **TIM2** | Motor 1 |
| **USART3** | Mensajes por puerto serie |

Es importante que el nivel de prioridad de las interrupciones de USART sea lo más bajo posible, ya que se envían y procesan mensajes constantemente y si no se establece dicho nivel de prioridad, estos mensajes pueden interrumpir el funcionamiento del motor 1.

## Programa

Para abordar el programa desarrollado en lenguaje c, se dividirá la explicación de este en las siguientes partes

1. Variables globales y estados de los elementos
2. Inicialización del sistema
3. Funcion menú para la selección de opciones
4. Captura y proceso de las entradas del usuario
5. Interrupción de los sensores
6. Movimiento del motor1
7. Mensajes por LCD

De esta forma se espera un mejor entendimiento del programa desarrollado en c que se

### Variables globales

Aquí las variables importantes son las que se usaran para llevar la cuenta de los pulsos del motor y la configuración de la distancia entre sensores. En la parte del movimiento del motor se explicarán los valores de dichas variables.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

En la sección estados se identifican los diferentes estados para cada elemento del sistema. En el codigo expuesto se observan los siguientes elementos

|  |  |
| --- | --- |
| **estado\_sistema** | Determina el estado del proceso de derivar cajas |
| **motor\_1** | Determina la velocidad del motor 1 |
| **motor\_2** | Determina la posición del motor 2 |
| **cola\_de\_cajas** | Para conocer si existe alguna caja en cola |
| **opciones** | Para manejar las opciones del usuario en la recepción de datos |

### Funcion menú para la selección de opciones

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Estos serán los diferentes mensajes que se mostrarán por consola. Se declaran con anterioridad para usarlos directamente con la funcion HAL\_UART\_Transmit(huart, pData,Size,Timeout).

Luego se define la funcion menu\_seleccion(), que manejará las vistas por consola para cada momento del proceso.

Texto

Descripción generada automáticamente con confianza media

### Inicialización del sistema

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

La funcion principal inicializa los estados del sistema, inicia la comunicación con el LCD y genera el menú de opciones para el usuario.

### Captura y proceso de las entradas del usuario

La captura de las entradas del usuario se realiza mediante la interrupción de la USART, donde la funcion HAL\_UART\_Receive\_IT(huart, pData, Size) habilita la interrupción cada vez que se recibe un carácter desde el pin Rx.

Esta interrupción se maneja con la funcion de callback HAL\_UART\_RxCpltCallback(UART\_HandleTypeDef \*huart)



Y dentro de dicha funcion encontramos los diferentes bloques para procesar las opciones del usuario, siendo estos

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

### Interrupción de los sensores

Las interrupciones externas se manejan con la funcion HAL\_GPIO\_EXTI\_Callback(uint16\_t GPIO\_Pin), que se llamará cada vez que se detecte una interrupción por parte de alguno de los sensores o del pulsador de parada.

Texto

Descripción generada automáticamente con confianza media

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

### Movimiento del motor 1

Como se habrá visto en la sección de variables globales y estados, se declaran algunas relacionadas con los pulsos del motor, pulsos del proceso y para las velocidades del motor se usan valores de 240, 120 y 80.

Para llegar a estos valores primero se empieza conociendo que el motor paso a paso 28BYJ-48 necesita de 2048 pulsos de paso completo para que el eje de este realice una revolución.

Cuando trabaja con una frecuencia de 100Hz, a la salida del eje obtenemos una velocidad de

Esta velocidad, en una rueda de 10cm de diámetro que mueve a la cinta transportadora, se traduce en una velocidad lineal de 0,015 m/s. Si mantenemos una distancia de dos metros entre los dos sensores, una caja tardaría 133 segundos o unos 2 minutos y medio.

Por lo tanto, para agilizar el proceso y ofrecer al usuario una gama de velocidades variada, se opta por incluir una transmisión que aumente la velocidad a la salida.

Diagrama, Diagrama de Venn

Descripción generada automáticamente

Figura Nº 21: Relación de transmisión de la cinta transportadora.

Si tomamos los siguientes valores para los diámetros del juego de engranajes de la figura Nº21

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ø1 | 6 cm |  | Ø2 | 1 cm |  | Ø3 | 10 cm |

Y planteando diferentes frecuencias de trabajo, se obtienen las siguientes velocidades en la cinta transportadora

|  |  |  |  |
| --- | --- | --- | --- |
| Frecuencia de trabajo | Valor de ARR en TIM2 | Velocidad | Tiempo por metro recorrido |
| 100 Hz | 240 | 0,09 m/s | 10,86 s/m |
| 200 Hz | 120 | 0,18 m/s | 5,43 s/m |
| 300 Hz | 80 | 0,27 m/s | 3,62 s/m |

Y finalmente, la relación más importante para el control de las cajas es la cantidad de pulsos que debe realizar el motor para que una caja recorra un metro, siendo este valor de 1086 pulsos por metro recorrido.

Esto explica la constante PULSOS\_XM, que determina cuantos pulsos se necesitan para que la caja recorra un metro y así determinar cuántos pulsos se necesitan para recorrer la distancia entre los sensores, definida como D\_SENSORES.

Para movilizar al motor paso a paso se usan las siguientes funciones

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Donde

Calendario

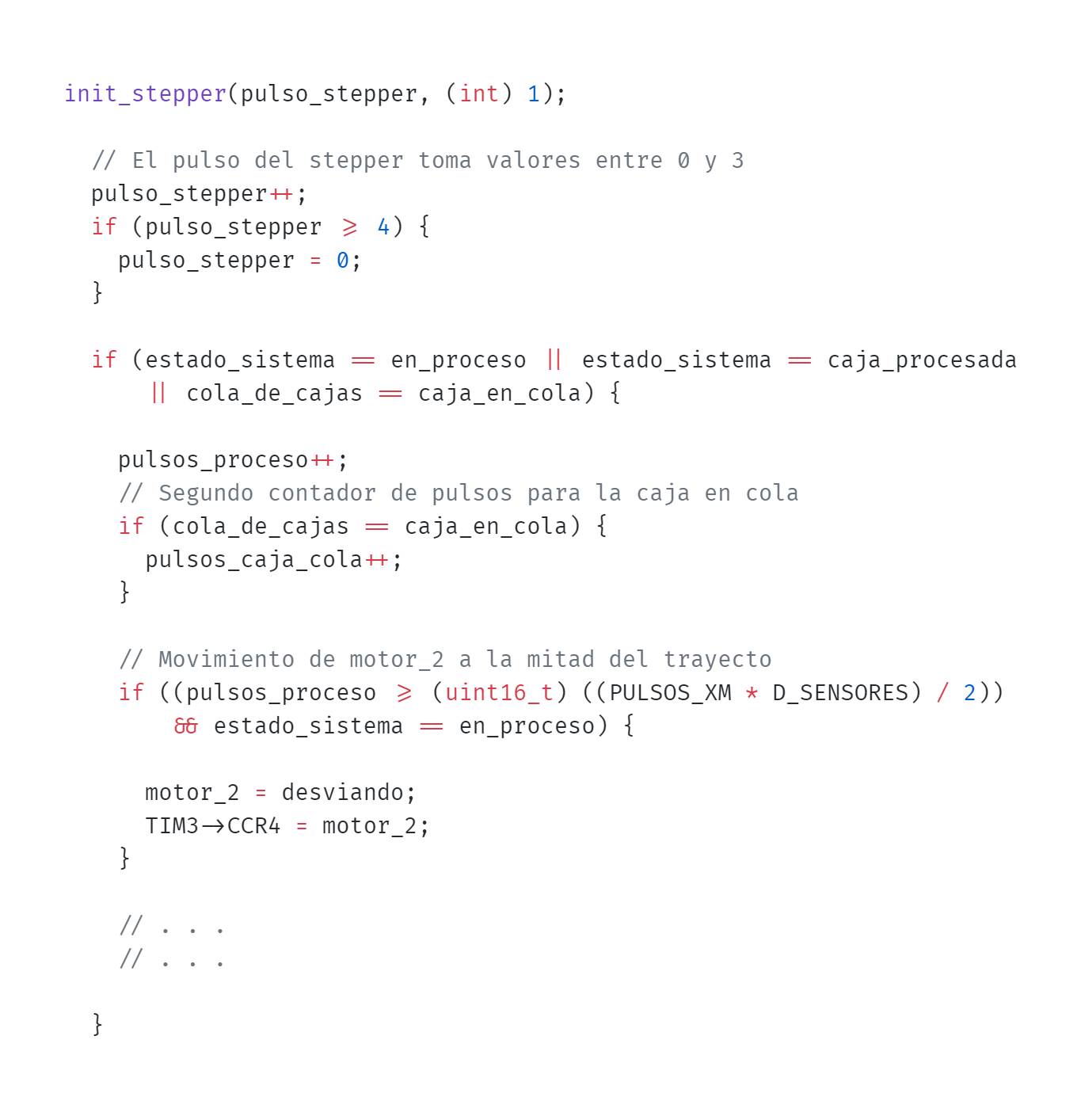
Descripción generada automáticamente

Ahora que sabemos cómo se moviliza al motor1 y cuáles son las constantes que ayudan a llevar la cuenta de los pulsos, se presenta el codigo que evalúa cada uno de los pulsos determinado por la velocidad seleccionada. Este codigo se ejecuta dentro de la funcion de callback HAL\_TIM\_PeriodElapsedCallback(TIM\_HandleTypeDef \*htim).

Interfaz de usuario gráfica, Texto

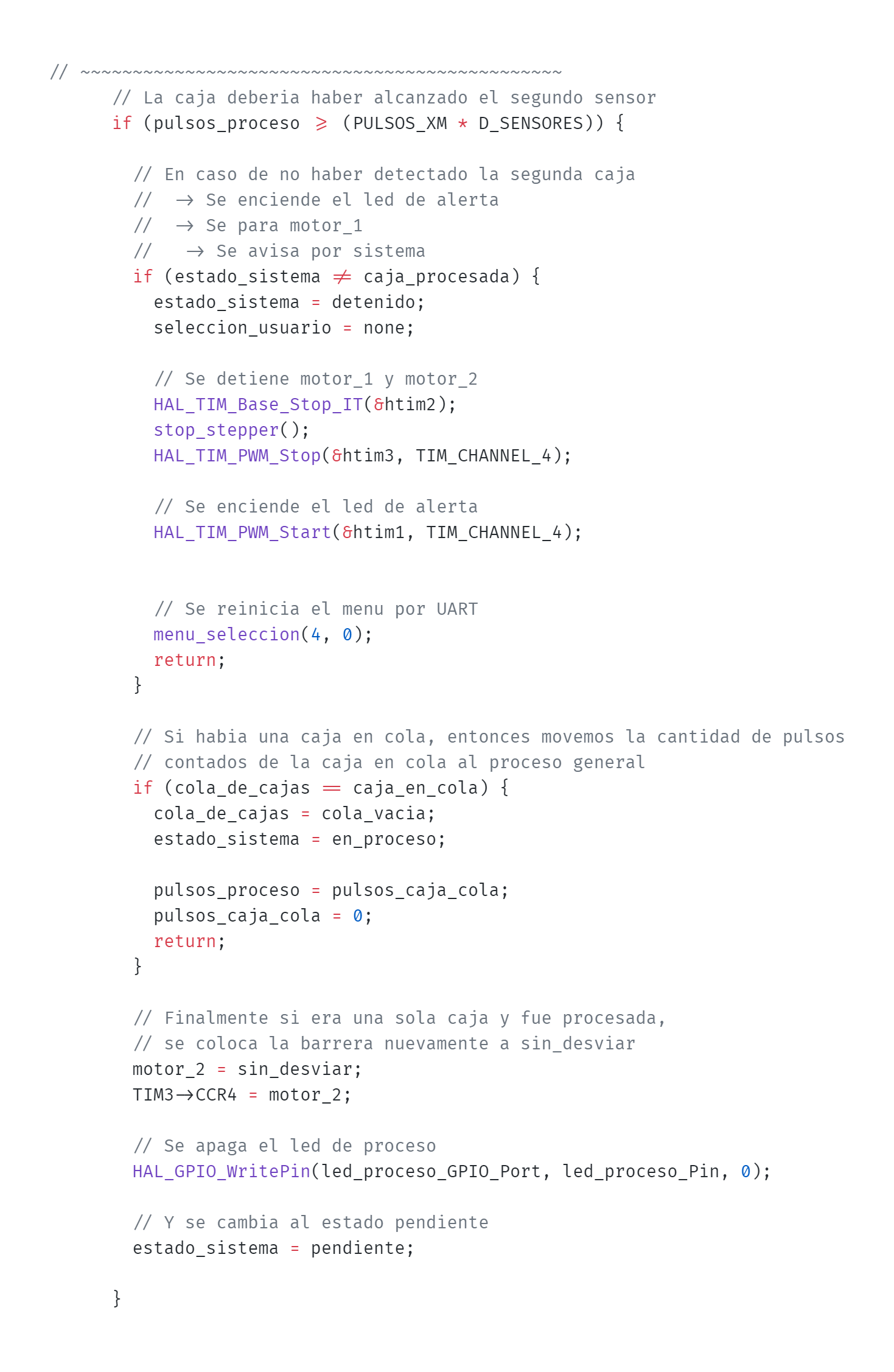
Descripción generada automáticamente

Que contiene la siguiente parte



Los pulsos de proceso llevarán la cuenta de la cantidad de pulsos que necesita el motor 1 para trasladar la caja desde el primer sensor hasta el segundo sensor. Es por ello por lo que esta variable solo aumenta cuando el sistema se encuentre en proceso, haya una caja en cola o la caja haya sido procesada lo que habilitará pasar los pulsos de la cola en la caja a los pulsos del proceso.

Finalmente, cuando los pulsos alcanzan los valores calculados anteriormente, se procede a realizar lo indicado en la siguiente porción de codigo.



### 

### Mensajes por LCD

Se usa una librería (ECUAL) para el manejo del LCD, facilitando así el desarrollo del programa y evitando posibles errores que puedan surgir al momento de realizar la comunicación con el mismo.

La librería cuenta con las siguientes funciones

Texto

Descripción generada automáticamente con confianza baja

Para un mejor manejo y a fin de evitar codigo repetitivo, se define la funcion LCD\_msg(char\*, unsigned char, unsigned char) para resumir las funciones de LCD\_Set\_Cursor(unsigned char, unsigned char) y LCD\_Write\_String(char\*).

Este nueva funcion pide como argumento la cadena de caracteres a mostrar y la posición de tal cadena de caracteres en el LCD.

# Etapas de montaje

El proyecto puede montarse fácilmente en una placa experimental o protoboard, y como se muestra en la figura Nº 22 las conexiones que hay en dicha placa son para alimentar a los diferentes elementos del sistema como los sensores y los motores.

Imagen que contiene tabla

Descripción generada automáticamente

Figura Nº 22

El circuito se alimenta con un transformador que cuenta con 5V y 1A de salida, y con ello es suficiente para que todo el sistema funcione correctamente.

|  |  |
| --- | --- |
| Imagen que contiene medidor  Descripción generada automáticamente | Imagen que contiene competencia de atletismo, tabla  Descripción generada automáticamente |

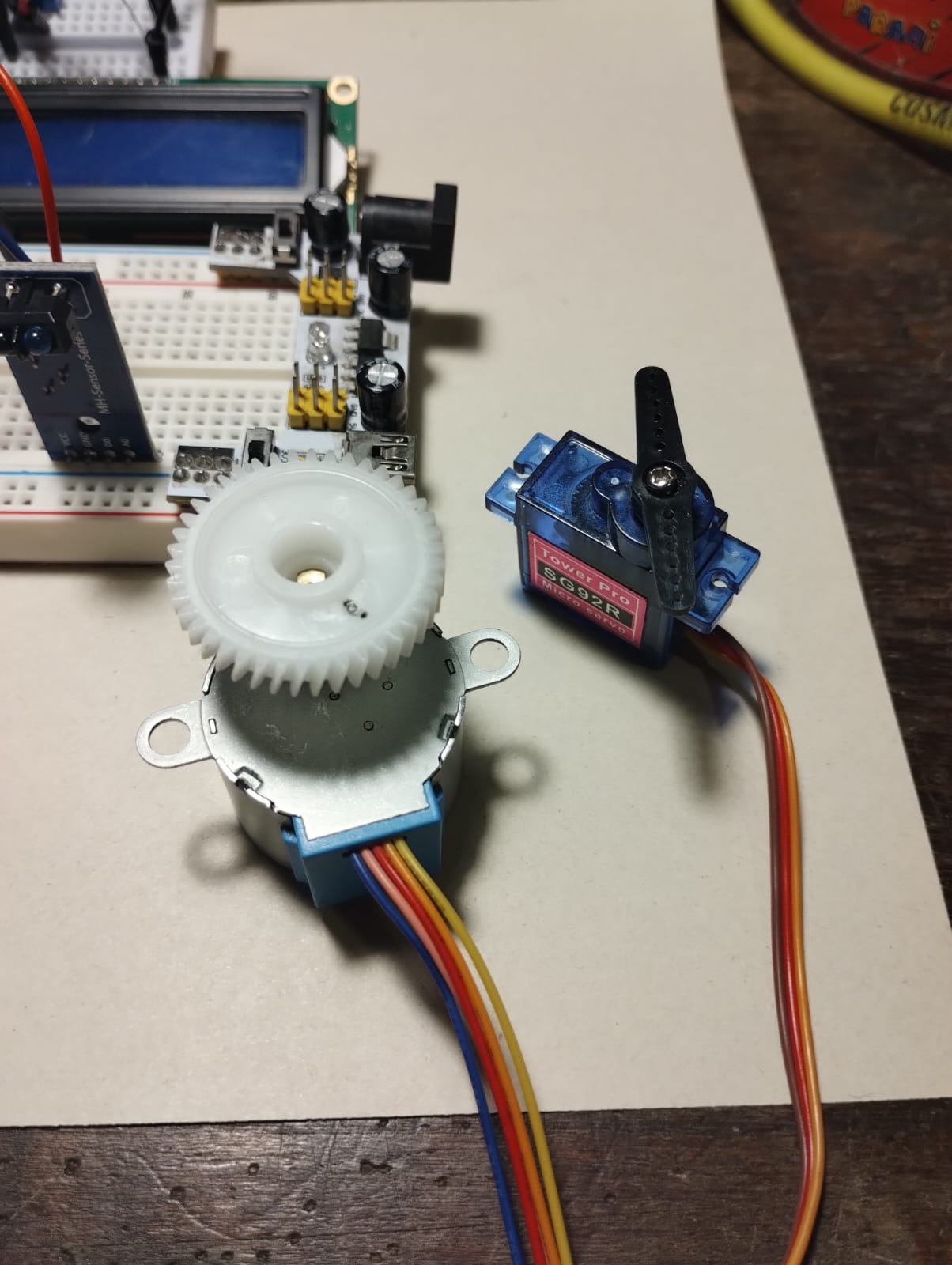
Figura Nº 23: En la figura no se ha incluido el LCD, aun así, todo el sistema funciona con el transformador de 5V y 1A .

En la figura Nº 24 se verifica que el primer sensor detecta colores blancos, pero no detecta un objeto con color negro.

|  |  |
| --- | --- |
|  | |
|  | |
|  |  |

Figura Nº 24: Sensores colocados en la protoboard, siendo el primer sensor el que está ubicado a la izquierda y el segundo sensor el ubicado a la derecha. Se comprueba en las imágenes inferiores que el sensor detecta (con un led rojo) el papel blanco, pero no detecta al papel pintado de negro.

En el montaje realizado a la fecha no se usa la transmisión comentada para lograr una mayor velocidad, sin embargo, se usa un engranaje colocado en el eje para visualizar el inicio del motor, el cambio de velocidad y el frenado de este.



De forma similar, el servomotor no cuenta con una barrea colocada, si no que trae acoplado un plástico alargado que nos facilita conocer la posición que tendría la barrera en su lugar.

|  |  |
| --- | --- |
|  |  |

Figura Nº 25: Posiciones que adoptara el servomotor. A la izquierda cuando se encuentra la barrera sin desviar, y a la derecha cuando se encuentra en posición de desviar.

En la protoboard se pueden encontrar implantados los leds para el proceso y para la alerta y el pulsador de parada.

|  |  |  |
| --- | --- | --- |
|  |  |  |

Finalmente, un elemento importante en la desviación de las cajas es el conteo de estas, y para ello se implementa el display LCD que nos dará dicha información sin tener que conectarnos por puerto seria al microcontrolador.



Figura Nº 26: El LCD se ha instalado en otra protoboard más pequeña con el fin de aliviar las conexiones en la protoboard principal.

# Ensayo de ingenieria

Si se requiere llevar el proyecto a una solucion industrial o real, es necesario realizar algunos cambios importantes respecto del prototipo desarrollado en este proyecto.

Los elementos para revisar serán

## Motores

El motor paso a paso implementado ha sido una buena opcion para el desarrollo y ha ayudado a resolver el problema del tiempo que pasa la caja entre sensor y sensor y en que momento debe ser desviado. Sin embargo, no se han contemplado el torque que debe ejercer para mover cargas reales. En una aplicación real se recomienda el uso de un motor de CC con encoders integrados y ajustados a las posibles variaciones de carga (torque variable) que haya en la cinta.

Para el segundo motor, el motor de la barrera, se ofrecen dos soluciones, una primer solucion consta de un motor de CC servocontrolado y para la segunda solucion se propone un accionamiento neumatico.

La primer opcion aumenta el coste del sistema pero es efectivo y en caso de mejorar al sistema, esta opcion es facilmente configurable.

El accionamiento neumatico es una opcion mas barata, pero requiere de la instalacion del sistema neumatico para el propio accionamiento. Ademas, el cierre y apertura de la barrera mediante este accionamiento es menos configurable.

## Sensores

Para un caso real, donde las cajas suponen equipaje, contamos con una gran variedad de colores y formas, sumado a que es la etiqueta del equipaje la que nos indica el avion que deben abordar. Entonces para detectar estos detalles se recomienda el uso de una camara que mediante el uso de vision artificial podemos usar para detectar y clasificar los distintos tipos de equipaje.

## Display

El uso del display LCD en este proyecto es meramente informativo, en un caso real no es necesario mostrar en un LCD el conteo de las cajas. Sin embargo, con lo que si deberia contar el sistema es con un SCADA, el cual mediante modulos de comunicación Modbus o CAN (sustituyendo a la comunicación serie mediante TTL a USB) nos informa en todo momento del estado del sistema.

Se suma a esto que el sistema es un solo derivador, pero en un caso real se cuenta con una mayor cantidad de derivaciones y por lo tanto diferentes cintas y barreras. Es por ello que con el SCADA tendriamos acceso al estado y la manipulacion de todos los dispositivos.

# Conclusiones

Inicialmente el proyecto contaba con un motor de corriente continua con escobillas (BDC) y a través de un encoder óptico se pensaba medir su velocidad. Si alguna perturbación ralentizaba la cinta, se media la velocidad y se podía calcular en qué momento debería posicionarse la barrera.

Este caso inicial presento varias complicaciones, sobre todo al momento del montaje, ya que, al no contar con un torque preciso sobre el eje del motor, la velocidad se media con el motor en vacío (sin carga que mover) y tal velocidad era muy alta, lo que causaba mediciones poco precisas.

Para resolver este problema se cambió el motor de corriente continua por un motor paso a paso, facilitando gran parte del trabajo ya que ahora el control de la velocidad de la cinta puede realizarse mediante los pulsos que alimentan las bobinas del motor paso a paso.

Otro elemento importante que no se consideró al iniciar el proyecto fue la implementación de la comunicación serie con el microcontrolador. Gracias a la idea de poder comunicar al usuario con el sistema se abrió un abanico de posibles interacciones o necesidades del usuario como el cambio de velocidad o el cambio en la posición inicial de la barrera.

Aunque el proyecto alcanzó los objetivos y expectativas propuestos, se plantean algunas mejoras para aplicar a futuro

* Montaje de una cinta para comprobar que los cálculos realizados para la relación pulsos/metro son correctos.
* Mejorar el programa aplicando funciones que resuman ciertas porciones de codigo como la selección del usuario y las decisiones a tomar cada vez que un sensor detecta una caja.
* Mejorar el menú de selección para que el usuario pueda inicialmente configurar la distancia entre sensores o una velocidad de cinta diferente a las ofrecidas.
* Usar un sensor RGB que detecte diferentes tipos de color en las cajas, de esta forma el usuario podría elegir qué tipo de cajas quiere derivar.
* Acceder a la memoria flash para guardar el conteo de cajas y permitir al usuario reiniciar esta cantidad por consola.

# Referencias

<https://lastminuteengineers.com/28byj48-stepper-motor-arduino-tutorial/>

<https://www.st.com/en/microcontrollers-microprocessors/stm32f1-series.html>

<https://deepbluembedded.com/>

<https://github.com/Khaled-Magdy-DeepBlue/STM32_Course_DeepBlue>