[[1]](#footnote-1)

Banda transportadora clasificadora

Aguilar Acevedo, Gabriel Eduardo

Resumen—

En este proyecto observamos lo que sería el proceso de manufactura automatizado de una banda transportadora clasificadora, este fue desarrollado con diferentes programas y elementos como lo es la Raspberry pi b+, utilizado para la simulación de un PLC (controlador lógico programable) y el programa Codesys que se utilizó para hacer una programación en escalera (LADDER). En este caso el proceso es clasificar el producto según el tamaño, color y tipo de productos, este proyecto también podría ser empleado para otros tipos de procesos como producción de latas o botellas, llenados y demás.

# **Introducción**

Aquí planteamos un Proyecto automatizado en el cual queremos mejorar el Sistema de producción así haciéndolo más eficaz en lo que es toda la manufactura de los productos y tratando de seguir llevando las industrias hacia el futuro y lograr mejores resultados.

# **Raspberry pi 3 b+**

*A.* 

*B.* **¿Qué es?**

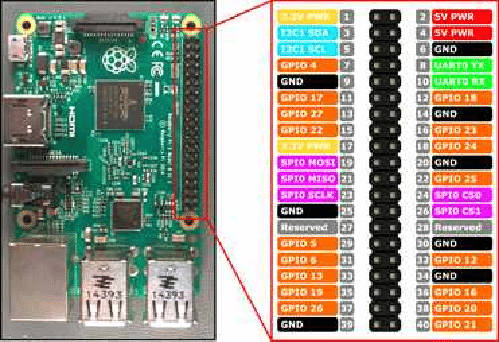
Raspberry Pi es un ordenador de placa reducida, ordenador de placa única u ordenador de placa simple (SBC) de bajo coste desarrollado en el Reino Unido por la Raspberry Pi Foundation.

Las placas de Raspberry se usan en todo tipo de proyectos. Puedes crear un ordenador simple conectándolo a un monitor, teclado y ratón. También puedes crear robots, controladores del internet de las cosas o prácticamente cualquier cosa que se te ocurra gracias a su gran versatilidad y tamaño.

*C.* **Especificaciones técnicas**

* CPU + GPU: Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
* RAM: 1GB LPDDR2 SDRAM
* Wi-Fi + Bluetooth: 2.4GHz y 5GHz IEEE 802.11.b/g/n/ac, Bluetooth 4.2, BLE
* Ethernet: Gigabit Ethernet sobre USB 2.0 (300 Mbps)
* GPIO de 40 pines
* HDMI
* 4 puertos USB 2.0
* Puerto CSI para conectar una cámara.
* Puerto DSI para conectar una pantalla táctil
* Salida de audio estéreo y vídeo compuesto
* Micro-SD
* Power-over-Ethernet (PoE)

*D. Pines GPIOS*

**

*GPIO (General Purpose Input/Output, Entrada/Salida de* Propósito General) es un pin genérico en un chip, cuyo comportamiento (incluyendo si es un pin de entrada o salida) se puede controlar (programar) por el usuario en tiempo de ejecución.

Los pines GPIO no tienen ningún propósito especial definido, y no se utilizan de forma predeterminada. La idea es que a veces, para el diseño de un sistema completo que utiliza el chip podría ser útil contar con un puñado de líneas digitales de control adicionales, y tenerlas a disposición ahorra el tiempo de tener que organizar circuitos adicionales para proporcionarlos*.*

**Usos***:*

* chips con pocos pines: IC, SoC, integrados y hardware a la medida, dispositivos lógicos programables (por ejemplo, FPGAs)
* chips multifunción: gestores de energía, códecs de audio, tarjetas de video.
* aplicaciones embebidas (por ejemplo, Arduino, BeagleBoard, Raspberry Pi) hacen un uso intensivo de GPIO para la lectura de varios sensores ambientales (IR, de vídeo, la temperatura, la orientación de 3 ejes, aceleración), y para enviar la salida a motores de corriente continua (mediante PWM), audio, LCD o pantallas LED para el estado.

**Capacidades**:

* Pines GPIO que pueden ser configurados para ser entrada o salida.
* Pines GPIO que pueden ser activados / desactivados.
* valores de entrada se pueden leer (por lo general alto = 1, bajo = 0)
* valores de salida de lectura / escritura.
* valores de entrada que a menudo se pueden utilizar como IRQ (típicamente para los eventos de activación)
* el dispositivo puede utilizar DMA para mover grandes cantidades de datos de manera eficiente dentro o fuera del dispositivo.
* Los periféricos GPIO varían muy ampliamente. En algunos casos son, simplemente, un grupo de pines que se puede cambiar en grupo ya sea de entrada o de salida. En otros, cada pin puede configurarse con flexibilidad para aceptar o entregar diferentes voltajes lógicos, y con la potencia de accionamiento configurables y tire ups / bajadas. Las tensiones de entrada y salida son típicamente la tensión de alimentación del dispositivo que contiene los GPIO, y pueden ser dañados por mayores tensiones. Algunos GPIO tienen entradas tolerantes de 5 V: incluso cuando el dispositivo tiene una tensión de alimentación baja (por ejemplo, 2 V), el dispositivo puede aceptar 5 V sin daño.

# **CODESYS**

CODESYS es un entorno de desarrollo para la programación de controladores conforme con el estándar industrial internacional IEC 61131-3. El término CODESYS es un acrónimo y significa Sistema de Desarrollo de Controladores.

Es desarrollado y comercializado por la empresa de software alemán 3S-Smart de Soluciones de Software situado en la ciudad bávara de Kempten. La Primera Versión (1,0) fue creada en 1994.

**Lenguajes de programación**

Los seis lenguajes de programación para aplicaciones vienen definidos en el IEC 61131-3 y están disponibles en el entorno de desarrollo CODESYS.

**Lenguajes de texto**

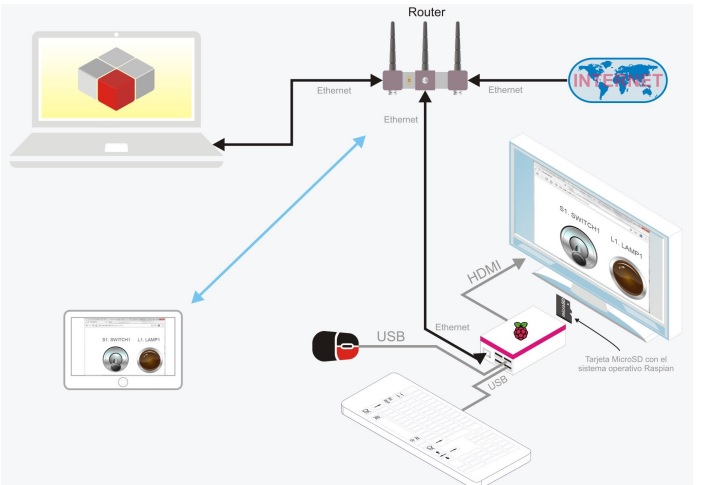
* IL (lista de instrucciones) es un lenguaje de programación parecido al lenguaje ensamblador.
* ST (texto estructurado) es similar a la programación en PASCAL o C

**Lenguaje gráficos**

* LD (Diagrama Ladder) permite al programador combinar los contactos de relé y las bobinas. Es el lenguaje de Programación de PLC por excelencia.
* FBD (diagrama de bloques de función) permite al usuario programar rápidamente, tanto expresiones como en lógica booleana.
* SFC (Bloques de función secuenciales) es conveniente para los procesos de programación secuencial
* Dispone también de un editor gráfico que no está definido en la norma IEC:
* CFC (Continuos Function Chart) es una especie de editor de FBD libre. Es un editor orientado a FBD donde las conexiones entre las entradas, salidas y los operadores se fijan automáticamente.

En este caso de nuestro proyecto utilizamos el lenguaje LADDER.

# **Comunicar CODESYS con Raspberry**

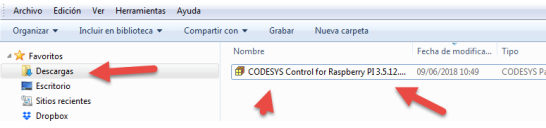


1. Debemos tener instalada alguna versión de CODESYS en nuestro PC. La descarga se puede hacer desde la página oficial de

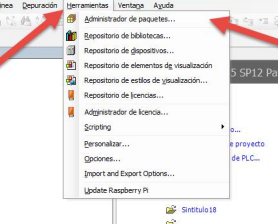
CODESYS, y recuerde que el programa permanece activo durante dos horas. Pasado este tiempo, se vuelve a arrancar el

programa y otras dos horas….

1. Se introduce Raspberry Pi, con el propósito de buscar la librería de CODESYS tiene del controlador de la frambuesa, es decir, CODESYS ha creado una extensión (librería, biblioteca…), para poder manejar la Raspberry desde su programa de automatización.
2. Una vez la descarga se ha completado, el paquete está guardado en Descargas.



1. No hace falta crear un nuevo proyecto. Desde la pantalla de inicio, se busca en Herramientas >> Administrador de paquetes.



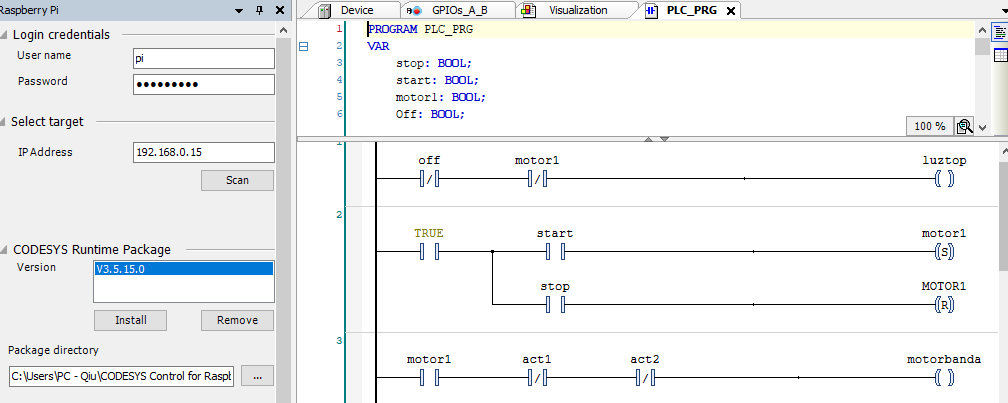
1. Se activa Instalando, y se busca la ruta en Descargas. Aparecen ejemplos de programaciones con la Raspberry y más documentos de interés. En cualquier caso, se da por terminada la instalación de la librería Raspberry Pi, para CODESYS
2. El siguiente paso, obligatoriamente se ha de crear un nuevo archivo de CODESYS, para que surtan los efectos de la incorporación de la librería.
3. Aquí se presenta un cambio importante. Como dispositivo de control hay que elegir CODESYS Control for Raspbberry Pi (3S– Smart Software Solutions GmbH). El lenguaje de programación ahora mismo no es primordial. Se ha elegido Ladder (LD).
4. Note, el cuadro de dispositivos. El control es para Raspberry, y en la parte inferior, el control GPIO, permitirá posteriormente controlar (vincular) las entradas y salidas de la Raspberry.

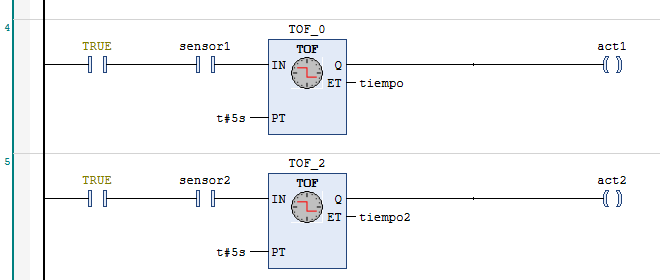
# **Proyecto**

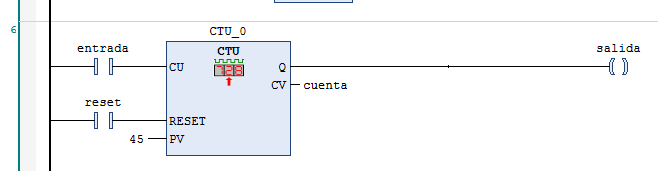
Consiste en lo que es una banda transportadora clasificadora en el cual se simula un PLC, se trabaja con el software de CODESYS y el hardware Raspberry pi que es el que simula el plc con sus pines GPIO.

Se realiza mediante un diagrama ladder con diferentes elementos como contactores, luces pilotos, motores, timers, entre otros.

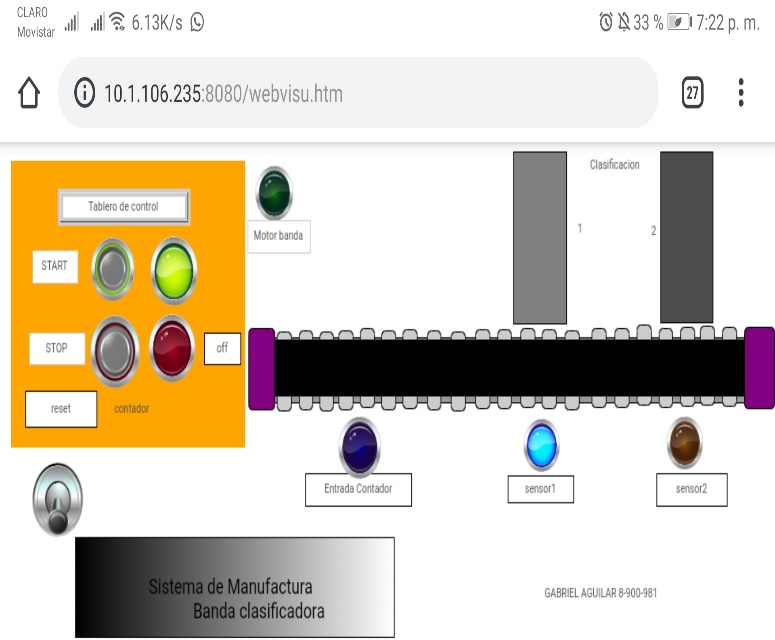
Programación ladder.







Visualizacion



# conclusión

Como pudimos observar el proyecto trato sobre la simulación de un PLC gracias a la utilización del programa codesys que por medio de unos paquetes especiales se logra conectar a la raspberry.

Se programó en lenguaje ladder y se simulo virtualmente en el codesys, para hacer esto en la vida real se ejecuta el programa en conjunto con el raspberry por conexión de red wifi y se envían las señales de salida y entrada por los puertos GPIO del raspberry pi 3 b+.

1. [↑](#footnote-ref-1)