

# Sistema elevador clasificador para paquetes

Jaime Aranda (1550106), Heberth Ardila (1250702), Daniel Marín (1450419), Sebastián Bolívar (1550268)

ortiz.jaime..., heberth.ardila..., marin.daniel..., juan.sebastian.bolivar... @correounivalle.edu.co

## Abstract

In this report the importance of automation in the industry, as well as the control of processes and automatic control systems will be exposed; In the same way, the development of the project will be explained, the implementations that were used, the type of components that were used, the technologies used, among others.

## Resumen

En el presente informe se expone la importancia de la automatización en la industria, así como el control de procesos y sistemas de control automáticos; de igual manera, se explicará el desarrollo del proyecto, las implementaciones que se hicieron, el tipo de componentes que se utilizaron, las tecnologías empleadas, entre otras.

## 1. Introducción

A partir de la inventiva, experiencia y demás virtudes que el hombre posee a podido generar miles de soluciones a sus problemas cotidianos, si unimos parte de estas virtudes con la necesidad de mejorar, ser más competitivo, reducir al máximo su participación en los trabajos pesados y generar para sí mismo mayores beneficios, encontramos soluciones tan creativas y a la vez tan avanzadas que nos permiten eliminar por un instante los límites que el hombre como ser racional posee.

En medio de su afán por mejorar sus producciones, conseguir mayores beneficios, y ser mejor cada día, el hombre de la mano de la tecnología, grandes investigaciones y una infinidad de posibilidades ha logrado crear sistemas automáticos, que de una u otra manera han hecho más fácil y a la vez más productiva la vida del hombre.

Dentro de las grandes posibilidades que se tiene a partir de lo existente, cualquier aparato, mecanismo o sistema es apto para sufrir modificaciones de este tipo, basta únicamente con aplicar conceptos de las ciencias básicas, con conceptos más nuevos, tecnológicos y de gran alcance.

Sistemas de cualquier tipo como mecánico, eléctrico, neumático, etc. son aquellos que sitúan dentro de la escala de evolución tecnológica al humano en cada espacio de tiempo, siendo cada uno de estos sistemas en la vida cotidiana del hombre indispensables para su funcionamiento normal dentro de cualquier ámbito.

La Automatización Industrial es la aplicación de diferentes tecnologías para controlar y monitorear un proceso, máquina, aparato o dispositivo que por lo regular cumple funciones o tareas repetitivas, haciendo que opere automáticamente, reduciendo al mínimo la intervención humana.

Lo que se busca con la Automatización industrial es generar la mayor cantidad de producto, en el menor tiempo posible, con el fin de reducir los costos y garantizar una uniformidad en la calidad.

La Automatización Industrial es posible gracias a la unión de distintas tecnologías, por ejemplo la *instrumentación* nos permite medir las variables de la materia en sus diferentes estados, gases, sólidos y líquidos, (eso quiere decir que medimos cosas como el volumen, el peso, la presión etc.), la *oleohidráulica*, la *neumática*, los *servos* y los *motores* son los encargados del movimiento, nos ayudan a realizar esfuerzos físicos (mover una bomba, prensar o desplazar un objeto), los *sensores* nos indican lo que está sucediendo con el proceso, donde se encuentra en un momento determinado y dar la señal para que siga el siguiente paso, los *sistemas de comunicación* enlazan todas las partes y los Controladores Lógicos Programables o por sus siglas *PLC* se encargan de controlar que todo tenga una secuencia, toma decisiones de acuerdo a una programación pre establecida, se encarga de que el proceso cumpla con una repetición, a esto debemos añadir otras tecnologías como el *vacío*, la *robótica*, *telemetría* y otras más.

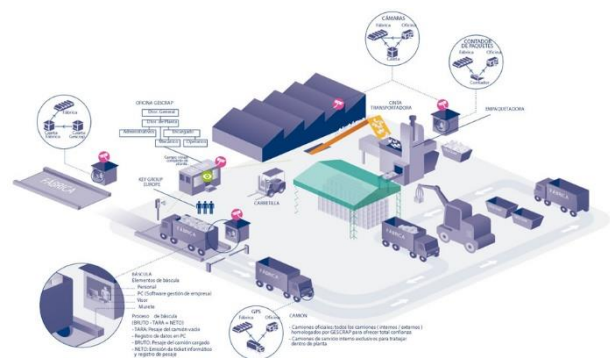


Figura 1. Control de Procesos.

La Automatización Industrial la encontramos en muchos sectores de la economía, como en la Fabricación de Alimentos, Productos Farmacéuticos, Productos Químicos, en la Industria Gráfica, Petrolera, Automotriz, Plásticos, Telecomunicaciones entre otros, sectores en los cuales generan grandes beneficios. No solo se aplica a máquinas o fabricación de productos, también se aplica la gestión de procesos, de servicios, a manejo

de la información, a mejorar cualquier proceso que con lleven a un desempeño más eficiente, desde la instalación, mantenimiento, diseño, contratación e incluso la comercialización.

Es por ello que *la importancia de la automatización industrial*, es reconocida hoy en día, en los ámbitos gerenciales y operativos dentro de la industria manufacturera moderna, sobre todo en la Comunidad Europea, como un elemento clave en el crecimiento y desarrollo de las naciones.

#### ❖ Bajo Costo

A pesar de que el costo de implementación de máquinas y software, puede resultar en primera instancia, el retorno de la inversión es rápido y seguro. Una vez la compañía ha recuperado la inversión realizada en automatización, los Empresarios percibe que los costes de producción se reducen en forma drástica, si se tiene en cuenta que un robot puede llegar a realizar la labor de varios trabajadores, sin afectar rubros de presupuesto tales como, seguridad social, vacaciones, bonificaciones, prestaciones sociales etc.

#### ❖ Seguridad Operativa

Una de las mayores preocupaciones de las grandes empresas, y que a la vez nos ayuda a entender la *importancia de la automatización industrial*, es la necesidad de mantener la excelencia en los procesos de producción, para asegurar productos certificados, sin incurrir en riesgos de accidentalidad laboral, que, todos los sabemos, son uno de los mayores costes que tienen las industrias del siglo XXI que no han sido objeto de procesos de automatización.



Figura 2. Sistema de Automatización Industrial.

Como proceso la automatización se compone de tres fases, el ingreso de datos, el procesamiento de dichos datos la salida y ejecución de los mismos.

En el ingreso de datos se encuentra que pueden ser operados por pulsadores, perillas, accionadores, así como las variables de entrada pueden ser de tipo físico como temperaturas, presión, humedad, etc. El manejo de estas variables permiten al operador o programador, generar una serie de características a un programa para que cumpla luego de ser analizadas las funciones requeridas por programador. Así por ejemplo a partir del ingreso a un sistema de un nivel de temperatura, el programador puede usar esta variable física como señal de

inicio para efectuar un proceso automático, como lo es la calefacción.

De acuerdo a lo anterior observamos como a partir de fenómenos naturales se puede llegar a dar solución a un problema dentro de la cotidianidad del hombre, es por ello que desde que existan dichas variables y al inventiva del hombre cualquier tipo de sistema, proceso o equipo puede ser fácilmente automatizado.

Durante el procesamiento de las entradas, y mediante el uso de tecnologías como la computación, los procesadores, etc., el hombre decodifica dichas entradas en lenguajes entendibles para la máquina, esto hace que haya un cambio de datos sin perder el fin al que se quiere llevar, así por ejemplo al recibir del exterior una temperatura  $x$  el sistema determina mediante programaciones anteriores que tipo de procedimiento debe seguir para ejecutar una acción.

Así pues si el procesador tiene como variable definida una temperatura de 20 grados como nivel de referencia, al ingresar una variable exterior, como por ejemplo 30° grados el procesador realizara a nivel interno una serie de acciones como comparar, ajustar, verificar ambas señales, una vez realizadas estas acciones determinara cual es la mejor solución, a partir de las necesidades del hombre. De esta manera el sistema de calefacción puede determinar que es momento de activar un sistema aparte que genere una similitud entre las variables.

Sin duda este momento es de vital importancia en cualquier sistema automatizado, pues de la preparación previa de la programación dependen en gran medida los resultados esperados.

Por último encontramos la salida de datos o ejecución de órdenes, después de haber realizado los procesos propios del análisis de datos de entrada y luego de encontrar una respuesta y solución al problema, el sistema mediante algún tipo de comunicación transmite sus conclusiones a sistemas generalmente de tipo mecánico, como lo son actuadores, motores, etc. A partir de este momento se evidencia en el espacio un cambio favorable a las condiciones iniciales de estado en el ambiente propuesto.

Sin duda dichos sistemas mecánicos que no son más que parte de la inventiva antigua del hombre, hacen posible después de una serie de pasos el poder contar con resultados eficientes a problemas cotidianos.



Figura 3. Sistema de Control Automático.

Este es en sí el principio y razón de ser de los sistemas automatizados, son sin lugar a dudas una de las grandes

invenciones del hombre, pues generan en todo sentido mayores beneficios, menos pérdidas de tiempo y grandes contribuciones hacia el desarrollo futurista de la humanidad.

## Sistemas de Control Escalables

El proyecto “Sistema elevador clasificador para paquetes” es un sistema de control escalable, éstos sistemas serán explicados.

Cuando la ISA (Sociedad para los Sistemas de Instrumentación y la Automatización, por sus iniciales en inglés) definió la norma ANSI/ISA-S88-01-1995 hace más de diez años, creó un lenguaje y unos modelos comunes para el diseño y la especificación de los sistemas de control para el procesamiento por lotes. Para los fabricantes de maquinaria, ya sometidos a las presiones habituales para que suministren máquinas más flexibles, más rápidamente y con menores costes, la norma presagiaba una notable reducción de la complejidad en el diseño de sistemas.

Conocida hoy en día simplemente como S88, la norma ANSI/ISA-S88.01-1995, su equivalente en la Comisión Internacional de Electrotécnica (IEC 61512-01) y las posteriores ampliaciones de la norma han allanado el camino para que los fabricantes de maquinaria puedan hacer un uso óptimo de los sistemas modulares con el fin de reducir los costes. Además, han eliminado la necesidad de escribir código personalizado para cada aplicación, reduciendo así los plazos de desarrollo, y han permitido la reutilización del código escrito para un sistema en sistemas posteriores, con independencia de la escala y la complejidad.

Por lo tanto, los fabricantes de maquinaria se benefician de una reducción de los plazos de comercialización gracias a los equipos estandarizados, las reutilizaciones eficaces del código, los controles escalables y las plataformas de información procedentes de redes de distribución de primera fila.

Creada en 1995, la norma trazó el camino a seguir en el desarrollo de sistemas de control por lotes (batch), definiendo modelos y terminología para la planta física, procedimientos y fórmulas de un proceso por lotes. La norma ha evolucionado con los años: En 2001 se publicó ANSI/ISA-88.00.02-2001, que trata sobre la estructura de los datos y las directrices para los lenguajes, y en 2004 vio la luz ANSI/ISA-88.00.03-2004, que trata de los modelos y representaciones de fórmulas generales y de plantas.

Además, de la norma S88 derivó la S95 para la integración de sistemas de control empresarial, que definía una terminología común para la descripción y comprensión de la información sobre fabricación en una empresa. También define el intercambio de información entre las funciones de control de fabricación y otras funciones de la empresa, incluidos los modelos de datos y las funciones de intercambio.

Entonces, ¿qué implica exactamente para los fabricantes de maquinaria trabajar para las industrias de transformación? El objetivo de los equipos de control por lotes para la industria de transformación es ser capaz de fabricar muchos productos con el mismo equipo, o llevar a cabo diferentes operaciones en el

mismo producto, sin necesidad de realizar grandes reconfiguraciones manuales del equipo de producción y sin tener que reescribir el software de control por lotes, una tarea potencialmente costosa y que lleva tiempo.

El modelo S88 de planta física define el equipo en función de celdas de proceso. Cada una de ellas contiene el equipo necesario para la producción de uno o más lotes.

Una celda de proceso se compone de todas las unidades: conjuntos de equipos de proceso, equipos de control y la lógica de control asociada que realiza las actividades de procesamiento.

Las capacidades básicas de procesamiento de la unidad pueden definirse mediante «módulos de equipo» y estos definen qué puede hacer una fórmula a la unidad, generalmente acciones como incorporaciones de material, transferencias de material, calentar o enfriar un recipiente, presurizar un recipiente y mezclar materiales en un recipiente. La norma S88 desglosa aún más los módulos de equipo en módulos de control, que son una combinación de la lógica básica de control y los diversos sensores y accionadores dentro de la unidad.

El efecto de todo ello es definir la gestión de las fórmulas jerárquicas y los marcos de segmentación de procesos que materializan el objetivo de separar los productos en fabricación de los equipos que los fabrican, y así permitir a los fabricantes disfrutar de importantes mejoras de la productividad.

Se puede usar el mismo equipo para fabricar múltiples productos o para realizar diferentes operaciones en un producto dado, con un desarrollo y despliegue sencillo de la fórmula y sin necesidad de contar con ingenieros especializados en sistemas de control.

Esto reduce de forma efectiva el coste total del diseño, desarrollo y suministro a través de una menor intervención de la ingeniería de proyectos (diseño, programación, cableado, ensayos, instalación, asistencia técnica), un ahorro de energía escalable y modular y sistemas abiertos de control e información de automatización.

El resultado es que los fabricantes de maquinaria pueden reducir espectacularmente los plazos de diseño de sistemas y suministrar soluciones de mayor capacidad y flexibilidad a los clientes con mayor rapidez. Al no tener que confiar en software y hardware personalizado, los costes del proceso de desarrollo se reducen drásticamente y las aplicaciones posteriores son más fáciles y rápidas de implantar.

En resumen, la S88 redefinió la forma en que se podían construir sistemas de control por lotes haciendo posible un enfoque modular del diseño de sistemas. Sin embargo, lo que de verdad ha hecho posible la materialización de los objetivos de la S88 es la llegada de herramientas informáticas estándar combinadas con equipos estándar, escalables y «llave en mano». Los fabricantes de maquinaria tienen por fin las herramientas que necesitan para construir sistemas de control por lotes de cualquier tamaño y nivel de complejidad para cualquier sector, de forma rápida y rentable.

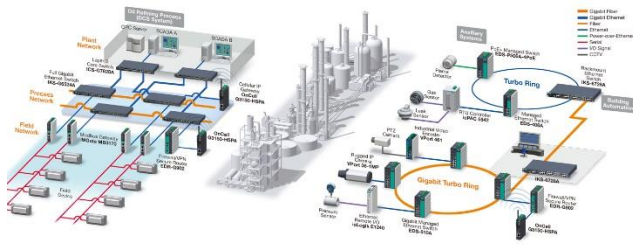


Figura 4. Sistema de control escalable o sistema de control distribuido (DSC).

## 2. Objetivo

Implementar un Sistema de Control Escalable con tecnologías Open Source basado en la arquitectura de los DCS, que permita interactuar en tiempo real con el sistema controlado.

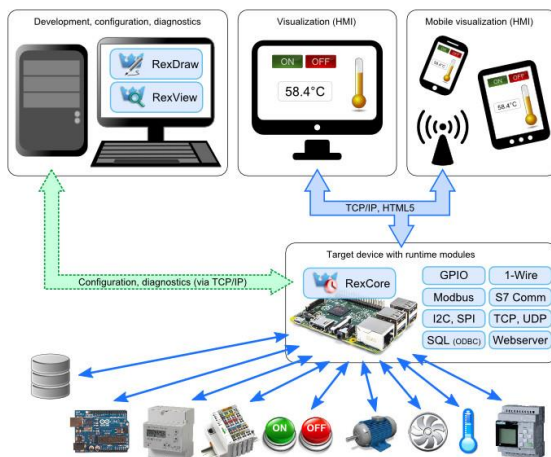


Figura 5. Sistema de control DCS. Implementación hecha para el proyecto.

### 2.1.Objetivos específicos

- ❖ Simular un entorno o proceso de control automático el cual estará monitoreado y controlado, este entorno estará adaptado a pruebas de error en la ejecución del sistema para probar el funcionamiento del SCS.
- ❖ Entender los diferentes tipos de protocolos permitidos para controlar dispositivos “IoT” y sensores.
- ❖ Aplicar las técnicas necesarias para linealizar los datos de los sensores a monitorear.
- ❖ Analizar las herramientas disponibles para diseñar el Dashboard.
- ❖ Interfazar todos los dispositivos que componen el SCS.
- ❖ Diseñar el Frontend o Panel de control.
- ❖ Publicar el SCS en la nube monitoreando el sistema.

## 3. Componentes del sistema

### 3.1.Componentes eléctricos y electrónicos

- ❖ Moto-reductores.
- ❖ Servo-motores.
- ❖ Puente H (L298N).

### 3.2.Sensores Análogos

- ❖ Sensor de fuerza resistivo (FSR402).

### 3.3.Dispositivos E/S de control para el sistema

- ❖ Microcontrolador PIC 18F2550.

### 3.4.Transmisores / Receptores

- ❖ ESP8266.

### 3.5.Componentes del SCS

- ❖ Plataforma de control: Raspberry Pi.
- ❖ Dashboard RT del SCS: Node-Red.

## 4. Especificación del Proyecto

Elevador clasificador para paquetes. El proceso se inicia con el transporte de uno de los paquetes de la báscula; una vez clasificado el paquete en la báscula, se encenderá una luz indicadora del tipo de paquete (luz 1 será paquete grande y luz 2 será paquete pequeño). A continuación, el paquete es transportado por la cinta 1 hasta el plano elevador. El cilindro C eleva los paquetes. Acto seguido, los paquetes son clasificados; los paquetes pequeños son colocados en la cinta 2 por el cilindro A, y los paquetes grandes son colocados en la cinta 3 por el cilindro B. El cilindro elevador C se recupera sólo cuando los cilindros A Y B llegan a la posición final. El sistema contará con botones locales de stop, start y reset; los cuales también aparecerán en la HMI, en donde se mostrará las fases del proceso y llevará la cuenta de los paquetes grandes y pequeños.

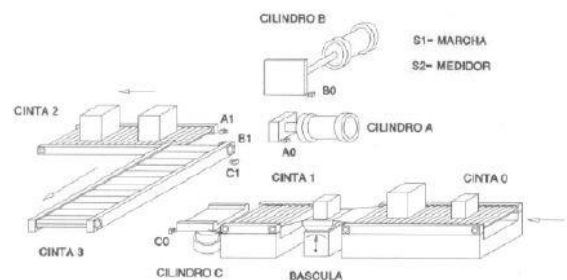


Figura 6. Sistema elevador clasificador para paquetes.



## 5. Parte 1: Construcción del Hardware del sistema

A continuación, se empezará a dar solución al Proyecto “Sistema elevador clasificador para paquetes”.

En primer lugar, se debió hacer el planeamiento de cómo iba a ser hecha la planta (arquitectura) del sistema. Luego de esto, se empezó a diseñar la idea que se tenía en ese entonces de utilizar madera Triplex como el principal material de la maqueta del proyecto.

En segundo lugar, debido a que después de que las cajas pasan de la banda 0 hacia la báscula de pesaje, nos dimos cuenta de que debíamos diseñar las bandas transportadoras con alturas diferentes para que las cajas cayeran de la primer banda hacia el sensor de peso. Entonces, se decidió que la primer banda debería ser levemente más alta que la segunda, esto debido a que necesitábamos que se creara un cambio de sectores, es decir, era necesario que el objeto transportado pasara de la cinta 0 al sensor de peso. Seguidamente, y después de tener la idea del material, era hora de empezar a construir, primeramente, las bandas transportadoras ya que estas nos podían determinar las dimensiones que tendría todo el sistema.



Figura 7. Medición del diseño de las bandas transportadoras.

Acto seguido, después de que se hicieron las dimensiones de la primer banda, se decidió hacer todas las bandas con las mismas dimensiones, para hacerlas más uniformes.



Figura 8. Construcción de las diferentes bandas transportadoras del sistema.

Llegado a este punto, y después de haber hecho el esqueleto de todas las bandas, ya era hora de decidir la altura que llevaran cada una de éstas, adicionar el soporte de la cinta, agregar el hule que es el que servirá para que el objeto sea transportado a través de toda la planta.

Con respecto al diseño de las cintas, se hizo la cinta 0 relativamente más alta que la báscula y la cinta 1, esto para que halla una transición de ésta al sensor. Luego, el sensor; después de sensar y determinar si el peso corresponde a una caja grande o pequeño; por medio de un servo-motor, envía una señal que hace que gire la plataforma en la que se encuentra en objeto a pesar para que éste pase a la siguiente banda.



Figura 9. Adición del soporte al igual que la adición del hule.



Figura 10. Adición del hule a las cintas transportadoras.



Figura 11. Diseño final de la construcción de la banda.

Luego de adicionar el soporte y el hule, se procedió a colocar el motor que servirá para que la banda gire (esto se hizo nuevamente a cada una de las cintas restantes).

Llegado a este punto, se empezó a diseñar el elevador que servirá para elevar los objetos a las demás bandas, las cuales será las correspondientes a las cajas pequeñas y grandes.



Figura 12. Diseño y construcción final del elevador de paquetes.

Por otra parte, luego de haber construido el elevador de paquetes era necesario la Implementación de lo que serán los cilindros que son los que clasificarán los paquetes, ya sean grandes o pequeños.



Figura 13. Acoplamiento de todos los sectores que componen el sistema.

Para terminar esta primer parte de la construcción de la maqueta del sistema era importante la Implementación del sistema de clasificación de paquetes. Para ello, se hizo uso de servomotes con acoples que permiten que al generar un PWM, éstos harán que un brazo actúe sobre las cajas haciendo que éstas se dirijan hacia la cinta A o la cinta B.

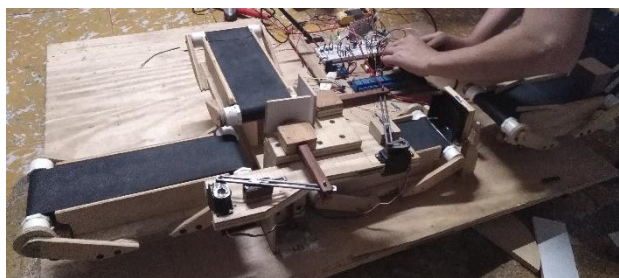


Figura 14. Sistema clasificador para paquete implementado por medio de servo-motores y acoplamiento de los brazos clasificadores.

## 6. Parte 2: Implementación electrónica y software de control del proceso

Luego de concluir con la construcción de la parte del hardware del sistema de elevación clasificación para paquetes, era hora de empezar a diseñar, programar e implementar la parte de la

electrónica y el software que controlarán todos los procesos que se requieran en el proyecto.

Primeramente, se decidió usar la Raspberry Pi 3, como servidor en el cual se enviarán todos los datos correspondientes al monitoreo y control del proceso que se llevará a cabo en el sistema.

Luego de esto, se implementó un módulo WIFI ESP8266 el cual se comunica con la Raspberry inalámbricamente, enviando los datos acerca del pesaje, clasificación; control manual de inicio, parada y reinicio; y visualización de la planta en tiempo real.

Llegado a este punto, se explicará el código empleado en el módulo WIFI ESP8266; pero antes de eso, se debe aclarar que el módulo trabaja por medio de comandos AT, los comandos AT son instrucciones codificadas que conforman un lenguaje de comunicación entre el hombre y un terminal modem. Ahora que se sabe qué es un comando AT, se debe decir que no se trabajó con este lenguaje, así que se decidió cambiar el FIRMWARE para poder hacer uso de un lenguaje mucho más completo como el C.

A continuación, se explicará el funcionamiento del código dentro del sistema:

Conexión de módulo Wi-Fi ESP8266 con servidor Mosquitto bajo el protocolo MQTT (M2M). Este código, es una modificación al ejemplo de conexión con el servidor MQTT de Arduino el cual utiliza la librerías ESP8266Wifi y PubSubClient, fue modificado y adaptado para el control de nuestro proceso, se conecta a un servidor de MQTT el cual se encuentra en una red alcanzable vía TCP/IP.

Funcionamiento:

Al conectarse con el servidor MQTT, imprime el mensaje "Hola mundo" indicando que se ha conectado correctamente, esto lo realiza con el topic "outopic", este hola mundo es utilizado sobre Node-Red para detectar que la planta se ha conectado con el servidor de aplicación.

- ❖ Se suscribe a un topic "inTopic" por el cual recibe los eventos desde el mismo servidor MQTT y son enviados por serial al PIC18f2550.
- ❖ Si el primer carácter del topic de entrada es un 1, enciende el LED integrado sobre el mismo ESP8266.
- ❖ Este se reconecta al servidor si en algún momento falla la comunicación.

Envía y recibe datos por el puerto serial el cual van y vienen del microcontrolador PIC18f4550. Para instalar la board ESP8266 sobre Arduino 1.6.4 o mayor agregue el siguiente al administrador de Boards de Arduino sobre Archivo, Preferencias, Administrador de boards adicionales

[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)

Despues habra "Herramientas > Board > Administrador de Boards", busque e instale el paquete ESP8266.

Para seleccionar el modulo, clic en Tool > Board.

Librerias:

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
```

Actualizar estos valores con los datos de la red Wifi y el servidor de aplicacion MQTT.

```
const char* ssid = "SSIDDeLaRed";
const char* password = "ClaveDeLaRedWifi";
const char* mqtt_server =
"DireccionIPoNombredelDominioodelServidor";
```

```
WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;
```

Inicia el serial, el wifi y la conexion con el servidor mqtt por el puerto 1883.

```
void setup () {
```

Inicializa el BUILTIN\_LED pin Como una salida

```
pinMode(BUILTIN_LED, OUTPUT);
Serial.begin(9600);
setup_wifi();
client.setServer(mqtt_server, 1883);
client.setCallback(callback);
}
```

Función que realiza la conexion a la red Wifi.

Imprime por el serial un. mientras intenta realizar la conexion con la red Wifi. Una vez conectado imprime por serial los datos de la conexion realizada y la ip asignada.

```
void setup_wifi() {

  delay(10);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}
```

Funcion que recoge el evento de llegada recogido por el servidor mqtt, lo almacena en buffer y lo imprime por serial.

```
void callback(char* topic, byte* payload, unsigned int
length) {
  for (int i = 0; i < length; i++) {
```

```
    Serial.print((char)payload[i]);
  }
```

Enciende el BUILTIN\_LED si se recibe un 1 sobre el primer espacio del buffer de datos recibido.

```
if ((char)payload[0] == '1') {
```

Enciende el led con logica 0

```
digitalWrite(BUILTIN_LED, LOW);

} else {
```

Apaga el led con logica 1

```
digitalWrite(BUILTIN_LED, HIGH);
}
}
```

```
void reconnect() {
```

Loop de reconexoin con el servidor MQTT.

```
while (!client.connected()) {
  Serial.print("Attempting MQTT connection...");
```

Intento de conexión.

```
if (client.connect("ESP8266Client")) {
  Serial.println("connected");
```

Una vez conectado, realiza una publicacion

```
client.publish("outTopic", "hello world");
```

Se suscribe a un topic de entrada.

```
client.subscribe("inTopic");
} else {
  Serial.print("failed, rc=");
  Serial.print(client.state());
  Serial.println("try again in 5 seconds");
```

Espera 5 segundos antes de intentar reconectar de nuevo.

```
delay(5000);
}
}
```

```
void loop() {
```

```
if (!client.connected()) {
  reconnect();
}
client.loop();
```

```
long now = millis();
if (now - lastMsg > 500) {
  lastMsg = now;
  ++value;
```

Buffer que recoge los datos leídos por el puerto serial y lo publica sobre el topic de salida.

```
char myArray[5];
Serial.readBytes(myArray,6);
client.publish("outTopic", myArray);
```



```
}
}
```

Luego de que el código del ESP8266 fuese completado, era momento de crear el código responsable de controlar todo el proceso de la planta, el cual fue implementado en el PIC 18F2550, al cual se le adecuaron las salidas para, el control del motor de la primer banda, el control de los otros motores para la tres bandas restantes (los cuales fueron colocados en paralelo para que funcionaran al mismo tiempo), el PWM del servo-motor que se encuentra en la plataforma del sensor y 2 PWMs que controlan los servos que se encuentran en la parte de la clasificación de los paquetes; también las conexiones para los botones de *star*, *stop* y *reset*; los LEDs indicadores del tipo de objeto que fue pesado (grande o pequeño), el control del sensor de peso y adecuación de la señal del mismo (ref. FSR 402) y por último la conexión con el ESP8266.

En este punto, se hará la explicación del código del PIC 18F2550:

Convertidor de Analogo a Digital:

```
#include <18F2550.h>
```

```
#fuses xt, nowdt, noprotect, put, CPUDIV1
```

Numero de bits a utilizar sobre el convertidor analogo:

```
#device adc = 10
#use delay(internal = 8000000)
#use rs232(baud = 9600, bits = 8, parity = N, xmit =
PIN_C6, RCV = PIN_C7, stop = 1, ERRORS)
```

```
#use fast_io(b)
#use fast_io(a)
#use fast_io(c)
```

Variables Globales:

Cálculo de la resolución en la salida:

```
float resolucion = (float) 5 / (float) 1023;
char dato_recibido = '0';
```

Se usa en los bucles:

```
int i;
```

Se calcula el promedio del peso:

```
float promedio = 0.0f;
```

```
boolean primera_etapa = false;
boolean segunda_etapa = false;
```

Se define el puerto ADC a utilizar y sus pines:

```
void define_puertos() {
```

Define el pin 7 y 0 como entrada y el resto del puerto como salidas. 3 y 4 leds indicadores de caja, 0 entrada finales de Carrera y 2 banda 1:

bin: 1110 0001.

```
set_tris_b(0xE1);
output_low(PIN_B1);
```

Define la entrada 0 del puerto A como entrada:  
bin: 0000 0001.

```
set_tris_a(0x01);
output_high(PIN_A5);
output_low(PIN_A3);
```

Se define el puerto c como salida:

```
set_tris_c(0x00);
output_c(0x00);
```

Define como entradas analoga el puerto AN0:

```
SETUP_ADC_PORTS(AN0);
```

Define el reloj interno para el uso del ADC:

```
SETUP_ADC(ADC_CLOCK_INTERNAL);
}
```

```
void parar_planta(){
    output_high(PIN_B1);
```

Detiene lo que esta funcionando:

```
// if(primer_etapa){
```

Se detiene la banda 1:

```
output_low(PIN_B2);
//} else {
```

Se detienen las bandas finales:

```
output_high(PIN_A5);
}
```

```
while(!input(PIN_B5)) {
    delay_ms(100);
}
```

```
if(primer_etapa){
```

Se reanuda la primera banda:

```
output_high(PIN_B2);
}else{
```

Se reanuda las bandas finales:

```
if(input(PIN_B0)){
```

Esta en las bandas:

```
output_low(PIN_A5);
delay_ms(3500);
output_high(PIN_A5);
}
```

```
output_low(PIN_B1);
}
```



Interupciones.

Interrupción de cambio en el puerto b:

```
#int_rb
void int_rb_pulsadores(){
```

Botón de *Reset*:

```
    if(input(PIN_B7)){
        printf("7.0000");
    }
```

Botón de *Stop*:

```
    if(input(PIN_B6)){
        parar_planta();
    }
}
```

Interrupcion de llegada de datos:

```
#int_rda
void rb_isr() {
    if (kbhit()) {
        dato_recibido = getc();
        delay_ms(4);
        if(dato_recibido == '1'){
            parar_planta();
        }
        dato_recibido = '0';
    }
}

float leerADC () {
    return resolucion * (float) (READ_ADC ());
}

void subir_ascensor () {
```

Subir ascensor:

```
    output_high(PIN_A1);
    output_low(PIN_A2);
```

Inicio de la subida con fuerza:

```
    for (i = 0; i < 40; i++) {
        output_high(PIN_A3);
        delay_ms (9);
        output_low(PIN_A3);
        delay_ms (11);
    }

    if (promedio >= 3.5) {
```

Cajas pesadas:

```
        while (!input(PIN_B0)) {
            output_high(PIN_A3);
            delay_ms (8);
            output_low(PIN_A3);
            delay_ms (12);
        }
    } else {
```

Cajas livianas:

```
        while (!input(PIN_B0)) {
            output_high(PIN_A3);
            delay_ms (6);
```

```
            output_low(PIN_A3);
            delay_ms (14);
        }
    }
}
```

```
void bajar_ascensor() {
```

Bajar ascensor:

```
    output_low(PIN_A1);
    output_high(PIN_A2);

    for (i = 0; i < 40; i++) {
        output_high(PIN_A3);
        delay_ms (12);
        output_low(PIN_A3);
        delay_ms (8);
    }

    for (i = 0; i < 80; i++) {
        output_high(PIN_A3);
        delay_ms (5);
        output_low(PIN_A3);
        delay_ms (15);
    }

    while (!input(PIN_B0)) {
        output_high(PIN_A3);
        delay_ms (3);
        output_low(PIN_A3);
        delay_ms (17);
    }
}
```

```
void main () {
```

Habilita las interrupciones por serial:

```
    enable_interrupts(int_rda);
```

Habilita las interrupciones puerto b:

```
    enable_interrupts(int_rb);
```

```
    enable_interrupts(global);
```

Flanco de subida:

```
    ext_int_edge(L_TO_H);
```

Configura los puertos:

```
    define_puertos ();
```

Se configura el canal del adc:

```
    SET_ADC_CHANNEL (0);

    while (true) {

        primera_etapa = true;
```

Inicia la banda 1:

```
        output_high(PIN_B2);
```

Se apagan los led indicadores de caja:

```
        output_low(PIN_B3);
```

```
output_low(PIN_B4);
```

Esta en este metodo hasta que pese algo:

```
while (true) {  
    delay_ms (200);  
    if (leerADC () > 1.0f) {  
        break;  
    }  
}
```

Se detiene la banda 1:

```
output_low(PIN_B2);
```

Espera 3 segundos mientras cuadra la caja:

```
delay_ms (3000);
```

Peso de la caja:

```
for (i = 0; i < 7; i++) {  
    delay_ms (300);  
    promedio += leerADC ();  
}  
promedio /= 7;
```

Se envía el peso por r232:

```
printf ("%1.4f", promedio);
```

Enciende los led indicadores:

```
if (promedio >= 3.5) {  
    output_high(PIN_B3);  
} else {  
    output_high(PIN_B4);  
}
```

Servo 1 (pasa a la otra banda):

```
delay_ms (500);
```

Brazo 90 grados:

```
for (i = 0; i < 15; i++) {  
    output_high(PIN_C2);  
    delay_us (1500);  
    output_low(PIN_C2);  
    delay_us (18500);  
}  
  
for (i = 0; i < 15; i++) {  
    output_high(PIN_C2);  
    delay_us (500);  
    output_low(PIN_C2);  
    delay_us (19500);  
}
```

Fin del servo 1:

Cambia los estados de las etapas:

```
primera_etapa = false;  
segunda_etapa = true;
```

Corre las 3 bandas del rele:

```
output_low(PIN_A5);  
delay_ms (3500);  
output_high(PIN_A5);  
subir_ascensor();  
  
delay_ms (500);
```

Condiciona para pesar las cajas:

```
if (promedio >= 3.5) {
```

Brazo de las cajas pesadas:

```
for (i = 0; i < 60; i++) {  
    output_high(PIN_C1);  
    delay_us (500);  
    output_low(PIN_C1);  
    delay_us (19500);  
}  
  
for (i = 0; i < 60; i++) {  
    output_high(PIN_C1);  
    delay_us (2500);  
    output_low(PIN_C1);  
    delay_us (17500);  
}  
}  
} else {
```

Brazo de las cajas livianas:

```
for (i = 0; i < 60; i++) {  
    output_high(PIN_C0);  
    delay_us (500);  
    output_low(PIN_C0);  
    delay_us (19500);  
}  
  
for (i = 0; i < 60; i++) {  
    output_high(PIN_C0);  
    delay_us (2500);  
    output_low(PIN_C0);  
    delay_us (17500);  
}  
}  
}
```

Corre las 3 bandas del rele:

```
output_low(PIN_A5);  
delay_ms (3500);  
output_high(PIN_A5);  
  
bajar_ascensor();  
  
segunda_etapa = false;  
}  
}
```

## 6.1. Parte 2: Implementación de la interfaz Gráfica

La interfaz gráfica fue diseñada por medio de la herramienta de programación Node-Red (<https://nodered.org/>), la cual es una herramienta de programación para conectar dispositivos de hardware, API y servicios en línea de maneras nuevas e interesantes.

Proporciona un editor basado en navegador que facilita el cableado de flujos utilizando la amplia gama de nodos de la

paleta que se puede implementar en su tiempo de ejecución con un solo clic.

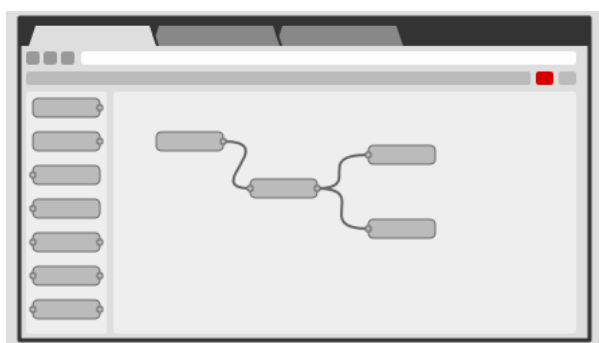


Figura 15. Flow de la herramienta Node-Red.

Node-Red funciona por medio de bloques interconectados entre ellos, los cuales, tienen funciones específicas que permiten que el desarrollo sea más interactivo y más ordenado.

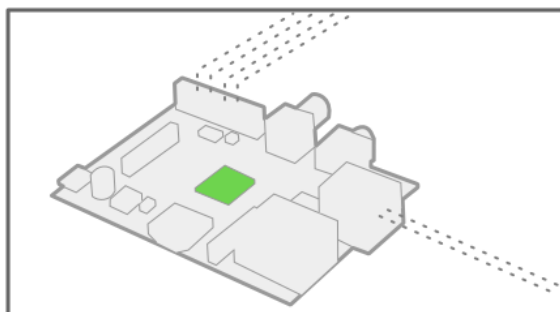


Figura 16. Implementación del Node-Red en la Raspberry Pi.

Como se mencionó anteriormente, la ventaja de Node-Red es que permite un fácil manejo del mismo, permitiendo que se pueda acceder a éste por medio de la implementación de una Raspberry Pi. Pudiendo llevarlo a todos lados.

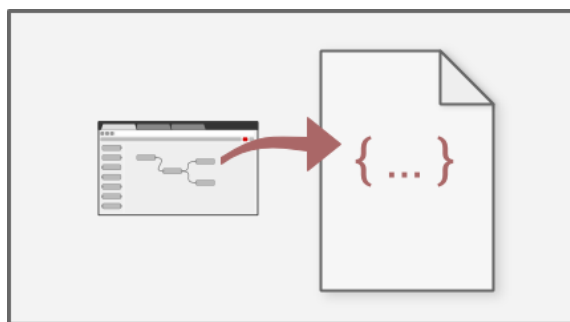


Figura 17. Los flows que componen el sistema creado en Node-Red son almacenados usando un lenguaje de programación llamado JSON.

Los flows de Node-Red (los bloques a la izquierda en la Figura 17) son programados en JavaScript, el cual, es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario; cada bloque se programa con este lenguaje pero el resultado final (el almacenamiento de todos estos flows) es un archivo .JSON el cual almacena toda la programación de la página. El código

JSON de la interfaz creada será referenciado en la sección anexos, la cual, tendrá un enlace para que se pueda observar todo el código que compone la interfaz gráfica.

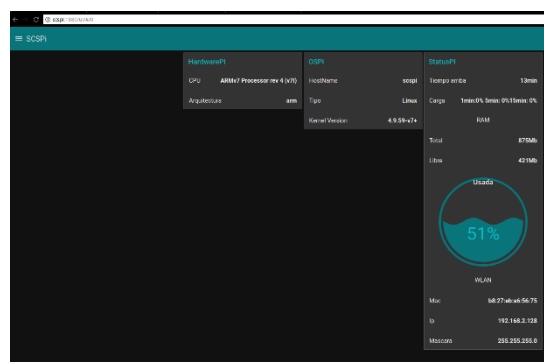


Figura 18. Dashboard de la Interfaz gráfica.

En la figura 18, se observa parte de la interfaz, en esta ventana se observan la arquitectura del sistema, al igual que las características de la Raspberry y el consumo de memoria.

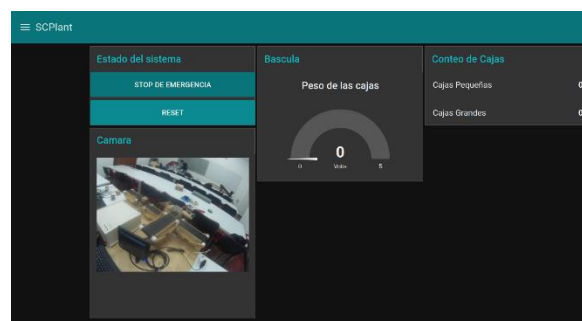


Figura 19. Visualización del sistema, el peso, número de caja y botones de stop y reset.

En esta figura se observa la planta, la cual esta siendo transmitida en tiempo real por medio de una camara de video. Se puede ver la simulación del sensor por peso y el número de cajas que han pasado por el proceso de la planta.

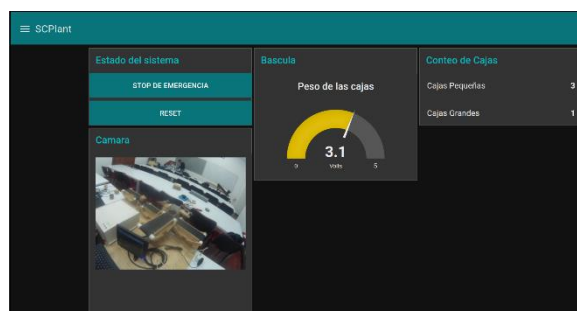


Figura 20. Visualización de la carga en el sensor.

En la figura 20 se puede apreciar que, al aplicar una fuerza sobre el sensor, este; en tiempo real, permite visualizar en tiempo real el voltaje del peso al cual se encuentra en el momento del pesa y determina si el voltaje corresponde a un paquete grande o pequeño.

## 7. Funcionamiento del sistema

Llegado a este punto, el funcionamiento del sistema se basa en el acoplamiento de la señal de información a través de todos los componentes de control.



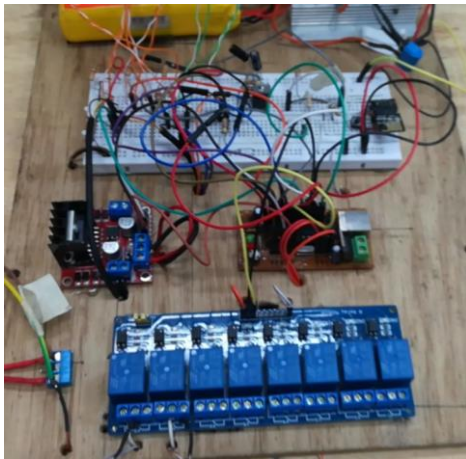


Figura 21. Sistema de control de la planta.

Primeramente, al iniciar el proceso, la banda 0 comienza a moverse, por lo cual, se procede a colocar un paquete (en este caso una caja) que pase a través de ella. Luego, la caja cae en el sensor, al caer en el sensor la primera banda se detiene y el sensor comienza a sensar el peso que resiste (en este caso el sensor presenta una variación en la resistencia), después del sensado, un servo-motor (por medio de un PWM) hace girar la plataforma de la báscula para que la caja pase del sensor a la cinta transportadora siguiente. Llegado a este punto, y luego de que la caja pasa de la segunda banda hacia el elevador, el puente H (L298N) es habilitado por medio de un pulso que se envía del PIC hacia éste, al hacer eso, el puente H alimenta al motor el cual, con un tornillo sin fin acoplado al rotor, empieza a girar y la plataforma (acoplada con una tuerca) empieza a subir, después de haber terminado el proceso de elevación, son activados cualquiera de los dos servo-motores que se encuentran en la parte de la clasificación; ya determinado por el PIC; el servo-motor al cual le corresponda el tipo de paquete (grande o pequeño) se activa y por medio de un brazo empuja al objeto hacia la banda correspondiente.

En medio de la explicación anterior, la cual es una explicación muy generalizada del funcionamiento del proceso, se procede a explicar la implementación de la conexión inalámbrica. En medio de la actividad de sensado del sensor, éste envía una señal hacia el módulo el cual, al estar conectado por medio inalámbrico a la Raspberry; y esta a su vez ejecutando el archivo .JSON el cual corresponde a la interfaz gráfica, es visualizado cuánto voltaje es con el que cuenta el objeto sensado. En medio de esto, el PIC con ayuda del código implementado determina si la caída de voltaje que se genera en la variación de la resistencia del sensor corresponde a una caja grande o una pequeña.

Para terminar, todo este proceso siempre es monitoreado por medio de una cámara que transmite en tiempo real sobre la interfaz (figura 19), con el cual se puede tener en cuenta cualquier problema que pueda surgir en medio del proceso.

## 8. Conclusiones

- ❖ Al haber trabajado con sistemas embebidos como la Raspberry, y también con el módulo WIFI nos dimos cuenta que el Internet de las Cosas es algo que actualmente ya está siendo implementado en todos los sistemas de automatización, esto, para proveer un

constante monitoreo sobre los procesos que se llevan a cabo, además de ser una herramienta de mucha utilidad, por el motivo de que todas las personas responsables del proceso son capaces de visualizar los procesos, controles, monitoreos, mediciones, cálculos, etc. En casi todo el mundo. Ofrece un servicio más sofisticado, más confiable y de mejor calidad.

- ❖ El control de procesos consiste en mediciones realizadas para controlar las variables críticas del proceso asociadas a la entrada, transformación y salida, el control de procesos es totalmente necesario para el buen manejo de una organización. El control es un mecanismo que permite corregir desviaciones a través de indicadores cualitativos y cuantitativos dentro de un contexto social amplio, a fin de lograr el cumplimiento de los objetivos claves para el éxito organizacional, es decir, el control se entiende no como un proceso netamente técnico de seguimiento, sino también como un proceso informal donde se evalúan factores culturales, organizativos, humanos y grupales, es un esfuerzo sistemático para establecer normas de desempeño con objetivos de planificación, para diseñar sistemas de retroinformación, para comparar los resultados reales con las normas previamente establecidas, para determinar si existen desviaciones y para medir su importancia, así como para tomar aquellas medidas que se necesiten para garantizar que todos los recursos de la empresa se usen de la manera más eficaz y eficiente posible para alcanzar los objetivos de la empresa.

## Referencias

- [1] <https://www.raspberrypi.org/>  
(Raspberry Pi: Sistema embebido en un solo Chip con capacidades de MultiProceso).
- [2] <https://www.raspberrypi.org/documentation/>  
(Documentación acerca de la Raspberry)
- [3] [https://cdn-shop.adafruit.com/product-files/2471/0A-ESP8266\\_Datasheet\\_EN\\_v4.3.pdf](https://cdn-shop.adafruit.com/product-files/2471/0A-ESP8266_Datasheet_EN_v4.3.pdf)  
(ESP8266: Módulo de comunicación Inalámbrica 802.11 b/g/n, DataSheet)
- [4] <http://ww1.microchip.com/downloads/en/DeviceDoc/39632e.pdf>  
(PIC18F2550: MicroControlador de la familia MicroChip, DataSheet)
- [5] <https://www.raspbian.org/>  
(Raspbian: Sistema Operativo Basado en Tecnologías GNU/Linux)
- [6] <https://www.raspbian.org/RaspbianDocumentation>  
(Documentación Raspbian).
- [7] <https://nodered.org/>  
(Node-RED: Herramienta de programación basada en Flows, editor de flujos con tecnologías basadas en APIs para conectar dispositivos de Hardware del Internet de las Cosas "IoT")
- [8] <https://nodered.org/docs/>

(Documentación acerca de Node-Red).

[9] <https://mosquitto.org/>

(Mosquitto: Servidor de aplicacion intermediario para los mensajes bajo el protocolo MQTT)

[10] <https://mosquitto.org/documentation/>

(Documentación Mosquito).

[11] <http://mqtt.org/>

(MQTT: Protocolo de comunicacion Maquina a Maquina (M2M)).

[12] <http://mqtt.org/documentation>

(Documentación protocolo MQTT).

[13] [https://github.com/knolleary/pubsubclient/tree/master/examples/mqtt\\_esp8266](https://github.com/knolleary/pubsubclient/tree/master/examples/mqtt_esp8266)

(Ejemplo de uso del protocolo MQTT).

[14] <https://www.arduino.cc/>

(Arduino: IDE de programacion para Chips basados en la arquitectura Atmel AVR).

[15] <https://www.arduino.cc/en/Guide/HomePage>

(Documentación Arduino).

[16] <http://arduino-esp8266.readthedocs.io/en/latest/>

(Referencia a la programacion del chip ESP8266 con Arduino).

[17] <https://github.com/esp8266/Arduino/tree/master/libraries/ESP8266WiFi>

(Libreria ESP8266Wifi).

[18] <https://github.com/knolleary/pubsubclient>

(Libreria PubSubClient, Cliente MQTT para el chip ESP8266).

[19] <http://www.ccsinfo.com/>

(Pic C compiler: IDE utilizado para programar Chips de la arquitectura MicroChip).

[20] [https://www.ccsinfo.com/downloads/ccs\\_c\\_manual.pdf](https://www.ccsinfo.com/downloads/ccs_c_manual.pdf)

(Documentación PIC C Compiler).

[21] <https://www.javascript.com/>

(Java Script: Lenguaje de Programacion Orientado a Objetos utilizado para programar sobre Node-Red).

[22] <https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia>

(Documentación sobre JavaScript).

[23] <http://estampacionesjom.com/importancia-de-la-automatizacion-industrial-dentro-del-panorama-actual/>

(Importancia De La Automatización Industrial)

[24] <http://www.redjbm.com/catedra/index.php/tecnologia/68-la-importancia-de-la-automatizacion>

(Importancia De La Automatización Industrial).

[25] <https://www.gruposothis.com/soluciones/automatizacion-control-procesos/>

(Control De Procesos Automatizados).

[26] <http://www.interempresas.net/Robotica/Articulos/18174-Soluciones-escalables-para-el-diseno-de-sistemas-de-control-por-lotes.html>

(Sistemas De Control Escalables).

[27] <https://www.br-automation.com/es/productos/software/automation-runtime/escalabilidad-completa-del-sistema-de-control-visualizacion-y-accionamiento/>

(Escalabilidad completa del sistemade control).

[28] <http://w3.siemens.com/mcms/automation/es/process-control-systems/pages/default.aspx>

(Sistemas de control de procesos).

[29] <https://www.opiron.com/2017/06/15/escalabilidad-en-automatizacion-industrial/>

(Escalabilidad en la automatización industrial).

[30] <https://es.slideshare.net/alleonchile/sistemas-de-control-distribuido-dcs-7298975>

(Sistemas de control distribuido).

## Anexos

[1] Enlace en el cual se podrá observar el archivo .JSON el cual engloba toda la programación de la interfaz gráfica.

<https://github.com/UavLabsColombia/ScsPi/blob/master/Scr/Flow-NodeRed/SCSPiDashboard.json>