

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA ELECTRÓNICA**

**Trabajo de titulación previo a la obtención del título de: INGENIERO
ELECTRÓNICO**

**TEMA:
DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL EN RED
(NCS), UTILIZANDO SENSORES Y ACTUADORES INTELIGENTES PARA
LOS MECANISMOS DE CONTROL: PID, PWM Y CONTROL ON/OFF,
MONITOREADOS MEDIANTE UNA APLICACIÓN POR LA NUBE.**

**AUTOR:
PAÚL AGUSTIN BONILLA AGUALONGO**

**TUTOR:
CARLOS GERMÁN PILLAJO ANGOS**

Quito, marzo del 2016

Cesión de derechos de autor

Yo Paúl Agustín Bonilla Agualongo, con documento de identificación N° 1720669264, manifiesto mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del trabajo de titulación intitulado: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL EN RED, UTILIZANDO SENSORES Y ACTUADORES INTELIGENTES PARA LOS MECANISMOS DE CONTROL: PID, PWM Y CONTROL ON/OFF, MONITOREADOS MEDIANTE UNA APLICACIÓN POR LA NUBE, mismo que ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



Nombre: Paúl Agustín Bonilla

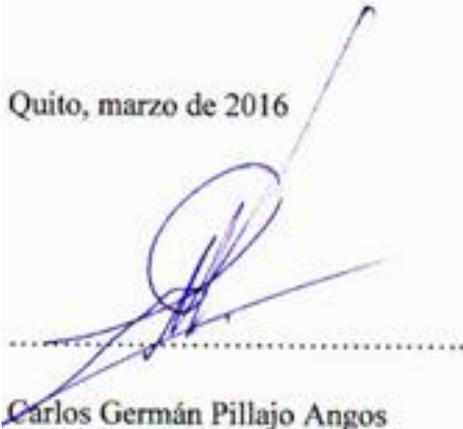
Cédula: 1720669264

Fecha: marzo de 2016

Declaratoria de coautoría del docente tutor

Yo, declaro que bajo mi dirección y asesoría fue desarrollado el trabajo de titulación: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL EN RED, UTILIZANDO SENSORES Y ACTUADORES INTELIGENTES PARA LOS MECANISMOS DE CONTROL: PID, PWM Y CONTROL ON/OFF, MONITOREADOS MEDIANTE UNA APLICACIÓN POR LA NUBE, realizado por Paúl Agustín Bonilla Agualongo, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana para ser considerados como trabajo final de titulación.

Quito, marzo de 2016



Carlos Germán Pillajo Angos

Cédula: 1709255119

DEDICATORIA

A Dios, a mis padres por apoyarme y guiarme en todos los aspectos de mi vida, por su amor y paciencia en los difíciles momentos.

Son el todo de mi vida

AGRADECIMIENTO

A la Universidad Politécnica Salesiana por abrirnos sus puertas y formarnos para los retos del mañana, además al Ing. Carlos Pillajo por su guía y apoyo en desarrollo del presente trabajo.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1	2
ANTECEDENTES	2
1.1 Título del proyecto.	2
1.2 Planteamiento del problema.	2
1.3. Objetivos.	2
1.3.1. Objetivo General.....	2
1.3.2. Objetivos específicos.	3
1.4. Alcance.	3
1.5. Justificación.....	4
CAPÍTULO 2	5
MARCO TEÓRICO	5
2.1. NCS	5
2.2. Control PID.	6
2.3. Control ON/OFF.....	7
2.4. Control PWM.	7
2.5. Redes de información.....	8
2.5.1. Modelo TCP/IP	9
2.5.2. DHCP.....	9
2.5.3. Router.....	9
2.6. Sensores y actuadores inteligentes.	10
2.7. Arduino.....	10
2.7.1. Módulo ethernet Arduino.....	10
2.7.2. Dragino Yun.	11

2.8. Sensor LM35.....	11
2.9. Sensor Sharp GP2Y0A41SK0F.....	11
2.10. HTML.....	11
2.11 Análisis de la situación actual.	12
CAPÍTULO 3.....	15
DISEÑO E IMPLEMENTACIÓN	15
3.1. Definición del Problema.....	15
3.2. Características del sistema.....	16
3.2.1. Costo.	16
3.2.2. Cantidad de módulos.	17
3.2.3. Volumen.....	17
3.2.4. Tiempo de respuesta de la red.....	17
3.2.5. Error.	17
3.3. Análisis Funcional.	18
3.3.1. Raspberry PI	18
3.3.2. Comparación en costos.	19
3.3.3. Comparación en tiempos de respuesta de la red.	19
3.3.4. Comparación en error.	20
3.4. Diseño y configuración de la red.....	20
3.4.1. Configuración del router.....	20
3.4.2. Direcciones IP de la red.	21
3.4.3. Configuración DHCP.....	21
3.5. Descripción de los Procesos de control.	23
3.5.1. Primera planta de control se simula el proceso Ball & Beam(PID).	24
3.5.2. Sensor inteligente de distancia Ethernet.	25

3.5.3. Actuator inteligente wi-fi.....	25
3.6. Segunda planta de control se simula un proceso de industria lanera, en específico de tinturado (PWM y control ON/OFF).....	28
3.6.1. Sensor-Actuator.	30
3.6.2. PWM en Arduino extra para la compensación de retardos.	32
3.7. Central de control.	32
3.8. Arduino Ethernet en la Central de control.....	32
3.8.1. Páginas Web.	33
3.9. Algoritmos de control del proceso de tintura.	36
3.10. Control del proceso PID en la red.	38
3.10.1. Algoritmos del proceso PID en la red.....	38
3.11. Dragino Yun como receptor de datos.....	44
3.11.1. Open Energy Monitor	44
3.11.2. Programación en Emoncms.	46
3.11.3 Dragino Yun como cliente de Emoncms.	47
CAPÍTULO 4.....	52
PRUEBAS Y RESULTADOS	52
4.1. NCS como servicio.....	52
4.2. Sistema de control PWM y Control On/Off en red.	53
4.3. Sistema de control PID en red.	55
4.3.1. Valores de las funciones lineales.	57
4.3.2. Coeficientes del algoritmo de control PID.	60
4.3.3. Emoncms con control PID.....	63
CONCLUSIONES.....	65
RECOMENDACIONES.....	67

REFERENCIAS68

ANEXOS69

ÍNDICE DE TABLAS

Tabla 1. Voz del ingeniero y usuario	16
Tabla 2. Diferencias entre Arduino y Raspberry PI.....	18
Tabla 3. Costos de Arduino Ethernet y wifi.....	19
Tabla 4. Router TP-WR740N	20
Tabla 5. Datos de los tipos de tintura.....	30
Tabla 6. Características de los dispositivos en la red.....	50
Tabla 7. Conexión entre dispositivos.....	51
Tabla 8. Valores del PID en la red.....	58
Tabla 9. Tiempo de estabilización del proceso PID.....	63

ÍNDICE DE FIGURAS

Figura 1. Sistemas de control con medio compartido	5
Figura 2. Modulación por ancho de pulso.....	8
Figura 3. Clasificación de la distribución de las acciones en función del instante de su activación.	13
Figura 4. Muestreo basado por tiempo y eventos	14
Figura 5. Diagrama de NCS.....	15
Figura 6. Configuración de Red de área local.....	21
Figura 7. Configuración de DHCP.....	22
Figura 8. Configuración de reservación de direcciones	23
Figura 9. Ball & Beam	24
Figura 10. Parámetros de red en Arduino	25
Figura 11. Configuración de Dragino Yun	26
Figura 12. Configuración de la red de trabajo en Dragino Yun.....	27
Figura 13. Gráfica de tintura de la lana.....	29
Figura 14. Parámetros de red de Arduino Ethernet en la Central de control.	33
Figura 15. Página principal en la dirección IP 192.168.0.155	34
Figura 16. Segunda página en la dirección IP 192.168.0.155:81.....	35
Figura 17. Tercera página en la dirección IP 192.168.0.155:9112	35
Figura 18. Cuarta página en la dirección IP 192.168.0.155:9113.....	36
Figura 19. Flujograma del Proceso de tintura	37
Figura 20. Diagrama de fuerzas en Ball & Beam	39
Figura 21. Función lineal generada por eventos	40
Figura 22. Programación de funciones lineales generadas por eventos.....	41
Figura 23. Programación de algoritmo PID en Arduino	42
Figura 24. Flujograma del proceso PID	43
Figura 25. OEM ofrece multiples sistemas de comunicación.....	44
Figura 26. Pantalla principal de Emoncms	45
Figura 27. Configuración de las características gráficas de Emoncms	46
Figura 28. Visualizacion de datos de entrada en Emoncms.....	47

Figura 29. Programación de claves de Emoncms en Arduino	47
Figura 30. Programacion de la petición al servidor de Emoncms en Arduino	48
Figura 31. Los diferentes datos son enviados en la petición al servidor de Emoncms ...	48
Figura 32. Conexión de los diferentes elementos en NCS.....	49
Figura 33. Usuarios de NCS.....	52
Figura 34. Selección del tipo de tintura	53
Figura 35. Botones de control del proceso de tintura	53
Figura 36. Los datos de los diferentes sistemas es presentado en la pantalla secundaria	54
Figura 37. Gráfica de la tintura blanco-negro, obtenida en Emoncms.....	54
Figura 38. Gráfica de la tintura a colores, obtenida en Emoncms	55
Figura 39. Contador entre eventos en el monitor serial de Arduino	56
Figura 40. Gráfica de contador vs eventos.....	57
Figura 41. PID con valor $k_p=0.5$	58
Figura 42. PID con valor $k_p=1$	59
Figura 43. PID con valor $k_p=1.5$	59
Figura 44. PID con valor $k_p=2$	59
Figura 45. PID con valores $k_p = 1.5$, $k_d =0.5$	61
Figura 46. PID con valores $k_p=1.5$, $k_d=1$	61
Figura 47. PID con valores $k_p= 1.5$, $k_d =1.5$	62
Figura 48. PID con valores $k_p = 1.5$, $k_d = 1$, $k_i =0.01$	62
Figura 49. Gráfica PID con Emoncms	64

ÍNDICE DE ECUACIONES

Ecuación 1	7
Ecuación 2	8
Ecuación 3	40

ÍNDICE DE ANEXOS

Anexo 1. Características de consumo eléctrico de los elementos de las estaciones.	69
Anexo 2. Fuente Switching	69
Anexo 3. Circuitos de Regulación de Voltaje.....	71
Anexo 4. Diseño de circuitos de salida para la estación PWM y Control On/Off.....	73
Anexo 5. Estación PID.....	74
Anexo 6. Estación PWM y Control On/Off.....	74
Anexo 7. Central de control.	75

RESUMEN

Los diversos sistemas de control existentes intercambian diferente tipo de datos entre dispositivos de campo, la gran variedad de redes y cableado dedicado para cumplir los requerimientos de los diferentes niveles de la pirámide de automatización presentan una elevada complejidad en configuración con altos costos de adquisición e instalación.

En el presente proyecto se propone diseñar e implementar un sistema de control en red, cuya característica principal es usar un medio compartido que permitirá comunicar al controlador con los sensores y actuadores del campo.

Una nueva propuesta en sistemas de control, con un protocolo y cableado general que responda las necesidades de todos los niveles de la pirámide de automatización, común para las redes de información e industriales

La red propuesta en este proyecto permitirá controlar y visualizar los diferentes procesos de planta, los múltiples usuarios podrán acceder a los datos de los sistemas de control de forma ágil y rápida por los diversos dispositivos dentro de la cobertura de la red.

ABSTRACT

The various existing control systems exchange different types of data between field devices, the wide variety of networks and dedicated wiring to meet the requirements of the different levels of the automation pyramid have a high complexity configuration with high costs of acquisition and installation .

In this project it proposes to design and implement a network control system, whose main characteristic is to use a shared medium that allows the controller to communicate with sensors and actuators in the field.

A new proposal on control systems, with a protocol and general wiring that meets the needs of all levels of the automation pyramid, common information networks and industrial

The proposal in this project network will monitor and display the various plant processes, multiple users can access data from control systems easily and quickly by the various devices within the network coverage.

INTRODUCCIÓN

La industria ha evolucionado acorde a las necesidades de nuevas eras con un mercado que exige sus productos de forma acelerada, los múltiples procesos deben ser monitorizados y controlados continuamente para garantizar la mayor calidad de los artículos de venta.

Varios elementos distribuidos por la planta no trabajan de manera aislada, las redes industriales surgieron como una solución para comunicarlos. Se puede agrupar en dos tipos de redes los diferentes niveles de la pirámide de automatización: información e industriales.

Las redes de información basadas en protocolo internet (por sus siglas en ingles IP – Internet Protocol) se han limitado a trabajar en los dos últimos niveles de la pirámide de automatización, esto es debido a que trabaja con gran volumen de datos y su respuesta no es en tiempo real, al presentarse retardos en la comunicación en un sistema de control la respuesta del controlador tiende al desequilibrar al proceso.

Las redes de información solo han sido aplicadas a tareas de transmisión de datos o comercio electrónico, un sistema de control basado en redes de información presenta una generalización de comunicaciones en la pirámide de automatización, aunque su mayor problema es los retardos aleatorios en la transmisión de información, existen varios métodos o estrategias que pueden solucionar o disminuir su efecto.

CAPÍTULO 1.

ANTECEDENTES

1.1 Título del proyecto

Diseño e implementación de un Sistema de Control en Red, utilizando sensores y actuadores inteligentes para los mecanismos de control: PID, PWM y Control on/off, monitoreados mediante una aplicación por la nube.

1.2 Planteamiento del problema

Las redes industriales presentan diferentes características para responder las necesidades de los niveles de la pirámide de automatización. El Desarrollo de múltiples protocolos facilitan la transmisión de datos según su funcionalidad, muchas marcas en el mercado presentan su propia estructura y gestión datos exclusiva para una red. Los protocolos industriales existentes presentan una elevada complejidad, enormes redes con cableado dedicado que incrementa los costos de instalación y mantenimiento, por ende muchas industrias prefieren prescindir de una red industrial.

Las redes de información tienen un gran potencial en sistemas de control, sin embargo los diferentes problemas que se presentan por ser un sistema de comunicación no determinístico limita su aplicación en varios campos de acción. El desarrollo de algoritmos o estrategias para reducir el impacto de los retardos de comunicación puede ser aplicado desde el nivel de comunicación o control.

1.3. Objetivos

1.3.1. Objetivo general

- Diseñar e implementar un Sistema de Control en Red (por sus siglas en inglés NCS – Network Control System) alámbrica e inalámbrica, utilizando sensores y actuadores inteligentes configurados con módulos Arduino, para los

mecanismos de control: PID, PWM y Control on/off, que serán monitoreados mediante una aplicación por la nube.

1.3.2. Objetivos específicos

- Estudiar la situación actual de NCS así como los sistemas de control inalámbrico (por sus siglas en ingles WCS – Wireless Control System).
- Implementar procesos en los que se pueda aplicar los conceptos de NCS y WCS, tales como procesos: ON/OFF, PID, PWM.
- Desarrollar interfaces amigables para monitorear y controlar los procesos vía web, utilizando módulos Arduino como gestor de datos y páginas dentro de la red.
- Implementar la red de comunicaciones con Arduino para una aplicación en la nube, transmitiendo datos y su posterior presentación gráfica que servirá de historial al proceso.
- Realizar pruebas de comunicación de los sistemas implementados, además de su ejecución en software y hardware realizado con módulos Arduino.

1.4. Alcance

El presente proyecto pretende elaborar un NCS, estableciendo un medio compartido entre controlador y sensores o actuadores de los diferentes procesos de planta. Los diferentes elementos de campo poseen su respectiva dirección IP para la correcta gestión y transmisión de los valores de las variables de interés a su correcto destino.

Los diferentes elementos de campo se conectaran mediante un router vía wifi y Ethernet, cualquier dispositivo capaz de adquirir una dirección IP podrá acceder a la red para controlar y visualizar los diferentes parámetros de los procesos de planta.

Las variables de interés serán monitorizadas por una aplicación de internet, mostrando la evolución temporal de las señales captadas por los sensores.

1.5. Justificación

Las redes de información no han sido explotadas en todo su potencial, la conexión de dispositivos físicos a la red se presenta como una nueva alternativa de comunicación en sistemas de control. Los protocolos industriales presentan múltiples ventajas, pero sus elevados costos representan un gran problema para pequeñas y medianas industrias.

Los NCS presentan una gran adaptabilidad y escalabilidad por tener un protocolo general para los diferentes dispositivos del sistema, el ingreso a la red es ágil y rápido sin configuraciones complicadas.

Los NCS tienen un cableado general lo que representan una reducción en costos y espacio, la red también posee la capacidad inalámbrica facilitando la movilidad, conectividad y funcionalidad de los diferentes elementos de campo.

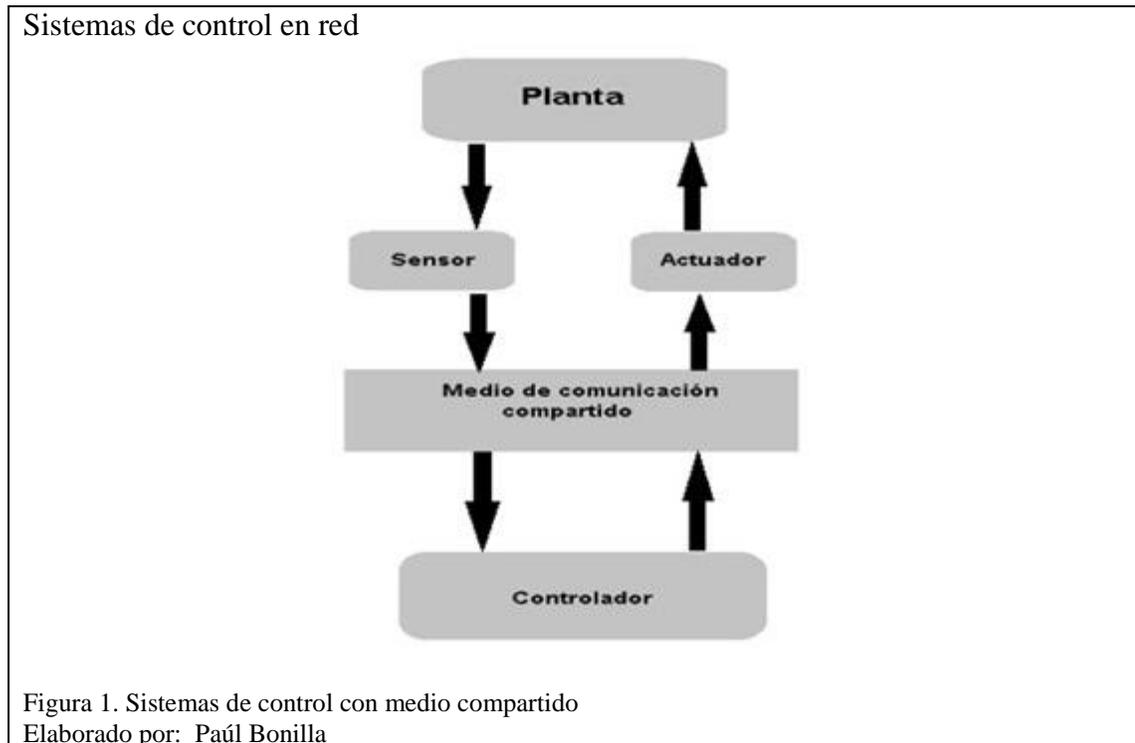
Los NCS son una tecnología en desarrollo que presenta múltiples problemas por ser sistemas de comunicación no determinísticos, el presente proyecto se plantea establecer conocimiento práctico y teórico para la aplicación de redes de información en sistemas de control.

CAPÍTULO 2.

MARCO TEÓRICO

2.1. NCS

Los sistemas de control convencionales usan múltiples medios de comunicación exclusivos entre controlador y los diferentes actuadores y sensores de campo, los NCS se caracterizan por usar un único medio para comunicarse con todos los elementos de campo, no tan solo sensores y actuadores sino todo dispositivo capaz de obtener una dirección IP. El único medio existente se encarga cerrar los diferentes bucles en un sistema a lazo cerrado (Casanova, 2005, pág. 2).



Uno de los principales problemas de NCS son los retardos aleatorios en transmisión de datos, las redes de información basadas en direcciones IP son sistemas no determinísticos con períodos asíncronos en la comunicación de la red. Al existir un único medio de comunicación la cantidad de dispositivos comprometen la eficiencia de la red incrementando los tiempos de retardo (Casanova, 2005, pág. 14).

Al realizar un sistema de control basado en redes de información, los retardos en la transmisión de datos entre planta y controlador tienen una influencia negativa en la acción de control, la respuesta del controlador tiende a desequilibrar al sistema ya que pierde información de la variable de interés hasta que se actualice.

La acción del controlador depende de la evolución temporal de la variable de interés, muchos procesos requieren de respuestas en tiempo real con valores de frecuencias de muestreo elevadas. Las redes industriales se desarrollaron por la necesidad de sistemas de comunicación determinísticos, a diferencia de la redes de información los protocolos industriales trabajan con pequeñas cantidades de datos que van desde bits hasta paquetes mejorando la eficiencia y velocidad de la red cumpliendo con los requerimientos de los diferentes niveles de la pirámide de automatización en tiempo real.

Las posibles soluciones de NCS pueden ser abarcadas desde dos áreas relacionadas al sistema: área de comunicaciones, teniendo en consideración la cantidad de dispositivos que trabajan en la red con un adecuado valor de ancho de banda, acceder al medio de comunicación cuando se cumplan ciertos requisitos en los procesos para evitar el uso innecesario de la red, área de control, desarrollando algoritmos o estrategias que respondan ante la ausencia de datos reduciendo el efecto de los retardos en el sistema de control. Las soluciones que se aplican a los diferentes sistemas de control dependen de la sensibilidad de la variable de interés ante una respuesta, cada proceso posee características particulares que deben de ser consideradas para una acción óptima de un controlador que trabaja con retardos (Casanova, 2005, pág. 16).

2.2. Control PID

En un sistema a lazo cerrado la diferencia entre valor medido y valor de referencia se conoce como error, el mecanismo de control PID realiza diferentes operaciones sobre el valor de error que permite elaborar una respuesta para equilibrar al sistema.

$$u(t) = k_p * e(t) + k_i * \int e(t)dt + k_d \frac{de(t)}{dt}$$

Ecuación 1

Los tres valores de este mecanismo de control son: Proporcional, Integral, Derivativo con sus coeficientes k_p , k_i , k_d respectivamente, cuya función es amplificar la acción de cada uno. El valor proporcional trabaja con los errores actuales, el valor integral trabaja con la suma de los errores pasados, el valor derivativo trabaja con la derivación de la función de error en el tiempo, prediciendo un futuro posible error. La suma de estos tres parámetros permite obtener un valor de salida acoplado a un actuador para controlar un proceso.

Los valores de un controlador PID pueden ser calculados estableciendo una función de transferencia del sistema analizado, las múltiples características y variables de un proceso son difíciles de relacionar, debido a la complejidad de este método muchos desarrolladores de sistemas de control optan por otras alternativas como: ajuste por prueba y error, método de Ziegler y Nichols, etc. (Morilla, 2006, pág. 5).

2.3. Control ON/OFF

Mecanismo de control a lazo cerrado en el que solo presenta dos estados, los sistemas que trabajan bajo este método presentan un error oscilante por la continua variación del actuador. Al usar dispositivos electromecánicos como relés que no pueden conmutar a altas velocidades, se agregan tiempos de apagado y encendido en la programación del controlador para no dañar al actuador.

2.4. Control PWM

Su fundamento se basa en dos estados que son encendido o apagado pero aplicado a una señal periódica, para realizar el control de un motor por Modulación por ancho de pulso (por sus siglas en inglés PWM – Pulse Width Modulation) se tienen dos señales: la señal de potencia y la señal de control. La señal de control está continuamente conmutando, en

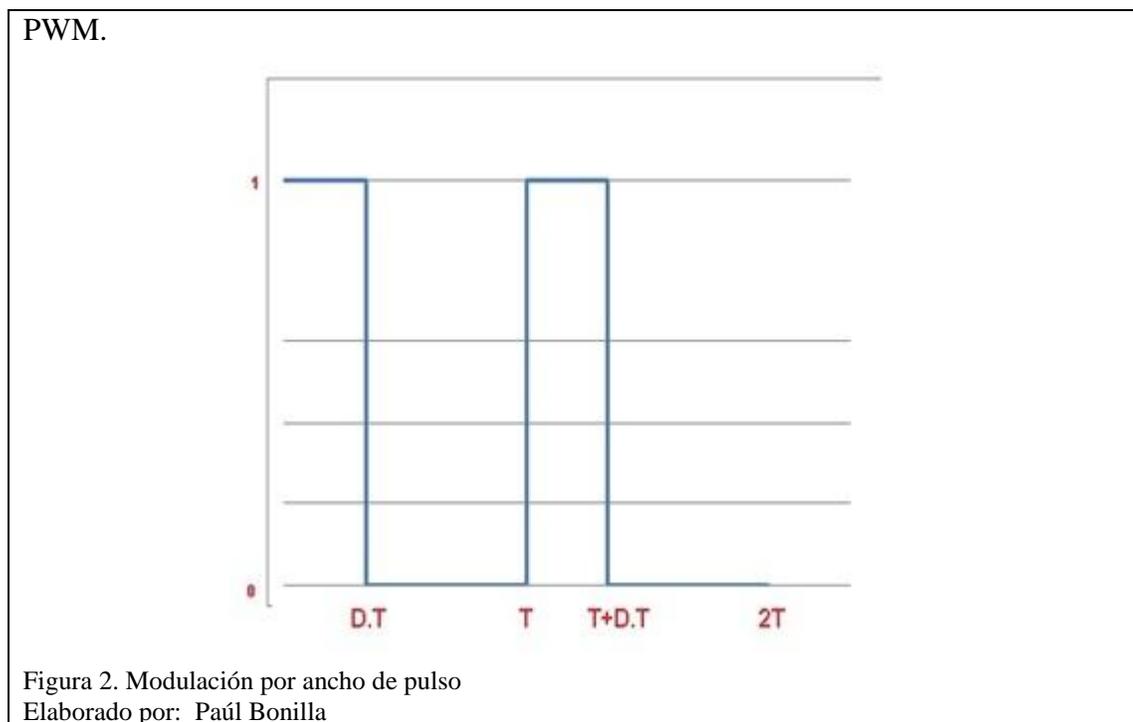
el lapso de tiempo en que está sin energía eléctrica en el motor existe un residual de energía mecánica que lo mantiene en movimiento, antes que el motor pare totalmente se energiza nuevamente manteniendo el giro del rotor, por medio de este proceso se realiza el control de velocidad de motores a corriente continua.

Se pueden establecer tres parámetros en PWM: el ciclo de trabajo D, periodo de la señal T, tiempo en el que la señal energiza el motor L.

$$D = \frac{L}{T}$$

Ecuación 2

La velocidad del motor es directamente proporcional al ciclo de trabajo del PWM.



2.5. Redes de información

Los fundamentos y características de redes de información basadas en direcciones IP son un tema sumamente extenso que abarca diferentes aspectos de gran complejidad, es necesario explicar de forma rápida algunos conceptos.

2.5.1. Modelo TCP/IP

Conjunto de protocolos que permite comunicar los diferentes dispositivos de la red, estableciendo características de la información circundante en el medio entre emisor y receptor, los diferentes tipos de datos pasan por varios procedimientos que pueden ser agrupados en niveles o capas. Modelo TCP/IP (por sus siglas en inglés Transmission Control Protocol/ Internet Protocol, en español Protocolo de Control de Trasmisión / Protocolo Internet) presenta 4 niveles que son: nivel de Acceso a Red, nivel de internet, nivel de transporte, nivel de aplicación.

Cada dispositivo dentro de la red es identificado por una dirección IP. Existen 2 tipos de versiones que son: IPv4, contiene 32 bits que se pueden dividir en 4 octetos, IPv6 contiene 128 bits y surge por el límite de direcciones existente en IPv4.

Los dispositivos de red poseen otro tipo de dirección conocida como MAC (por sus siglas en inglés Media Access Control, en español Control de acceso al Medio), que contiene 48 bits dividido en 6 bloques hexadecimales. A diferencia de la dirección IP, la dirección MAC no puede ser cambiada y es única de cada dispositivo de la red.

2.5.2. DHCP

El protocolo de configuración dinámica de host (por sus siglas en inglés DHCP-Dynamic Host Configuration Protocol) otorga ciertos parámetros de configuración a los diferentes dispositivos de la red. DHCP permite crear una lista de direcciones IP que pueden ser usadas por elementos pasajeros de la red.

2.5.3. Router

Dentro de la red un router se encarga de encaminar los diferentes paquetes de información, también permite conectividad con redes externas. Los diferentes router en

el mercado proveen diferentes tipos de protocolos, configuraciones de seguridad, permisos a diferentes procesos, direccionamiento, etc.

2.6. Sensores y actuadores inteligentes

Los sensores y actuadores inteligentes tienen un conjunto de funciones diseñadas o programadas en un único dispositivo. Poseen uno o varios procesadores que cumplen diferentes funciones: conversor análogo-digital, memoria, gestión de datos en la red, comunicación con otros dispositivos, decisiones de acción según eventos (Custodio, 1999, pág. 19). Los sensores y actuadores inteligentes en redes de información tienen la capacidad de enviar o recibir información a los diferentes dispositivos de red por medio de su respectiva dirección IP, trabajando como cliente o servidor.

2.7. Arduino

La plataforma de software y hardware libre arduino presenta múltiples herramientas de gran potencial en diferentes áreas, muchos diseñadores han aceptado esta plataforma por su versatilidad y simplicidad, reúne los elementos desordenados de programación de un microcontrolador para adaptarlo en un proceso simple de trabajo (Arduino m. , 2015).

2.7.1. Módulo ethernet Arduino

El módulo Ethernet arduino permite la conexión a internet por medio del chip Wiznet W5100 que le permite trabajar con los protocolos TCP/IP (Arduino E. , 2015). El modulo arduino Ethernet puede ser adaptado a diferentes redes por medio de su conector RJ-45, combinando las características de un microcontrolador y red en un solo dispositivo, trabajando como cliente haciendo peticiones a diferentes elementos de red o como servidor programando páginas en html.

2.7.2. Dragino Yun

El módulo dragino yun permite conectividad con internet vía Ethernet y wi-fi, posee otro procesador Atheros AR9331 que soporta Linux, la librería Bridge permite la comunicación entre los 2 procesadores vía serial. El procesador ATmega 32u4 se encarga de la programación USB, el procesador AR9331 (Linino) se encarga de host usb, interfaces de red y tarjeta SD (Arduino Y. , 2015).

2.8. Sensor LM35

Sensor de temperatura que tiene un rango de -55°C a 150°C , su salida es lineal y la razón de cambio es 10 mv equivale a un grado Celsius. No requiere de calibración externa y para adaptarlo a cualquier procesador solo basta con introducir el pin de salida a un conversor análogo-digital y programar su equivalencia a temperatura (Texas, 2016, pág. 1).

2.9. Sensor Sharp GP2Y0A41SK0F

Sensor de distancia que tiene un rango de 4cm a 30cm, su salida es analógica no lineal. Compuesto por 2 circuitos, detector de posición y diodo emisor de infrarrojo. A diferencia de otros sensores de distancia, la superficie del objeto no afecta su lectura (Sharp, 1999, pág. 1). Para adaptarlo a un procesador se deben acoplar diferentes pendientes por intervalos a lo largo la de la curva de salida del sensor para obtener la distancia.

2.10. HTML

Lenguaje de marcas de hipertexto (por sus siglas en inglés HTML- Hyper Text Markup Language) es usado para la elaboración de páginas Web. No se requiere de un programa especial para su elaboración, se requiere de un editor básico ya que el código HTML es texto ASCII.

HTML se caracteriza por ser un lenguaje simple cuya información no ocupa un espacio significativo. Sus diferentes comandos permiten crear, estructurar y vincular páginas Web

2.11 Análisis de la situación actual

Los NCS presentan múltiples problemas e inconvenientes por ser un sistema de comunicación no determinístico, las características que presenta ha despertado interés de varios expertos en el área por los diversos beneficios que se obtienen: reducción costos de cableado, diferentes dispositivos funcionando con un protocolo general, control centralizado. (Casanova, 2005, pág. 15).

Los sistemas de control convencionales tiene una frecuencia de elevado valor y síncrona, la respuesta del sistema es activada por el tiempo ya que existe fundamentos teóricos establecidos y bien desarrollados para señales muestreadas periódicamente. (Dormido, Sánchez, & Kofman, 2008, pág. 19).

Dependiendo del proceso de control el período de muestreo debe ser menor de un segundo. NCS no dispone tiempos de muestreo tan bajos, no existe garantía alguna de que la información necesaria llegue en un tiempo determinado. Los NCS trabajan con tiempos de muestreo semisíncronos y asíncronos. (Dormido, Sánchez, & Kofman, 2008, pág. 2).

Las aplicaciones de NCS son muy escasas a nivel académico e industrial, especialmente en sistemas de control a lazo cerrado por trabajar con dinámicas rápidas. (Dormido, Sánchez, & Kofman, 2008, pág. 4) . Muchos investigadores han propuesto diferentes tipos de soluciones que pueden reducir el efecto de los retardos aleatorios en la comunicación.

Clasificación de sistemas por acción.

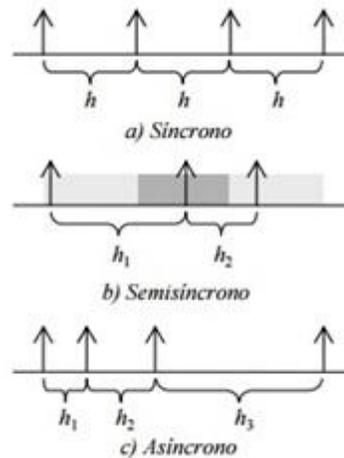


Figura 3. Clasificación de la distribución de las acciones en función del instante de su activación. Elaborado por: (Dormido, Sánchez, & Kofman, 2008, pág. 2).

Dentro de las diferentes estrategias se plantea una respuesta durante la ausencia de información de la red, el controlador puede interpretar o intuir la tendencia que podría tomar la variable de interés por medio de las características físicas del proceso generando una respuesta alternativa. Al presentar una solución de dichas características, se desarrolla un algoritmo de compensación exclusivo para el proceso controlado.

Otra solución de gran relevancia es los sistemas de control por eventos que se basa en la evolución dinámica del sistema para realizar una acción de control, en los sistemas convencionales el avance del tiempo define las acciones de procesamiento y comunicación. Los múltiples beneficios que se obtienen de control por eventos son básicamente la liberación de carga innecesaria de trabajo para el controlador y red ya que solo es necesario realizar una respuesta bajo ciertos parámetros del sistema y no del tiempo. (Dormido, Sánchez, & Kofman, 2008, pág. 2).

Sistemas de muestreo por tiempo y eventos.

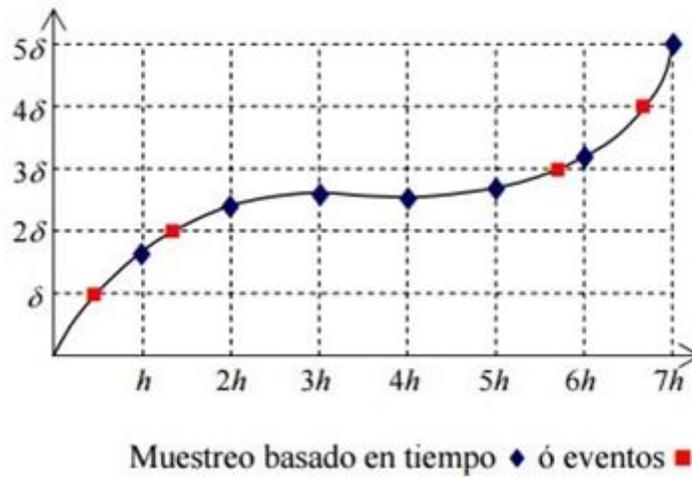


Figura 4. Muestreo basado por tiempo y eventos
Elaborado por: (Dormido, Sánchez, & Kofman, 2008, pág. 2) .

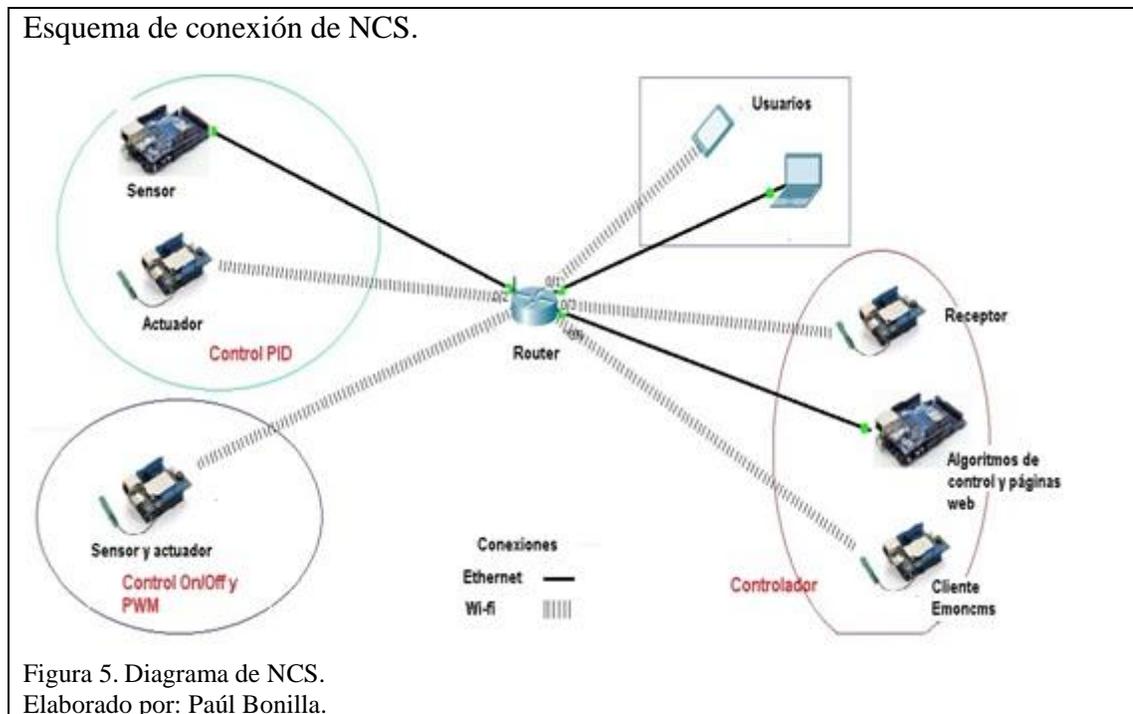
CAPÍTULO 3. DISEÑO E IMPLEMENTACIÓN

En el presente capítulo se detalla el modelado de los diferentes módulos de NCS.

3.1. Definición del problema

Se necesita diseñar e implementar un sistema de control en red, los diferentes sensores y actuadores de los módulos deben enviar y recibir información a un controlador central vía wi-fi o Ethernet.

La NCS creada debe presentar interfaces amigables para los distintos usuarios que ingresan a la red, las diferentes variables de los sistemas llevan un registro gráfico por medio de una aplicación de internet. Los diferentes módulos deben cumplir su objetivo de control manteniendo un error aceptable acorde al proceso establecido.



3.2. Características del sistema

Se debe de analizar los diferentes requerimientos o beneficios extra que el sistema puede ofrecer, las demandas las plantea el usuario y las respuestas las genera el ingeniero con sus conocimientos y criterios técnicos.

Tabla 1.
Voz del ingeniero y usuario

#	Voz del usuario	Voz del ingeniero
1	Que no sea muy cara	Costo
2	Fácilmente desmontable	Cantidad de Módulos
3	Que no sea muy pesada	Volumen
4	Responda a los diferentes elementos de red	Tiempo de respuesta de la red
5	Cumpla los objetivos de control de los diferentes procesos	Error

Nota. Elaborado por: Paúl Bonilla

Cada una de las características de la voz del ingeniero tienen una gran importancia, sin embargo se hará mayor énfasis en las 2 últimas. Las características 4 y 5 responden a la funcionalidad del proceso.

3.2.1. Costo

El factor económico define muchos de los aspectos del proyecto terminado, gran cantidad de detalles a nivel de software y hardware pueden ser elaborados o reciclados para ahorrar recursos, sin embargo siempre se debe considerar la alternativa funcional que responda las demandas del proceso con relación costo-beneficio.

3.2.2. Cantidad de módulos

Al ser un sistema que depende de varios procesos relacionados conectados vía alámbrica e inalámbrica, se elaborarán diferentes estaciones que albergan los elementos de control y comunicación. Se establecen 3 estaciones: PID, Control on/off y PWM, Controlador o Central (Ver fig. 5).

3.2.3. Volumen

Las dimensiones de cada módulo define la comodidad y confort para movilizar o trabajar con las diferentes estaciones, se estipulan realizar bases de tabla triplex prensada con las siguientes dimensiones: 28cm x 28 cm x 2cm. Para proteger los diferentes elementos se realizarán cajas de acrílico, un material de fácil uso, muy estético y permite visualizar la parte interna por su transparencia.

3.2.4. Tiempo de respuesta de la red

La eficiencia y velocidad de la red están establecidos por la cantidad de dispositivos en la red por el ancho de banda, en redes de información los retardos no implican mayor problema en su área de trabajo, en redes industriales tienden a deteriorar al proceso. Es lógico pensar que para NCS la presencia de retardos que afecten al proceso es algo inevitable, pero se plantean soluciones (control eventos, algoritmos de compensación) para presentar una red que cumpla con los requerimientos establecidos.

3.2.5. Error

Cuando el error en un sistema tiende a ser de valores bajos o despreciables, se considera que el mecanismo de control ha cumplido sus objetivos. En NCS la respuesta del controlador esta vincula al único medio comunicación con retardos en la transmisión de datos, tanto las soluciones a nivel de software y hardware tiene que responder los inconvenientes de estabilidad del sistema.

3.3. Análisis funcional

Las diferentes estaciones se crearán usando la plataforma Arduino como un sistema independiente capaz de generar control y comunicación. Se plantea comparar las tres características más importantes de la voz del ingeniero con otra plataforma para establecer criterios de beneficios y optimización. Las características de cantidad de módulos y volumen no serán consideradas en este análisis ya que las dos plataformas a comparar (Arduino y Raspberry PI), tienen dimensiones similares.

3.3.1. Raspberry PI

Raspberry PI es una computadora de tamaño reducido. Fue desarrollada con fines educativos en Reino Unido para estimular el interés de múltiples estudiantes de programación. Arduino y Raspberry PI tienen diferentes características a nivel de software y hardware, los diferentes enfoques al solucionar un problema dependen de las herramientas de las que se dispone en un proyecto.

Tabla 2.
Diferencias entre Arduino y Raspberry PI

	Arduino uno	Raspberry PI modelo B
Precio	30\$	70\$
Tamaño	7.6cm X 1.9cm X 6.4cm	8.6cm X 5.4 cm X 1.7cm
Memoria RAM	2kB	512MB
velocidad de reloj	16MHz	700Mhz
Multitarea	No	Si
Voltaje de entrada	7 a 12 Vdc	5 Vdc
Puertos USB	Uno	Dos
Entorno de desarrollo integrado	Arduino IDE	Scratch, IDLE, cualquiera con soporte Linux

Nota. Elaborado por : Paúl Bonilla

3.3.2. Comparación en costos

Arduino no tiene incorporados en su placa conectores RJ-45 o wi-fi para conexión a internet, se requiere adaptar otros módulos extra.

Tabla 3.
Costos de Arduino Ethernet y wifi

	Precio
Módulo ethernet	20\$
Dragino yun	50\$
Arduino + Módulo ethernet	50\$
Arduino + Dragino yun	80\$

Nota. Elaborado por : Paúl Bonilla

El precio para obtener un módulo activo en NCS no tiene mayor diferencia entre las dos plataformas, pero la capacidad de procesamiento de Raspberry PI es mucho más elevada respecto Arduino. Los diferentes sensores y actuadores de las estaciones de NCS no requieren de gran potencia de cálculo, para la central se usará Arduino Mega con memoria RAM de 8KB por tener la mayor carga de trabajo con un incremento de costo de 5\$.

3.3.3. Comparación en tiempos de respuesta de la red

La capacidad de procesamiento de Raspberry PI mejoraría la eficiencia de la red al disminuir los tiempos de retardo, más no eliminarlos. El mayor factor que incrementa el tiempo de los retardos es la propia naturaleza no determinista de la red. La capacidad de Arduino está correctamente valorada para los diferentes mecanismos, se sobredimensionaría al usar una Raspberry para hacer un sensor o actuador en la red.

3.3.4. Comparación en error

Ambas plataformas pueden realizar los diferentes algoritmos obteniendo sistemas de control con bajos valores de error, pero Arduino tiene gran facilidad de uso y flexibilidad para acoplarse a cualquier proceso. Arduino es una plataforma de software y hardware libre que es adaptable a varios sistemas operativos, Raspberry PI es computadora que trabaja en sistema operativo basado en Linux. La simplicidad de Arduino es un factor que atrae a varios diseñadores, la gran cantidad de información y librerías creadas por programadores independientes facilitan la elaboración de diferentes proyectos.

3.4. Diseño y configuración de la red

3.4.1. Configuración del router

Los diferentes parámetros de red se configuran en router modelo TL-WR740N, con las siguientes características más relevantes para el proyecto (ver tabla 4.).

Tabla 4.
Router TP-WR740N

Velocidad de transmisión de inalámbrica de 150 Mbps
Soporta direcciones IP estáticas y dinámicas
Rango de frecuencia: 2.4 a 2.4835 GHz
Integrado con un Switch de 4 puertos
Estándar :802.11 n/b/g/
Dimensiones: 174mm X 118 mm X 33 mm
Soporta protocolos: Traducción de direcciones de red (por sus siglas en ingles NAT- Network Address Translation), DHCP.

Nota. Elaborado por : Paúl Bonilla

3.4.2. Direcciones IP de la red

Se configura la red con el rango de direcciones 192.168.0.X con máscara de red 255.255.255.0, bajo estos parámetros se pueden obtener 254 direcciones para los diferentes dispositivos de la red, en realidad se obtiene 256 direcciones pero la primera dirección es de la red (192.168.0.0) y la última es de broadcast (192.168.0.255). Se configura en el router la dirección 192.168.0.1 con la respectiva máscara de subred.



3.4.3. Configuración DHCP

Cuando un dispositivo ingresa a la red necesita establecer parámetros de configuración, las diferentes direcciones IP permiten identificar origen y destino de la información, DHCP configura automáticamente nuevos dispositivos para ingresar a la red. Los parámetros de DHCP se configuran en el router.

Parámetros de DHCP en el Router

The screenshot displays the DHCP configuration interface of a TP-LINK router. The browser address bar shows the IP address 192.168.0.1. The left sidebar contains a menu with the following items: Status, Quick Setup, QSS, Network, Wireless, DHCP (highlighted), DHCP Settings (highlighted), DHCP Clients List, Address Reservation, Forwarding, Security, Parental Control, Access Control, Advanced Routing, Bandwidth Control, and IP & MAC Binding. The main content area is titled 'DHCP Settings' and includes the following fields:

- DHCP Server:** Radio buttons for 'Disable' and 'Enable' (selected).
- Start IP Address:** Text input field containing '192.168.0.100'.
- End IP Address:** Text input field containing '192.168.0.199'.
- Address Lease Time:** Text input field containing '120' minutes (1-2880 minutes, the default value is 120).
- Default Gateway:** Text input field containing '192.168.0.1' (optional).
- Default Domain:** Text input field (optional).
- Primary DNS:** Text input field containing '0.0.0.0' (optional).
- Secondary DNS:** Text input field containing '0.0.0.0' (optional).

A 'Save' button is located at the bottom right of the settings area.

Figura 7. Configuración de DHCP

Elaborado por: Paúl Bonilla.

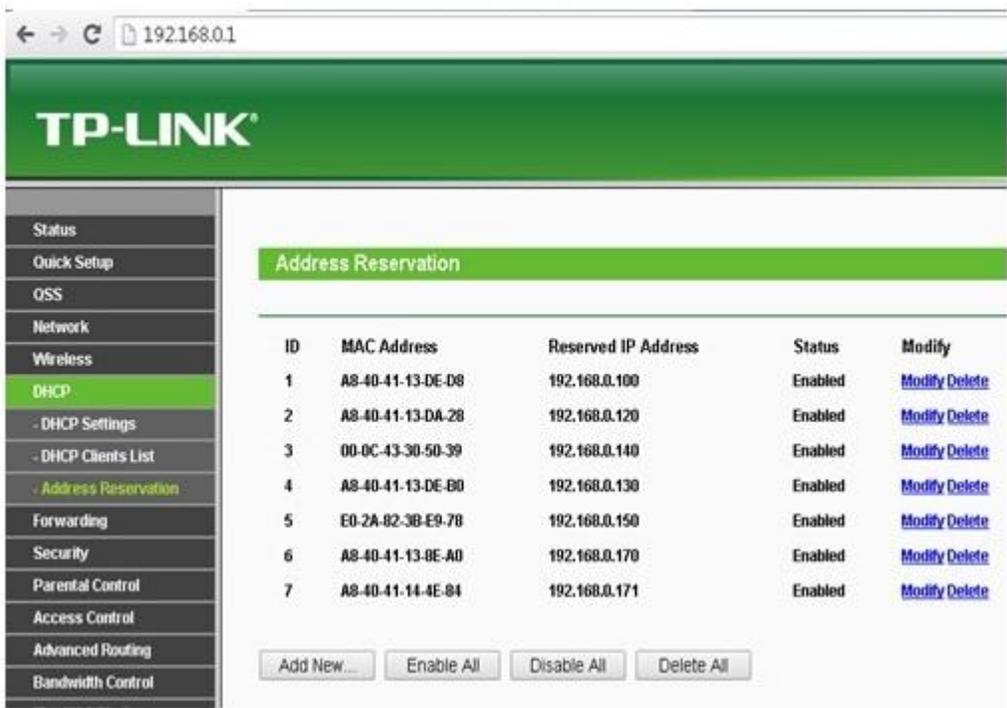
Se activa DHCP estableciendo un rango de direcciones que inicia en 192.168.0.100 hasta 192.168.0.199, también se define la cantidad de tiempo en la que los dispositivos mantienen los parámetros de configuración otorgadas por el protocolo, se deja valor por defecto de 120 minutos.

Los diferentes dispositivos de NCS no pueden tener direcciones IP dinámicas, cada elemento de la red envía información específica para un proceso definido, DHCP puede otorgar hasta 100 direcciones diferentes a un solo dispositivo.

Se deben definir direcciones estáticas para cada elemento de red, en los dispositivos conectados vía Ethernet se puede configurar direcciones estáticas en el propio host, teniendo en cuenta que la dirección pertenezca al rango definido, en los dispositivos que se conectan vía wi-fi, DHCP define sus parámetros. Para solucionar este problema se puede definir una dirección IP fija por medio de la dirección MAC del dispositivo. Cuando un nuevo dispositivo entra a la red envía datos propios del host, como la

dirección MAC que es única de cada dispositivo. Con la dirección MAC se puede establecer una IP fija para un dispositivo de red.

Reservación de direcciones del pool de DHCP



ID	MAC Address	Reserved IP Address	Status	Modify
1	A8-40-41-13-DE-D8	192.168.0.100	Enabled	Modify Delete
2	A8-40-41-13-DA-28	192.168.0.120	Enabled	Modify Delete
3	00-0C-43-30-50-39	192.168.0.140	Enabled	Modify Delete
4	A8-40-41-13-DE-B0	192.168.0.130	Enabled	Modify Delete
5	E0-2A-82-3B-E9-78	192.168.0.150	Enabled	Modify Delete
6	A8-40-41-13-DE-A0	192.168.0.170	Enabled	Modify Delete
7	A8-40-41-14-4E-84	192.168.0.171	Enabled	Modify Delete

Figura 8. Configuración de reservación de direcciones
Elaborado por: Paúl Bonilla.

El resto de dispositivos tendrán direcciones IP dinámicas, al no estar la dirección MAC en esta lista, DHCP definirá sus parámetros. Los diferentes datos generados por los usuarios tienen gran relevancia para los procesos de control, pero son información general. La central de NCS es un servidor que responde a la petición de cualquier dirección IP de la red.

3.5. Descripción de los procesos de control

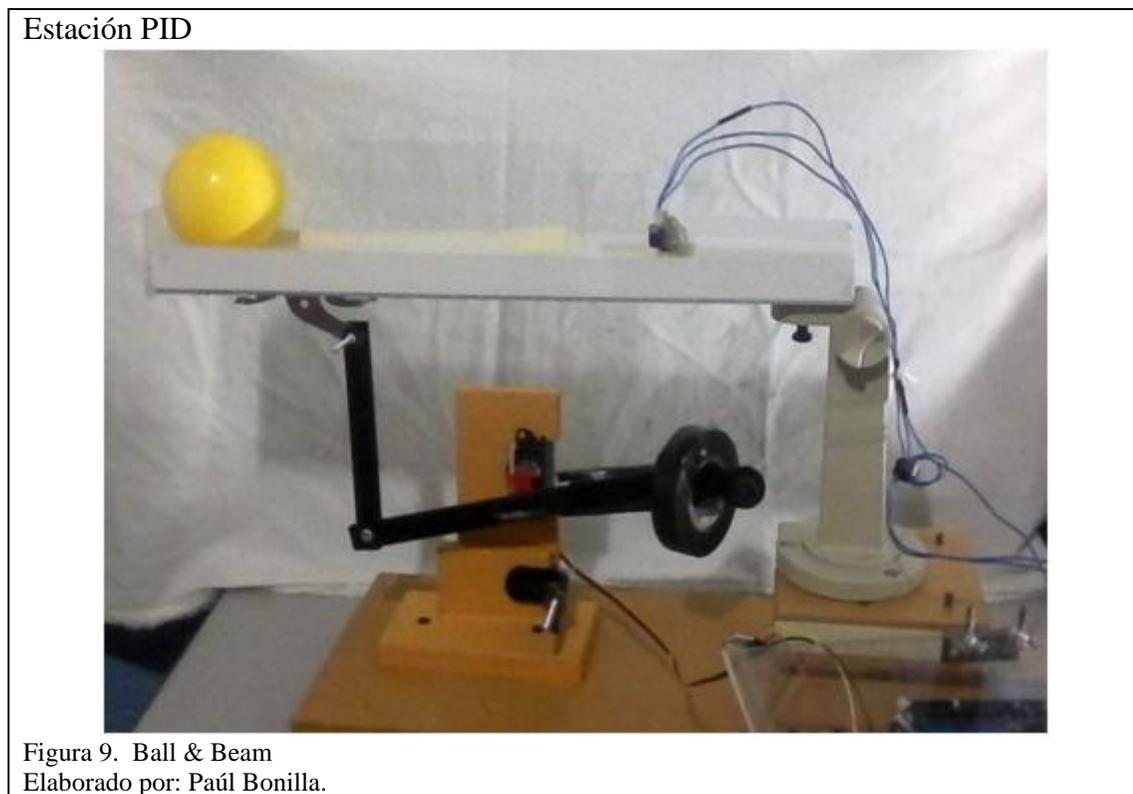
Para la creación de NCS se tomaron tres procesos donde estarán distribuidos los diferentes sensores y actuadores inteligentes vinculados vía Ethernet o Wi-fi al controlador central. El controlador central también envía los valores de las variables de

interés a una aplicación de internet (Emoncms) que permite graficar y registrar su evolución temporal.

Cada sensor y actuador tiene su propia dirección IP que le permite gestionar los diferentes datos como cliente o servidor. El router de la red configura características como: direcciones fijas, contraseñas, reconocimiento de direcciones MAC, conectividad con redes externas y nuevos usuarios en la red.

3.5.1. Primera planta de control se simula el proceso Ball & Beam(PID)

Ball & Beam es un sistema que consiste en una pelota que puede rodar por la inclinación de una superficie que está conectada a unos ejes con un servo motor. La posición de la pelota depende básicamente del ángulo del servo motor, existen otros parámetros que se deben de considerar del sistema como: peso de la pelota, rozamiento, material de la pelota y superficie, dimensión de los ejes de transmisión.



3.5.2. Sensor inteligente de distancia Ethernet

El módulo Ethernet Arduino debe recibir la información del sensor distancia Sharp GP2Y0A41SK0F y crear un servidor en la dirección IP estática 192.168.0.177 en el puerto 9911. El sensor inteligente es conectado al router, todo elemento de red puede hacer consulta del valor de distancia, el servidor tiene un programa en HTML que presenta esta información. El módulo Ethernet de arduino permite configurar direcciones IP, puertos y dirección MAC. A diferencia de otros dispositivos la dirección MAC si puede ser modificada en Arduino.

Configuración de parámetros de red en Arduino.

```
byte mac[] = {0xAE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};  
IPAddress ip(192, 168, 0, 177);  
EthernetServer server(9911);
```

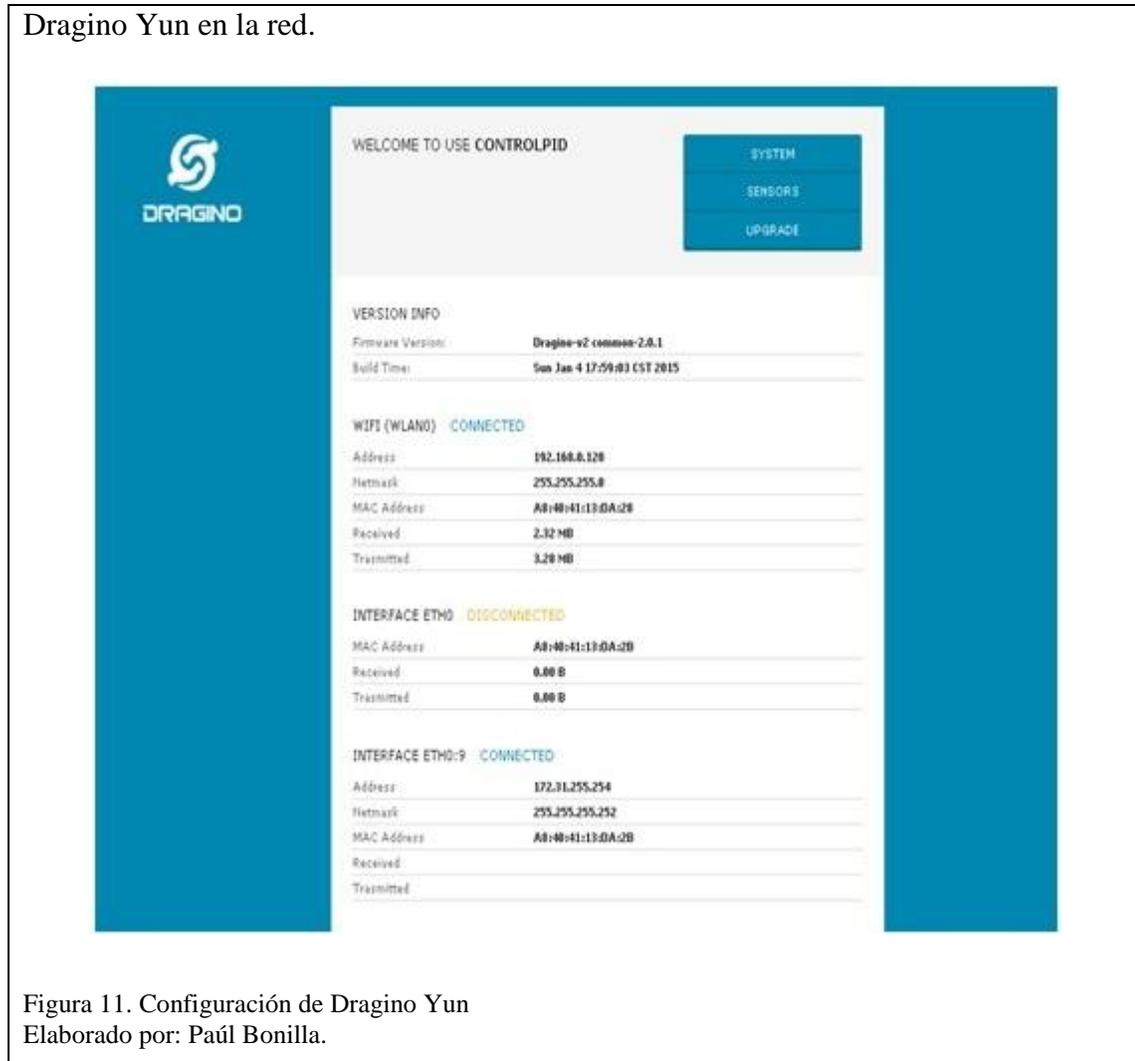
Figura 10. Parámetros de red en Arduino
Elaborado por: Paúl Bonilla.

El sensor distancia Sharp GP2Y0A41SK0F tiene una salida que no puede ser expresada en una función lineal, se debe establecer ciertas condiciones para obtener la relación de voltaje-distancia. Se trazan líneas por diferentes tramos para cubrir la trayectoria de la señal, la conducta del sensor no siempre es exacta a la gráfica, por lo que es preferible calibrar al sensor trabajando en el proceso.

3.5.3. Actuador inteligente wi-fi

Se debe configurar Dragino Yun a la red LAN para establecer las características del dispositivo dentro de la red. Por medio de una computadora vía wi-fi se establece la conectividad con la red de trabajo, se conecta con la red Dragino y se hace una petición

en la dirección 192.168.240.1, para visualizar los parámetros de configuración del dispositivo.

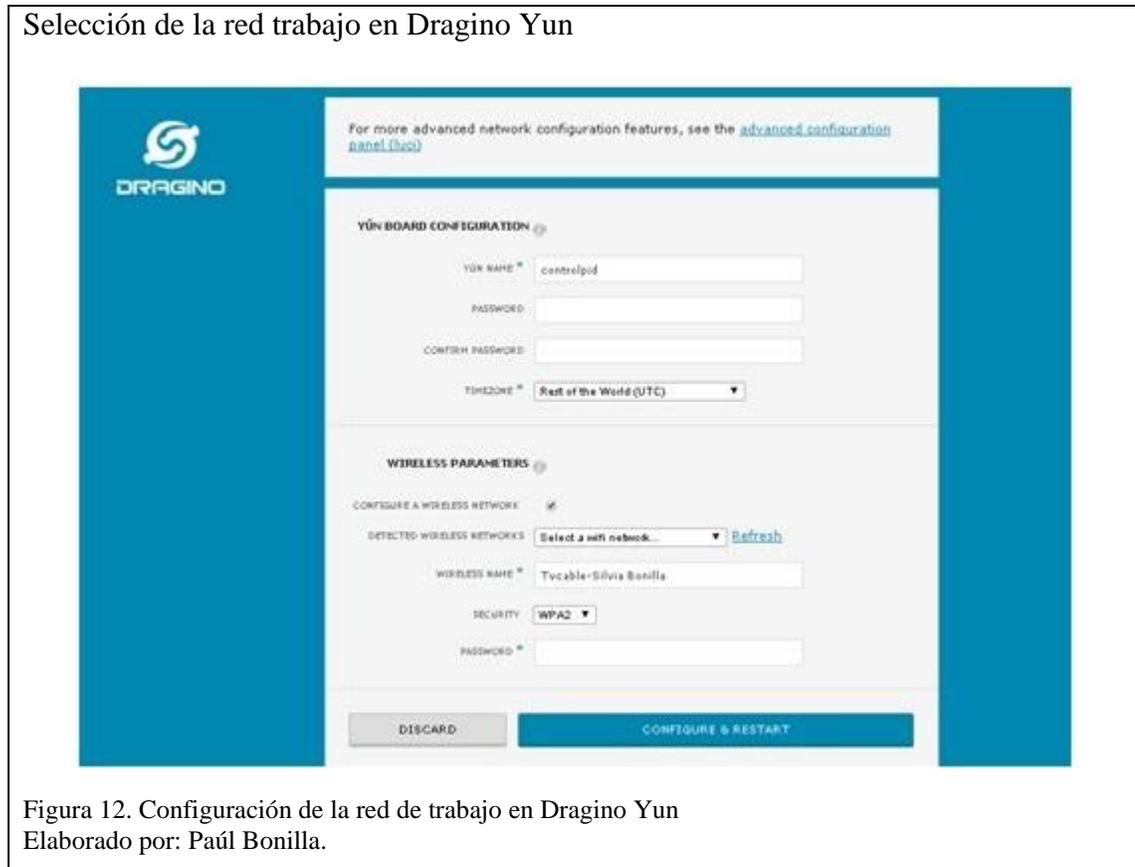


La pantalla principal presenta datos propios del dispositivo como: dirección IP, parámetros de conexión Wi-fi y Ethernet, dirección MAC, información del hardware de Arduino, etc. A diferencia de Arduino Ethernet la dirección MAC no puede ser cambiada.

Dragino Yun permite configurar diferentes parámetros de seguridad como claves de ingreso para su configuración, dentro de los parámetros inalámbricos permite conectarse cualquier red wifi que este dentro de su cobertura. Se selecciona la red trabajo y se

ingresa su respectiva contraseña, el router identifica al dispositivo y le asigna una dirección IP mediante DHCP.

La información generada por el actuador inteligente va a ser específica del dispositivo dentro del proceso, para evitar que DHCP le asigne diferentes direcciones, se reserva su dirección IP con su respectiva dirección MAC.



El actuador inteligente Wi-fi se configura como un cliente que receptorá diferente información de la central control , las librerías de Dragino Yun tiene otras características en comparación de Arduino Ethernet, es debido a que Dragino Yun trabaja con otro procesador basado en Linux. Se usa la librería HttpClient para generar constantes peticiones a la central de control.

Al actuador inteligente del proceso PID recibe los valores del ángulo del servomotor, esta información es enviada por la central que genera una respuesta en la dirección IP 192.168.0.155:9113. Los datos enviados se encuentran en variables tipo String, se debe de transformar a tipo entero para su posterior ejecución en el servomotor.

Para realizar diferentes consultas Dragino Yun presenta la librería Console que permite visualizar información previamente programada. En el IDE de Arduino, se direcciona en el serial monitor la dirección IP del actuador para obtener el valor del ángulo del servomotor.

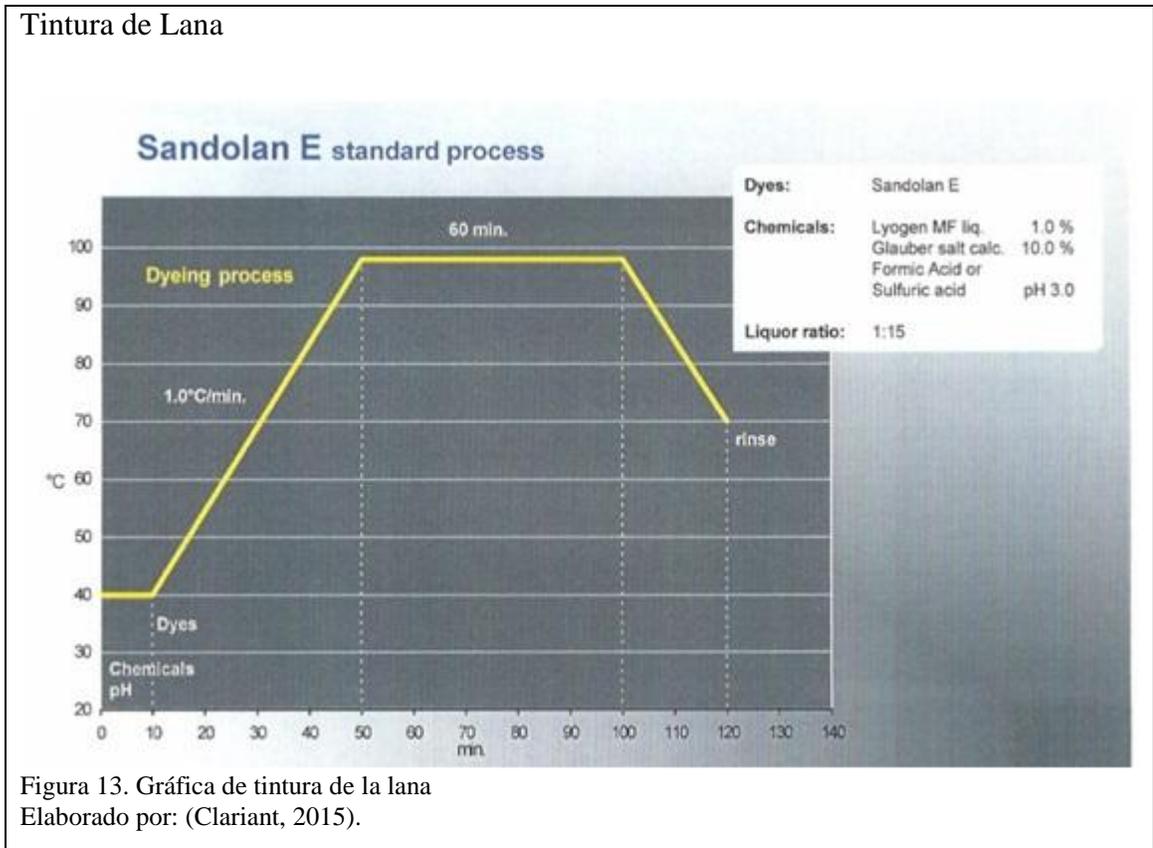
La estación o módulo de sensores y actuadores PID está conectada a la fuente switching, en la entrada del módulo se encuentra una etapa de regulación para energizar todos elementos que requieren diferentes niveles de voltajes, para evitar el calentamiento en el diseño de circuito se incorpora una salida extra para un ventilador (Ver Anexo 5).

El módulo Ethernet Arduino genera una señal muy débil, la distancia entre el módulo y router es de tres metros. El módulo Ethernet Arduino directamente conectado al router con un cable Ethernet de cinco metros, presenta conflictos en la información por la atenuación de la señal. Para solucionar este problema se usa un switch previo al router para amplificar la señal generada por Arduino Ethernet (Ver Anexo 5).

3.6. Segunda planta de control se simula un proceso de industria lanera, en específico de tinturado (PWM y control ON/OFF)

En la industria lanera el proceso de tinturado presenta múltiples problemas en la obtención de un producto final homogéneo que retenga todo el colorante aplicado. Los diferentes factores que intervienen en el proceso son: tipos de colorantes, auxiliares químicos, pH de la solución, temperatura, etc. El proceso de tinturado de Lana no tratada, consiste en depositar la lana en solución química que es sometida a diferentes temperaturas (Lancovici, 2000, pág. 1).

La temperatura debe variar lentamente iniciando desde 30 o 40°C en un intervalo de tiempo, después la temperatura debe incrementar en diferentes variaciones de grado por minuto, esto depende del tipo de colorante que se use. Para colores pastel y tonos claros la temperatura tope es de 80°C, para el resto de colores es de 92°C a 96°C, para garantizar la correcta migración del colorante hacia la lana se deja reposar durante 15 o 20 minutos en la temperatura tope (Lancovici, 2000, pág. 2).



El fundamento teórico para tinturar lana es sumamente extenso que amerita un estudio muy exhaustivo, para el proceso aplicado en NCS solo se realiza el control de temperatura.

Se plantean diferentes tiempos y temperaturas de valor reducido para la aplicación del módulo PWM y control On/Off. Se establecerán dos tipos de tintura, a colores y blanco

y negro. Cada tipo de tintura tiene sus propios parámetros de trabajo, que pueden ser seleccionados por el usuario que ingrese a la red (Ver Tabla 5.).

Tabla 5.
Datos de los tipos de tintura.

	Paso 1		Paso 2		Paso 3	
Tipo	Temperatura [°C]	tiempo [min]	Temperatura [°C]	tiempo [min]	Temperatura [°C]	tiempo [min]
B/N	40	8	40—70	8	70	5
Colores	40	5	40—70	10	70	6

Nota. Elaborado por : Paúl Bonilla

3.6.1. Sensor-Actuador

A diferencia del modulo PID, existe un solo Dragino Yun que está conectado a los diferentes elementos de el proceso de tintura. Las diferentes librerías de Dragino Yun permiten establecer un servidor-cliente, crear una página en HTML que presente los valores de la variable de interés y hacer peticiones a la central de control para obtener los datos de los actuadores.

A diferencia de Arduino Ethernet, las librerías de Dragino Yun facilitan la interacción con la red, pero la gestión de datos somete a los procesadores a tener una respuesta muy lenta respecto a la programación lineal existente.

Al realizar una petición a un servidor de la red desde Dragino Yun, mientras no exista una respuesta o finalice todo el mensaje, la siguiente línea de programación no se ejecutara. El tamaño de los datos enviados influye negativamente en la respuesta de Dragino Yun.

El sensor lm35 es usado para medir la temperatura del proceso de tintura, se controla la temperatura de un caudín que es encendido o apagado por un relé, PWM controla la velocidad del ventilador, que actúa proporcionalmente con error del sistema.

El módulo genera información de carácter específico, DHCP otorga una dirección dinámica lo cual representa un problema en la gestión de datos. Al igual que en el módulo anterior, se reserva una dirección IP por medio de la dirección MAC única del dispositivo.

Se realiza una petición a la central de control en la dirección IP 192.168.0.155:9112, la central genera una respuesta en un formato String, este mensaje tiene dos identificadores que se separan en una respuesta para el relé y ventilador. Identificados los valores de cada actuador, se los transforma en variables tipo entero para su posterior ejecución.

En el programa se introduce un retardo de dos milisegundos que no afecta directamente la acción del PWM, pero observa un incorrecto funcionamiento del ventilador. Al incluir las diferentes acciones de servidor-cliente, el procesador no ejecuta las acciones que siguen en el programa hasta que la transmisión de datos se haya completado. Los diferentes retardos introducidos por la red oscilan entre dos o tres segundos, por esta razón la acción del control de velocidad del ventilador no se cumple óptimamente. La acción del relé no se ve afectada, debido a que conmuta en períodos de cuatro o más segundos. El problema fundamental radica en la respuesta de la red, mientras el dispositivo que solicitó o envió información no finaliza esta operación, introduce un retardo equivalente a tener un tiempo variable con el comando delay. El control PWM se deteriora por este período de espera introducido en la señal cuadrada de control.

Se crea también un servidor que presenta los valores de temperatura del proceso de tintura en código HTML, cualquier dispositivo de la red puede visualizar esta información al ingresar a la dirección IP del módulo.

3.6.2. PWM en Arduino extra para la compensación de retardos

Para solucionar el tiempo de espera introducido en los períodos de PWM por la gestión de datos de la red, se introduce otra tarjeta Arduino que recepta la información de PWM vía Serial de Dragino Yun (Ver ANEXO 6) .

3.7. Central de control

En la Central de control existen tres dispositivos Arduino que están conectados entre sí vía comunicación Serial. En el módulo Ethernet Arduino se encuentran los diferentes algoritmos de control, la creación de páginas que presentan la información de las diferentes variables de los sistemas, servidores que presentan los datos de actuadores, gestión de datos entre las tarjetas Arduino de la Central de control. Los dos módulos inalámbricos Dragino Yun se encargan de la recepción de datos y establecer comunicación con una aplicación con Internet, la aplicación permite graficar y guardar un registro de la evolución temporal de las variables captadas por los sensores de los diferentes sistemas desarrollados.

3.8. Arduino Ethernet en la central de control

Arduino Ethernet en la central de control está configurado como un servidor, con una dirección IP estática, se configuran otros puertos para elaborar diferentes páginas que responden a las peticiones de los diferentes elementos de los sistemas.

Programa de Arduino Ethernet

```
#include <SPI.h>
#include <Ethernet.h>
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192,168,0, 155);
EthernetServer server(80);
EthernetServer server2(9112);
EthernetServer server3(81);
EthernetServer server4(9113);
```

Figura 14. Parámetros de red de Arduino Ethernet en la Central de control.
Elaborado por: Paúl Bonilla.

3.8.1. Páginas Web

Existen cuatro páginas Web que son desarrolladas en la Central de control, la finalidad de cada página es contener cierta información exclusiva para ciertos elementos de red. Todo el código HTML generado en Arduino es almacenado en una variable String, algunos elementos solo necesitan un dato de pequeñas dimensiones, cualquier información extra en una página cargaría de trabajo innecesario al procesador del cliente o servidor.

La primera página creada en el puerto 80, es la página que presenta la información general del sistema, el código HTML es el más extenso debido a que es la página que el usuario accede para interactuar en los procesos. Se desarrolla un interfaz agradable, que permite configurar los valores como: coeficientes del PID, botón de inicio, botón de paro, botón de reset, tipo de tintura.

También se introduce una tabla que permite la visualización de los parámetros de los procesos como: tiempo del tipo de tintura, valor de referencia (set point), valor medido (set value), coeficientes que están trabajando en el PID, tipo de tintura. La primera página solo se actualiza cuando se realiza una acción como: enviar un dato, seleccionar algún icono o recargando la página.

Página principal

UNIVERSIDAD POLITÉCNICA
SALESIANA
ECUADOR

Sistema PID

USUARIO: PASSWORD: CODIGO:

Login

Salir

Distancia:	29.28 m	Temperatura:	27.58 °C
P1:	0.00	Tipo:	No definido
P2:	0.00	Tiempo:	0.0 min
P3:	0.00	P4:	0.00

Seleccione el tipo de sistema (zona de temperatura)

Salir

Gráfico

Figura 15. Página principal en la dirección IP 192.168.0.155
Elaborado por: Paúl Bonilla.

La segunda página contiene toda la información de los sistemas teniendo una actualización cada dos segundos. A diferencia de la primera página no permite introducir datos. Cada vez que existe una actualización toda la página se reinicia borrando los datos que el usuario está introduciendo en la central, debido a este problema se crean dos páginas, la primera permite introducir datos sin que las actualizaciones interrumpen su ingreso, la segunda página se actualiza cada dos segundos permitiendo observar cómo evoluciona las variables de interés en el tiempo.

Segunda página

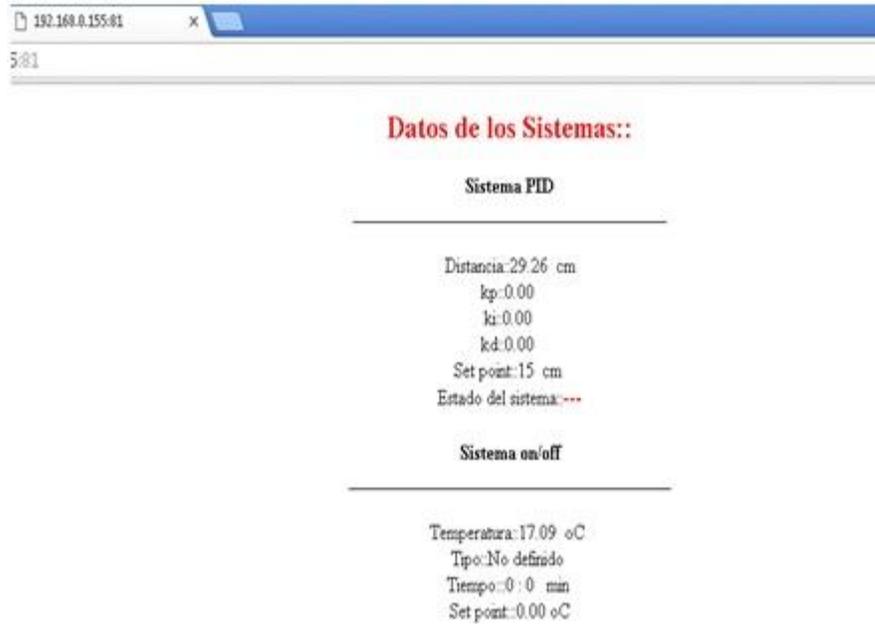


Figura 16. Segunda página en la dirección IP 192.168.0.155:81
Elaborado por: Paúl Bonilla.

La tercera página contiene la información exclusiva de la estación PWM y control On/Off, tiene los valores de ciclo de trabajo de PWM y del estado del relé. El código HTML es muy simple, responde a una línea que contiene los valores mencionados. Cuando la estación del proceso de tintura necesita saber qué hacer con los actuadores ejecuta una petición a esta página en la dirección IP 192.168.0.155:9112.

Tercera página

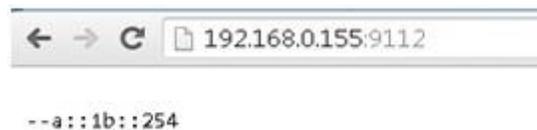


Figura 17. Tercera página en la dirección IP 192.168.0.155:9112
Elaborado por: Paúl Bonilla.

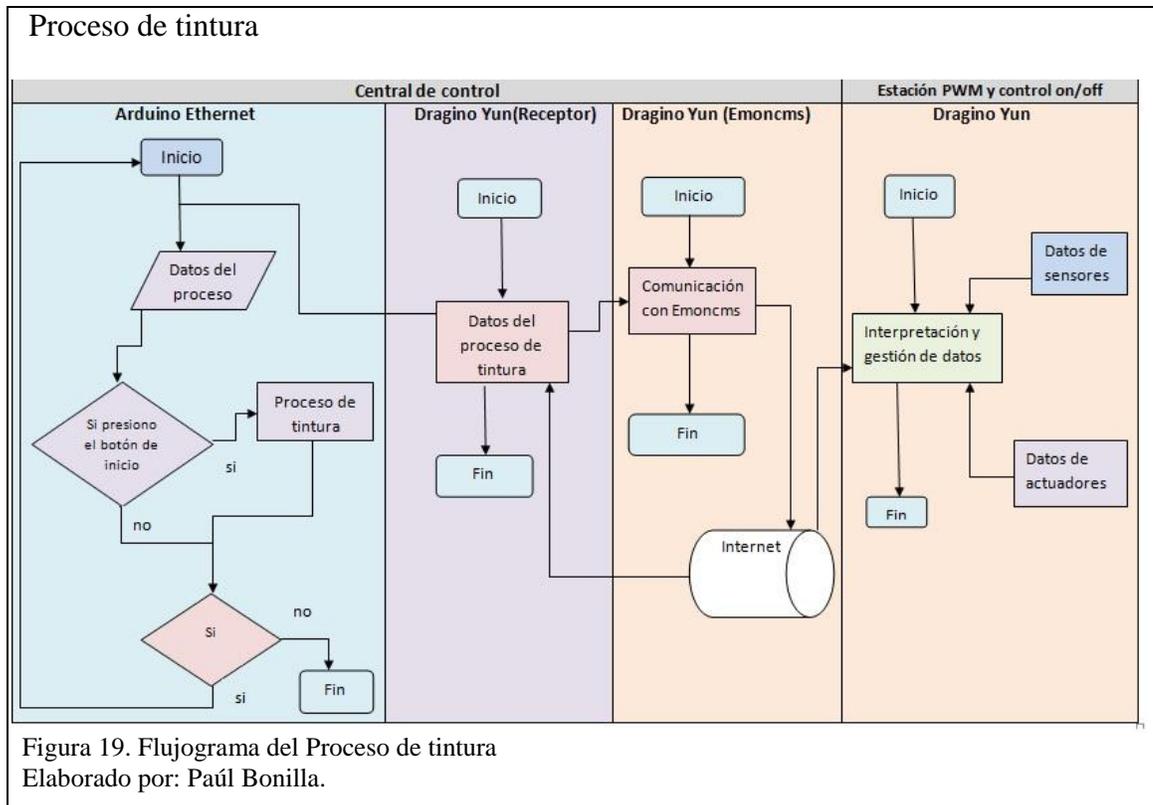
La cuarta página contiene la información exclusiva de la estación PID, la estación remota PID hace peticiones a la cuarta página para consultar el ángulo del servomotor, su respectiva dirección IP es 192.168.0.155:9113.



3.9. Algoritmos de control del proceso de tintura

La variable medida en el proceso de tintura no es tan volátil ante una acción, mantiene una cierta estabilidad temporal. Se puede definir al proceso de tintura como un sistema de reacción lenta. La central de control siempre está recibiendo el valor de temperatura del proceso de tintura, sin embargo esta actualización depende del desempeño de la red. Por el mismo medio de comunicación están circulando todos los datos que requieren los diferentes clientes, servidores y usuarios, es natural que exista un retardo considerable.

La respuesta de la red, incluyendo los retardos permite controlar el proceso de tintura sin desarrollar un algoritmo de compensación. A continuación se detalla el algoritmo efectuado por la central de control y la estación de tintura (Ver Fig. 19).



La Central de control recibe el valor de temperatura del proceso de tintura que es comparado con el valor de referencia para obtener el error. Dependiendo si el error es positivo o negativo, el relé encenderá o apagará el caudín con un tiempo de conmutación que no permitirá un desgaste de las partes mecánicas del relé.

PWM actuará dependiendo del error, el ciclo de trabajo del ventilador es directamente proporcional al error positivo, si el error es negativo el ventilador se apagará.

Se crea un temporizador para controlar el valor referencial que aumenta según el tiempo del proceso (Ver Tabla 5). También existe un segundo temporizador que controla el tiempo de conmutación relé. Cuando el proceso de tintura termine, automáticamente el ciclo de trabajo del PWM llegara a 100 % para enfriar el caudín hasta que llegue a una temperatura menor a los 25°C.

3.10. Control del proceso PID en la red

En el proceso Ball & Beam la variable medida presenta una elevada sensibilidad ante una acción en el sistema. Para sistemas de este tipo se requiere de una elevada frecuencia de muestreo, la variable medida cambia drásticamente en valores menores al segundo. La respuesta del actuador no equilibra el sistema por los retardos introducidos por la red.

Entre cada actualización de la distancia de la pelota en el proceso Ball & Beam, se pierde información del comportamiento del sistema. No existe un tiempo específico que garantice la recepción de datos en la central de control. También existe un retardo en el receptor Dragino Yun, ya que recibe los datos de las dos estaciones. Para mejorar la recepción de la información del sensor de distancia inteligente Ethernet, se incluye un receptor en la central de control, se incrementa un Dragino Yun más para recibir exclusivamente la información de la estación PID.

3.10.1. Algoritmos del proceso PID en la red

Los problemas presentes por los retardos introducidos por trabajar en una red no determinista, tienen un grave efecto en la estabilidad de la estación PID. Se considera un evento cada vez que un dato llega a la central de control, durante este evento se introduce esta variable en el algoritmo del PID. Cada evento no tiene un período establecido, dependiendo de la eficiencia de la red la respuesta puede llegar en diferentes intervalos de tiempo (Control por Eventos). Las diferentes muestras de la estación PID no son establecidas por el paso del tiempo, sino por una acción asíncrona propia de la red de comunicación.

A nivel de red y hardware no existe alguna solución que elimine los retardos, a nivel de software se puede introducir algoritmos de control que definan que hacer mientras no existe información del sistema. Se deben analizar las propiedades físicas que influyen sobre la variable medida (Ver Fig. 20).

Sistema Ball & Beam

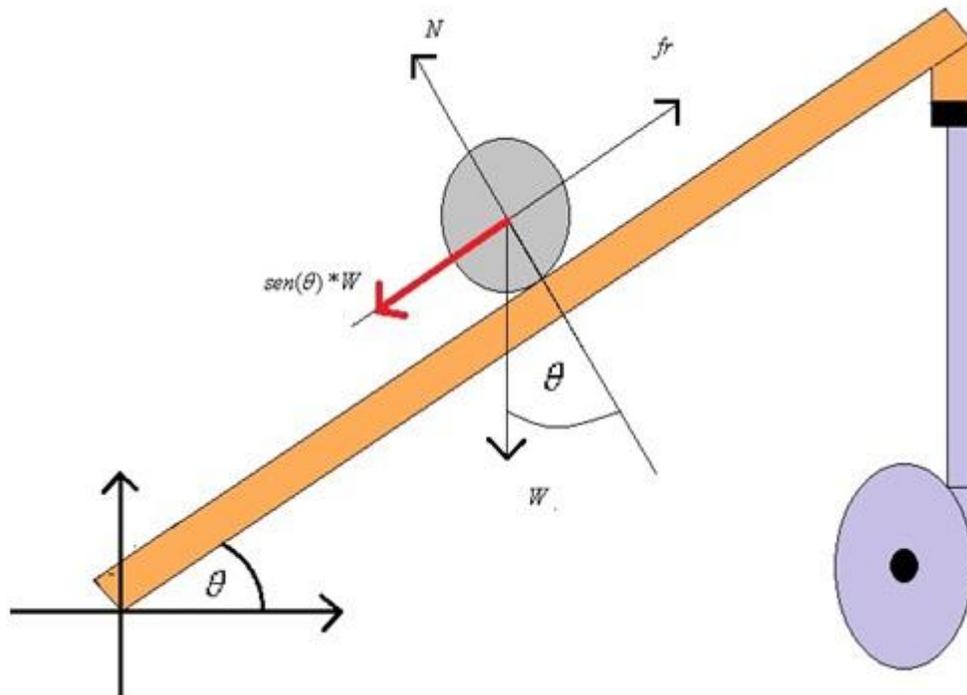
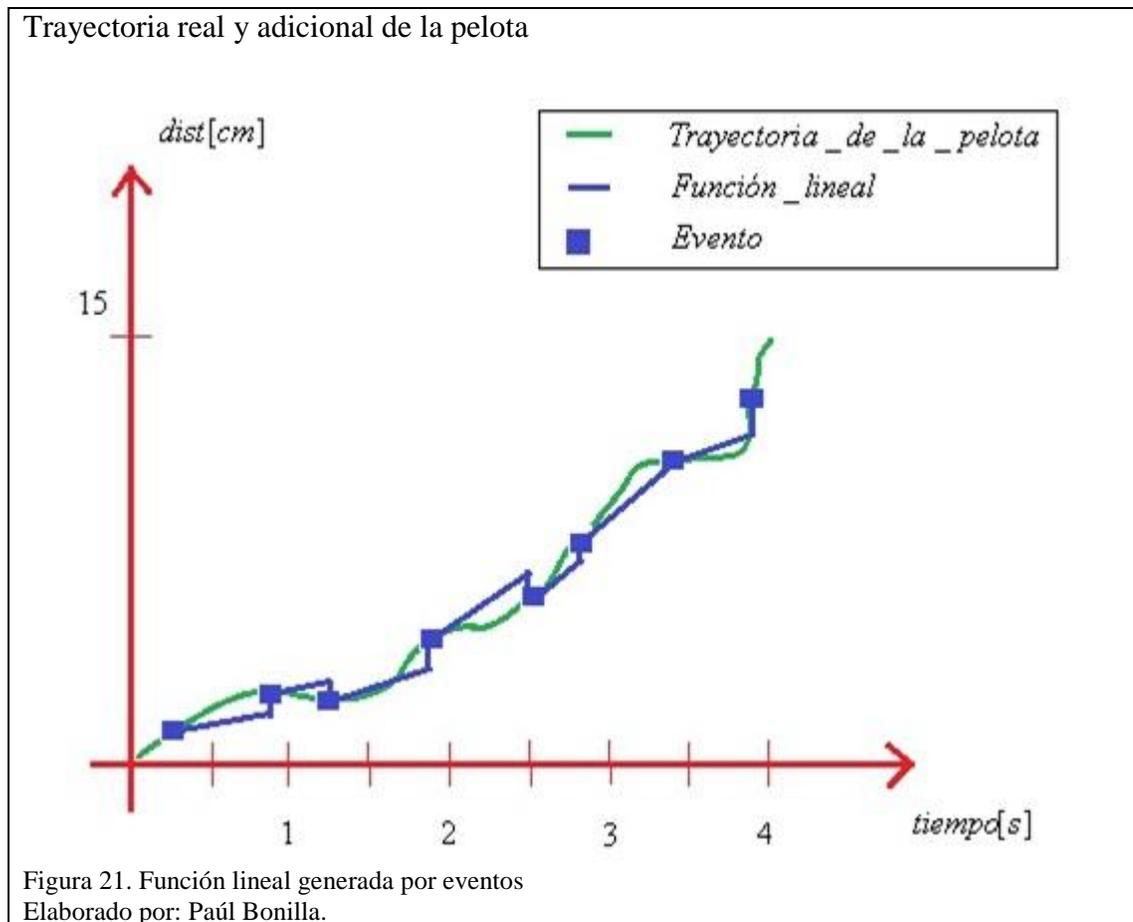


Figura 20. Diagrama de fuerzas en Ball & Beam
Elaborado por: Paúl Bonilla.

Asumiendo una mínima fuerza de rozamiento por las características físicas de la maqueta de Ball & Beam, la pelota tiende a rodar por la inclinación del plato, el servomotor y los ejes definen la variación de este ángulo. La fuerza que ejerce el movimiento sobre la pelota tiene dos componentes, el peso de la pelota que es una constante y el ángulo de giro del servomotor en el que se ejerce directamente la repuesta del controlador.

La repuesta del sistema ante una acción tiende a dar una intuición de lo que puede pasar en un futuro sin necesidad de tomar una muestra del sistema. Se debe programar ciertos criterios de las características físicas del sistema y programarlas en el controlador. Los valores intuitivos llenarán los espacios que existen mientras la red no tiene una respuesta, es decir, cada evento desplegará una serie de cálculos para establecer una respuesta adicional.

Dentro de la programación se incluye un contador que vuelve a cero cada vez que llega un dato de la estación PID. El algoritmo PID solo funciona cuando el contador es cero, es decir, cuando llega un dato legítimo del sensor de distancia de la estación PID. Durante el tiempo variable entre datos legítimos, se superpone una función lineal que considera los últimos datos del servomotor y evento.



La función lineal que se activa durante el evento se describe en la siguiente ecuación:

$$y = mx + b;$$

$$dis\ tan\ cia_int\ uida = m * contador + dato_dis\ tan\ cia_legitima$$

Ecuación 3

Cuando el evento llega el contador se establece en cero, se toma la distancia legítima para iniciar la función lineal, el contador incrementa hasta que llegue otro evento, estableciendo nuevos datos al algoritmo de control PID.

Para establecer la pendiente de la función lineal, se toma en cuenta el ángulo del servomotor, se estableció un rango de giro de 65° hasta 105° . Al estar el servo motor en un ángulo en un intervalo de 85° a 89° , la pendiente es cero debido a que la fuerza que mueve a la pelota actúa solo en el eje y. La pendiente es positiva y es inversamente proporcional al ángulo del servomotor cuando se encuentra en ángulos menores a 85° . La pendiente es negativa y es directamente proporcional al ángulo del servomotor cuando se encuentra en ángulos superiores de 89° .

Funciones lineales.

```
void datos_falsos(void){
  if (inicio_proceso_pid == 10){
    if (salida >= 89 && salida <=105){
      equilibrio=-0.35;
    }
    if (salida < 89 && salida >=85){
      equilibrio=0.0;
    }
    if (salida >= 65 && salida <85){
      equilibrio=0.08;
    }
    if (t_dato_original % 35 == 0 ){

      distancia_b_falsa = equilibrio*(aumento_falso)+distancia_b;
      control_pid_proceso();
      aumento_falso++;
    }
  }
}
```

Figura 22. Programación de funciones lineales generadas por eventos
Elaborado por: Paúl Bonilla.

El resultado de la función lineal llena los datos faltantes teniendo un criterio de que puede ocurrir en una determinada situación, la distancia falsa introducida presenta un error en comparación con la distancia real, pero la generación de estos datos intuitivos

reducen en gran medida el efecto de los retardos propios de la red. No se recogen todos los datos generados por el contador sino solo ciertos datos programados, al existir un retardo de elevado valor, el contador tendera a introducir una función lineal con resultados fuera de escala.

Para solucionar este problema se crea un segundo contador, que solo acepta ciertos valores bajo criterios de múltiplos.

El algoritmo PID trabaja con el valor legítimo de distancia y con los valores introducidos por la función lineal, el algoritmo PID solo se activa bajo el criterio del segundo contador.

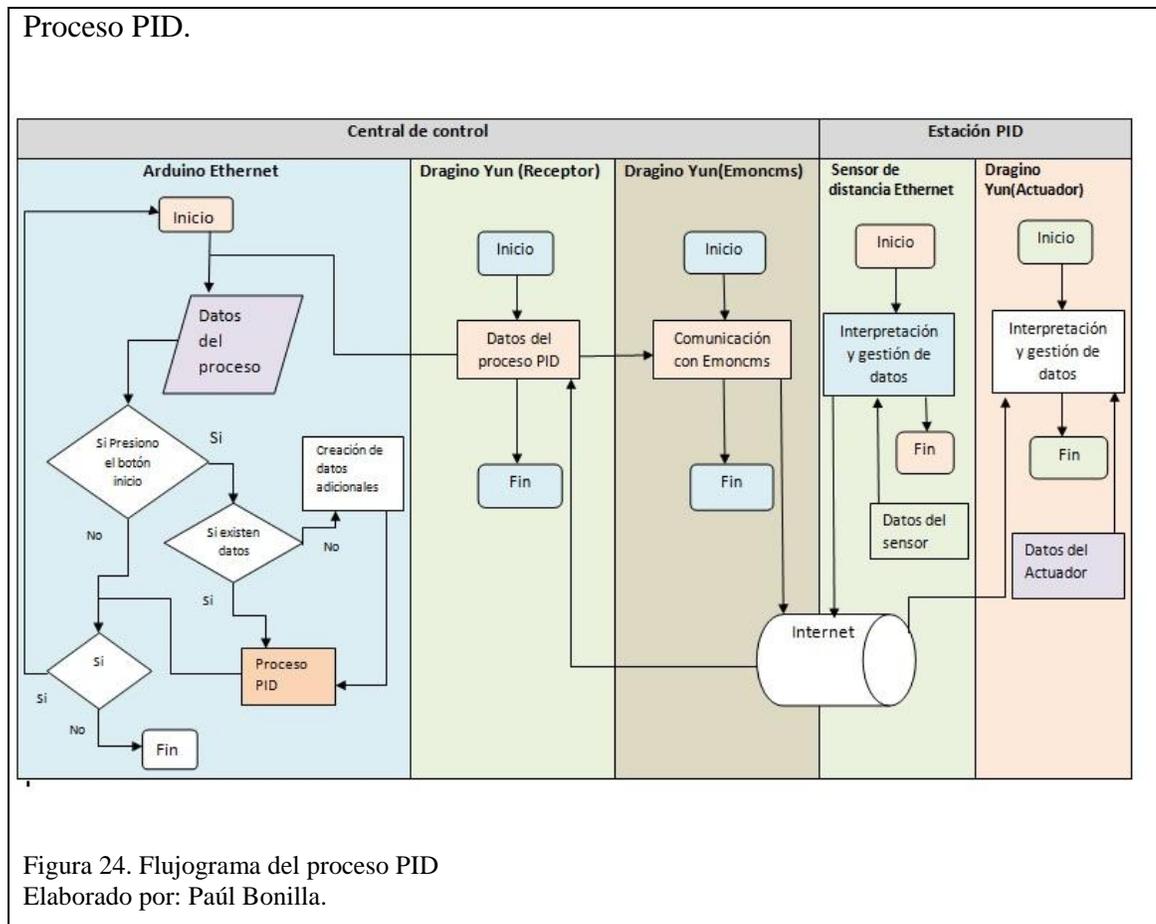
Algoritmo PID.

```
void control_pid_proceso(void){
  if (inicio_proceso_pid == 10){
    error = sp - distancia_b_falsa;
    proporcional=error;
    integral = integral + error;
    derivativo = error - error_anterior;
    pid=((kp_f*proporcional)+(ki_f)*integral)+(kd_f*derivativo));
    error_anterior=error;
    salida = map(pid , -30 , 30 , 105 ,65);
    if (pid >=30){
      pid=30;
    }
    if (pid<=-30){
      pid = -30;
    }
    if (salida >= 105) {
      salida = 105;
    }
    if (salida <= 65) {
      salida = 65;
    }
  }
}
```

Figura 23. Programación de algoritmo PID en Arduino
Elaborado por: Paúl Bonilla.

La calibración del sistema se la realiza observando su comportamiento al introducir los diferentes parámetros previamente establecidos, se debe de tener en consideración que la respuesta de la red es siempre variable, siempre existe un error aceptable entre el valor real de la distancia y los introducidos por la función lineal, pero el objetivo de las respuestas creadas es dar un criterio o predicción al controlador de lo que podría pasar si ejecuto una acción.

Los diferentes valores del algoritmo PID se explican con más detalle en el capítulo 4 al obtener los valores de retardos de la red, observando el comportamiento del sistema con diferentes valores.

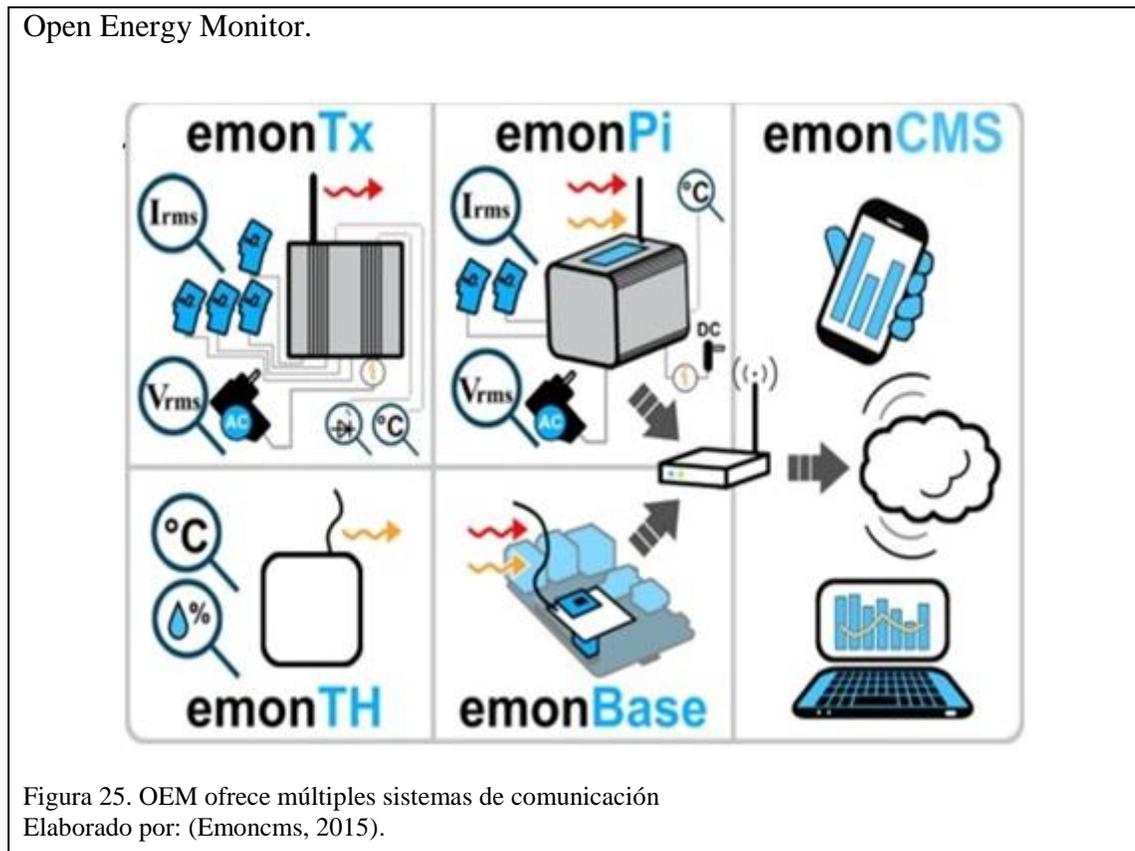


3.11. Dragino Yun como receptor de datos

La central de control tiene dos Dragino Yun exclusivos para la recepción de datos provenientes de cada estación, es decir, los dos son programados como clientes que continuamente están haciendo peticiones a las respectivas direcciones IP de las estaciones de proceso de tintura y PID. Los diferentes datos son enviados a Arduino Ethernet vía comunicación Serial.

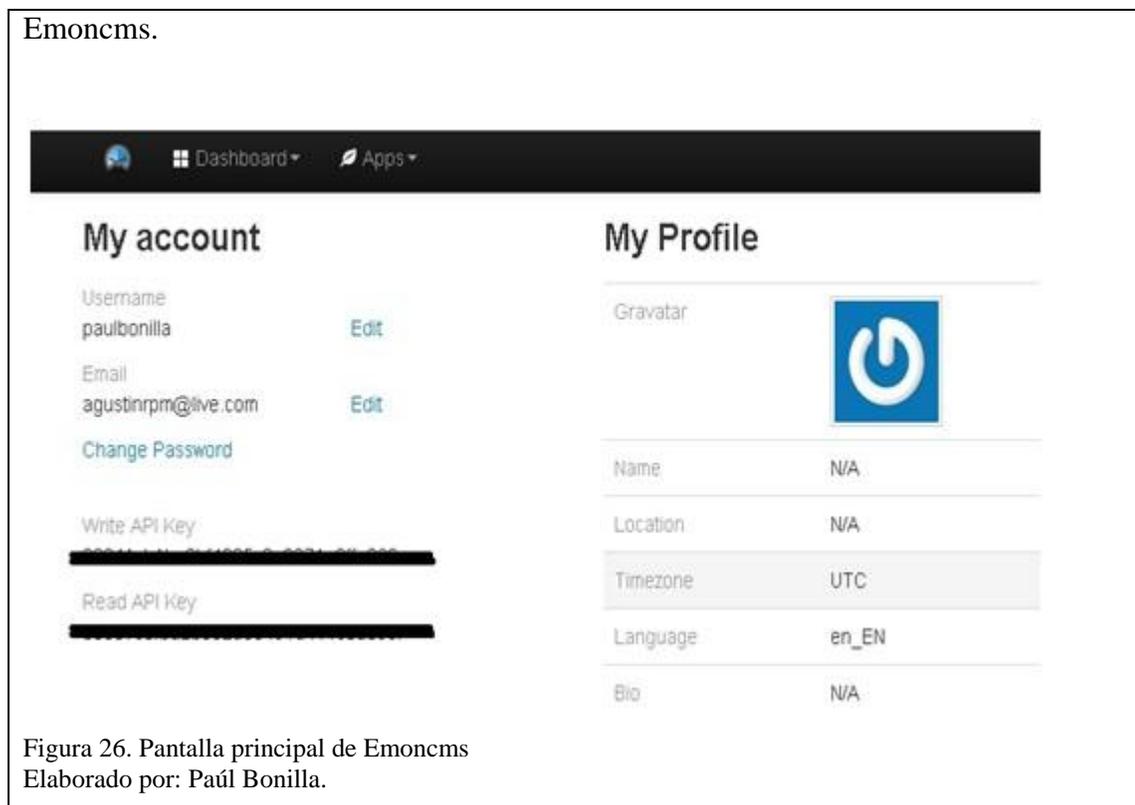
3.11.1. Open Energy Monitor

Open Energy Monitor es una herramienta capaz de monitorizar y almacenar diferentes tipos de variables, la principal ventaja es que es de código abierto. Desarrollado por Glyn Hudson y Trystan Lea, se diseñó como un sistema modular que trabaja en distintas plataformas para el análisis de datos, incorporando elementos propios de hardware.



Emoncms es una aplicación Web para el almacenamiento y visualización de datos, pudiendo ser instalado en un servidor local o conectarse en un servidor externo en la red. Las herramientas gratuitas que ofrece Emoncms son suficientes para el desarrollo de este proyecto, existen otros beneficios que se adquieren si cancela una cierta cantidad de dinero.

Se ingresa a la página web emoncms.org, se introduce un nombre de usuario y contraseña para adquirir una cuenta. En la página principal Emoncms te otorga nuevas claves para poder gestionar datos desde diferentes dispositivos o nodos (Ver Fig. 26).



Api key son claves que Emoncms otorga, y deben ser configuradas en los dispositivos que trabajaran con la aplicación.

3.11.2. Programación en Emoncms

Emoncms permite crear DashBoard, que gráfica variables previamente establecidas desde los diferentes nodos de la red. Se puede configurar las diferentes dimensiones de pantallas, variable, escalas y aspectos estéticos (Ver Fig. 27).

Cuando un dispositivo empieza a enviar datos, automáticamente en la venta de inputs aparecerán los valores de las variables de interés, mostrando el intervalo de actualización (Ver Fig. 28).



Entradas en Emoncms.



Node	Key	Name	Process list	last updated	value
0	distancia			6.9 days ago	9
0	Temp			6.9 days ago	19

Figura 28. Visualización de datos de entrada en Emoncms
Elaborado por: Paúl Bonilla.

3.11.3 Dragino Yun como cliente de Emoncms

Dragino Yun puede establecer comunicación con Emoncms, enviando una petición a un servidor a través de la red. Se programan las dos claves que Emoncms otorga para la transferencia y gestión de datos (Ver Fig. 29).

Claves de Emoncms en Dragino Yun

```
#include <Process.h>

#define APIKEY      "Write API KEY"
#define USERAGENT  "Read API Key"
```

Figura 29. Programación de claves de Emoncms en Arduino
Elaborado por: Paúl Bonilla.

Por razones de seguridad las claves de Emoncms han sido cambiadas o tachadas. La información de los dos Draginos Yun receptores es enviada vía comunicación Serial al Dragino que trabaja como cliente de Emoncms. Las variables son almacenadas en una variable String con dos identificadores para diferenciar los datos.

Se crean dos variables String adicionales, la variable apiString se coloca las clave que Emoncms otorga a los dispositivos para la gestión de datos, en la variable url se coloca el url de la pagina web (Ver Fig. 30). En la variable String url se encadenan todas las variables String previamente establecida quedando como resultado una petición que tiene clave, datos, petición Post (Ver Fig. 31).

Petición al servidor de Emoncms.

```
void sendData() {  
  String apiString = "apikey=";  
  apiString += APIKEY;  
  String url = "http://emoncms.org/api/post?";  
  url += "json=";  
  url += dataString;  
  url += "&";  
  url += apiString;  
  Process emoncms;  
  Console.print("\n\nSending data... ");  
  emoncms.begin("curl");  
  emoncms.addParameter("-g");  
  emoncms.addParameter(url);  
  emoncms.run();  
}
```

Figura 30. Programacion de la petición al servidor de Emoncms en Arduino
Elaborado por: Paúl Bonilla.

Petición de Dragino(Emoncms)



Figura 31. Los diferentes datos son enviados en la petición al servidor de Emoncms
Elaborado por: Paúl Bonilla.

La librería process de Arduino Yun trabaja en el procesador de Linux, se inicia el proceso Emoncms, el comando “curl” permite obtener datos de cualquier servidor web

y ponerlo a nuestra disposición para el trabajo. Las gráficas generadas por Emoncms son añadidas a la página principal programada en Arduino Ethernet (Ver Fig. 15).

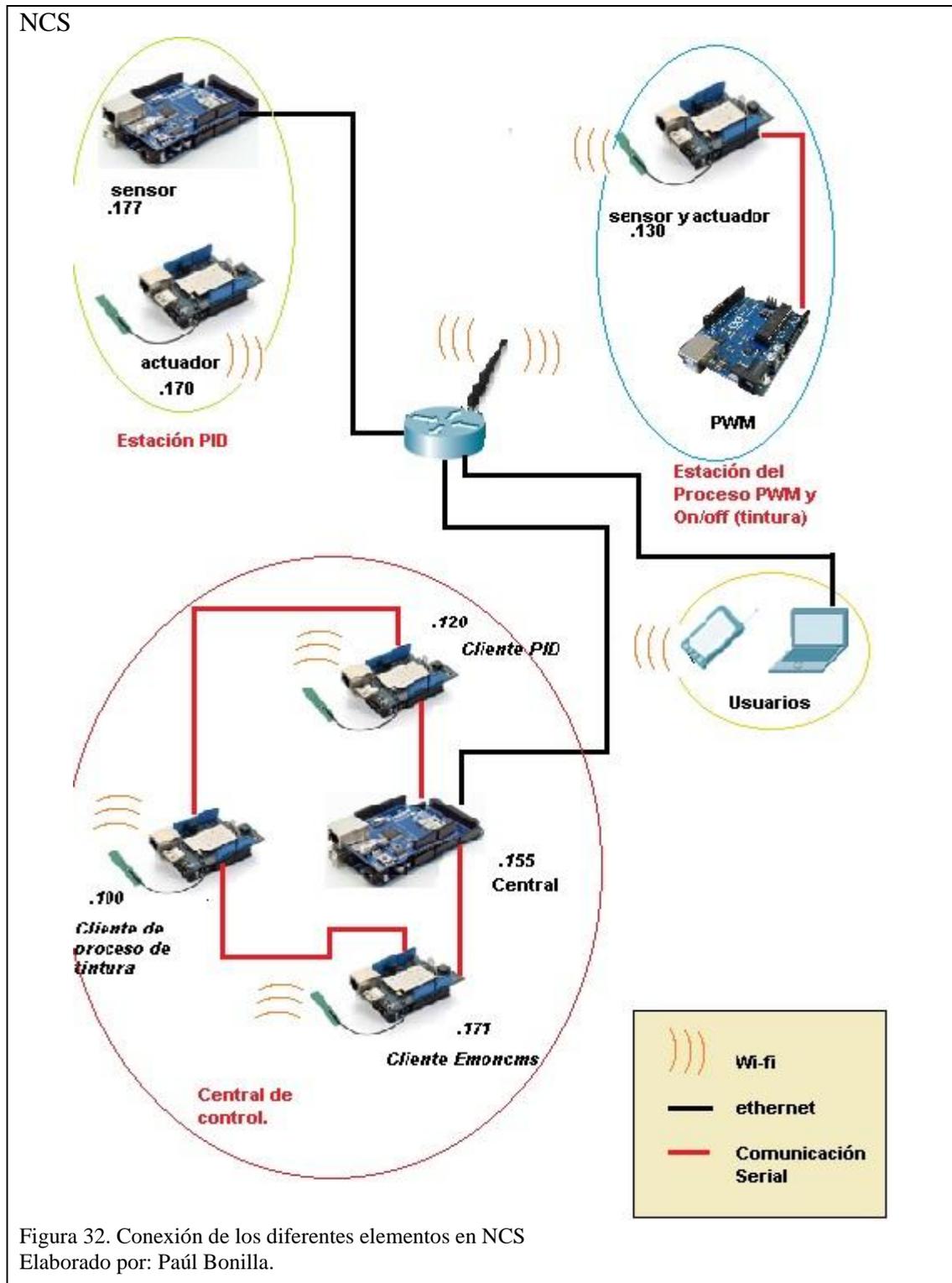


Tabla 6.
Características de los dispositivos en la red.

Central de control				
Elemento	Dirección IP	Dirección Mac	Función	Dirección IP con la que se comunica
Arduino Ethernet	192.168.0.155:80 192.168.0.155:81 192.168.0.155:9112 192.168.0.155:9113	DE-AD-BE-EF-FE-ED	Servidor, genera las páginas: Principal(usuario), secundaria (usuario), datos de actuador (PWM -control On/Off), actuador (PID). Recibe la información de los receptores via comunicación Serial.	192.168.0.170 192.168.0.130
Dragino Yun	192.168.0.120	A8-40-41-13-DA-28	Cliente, hace peticiones al servidor de la estación PID. Envía datos via comunicación Serial a Arduino ethernet y al Dragino Yun(cliente de Emoncms).	192.168.0.177:9911
Dragino Yun	192.168.0.100	A8-40-41-13-DE-D8	Cliente, hace peticiones al servidor de la estación PWM y control On/off. Envía datos via comunicación Serial a Arduino ethernet y al Dragino Yun(cliente de Emoncms).	192.168.0.130
Dragino Yun	192.168.0.171	A8-40-41-14-4E-84	Cliente, se comunica con el servidor de Emoncms para almacenar y visualizar los datos de las dos estaciones.	Servidor de Emoncms
Estación PID				
Arduino Ethernet	192.168.0.177:9911	AE-AD-BE-EF-FE-ED	Servidor, genera la página que contiene la información del sensor de la estación PID.	192.168.0.120
Dragino Yun	192.168.0.170	A8-40-41-13-8E-A0	Cliente, hace peticiones a la central de control	192.168.0.155:9113
Estación PWM y Control On/Off				
Dragino Yun	192.168.0.130	A8-40-41-13-DE-B0	Servidor, genera la página que contiene la información del sensor de la estación PWM y Control On/Off. Cliente hace petición a la central de control	192.168.0.155:9112 192.168.0.100

Nota. Elaborado por : Paúl Bonilla

Tabla 7.
Conexión entre dispositivos.

Central de control			
Elemento	Terminal propia	Función	Terminal o elemento con el que se conecta
Arduino Ethernet (192.168.0.155)	RX1	Recibe los datos de la estación PID	Dragino Yun (192.168.0.120), TX1 envía los datos de la estación PID
	RX2	Recibe los datos de la estación PWM y Control On/Off	Dragino Yun(192.168.0.100), TX2 envía los datos de la estación PWM y Control On/Off
Dragino Yun (192.168.0.120)	TX1	Envía datos para el Servidor Central	Arduino Ethernet (192.168.0.155), RX1 recibe los datos de la estación PID
	TX2	Envía datos para el cliente Emoncms	Dragino Yun (192.168.0.171), RX1 recibe los datos de la estación PID
Dragino Yun (192.168.0.100)	TX2	Envía datos para el Servidor Central	Arduino Ethernet (192.168.0.155), RX2 recibe los datos de la estación PWM y Control On/Off.
	TX1	Envía datos para el cliente Emoncms	Dragino Yun (192.168.0.171), RX2 recibe los datos de la estación PWM y Control On/Off.
Dragino Yun (192.168.0.171)	RX1	Recibe los datos de la estación PID	Dragino Yun(192.168.0.120), TX2 envía datos de la estación PID
	RX2	Recibe los datos de la estación PWM y Control On/Off	Dragino (192.168.0.100), TX1 envía datos de la estación PWM y Control On/Off.
Estación PID			
Arduino Ethernet 192.168.0.177:9911	A1	Entrada analógica	Sensor de distancia
	5v	5 Voltios	
	Gnd	GND	
Dragino Yun (192.168.0.170)	8	Salida para el servomotor	Servomotor
	Gnd	GND	
Regulador de voltaje 7805	Vout	Energiza al servomotor	Servomotor
	Gnd	GND	Dragino Yun (192.168.0.170), GND
Estación PWM y Control On/Off			
Dragino Yun (192.168.0.130)	TX1	Envía datos para PWM	Arduino PWM, RX1 recibe los datos de PWM
	52	Salida para rele	Rele
	A1	Entrada analógica	Sensor de temperatura
	5V	5 Voltios	
	Gnd	GND	
Arduino PWM	RX1	Recibe los datos de PWM	Dragino Yun(192.168.0.130), TX1 envía datos PWM
	9	Salida PWM	Ventilador

Nota. Elaborado por : Paúl Bonilla

CAPÍTULO 4.

PRUEBAS Y RESULTADOS

4.1. NCS como servicio

Cualquier dispositivo que este en la capacidad de conseguir una dirección IP por el router de la red, puede ser un usuario de la aplicación de control.

NCS como servicio.

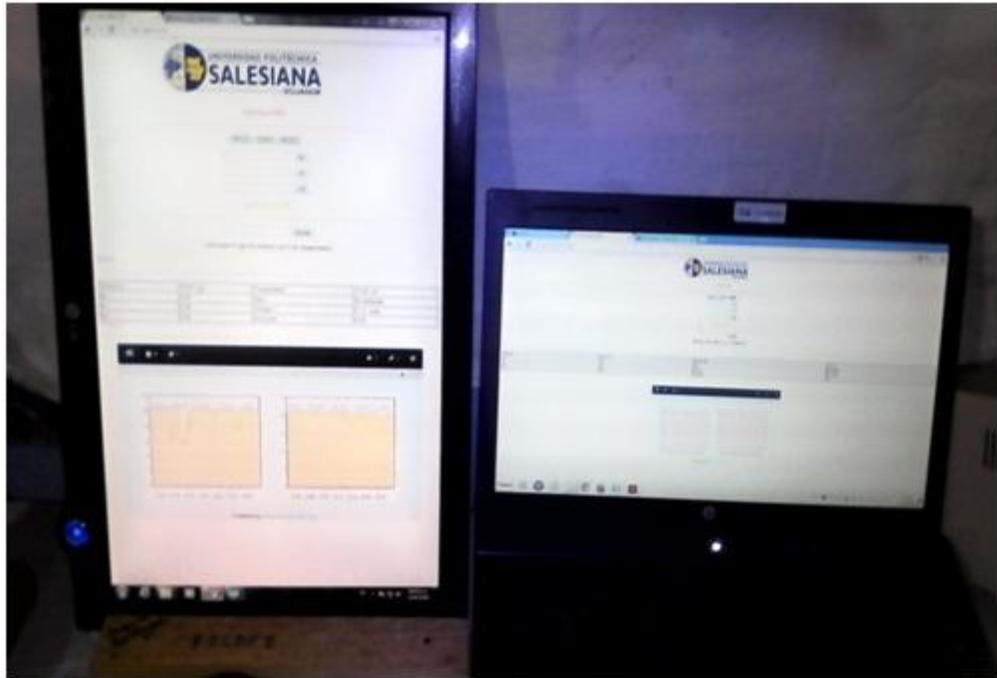
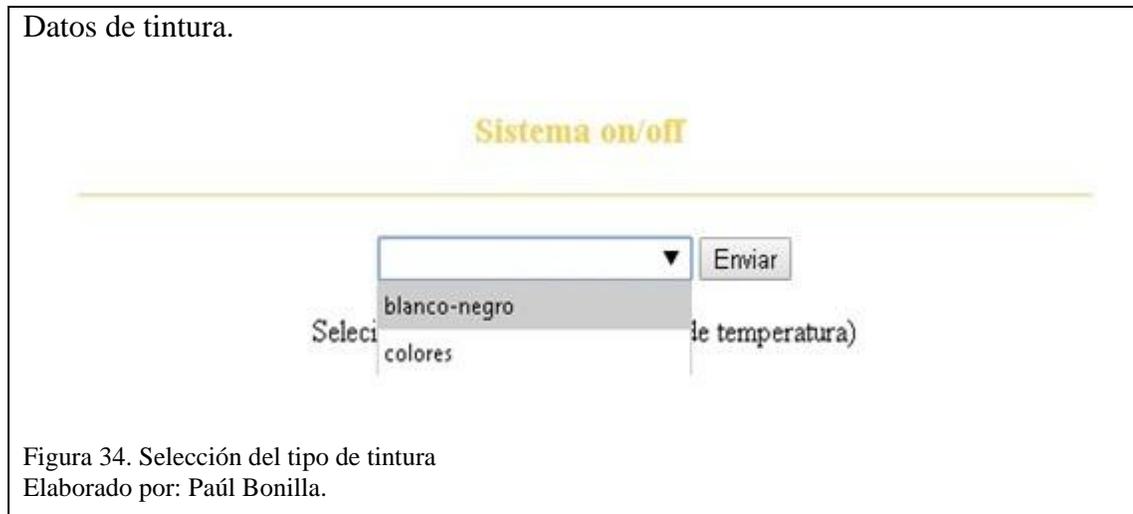


Figura 33. Usuarios de NCS
Elaborado por: Paúl Bonilla.

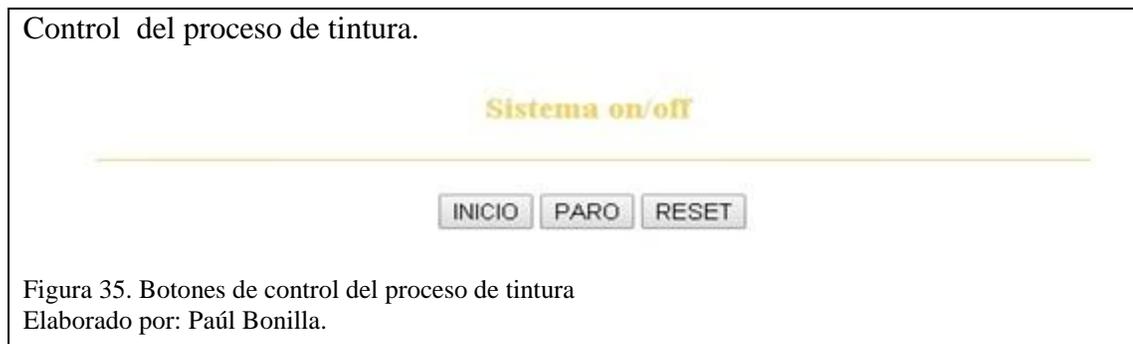
Los dispositivos pueden comunicarse con los diferentes sistemas de control conectándose vía wifi o Ethernet con el router de la red. Las gráficas que genera Emoncms no tiene soporte para trabajar en Sistema Operativo Android, en un dispositivo móvil aparece los cuadros de Emoncms en blanco.

4.2. Sistema de control PWM y Control On/Off en red

El usuario puede escoger entre los dos tipos de tintura, blanco-negro y colores.



Seleccionado el tipo de tintura, se presenta automáticamente tres botones: Inicio, Paro, Reset.



Al presionar en la tecla de inicio el proceso empieza, la pantalla secundaria indica los parámetros establecidos actualizándose cada dos segundos.

Pantalla secundaria.

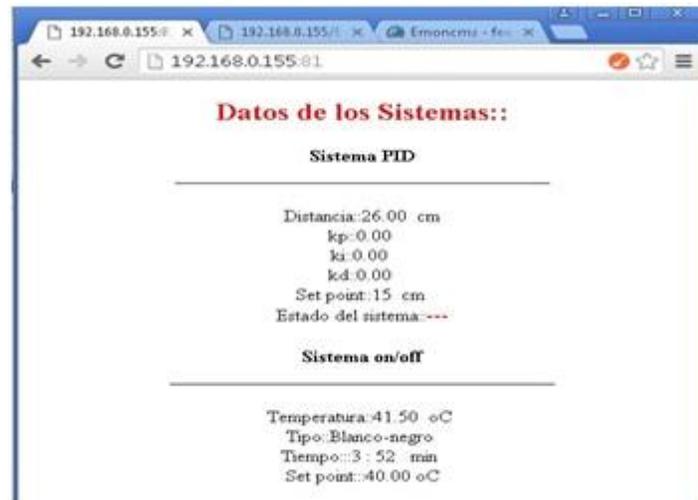


Figura 36. Los datos de los diferentes sistemas es presentado en la pantalla secundaria
Elaborado por: Paúl Bonilla.

El tipo de tintura Blanco-Negro y colores, deben de cumplir con los parámetros establecidos (Ver Tabla 5). Al finalizar el proceso automáticamente el ventilador se enciende a la máxima velocidad para bajar la temperatura del caudín.

Tintura Blanco-Negro.

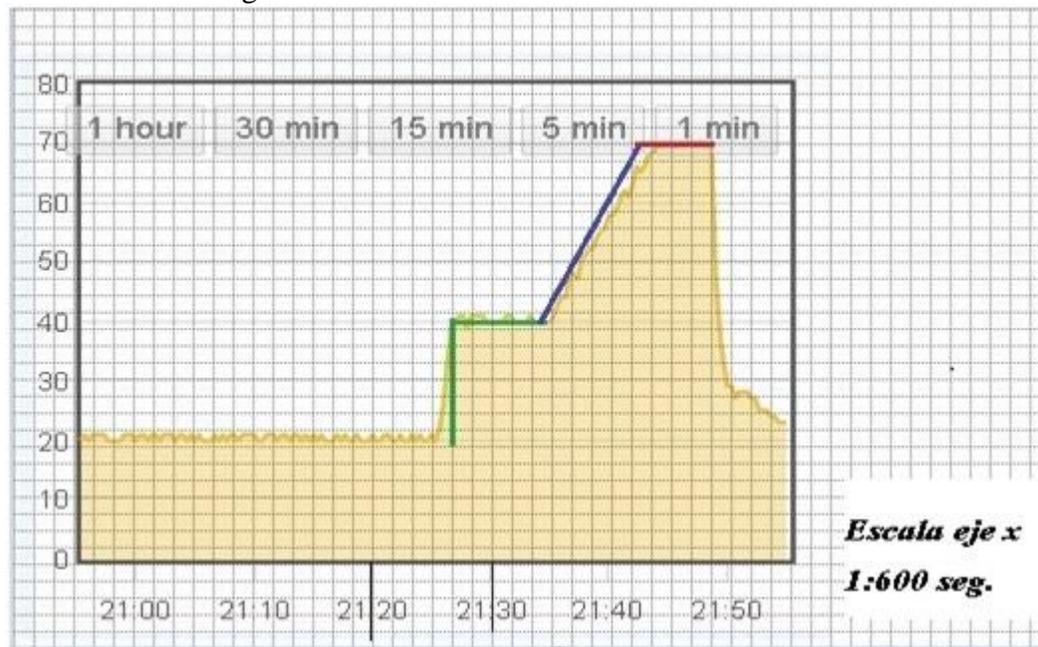


Figura 37. Gráfica de la tintura blanco-negro, obtenida en Emoncms
Elaborado por: Paúl Bonilla.

Tintura Colores.

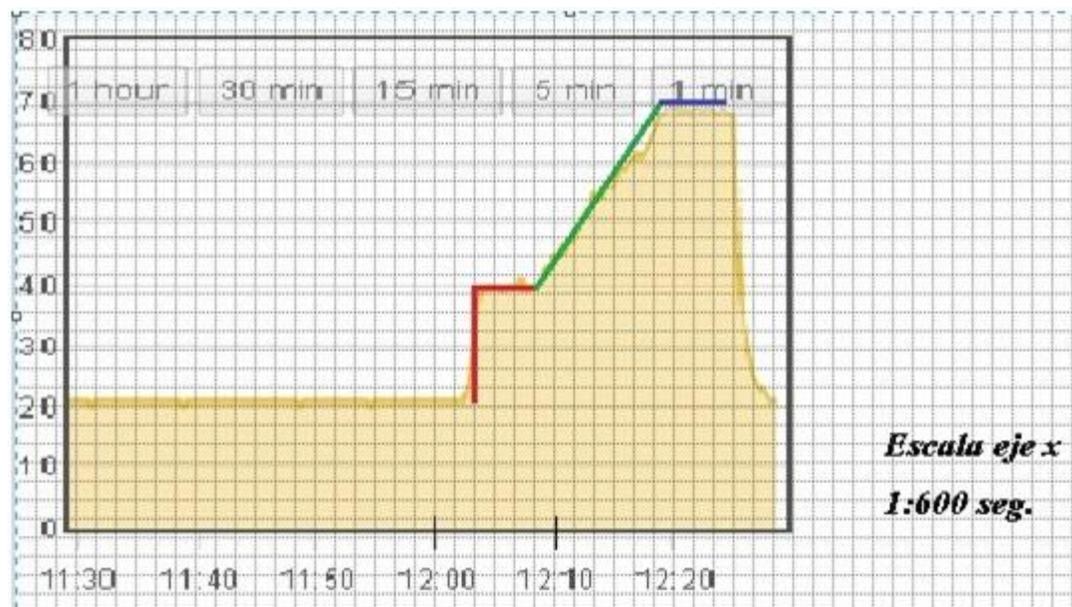


Figura 38. Gráfica de la tintura a colores, obtenida en Emoncms
Elaborado por: Paúl Bonilla.

Con los datos obtenidos en Emoncms, se compara la evolución temporal de la temperatura ideal y medida. Se puede observar la simetría entre las dos gráficas para los dos tipos de tintura (Ver Fig. 37 y Fig. 38).

4.3. Sistema de control PID en red

Para obtener los valores y parámetros del sistema PID, se somete a actuar al proceso por calibración basado en prueba y error, se observa cómo responde el control ante los retardos. Las diferentes lecturas del sensor y los valores introducidos por la función lineal dependen básicamente de la eficiencia de la red.

Se introduce un contador extra en la programación para contabilizar cuanto tiempo se demora cada evento. El programa de la Central de control tiene un retardo de un milisegundo, pero existen una gran cantidad de procesos que no permiten un correcto avance del contador en el tiempo (Ver Fig. 39).

Contador entre eventos



Figura 39. Contador entre eventos en el monitor serial de Arduino
Elaborado por: Paúl Bonilla.

El valor menor que se obtiene de la muestra tomada de los retardos es 47 milisegundos, el valor mayor es 489 milisegundos. En la realidad el tiempo entre cada evento es mucho mayor debido a la programación lineal, existen procesos que demoran el incremento del contador. Se observa como el tiempo entre eventos tiene un valor asíncrono sin ningún patrón que permita su interpretación. En el peor de los casos el contador puede llegar a superar los 1000 milisegundos

Contador vs eventos.

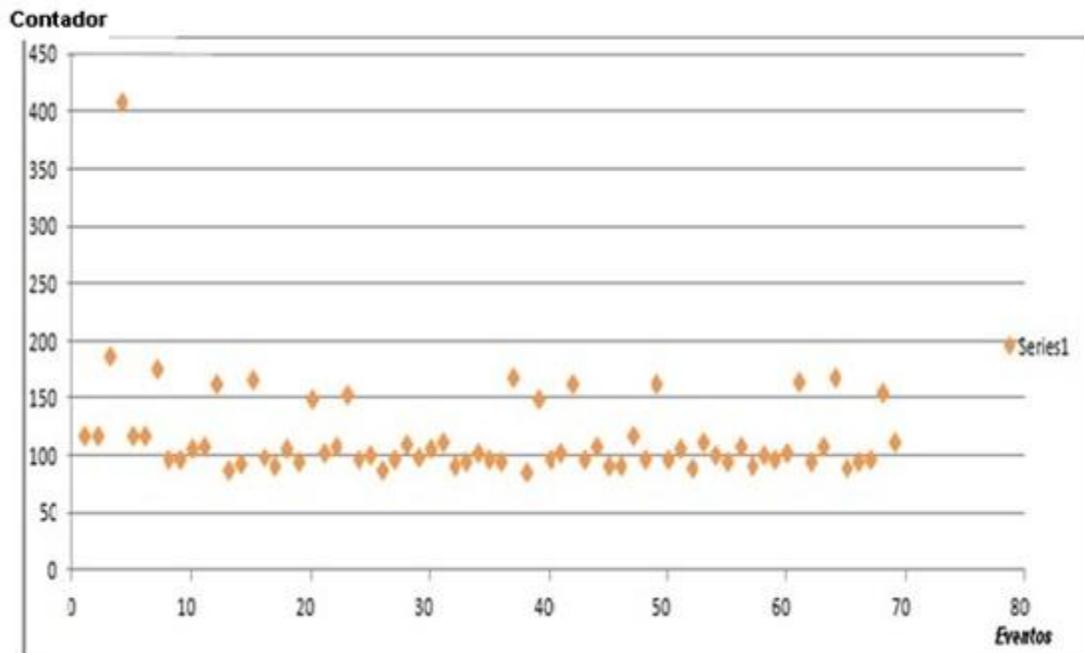


Figura 40. Gráfica de contador vs eventos
Elaborado por: Paúl Bonilla.

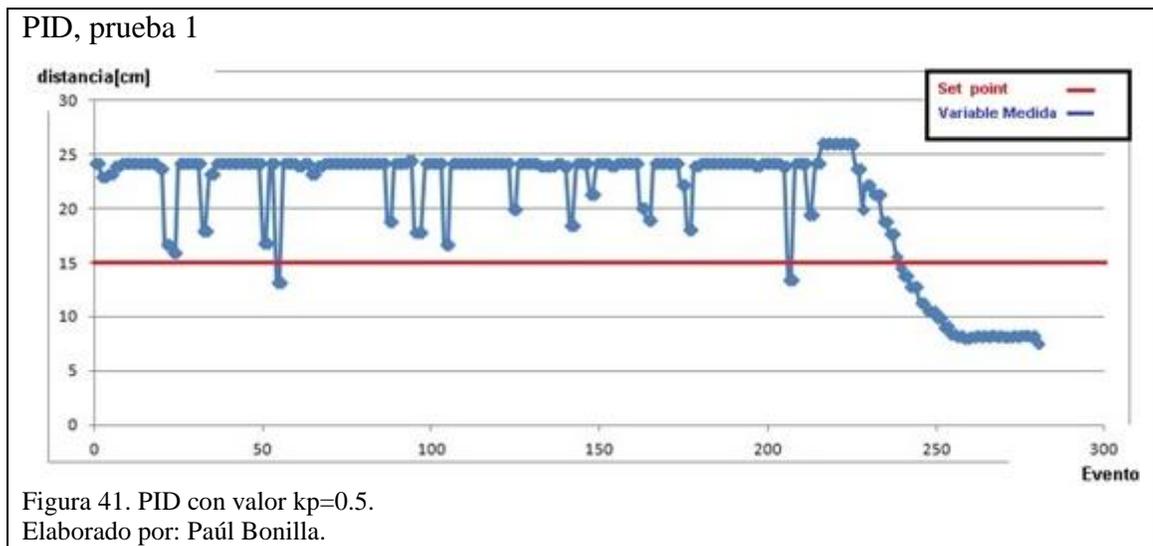
4.3.1. Valores de las funciones lineales

Entendiendo la naturaleza asíncrona del sistema, el proceso es sometido a valores iniciales para analizar el comportamiento del PID con los valores adicionales de las funciones lineales. Se inicia con valores bajos en las pendientes y k_p , el objetivo es ver como las funciones lineales se comportan con los eventos. El criterio para definir el valor de la pendiente, es que los datos generados tengan una coherencia que ayude a definir el comportamiento de la pelota, una tendencia que complemente la ausencia de datos en el PID.

Tabla 8.
Valores del PID en la red.

Valores del PID				
Pendiente			Kp	Figura
Angulo del servomotor(Intervalos)				
[65° - 88°]	[88° - 89°]	[89° - 105°]		
0.01	0	-0.01	0.5	41
0.01	0	-0.01	1	42
0.01	0	-0.01	1.5	43
0.01	0	-0.01	2	44

Nota. Elaborado por : Paúl Bonilla



PID, prueba 2

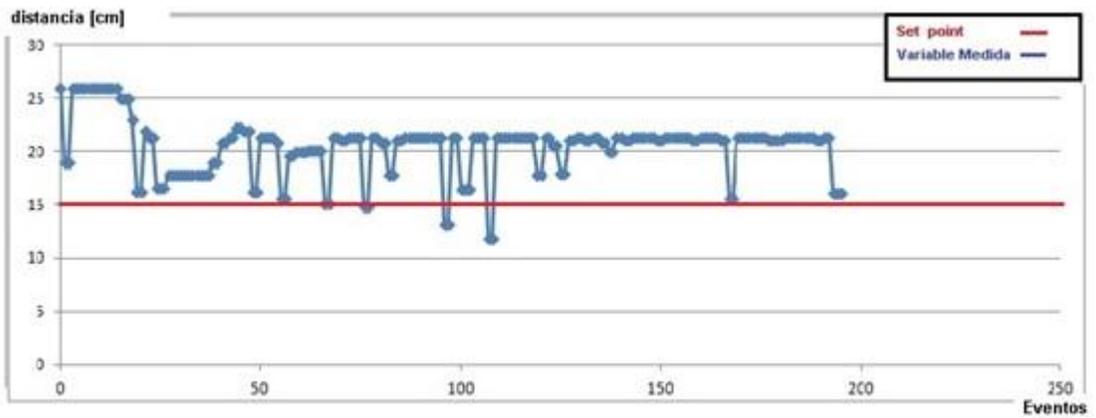


Figura 42. PID con valor $k_p=1$
Elaborado por: Paúl Bonilla.

PID, prueba 3

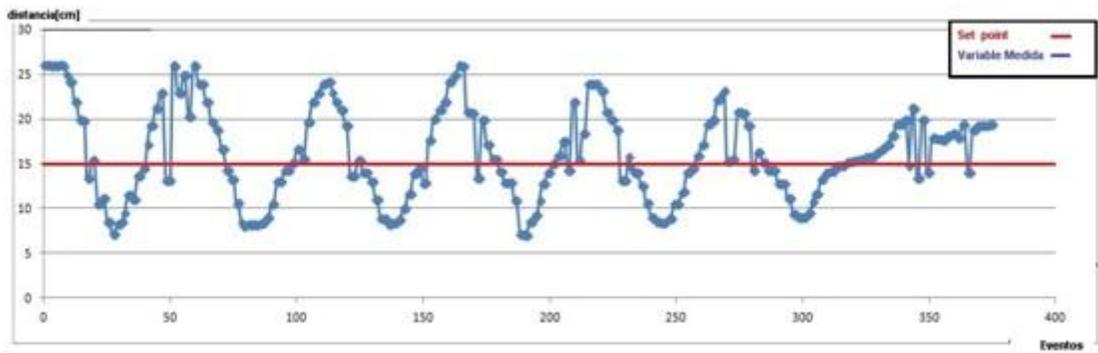


Figura 43. PID con valor $k_p=1.5$
Elaborado por: Paúl Bonilla.

PID, prueba 4

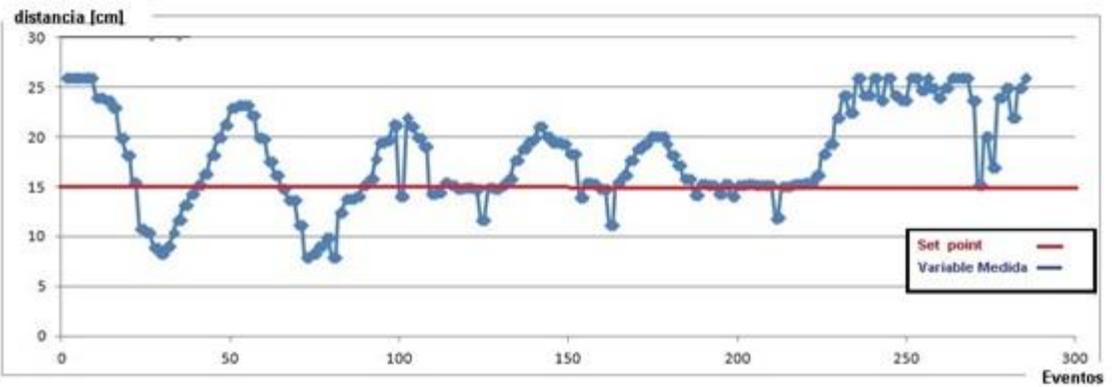


Figura 44. PID con valor $k_p=2$
Elaborado por: Paúl Bonilla.

Las gráficas presentan una información importante del comportamiento de las funciones lineales, todos los valores legítimos y los adicionales de la función lineal son incorporados en la gráfica, se observa una coherencia en su variación. El número de datos que existe entre cada evento varía, depende de la red por su respuesta asíncrona.

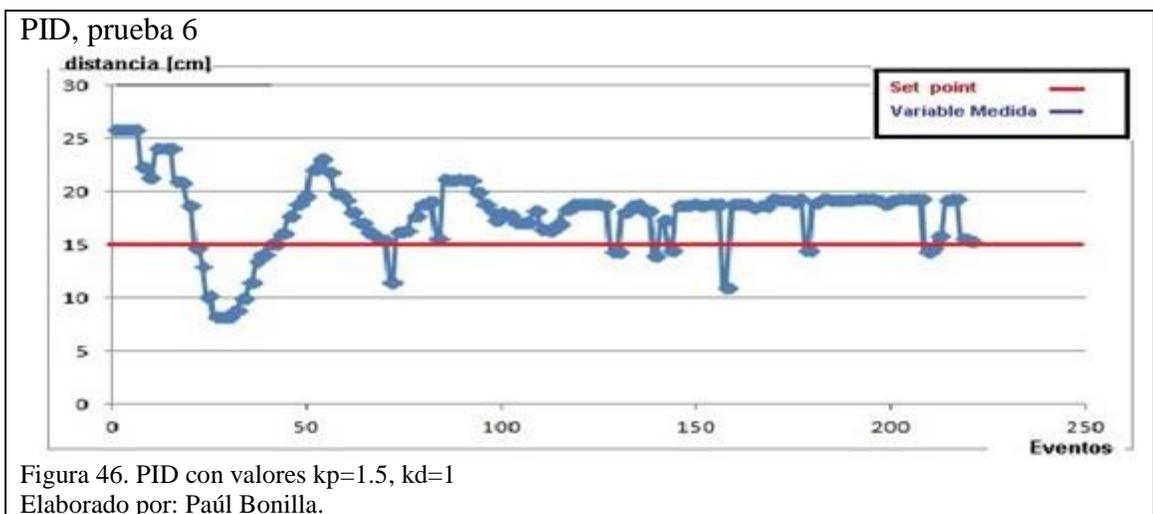
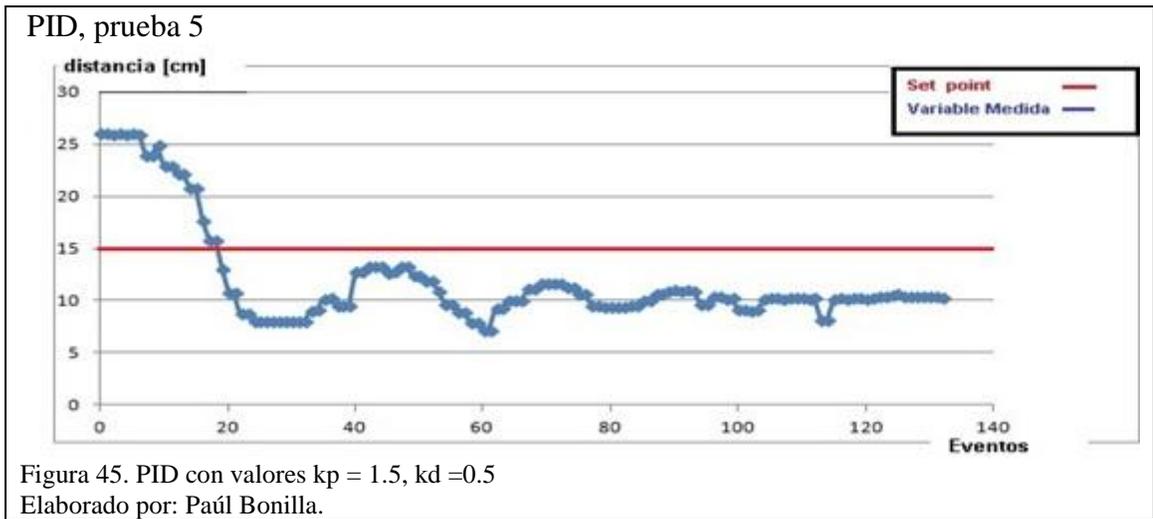
En la gráfica con $k_p=0.5$, se observa que la variable medida no se estabiliza en el valor de referencia o Set Point (15cm), el sistema tiende a presentar un error muy elevado. El coeficiente K_p establecido provoca una reducción de la escala de la acción del actuador, es decir, el valor del ángulo del servomotor generado por el Proporcional, no es suficiente para provocar un sistema en equilibrio (Ver Fig. 41). La gráfica con $k_p=1$, presenta una situación muy similar al sistema con el coeficiente $k_p=0.5$, pero debido al incremento del valor de K_p el error tiende a bajar pero no a equilibrar el sistema (Ver Fig. 42). En la gráfica con valor $k_p = 2$, se observa que el sistema tiene un sobrepulso, es decir, el valor de la escala del ángulo es muy elevado que tiende a desequilibrar al sistema, requiere de un variación de menor valor (Ver Fig. 44).

La gráfica con $k_p= 1.5$, tiende a oscilar la pelota sin estancarla (Ver Fig. 43). Se observa una oscilación que tiende a reducir la amplitud acercándose al Set Point. Existe un error positivo y negativo, en promedio el error se aproximaría a cero. En las otras gráficas se presenta un error constante de un solo signo (Ver Fig. 41 y Fig. 42). El coeficiente $k_p=1.5$ es la mejor opción para empezar a calibrar al sistema.

4.3.2. Coeficientes del algoritmo de control PID

El usuario puede introducir los coeficientes del sistema PID en la página principal, también tiene tres botones: Inicio, Paro, Reset (Ver Fig. 15). El valor de k_p es 1.5 por el análisis hecho previamente, para definir el coeficiente k_d se incorporan varios valores observando la respuesta del sistema. Se debe considerar que k_d es una respuesta a los errores futuros. Su repuesta amortiguará la pelota en la posición de equilibrio, considerando que en la derivada se encuentra los valores de la función lineal introducida por eventos. Se mantiene el coeficiente k_p en 1.5, se define el valor del

coeficiente k_d en 0.5. Se observa como la acción derivativa reduce la amplitud de las oscilaciones, pero la variable medida no se estabiliza en el valor de referencia, se mantiene en un rango mucho más bajo (Ver Fig. 45). Se cambia el valor de k_d a 1.5, se observa que la amplitud de las oscilaciones de disminuye paulatinamente en menos tiempo, pero el sistema no se equilibra en el valor de referencia (Ver Fig. 47). Se cambia el valor k_d a 1, se observa una mayor oscilación, pero tiende a mantenerse cerca del valor de referencia (Ver Fig. 46). k_d se mantiene en 1 para el análisis del sistema.



PID, prueba 7

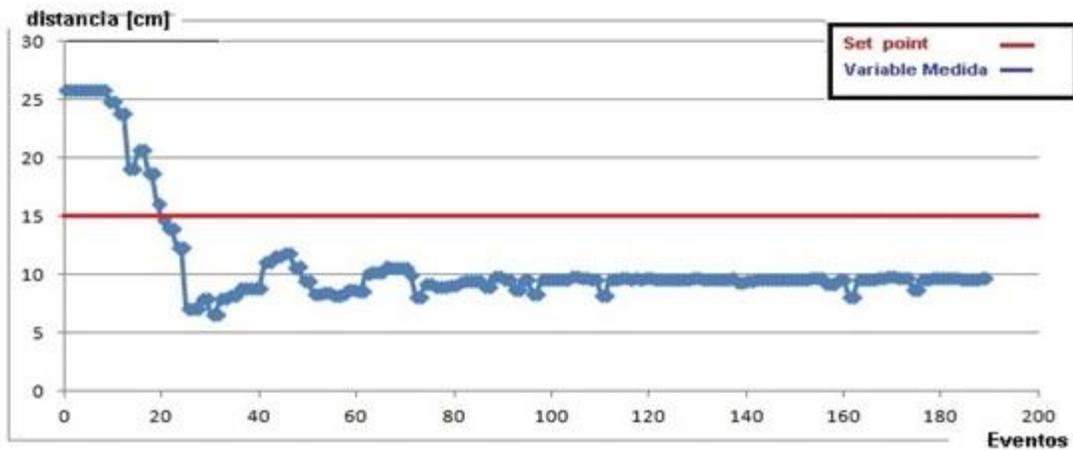


Figura 47. PID con valores $k_p= 1.5$, $k_d=1.5$
Elaborado por: Paúl Bonilla.

Se mantienen los coeficientes k_p y k_d en los valores definidos previamente, la acción integral considera todos los errores pasados, al ser un sistema muy sensible la respuesta integral puede tender a desequilibrar al sistema, se inicia con un valor mínimo de 0.01.

PID, prueba 8

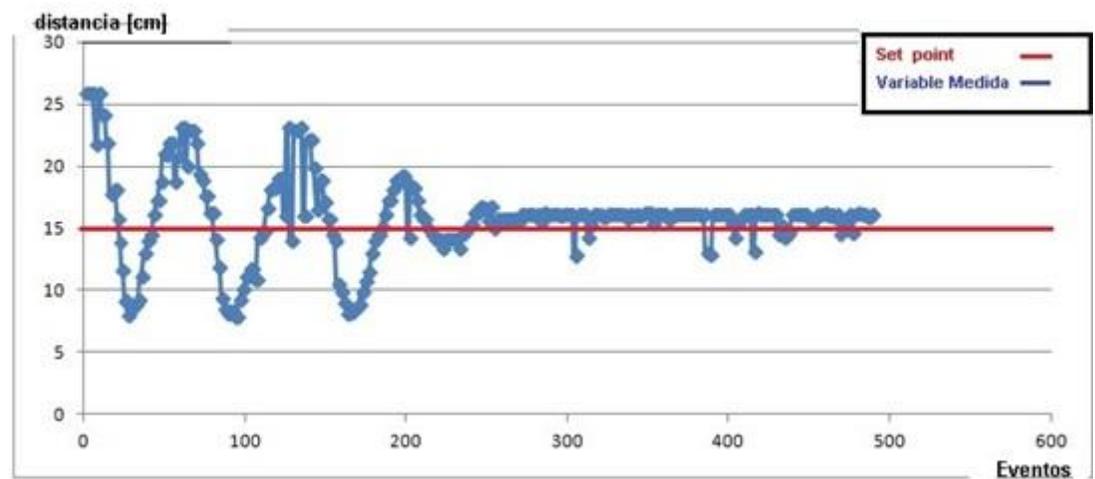


Figura 48. PID con valores $k_p = 1.5$, $k_d = 1$, $k_i = 0.01$.
Elaborado por: Paúl Bonilla.

Se observa como en el sistema tiende a equilibrarse en el valor de referencia o Set Point (Ver Fig. 48). Las oscilaciones incrementan debido a la acción integral que acumula el

error y afecta el resultado de la respuesta del PID, para encontrar los coeficientes y los valores de las funciones lineales, se inicia un proceso de calibración del sistema por experimentación. Se debe interpretar los datos adicionales que se introducen al PID, si el sistema cambia en sus características físicas tales como: los ángulos para el servo, dimensiones de la transmisión, longitud del plato y pelota, cambiar superficie entre pelota y plato, posiblemente los datos adicionales tiendan a ser incorrectos. Si el PID trabaja con datos incorrectos que presentan un error muy elevado el sistema no se equilibrará. En todas las gráficas del proceso PID, la distancia no varía en el tiempo sino en función de los eventos. El tiempo entre eventos es variable pero el sistema se equilibra aproximadamente a los 7 o 15 segundos.

Tabla 9.
Tiempo de estabilización del proceso PID.

Sistema PID	
Tiempo de estabilización	
n° de muestras	tiempo[s]
1	7
2	8
3	6
4	12
5	15
6	12
7	9
8	8
9	15
10	16
Promedio	10.8

Nota. Elaborado por : Paúl Bonilla

4.3.3. Emoncms con control PID

Emoncms actualiza datos en un promedio de 8 a 15 segundos, en control PID este tiempo es insuficiente para llevar un análisis de su respectiva gráfica. Al generar una

petición con servidor externo, se consume mucho tiempo para generar una actualización de datos que permita una mayor frecuencia de muestreo.



CONCLUSIONES

Los NCS y WCS son una nueva propuesta en sistemas de control que presentan un gran número de beneficios e inconvenientes, por que al ser un sistema de control con un protocolo y cableado general que responde los diferentes requerimientos de comunicación de todos los elementos de planta, eliminando hardware y software extra dedicado para cada nivel de la pirámide de automatización. Por el contrario, al implementar una red de comunicación no determinística en un sistema de control, se tiene como resultado un sistema inestable en tiempo real; en los diferentes procesos se puede analizar la influencia negativa que tienen los retardos aleatorios sobre la acción de control, especialmente en el proceso PID.

En el presente proyecto de titulación se implementó dos procesos, aplicados a NCS y WCS los que se pueden dividir en sistemas de reacción lenta y rápida. El Proceso Ball & Beam es un sistema de reacción rápida, debido a que una pequeña variación del actuador cambia drásticamente el valor de la variable medida. La razón de cambio de la variable medida en el controlador pierde información que retarda la acción de control. El proceso de tintura es de reacción lenta, existe un bajo valor de la variación de la variable medida ante las acciones de los actuadores, el controlador tiene el comportamiento casi real del sistema a pesar de los retardos, por lo tanto es más conveniente trabajar con sistemas de reacción lenta con estos sistemas de control.

Los módulos Arduino pueden ser utilizados como nodos de red, que permiten controlar y monitorear los diferentes procesos de planta, también permiten desarrollar interfaces amigables incorporando todas las características de diseño web por el formato HTML, esto se observo porque el usuario puede acceder a diferentes páginas web que le permiten interactuar con los diferentes sensores y actuadores de planta, de forma fácil y dinámica. Cada elemento posee su propia dirección IP que presenta e intercambia información específica de su trabajo.

Los módulos Arduino se pueden configurar dentro de una red de comunicación local, la misma que tiene la facilidad de interactuar con redes externas, por que los diferentes valores de los sensores de planta se enviaron por la URL de Arduino a una aplicación en la nube, la que presenta al usuario un registro gráfico de los valores de las variables que intervienen en los procesos.

Se desarrollo una red de comunicación con módulos Arduino para estructurar un sistema de comunicación que contenga todos los elementos de un lazo de control conectados con un medio físico único y compartido. El controlador recibe información de los sensores, procesa la misma y obtiene acciones de control que son enviadas a los actuadores, esto se aprecio a lo largo de las pruebas realizadas al observar como en los sistemas el error disminuye

RECOMENDACIONES

- Para aplicar NCS desarrollado en otras redes de datos se debe analizar cuál es la eficiencia de la red, observar cómo reacciona el sistema de control. En caso de presentarse conflictos se debe de establecer nuevamente el proceso de calibración de controladores, ya que los retardos de otras redes podrían tener otros valores.
- Al aplicar NCS a otros procesos de control, se debe analizar la velocidad de cambio de las variables que intervienen en el proceso, ya que la sensibilidad de un sistema depende de sus propiedades físicas.
- Para realizar un análisis de las diferentes variables de control como de red, se debe desarrollar una interfaz local que presenta gráficas en tiempo real para su análisis, también los diferentes valores de los algoritmos puedan ser modificados con facilidad sin necesidad de cambiar líneas de código del programa, incorporando una nueva área en la página principal para el usuario.
- Existen otras soluciones que pueden reemplazar la ausencia de datos en NCS tales como: funciones, pesos que varían por la última lectura, decisiones automáticas de sensores y actuadores bajo ciertas condiciones, incluso se podría incorporar lógica difusa. Estas soluciones serán consideradas en aplicaciones futuras.

REFERENCIAS

- Arduino, E. (20 de 12 de 2015). Obtenido de <https://www.arduino.cc/en/Main/ArduinoEthernetShield>.
- Arduino, m. (10 de diciembre de 2015). Obtenido de <http://arduino.cl/que-es-arduino/>
- Arduino, Y. (5 de 11 de 2015). Obtenido de <https://www.arduino.cc/en/Main/ArduinoBoardYun>.
- Casanova, V. (2005). *Sistemas de control basados en red. Modelado y diseño de estructuras de control*. Valencia: Universidad Politécnica de Valencia.
- Clariant. (2 de diciembre de 2015). *A NEW SERIES OF ORGANIC PIGMENTS*. Obtenido de <http://www.clariant.com/en/Innovation/Innovation-Spotlight-Videos/EDW-Pigments>
- Custodio, A. (1999). *Sensores Inteligentes: La revolución de la instrumentación*. Madrid: Departamento de electrónica UNEXPO.
- Dormido, S., Sánchez, J., & Kofman, E. (2008). *Muestreo, control y comunicación basada en eventos*. España: Comité Español de Automática(C.E.A.).
- Emoncms. (10 de octubre de 2015). *Emoncms*. Obtenido de <https://openenergymonitor.org/emon/>
- Lancovici, A. (2000). *Algunos problemas de la tintura de lana y sus nuevas técnicas*. International Wool Dyeing Service.
- Morilla, F. (2006). *Contraladores PID, ajuste Empírico*. Madrid: Dpto. de Informática y Automática ETSI de Informática, UNED.
- Pillajo, C., Hincapie, R., & Pilatasig, E. (2015). *Implementation of a network control system for a Robotic Manipulator as cloud service*. Chile: Chilecon.
- Sharp. (1999). *GP2Y0A41SK0F*.
- Texas, I. (2016). *LM35 Precision centigrade Temperature Sensors*. Texas Instruments.

ANEXOS

Anexo 1. Características de consumo eléctrico de los elementos de las estaciones.

Los diferentes elementos de la estaciones tienen varios valores de consumo que deben ser analizados para un posterior diseño.

Valores de voltaje y amperaje de diferentes dispositivos de las estaciones.

Numero	Elemento	Voltaje[V]	Amperaje[mA] (medidos en trabajo)	Total de amperaje [mA]
1	Arduino uno y ethernet	7 a 12	320	320
1	Arduino mega y ethernet	7 a 12	350	350
1	Arduino uno y Dragino Yun	7 a 12	350	350
3	Arduino mega y Dragino Yun	7 a 12	380	1140
1	Servomotor	4.8 a 6	350	350
3	Ventiladores	12	160	480
			Total	2990

Nota. Elaborado por: Paúl Bonilla

Anexo 2. Fuente Switching.

Las fuentes Switching toman la energía de entrada en periodos de tiempo definidos por un oscilador, para obtener en la salida diferentes valores de corriente y tensión. Presentan mayor eficiencia en comparación a fuentes lineales.

A continuación, se presenta las diferentes características de la fuente Switching usada en el proyecto.

Fuente Switching



Fuente de 12Vdc y 10A
Elaborado por: Paúl Bonilla.

Características de la fuente Switching S-120-12.

Modelo	S-120-12
Voltaje de entrada	110 - 220 Vac
Frecuencia de entrada	50/60 Hz
Salida	12 Vdc ,10 A
Dimensiones	20 x 10 x 4.3 cm
Protecciones	Cortocircuitos, sobrecarga, sobrevoltaje

Nota. Elaborado por: Paúl Bonilla

El amperaje de salida sobreaabastece los requerimientos de la carga, permitiendo adaptar otros elementos o cambios en las estaciones sin tener preocupación por la fuente de alimentación. Uno de los grandes problemas de las fuentes Switching es la presencia de ruido a la salida, normalmente se usan capacitores y reguladores lineales en serie para obtener una señal con mejores características de rizado. Arduino posee su propio regulador interno que podría usarse directamente conectando a la fuente Switching. Se debe de considerar que los reguladores lineales de voltaje tienden a calentarse al disipar el excedente de energía, la tarjeta Arduino se comporta de manera errática cuando

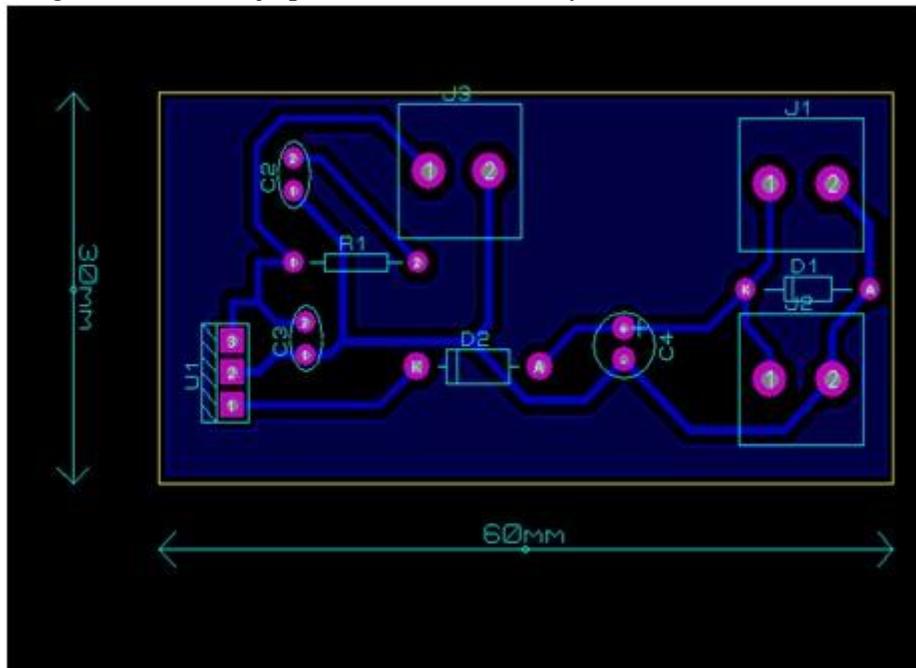
trabaja en rangos mayores a 40°C en por largos tiempos. No se puede acoplar un disipador de correctas dimensiones en el regulador interno de Arduino por el espacio reducido de la placa. Para solucionar este problema se usa un regulador externo con ventilación y un elemento que le ayude a disipar el calor fuera de la placa.

Anexo 3. Circuitos de Regulación de Voltaje

Arduino puede ser alimentado con un rango de voltaje de 7 a 12 Vdc. Se usa el regulador de voltaje 7808 que se halla dentro de los rangos que tolera Arduino. El regulador lineal 7808 tiene tres terminales que corresponden a: Entrada, Gnd y Salida.

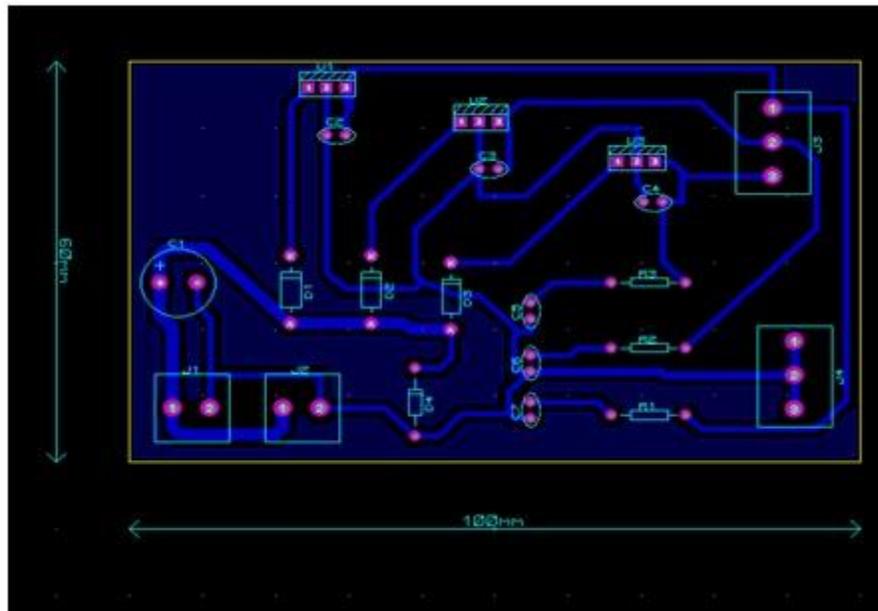
El terminal de entrada acepta valores de voltaje desde 10.5 a 23 Vdc, la fuente Switching tiene una salida de 12.7Vdc que garantice el correcto trabajo del regulador de voltaje. El regulador 7808 con encapsulado T0-220 tiene una corriente de salida de 1A, que abastece los diferentes elementos de los módulos ya que el mayor de los consumos no llega a 350mA. En la placa se incorporan diodos de protección en caso de polarizar erróneamente la placa, capacitores de entrada y salida para eliminar ruido y leds de visualización.

Circuito regulador de voltaje para Módulo On/Off y PWM



Circuito regulador de voltaje realizado en Proteus
Elaborado por: Paúl Bonilla.

Circuito regulador de voltaje para PID y Central de control

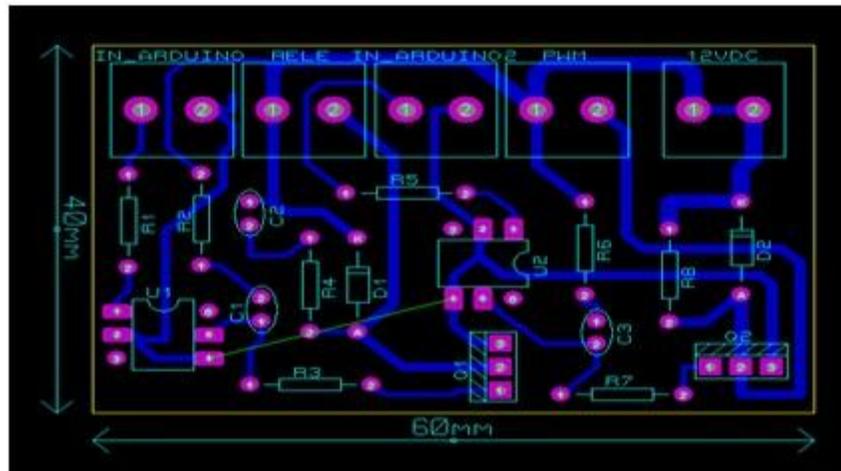


Circuito regulador realizado en Proteus
Elaborado por: Paúl Bonilla.

Anexo 4. Diseño de circuitos de salida para la estación PWM y Control On/Off.

Cada salida de Arduino está conectada a un octoacoplador, cuya salida es de tipo transistor. El colector está conectado al Gateway, activando o desactivando al Mosfet. El ventilador y relé se encuentran en el drenaje de su respectivo Mosfet.

Circuito de los actuadores: relé y ventilador



Circuito para actuadores realizado en Proteus
Elaborado por: Paúl Bonilla.

Anexo 5. Estación PID.

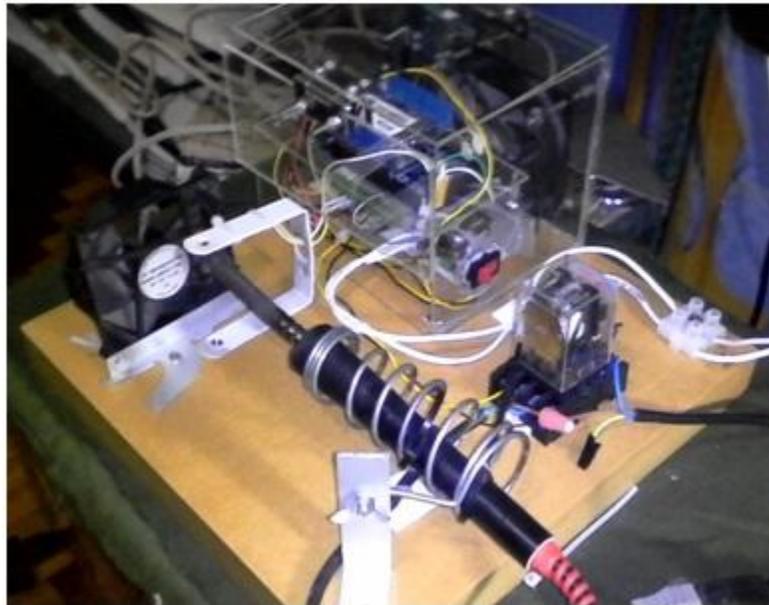
Estación PID



Módulo o estación de sensores y actuadores PID
Elaborado por: Paúl Bonilla.

Anexo 6. Estación PWM y Control On/Off.

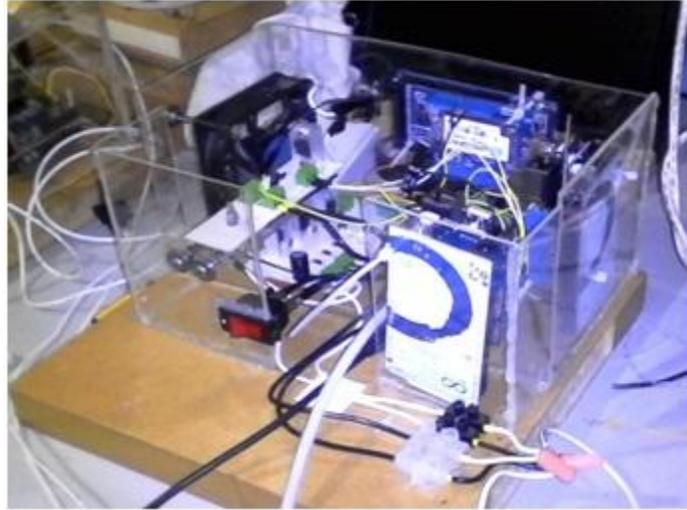
Estación PWM y control ON/OFF



Módulo o estación de sensores y actuadores para PWM y control On/Off
Elaborado por: Paúl Bonilla.

Anexo 7. Central de control.

Central de control.



Central de control de NCS
Elaborado por: Paúl Bonilla.