



**UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MÉXICO**

CENTRO UNIVERSITARIO UAEM TEXCOCO

**“AUTOMATIZACIÓN DE MEZCLA DE
SOLUCIONES PARA FERTIRRIGACIÓN”**

T E S I S

**QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN**

P R E S E N T A

REY JONATHAN HILDEBRANDO OSORIO GONZALEZ

DIRECTOR

DR. EN C. OZIEL LUGO ESPINOSA

REVISORES

M. EN C.C. MA. DOLORES ARÉVALO ZENTENO

LIC. EN INF. JESÚS ENRIQUE CONTRERAS GONZÁLEZ

TEXCOCO, ESTADO DE MÉXICO, ENERO DE 2016.

Contenido

Lista de Figuras.....	VIII
Lista de Tablas.....	X
Introducción.....	1
Definición del Problema	2
Metodología.....	3
Justificación.....	4
Objetivos	5
Objetivo General	5
Objetivos Específicos.....	5
Capítulo 1. Estado del Arte.....	6
Generalidades.....	6
Los Desarrollos que se Presentan son:	7
Sistema de Control Solar Automatizado para Fertirrigación Cucumis Melo L. Cultivo en Green House.....	8
Capítulo 2. Automatización.....	10
El Autómata Programable como Elemento de Control de Procesos Secuenciales	10
Controladores Lógicos Programables (PLC)	10
Arquitectura Interna	11
Ciclo de Funcionamiento	12
Tiempo de Ejecución y Control en Tiempo Real.....	16
Posibilidades de Configuración.....	22
Interfaces de Entrada/Salida.....	24
Interfaces Específicas.....	27
Elementos de Control de Procesos Industriales: Sensores y Actuadores.....	28
Detectores de Proximidad.....	30
Medidores de Posición o Distancia.....	33
Medidores de Pequeños Desplazamientos y Deformaciones.....	36
Transductores de Temperatura	37

Accionamientos: Clasificación	39
Capítulo 3. Plataforma Arduino.	41
¿Qué es Arduino?	41
¿Qué Quiere Decir que Arduino sea “Hardware Libre”?	41
¿Porque Elegir Arduino?	42
Elementos de Arduino	42
Hardware Elementos	42
Software	46
La Finalidad de Arduino	49
Ejemplos	50
Capítulo 4. Prototipo	53
Elementos Principales	53
Sensor de Flujo	53
Shield para 8 Relés	54
Electroválvula	54
Arduino Leonardo	55
Fuente de Poder ATX 300W	56
Conexiones	57
Ensamblado Final	63
Interfaz de Usuario	65
Fase de pruebas	66
Presupuesto de prototipo	66
Nombre	66
Discusiones	68
Conclusiones	69
Bibliografía	70
Anexo 1	72
Interfaces Hombre-Máquina	72
Usabilidad y Guías de Diseño	72
Técnicas de Interacción	76
Anexo 2	80

Código en Arduino	80
Anexo 3	87
Código en Java Envió de Valores	87

Lista de Figuras

ILUSTRACIÓN 1.ARQUITECTURA DESARROLLADA DE LA MEZCLA DE NUTRIENTES Y SISTEMA DE RIEGO (J.E. MOHD SALIH, 2012)	9
ILUSTRACIÓN 2.CONTROLADOR LÓGICO PROGRAMABLE (PLC).....	10
ILUSTRACIÓN 3.ESTRUCTURA INTERNA DE PLC'S (FACULTAD DE CIENCIAS, TECNOLOGÍAS VIRTUAL, ELECTRÓNICA II, 2015)	11
ILUSTRACIÓN 4.CICLO DE SCAN.....	13
ILUSTRACIÓN 5.CONFIGURACIÓN COMPACTA Y MODULAR DE LA UNIDAD DE CONTROL (JOSEP BALCELLS, 2000)	24
ILUSTRACIÓN 6.SEÑAL DIGITAL.....	26
ILUSTRACIÓN 7.SEÑAL ANALÓGICA	26
ILUSTRACIÓN 8.DIAGRAMA GENERAL DE UN SENSOR.....	29
ILUSTRACIÓN 9.ARDUINO ETHERNET SHIELD (ELECTRONICA ESTUDIO.CO, INGENIERÍA ELECTRÓNICA Y PROYECTOS PICMICRO, S.F.)	46
ILUSTRACIÓN 10.DETECTOR DE MENTIRAS (MONK, 2012)	50
ILUSTRACIÓN 11.OSCILOSCOPIO (MONK, 2012)	51
ILUSTRACIÓN 12.VENTILADOR CONTROLADO POR ORDENADOR (MONK, 2012).....	52
ILUSTRACIÓN 13.SENSOR DE FLUJO YF-S201	53
ILUSTRACIÓN 14.SHIELD PARA 8 RELÉS DE 10 AMP	54
ILUSTRACIÓN 15.ELECTROVÁLVULA	55
ILUSTRACIÓN 16.ARDUINO LEONARDO	55
ILUSTRACIÓN 17.FUENTE DE PODER ATX 300W	57
ILUSTRACIÓN 18.CONECTOR MOLEX MACHO SIN ENSAMBLAR	57
ILUSTRACIÓN 19.CONECTOR MOLEX DE LA FUENTE AL SHIELD DE RELÉS	58
ILUSTRACIÓN 20.CONECTOR MOLEX DE LA FUENTE A LAS TRES ELECTROVÁLVULAS	58
ILUSTRACIÓN 21.TERMINALES O ZAPATAS	59
ILUSTRACIÓN 22.CONEXIÓN DE LAS TERMINALES CON LA ELECTROVÁLVULA	59
ILUSTRACIÓN 23.CONEXIÓN DEL SENSOR DE FLUJO CON LA PLACA ARDUINO	60

ILUSTRACIÓN 24.CONEXIÓN CON HEADERS AL SENSOR DE FLUJO	61
ILUSTRACIÓN 25.IDENTIFICACIÓN DE LOS PINES QUE RECIÉN LOS DATOS DEL SENSOR DE FLUJO	61
ILUSTRACIÓN 26.IDENTIFICACIÓN DE LOS PINES TIERRA GND Y VOLTAJE 5V	62
ILUSTRACIÓN 27.CABLEADO INTERNO DEL PROTOTIPO	62
ILUSTRACIÓN 28.VISTA PARTE FRONTAL DEL PROTOTIPO	63
ILUSTRACIÓN 29.VISTA TRASERA DEL PROTOTIPO	64
ILUSTRACIÓN 30.INTERFAZ DE USUARIO DEL PROTOTIPO.....	65
ILUSTRACIÓN 31.USABILIDAD	72
ILUSTRACIÓN 32.INTERFAZ DE LÍNEA DE COMANDOS	77
ILUSTRACIÓN 33.INTERFAZ GRÁFICA DE USUARIO (GUI)	78
ILUSTRACIÓN 34.INTERFAZ DE USUARIO NATURAL (ONYSUS CREATE WONDER , S.F.).....	79

Lista de Tablas

TABLA 1.RETARDOS DE CONMUTACIÓN TÍPICOS DE INTERFACES DE AUTÓMATAS (MS.).....	18
TABLA 2.CARACTERÍSTICAS GENERALES DE VERSIONES DE ARDUINO	44
TABLA 3.TIPOS DE DATOS EN ARDUINO (ARTERO, 2013).....	49
TABLA 4.CARACTERÍSTICAS SENSOR DE FLUJO YF-S201	53
TABLA 5.CARACTERÍSTICAS SHIELD PARA 8 RELÉS DE 10 AMP	54
TABLA 6.CARACTERÍSTICAS ELECTROVÁLVULA	55
TABLA 7.CARACTERÍSTICAS ARDUINO LEONARDO.....	56
TABLA 8.ELEMENTOS PRINCIPALES Y COSTOS.....	66
TABLA 9.OTROS ELEMENTOS Y PRECIOS.....	67
TABLA 10.COSTO DE DESARROLLO	67

Introducción

La Fertirrigación es una técnica de aplicación de abonos disueltos en el agua de riego a los cultivos. El objetivo principal de la Fertirrigación es el aprovechamiento del flujo de agua del sistema de riego para transportar los elementos nutritivos que necesita la planta hasta el lugar donde se desarrollan las raíces, con lo cual se optimiza el uso del agua, los nutrientes y la energía, y se reducen las contaminaciones si se maneja de manera adecuada. Calcular las necesidades de fertilización de cualquier cultivo no es una tarea sencilla ya que cada variedad de planta tiene necesidades particulares y la selección de la solución para Fertirrigación depende de diversos factores como propiedades del suelo, requerimientos del cultivo, disponibilidad y costo de los fertilizantes, compatibilidad entre fertilizantes, entre otros.

La Ingeniería en Computación estudia el desarrollo de sistemas automatizados y el uso de los lenguajes de programación, de igual forma se enfoca en el análisis, diseño y la utilización de hardware y software para lograr la implementación de las nuevas tecnologías y así lograr su implementación y aplicación.

En base a lo anterior, se pretende automatizar el proceso para la realización de soluciones para la Fertirrigación donde el objetivo principal es diseñar un prototipo y un programa de computadora con una interfaz de usuario amigable para los productores o personas que necesitan preparar soluciones de Fertirrigación. El software interactuará con la plataforma Arduino como elemento electrónico para controlar actuadores y leer datos de sensores. Los actuadores serán electroválvulas solenoides para permitir el paso de sustancias hacia un contenedor para la preparación de la solución y se usarán sensores de flujo para determinar la cantidad necesaria de cada sustancia en cuestión. El software libre (Java) y hardware libre (Arduino) con que se construirá el prototipo, permitirá una disminución en costos.

Definición del Problema

Se sabe que el hombre comenzó a cultivar las tierras desde hace miles de años, pero la historia de la fertilización se inicia cuando los agricultores primitivos descubrieron que determinados suelos dejaban producir rendimientos aceptables si se cultivaba continuamente, y que al añadir estiércol o residuos vegetales se restauraba su fertilidad. El origen de la industria mundial de fertilizantes se inició a mediados del siglo XIX.

El incremento de la población mundial en los últimos años exige un constante reto a la agricultura para proporcionar un mayor número de alimentos, tanto en calidad como en cantidad.

Para alcanzar los retos de poder incrementar la producción agrícola para abastecer el crecimiento de la población, únicamente existen dos factores posibles.

- Aumentar la superficie de cultivo, posibilidad cada vez más limitada, lo que provocaría el deterioro de las grandes masas forestales.
- Proporcionar a los suelos fuentes de nutrientes adicionales en formas asimilables por las plantas, para incrementar el rendimiento de los cultivos.

El fertirriego se puede definir como el proceso por el cual los fertilizantes son aplicados con el agua de riego, en éste contexto México requiere de apoyo para mejorar la productividad y competitividad, ya que la mayoría de la tecnología que se utiliza proviene principalmente del extranjero, cuyo costo es alto. Según datos de la Secretaría de la Agricultura, Ganadería, Desarrollo Rural, Pesca y Alimentación en los últimos años se ha incrementado el número de invernaderos, por lo que es importante desarrollar técnicas, conocimientos y fundamentos para lograr los objetivos. El papel de Ingeniería en Computación en éste campo es el diseño de sistemas que impliquen software y hardware para mejorar la calidad del producto, en las diferentes etapas de crecimiento del cultivo.

Metodología

Para el trabajo de investigación se usará el método de análisis-síntesis, el cual consiste en la separación de las partes de un todo para estudiarlas en forma individual (análisis), y la reunión racional de los elementos dispersos para estudiarlos en su totalidad (síntesis).

Para aplicar el método de análisis-síntesis, se requiere:

- Observación de la preparación de soluciones para el fertirriego, partes y componentes.
- Descripción. Identificación de todos sus elementos, partes y componentes para poder entenderlo.
- Examen crítico. Es la revisión rigurosa de cada uno de los elementos de Fertirrigación con Tecnología.
- Descomposición. Análisis exhaustivo de todos los detalles de la automatización de Fertirriego, comportamiento y características de cada uno de los elementos constituidos.
- Enumeración Desintegración de los componentes a fin de identificarlos, registrarlos y establecer sus relaciones con los demás.
- Ordenación. Volver a armar y reacomodar cada una de las partes del todo descompuesto con el fin de restituir su estado original, con ello tener un prototipo propio.
- Clasificación. Ordenación de cada una de las partes por clases, siguiendo el patrón del fenómeno analizado, para conocer sus características, detalles y comportamiento.
- Conclusión. Analizar los resultados obtenidos, estudiarlos y dar una explicación del fenómeno observado.

Justificación

El costo de la tecnología que proviene de otros países es muy alto y genera dependencias de capacitación y compra de refacciones en el extranjero, esto puede llegar a ser muy costoso con respecto al tiempo en un proceso de trabajo continuo.

La necesidad de obtener actualmente nuevas fuentes de energía abre un nuevo campo para la agricultura, y la aplicación adecuada de fertilizantes debe contribuir a conseguir este objetivo.

Gracias a los fertilizantes se es posible alcanzar los siguientes retos.

- Asegurar la productividad y calidad nutricional de los cultivos, ofreciendo una seguridad alimenticia e incrementando el contenido de nutrientes de las cosechas.
- Evitar la necesidad de incrementar la superficie agrícola, ya que sin los fertilizantes habría que destinar millones de hectáreas adicionales a la agricultura.
- Conservar el suelo y evitar su degradación, y mejorar la calidad de vida del agricultor y su entorno.
- Contribuir a la mayor producción de materia prima para la obtención de energías alternativas.

Se beneficia a campos de cultivos, implementando tecnología e innovación, implementando un prototipo con el fin de optimizar de manera eficiente las cantidades exactas para la creación de soluciones.

Con este método se pueden lograr incrementos de productividad en comparación con la forma tradicionales de riego. También permite un ahorro en el uso de fertilizante y agua así como mejorar la calidad del producto ya que se controlan mejor las condiciones de humedad, se protege a los cultivos de deshidratación o exceso de agua, aplicando la solución directamente en las raíces de la planta.

Objetivos

Objetivo General

Desarrollar un prototipo para la optimización en la preparación de soluciones para la fertirrigación a través de software y hardware de plataforma libre.

Objetivos Específicos

- Controlar componentes de hardware para la mezcla de soluciones y así proporcionar la cantidad exacta de cada componente a mezclar, para sus distintas pruebas a realizar.
- Desarrollar un software en un lenguaje de programación (JAVA), para la generación de una Interfaz amigable para el usuario, y así realizar la preparación de soluciones.
- Integrar la plataforma de hardware libre (Arduino), para el control de elementos electrónicos, el cual permitirá el control de los destinos actuadores.
- Realizar la conexión Java con Arduino para obtener los datos que arrojan los sensores de flujo.

Capítulo 1. Estado del Arte.

Generalidades

En la actual producción de alimentos, la creciente industrialización de la agricultura se proyecta sobre estos bienes de capital materializándose en innovaciones tecnológicas continuas que se traducen en la productividad.

En la actualidad se ha implementado en el sector agrícola tecnologías y mecanismos de precisión sumándose el uso de la electrónica como innovación, lo que facilita la recopilación de datos en tiempo real, el uso exacto de nutrientes o fertilizantes en el riego y aplicaciones diversas, permitiendo así la posible corrección de las acciones realizadas al instante. Esto con el fin de obtener una mayor precisión en la agricultura.

Las investigaciones y futuros desarrollos apuntarán a incrementar la eficiencia productiva y generar un impacto positivo en el desarrollo de la agricultura. La innovación es entendida como la capacidad de generar e incorporar conocimientos, permitiendo no sólo maximizar las potenciales productivas, sino que también puede contribuir a una utilización más racional de los recursos.

La tecnología desarrollada con el objetivo de lograr una alta eficiencia en la explotación de los sistemas de riego localizados, con un uso más racional y eficiente del agua y la energía.

Su propósito es el desarrollo y/o adaptación de tecnologías necesarias para incrementar la eficiencia de los productos, empleando diversos sistemas electrónicos e informáticos así como el uso de las distintas tecnologías, transmisión de datos, sistema de control entre otros.

En el Instituto de Investigaciones de Riego y Drenaje (Reinaldo Rey García, 2007), tecnología cubana se han desarrollado Tecnologías de Automatización para los sistemas de riego.

Los Desarrollos que se Presentan son:

- a. Bajante con boquilla de baja presión para las máquinas de pivote central.
- b. Tecnología para los sistemas de riego localizado.
- c. Tecnología de riego por gravedad de flujo intermitente.
- d. Tecnología de riego por aspersión de baja intensidad.

Bajante con Boquilla de Baja Presión para las Máquinas de Riego de Pivote Central y Modernización de la Actual

Con el desarrollo de la tecnología de las máquinas de pivote central para el riego, la tendencia en los últimos años, en cuanto al manejo de estos equipos es reducir la presión de trabajo, sustituyendo los aspersores de impacto por difusores de baja presión y disminuir la altura de colocación de estos emisores respecto al suelo, permitiendo el incremento en los rendimientos de producción, aumentando la eficiencia del agua hasta un 85%, así como, el ahorro de energéticos y materias primas en la actividad de riego.

Tecnologías de Riego Localizado

Tecnología cubana obtenida a partir de la transferencia de las tecnologías avanzadas empleadas en los países de que están a la vanguardia a nivel mundial.

Esta tecnología se desarrolló con el objetivo de lograr una alta eficiencia en la explotación de los sistemas de riego localizado, con un uso más racional y eficiente del agua y la energía

Consta de los siguientes elementos:

- Programadores de riego.

- Una familia de válvulas hidráulicas básicas.
- Un cuadro de mando hidráulico.
- Elementos de trasmisión de las señales de accionamientos.
- Accesorios varios.

Tecnologías de Riego por Gravedad de Flujo Intermitente

A partir de estudios de investigación de campo, ha propuesto la tecnología riego por gravedad de flujo intermitente, debido a su alta eficiencia en la aplicación y distribución del agua de riego

Tecnología de Riego por Aspersión de Baja Intensidad

La utilización de esta tecnología en sustitución de los sistemas de riego de alta precisión, contribuye al ahorro de agua y a la preservación de los suelos, así como la sustitución de importaciones. Con ello se garantiza la reposición de los elementos que constituyen el sistema de riego, constituyendo a su sostenibilidad.

Sistema de Control Solar Automatizado para Fertirrigación Cucumis Melo L. Cultivo en Green House.

La producción de frutas y vegetales en Malasia utilizan métodos de fertirrigación de crecimiento acelerado. Ofreciendo a los agricultores de forma automática, cantidades adecuadas de nutrientes y concentraciones mediante el riego por goteo para las plantas aplicándola en raíz a lo largo del ciclo de cultivo. Utilizando tres temporizadores digitales predeterminados que sirven para activar o desactivar las bombas de los inyectores de riego para la mezcla de fertilizantes.

Diseño Mecánico

Se creó un prototipo para ello, el sistema fue desarrollado utilizando tres tanques cilíndricos de polietileno. Colocando los fertilizantes en los dos tanques de 30 litros y un tercer tanque de 60 litros que fue utilizado para la mezcla de la solución. Los inyectores de fertilizante (bomba A y la bomba B), son bombas de agua con un

caudal de 55 litros / hora de 6v y la irrigación de la bomba tenía una velocidad de flujo de 250 litros / hora de 12v. Tres unidades de válvulas solenoide de 12v utilizados para controlar el flujo de agua al tanque de mezcla y para controlar el flujo de nutrientes al sistema de tuberías de riego. Un sensor ultrasónico utilizado como sensor de nivel que permite monitorear el nivel en el tanque de mezclas.

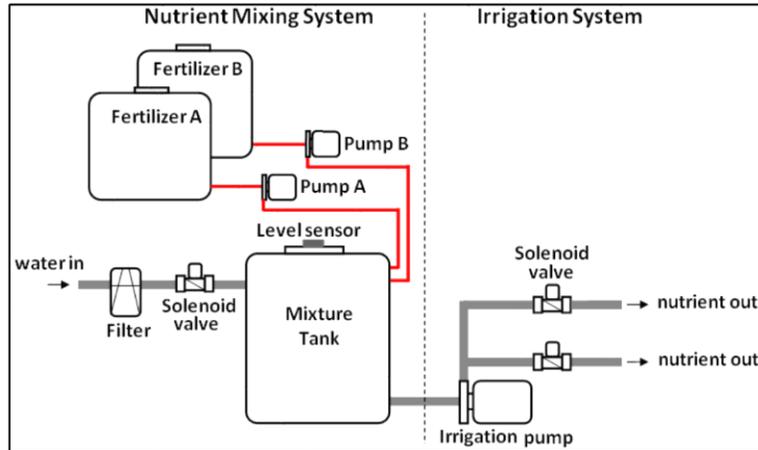


ILUSTRACIÓN 1. ARQUITECTURA DESARROLLADA DE LA MEZCLA DE NUTRIENTES Y SISTEMA DE RIEGO (J.E. MOHD SALIH, 2012)

Algoritmo de Control

Fue programado para ejecutar el proceso de mezclas de nutrientes y riego basado en el nivel establecido por el usuario utilizando un medidor de potencial en función del periodo de siembra. El nivel seleccionado por el usuario determinara el periodo para ambas bombas inyectoras. El volumen total dentro de los tanque es medido por el sensor de nivel. Una subrutina convierte la distancia en centímetros que se recibirá el sensor que será el volumen de nutrientes en el interior del tanque. Los parámetros del proceso, periodos de siembra en número de semanas, días y horas se mostraran en pantalla. El usuario también puede controlar cualquier otra condición de parámetros de proceso tales como nivel de solución de nutrientes, el volumen de nutrientes.

Al ser un sistema de fertilizantes por goteo completamente operado con energía solar se puede instalar en lugares rurales o remotos.

Capítulo 2. Automatización.

El Autómata Programable como Elemento de Control de Procesos Secuenciales

Controladores Lógicos Programables (PLC)

Un PLC (Programmable Logic Controller) o autómata programable, es una máquina electrónica programable capaz de ejecutar un programa, o sea, un conjunto de instrucciones organizadas de una forma adecuada para solventar un problema dado, y diseñada para trabajar en un entorno industrial y por tanto hostil. Las instrucciones disponibles para crear programas serán de una naturaleza tal que permitirán controlar procesos, por ejemplo: funciones lógicas, operaciones aritméticas, de conteo de eventos, de temporización, etc. Además, el PLC estará diseñado de forma tal que la conexión del mismo con el proceso a controlar será rápida y sencilla por medio de entradas y salidas de tipo digital o analógico. (Universidad de Oviedo Área de Ingeniería de Sistemas y Automática, 2008).

Un PLC es un controlador basado en un microprocesador con múltiples entradas y salidas, utiliza una memoria programable para almacenar las instrucciones y llevar a cabo las funciones de control y proceso (Collins, Mechatronickai És Autotechnikai Intézet, 2015).

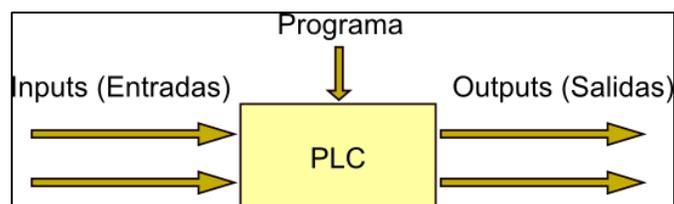


ILUSTRACIÓN 2. CONTROLADOR LÓGICO PROGRAMABLE (PLC)

Ventajas de un PLC (Facultad de Ciencias, Tecnologías Virtual, Electrónica II, 2015):

- Menor tiempo empleado en la elaboración de proyectos.
- Posibilidad de hacer modificaciones sin alterar el cableado y/o añadir elementos.
- Posibilidad de controlar varias máquinas con el mismo Autómata.
- Reducido espacio de ocupación.
- Menor costo.
- Confiable.
- Reprogramable.

Arquitectura Interna

Para poder interpretar el funcionamiento de un PLC (Facultad de Ciencias, Tecnologías Virtual, Electrónica II, 2015), se muestra un esquema de su estructura interna. Donde podemos distinguir cinco bloques en la estructura de los Autómatas Programables.

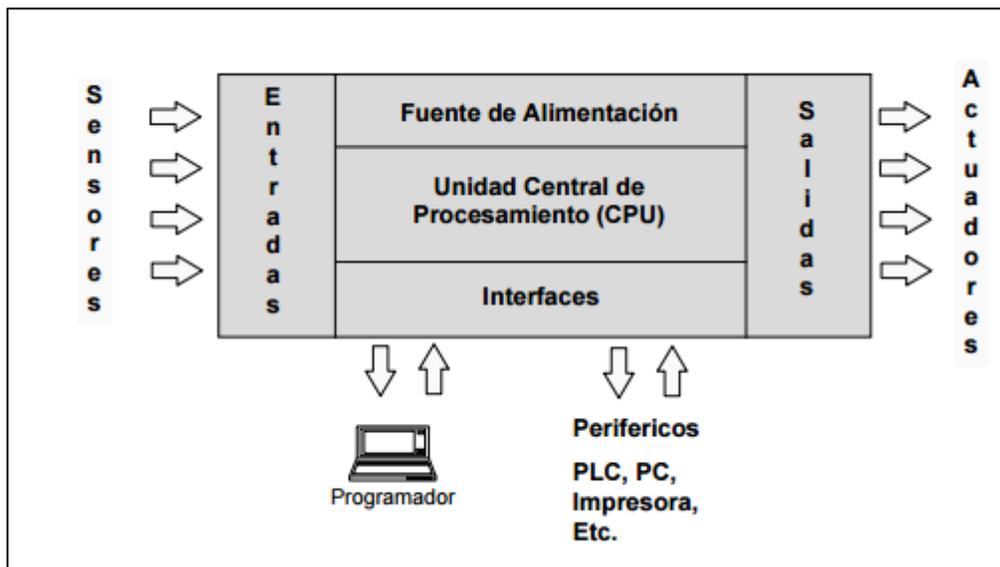


ILUSTRACIÓN 3. ESTRUCTURA INTERNA DE PLC'S (FACULTAD DE CIENCIAS, TECNOLOGÍAS VIRTUAL, ELECTRÓNICA II, 2015)

- **Bloque de Entrada.** En él se reciben las señales que proceden de los sensores. Estas son adaptadas y codificadas de tal forma que sean comprendidas por la CPU, teniendo como misión proteger los circuitos electrónicos internos del PLC, realizando una separación eléctrica entre estos y los sensores.
- **Bloque de Salida.** Trabaja de forma inversa al anterior. Interpreta las órdenes de la CPU, las decodifica y las amplifica para enviarlas a los actuadores. También tiene una interface para asilar la salida de los circuitos internos.
- **Unidad Central de Procesamiento (CPU).** En ella reside la inteligencia del sistema. En función de las instrucciones del usuario (programa) y los valores de las entradas, activando las salidas.
- **Fuente de Alimentación.** Su misión es adaptar la tensión de red (220V/50Hz) a los valores necesarios para los dispositivos electrónicos internos, generalmente (24Vcc y 5Vcc)
- **Interfaces.** Son los canales de comunicación con el exterior. Por ejemplo con:
 - equipos de programación
 - otros autómatas
 - computadoras

Ciclo de Funcionamiento

Un PLC una vez conectado a la red eléctrica tiene básicamente dos modos de funcionamiento (Universidad de Oviedo Área de Ingeniería de Sistemas y Automática, 2008).

- **Stop.** En este modo de funcionamiento no se ejecuta el programa de control.
- **Run.** En este modo de funcionamiento el programa de control se está ejecutando de manera indefinida hasta que el PLC pasa al modo Stop o bien se desconecta de la alimentación.

Siendo este último modo de funcionamiento el más interesante. Cuando el autómatas se encuentra en esta situación, el programa de control que está grabado en su memoria se ejecuta **cíclicamente** describiendo lo que se ha dado en llamar “**Ciclo de Scan**”.

Un ciclo de scan consiste básicamente en cuatro pasos bien diferenciados:

1. Lectura de las entradas del PLC.
2. Ejecución de programa de control.
3. Escritura de las salidas del PLC.
4. Tareas internas del PLC.

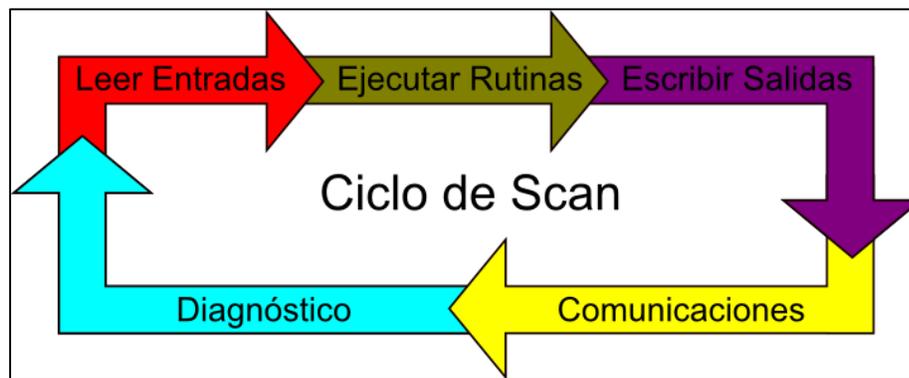


ILUSTRACIÓN 4. CICLO DE SCAN

Lectura de Entradas

Al comienzo de cada ciclo de scan, el sistema operativo del PLC comprueba el estado en que se encuentra todos y cada uno de los elementos de entrada (sensores, pulsadores, etc.) que están conectados en los distintos módulos de entrada del PLC.

Si un sensor está activado el PLC pondrá un “1” lógico en una posición determinada de una zona de memoria especial llamada “Memoria de Entrada” o “Imagen de Proceso de Entradas”. Si por el contrario ese sensor no estuviera activado, entonces el PLC pondrá un “0” lógico en la posición de memoria de entrada asignada para esa entrada. Si el sensor fuese analógico una vez de escribir un “1” o un “0”, se

convertiría en el valor de la magnitud física a un valor numérico que también se depositaría en una zona de la memoria de entradas analógicas.

Esta operación de lectura de las entradas conlleva un cierto tiempo para ejecutarse totalmente, el cual debe ser tenido en cuenta a la hora de calcular la duración del ciclo scan. En cualquier caso, este tiempo suele ser despreciable con respecto a la duración de la ejecución del programa de control.

Al hacer esta operación de lectura de todas las entradas al comienzo de cada ciclo se asegura que todas las instrucciones del programa de control son ejecutadas sobre una “imagen” del estado del proceso coherente. Es decir, que si una vez comenzado a ejecutar el programa de control, algún sensor del proceso cambia de valor, este nuevo valor no se tomará en cuenta hasta el siguiente ciclo.

Ejecución de Programa de Control

Una vez que la memoria de entrada ha sido totalmente actualizada el sistema operativo del PLC, comienza a ejecutar las instrucciones del programa albergado en su memoria. Lo hará secuencialmente comenzando por la primera instrucción del módulo del programa que se considere el principal.

La ejecución secuencial no implica ejecución lineal, es decir, que un programa puede contener instrucciones especiales que permitan hacer saltos hacia adelante y hacia atrás, e incluso es posible que haya subrutinas e interrupciones. Pero en cualquier caso, la ejecución seguirá siendo secuencial siendo posible alterar esa secuencia de manera dinámica. Esa secuencia acabará teniendo una última instrucción que tras ser ejecutada pondrá fin a este paso del ciclo de scan.

Dado que la secuencia de ejecución de un programa puede ser alterado por medio de instrucciones de programa será posible hacer un programa de control que no termine nunca, lo que en informática se conoce como un “bucle infinito” que haga que el PLC se “cuelgue”. El autómatas dispone de mecanismos para detectar esta situación y provocar la finalización de la ejecución del programa, y será

responsabilidad del programador contemplar esta anomalía pasa subsanar sus posibles efectos. Lo deseable es no generar nunca un bucle infinito, teniendo en cuenta que un PLC es un dispositivo dedicado a controlar un proceso y en caso de presentarse una situación de esa naturaleza, el proceso se descontrolaría poniendo en peligro la integridad de las personas y de la propia instalación.

La normal ejecución del programa de control hará que los valores de la memoria de datos del PLC vayan cambiando a medida que avanza, Especial importancia tienen los cambios que se registren en la memoria de salida ya que serán los que finalmente se traduzcan en acciones sobre el proceso. Por último es de destacar que la ejecución del programa de control durara una determinada cantidad de tiempo que será directamente proporcional a la velocidad del procesador (CPU) del PLC y que por lo tanto dependerá del modelo del mismo.

Escrituras de Salidas

Cuando el sistema operativo del PLC detecta que se ha ejecutado la última instrucción del programa de control, éste comienza a revisar una por una todas las posiciones de su memoria de salidas. Si esta posición lee un “1” lógico, el PLC activará la salida correspondiente en el módulo de salidas. Es de esperarse que conectado a ese módulo de salida en esa posición concreta se encuentre un preaccionador o un accionador (por ejemplo una electroválvula) el cual se activará al ser activada esta salida llevando a cabo la acción correspondiente (por ejemplo cortar paso de agua de una vivienda) sobre algún elemento del proceso.

Si el programa de control tras su ejecución genera señales analógicas en forma de valores digitalizados en la memoria de salida analógicas del PLC, en esta fase son convertidas en valores determinados de corriente y tensión por medio de los módulos de salida analógicas correspondientes. Estos valores de corriente y tensión provocarán una acción proporcional sobre algún componente del proceso.

Aunque este proceso consume cierta cantidad de tiempo, su duración es despreciable con respecto a la de la fase de ejecución del programa de control

Tareas Internas

Antes de comenzar un nuevo ciclo de scan, el PLC necesita realizar ciertas tareas internas como por ejemplo comprobar si se han producido de errores, almacenar la duración del ciclo scan, actualizar los valores internos de sus tablas de datos, etc.

De nuevo la duración de esta fase puede considerarse despreciable con respecto a las otras tres.

Una vez que esta fase ha terminado el sistema operativo del PLC comenzará a ejecutar un nuevo ciclo scan.

Tiempo de Ejecución y Control en Tiempo Real

El tiempo total que el autómatas emplea para realizar un ciclo de operación se llama “tiempo de ejecución de ciclo” o sencillamente “tiempo de ciclo” (Scan Time) (Josep Balcells, 2000). Dicho tiempo depende de:

- El número de entradas/salidas involucradas.
- La longitud del programa de usuario.
- El número y tipo de periféricos conectados al autómatas.

El tiempo total de ciclo es entonces la suma de los tiempos empleados en realizar las distintas operaciones del ciclo, a saber:

- Autodiagnóstico.
- Actualización de E/S.
- Ejecución del programa.
- Servicio a periféricos.

Los tiempos empleados en chequeos y comprobaciones cíclicas (autodiagnósticos) son del orden 1 a 2 μ s, al igual que los consumidos en la atención a periféricos externos.

Los tiempos de ejecución de instrucciones se miden en unidades de microsegundos (μs), resultando un tiempo de ejecución del programa variable en función del número y tipo de instrucciones contenidas. Precisamente, éste tiempo es uno de los parámetros que caracterizan a un autómata, expresado normalmente en milisegundos por cada mil instrucciones ($\mu\text{s} / \text{K}$).

Lógicamente, el tiempo resultante dependerá del tipo de instrucciones elegidas, por lo que el fabricante suele indicar las utilizadas como base de cálculo (booleanas, aritméticas, de temporización, entre otros), y las proporciones entre ellas dentro del programa muestra. Son tiempos típicos de 5 a 15 $\mu\text{s} / \text{K}$ instrucciones, aunque la aplicación de nuevas tecnologías y diseños a la CPU están reduciendo los tiempos a valores menores a 1 $\mu\text{s} / \text{K}$ instrucciones.

Los tiempos de acceso al exterior para lectura/escritura de datos son del orden 1 a 5 μs , dependiendo del número de E/S involucradas, de la CPU del autómata, de si este es compacto o modular, si se utilizan o no entradas/salidas remotas etc.

El autómata estándar de gama media, para una aplicación que necesite aproximadamente unas 1000 instrucciones, el tiempo de ciclo puede estimarse de orden 20 milisegundos.

El tiempo de ciclo será uno de los factores determinantes en el diseño del sistema de control cuando en éste aparezcan muy bajas constantes de tiempo (sistema de respuesta rápida), pero no el único a considerar, ya que también la electrónica de las interfaces introduce retardos adicionales.

La lectura/escritura de señales, realizada a través de las interfaces de entrada/salida, está afectada por los retardos propios de estos circuitos que, al incluir filtros que tratan de evitar la entrada de ruidos eléctrico en el autómata, introducen también retardos en las comunicaciones de la señal, limitando la frecuencia máxima de esta a valores inferiores a 100 Hz.

La Tabla 1 representa los rangos de retardo máximo típicos, en milisegundos, para distintas interfaces de entrada y salida en las comunicaciones de encendido a apagado y viceversa

INTERFAZ CONMUTACION	ENTRADA ($T_{INPUT\ delay}$)			SALIDA ($T_{OUTPUT\ delay}$)	
	AC	DC	RELE	TRANSISTOR	TRIAC
	OFF-ON	15-30	5-10	5-15	0.5-2
ON- OFF	15-30	5-10	5-20	1-2	11-12

TABLA 1. RETARDOS DE CONMUTACIÓN TÍPICOS DE INTERFACES DE AUTÓMATAS (MS.)

Como se observa, las entradas en DC son más rápidas que en AC, por la mayor necesidad de filtrado de las tensiones de red, mientras las salidas estáticas son también más rápidas que las salidas por relé.

Si el proceso utiliza además señales analógicas, será preciso considerar también los tiempos de conversión A/D y D/A, lo que, dado el procedimiento de conversión por doble rampa normalmente utilizado, obliga a considerar retardos aún mayores, de 15 a 20 μ s.

Dada una señal de mando (salida) función de una o varias señales de planta (entradas), se denomina tiempo de respuesta. El tiempo dependerá, por tanto, de los retardos de comunicación y adaptación de señal en las interfaces de entrada y salida $T_{INPUT\ delay}$ y $T_{OUTPUT\ delay}$ y del tiempo de ciclo del autómata.

Aunque el tiempo de respuesta es variable dependiendo del instante en que cambia la entrada respecto al ciclo de operación, se mueve siempre entre dos valores límite máximo y mínimo.

Estos valores límites son:

- Valor máximo (la señal está disponible en la interfaz justo después de la lectura de entradas).

$$T_{RES\ max} = T_{INPUT\ delay} + 2 T_{CICLO} + T_{OUTPUT\ delay}$$

- Valor mínimo (la señal está disponible en la interfaz justo antes de la lectura de entradas).

$$T_{RES\ min} = T_{INPUT\ delay} + T_{CICLO} + T_{OUTPUT\ delay}$$

El tiempo de retardo en las interfaces de entradas y salidas depende del tiempo de comunicación. ON/OFF o bien OFF/ON, así que para el cálculo del tiempo de respuesta máximo habrá que tomar los mayores posibles que indique el fabricante.

Los valores de tiempo de respuesta resultante, y de su inversa, la frecuencia de operación, pueden resultar depreciables frente a las constantes de tiempo de los sistemas electromecánicos (relés, solenoides, electroválvulas, etc.) a los que el autómatas sustituyó o controlaba en su primer aplicación de mercado, pero no lo son tanto comparados con los tiempos extremadamente cortos que intervienen en la lectura y control de elementos electrónicos, típicos en procesos de posicionamiento de motores (lectura de pulsos de encoders), respuesta a alarma y detección de móviles a alta velocidad.

Si se trata de controlar uno de estos procesos relativamente rápidos, puede darse el caso de que la respuesta del autómatas resulte demasiado lenta, con lo que el control no actuará como se espera.

Se dice que el autómatas es capaz de controlar en tiempo real a un proceso, si sus tiempos de respuesta o retardos resultan muy pequeños frente a los tiempos de reacción del mismo.

Elementos de Proceso Rápido

Hay tres tipos de actividades que exigen rápidas respuestas del autómatas:

- Ejecución de subrutinas o programas a intervalos menores de los que permite el tiempo de ciclo general del autómata (por ejemplo, para el procesado de alarmas).
- Lectura de impulsos de entrada a alta frecuencia (por ejemplo, procedentes de encoders).
- Detección de señales de muy corta duración, menor que el tiempo de ciclo (por ejemplo, para detección de móviles a alta velocidad).

Cada una de las circunstancias anteriores plantea problemas distintos, que obligan también a buscar distintas soluciones en el autómata.

Para el primer supuesto, reducción del tiempo de respuesta para una señal determinada, puede ordenarse la ejecución del bloque de programa que la controla de forma independiente al resto del ciclo, iniciándose ésta:

- por medio de un señal periódica (para reducir el tiempo de ciclo).
- por medio de una señal de interrupción exterior (para obtener respuestas rápidas).

En el primer caso, activación de una rutina rápida mediante una señal periódica (normalmente, una temporización interna), se asegura la reducción del tiempo de ciclo para las señales E/S manipuladas dentro de la rutina, mientras que en el segundo caso, activación mediante interrupción exterior, se permite al autómata responder con rapidez frente a estímulos externos.

En el segundo supuesto, lectura de impulsos de alta frecuencia, la pérdida de información que supone la aparición de varios impulsos dentro de un mismo ciclo de ejecución, al ser leído sólo el que coincida con el instante de lectura, puede evitarse disponiendo de un contador rápido que, aprovechando un hardware específico con entradas débilmente filtradas y circuito propio, sea capaz de leer señales de alta frecuencia.

Por último, para la detección de señales de muy corta duración podría emplearse cualquiera de las dos soluciones anteriores, aunque algunos autómatas que no las incluyen, sobre todo en gamas bajas, pueden tener en cambio una o varias entradas detectoras de flancos, capaces de captar y mantener en un relé interno, y hasta su lectura por parte de la CPU, señales individuales de tan corta duración como 1 μ s (menores de un Scan).

Procesado Rápido de Programas

Con el procesamiento rápido de programa se elimina la dependencia de algunas señales con el tiempo de ciclo del programa base, mejorando la velocidad de respuesta de estas señales. A fin de reducir al máximo el tiempo de respuesta, es frecuente que los autómatas con capacidad de procesamiento rápido tengan definidas algunas entradas y salidas con tiempo de conmutación más pequeños que las ordinarias (menores filtrados).

La combinación de procesos rápidos de instrucciones con el uso de estas entradas/salidas no filtradas o con filtrados reducidos, disminuye la respuesta entre la aparición de las señales y la ejecución de la señal de mando a valores.

Contador de Alta Velocidad

El contador rápido es un módulo de hardware independiente de la CPU, capaz de controlar impulsos exteriores de alta frecuencia procedentes de captadores como decodificadores incrementales o detectores de proximidad. El contador cuenta los impulsos de forma asíncrona con el resto del autómata, utilizando sus propias entradas débiles filtradas y específicas para esta aplicación.

Entradas detectoras de flanco

En ocasiones, el autómata debe ser capaz de leer señales de entrada de muy corta duración, mucho menores que su tiempo de ciclo.

Si se dispone de contador rápido, pueden leerse estas señales cortas preseleccionando el valor de cuenta a la unidad: cuando se reciba la señal, el contador alcanzará el final de cuenta y podrá leerse la conmutación de su salida desde el programa de ejecución normal, por consulta o interrupción.

Si el autómata puede ejecutar rutinas rápidas controladas por entradas de interrupción, puede utilizarse la señal a leer como disparo de un programa rápido, el cual conmutará cualquier relé interno para que el programa base pueda leer sobre él la existencia o no de la señal.

Los autómatas compactos de pequeño tamaño y aplicaciones restringidas no poseen el tratamiento rápido de señales, pero pueden incluir una función especial para la lectura de señales de corta duración.

Si la función existe, el flanco de señal de entrada, la cual debe mantenerse en ON durante un tiempo mínimo, activará un relé interno que mantiene su estado hasta el final del siguiente "Scan".

Posibilidades de Configuración

Por configuración del autómata se entiende la estructura que adopta su sistema físico, esencialmente la unidad de control, el sistema de entradas/salidas y la memoria de la máquina a fin de adaptarlo a las particularidades de la aplicación: número de entradas salidas, ubicación de la misma, potencia de cálculo, capacidad de almacenamiento, etc. (Josep Balcells, 2000).

Existen básicamente dos configuraciones posibles para la unidad de control:

- Unidad de control compacta (control centralizado).
- Unida de control modular (control distribuido).

Unidad de control compacta (control centralizado). Una sola CPU o unidad central de proceso soporta varios módulos de entradas/salidas, que contienen

exclusivamente interfaces E/S. Puesto que estos módulos no pueden funcionar de forma autónoma, el control queda centralizado en la CPU única.

Unida de control modular (control distribuido). Existen varios módulos con sus propias unidades de proceso, y, en general con sus propias interfaces de E/S. Estos módulos se encuentran conectados a una unidad de central (CPU principal, maestra, supervisora, etc.) que gestiona de forma general el sistema y permite el intercambio de datos entre los elementos, procesadores o interfaces del mismo.

En cuanto a la configuración del sistema de entradas/salidas, éste puede ser a su vez:

- Centralizados
- Distribuidos

Las interfaces de entrada/salida son centralizadas cuando se comunican con el autómata a través de su bus interno, sin necesidad de utilizar procesadores de comunicaciones intermedios.

Por el contrario, si la comunicación necesita de estos procesadores, conectados sobre el bus interno y unidos a los periféricos con cables de conexión apropiados, el sistema de entradas/salidas se denomina distribuido.

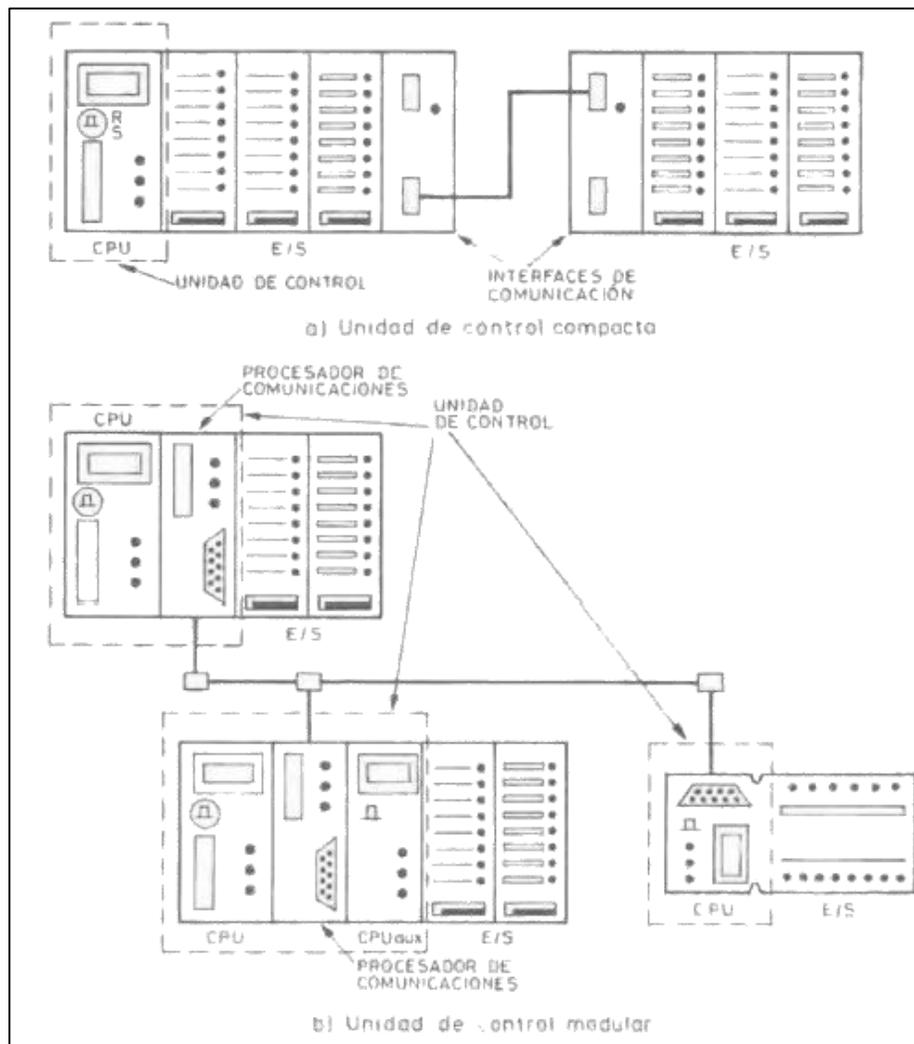


ILUSTRACIÓN 5. CONFIGURACIÓN COMPACTA Y MODULAR DE LA UNIDAD DE CONTROL (JOSEP BALCELLS, 2000)

Interfaces de Entrada/Salida

El control del proceso requiere siempre un dialogo operador-máquina y una comunicación entre máquina y sistema de control. Para el sistema de control, y más concretamente si éste es un autómeta, se traduce en que éste reciba una serie de señales externas, unas de mando y otras de realimentación que en conjunto constituyen las entradas. Por otro lado el operador debe conocer algunos datos sobre la marcha del proceso y los accionamientos, necesita recibir las órdenes oportunas para controlarlo. Todo ello se realiza a través de las salidas del autómeta (Josep Balcells, 2000).

Unidades de Entradas

Son los dispositivos básicos por donde llega la información de los sensores. Distinguimos dos tipos de entradas (Facultad de Ciencias, Tecnologías Virtual, Electrónica II, 2015):

- **Analógicas.** Se debe usar cuando la entrada corresponde a una medida por ejemplo: temperatura, presión o caudal. En su interior tienen un dispositivo que convierte la señal analógica en digital (convertor A/D). Vienen en distintos rangos de tensión e intensidad. (Por ejemplo 0 a 10V, 0 a +-10V, 4 a 20 mA, etc.). La resolución puede ser de 8 a 12 bits.
- **Digitales.** Son las más utilizadas y corresponden a señales activado-desactivado (0 o 1). O sea la presencia o no de una tensión (por ejemplo de fines de carrera, selectores, termostatos, pulsadores, etc.). Esta tensión puede ser alterna (0-220V, 0-110V) o continua (generalmente 0-24V).

Unidades de Salida

Son los bloques básicos que excitarán los actuadores. Al igual que las entradas, pueden ser analógicas o digitales (Facultad de Ciencias, Tecnologías Virtual, Electrónica II, 2015).

- **Analógicas.** Se debe usar cuando el actuador que se debe activar es analógico (por ejemplo válvula modulante, válvulas de control de flujo, un variador de velocidad, etc.). En este caso se dispone de un dispositivo interno que realiza el proceso inverso al de las entradas analógicas, un convertor D/A.
- **Digitales.** Vienen en tres tipos. Con salida a triac, a relé o a transmisor. En el primer caso es exclusivamente para corriente alterna. En el segundo caso puede ser para continua o alterna. En el tercer caso de salida a transitar es exclusivamente para continua. Soportan en todos los casos corriente entre 0.5 y 2 A.

Las entradas y salidas digitales vienen normalmente integradas en el propio PLC, particularmente en el caso de autómatas compactos como el CPM2A. Por el contrario, para las analógicas debemos disponer de un modelo de expansión adicional, (como el MAD01 que dispone de 2 entradas y 1 salida digitales, de 8 bits de resolución) y se conectan a través de una bahía de expansión de periféricos. No todos los autómatas admiten módulos de expansión analógicos, aunque cada vez es más frecuente incluso en módulos básicos.

A continuación se muestran las dos distintos tipos de señales:

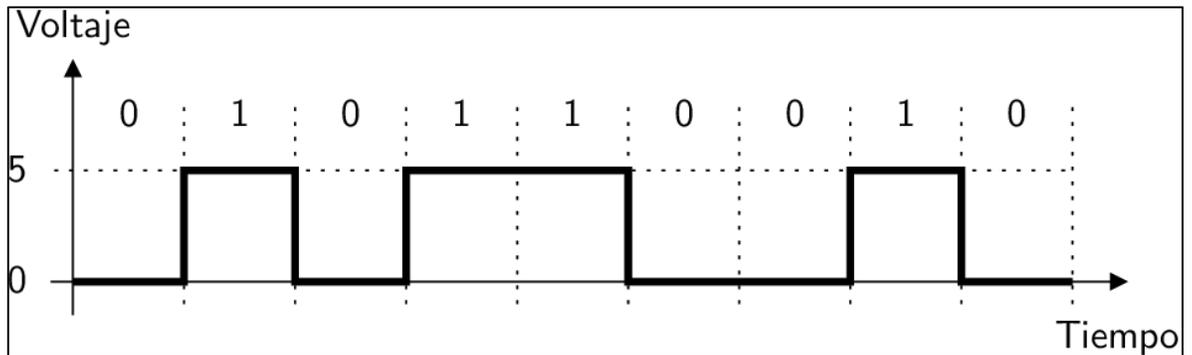


ILUSTRACIÓN 6. SEÑAL DIGITAL

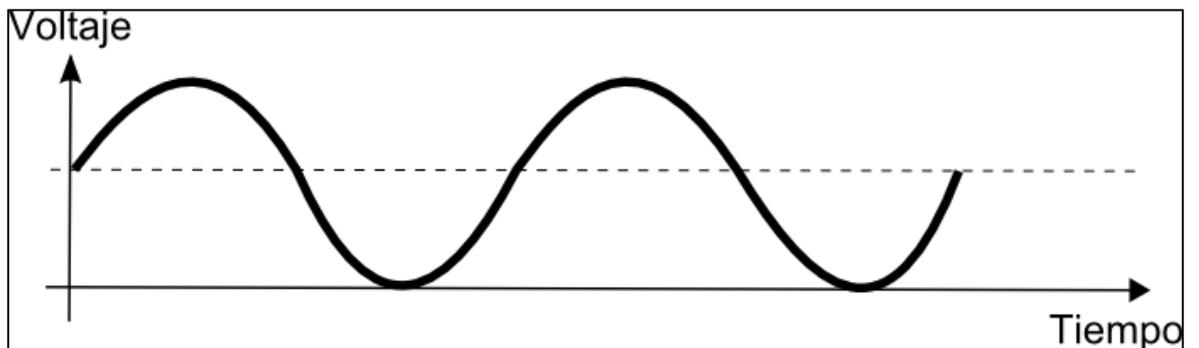


ILUSTRACIÓN 7. SEÑAL ANALÓGICA

Interfaces Específicas

Las interfaces específicas permiten la conexión con elementos o procesos particularmente de la planta, realizando funciones que van desde la lectura directa de termopares hasta la presentación de sinópticos y control SCADA, pasando por regulación o posicionamiento de ejes. (Josep Balcells, 2000).

Estas interfaces específicas están disponibles para autómatas modulares gama media y alta, máquinas que exigen, por su mayor campo de aplicaciones, mayor personalización que los micro miniautómatas.

Aunque a todas estas interfaces se les denomina en general específicas por lo concreto de la función que realizan, se puede distinguir entre ellos tres grupos bien diferenciados:

- Entradas/salidas especiales
- Entradas/salidas inteligentes
- Procesadores periféricos inteligentes

Entradas/salidas especiales. Se caracterizan por entregar o recibir señales particulares, bien por su forma, bien por su aplicación, pero sin incluir en ningún caso control sobre variables de planta, sino solo tratamiento de la señal para hacerla legible a la CPU (interfaces de entrada), o a la planta (interfaces de salida). Este tratamiento está predeterminado y no es modificable por el usuario, que únicamente puede actuar, bien por instrucciones de programa, bien mediante microinterruptores, sobre los modos de trabajo y sobre algunos parámetros de la tarjeta.

Entradas/salidas inteligentes. Admiten, en cambio, múltiples modos de configuración, ordenados por el programa y disponen de salidas binarias controladas desde la misma tarjeta, lo que permite implementar lazos de control ON/OFF sobre variables de planta, en funcionamiento transparente para la CPU.

Las consignas y controles necesarios son enviados por el programa de usuario, desde la CPU principal.

De esta forma se descarga el trabajo a la unidad central y se mejora indirectamente su capacidad de direccionamiento al abrirse la posibilidad de acceder a señales de entrada/salida que no tienen por qué aparecer en la memoria.

Procesadores periféricos inteligentes. Este grupo de interfaces específicas cuenta con características diferenciales propias, es el de los procesadores periféricos inteligentes, tarjetas o módulos que incluyen su propio procesador, memoria y puntos auxiliares de entrada/salida.

Este procesador contiene en origen un programa o interprete de programa especializado en ejecución de una tarea concreta, al que le basta conocer los puntos de consigna y los parámetros de la aplicación para ejecutar, de forma autónoma e independiente de la CPU principalmente, el programa de control.

A diferencia de las entradas/salidas inteligentes, que limitan a unos pocos los parámetros necesarios para su funcionamiento (funciones muy especializadas), los procesadores periféricos de uso más general, necesitan más información para definirlo, no solo la configuración de periférico, sino además:

- Condiciones de aplicación, que definen las condiciones del entorno.
- Consignar a seguir.
- Condiciones de control, que definen la respuesta según la evolución del proceso.

Elementos de Control de Procesos Industriales: Sensores y Actuadores

Los términos "sensor" y "transductor" se suelen aceptar como sinónimos, aunque, si hubiera que hacer alguna distinción, el término transductor es quizás más amplio, incluyendo una parte sensible o "captador" propiamente dicho y algún tipo de

circuito de acondicionamiento de la señal detectada. Si nos centramos en el estudio de los transductores cuya salida es una señal eléctrica, podemos dar la siguiente definición:

Un transductor es un dispositivo capaz de convertir el valor de una magnitud física en una señal eléctrica codificada, va sea en forma analógica o digital (Josep Balcells, 2000).

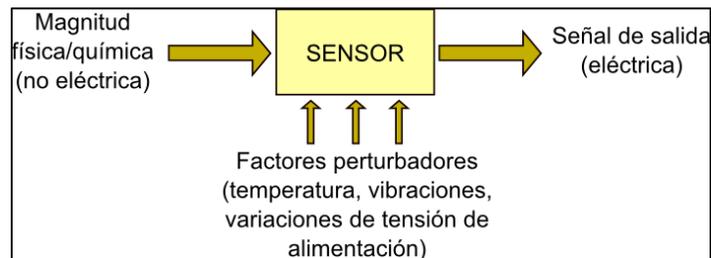


ILUSTRACIÓN 8. DIAGRAMA GENERAL DE UN SENSOR

No todos los transductores tienen por qué dar una salida en forma de señal eléctrica. Como ejemplo puede valer el caso de un termómetro basado en la diferencia de dilatación de una lámina bimetálica, donde la temperatura se convierte directamente en un desplazamiento de una aguja indicadora.

Los transductores basados en fenómenos eléctricos o magnéticos, éstos suelen tener una estructura general, en la cual podemos distinguir las siguientes partes:

- Elemento sensor o captador. Convierte las variaciones de una magnitud física en variaciones de una magnitud eléctrica o magnética, que denominaremos habitualmente señal.
- Bloque de tratamiento de señal. Si existe, suele filtrar, amplificar, linealizar y, en general, modificar la señal obtenida en el captador, por regla general utilizando circuitos electrónicos.
- Etapa de salida. Esta etapa comprende los amplificadores, interruptores, conversores de código, transmisores y en general, todas aquellas partes que adaptan la señal a las necesidades de la carga exterior.

Detectores de Proximidad

Los detectores de proximidad pueden estar basados en distintos tipos de captadores (Josep Balcells, 2000), siendo los más frecuentes:

- Detectores inductivos
- Detectores capacitivos
- Detectores ópticos
- Detectores ultrasónicos

Por lo general se trata de sensores con respuesta todo o nada, con una cierta histéresis (un transductor presenta histéresis cuando, a igualdad de la magnitud de entrada, la salida depende de si dicha entrada se alcanzó con aumentos en sentido creciente o en sentido decreciente) en la distancia de detección y con salida a base de interruptor estático (transistor, tiristor o triac), pudiendo actuar como interruptores de CC o CA. Pero algunos de ellos pueden llegar a da una salida analógica proporcional a la distancia.

Detectores Inductivos

Este tipo de detectores sirven para detectar la proximidad de piezas metálicas en un rango de distancia que va desde 1 mm a unos 30 mm, con una posible resolución de orden de décimas de milímetro. La ejecución mecánica y eléctrica está normalizada a nivel europeo por CENELEC (normas EN 50.032, EN 50.036, EN 50.037, EN 50.038)

La distancia de detección está definida según la norma para una placa cuadrada de acero ST57 de 1 mm de espesor y de dimensiones acordes al diámetro del cabezal sensible. Para otros tipos de metal y otras dimensiones la distancia nominal de detección debe corregirse con un factor de valores entres 0.4 a 0.6 y 1.

La variación de amplitud de la oscilación, provocada por la presencia del metal frente al cabezal detector, puede utilizarse para obtener una señal analógica de posición. El detector de proximidad da entonces una señal que es proporcional a la

distancia. Sin embargo, la medida es muy precisa, depende mucho del tipo de metal y de las condiciones del ambiente.

El campo de aplicación más importante de los detectores inductivos es como interruptor final de carrera con algunas ventajas con respecto a los electromecánicos, tales como: ausencia de contacto con el objeto a detectar, robustez mecánica, resistencia a ambientes agresivos y altas temperaturas y bajo precio.

Detectores Capacitivos

El principio de funcionamiento y las características son análogas a las descritas para los detectores inductivos, pero en este caso el elemento sensible es el conductor del circuito oscilante, formado por dos aros metálicos concéntricos situados en la cara sensible, y cuyo dieléctrico es el material de la zona sensible.

Este tipo de sensores permite detectar materiales metálicos o no, pero su sensibilidad se ve afectada por el tipo de material y por el grado de humedad y del cuerpo a detectar. Por ello se utilizan exclusivamente como detectores de todo-nada, con una respetabilidad bastante dependiente de las condiciones ambientales.

Para lidiar con el problema de dependencia de la sensibilidad con el tipo de material, se suelen construir con un ajuste sensible que permite utilizarlos para la detección de algunos materiales entre otros (por ejemplo, aluminio entre cobre o latón). Las aplicaciones típicas son la detección de materiales no metálicos como vidrio, cerámica, plástico, madera, aceite, agua, cartón y papel.

En cuanto a las formas de ejecución mecánica, tipos de alimentación y formas de conexión son idénticas a las de los detectores inductivos.

Detectores Ópticos

Los detectores ópticos emplean fotocélulas como elementos de detección. Algunos tipos disponen de un cabezal que incorpora un emisor de luz reflejado sobre el

objeto que se pretende detectar. Otros tipos trabajan a modo de barrera y están previstos para detección a mayores distancias con fuentes luminosas independientes del cabezal detector. Ambos tipos suelen trabajar con frecuencia luminosa en la gama de los infrarrojos.

Las características particulares de los detectores de proximidad óptica, respecto a otros detectores de proximidad son:

- Elevada inmunidad a perturbaciones electromagnéticas (EMI) externas.
- Distancia de detección grandes respecto a los inductivos o capacitivos. Se obtienen fácilmente hasta 500 m en modo barrera, y hasta 5 m por reflexión.
- Alta velocidad de respuesta y frecuencia de comunicación.
- Permiten la identificación de colores.
- Capaces de detectar objetos del tamaño de decimas de milímetros.

Otras variantes de detectores ópticos son de fibra óptica, que tienen los puntos de emisión/recepción de luz separados de la unidad generadora, y unidos a ella mediante la fibra. De esta forma, la detección puede llevarse a puntos inaccesibles para las fotocélulas de barrera o reflexión, aprovechando la flexibilidad de la fibra. Estos detectores, disponibles en versiones de reflexión y de barrera, tienen distancia de detección desde 3 mm a 10 metros, pudiendo detectar objetos pequeños como marcas de 1 mm sobre piezas que se mueven a gran velocidad (30 m/s), u objetos estáticos de hasta unas 20 micras de diámetro.

Detectores Ultrasónicos

Estos detectores están basados en la emisión-recepción de ondas ultrasónicas. Cuando un objeto interrumpe el haz, el nivel de recepción varía y el receptor lo detecta.

Como ventaja frente a las fotocélulas, los detectores ultrasónicos pueden detectar con facilidad objetos transparentes, como cristal y plásticos, materiales que ofrecen dificultades para la detección óptica.

Sin embargo, y dado que estos detectores utilizan ondas ultrasónicas que se mueven por el aire, no podrán ser utilizados en lugares donde este circule con violencia (bocas de aire acondicionado, cercanías de puertas, etc.), o en medios de elevada contaminación acústica (prensas, choque entre metales, etc.).

Medidores de Posición o Distancia

Dentro de los transductores de posición podemos distinguir dos grandes grupos: (Josep Balcells, 2000).

- Los indicadores de posición lineal o angular para grandes distancias, conocidos también como sistemas de medición de coordenadas.
- Los detectores de pequeñas deformaciones o detectores de presencia de objetos a una cierta distancia que dan una señal analógica o digital proporcional a dicha distancia.

Los *medidores de Coordenadas* se utilizan, por lo general para determinar la posición relativa de partes móviles de una máquina. Se trata de transductores de desplazamiento relativo previstos para medición indirecta de distancias, se dice medición indirecta por que no permite determinar la distancia entre objetos estáticos, sino únicamente la posición relativa del objetos a partir de un origen de desplazamiento. Su característica esencial es que permite medir grandes distancias con una excelente resolución y se usan, sobre todo, en el campo de la robótica.

El segundo grupo de detectores, es decir, los *detectores de pequeñas distancias*, permiten determinar distancias entre un objeto estático o en movimiento respecto al cabezal del sensor.

Potenciómetro

El potenciómetro es un transductor de posición angular, de tipo absoluto y con salida de tipo analógico. Consiste en una resistencia de hilo bobinado o en unas de material conductor, distribuida a lo largo de un soporte en forma de arco y un cursor solidario a un eje de salida, que puede deslizarse sobre dicho conductor. El

movimiento del eje arrastra el cursor provocando cambios de resistencia entre este y cualquiera de los extremos de la resistencia con una tensión constante.

Encoders

Los Encoders son dispositivos formados por un rotor con uno o varios grupos de bandas opacas y traslúcidas alternadas y por una serie de captadores ópticos alojados en el estator, que detectan la presencia o no de la banda opaca frente a ellos. Existen dos tipos de Encoders: los *incrementales* y los *absolutos*.

Los primeros dan un determinado número de impulsos por vuelta y requieren de un contador para determinar la posición a partir del origen de referencia. Los absolutos, en cambio, disponen de varias bandas en el rotor ordenados según un código binario. Los captadores ópticos detectan, pues, un código digital completo, que es único para cada posición del rotor.

Sincros y Resolvers

Un síncrono es un transductor de posición angular de tipo electromagnético, cuyo principio de funcionamiento puede resumirse diciendo que se trata de un transformador con uno de sus devanados rotativo.

Existen diversos tipos de síncronos, dependiendo el número de devanados y su posición, pero las configuraciones más frecuentes son:

- Primario alojado en el rotor y, en general, monofásico.
- Segundo alojado en el estator y, en general, trifásico.

En algunos servos de posición se utilizan pares de síncronos en una configuración denominada transmisor-receptor o “maestro-esclavo” que permite generar una señal proporcional a la diferencia de ángulos de los dos síncronos interconectados. Esta configuración encuentra su aplicación en servos de seguimiento, copiadoras, pilotos automáticos, etc.

El síncrono maestro es alimentado por el rotor, el tiempo de dicho rotor es accionado mecánicamente siguiendo, por ejemplo, un contorno en el caso de las copadoras.

Una configuración particular de síncronos es la de los resolver, cuyo principio de funcionamiento es análogo al indicado anteriormente, con las siguientes particularidades constructivas:

- Primario alojado en el estator, y generalmente bifásico.
- Segundo alojado en el rotor, que puede ser monofásico o bifásico.

Los devanados del estator se alimentan generalmente en serie (sistema monofásico), dando un campo estacionario sobre el eje y los devanados del rotor (bifásico).

Inductosyn

El Inductosyn es un transductor electromagnético utilizado para la medida de desplazamientos lineales, con una precisión del orden de micras. Se utiliza en la actualidad como medidor de coordenadas en muchísimas máquinas-herramientas y de control numérico.

Sensor Laser

Los sensores laser pueden utilizarse como detectores de distancia, utilizando técnicas de reflexión y triangulación parecidas a las de los otros detectores óptico, o permiten también la utilización como detectores de desplazamiento por análisis de interferencia en la emisión-recepción de un mismo rayo. En este último caso la medición de distancia se hace constando crestas y valles en la interferencia y, por lo tanto, por un principio similar a un encoder incremental.

El principio de funcionamiento del interferómetro laser se basa en la superposición de dos ondas de igual frecuencia, una directa y otra reflejada. Si las ondas están en fase, la superposición es aditiva, y si están en contrafase, sustractiva. La onda

resultante de la superposición pasa por valores máximos y mínimos al variar la fase de la señal reflejada.

Los sensores industriales basado en este principio general un haz de luz que se divide en dos partes ortogonales mediante un separador. Un haz se aplica directamente sobre un espejo plano fijo, mientras que el otro se refleja en el objeto cuya distancia se quiere determinar. Los dos haces se superponen de nuevo en el separador, de forma que al desplazarse el objeto a detectar se generan máximos y mínimos de amplitud a cada múltiplo de la longitud de onda del haz. El desplazamiento, o diferencia relativa de posiciones, se determina contando dichas oscilaciones o franjas, obteniéndose una salida digital de elevada precisión, con resolución de orden de la longitud de ondas de luz empleadas (unos 50 μm en sensores industriales).

Sensores Ultrasónicos

Los sensores ultrasónicos emiten una señal de presión hacia el objeto cuya distancia se pretende medir, y miden el tiempo transcurrido entre la emisión del pulso hasta la recepción del eco reflejado en dicho objeto. El más conocido de estos es quizás el “sonar” que incorporan los submarinos y actualmente los sistemas de ecografía o incluso en el medio natural, la forma de detectar obstáculos, que emplean los murciélagos en la oscuridad.

En el campo industrial se suelen emplear para controlar niveles de sólidos o áridos en silos, presencia de obstáculos en el campo de alcance de robots, etc. La frecuencia de la perturbación emitida está en el rango de 20 a 40 kHz.

Medidores de Pequeños Desplazamientos y Deformaciones

Este tipo de sensores se utiliza para la detección de pequeños desplazamientos, deformaciones y planitud de superficies. Se emplean también unidos sólidamente a sólidos deformables, como transductores indirectos de esfuerzos, par o impar (Josep Balcells, 2000).

Transductores Piezoeléctricos

Algunos elementos cristalinos como el cuarzo, y otros minerales sintéticos poseen la propiedad de adquirir una polarización en la dirección de los denominados ejes eléctricos, cuando se les somete a un esfuerzo y se deforman según la dirección de los llamados ejes mecánicos.

El fenómeno se debe al desplazamiento que sufre el centro de gravedad de las cargas o iones positivos y negativos, generándose un efecto de *dipolo eléctrico*. Esta propiedad se aprovecha para obtener sensores de deformación o indirectamente de fuerza, par o impar. Las formas de aprovechar este efecto pueden ser básicamente dos:

- Medición de carga de polarización.
- Medición de la frecuencia de oscilación.

Los sensores basados en el primer método miden la densidad de carga superficial que aparece en la dirección de los ejes eléctricos al someter al cristal a una presión en la dirección de los ejes mecánicos. Dicha densidad de carga es proporcional a la presión.

El segundo método aplicable con los cristales piezoeléctricos se basa en la medición de su frecuencia de oscilación cuando se les coloca en un circuito con realimentación positiva. Un cristal con electrodos dispuestos en la dirección de los ejes eléctricos tiene una frecuencia propia de oscilación que depende básicamente de sus dimensiones.

Transductores de Temperatura

La temperatura es otro de los parámetros que muchas veces debe controlarse en los procesos industriales. Atendiendo al principio de funcionamiento de la mayoría de los sensores industriales (Josep Balcells, 2000), podemos distinguir tres grandes grupos de sensores térmicos:

- Termostatos todo-nada: Interruptores que conmutan a un cierto valor de temperatura, en general con una cierta histéresis.
- Termorresistencia: Sensores pasivos de tipo analógico basado en el cambio de resistividad eléctrica de algunos metales o semiconductores con la temperatura.
- Pirómetros de radiación: Sensores de tipo analógico, utilizables en general para altas temperaturas, que están basados en la radiación térmica emitida por los cuerpos calientes.

Termostatos

Los termostatos son señores con salida de tipo todo o nada que conmuta a un cierto calor de la temperatura. Los más simples están basados generalmente en la diferencia de dilatación de metales y los más sofisticados se suelen construir a base de un sensor de tipo analógico y uno o varios comparadores con histéresis.

Los de tipo bimetálico se utilizan típicamente en sistemas de climatización y algunas aplicaciones industriales como interruptores de protección (falta de ventilación, etc.).

Los formados por una sonda analógica y un sistema de comparación tienen la ventaja de ser, en general, regulables y de poder utilizar sondas de muy pequeño tamaño que pueden ubicarse en el interior del bobinado, maquinas, recintos con atmosfera explosiva u otros emplazamientos donde se requiere que ocupen poco espacio o que se produzca arco eléctrico por apertura de un circuito.

Termopares

Los termopares son sensores activos de tipo analógico basados en el efecto Seebeck. Dicho efecto consiste en la aparición de una tensión eléctrica entre dos piezas de distintos metales unidas o soldadas por un extremo, cuando éste se calienta (unión caliente) y se mantienen los otros dos extremos a una misma temperatura inferior (unión fría).

La fuerza electromotriz generada depende de la diferencia de temperaturas entre la unión fría y la unión caliente.

Para ciertos materiales existe una relación bastante lineal entre la diferencia de temperaturas y la fuerza electro magnética generada, siendo, por tanto, muy adecuados como transductores. Sin embargo, para recoger esta f.e.m. se deberán conectar los extremos fríos a conductores de cobre u otro metal y por el mismo efecto aparecerá una f.e.m. de contacto, que sólo se compensarán en el caso de que ambas uniones frías se mantengan a idéntica temperatura.

Por otro lado, para que la tensión de salida sea proporcional a la temperatura en la unión caliente, debe mantenerse constante la temperatura de las uniones frías, o compensar la fuerza electro magnética que darían lugar sus variaciones mediante un circuito adicional. Si se requiere una buena precisión, se prefiere generalmente compensar las variaciones de temperatura en la unión fría, aunque esto requiere un sensor adicional, que suele ser una NTC.

Accionamientos: Clasificación

El accionamiento puede estar bajo el control directo de la parte de mando o puede requerir algún preaccionamiento para amplificar la señal de mando. (Josep Balcells, 2000).

La gama de posibles accionamientos que puede controlar un autómata programable es enormemente extensa y variada. Entre los más habituales se encuentran los destinados a producir movimiento (motores y cilindros), los destinados a trasiego de fluidos (bombas) y los de tipo térmico (hornos, intercambiadores, etc.).

Para empezar podemos establecer una clasificación atendiendo a la tecnología o, si se quiere, dependiendo del tipo de energía empleada en el accionamiento.

Según esto podemos distinguir:

- Accionamientos eléctricos.

- Accionamientos hidráulicos.
- Accionamientos neumáticos.
- Accionamientos térmicos.

Dentro de cada una de estas tecnologías encontramos, a su vez, accionamientos de dos tipos:

- Accionamiento todo o nada.
- Accionamiento de tipo continuo.

Accionamientos Eléctricos

De alguna manera, todos los preaccionamientos que se conectan a los autómatas suelen tener mando eléctrico. Pero dentro de este apartado nos referiremos únicamente a relés, contactores y servomotores de tipo eléctrico.

- Relés y contactores
- Servomotores de CC
- Servomotores de CA
- Motores paso a paso

Accionamientos Hidráulicos y Neumáticos

Los accionamientos hidráulicos y neumáticos se aplican de forma masiva en los automatismos industriales, gracias a su robustez y facilidad de control.

- Válvulas
- Servoválvulas
- Cilindros
- Sujeción por vacío
- Bombas y motores hidráulicos

Capítulo 3. Plataforma Arduino.

¿Qué es Arduino?

Arduino es una plataforma de hardware libre (open source), basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

Se puede usar Arduino para crear objetos interactivos, leyendo datos de una gran variedad de interruptores y sensores y controlar multitud de tipos de luces, motores y otros actuadores físicos. Los proyectos con Arduino pueden ser autónomos o comunicarse con un programa (software) que se ejecute en tu ordenador. La placa puedes montarla tú mismo o comprarla ya lista para usar, y el software de desarrollo es abierto y lo puedes descargar gratis desde la página (Arduino).

¿Qué Quiere Decir que Arduino sea “Hardware Libre”?

El hardware libre (también llamado “open-source” o “de fuente abierta”) comparte muchos de los principios y metodologías del software libre. En particular, el hardware libre permite que la gente pueda estudiarlo para entender su funcionamiento, modificarlo, reutilizarlo, mejorarlo y compartir dichos cambios. Para conseguir esto, la comunidad ha de poder tener acceso a los ficheros esquemáticos del diseño del hardware en cuestión. Estos ficheros detallan toda la información necesaria para que cualquier persona con los materiales, herramientas y conocimientos adecuados pueda reconstruir dicho hardware por su cuenta sin problemas, ya que consultando estos ficheros se puede conocer qué componentes individuales integran el hardware y qué interconexiones existen entre cada uno de ellos. (Artero, 2013)

¿Porque Elegir Arduino?

Existen muchas otras placas de diferentes fabricantes que, aunque incorporan diferentes modelos de microcontroladores, son comparables y ofrecen una funcionalidad más o menos similar a la de las placas Arduino. Todas ellas también vienen acompañadas de un entorno de desarrollo agradable, cómodo y de un lenguaje de programación sencillo y completo. No obstante, la plataforma Arduino (hardware más software) ofrece una serie de ventajas: (Artero, 2013)

- Arduino es libre y extensible.
- Arduino tiene una gran comunidad.
- Su entorno de programación es multiplataforma.
- Su entorno y el lenguaje de programación son simples y claros.
- Las placas Arduino son baratas.
- Las placas Arduino son reutilizables y versátiles

Elementos de Arduino

Hardware Elementos

La placa consiste en un microcontrolador Atmel AVR de 8 bits, que permite facilitar la programación y nos permite incorporarlos con otros circuitos. Un aspecto importante es la manera en que los conectores son expuestos permitiendo a la CPU conectarse con otros módulos expansivos (Shields). Utiliza la serie de chips megaAVR, especialmente los ATmega8, ATmega168, ATmega328 y ATmega1280.

La mayoría de las placas incluyen un regulador de 5 volts y un oscilador de cristal de 16 MHz. Regula a 5 volts estabilizados cualquier tensión (entre 7 y 12 voltios)

Cuando se utiliza su software, este establece una conexión serial RS-232, pero la manera en la que se implementa en el hardware varía en cada versión. Actualmente existen algunos métodos diferentes para realizar la transmisión de datos, mediante

las placas programables vías USB, incorporadas a través de un chip adaptador USB para serial.

Las placas tienen 14 pines digitales, cada uno de los cuales puede fijarse como una entrada o salida, y seis entradas analógicas. Además seis de los pines digitales se pueden programar para proporcionar salidas analógicas de modulación de ancho de pulsos (PWM). Cuenta con una variedad de protocolos de comunicación incluyendo el de serie, bus de interfaz periférica en serie (SPI), y I2c/TWI incluida en cada placa como característica estándar y un botón de reinicio.

Placas Arduino

- Arduino Uno
- Arduino 101
- Arduino Pro
- Arduino Pro Mini
- Arduino Micro
- Arduino Nano
- Arduino Leonardo
- Arduino Gema
- Arduino Mega 2560
- Arduino Mega ADK
- Arduino BT (Bluetooth)
- Arduino Due
- Arduino Yun
- Arduino Ethernet
- Arduino Fio
- Arduino Lilypad

Modelo	Microcontrolador	Voltaje	I/O Digitales	Entradas Analógicas	Memoria Flas	Velocidad de reloj
UNO	ATmega328	5	14	6	32 KB	16 MHz
Leonardo	ATmega32u4	5	20	12	32 KB	16 MHz
Due	AT91SAM3X8E	3.3	54	12	512 KB	84 MHz
Yún	ATmega32u4	5	20	12	32 KB	16 MHz
Robot	ATmega32u4	5	5	4	32 KB	16 MHz
Explora	ATmega32u4	5			32 KB	16 MHz
Mega ADK	ATmega2560	5	54	16	256 KB	16 MHz
Ethernet	ATmega328	5	14	6	32 KB	16 MHz
Mega 2560	ATmega2560	5	54	16	256 KB	16 MHz
Mini	ATmega328	5	14	8	32 KB	16 MHz
Nano	ATmega168	5	14	8	16 KB	16 MHz
Pro Mini	ATmega168	3.3	14	8	17 KB	8 MHz
Pro	ATmega168	3.3	14	6	16 KB	8 MHz
Micro	ATmega32u4	5	20	12	32 KB	16 MHz
Fio	ATmega328P	3.3	14	8	32 KB	8 MHz

TABLA 2. CARACTERÍSTICAS GENERALES DE VERSIONES DE ARDUINO

“Shield” Arduino

En Arduino el término Shield se utiliza denominar el hardware que permite otras funciones específicas como movimiento de brazos mecánicos y comunicación en redes.

A continuación se dan ejemplos de Shields:

Arduino Motor Shield.- Este Shield nos permite dominar la velocidad y sentido de giro de hasta dos motores DC de forma independiente o bien estas dos magnitudes de un motor pasó a pasó. También podemos realizar diferentes medidas sobre las capacidades de los motores conectados.

Está diseñado para controlar componentes que contienen inductores (bobinas) en su estructura interna, tales como relés, solenoides, motores DC (corriente continua) o motores paso a paso, entre otros.

Arduino Proto Shield.- Este Shield permite de forma fácil diseñar e implementar circuitos personales. Ofrecer un área de trabajo donde se pueden soldar los diferentes componentes electrónicos que necesitemos para montar nuestro proyecto. Teniendo de forma muy compacta todo un circuito completo.

Arduino Ethernet Shield.- Este Shield permite a una placa Arduino la capacidad de conectarse a una red cableada TCP/IP, configurándose con la librería “Ethernet” la cual ya viene por defecto en el lenguaje de Arduino.

Arduino GSM Shield.- Este Shield permite hacer llamadas telefónicas, enviar SMS y conectarse a internet utilizando la red inalámbrica GPRS. Solo se tiene que conectar una tarjeta SIM de un operador de cobertura GPRS. Para realizar y/o recibir las llamadas de voz es necesario un circuito de altavoz y micrófono externo.

Arduino WiFi Shield.- Este Shield permite a una placa Arduino la capacidad de conectarse Inalámbricamente a una red TCP/IP. Incorpora el chip HDG104 del

fabricante H&D Wireless, el cual incluye una antena integrada y viene preprogramado de fábrica para proporcionar una IP completa.

Se puede conectar a redes abiertas, o protegidas con encriptado de tipo WEP o WPA2. Para poder gestionar este Shield es necesario utilizar la librería “Wifi” la cual viene por defecto en el lenguaje de Arduino.

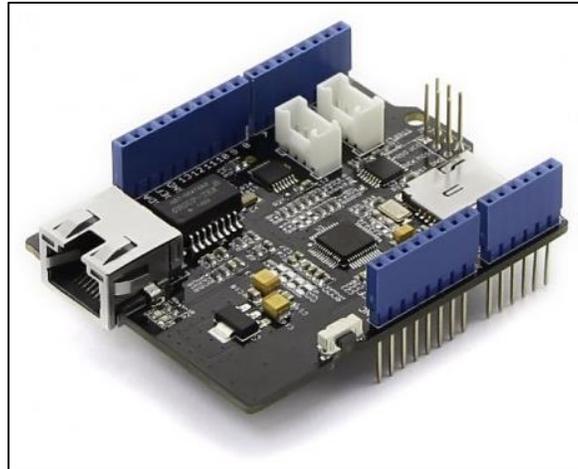


ILUSTRACIÓN 9. ARDUINO ETHERNET SHIELD (ELECTRONICA ESTUDIO.CO, INGENIERÍA ELECTRÓNICA Y PROYECTOS PICMICRO, S.F.)

Software

IDE

Las siglas IDE vienen de Integrated Development Environment (Entorno de Desarrollo Integrado). Esto es un conjunto de herramientas que permite a los programadores poder desarrollar sus propios programas con comodidad. En el caso de Arduino, necesitamos un IDE que nos permita escribir y editar nuestro programa (también llamado “sketch” en el mundo de Arduino), que nos permita comprobar que no hayamos cometido ningún error y que además nos permita grabarlo en la memoria del microcontrolador de la placa Arduino para que este se convierta a partir de entonces en el ejecutor autónomo de dicho programa.

Para poder empezar a desarrollar nuestros propios sketches, deberemos instalar en nuestro computador el IDE que nos proporciona el proyecto Arduino.

El software de Arduino se ejecuta en sistemas operativos Windows, Macintosh OSX y GNU/Linux.

IDEs Alternativos

Actualmente existe una variedad de IDEs alternativos que aportan más características que el IDE original o simplemente que cambia la manera de trabajar. Algunos IDEs alternativos pueden ser los siguientes:

- **CodeBlocks:** IDE libre y multiplataforma para el desarrollo de aplicaciones escritas en el lenguaje C y C++.
- **Gnduino:** IDE libre con aspecto y funcionalidad similar al oficial, pero evita su dependencia con Java ya que está escrito en el lenguaje Python, solo funciona en Linux.
- **Codebernder:** aplicación web totalmente “online”. Incluye, editor de texto, compilador y cargador de sketch.
- **Visualmicro:** es un complemento (plug-in) para usar el lenguaje Arduino dentro del entorno de programación de Visual Studio, no funciona en la versión Express solo en ediciones superiores.
- **EmbedXcode:** es un complemento (plug-in) del entorno de programación oficial de Apple, llamado XCode. Permite trabajar con un IDE “todo en uno” en el cual se puede programar de forma unificada diversas plataformas (Arduino, chipKIT, Maple, Wiring), cada una con su lenguaje de programación propio.

La Comunicación Serie con la Placa Arduino

El microcontrolador dispone de un receptor/transmisor serie de tipo TTL-UART que permite comunicar la placa Arduino con otros dispositivos (normalmente, nuestro computador), para así poder transferir datos entre ambos. El canal físico de comunicación en estos casos suele ser el cable USB, pero también pueden ser los pines digitales 0 (RX) y 1 (TX) de la placa. Hay que tener en cuenta que si se utilizan

estos dos pines para la comunicación serie, no podrán ser usados entonces como entradas/salidas digitales estándar.

Estructura del Programa

La estructura básica del lenguaje de programación Arduino es bastante simple y se organiza en al menos dos partes o funciones que encierran bloques de declaraciones.

```
void setup()
```

```
{  
  
    declaraciones;  
  
}
```

```
void loop()
```

```
{  
  
    declaraciones;  
  
}
```

Ambas funciones son requeridas para que el programa funcione.

Las instrucciones escritas dentro de la sección “**void setup()**” se ejecutan una única vez, en el momento de encender (o iniciar) la placa Arduino.

Las instrucciones escritas dentro de la sección “**void loop()**” se ejecutan justo después de las de la sección “**void setup()**” infinitas veces hasta que la placa se apague (o se reinicie). En ésta sección se escriben las instrucciones para enviar códigos al demás hardware que debe realizar una actividad específica, además de poder enviar datos a la Interfaz realizada en Java para tener una comunicación con el usuario.

Tipos de Datos en Arduino

Como todo lenguaje de programación es necesario utilizar variables, que son áreas de memoria de la placa Arduino, donde se podrá almacenar datos para poder manipularlos, por lo cual es necesario especificar el tipo de dato que contendrá. En la Tabla 3 se mencionan distintos de datos.

Tipo de dato	Característica
boolean	Valores de cierto o falso
char	Solo un carácter (letra, dígito, signo de puntuación...)
byte	Numero entero entre 0 y 255
int	Numero entero entre -32768 y 32767
float	Valor con numero decimal
double	Aporta el doble de precisión que un flotante
array	También llamado vector, se distinguen entre sí por un número de índice.
string	Cadena de caracteres
long	Numero entero entre -2,147,483,648 y 2.147,483,647

TABLA 3.TIPOS DE DATOS EN ARDUINO (ARTERO, 2013)

La Finalidad de Arduino

La principal finalidad del Arduino en un sistema es facilitar el prototipado, implementación o emulación del control de sistemas interactivos, a nivel doméstico, comercial o móvil, de la misma forma que el PLC controla sistemas de funcionamiento industriales. Con este es posible enviar o recibir informaciones de básicamente cualquier sistema electrónico, como identificar la aproximación de una

persona y variar la intensidad de la luz del ambiente a su llegada. O abrir las ventanas de una oficina según la intensidad de la luz del sol y la temperatura ambiente.

Los campos de actuación para el control de sistemas son inmensos, pudiendo tener aplicaciones en el área de impresión 3D, robótica, ingeniería de transportes, ingeniería agrónoma, musical entre otras. (Pedraza)

Ejemplos

Detector de mentiras

El detector de mentiras utilizara un efecto el cual se conoce como “respuesta galvánica de piel” (resistencia cutánea). Utilizando una entrada analógica se podrá medir la resistencia cutánea. Cuanto menor sea la resistencia cutánea, más se acercara la entrada analógica 0 hacia los 5v, y cuanto mayor sea se acercara a GND o 0v que hará sonar o no un piezozumbador.

Se utiliza un LED multicolor, rojo indicara la mentira, verde indicara la verdad, azul mostrara que el detector de mentiras debe ajustarse mediante un potenciómetro. El potenciómetro se utiliza para ajustar el punto de consigna de la resistencia, y las terminales de mediación.

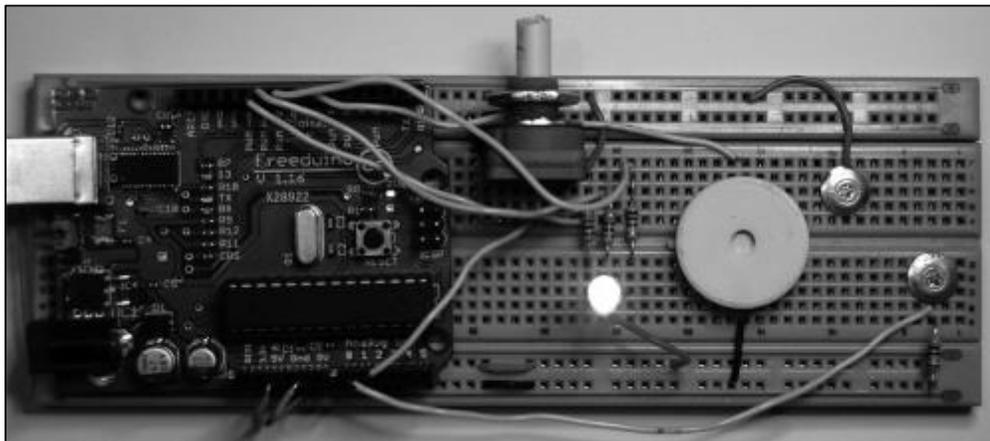


ILUSTRACIÓN 10.DETECTOR DE MENTIRAS (MONK, 2012)

Osciloscopio

Este proyecto lee los valores de la entrada analógica y los envía al ordenador a través de una conexión USB. En lugar de ser recibidos por el Serial Monitor

El circuito tiene dos partes. R1 y R2 son resistencias de alto valor que polarizan la señal que va a la entrada analógica a 2.5 V, haciendo de divisor de tensión 2.5 V condensador C1 permite a la señal de AC (comente alterna) pasar sin ninguna componente de corriente continua (DC).

R3, R4, C2 y C3 proporcionan un voltaje estable de referencia de 2.5 V. Esto es así para que nuestro osciloscopio pueda mostrar tanto señales positivas como negativas. Es decir, un voltaje positivo significará un valor en la entrada analógica de más de 2.5 V y un valor negativo de menos de 2.5 V.

La manera más sencilla de probar el osciloscopio es utilizar una señal que tenemos fácilmente disponible a nuestro alrededor y que es el zumbido de la red eléctrica. La red eléctrica oscila a 50 ó 60 Hz.

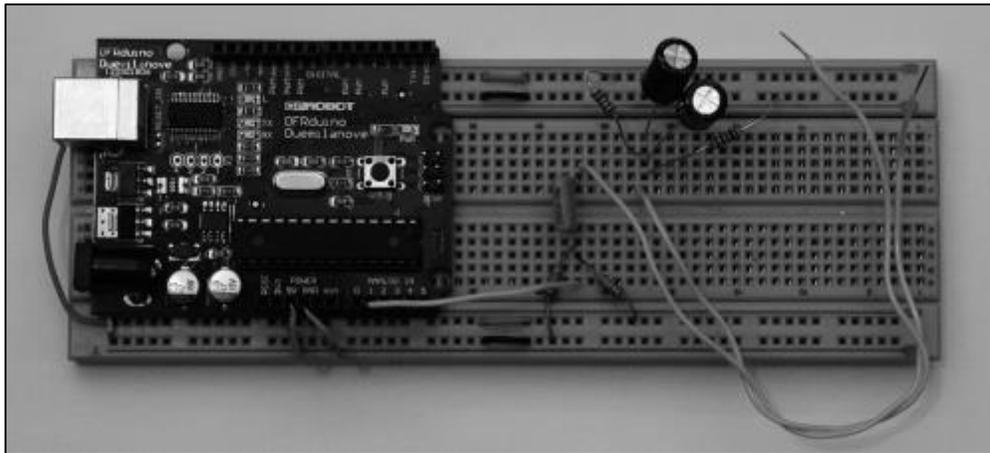


ILUSTRACIÓN 11.OSCILOSCOPIO (MONK, 2012)

Ventilador controlado por ordenador

Para este proyecto se utilizara un ventilador de una PC, al cual se le quiere controlar la velocidad mediante un ordenador.

Se puede controlar la velocidad del ventilador utilizando la salida analógica (PWM) y haciendo que maneje un transistor de potencia para controlar el motor. Puesto que estos ventiladores de ordenador suelen ser de 12 V, se utiliza una fuente de alimentación externa para proporcionar la energía para alimentar el ventilador.

Esencialmente, lo único que necesita es leer un número del 0 al 9 del USB y ejecutar un analogWrite al motorPin de ese valor, multiplicado por 28, para escalarlo hasta un número entre 0 y 252.

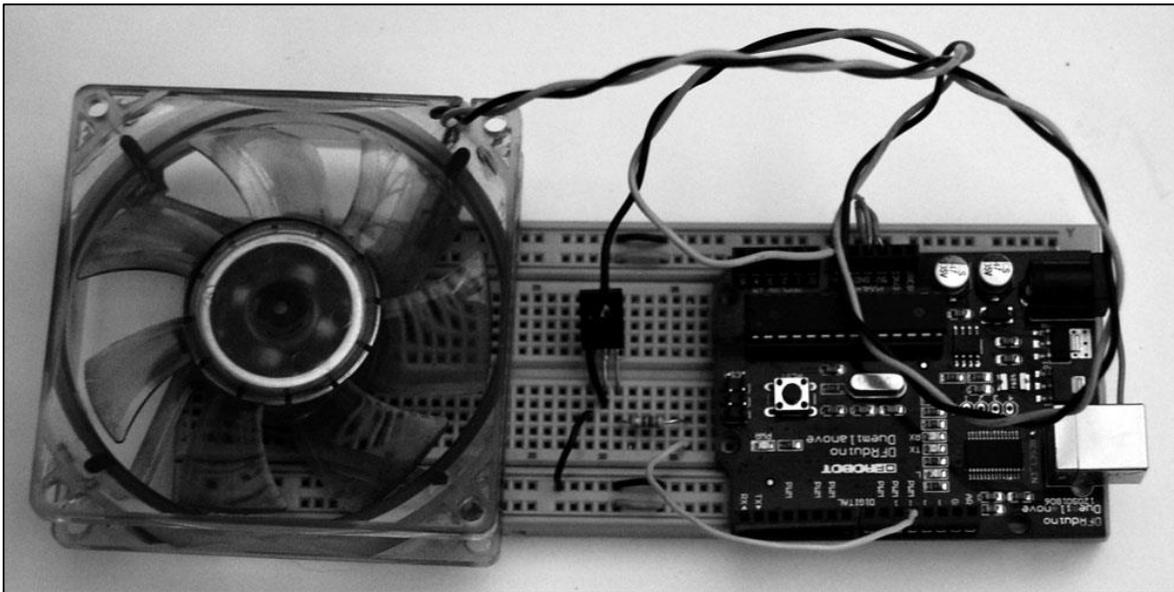


ILUSTRACIÓN 12. VENTILADOR CONTROLADO POR ORDENADOR (MONK, 2012)

Capítulo 4. Prototipo.

En este capítulo se dará a conocer el proceso que conlleva el armado del prototipo, así como es el ensamblaje y las conexiones necesarias para que el prototipo funcione, como también la interfaz de usuario auxiliar para el manejo del mismo.

Elementos Principales

Sensor de Flujo

El sensor de flujo es un dispositivo que al ser instalado en una tubería permite determinar la circulación de un fluido, es decir, el sensor nos va a indicar la ausencia o presencia de flujo, mandando la cantidad que va pasando por el mismo mediante el puerto serie al Arduino, este sensor funciona para agua y líquidos no corrosivos. Su alimentación es mediante la placa Arduino Leonardo ya que el sensor trabaja a 5V.



ILUSTRACIÓN 13.SENSOR DE FLUJO
YF-S201

- Rango 1/30 L por minuto
- Presión máxima: 2.00 Mpa
- Alimentación: 5V-24V DC
- Corriente máxima: 15mA a 5V DC
- rojo (+), negro (-), amarillo (señal)
- Conexión para manguera de 1/2 pulgada

TABLA 4.CARACTERÍSTICAS SENSOR DE FLUJO YF-S201

Shield para 8 Relés

Los relevadores funcionaron como un interruptor, permitiendo o negando el paso de la corriente eléctrica, es necesario utilizar un relevador por electroválvula dejando 5 relés libres por si se desea agregar más elementos. Para su alimentación se requiere una fuente de alimentación externa para PC, ya que su alimentación es de 12V y su amperaje es bastante considerable. Nuestro Arduino va empotrada en la placa de relés ya que esta nos hace la función de un Shield.

	<ul style="list-style-type: none">• 8 relés de 10 Amp• Alimentación 9 a 12 VCD
--	---

ILUSTRACIÓN 14. SHIELD PARA 8 RELÉS DE 10 AMP

TABLA 5. CARACTERÍSTICAS SHIELD PARA 8 RELÉS DE 10 AMP

Electroválvula

Las electroválvulas son dispositivos que responden a los pulsos eléctricos. Gracias a la corriente que circula a través del solenoide es posible abrir o cerrar la válvula controlando, de esta forma, el flujo de fluidos. Su alimentación es de 12V por lo que se requiere una fuente de alimentación externa para PC y alimentar las electroválvulas independientemente cada una con una línea directa de 12V y su tierra física para su mejor funcionamiento

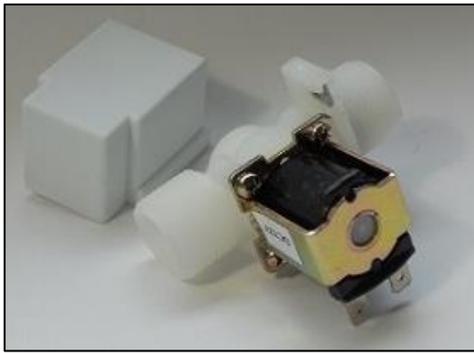


ILUSTRACIÓN
155.ELECTROVÁLVULA

- Válvula solenoide de 3/4" (12V)
- Alimentación 12V
- Material: Plástico
- Ambiente de trabajo: Agua, Gas y Aceite
- Temperatura de trabajo: 0- 100 °C
- Voltaje: AC/DC 6/9/12/24/36110/220V

TABLA 6.CARACTERÍSTICAS ELECTROVÁLVULA

Arduino Leonardo

Arduino Leonardo es el principal elemento ya que sin esta placa, el funcionamiento de los demás elementos no se podrían llevar acabo, en dicha placa se programará las acciones de cada uno de los componentes (ver Anexo 2) en conjunto con la una interfaz de usuario en Java (Netbeans) con una comunicación por el puerto serie COM En donde Leonardo recibirá los datos y empezará a trabajar con ellos.

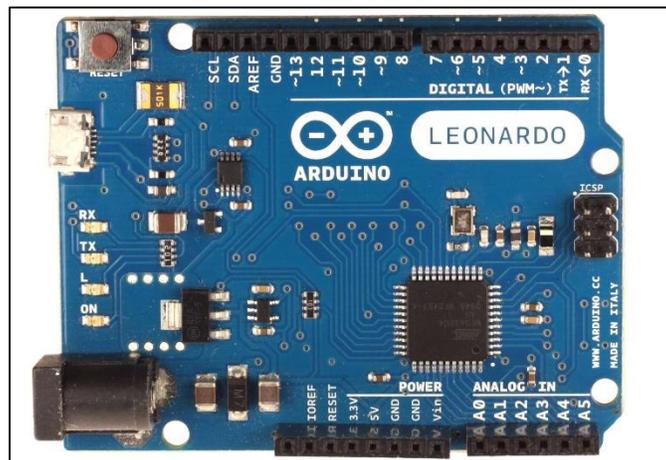


ILUSTRACIÓN 16.ARDUINO LEONARDO

Microcontrolador	ATmega32u4
Voltaje de funcionamiento	5V
Alimentación (recomendada)	7-12V
Voltaje máximo de entrada(no recomendado)	20V
Pines digitales I/O	20 (de los cuales 7 dan salida PWM)
Pines de entrada analógica	12
Corriente DC por I/O Pin	40 mA
Corriente DC para el pin 3.3V	50 mA
Memoria Flash	32 KB
SRAM	3.3 KB
EEPROM	1 KB
Velocidad de reloj	16 MHz

TABLA 7. CARACTERÍSTICAS ARDUINO LEONARDO

Fuente de Poder ATX 300W

La fuente alimentará el Shield para 8 Relés y las electroválvulas proporcionando una línea de 12V independientes a cada elemento del prototipo, la fuente dará un amperaje estable y permitirá que las electroválvulas funcionen adecuadamente y no se pierda tanto el voltaje como el amperaje.



ILUSTRACIÓN 17.FUENTE DE PODER ATX 300W

Conexiones

Tanto para la conexión del Shield de 8 relés como para las electroválvulas fue necesario hacer una conexión mediante los conectores molex macho a la fuente de Poder ATX 300W ya que esta fuente cuenta con conectores molex tipo hembra, cabe señalar que se venden cables molex ya armados los cuales no fueron implementados ya que se era necesario conectar un extremo a la fuente y otro a la electroválvula y a la placa Shield, por lo que fue necesario comprar el conector sin armar como se muestra en la Ilustración 18 y soldar cada parte utilizando solo dos cables corriente y tierra física como se muestra en la Ilustración 19 y 20.



ILUSTRACIÓN 18.CONECTOR MOLEX MACHO SIN ENSAMBLAR

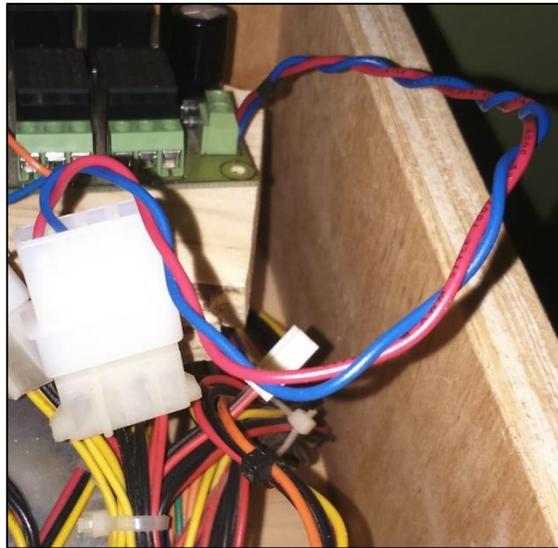


ILUSTRACIÓN 19. CONECTOR MOLEX DE LA FUENTE AL SHIELD DE RELÉS

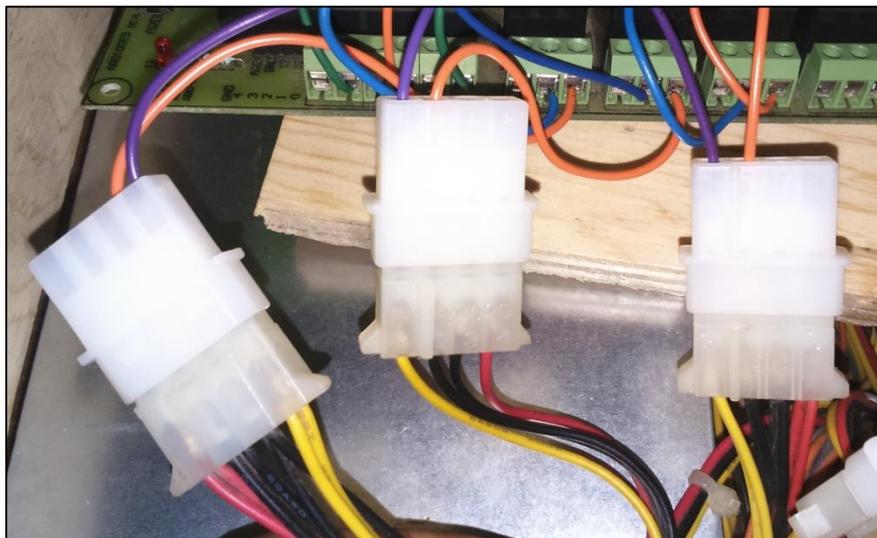


ILUSTRACIÓN 20. CONECTOR MOLEX DE LA FUENTE A LAS TRES ELECTROVÁLVULAS

Para la conexión de las electroválvulas con la fuente de poder también fue necesario unas terminales también conocidas como zapatas Ilustración 21 las cuales se quitan y ponen en los conectores de las electroválvulas esto para poder tener un mejor manejo y poder ensamblar con facilidad el prototipo y sin necesidad de soldarlo todo por completo. Mostrándose como en la Ilustración 22.



ILUSTRACIÓN 21.TERMINALES O ZAPATAS



ILUSTRACIÓN 22.CONEXIÓN DE LAS TERMINALES CON LA ELECTROVÁLVULA

Se repitió el mismo procedimiento hasta que las tres electroválvulas estuvieran conectadas cada uno con su línea de 12V y si tierra física

Para la conexión del sensor del flujo fue necesario soldar tres cables que van de la placa a los sensor como se muestra en la Ilustración.23, ya que la placa Leonardo

va empotrada en el Shield de 8 relés, identificando los pines que ocupaban ese puerto y elaborando los cables que nos permitieran la conexión del sensor con la placa, quedado como se muestra en las Ilustraciones 24 y 25 y 26.

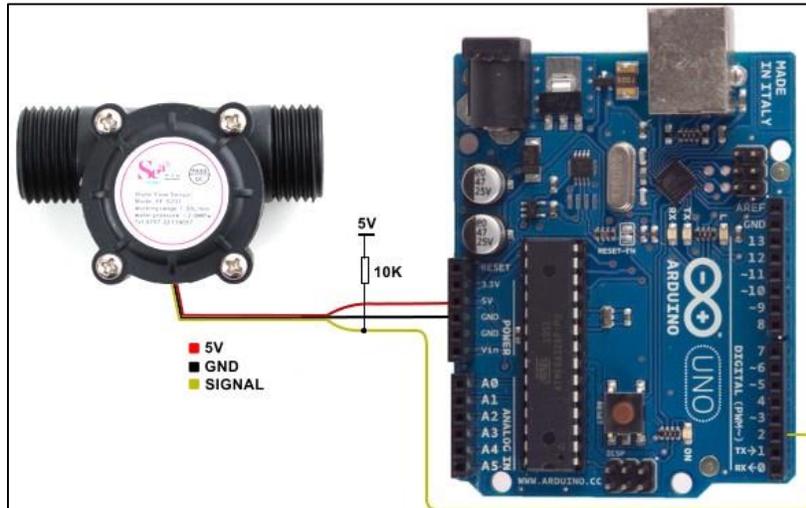


ILUSTRACIÓN 23. CONEXIÓN DEL SENSOR DE FLUJO CON LA PLACA ARDUINO

Como el sensor tiene entradas tipo hembra fue necesario soldar unos headers tipo macho en los cables que vienen de la placa hacia el sensor para poder tener un mejor manejo y no ocasionara un falso contacto a la hora de ejecución.



ILUSTRACIÓN 24. CONEXIÓN CON HEADERS AL SENSOR DE FLUJO

Una vez teniendo identificado los pines que se utilizarán para la recepción de datos Ilustración 25 y la alimentación que se le dará al sensor, se observó que en los pines de corriente y tierra eran conectores headers tipo macho por lo que se necesitó implementar headers tipo hembra y soldarlos para que el sensor estuviera alimentado con los 5V y su GND como se muestra en la Ilustración.26.



ILUSTRACIÓN 25. IDENTIFICACIÓN DE LOS PINES QUE RECIÉN LOS DATOS DEL SENSOR DE FLUJO

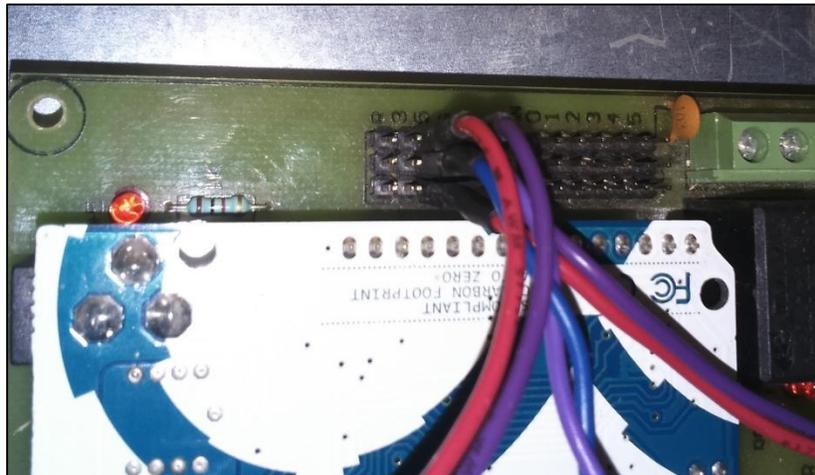


ILUSTRACIÓN 26. IDENTIFICACIÓN DE LOS PINES TIERRA GND Y VOLTAJE 5V

Una vez hechas estas conexiones, el sensor de flujo está listo para mandar datos al Arduino por el puerto serie (COM), ya que se identificaron los pines donde se transmiten los Datos, GND y Voltaje.

Todo el cableado interno Ilustración 27, fue colocado dentro de un contenedor con el fin de proteger el cableado y evitar deterioro y posibles fallas por la exposición de los elementos, así también como una mejor presentación del prototipo.

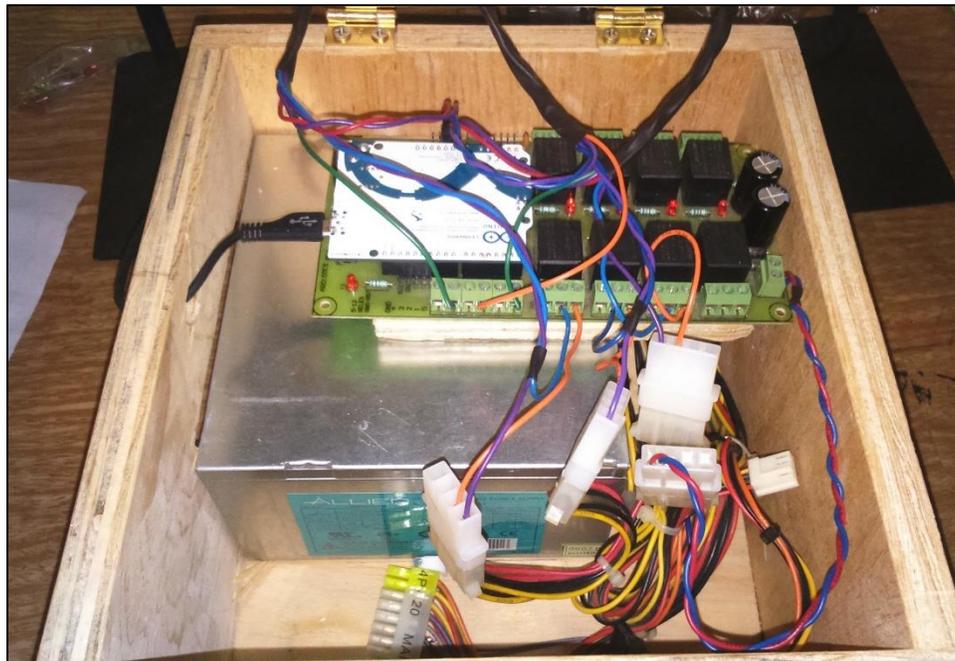


ILUSTRACIÓN 27. CABLEADO INTERNO DEL PROTOTIPO

Ensamblado Final

Una vez teniendo todas las conexiones listas y probadas, se procede a ensamblar todas las piezas que participarán en desarrollo y ejecución del prototipo.

El prototipo se elaboró de esta forma con la finalidad de que fuera fácil de ensamblar, y para que cualquier persona pueda ensamblarlo y transportarlo.

A continuación, se muestra el ensamblado final del prototipo y algunas tomas fotográficas de cómo luce el aparato.

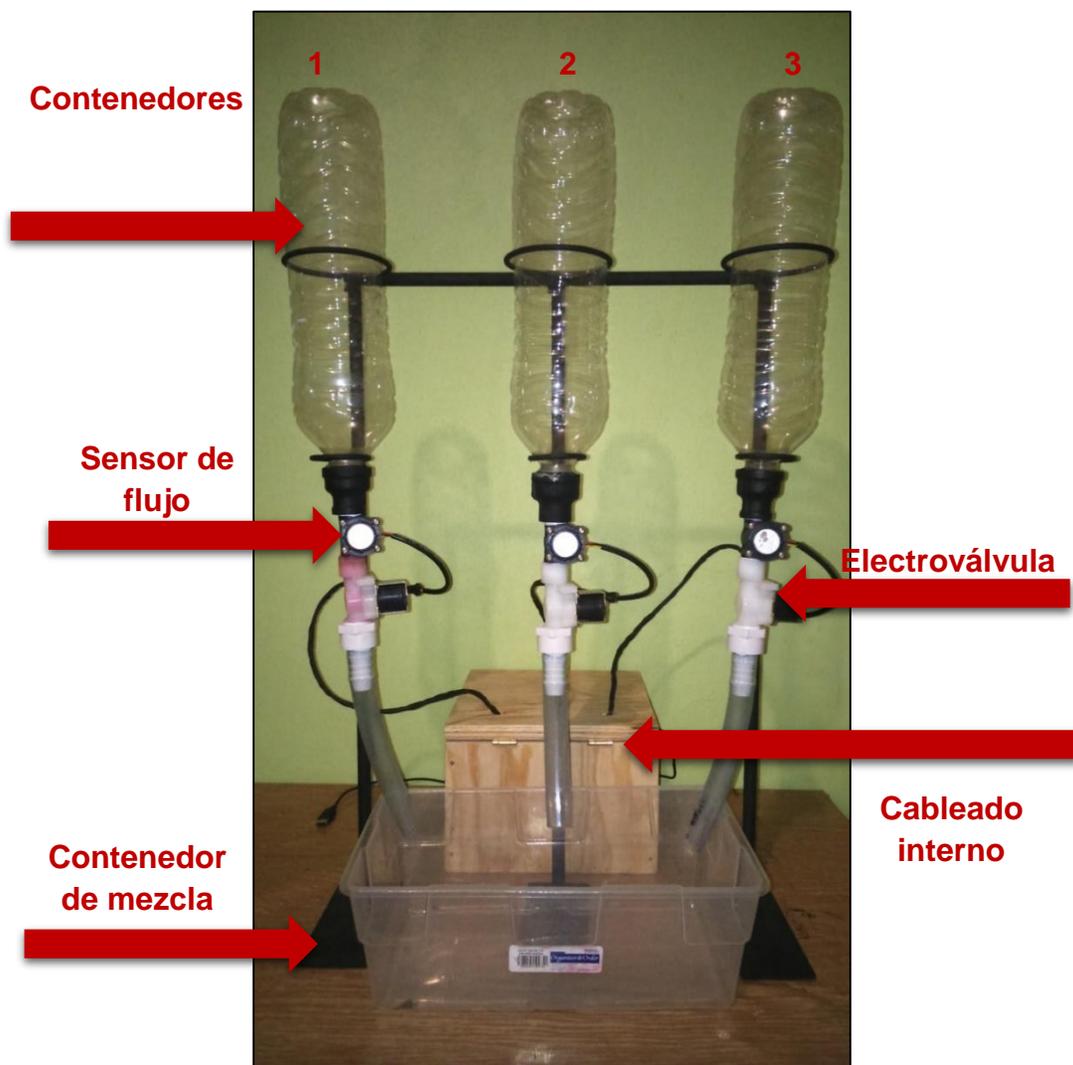


ILUSTRACIÓN 28.VISTA PARTE FRONTAL DEL PROTOTIPO

Una vez llenos los tres tanques y con ayuda de una interfaz de usuario se mandara la cantidad que se desees al Arduino y esta a su vez ejecutara el programa que ponga en marcha todo el sistema y deteniendo el flujo en la cantidad deseada, teniendo así una automatización para la preparación de mezclas de solución para el fertirriego.

En la parte que se encuentra atrás es donde se encuentra todo el cableado protegido como anterior mente se dijo por una caja contenedora donde se efectúa la comunicación serial del Arduino con Java, es aquí donde se encuentra el usuario haciendo uso de la interfaz que se ha realizado para que interactúe dinámicamente con el prototipo ya que dicha interfaz es intuitiva y fácil de usar.



ILUSTRACIÓN 29.VISTA TRASERA DEL PROTOTIPO

Interfaz de Usuario

Con base en la Ingeniería de Software e implementando las Guías de Diseño (ver Anexo 1) se implementó una Interfaz gráfica que servirá como el medio de interacción con el prototipo. Una interfaz de usuario dinámica, intuitiva y fácil de usar para que el usuario sea capaz de hacer uso del mismo y hacer eficiente el proceso de automatización.

En la interfaz se puede mostrar tres tanques de diferente color que hacen referencia a los tanques físicos con los que cuenta el prototipo, estos tres tanques tienen una pequeña animación cuando se pasa el cursor sobre el mismo esto con el fin de no hacer una interfaz aburrida y que no llame la atención.



ILUSTRACIÓN 30. INTERFAZ DE USUARIO DEL PROTOTIPO

La interfaz cuenta con tres Spinner, donde por medio de flechas se selecciona la cantidad de solución que se desea, una por cada tanque de solución, en donde la

cantidad a ingresar es por litros teniendo un inicio en 0.00 litros y un límite de 1.50 litros.

La interfaz cuenta con tres botones importantes: Play, Stop y Reset, los cuales mandarán datos al Arduino para su ejecución (ver Anexo 3).

Fase de pruebas

Fue necesario realizar las pruebas necesarias para verificar si los objetivos fueron alcanzados o no. Se necesitó llenar los tanques y realizar las posibles combinaciones que podrían realizarse para denotar si se encontraba alguna falla en el sistema.

Una vez que se realizaron las pruebas se determinó si el prototipo había cumplido con sus objetivos establecidos.

El prototipo cumple y se optimizó la preparación de soluciones a través de software y hardware de plataforma libre.

Se desarrolló una Interfaz de grafica amigable para el usuario realizando una conexión Java y Arduino integrando así la plataforma de hardware libre (Arduino).

Presupuesto de prototipo

Nombre	Precio	Cantidad	Total
Arduino Leonardo	\$484.88	1	\$484.88
Electroválvula de 3/4" (12v)	\$157.30	3	\$471.90
Sensor de Flujo YF-S201	\$139.15	3	\$417.45
Shield para 8 relés de 10 Amp	\$332.06	1	\$332.06
Fuente de Poder Allied	\$450.00	1	\$450.00
		COSTO TOTAL	\$2,156.29

TABLA 8.ELEMENTOS PRINCIPALES Y COSTOS

Nombre	Precio	Cantidad	Total
Cable #22	\$4.00	16	\$64.00
Resistencia 12 K	\$1.00	4	\$4.00
Termo fit delgado	\$12.00	1	\$12.00
Termo fit grueso	\$15.00	3	\$45.00
Cautín de lápiz	\$65.00	1	\$65.00
Terminales	\$1.50	6	\$9.00
Reducción de 1" a ½ " pulgada	\$20.00	3	\$60.00
Conector molex macho	\$10.00	3	\$30.00
Estructura de metal	\$200.00	1	\$200.00
Conector para manguera ½"	\$8.00	3	\$24.00
Manguera ½" Transparente	\$15.00	1	\$15.00
Recipiente transparente	\$17.50	1	\$17.50
COSTO TOTAL			\$545.50

TABLA 9. OTROS ELEMENTOS Y PRECIOS

Nombre	Precios
Elementos principales	\$2,156.29
Otros elementos	\$545.50
TOTAL	\$2,701.79

TABLA 10. COSTO DE DESARROLLO

Discusiones

- El prototipo puede ser considerado para implementarse con tanques de mayor capacidad e incluso incrementar el número de tanque, solo hay que considerar que el sensor de flujo tiene una rosca de $\frac{1}{2}$ pulgada y se deberá buscar una reducción para que este tenga un funcionamiento adecuado.
- El funcionamiento de las electroválvulas requiere de un amperaje alto para un buen funcionamiento, por lo cual se utilizó una fuente de alimentación externa para PC; de no contar con este elemento, provocara que las electroválvulas no reciban el voltaje necesario para que realice su función correctamente.
- El prototipo puede ser aún más autónomo si se cambia la fuente de alimentación externa por una pila (carro montable) que arroje 12 volts de voltaje pero su amperaje no se caiga al conectar las tres electroválvulas. Así teniendo unos sistemas cada vez más autónomo. Solo debiendo considerar la duración de dicha pila
- Existen varias opciones para establecer un sistema de mezclas de solución. Los más sencillos son fáciles y baratos de implementar mientras que aquellos sistemas automatizados pueden resultar de alto costo oscilando en precios aproximados de \$10,000.00 o más.
- La automatización de todo el sistema implica costos superiores a \$1,000.00 y puede llegar a \$3,000.00.

Conclusiones

- La construcción del prototipo es de bajo costo, ya que se implementó con software libre y hardware de plataforma abierta.
- Ya que el prototipo funciona en base de software libre y hardware de plataforma abierta se le pueden agregar más tanques para preparar diversas mezclas de solución, es decir, el prototipo es escalable.
- Dado que la interfaz gráfica de usuario, es de fácil uso e intuitiva, su interacción por parte del usuario no requiere de conocimientos detallados.
- El prototipo cumple con el objetivo deseado, la automatización y optimización de la mezcla de soluciones para la fertirrigación, así como la reducción de tiempo de elaboración del mismo.
- Se puede aplicar en diversos lugares, donde se requieran de mezclas de soluciones, como por ejemplo en la ingeniería agrónoma.
- El prototipo puede elaborar mezclas de soluciones tanto químicas como orgánicas.

Bibliografía

- (s.f.). Recuperado el 23 de Abril de 2015, de ELECTRONICA ESTUDIO.CO, Ingeniería Electrónica y Proyectos PICmicro: <http://www.electronicaestudio.com/sensores.htm>
- (s.f.). Recuperado el 16 de Abril de 2015, de Asociación Nacional de Fabricantes de Fertilizantes : <http://www.anffe.com/noticias/2008/2008-06-02%20La%20importancia%20de%20los%20fertilizantes%20en%20una%20agricultura%20actual%20productiva%20y%20sostenible/LA%20IMPORTANCIA%20DE%20LOS%20FERTILIZANTES.pdf>
- (s.f.). Recuperado el 24 de Abril de 2015, de Arduino: <http://www.arduino.cc/>
- (25 de Marzo de 2008). Obtenido de Universidad de Oviedo Área de Ingeniería de Sistemas y Automática: http://isa.uniovi.es/docencia/iea/teoria/plc_resumen.pdf
- Arduino. (s.f.). *Arduino*. Recuperado el 24 de Abril de 2015, de <http://www.arduino.cc/>
- Artero, Ó. T. (2013). *Arduino Curso práctico de formación*. México: Alfaomega.
- Atlantic International University. (s.f.). Obtenido de <http://cursos.aiu.edu/Automatizacion%20y%20Control/PDF/Tema%205.pdf>
- BEREA, P. (28 de Mayo de 2013). (A. I. INFORMATIVA, Editor) Recuperado el 2015 de Marzo de 12, de Instituto Tecnológico y de Estudios Superiores de Monterrey: <http://www.itesm.mx/wps/wcm/connect/snc/portal+informativo/por+tema/investigacion/sistemaferirrigacionremoto28may13>
- Chacra, R. (s.f.). Recuperado el 12 de Marzo de 2015, de Agritotal.com: <http://www.agritotal.com/0/vnc/nota.vnc?id=547>
- Collins, K. (29 de Abril de 2015). *Mechatronickai És Autotechnikai Intézet*. Obtenido de http://siva.bgk.uni-obuda.hu/jegyzetek/Mechatronikai_alapismeretek/English_Mechatr/PLC_Control/plcprogramming.pdf
- Cruz Bautista, F. (Ed.). (2007). Recuperado el 17 de Abril de 2015, de Automatización de un sistema de Fertirrigación por goteo. Tesis de Maestría. Colegio de Postgraduados. México:

http://www.biblio.colpos.mx:8080/xmlui/bitstream/handle/10521/1249/Cruz_Bautista_F_MC_Hidrociencias_2007.pdf?sequen

Facultad de Ciencias, Tecnologías Virtual, Electrónica II. (Agosto de 2015).

Obtenido de <http://www.herrera.unt.edu.ar/eiipc/material/apuntes/Automatas%20Programables.pdf>

Ing. Mario Alberto Perez, I. A. (2008). *Universidad Nacional de San Juan Facultad de Ingeniería.* Obtenido de

<http://dea.unsj.edu.ar/control1b/teoria/unidad1y2.pdf>

J.E. Mohd Salih, ,. A. (13 de Diciembre de 2012). *ScienceDirect.* Obtenido de <http://www.sciencedirect.com/science/article/pii/S2212670812001844>

Josep Balcells, J. L. (2000). *Autómatas Programables.* Barcelona: MARCOMBO, S.A.

Kendall, K. E. (2011). *Análisis y Diseño de Sistemas.* Pearson Educación.

Monk, S. (2012). *30 Proyectos con Arduino .* España: Estribor.

Onysus Create Wonder . (s.f.). Obtenido de <http://www.onysus.com/interfaz-natural-de-usuario-nui-en-una-cascara-de-nuez/>

Pedraza, J. G. (s.f.). *Fundamentos de Arduino.*

Reinaldo Rey García, L. R. (Julio de Julio de 2007). *Red Iberoamericana de Riegos.* Obtenido de

http://ceer.isa.utl.pt/cyted/2007/ecuador2007/13_Rodriguez.pdf CYTED:

Rueda, D. E. (Ed.). (s.f.). (U. Facultad de química, Productor) Recuperado el 2015 de Abril de 2015, de Evolución de la industria de fertilizantes en México: http://www.quimica.unam.mx/IMG/doc/Resumen_platica_E_Bazua.doc

Saldaña, T. M. (Ed.). (s.f.). Recuperado el 16 de Abril de 2015, de Historia De La Agricultura En México: <http://www.bvsde.ops-oms.org/bvsacd/eco/003106/03106-03-A.pdf>

Toni Granollers i Saltiveri, J. L. (2011). *Diseño de sistemas interactivos centrados en el usuario.* Editorial UOC.

Anexo 1

Interfaces Hombre-Máquina

Usabilidad y Guías de Diseño

Se analizan los conceptos que nos sirven como referencia al momento de diseñar e implementar interfaces que nos sirven como medio de interacción entre el humano y el computador.

Usabilidad

Jakob Nielsen, pionero en la difusión de la usabilidad, indica que la usabilidad es un término multidimensional. Señala que un sistema usable debe de poseer los siguientes atributos: capacidad de aprendizaje, eficiencia en el uso, facilidad de memorizar, tolerante a errores y subjetivamente satisfactorio. (Toni Granollers i Saltiveri, 2011)

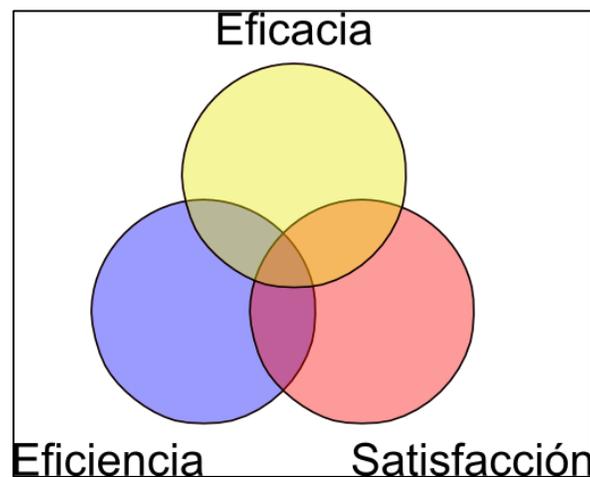


ILUSTRACIÓN 31.USABILIDAD

La usabilidad hace referencia a la rapidez y facilidad con la que las personas llevan a cabo sus tareas propias mediante el uso del producto con el que están trabajando. Esta idea descansa en cuatro puntos

- Una aproximación al usuario: usabilidad significa centrarse en los usuarios. Para desarrollar un producto usable se tiene que conocer, entender y trabajar con las personas que representan a los usuarios actuales o potenciales del producto.
- Un amplio conocimiento del contexto de uso: las personas utilizan los productos para incrementar su propia productividad. Un producto se considera fácil de aprender y de usar en términos del tiempo que toma el usuario para llevar a cabo su objetivo, el número de pasos que debe realizar para esto y el éxito que tiene en predecir la acción apropiada para llevar a cabo. Para desarrollar productos usables, hay que entender los objetivos del usuario, es necesario conocer los trabajos y tareas del usuario que el producto automatiza o modifica.
- El producto debe satisfacer las necesidades del usuario y adaptarse a sus modelos mentales: los usuarios son gente muy diversa y ocupada, que intenta llevar a cabo una tarea. Se va a relacionar usabilidad con productividad y calidad. El hardware y el software son las herramientas que ayudan a la gente ocupada a realizar trabajos ya disfrutar de su ocio.
- Son los usuarios, y no los diseñadores o desarrolladores, los que determina cuando un producto es fácil de usar.

Implementación de las Buenas Prácticas de la HCI

Es importante reconocer que la interfaz es el sistema para la mayoría de los usuarios. Sin importar qué tan bien o mal diseñada esté, representa al sistema y refleja su competencia como analista de sistemas. Una interfaz bien diseñada mejora el ajuste entre la tarea, la tecnología y el usuario.

Su objetivo debe ser diseñar interfaces que ayuden a los usuarios y las empresas a introducir la información al sistema y obtener de éste la información que necesitan, para lo cual hay que considerar los siguientes objetivos: (Kendall, 2011).

- Hacer que la interfaz de usuario corresponda con la tarea.

- Hacer la interfaz de usuario eficiente.
- Proveer una retroalimentación apropiada a los usuarios.
- Generar consultas que se puedan utilizar.
- Mejorar la productividad de los usuarios de computadora.

Lineamientos para el Diseño del Diálogo

El diálogo es la comunicación entre la computadora y una persona. Un diálogo bien diseñado facilita a las personas el proceso de usar una computadora y reduce su frustración con el sistema de cómputo. Hay varios puntos clave para diseñar un buen diálogo: (Kendall, 2011).

1. Una comunicación significativa.
2. Una mínima acción por parte del usuario.
3. Operación y consistencia estándar.

1.- Comunicación significativa: El sistema debe presentar con claridad la información al usuario. Esto significa tener un título apropiado para cada pantalla, minimizar el uso de las abreviaturas y proveer una retroalimentación clara al usuario. Hay que proveer instrucciones para el usuario en relación con los detalles, como las asignaciones disponibles para las teclas de función. En una interfaz gráfica de usuario, el cursor puede cambiar de forma dependiendo del trabajo que se esté realizando.

Los usuarios con menos habilidad para usar la computadora o los que realizan sus tareas con una computadora requieren de mayor comunicación. La información de la línea de estado para las pantallas de GUI es otra forma de proveer instrucciones para los usuarios.

Muchas GUI incorporan la ayuda mediante cuadros de información sobre las herramientas, donde se muestra un pequeño mensaje de ayuda para identificar la función de un botón de comando cuando se coloca el cursor encima de éste. El otro lado de la comunicación es que la computadora debe comprender lo que introdujo

el usuario. Por ende, hay que editar todos los datos que se introduzcan en la pantalla para que sean válidos.

2.- Mínima acción por parte del usuario: La acción de teclear es a menudo la parte más lenta de un sistema de cómputo, por lo que un buen diálogo minimizará el número de pulsaciones de tecla requeridos. Podemos lograr esto de varias formas distintas:

- Teclear códigos en las pantallas de entrada, como los códigos de aeropuertos al hacer una reservación de un vuelo, en vez de palabras completas.
- Introducir sólo datos que no estén ya almacenados en los archivos.
- Proveer los caracteres de edición (por ejemplo, las barras diagonales como separadores de campos de fecha).
- Usar valores predeterminados para los campos en las pantallas de entrada.
- Diseñar un programa de consulta (o de modificación, o eliminación) de manera que el usuario necesite introducir sólo los primeros caracteres de un nombre o la descripción de un artículo.
- Proveer pulsaciones de tecla para seleccionar las opciones de los menús desplegables.
- Usar botones de opción y listas desplegables para controlar las pantallas de nuevas páginas Web o modificar formularios Web.
- Proveer control del cursor para los formularios Web y otras pantallas, de manera que el cursor se desplace al siguiente campo cuando se haya introducido el número correcto de caracteres.

3.- Operación y consistencia estándar: El sistema debe ser consistente a través de todas sus pantallas y todos sus mecanismos para operarlas en distintas aplicaciones. La consistencia facilita a los usuarios el proceso de aprender a usar las nuevas porciones del sistema una vez que estén familiarizados con un componente. Podemos lograr la consistencia mediante los siguientes puntos:

- Ubicar los títulos, la fecha, hora y los mensajes del operador y de retroalimentación en los mismos lugares en todas las pantallas.
- Salir de cada programa mediante el uso de la misma clave u opción del menú.
- Cancelar una transacción de manera consistente, como usar la tecla **esc**.
- Obtener ayuda de una manera estandarizada, como usar una tecla de función.
- Estandarizar los colores utilizados para todas las pantallas.
- Estandarizar el uso de iconos para las operaciones similares al usar una interfaz gráfica de usuario.
- Usar una terminología consistente en una pantalla de visualización.
- Proveer una manera consistente de navegar por el diálogo.

Técnicas de Interacción

Es importante señalar las formas con las cuales los usuarios pueden interactuar con el sistema. Una técnica de interacción no es más que la fusión entre la entrada y la salida, consistiendo de todos los elementos de software y hardware, que permite realizar y completar una tarea.

Línea de Comando

Las interfaces basadas en líneas de comando, o CLI (Command Line Interface) por sus siglas en inglés, fueron de las primeras técnicas desarrolladas para poder interactuar con los equipos informáticos, donde los usuarios digitan una serie de instrucciones y reciben algún tipo de respuesta por parte del sistema.

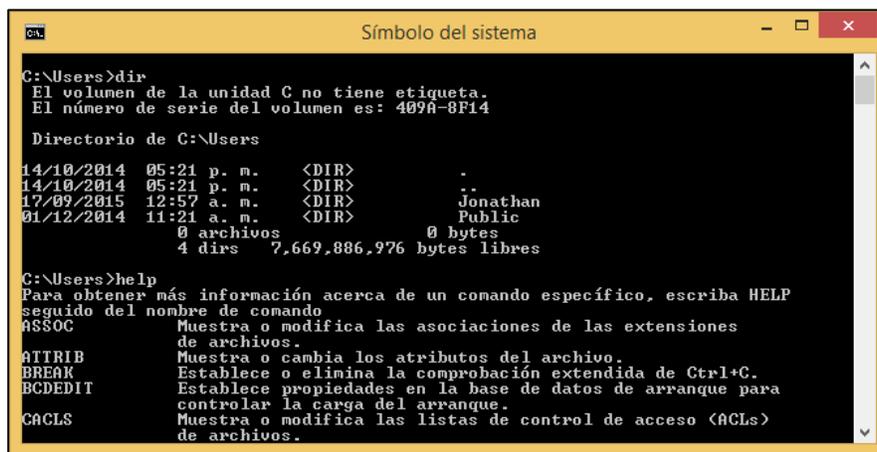
Una interfaz de lenguaje de comandos permite al usuario controlar la aplicación mediante una serie de pulsaciones de tecla, comandos, frases o alguna secuencia de estos tres métodos. Las sintaxis simples de los lenguajes de comandos se consideran cercanas al lenguaje natural. (Kendall, 2011)

El lenguaje de comandos no tiene un significado inherente para el usuario; esto lo diferencia de las demás interfaces. Los lenguajes de comandos manipulan la

computadora como una herramienta al permitir que el usuario controle el diálogo. El lenguaje de comandos ofrece al usuario más flexibilidad y control en general.

Cuando el usuario emplea el lenguaje de comandos, el sistema ejecuta el comando de inmediato; así, el usuario puede darle otro comando.

Los lenguajes de comandos requieren la memorización de reglas de sintaxis, lo que puede constituir un obstáculo para los usuarios inexpertos. Los usuarios expertos tienden a preferir los lenguajes de comandos, posiblemente debido a que se llevan a cabo con mucha más rapidez.



```
C:\Users>dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 409A-8F14

Directorio de C:\Users

14/10/2014 05:21 p. m. <DIR> -
14/10/2014 05:21 p. m. <DIR> ..
17/09/2015 12:57 a. m. <DIR> Jonathan
01/12/2014 11:21 a. m. <DIR> Public
0 archivos 0 bytes
4 dirs 7,669,886,976 bytes libres

C:\Users>help
Para obtener más información acerca de un comando específico, escriba HELP
seguido del nombre de comando
ASSOC Muestra o modifica las asociaciones de las extensiones
de archivos.
ATTRIB Muestra o cambia los atributos del archivo.
BREAK Establece o elimina la comprobación extendida de Ctrl+C.
BCDEDIT Establece propiedades en la base de datos de arranque para
controlar la carga del arranque.
CACLS Muestra o modifica las listas de control de acceso (ACLs)
de archivos.
```

ILUSTRACIÓN 32. INTERFAZ DE LÍNEA DE COMANDOS

Interfaz Gráfica de Usuario

Las GUI acompañadas de la invención del mouse empezaron a hacer uso de ventanas, iconos, menús y punteros, forma de interacción que en la actualidad sigue usándose en la mayoría de los sistemas. (Kendall, 2011)

La interfaz gráfica tiene como objetivo el presentar la información y opciones disponibles de una manera visual y comprensible al usuario permitiéndole navegar entre las diferentes opciones que puede ofrecer una aplicación invitando al descubrimiento de características y posibilidades, donde a diferencia de las interfaces basada en líneas de comando el usuario no se ve en la necesidad de recordar comandos y parámetros.

La clave para las interfaces gráficas de usuario (GUI) es la retroalimentación constante en la realización de las tareas que proveen a los usuarios. La retroalimentación continua sobre el objeto manipulado significa que se pueden cambiar o invertir las operaciones con rapidez, sin incurrir en mensajes de error.

La creación de GUI impone un reto, pues hay que inventar un modelo apropiado de la realidad o un modelo conceptual aceptable de la representación. La mayoría de los usuarios desconocidos para el desarrollador, por lo que el diseño debe estar bien definido. La elección de iconos, lenguaje, etc., se convierte en todo un conjunto completo de decisiones y suposiciones sobre los tipos de usuarios que espera. El diseñador también debe adherirse a las convenciones que los usuarios esperan.

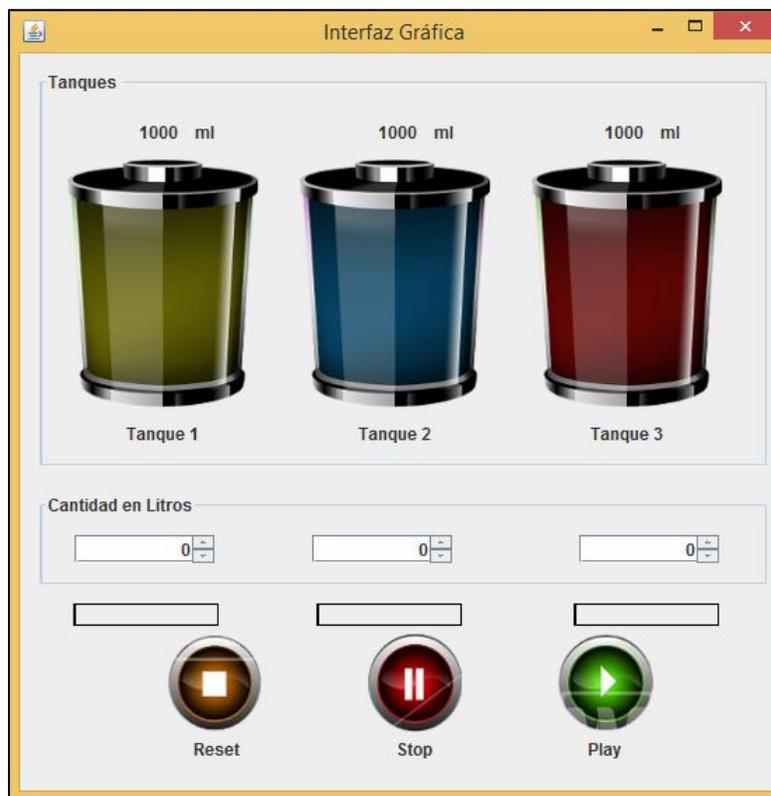


ILUSTRACIÓN 33.INTERFAZ GRÁFICA DE USUARIO (GUI)

Interfaz de Usuario Natural

Las interfaces de lenguaje natural son tal vez el sueño y el ideal de los usuarios sin experiencia, ya que no requieren tener habilidades especiales y pueden interactuar con la computadora mediante el uso de lenguaje natural. (Kendall, 2011)

Las ambigüedades naturales de algunos idiomas, como el inglés y en menor medida el español, producen arduos problemas de programación. Los intentos de crear interfaces en lenguaje natural para ciertas aplicaciones en las que cualquier otro tipo de interfaz no sería viable (por ejemplo, en el caso de un usuario discapacitado) están teniendo algo de éxito; sin embargo, comúnmente estas interfaces son costosas. Hasta ahora, los problemas de implementación y la extraordinaria demanda de recursos de cómputo han mantenido el uso de las interfaces de lenguaje natural al mínimo. Pero la demanda existe, por lo que muchos programadores e investigadores están trabajando con diligencia en ellas. Es un área de crecimiento, por lo que amerita mantenerse al tanto de ella.

Una de las primeras aproximaciones que se le está dando a este tipo de interfaces es la del dispositivo de juegos Xbox Kinect, el cual por medio de cámaras y sensores convierte a la propia persona en el control del dispositivo manipulando totalmente las acciones del sistema con su cuerpo y voz.



ILUSTRACIÓN 34. INTERFAZ DE USUARIO NATURAL (ONYSUS CREATE WONDER , S.F.)

Anexo 2

Código en Arduino

```
// Defino los pines de los sensores de flujo
#define PIN_SENSOR_1 4
#define PIN_SENSOR_2 2
#define PIN_SENSOR_3 0

// Defino los pines de los relevadores
int RB0 = 5;
int RB1 = 6;
int RB2 = 7;
int RB3 = 8;
int RB4 = 9;
int RB5 = 10;
int RB6 = 11;
int RB7 = 12;
int Pin13 = 13; //este es de chocolate

String cadena = ""; // Concatena la Cadena
String cadenafinal;
char caracter; // Recibe los caracteres del puerto Serial
String cadLitros1=""; //Primer valor del estanque 1
String cadLitros2=""; //Segundo valor del estanque 2
String cadLitros3=""; //Tercer valor del estanque 3
int arrobaPosition; // La posicion de la siguiente arroba "@"
en el String
int commaPosition; // La posicion de la siguiente coma "," en
el String
float Litros1=0; //Se definen los Litros
float Litros2=0;
float Litros3=0;
float Limite1=0; //Limite de los Litros
float Limite2=0;
float Limite3=0;

// Numero de pulsos totales por los sensores de flujo
volatile uint16_t Pulsos1 = 0;
volatile uint16_t Pulsos2 = 0;
volatile uint16_t Pulsos3 = 0;

// Ultimo estado del pin
volatile uint8_t Ultimo_Estado_Pin_1;
volatile uint8_t Ultimo_Estado_Pin_2;
```

```
volatile uint8_t Ultimo_Estado_Pin_3;

// usted puede tratar de mantener el tiempo de cuánto tiempo
pasa entre Pulsos
volatile uint32_t Timer1 = 0;
volatile uint32_t Timer2 = 0;
volatile uint32_t Timer3 = 0;

// Y usarlo para calcular la Velocidad de flujo
volatile float Velocidad_Flujo1;
volatile float Velocidad_Flujo2;
volatile float Velocidad_Flujo3;

// La interrupción se llama una vez cada miliSegundo, busca
cualquier Pulso del sensor!
SIGNAL(TIMER0_COMPA_vect) {
  uint8_t x1 = digitalRead(PIN_SENSOR_1);
  uint8_t x2 = digitalRead(PIN_SENSOR_2);
  uint8_t x3 = digitalRead(PIN_SENSOR_3);
  // Lecturas del sensor 1
  if (x1 == Ultimo_Estado_Pin_1) {
    Timer1++;
    //return; // nothing changed!
  }
  else {
    if (x1 == HIGH) {
      //low to high transition!
      Pulsos1++;
    }

    Ultimo_Estado_Pin_1 = x1;
    Velocidad_Flujo1 = 1000.0;
    Velocidad_Flujo1 /= Timer1; // in hertz
    Timer1 = 0;
  }
  // Lecturas del sensor 2
  if (x2 == Ultimo_Estado_Pin_2) {
    Timer2++;
    //return; // nothing changed!
  }
  else {
    if (x2 == HIGH) {
      //low to high transition!
      Pulsos2++;
    }

    Ultimo_Estado_Pin_2 = x2;
```

```
Velocidad_Flujo2 = 1000.0;
Velocidad_Flujo2 /= Timer2; // in hertz
Timer2 = 0;
}
// Lecturas del sensor 3
if (x3 == Ultimo_Estado_Pin_3) {
    Timer3++;
    //return; // nothing changed!
}
else {
    if (x3 == HIGH) {
        //low to high transition!
        Pulsos3++;
    }

    Ultimo_Estado_Pin_3 = x3;
    Velocidad_Flujo3 = 1000.0;
    Velocidad_Flujo3 /= Timer3; // in hertz
    Timer3 = 0;
}
}

void useInterrupt(boolean v) {
    if (v) {
        //Timer0 ya se utiliza para millis () - sólo tendremos
que interrumpir en algún lugar
        //En el medio y llamar a la "comparación A" función
anterior
        OCROA = 0xAF;
        TIMSK0 |= _BV(OCIE0A);
    }
    else {
        // No llamar a la función COMPA interrupción más
        TIMSK0 &= ~_BV(OCIE0A);
    }
}

void setup() {

    // PARA LOS RELES
    pinMode(RB0, OUTPUT);
    pinMode(RB1, OUTPUT);
    pinMode(RB2, OUTPUT);
    pinMode(RB3, OUTPUT);
    pinMode(RB4, OUTPUT);
    pinMode(RB5, OUTPUT);
    pinMode(RB6, OUTPUT);
}
```

```
pinMode(RB7, OUTPUT);

Serial.begin(9600);

pinMode(PIN_SENSOR_1, INPUT);
pinMode(PIN_SENSOR_2, INPUT);
pinMode(PIN_SENSOR_3, INPUT);

digitalWrite(PIN_SENSOR_1, HIGH);
digitalWrite(PIN_SENSOR_2, HIGH);
digitalWrite(PIN_SENSOR_3, HIGH);

Ultimo_Estado_Pin_1 = digitalRead(PIN_SENSOR_1);
Ultimo_Estado_Pin_2 = digitalRead(PIN_SENSOR_2);
Ultimo_Estado_Pin_3 = digitalRead(PIN_SENSOR_3);

useInterrupt(true);
}
//void(*resetFunc)(void)=0; //Segun Reseta arduino

void loop() {
  //resetFunc();
  if(Serial.available() > 0)
  {
    caracter = char(Serial.read());
    cadena.concat(caracter);
  }
  if(cadena.endsWith("\n"))
  {
    int indexn = cadena.indexOf("\n");
    cadenafinal=cadena.substring(0, indexn);
  }
  //Serial.print("La cadena concatenada ");
  Serial.println(cadenafinal); // show the source string
  //Separamos los valores de la cadena
  arrobaPosition = cadenafinal.indexOf('@');
  if(arrobaPosition != -1){
    //Se exxtrae el pimer valor antes del "@"
    cadLitros1 = cadenafinal.substring(0, arrobaPosition);
    commaPosition = cadenafinal.indexOf(',');
    cadLitros2 = cadenafinal.substring(arrobaPosition+1,
    commaPosition);
    //Se extrae el tercer valor depues de la "," hasta fin de
    cadena
    cadLitros3 = cadenafinal.substring(commaPosition+1,
    cadenafinal.length());
    //Se convierte la cadena en valor en numero flotante
```

```
    Limite1 = cadLitros1.toFloat();
    //Serial.print("Litros1:"); Serial.println(Limite1);
    Limite2 = cadLitros2.toFloat();
    //Serial.print("Litros2:"); Serial.println(Limite2);
    Limite3 = cadLitros3.toFloat();
    //Serial.print("Litros3:"); Serial.println(Limite3);
    Serial.println("-----");
}
else{ // here after the last comma is found
    if(cadenafinal.length() > 0)
        Serial.println(cadenafinal); // if there is text after
the last comma,
}

Litros1 = Pulsos1;
Litros1 /= 7.5;
Litros1 /= 60.0;

Litros2 = Pulsos2;
Litros2 /= 7.5;
Litros2 /= 60.0;

Litros3 = Pulsos3;
Litros3 /= 7.5;
Litros3 /= 60.0;

//La electrovalvula se mantiene abierta con "LOW" y cerrada
con "HIGH"
digitalWrite(RB0, HIGH); //Se mantienen las electrovalvulas
cerradas
digitalWrite(RB1, HIGH);
digitalWrite(RB2, HIGH);

//Convinacion 000 Funciona
if(Limite1<=0 && Limite2<=0 && Limite3<=0){
Serial.println("Convinacion 000");
digitalWrite(RB0, HIGH);
digitalWrite(RB1, HIGH);
digitalWrite(RB2, HIGH);
}

//Convinacion 001 Funcioona
if(Limite1<=0 && Limite2<=0 && Limite3>0){
Serial.println("Convinacion 001");
digitalWrite(RB2, LOW);
if (Litros3 >= Limite3){ digitalWrite(RB2, HIGH); }
}

//Convinacion 010 Funciona
```

```
if(Limite1<=0 && Limite2>0 && Limite3<=0){
Serial.println("Convinacion 010");
digitalWrite(RB1, LOW);
if (Litros2 >= Limite2){ digitalWrite(RB1, HIGH); }
}

//Convinacion 011 Funciona
if(Limite1<=0 && Limite2>0 && Limite3>0){
Serial.println("Convinacion 011");
digitalWrite(RB1, LOW);
digitalWrite(RB2, LOW);
if (Litros2 >= Limite2){ digitalWrite(RB1, HIGH); }
if (Litros3 >= Limite3){ digitalWrite(RB2, HIGH); }
}

//Convinacion 100 Funciona
if(Limite1>0 && Limite2<=0 && Limite3<=0){
Serial.println("Convinacion 100");
digitalWrite(RB0, LOW);
if (Litros1 >= Limite1){ digitalWrite(RB0, HIGH); }
}

//Convinacion 101 Funciona
if(Limite1>0 && Limite2<=0 && Limite3>0){
Serial.println("Convinacion 101");
digitalWrite(RB0, LOW);
digitalWrite(RB2, LOW);
if (Litros1 >= Limite1){ digitalWrite(RB0, HIGH); }
if (Litros3 >= Limite3){ digitalWrite(RB2, HIGH); }
}

//Convinacion 110 Funciona
if(Limite1>0 && Limite2>0 && Limite3<=0){
Serial.println("Convinacion 110");
digitalWrite(RB0, LOW);
digitalWrite(RB1, LOW);
if (Litros1 >= Limite1){ digitalWrite(RB0, HIGH); }
if (Litros2 >= Limite2){ digitalWrite(RB1, HIGH); }
}

//Convinacion 111 Funciona
if(Limite1>0 && Limite2>0 && Limite3>0){
Serial.println("Convinacion 111");
digitalWrite(RB0, LOW);
digitalWrite(RB1, LOW);
digitalWrite(RB2, LOW);
if (Litros1 >= Limite1){ digitalWrite(RB0, HIGH); }
if (Litros2 >= Limite2){ digitalWrite(RB1, HIGH); }
if (Litros3 >= Limite3){ digitalWrite(RB2, HIGH); }
}

Serial.println("-----");
```

```
Serial.println("Litros: ");
Serial.print("1: "); Serial.println(Litros1);
Serial.print("2: "); Serial.println(Litros2);
Serial.print("3: "); Serial.println(Litros3);
Serial.println("-----");

delay(100);
}

void reseteo(){
  //delay(2000);
  Timer1=0;
  Timer2=0;
  Timer3=0;
  Pulsos1=0;
  Pulsos2=0;
  Pulsos3=0;
  Litros1=0;
  Litros2=0;
  Litros3=0;
  Velocidad_Flujo1=0;
  Velocidad_Flujo2=0;
  Velocidad_Flujo3=0;
  Ultimo_Estado_Pin_1=0;
  Ultimo_Estado_Pin_2=0;
  Ultimo_Estado_Pin_3=0;

  //cadena = "";
  //adenafinal="";
  caracter;
  //cadLitros1="";
  //cadLitros2="";
  //cadLitros3="";
  //arrobaPosition=0;
  //commaPosition=0;
  Limite1=0;
  Limite2=0;
  Limite3=0;

  digitalWrite(RB0, HIGH);
  digitalWrite(RB1, HIGH);
  digitalWrite(RB2, HIGH);
}
```

Anexo 3

Código en Java Envió de Valores

```
float valor1=0 ; //con valor uno funciona
float valor2=0 ;
float valor3=0 ;
String cadena;

//valor = ((int) jSpinner1.getValue()+ (int)
jSpinner2.getValue()+ (int) jSpinner3.getValue());
    valor1 = (float) jSTanque1.getValue();
    valor2 = (float) jSTanque2.getValue();
    valor3 = (float) jSTanque3.getValue();

    //Concatena los valores de los Spinners
cadena = ""+valor1+"@"+valor2+", "+valor3+"\r\n";

    //System.out.println("valor-->" + valor);
    //System.out.println("cadena-->" + cadena);

    try {
        CommPortIdentifier portId = null;
        try {
            portId =
CommPortIdentifier.getPortIdentifier("COM4");
        } catch (NoSuchPortException npe) {

        }
        SerialPort port = (SerialPort)
portId.open("Título comunicacion serial", 9600);
        serialOut = port.getOutputStream();
        serialInt = port.getInputStream();
        port.setSerialPortParams(9600,
            SerialPort.DATABITS_8,
            SerialPort.STOPBITS_1,
            SerialPort.PARITY_NONE);

        while (true) {
            serialOut.write(cadena.getBytes());
            //serialOut.write(valor1);
            //serialOut.write(valor2);
```

```
//serialOut.write(valor3);

//serialOut.toString();

while (serialInt.available() > 0) {
    System.out.print((char)
(serialInt.read()));
    //System.out.println("No esta mandando
nada ");
    //serialOut.write(cadena.getBytes());
}
}
} catch (Exception e) {
    e.printStackTrace();
}
```