



Tecnológico Nacional de México

Instituto Tecnológico de Tuxtla Gutiérrez



Informe técnico de residencia profesional.

Automatizar el control de llenado de contenedores de botella.

Carrera:

Ingeniería mecánica.

Presenta:

Marco Antonio Morales Hernández.

Periodo:

Enero- junio 2018

Ing. Ignacio arrijoja cárdenas

Ing. José Angel Bautista Meneses

Asesor Interno

Asesor Externo

INTRODUCCION.

En ALPLA, estamos convencidos de que las soluciones para los envases competitivos con importantes prestaciones solo son posibles si comprendemos el mercado y sus exigencias en permanente cambio. El trabajo intensivo asociado a nuestros clientes y la investigación proactiva nos permiten seguir siendo relevantes con nuestros productos.

En la actualidad los sistemas de control automáticos son de gran importancia para las industrias ya que sin ellos no se podrían mantener los diferentes procesos con eficiencia, calidad y a menor tiempo siendo necesario ajustar diversos parámetros de acuerdo con las condiciones de la planta y del operador, de acuerdo con la problemática que se presentan como, por ejemplo, la merma por botellas que están aplastadas o en su caso defectuosa, contaminación del producto, por derrame o caídas antes de llegar al destino deseado, porque es una empresa comprometida en entregar un producto de calidad al cliente.

El proyecto por desarrollar es más que nada el sistema automático del llenado de un contenedor de botellas llamado silo ya que es de gran importancia para la industria ya que permite suministrar botellas en caso de paro de producción, para evitar el aplastamiento del producto por exceso de llenado.

Este proceso es necesario para que el operador no necesite estar pendiente en ir checar, aplastamiento de botellas o contaminación del producto que son perdidas para la empresa, para evitar esto la empresa requiere que se haga un sistema donde avise que ya está lleno al nivel deseado por medio de una alarma y torreta, así también se esté monitoreando por medio de un software y estar visualizando.

Contenido.

Índice	
Introducción	2
Capítulo 1: Generalidades del proyecto.	7
1.1 Justificación.	7
1.2 Objetivos.	9
1.2.1 Objetivo general.	9
1.2.2 Objetivo específico.	9
1.3 Problemas a resolver.	10
1.4 Alcances.	11
1.5 Limitaciones.	11
Capítulo 2: Datos de la empresa.	12
2.1 Antecedentes de la empresa.	12
2.2 Ubicación.	13
2.3 Política de la empresa.	14
2.3.1 Misión.	14
2.3.2 Visión.	14
Capítulo 3: Fundamentación teórica.	15
3.1 Sensor ultrasónico.	15
3.2 Métodos de medición por ToF.	17
3.2.1 Umbral Simple.	18
3.2.2 Umbral Variable.	18
3.2.3 Ganancia Variable.	18
3.3 Problemáticas Asociadas.	18
Capítulo 4: Cálculos de diseño	19
4.1 Temperatura y humedad.	19
4.2 Entorno.	21
4.3 Sensor para la detección de objetos (ultrasonidos hc-sr04).	22
4.4 Características.	23
Capítulo 5: Historia al entorno labview.	24

5.1 Labview 1.0	24
5.2 Presentación del entorno LabVIEW.	25
5.3 Instrumentación virtual.	26
Capítulo 6: Desarrollo y resultado del proyecto físico.	28
6.1 El Arduino.	28
6.1.2 Arduino mega.....	29
6.2 Programación para el control de nivel de llenado.....	31
6.3 Diagrama de flujo.	45
6.4 Símbolos del diagrama de flujo.....	46
6.5 Materiales que se utilizaron en la construcción del proyecto.....	48
6.6 Esquema de la placa arduino mega.....	51
6.7 Diagrama eléctrico de conexiones	52
6.8 Manual de uso de tablero de control de llenado de silos.	59
6.9 Cálculos de llenado de silos.....	60
Capítulo 7: Resultados.....	61
7.1 Resultados de la plataforma labview.	61
Capítulo 8: Conclusiones y recomendaciones.....	64
8.1 CONCLUSIONES.....	64
8.2 Recomendaciones.	65

Contenido de figuras.

Figura 1 Planta de coca cola.....	12
Figura 2 Ubicación de la planta.....	13
Figura 3 Ubicación vía satélite.....	13
Figura 4 Señal enviada y recibida por sensor.....	15
Figura 5 Funcionamiento del sensor.....	17
Figura 6 Proyección por medio del objeto.....	20
Figura 7 Error vs distancia.....	22
Figura 8 Sensor ultrasónico conectado al arduino.....	23
Figura 9 Diagrama de conexión sensor y arduino.....	23
Figura 10 Estructura de instrumento virtual.....	27
Figura 11 Diagrama de bloques de la placa de E/S de arduino.....	28
Figura 12 Caja de acrílico.....	47
Figura 13 Se está pintando, para mejor estética.....	47
Figura 14 Relevador.....	48
Figura 15 Sensor ultrasónico.....	48
Figura 17 Conector rj45 hembra.....	48
Figura 16 Resistencia 1/2 Kohm.....	48
Figura 18 Cable utp.....	48
Figura 19 Jumpers (macho).....	48
Figura 20 Regulador de 5v.....	49
Figura 21 Cargador de 9v.....	49
Figura 22 Plug hembra 3.5mm.....	49
Figura 23 Jumpers (hembra).....	49
Figura 24 Torreta 12v.....	49
Figura 25 Placa arduino mega 250.....	49
Figura 26 Sirena 12v.....	49
Figura 27 Botones.....	49
Figura 28 Base de acrílico del sensor.....	50
Figura 29 Plug macho 3mm.....	50
Figura 30 Brazo o soporte del sensor ultrasónico.....	50
Figura 31 Conector Jack alimentación hembra.....	50
Figura 32 Selector swich.....	50
Figura 33 Esquema 1.....	51
Figura 34 Esquema 2.....	52
Figura 35 Esquema 3.....	53
Figura 36 Diagrama de eléctrico de conexiones del proyecto.....	54
Figura 37 Se están realizando pruebas.....	55

Figura 38 Aquí se realizan pruebas previas antes del montaje posterior.	55
Figura 39 Soporte colocado.	56
Figura 40 Ultrasónico unido al soporte.	56
Figura 41 Tablero del control de llenado de silos.	57
Figura 42 En funcionamiento en corrida cc3.0 lt en posición silo 2.	57
Figura 43 El tablero se encuentra en paro ya que se alarma y el operador intervino.	58
Figura 44 Manual de usuario del tablero de llenado de silos.	59
Figura 45 Análisis de llenado de diferentes presentaciones.	60
Figura 46 Las flechas de colores indica el nivel de llenado de cada presentación de botellas.	60
Figura 47 Nivel bajo de llenado.	61
Figura 48 Nivel medio en un 50% de llenado.	61
Figura 49 Nivel lleno de acuerdo con la presentación que eligió el operador.	62
Figura 50 Diagrama de bloques de la plataforma labview.	62
Figura 51 Esquema de diagrama de bloques.	63

Capítulo 1: Generalidades del proyecto.

1.1 Justificación.

En la industria, el llenado de contenedores (silos) son muy importantes, para cuando se requiere botellas para la industria refresquera, cuando la maquinaria está en mantenimiento o en paro por fallas ocasionales que se presentan en producción.

Las razones por las cuales se requiere medir el control de llenado de botellas son para evitar que aplaste o dañe las botellas que quedan por debajo de las demás que están cayendo, así como también evitar el derrame de estas ya que debido a que se manejan diversas presentaciones se debe tener cuidado.

El mantener condiciones seguras de operación, la optimización y control de proceso, el ámbito de la investigación y el desarrollo entre otros.

La medición del nivel por capacidad cubre un amplio rango de aplicaciones que no se limita la ingeniería de procesos. Unas sondas simples que ofrecen una gran variedad de posibilidades para el control del nivel en este caso de botellas.

En este proyecto se presenta un ahorro de energía en las bandas en las cuales se transportan las botellas para su posterior llegada a los silos, así como también mantener monitoreado el control de llenado y vaciado del mismo para que a su vez nos avise una alarma del control del silo.

En este caso se utilizan un tipo de sensores llamados capacitivos, estos tienen la capacidad de captar la presencia de sólidos y un contador que active el circuito que lo mantenga enclavado o bloqueado.

El sistema será capaz de avisar a través de una alarma y detenerse para su mayor control de acuerdo con la demanda a los niveles de llenado.

En las industrias los sistemas de control de llenado de silos son indispensables para en casos de emergencias, la mayoría de estas en controlados o manejas con la ayuda visual del operador de producción para que pare o arranque el llenado, vaciado según sea en caso necesario en ese momento.

Las técnicas utilizadas de medición de nivel están presentes en los campos del sector alimenticio para su manipulación, así como entre otras que requieren controlar cierto nivel que desean monitorear.

La propuesta de este proyecto es la mejora para automatizar el control de llenado de botellas en diferentes presentaciones, así determinar en qué momento queremos llenar o vaciar y a través de una alarma avisar en qué estado se encuentra, dependiendo el caso que se requiera para su posterior necesidad para su manipulación de ello.

En la actualidad el llenado de silos es de suma importancia dependiendo de la demanda del mercado, que quiere mayor cantidad de productos por ello se requiere el monitoreo y control de este para facilitar al operador en la fabricación del producto.

1.2 Objetivos.

1.2.1 Objetivo general.

- Diseñar, ensamblar e implementar la optimización de un sistema capaz de detectar el llenado de los silos por medio de una alarma y torreta de forma automática, así como permitir el control digital para el monitoreo de este.

1.2.2 Objetivo específico.

- Implementar el ahorrar de derrame de botellas en el piso por cuestión de inocuidad.
- Ahorrar energía eléctrica en los motores de las bandas transportadoras de botellas hacia los silos.
- Comparar con los sistemas que ya existen con los que se desean implementar.
- Investigar sobre la factibilidad en el diseño del proyecto.
- Diseñar e implementar un sistema de monitoreo y control digital para el llenado de los silos.
- Diseñar un entorno grafico a través de computadora que se encargara de visualizar las variables a medir.

1.3 Problemas a resolver.

Debido a que las pequeñas, medianas y grandes industrias requieren sistemas de producción más eficientes, surge la necesidad de implementar sistemas de control automáticos. En estos tipos de sistemas, es de vital importancia que los dispositivos que actúan como elementos integradores de este, ofrezcan un nivel de seguridad que permita garantizar el desarrollo completo del proceso en ejecución en industrias.

Esta es la razón por la cual se deben utilizar señales de un sensor ultrasónico en la medida de nivel de botellas. Para resolver este problema estamos diseñando un sistema que nos permita manipular y controlar en el llenado de los silos en industria de la fabricación de botellas y monitorear a través de computadora, utilizando (*TARJETA PROGRAMABLE Y LABVIEW*), con el objetivo principal de evitar el desperdicio de botellas y por otra parte evitar el alto consumo de energía eléctrica. Por otro lado, el sistema será capaz de funcionar sin la intervención del ser humano, para lograrlo utilizaremos sensores ultrasónicos, que es un dispositivo económico, fáciles de manipular y pueden detectar objetos sin necesidad de algún filtro o adaptación especial.

Este principio de medición sirve asimismo para aplicaciones con sólidos debido a que no es sensible ni a polvos, ni a ruidos de fondo. El método de medición por ultrasonidos es una solución probada, comprobada y económica para la medición de nivel en aplicaciones de objetos.

1.4 Alcances.

- Capacidad de analizar los parámetros de distancias y poder revisar constantemente.
- El económico costo para la empresa el realizar el proyecto de automatizar el control de monitorear el llenado.
- Ahorra tiempo para el operador encargado de la producción de la empresa
- Evitar las pérdidas de dinero a la empresa.
- El reducir el consumo de energía eléctrica.

1.5 Limitaciones.

- No intervenir en equipos de planta para posible mejora.
- Poco interés en aprobación de mejoras para pruebas y no se implemente por fallo.
- El poco conocimiento de realizar automatización en planta a bajos costos.

Capítulo 2: Datos de la empresa.

2.1 Antecedentes de la empresa.

Coca-Cola Company como la empresa líder en bebidas carbonatadas y no carbonatadas, sin embargo, ese espacio no es un lugar que se lo han ganado solos, porque parte de la demanda de sus productos depende de la calidad del empaque o embalaje que utilizan.

Lo anterior lo han conseguido a través de diversas alianzas históricas para esta compañía, una de esas alianzas la realizaron con el Corporativo Alpla México & Centroamérica

La planta embotelladora abrió sus puertas en 1994, de Coca-Cola en la comunidad indígena de San Felipe Ecatepec, en el estado de Chiapas, en Periférico Norte Poniente, Barrio de Fátima.

Según, Ivo José Fuentes, Gerente de Planta de Alpla, su relación única es de Cliente-Proveedor misma que está sustentada en una base comercial, es decir, no tienen más Cliente que Coca-Cola FEMSA.

Es por ello que alpla se caracteriza por ser una empresa en la vanguardia mundialista como innovadora, emprendedora y con todo el talento humano para desarrollar soluciones de empaque de plástico que satisfacen en forma óptima las necesidades del mercado” argumentó.

Como Alpla a nivel mundial, la empresa ha innovado en la producción de botellas con material reciclado como empresa comprometida con el medio ambiente y cuidado de los recursos, tal que hoy día ya se obtienen envases PET con mezcla de resinas PET en un alto porcentaje de material reciclado (50%).



Figura 1 Planta de coca cola.

2.2 Ubicación.

La empresa alpa trading se ubica dentro de las instalaciones de coca cola femsa.

Se localiza Periférico Norte Poniente #89, Barrio de Fátima, 29264 San Cristóbal de las Casas, Chis.



Figura 2 Ubicación de la planta.



Figura 3 Ubicación vía satélite.

2.3 Política de la empresa.

Alpla es líder mundial en el desarrollo y la producción de soluciones de envases de plástico. Producimos sistemas innovadores de envase, botellas, tapas y piezas moldeadas por inyección para una amplia gama de industrias. Nuestra empresa y nuestros productos se caracterizan por nuestra tradición como empresa familiar, nuestro uso de las últimas tecnologías, nuestro deseo de soluciones innovadoras y la experiencia y el compromiso de nuestros empleados.

2.3.1 Misión.

ALPLA crea soluciones para embalajes de plástico que satisfacen de forma óptima los requisitos del mercado.

2.3.2 Visión.

Alcanzar el liderazgo en tecnología y mercado es nuestro objetivo.

Capítulo 3: Fundamentación teórica.

3.1 Sensor ultrasónico.

Un sensor ultrasónico calcula la distancia utilizando un transductor que emite “paquetes” de ultrasonido que guardan dentro una serie de ondas sonoras intermitentes. El paquete se emite en forma cónica, rebota en la superficie objetivo y regresa en un transductor.

El tiempo requerido por el sonido para ir y volver se mide y se convierte a unidades de distancia; este tiempo se mide en milisegundos. Este sensor tiene el mismo funcionamiento que un radar ya que envía pulsos a alta frecuencia en este caso ultrasónicos. Está compuesto por dos piezoeléctricos: un transmisor y receptor ambos en forma de cilindro.

El transmisor se encarga de emitir la señal ultrasónica para luego ser rebotada a través de un objeto llegando al cilindro receptor. Debido a que la señal le tardará un tiempo en regresar desde que se emite, es obvio pensar que la velocidad con la cual llegará al receptor está íntimamente relacionada con la distancia de transmisión y el rebote.

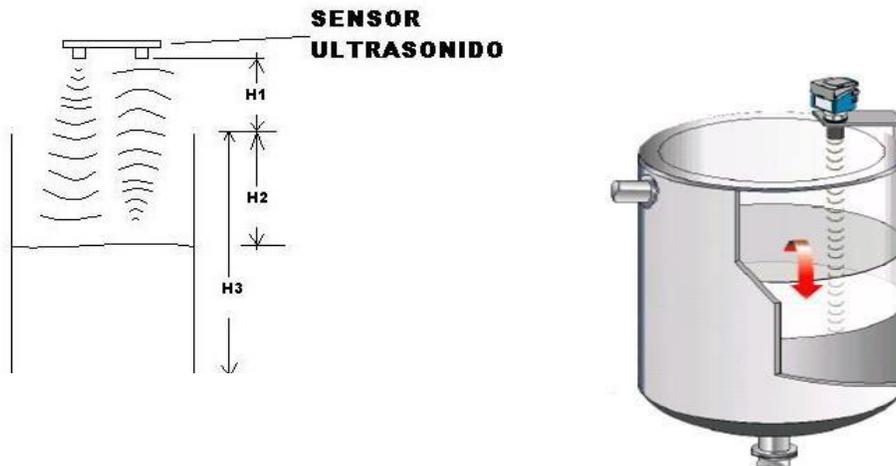


Figura 4 Señal enviada y recibida por sensor.

La automatización y monitoreo de procesos industriales, la investigación robótica, y mucho otros campos, necesitan de elementos que permitan al sistema de procesamiento obtener información del entorno, a estos dispositivos se llama

transductores. Uno de los transductores más usados es el ultrasónico, debido a su bajo coste, robustez, y sencillez circuital, los cuales permiten medición de distancia, creación de radares, mapeo de entornos, ecografías, entre otras aplicaciones.

Una señal ultrasónica es una onda sonora que está en el rango de frecuencias (20kHz, 100MHz) el cual no es audible para ciertos seres vivientes como el hombre.

Como onda sonora que es su velocidad es dependiente de la temperatura del medio:

$$v_s = v_0 \sqrt{1 + \frac{T}{273}} \quad \dots\dots (1)$$

Donde V_0 es la velocidad del sonido en el medio a una temperatura de 0°C y T es la temperatura actual. En el aire a una temperatura ambiente de 20°C y tomando V_0 de 331m/s la velocidad del sonido es entonces 343m/s .

Como se mencionó, las señales ultrasónicas son usadas para extraer información del entorno, uno de estos casos es la distancia a un objeto. Comúnmente se usa una topología con 2 transductores ultrasónicos, un transmisor (Tx) que emite la señal al medio y un receptor (Rx) que recibe el eco proveniente de los obstáculos dentro del haz de medición.

La manera como se genera la onda transmitida, y el posterior procesamiento del eco, puede ser realizado mediante varios métodos que se clasifican en dos grupos: Generación de Ondas Continuas o Generación de Ondas en Forma de Pulso.

El primero mide la diferencia de fase o frecuencia en una señal modulada en amplitud o frecuencia respectivamente. El segundo grupo se basa en la medición del Tiempo de Vuelo de la Señal (Time of Fly ToF), es decir el tiempo que tarda un pulso de onda en recorrer la distancia entre el sensor y el objetivo.

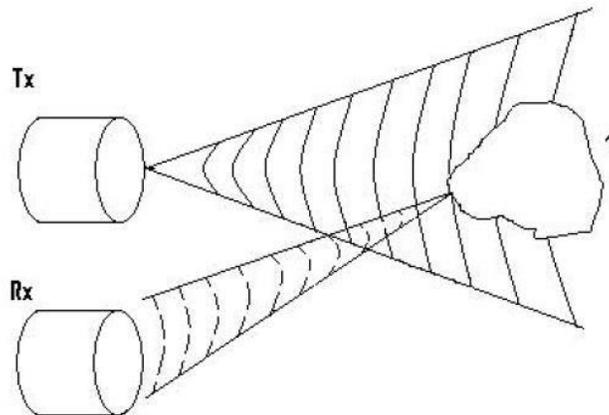


Figura 5 Funcionamiento del sensor.

Emisión de una señal ultrasónica y recepción del eco proveniente del objeto 1

3.2 Métodos de medición por ToF.

La ecuación que define el método de ToF es:

$$X = V_s * \frac{t_v}{2} \quad \dots(2)$$

Donde:

V_s: es la velocidad del sonido en el entorno. t_s: es el tiempo de vuelo de la señal.

X: la distancia a la que se encuentra el objeto.

Se divide por 2 el tiempo de vuelo, ya que este es la duración que tiene el recorrido de la onda desde el transmisor al objeto y desde el objeto al receptor.

Algunos de los métodos para la medición de distancia usando la técnica de ToF son:

3.2.1 Umbral Simple.

La señal de eco recibida es amplificada y luego comparada con un nivel de voltaje o umbral, que de ser superado indica la presencia de obstáculo en el ancho de haz del sensor. Esta topología, aunque es la más sencilla también es la más susceptible a ruido y su máximo y mínimo alcance son muy limitados.

3.2.2 Umbral Variable.

La señal de eco recibida es amplificada y luego comparada con un umbral variable en el tiempo. Este método aumenta el alcance máximo, pero genera una amplia zona muerta.

3.2.3 Ganancia Variable.

Conforme el tiempo de vuelo es mayor la amplitud de la señal disminuye, por lo cual este método da mayor ganancia a aquellas señales que corresponden a objeto más lejanos y luego las compara con un umbral simple. Éste último método permite la mayor precisión en las mediciones, pero posee la mayor complejidad circuital.

3.3 Problemáticas Asociadas.

La precisión de la medición de distancia por ultrasonidos depende de factores como la temperatura y humedad del entorno, tamaño, posición y material del objeto sobre el que incide la onda, ecosmúltiples y crosstalk, siendo este último uno de los más críticos.

Capítulo 4: Cálculos de diseño

4.1 Temperatura y humedad.

Aunque es importante conocer el valor exacto de la temperatura para el cálculo de la velocidad del sonido en el entorno, 1°C de desfase genera en la medición absoluta de distancia un error determinado por la siguiente ecuación:

$$Erm = R * 0.0018 \quad \dots\dots(3)$$

Donde:

R = Rango de distancia

Erm(R) = Error de imprecisión en la distancia del objetivo en metros para una imprecisión en la temperatura de 1°C cuando el rango R está en metros.

1) *Tamaño del Objeto.*

Conforme se aleja la onda de su fuente generadora, su frente de onda se expande. Éste debe ser siempre 10 veces menor que los objetos sobre los que incide, para obtener una máxima amplitud de eco, por lo tanto, objetos diminutos que se hallen lejos de la fuente generadora no serán detectados por el sistema, limitando así el rango de medida del ultrasonido.

2) *Impedancia Acústica Objeto.*

La impedancia acústica se define como el producto entre la presión acústica en un punto y la velocidad de vibración de la partícula situada en ese punto

$$Z = \rho * C \quad \dots\dots(4)$$

Donde:

Z = Impedancia Acústica

ρ = Densidad

C = Velocidad

La importancia de la impedancia acústica en la medición de distancia con ultrasonidos radica en que la amplitud de la onda ultrasónica reflejada depende de esta característica del objeto sobre el que incide la onda, mediante la ecuación:

$$R = \frac{Z - 1}{1 + Z} \dots(5)$$

R = Índice de Reflexión de onda Plana
Z = Impedancia Acústica

Siendo ambos números complejos

4) Posición del Objeto.

La señal ultrasónica tiene una máxima reflexión sobre el objeto, cuando la cara sobre la que incide la onda es perpendicular al eje acústico del sensor. Si esto no sucede la onda reflejada tendrá menor amplitud y conforme el ángulo en que incida la onda se aleje de 90°, la señal tenderá a perderse y no será percibida por el receptor.

5) Ecos múltiples y crosstalk

En todo entorno que sean utilizados ultrasonidos estarán presentes, además del objetivo, otros obstáculos que al incidir sobre ellos la onda ultrasónica generan reflexiones que son percibidas por el receptor, este caso se ilustra en la figura

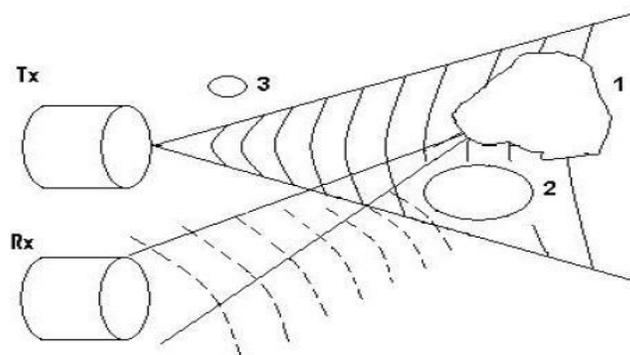


Figura 6 Proyección por medio del objeto.

Múltiples obstáculos en un entorno. 1. Objetivo 2. Obstáculo dentro del haz 3. Obstáculo fuera del Haz

En esta figura el objeto 1 es el objetivo, el 2 y el 3 son obstáculos adicionales, y la distancia medida por el sistema será la correspondiente al objeto 2, este error se debe a que el objeto 2 está dentro del haz de radiación y su eco es percibido por el receptor, y aunque el eco del objeto 1 también es recibido, el objeto 2 se encuentra más cercano. Asimismo, un haz de radiación inadecuado genera crosstalk por solapamiento de lóbulos, donde el haz de radiación interfiere con el haz de recepción, indicando al sistema que hay un objeto a una distancia cercana. Esta interferencia es de las más importantes, porque el máximo alcance conseguido depende en gran manera de como sea eliminada.

También la presencia de otros obstáculos en el entorno genera múltiples choques o ecos de una señal, que al llegar al receptor engañan al sistema indicando erróneamente la presencia de un obstáculo.

4.2 Entorno.

Las pruebas de este sonar se hicieron dentro de un laboratorio con varios obstáculos cercanos como mesas, sillas, osciloscopios, entre otros, que generan ecos de amplitud superior a los ecos recibidos de objetivos a distancias superiores de 110cm.

Para éste primer sensor de distancia desarrollado, en la figura 3 se exhibe el comportamiento del error en función de la distancia, donde se puede notar que la fiabilidad del sistema está garantizada solo hasta 110cm. (Jiménez Builes, 2008)

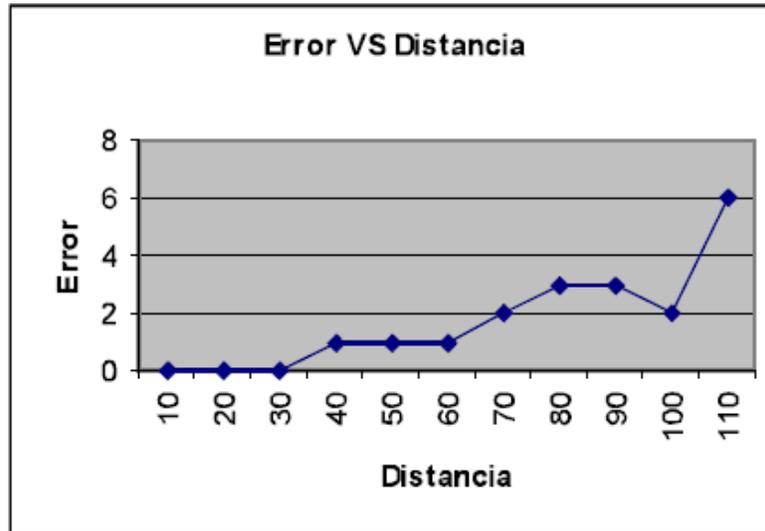


Figura 7 Error vs distancia.

Error en función de la distancia para el sistema de umbral simple desarrollado.

<https://www.robots-hop.com/media/files/pdf/arduinomega2560datasheet.pdf>

4.3 Sensor para la detección de objetos (ultrasonidos hc-sr04).

Este sensor tiene el mismo funcionamiento que un radar ya que envía pulsos a alta frecuencia en este caso ultrasónicos. Está compuesto por dos piezoeléctricos: un transmisor y receptor ambos en forma de cilindro. El transmisor se encarga de emitir la señal ultrasónica para luego ser rebotada a través de un objeto llegando al cilindro receptor.

Debido a que la señal le tardará un tiempo en regresar desde que se emite, es obvio pensar que la velocidad con la cual llegará al receptor está íntimamente relacionada con la distancia de transmisión y rebote.

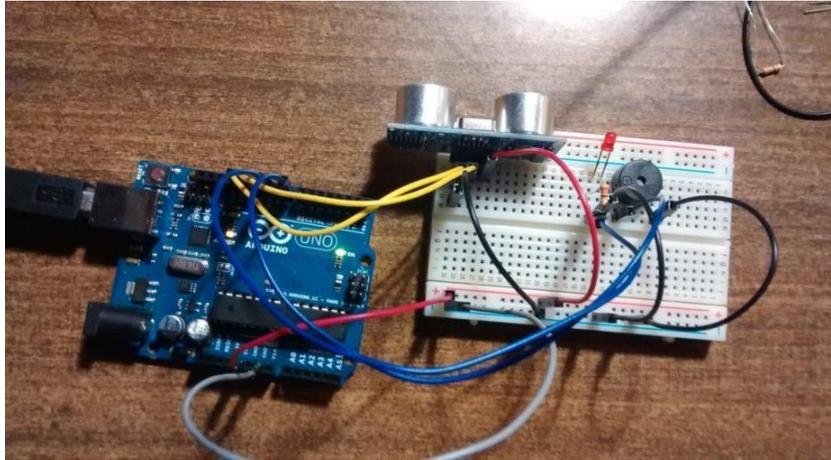


Figura 8 Sensor ultrasónico conectado al arduino.

El sensor consta de 4 pines: "VCC" conectado a la salida de 5V de la placa, "Trig" conectado al pin digital de la placa encargado de enviar el pulso ultrasónico, "Echo" al pin de entrada digital que recibirá el eco de dicho pulso y "GND" a tierra.

4.4 Características.

Este sensor tiene únicamente cuatro pines. Dependiendo del fabricante, puede tener desde tres hasta cinco pines. Para este caso se tiene:

- VCC: Voltaje de alimentación.
- TRIG: Pin de disparo.
- ECHO: Pin de eco.
- GND: Ground.
- Medición mínima: 2 cm.
- Medición máxima: 5 m.

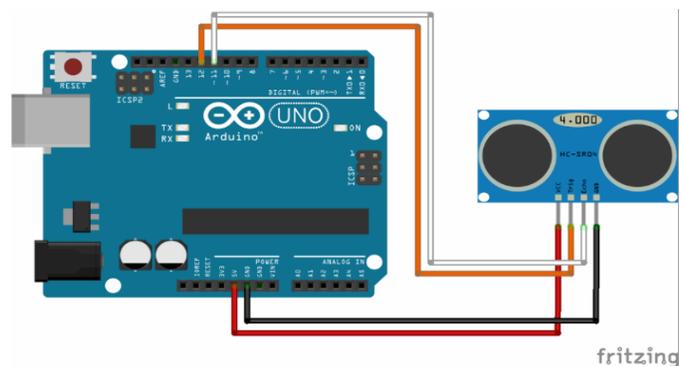


Figura 9 Diagrama de conexión sensor y arduino.

- Ángulo efectivo $< 15^\circ$.

Capítulo 5: Historia al entorno labview.

5.1 Labview 1.0

LabVIEW 1.0 fue lanzado al mercado en 1986 por la compañía National Instruments. Desde entonces integró características importantes que aun hoy después de dos décadas de desarrollo y mejoras siguen siendo el núcleo fundamental de este entorno de programación. El remplazo de líneas de texto por iconos gráficos y la escritura de código por el enlace de estos iconos ha logrado disminuir mucho los tiempos de desarrollo que se requieren para crear un instrumento virtual. Originalmente creado para computadores Mac, hoy este tipo de instrumentación abarca una gran variedad de sistemas operativos desde Windows hasta sistemas en tiempo real. La instrumentación virtual integra este lenguaje de programación con hardware especializado para crear soluciones que permiten adquirir, analizar y presentar datos en laboratorios de ciencia básica e ingeniería aprovechando al máximo las capacidades de cálculo y comunicación de los computadores modernos. El éxito de este tipo de instrumentación ha sido muy grande y puede atribuirse a la facilidad de uso del entorno gráfico de programación y a los dispositivos de hardware auto configurables plug and play que permiten a profesionales de muchas áreas trabajar en monitoreo y control de sistemas sin ser especialistas en software, programación o electrónica. Otro parámetro importante ha sido la capacidad de integración con áreas específicas del conocimiento por medio del desarrollo de Tools Kits especializados que permiten crear aplicaciones avanzadas utilizando herramientas prediseñadas.

5.2 Presentación del entorno LabVIEW.

Un programa desarrollado en LabVIEW es llamado un instrumento virtual y tiene la extensión .VI. Está compuesto por un panel frontal y un diagrama de bloques. El primero es la interface humano máquina en la cual se encuentran controles e indicadores que permiten al usuario interactuar con el instrumento virtual. Los controles e indicadores simulan la apariencia de un instrumento real. De esta forma el operador puede manipular las variables del programa sin tener ningún conocimiento

en programación. Este panel frontal ha sido uno de los grandes logros de LabVIEW ya que es fácilmente entendido y manejado por científicos e ingenieros de todas las ramas. Todos los controles pueden ser modificados por el usuario en posición, tamaño y color. Además, cuenta con algunos paquetes adicionales altamente especializados que permiten colocar controles e indicadores propios de campos muy específicos, algunos de esos paquetes son por ejemplo el DSSC (Data Login Sytem Control) que facilita la adquisición de datos y el control de instrumentos industriales y el Multisim para el diseño de aplicaciones con circuitos. Todos los controles e indicadores se encuentran en la paleta de controles que está disponible solo para el panel frontal, esta paleta se muestra en la figura.

En el diagrama de bloques se encuentra el código de programación el cual está basado en iconos que se conectan para establecer un flujo de datos, cada control e indicador del panel frontal tiene un terminal que lo representa en el diagrama de bloques y desde o hacia este terminal viajan datos a través de alambres de conexión. Todas las herramientas y funciones se obtienen de la paleta de funciones que está disponible en el diagrama de bloques, las diferentes funciones están agrupadas en sub paletas según su utilidad. Existen subpaletas tales como instrument i/o (para

comunicación con instrumentos externos), Mathematics (para cálculos estadísticos, ajuste de datos o trabajo en álgebra lineal) y signal processing para el tratamiento de señales, entre muchas otras. De esta manera y conociendo la organización de las funciones dentro de cada una de las subpaletas, la programación puede hacerse de manera totalmente intuitiva. Aunque LabVIEW puede entender subrutinas basadas en lenguajes de texto importadas desde otros lenguajes de programación, los instrumentos virtuales pueden ser hechos en su totalidad colocando iconos gráficos, que representan funciones, dentro de estructuras de ejecución.

5.3 Instrumentación virtual.

La instrumentación virtual nace de la utilización del computador como una herramienta de medida y análisis de datos en el laboratorio. Aunque los programas que se desarrollan en LabVIEW se llaman instrumentos virtuales, en su forma más general un instrumento virtual es la unión de un computador con el software y el hardware especializado que le permiten realizar las mismas funciones que un instrumento real. Estas funciones son adquirir, almacenar, analizar, mostrar y comunicar datos y tomar acciones de control sobre determinado proceso cuando así se requiera. Un instrumento virtual no es una simulación, el término virtual se refiere a que el instrumento no existe como un instrumento tradicional por separado si no como parte de un sistema basado en software. Software que, en la pantalla del computador simula la apariencia de un instrumento real. Las ventajas de utilizar instrumentación virtual son esencialmente todas aquellas que se derivan de la utilización de un computador y son entre otras: alta velocidad de procesamiento de datos, capacidad prácticamente ilimitada de almacenamiento de datos, comunicación de datos a diferentes niveles y altas velocidades, toma de decisiones según argumentos lógicos preestablecidos y capacidad comunicación con instrumentos externos o interfaces de adquisición de datos por diferentes tipos de interfaces o buses disponibles. (Quiñones, 2010)

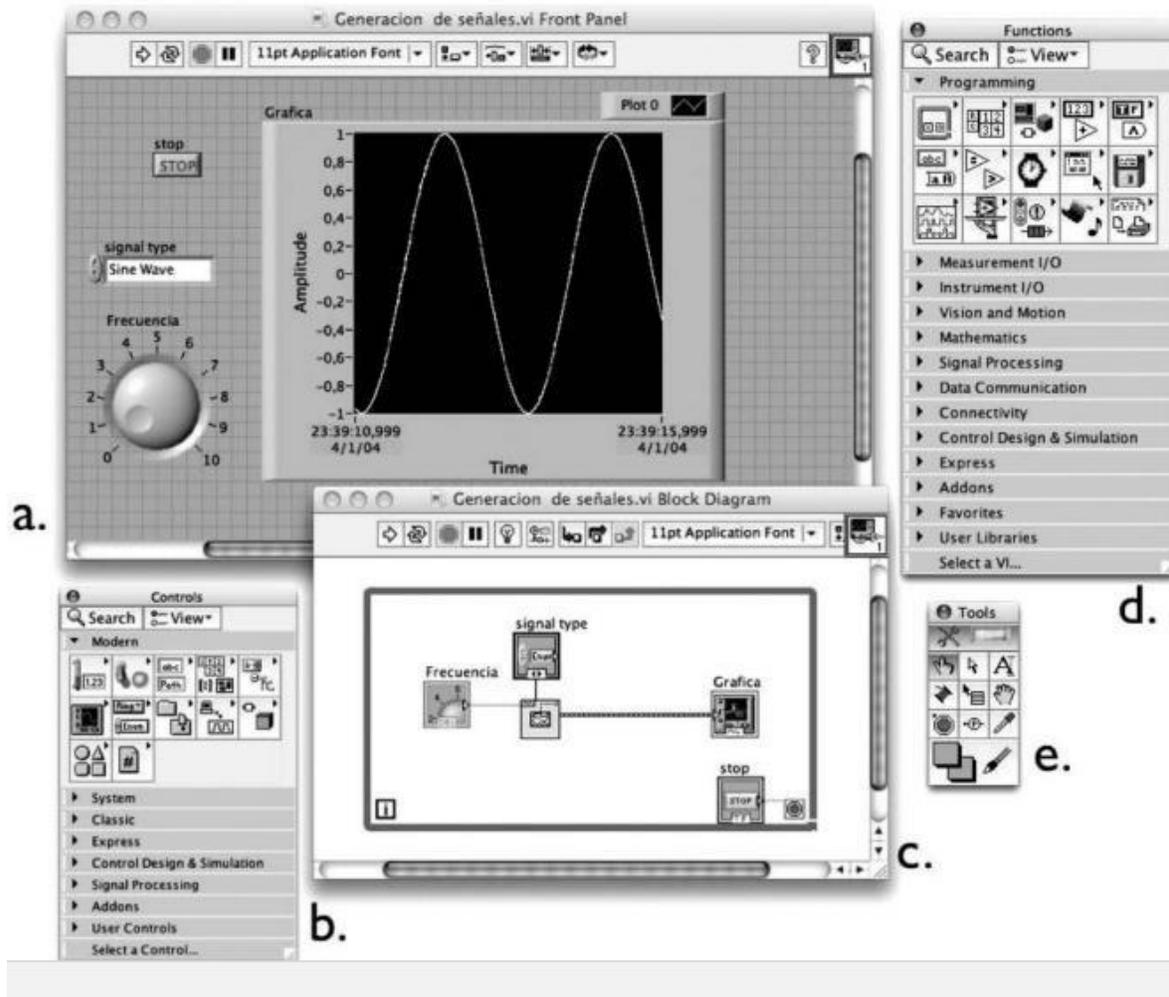


Figura 10 Estructura de instrumento virtual.

- a. Panel frontal con controles e indicadores.
- b. Paleta de controles.
- c. Diagrama de bloques con funciones y terminales
- d. Paleta de funciones
- e. Paleta de herramientas.

Capítulo 6: Desarrollo y resultado del proyecto físico.

6.1 El Arduino.

Debido a la historia y la evolución de Arduino, hay muchas variaciones en lo que se puede llamar un Arduino. La lista aumenta cada día. La oferta oficial del Equipo Arduino consiste en el Arduino Uno Y el más grande Arduino Mega 2560.

Cuando la mayoría de la gente piensa en Arduino, se imaginan el pequeño, rectangular (y probablemente azul) Placa de circuito impreso (PCB). Esto se denomina correctamente la tarjeta de E / S.

La placa de E / S es la parte físicamente tangible del sistema Arduino. Técnicamente hablando, el término Arduino cubre el hardware, software, equipo de desarrollo, filosofía de diseño y esprit de corps de la comunidad de usuarios. Sin embargo, a menudo se escucha a la gente decir cosas como, "Por favor, dame que Arduino", o "cuidado con ese Arduino, Eugene. Arduino fue desarrollado originalmente en Ivrea, Italia. Arduin de Ivrea era el rey de Italia sobre un hace mil años y se celebra en la historia local. La Piazza Gioberti alberga un pub llamado así famoso rey, que algunos dicen que sólo se nombra después de la carretera está en, la Vía Arduino. El nombre Arduino es un nombre italiano masculino que significa "amigo fuerte". Ser un nombre propio, Arduino siempre está capitalizado.

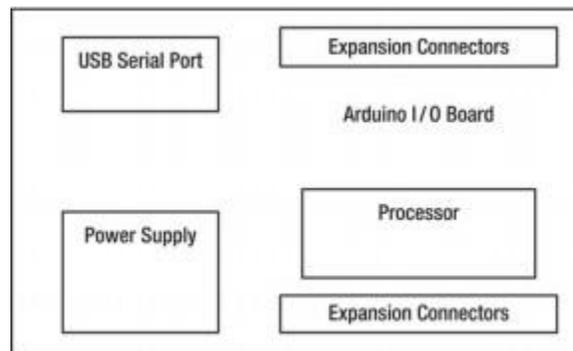


Figura 11 Diagrama de bloques de la placa de E/S de arduino.

6.1.2 Arduino mega.

Arduino Mega es una tarjeta de desarrollo open-source construida con un microcontrolador modelo Atmega2560 que posee pines de entradas y salidas (E/S), analógicas y digitales. Esta tarjeta es programada en un entorno de desarrollo que implementa el lenguaje Processing/Wiring. Arduino puede utilizarse en el desarrollo de objetos interactivos autónomos o puede comunicarse a un PC a través del puerto serial (conversión con USB) utilizando lenguajes como Flash, Processing, MaxMSP, etc. Las posibilidades de realizar desarrollos basados en Arduino tienen como límite la imaginación.

El Arduino Mega tiene 54 pines de entradas/salidas digitales (14 de las cuales pueden ser utilizadas como salidas PWM), 16 entradas análogas, 4 UARTs (puertos serial por hardware), cristal oscilador de 16MHz, conexión USB, jack de alimentación, conector ICSP y botón de reset. Arduino Mega incorpora todo lo necesario para que el microcontrolador trabaje; simplemente conéctalo a tu PC por medio de un cable USB o con una fuente de alimentación externa (9 hasta 12VDC). El Arduino Mega es compatible con la mayoría de los shields diseñados para Arduino Duemilanove, diecimila o UNO.

Esta nueva versión de Arduino Mega 2560 adicionalmente a todas las características de su sucesor utiliza un microcontrolador ATmega8U2 en vez del circuito integrado FTDI. Esto permite mayores velocidades de transmisión por su puerto USB y no requiere drivers para Linux o MAC (archivo inf es necesario para Windows) además ahora cuenta con la capacidad de ser reconocido por el PC como un teclado, mouse, joystick, etc.

Características:

Microcontrolador ATmega2560.

Voltaje de entrada de – 7-12V.

54 pines digitales de Entrada/Salida (14 de ellos son salidas PWM).

16 entradas análogas.

256k de memoria flash.

Velocidad del reloj de 16Mhz.

6.2 Programación para el control de nivel de llenado.

```
Archivo Edición Formato Ver Ayuda
// definir pines de sensor
const int trigPin = 9;
const int echoPin = 10;

const int trigpin = 6;
const int echopin = 4;

int tiempo=0;
// definir pines de alarma y estrobo
const int buzzer = 11;
const int ledPin = 13;
// definir leds de botones
const int ledPin1 = 24;
const int ledPin2 = 25;
const int ledPin3= 26;
const int ledPinP = 27;
const int ledPinS = 28;
// definir botones
const int BOTONP = 22;
const int BOTONS = 5;
const int BOTON3 = 12;
const int BOTON2 = 8;
const int BOTON = 7; // pin de entrada botón

// defines variables
long duration;
int distance;
int safetyDistance;

long duratioN;
int distance2;
int safetyDistance2;

int val = 0; //val se emplea para almacenar el estado
// del botón
int state = 0; // 0 LED apagado, mientras que 1 encendido
<
```

```
Archivo Edición Formato Ver Ayuda
int state = 0; // 0 LED apagado, mientras que 1 encendido
int old_val = 0; // almacena el antiguo valor de val// setup de parámetros

int val2 = 0; //val se emplea para almacenar el estado
// del botón
int state2 = 0; // 0 LED apagado, mientras que 1 encendido
int old_val2 = 0;

int val3 = 0; //val se emplea para almacenar el estado
// del botón
int state3 = 0; // 0 LED apagado, mientras que 1 encendido
int old_val3 = 0;
int val4 = 0;
int state4 = 0; // 0 LED apagado, mientras que 1 encendido
int old_val4 = 0;
int val5 = 0;
int state5 = 0; // 0 LED apagado, mientras que 1 encendido
int old_val5 = 0;
int val6 = 0;
int state6 = 0; // 0 LED apagado, mientras que 1 encendido
int old_val6 = 0;

int val7= 0;
int state7 = 0; // 0 LED apagado, mientras que 1 encendido
int old_val7 = 0;

void setup() {

pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
pinMode(echoPin, INPUT); // Sets the echoPin as an Input

pinMode(trigpin, OUTPUT);
pinMode(echopin, INPUT);

pinMode(BOTONP, INPUT);
pinMode(BOTONS, INPUT);
<
```

```
Archivo Edición Formato Ver Ayuda
pinMode(BOTONS, INPUT);|
pinMode(BOTON, INPUT);
pinMode(BOTON2, INPUT);
pinMode(BOTON3, INPUT);

pinMode(buzzer, OUTPUT);
pinMode(ledPin, OUTPUT);
pinMode(ledPin1, OUTPUT);
pinMode(ledPin5, OUTPUT);
pinMode(ledPin2, OUTPUT);
pinMode(ledPin3, OUTPUT);
pinMode(ledPinP, OUTPUT);
Serial.begin(9600); // Starts the serial communication

pinMode(BOTON, INPUT); //indicamos el pin de entrada
}

void loop ()
{
  if (digitalRead (BOTON) == HIGH && digitalRead (BOTON2) == LOW && digitalRead (BOTON3) == LOW && digitalRead (BOTONS) == LOW && digitalRead (BOTONP) == LOW )
    val= digitalRead(BOTON);
}

// lee el estado del Boton
// chequear si el boton esta presionado o no
if ((val == HIGH) && (old_val == LOW) ) {
  state=1-state;
  delay(10);
}
old_val = val; // val is now old, let's store it
if (state==1){
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
}

// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
```

```
Archivo Edición Formato Ver Ayuda
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
digitalWrite(ledPinP, LOW);
digitalWrite(ledPin1, HIGH);

// Reads the echoPin, returns the sound wave travel time in microseconds
duration = pulseIn(echoPin, HIGH);

// Calculating the distance
distance= duration*0.034/2;

safetyDistance = distance;
if (safetyDistance <= 30){
  delay(1000);
  tiempo++;
}
if(tiempo==30){
  digitalWrite(buzzer, HIGH);
  digitalWrite(ledPin, HIGH);
}
else if (digitalRead(BOTONP) == HIGH)
{
  digitalWrite(buzzer, LOW);
  digitalWrite(ledPin, LOW);
  digitalWrite(trigPin, LOW);
  digitalWrite(echoPin, LOW);
  digitalWrite(ledPinP, LOW);
  (state = 0);
}

// Prints the distance on the Serial Monitor
Serial.print("Distance: ");
Serial.println(distance);
}
```

<

```
Archivo Edición Formato Ver Ayuda  
}  
  
if (digitalRead (BOTON) == LOW && digitalRead (BOTON2) == LOW && digitalRead (BOTON3) == LOW && digitalRead (BOTONS) == LOW && digitalRead (BOTONP) == HIGH )  
{  
    digitalWrite(buzzer, LOW);  
    digitalWrite(ledPin1, LOW);  
    digitalWrite(ledPin2, LOW);  
    digitalWrite(ledPin3, LOW);  
    digitalWrite(ledPin5, LOW);  
    digitalWrite(ledPin, LOW);  
    digitalWrite(echoPin, LOW);  
    digitalWrite(trigPin, LOW);  
    digitalWrite(ledPinP, HIGH);  
    digitalWrite(echoPin, LOW);  
    digitalWrite(trigpin, LOW);  
  
    (state3 = 0);  
    (state2 = 0);  
    (state = 0);  
  
    (state5 = 0);  
    (state6 = 0);  
    (state7 = 0);  
  
}  
if (digitalRead (BOTON) == LOW && digitalRead (BOTON2) == LOW && digitalRead (BOTON3) == HIGH && digitalRead (BOTONP) == HIGH )  
{  
    digitalWrite(buzzer, LOW);  
    digitalWrite(ledPin1, LOW);  
    digitalWrite(ledPin2, LOW);  
    digitalWrite(ledPin3, LOW);  
    digitalWrite(ledPin5, LOW);  
    digitalWrite(ledPin, LOW);  
    digitalWrite(echoPin, LOW);  
    digitalWrite(trigpin, LOW);  
}
```

Archivo Edición Formato Ver Ayuda

```
digitalWrite(echoPin, LOW);  
digitalWrite(trigPin, LOW);  
digitalWrite(ledPin, HIGH);  
digitalWrite(echoPin, LOW);  
digitalWrite(trigPin, LOW);  
  
(state3 = 0);  
(state2 = 0);  
(state = 0);  
  
(state5 = 0);  
(state6 = 0);  
(state7 = 0);  
  
}  
if (digitalRead (BOTON1) == LOW && digitalRead (BOTON2) == HIGH && digitalRead (BOTON3) == LOW && digitalRead (BOTON4) == LOW && digitalRead (BOTON5) == LOW && digitalRead (BOTON6) == LOW )  
    val2= digitalRead(BOTON2);  
{  
    // lee el estado del Boton  
    // chequear si el boton esta presionado o no  
    if ((val2 == HIGH) && (old_val2 == LOW) ) {  
        state2=1-state2;  
        delay(10);  
    }  
    old_val2 = val2; // val is now old, let's store it  
    if (state2==1){  
        digitalWrite(trigPin, LOW);  
        delayMicroseconds(2);  
  
        // Sets the trigPin on HIGH state for 10 micro seconds  
        digitalWrite(trigPin, HIGH);  
        delayMicroseconds(10);  
        digitalWrite(trigPin, LOW);  
        digitalWrite(ledPin, LOW);  
        digitalWrite(ledPin2, HIGH);  
    }  
}
```

Archivo Edición Formato Ver Ayuda

```
// Reads the echoPin, returns the sound wave travel time in microseconds
duration = pulseIn(echoPin, HIGH);

// Calculating the distance
distance= duration*0.034/2;

safetyDistance = distance;
if (safetyDistance <= 60 ){

    delay(1000);
    tiempo++;
}
if(tiempo==30){
    digitalWrite(buzzer, HIGH);
    digitalWrite(ledPin, HIGH);
}
else if (digitalRead(BOTONP) == HIGH)
{
    digitalWrite(buzzer, LOW);
    digitalWrite(ledPin, LOW);
    digitalWrite(trigPin, LOW);
    digitalWrite(echoPin, LOW);
    digitalWrite(ledPinP, LOW);

    (state2 = 0);

}

// Prints the distance on the Serial Monitor
Serial.print("Distance: ");
Serial.println(distance);
}
}
```

```
Archivo Edición Formato Ver Ayuda
if (digitalRead (BOTON) == LOW && digitalRead (BOTON2) == LOW && digitalRead (BOTON3) == HIGH && digitalRead (BOTONS) == LOW && digitalRead (BOTOMP) == LOW )
  val3= digitalRead(BOTON3);
{
  // Lee el estado del boton
  // chequear si el boton esta presionado o no
  if ((val3 == HIGH) && (old_val3 == LOW) ) {
    state3=1-state3;
    delay(10);
  }
  old_val3 = val3; // val is now old, let's store it
  if (state3==1){
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
  }
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  digitalWrite(ledPin, LOW);
  digitalWrite(ledPin3, HIGH);
}
// Reads the echoPin, returns the sound wave travel time in microseconds
duration = pulseIn(echoPin, HIGH);
// Calculating the distance
distance= duration*0.034/2;
safetyDistance = distance;
if (safetyDistance <= 90 ){
  delay(1000);
  tiempo++;
}
if(tiempo==30){
  digitalWrite(buzzer, HIGH);
  digitalWrite(ledPin, HIGH);
}
<
```

```
Archivo Edición Formato Ver Ayuda
}
else if (digitalRead(BOTONP) == HIGH)
{
digitalWrite(buzzer, LOW);
digitalWrite(ledPin, LOW);
digitalWrite(trigPin, LOW);
digitalWrite(echoPin, LOW);
digitalWrite(ledPin, LOW);
(state3 = 0);
}

// Prints the distance on the Serial Monitor
Serial.print("Distance: ");
Serial.println(distance);
}
}

if (digitalRead (BOTON) == HIGH && digitalRead (BOTON2) == LOW && digitalRead (BOTON3) == LOW && digitalRead (BOTONS) == HIGH && digitalRead (BOTONP) == LOW )
{
val15 = digitalRead(BOTON);

// lee el estado del Boton
// chequear si el boton esta presionado o no
if ((val15 == HIGH) && (old_val15 == LOW)) {
state5 =1-state5;
delay(10);
}
// val is now old, let's store it
old_val15 = val15;
}
```

```
Archivo Edición Formato Ver Ayuda
old_val5 = val5;|
if (state5==1){
digitalWrite(trigpin, LOW);
delayMicroseconds(3);

// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigpin, HIGH);
delayMicroseconds(11);
digitalWrite(trigpin, LOW);
digitalWrite(ledPinP, LOW);
digitalWrite(ledPinS, HIGH);
digitalWrite(ledPin1, HIGH);

// Reads the echoPin, returns the sound wave travel time in microseconds
duratioN = pulseIn(echopin, HIGH);

// Calculating the distance
distance2= duratioN*0.034/2;

safetyDistance2 = distance2;
if (safetyDistance2 <= 30){
delay(1000);
tiempo++;
}
if(tiempo==30){
digitalWrite(buzzer, HIGH);
digitalWrite(ledPin, HIGH);
}
else if (digitalRead(BOTONP) == HIGH)
{
digitalWrite(buzzer, LOW);
digitalWrite(ledPin, LOW);
digitalWrite(trigpin, LOW);
digitalWrite(echopin, LOW);
digitalWrite(ledPinP, LOW);

(state5 = 0);
```

<

```
Archivo Edición Formato Ver Ayuda
(states = 0);
}

// Prints the distance on the Serial Monitor
Serial.print("Distance: ");
Serial.println(distance);
}
}

if (digitalRead (BOTON) == LOW && digitalRead (BOTON2) == HIGH && digitalRead (BOTON3) == LOW && digitalRead (BOTON5) == HIGH && digitalRead (BOTON6) == LOW )
    val6 = digitalRead(BOTON2);
{

    // lee el estado del boton
    // chequear si el boton esta presionado o no
    if ((val6 == HIGH) && (old_val6 == LOW)) {

        state6 = !-state6;
        delay(10);
    }

    // val is now old, let's store it
    old_val6 = val6;
    if (state6==1){
        digitalWrite(trigpin, LOW);
        delayMicroseconds(3);

        // Sets the trigPin on HIGH state for 10 micro seconds
        digitalWrite(trigpin, HIGH);
        delayMicroseconds(11);
        digitalWrite(trigpin, LOW);
        digitalWrite(ledPin, LOW);
        digitalWrite(ledPin, HIGH);
        digitalWrite(ledPin, HIGH);
    }

    // Reads the echoPin, returns the sound wave travel time in microseconds
```

```
Archivo Edición Formato Ver Ayuda
// Reads the echoPin, returns the sound wave travel time in microseconds
duration = pulseIn(echoPin, HIGH);

// Calculating the distance
distance2= duration*0.034/2;

safetyDistance2 = distance2;
if (safetyDistance2 <= 60){
  delay(1000);
  tiempo++;
}
if(tiempo==30){
  digitalWrite(buzzer, HIGH);
  digitalWrite(ledPin, HIGH);
}
else if (digitalRead(BOTONP) == HIGH)
{
  digitalWrite(buzzer, LOW);
  digitalWrite(ledPin, LOW);
  digitalWrite(trigpin, LOW);
  digitalWrite(echoPin, LOW);
  digitalWrite(ledPin, LOW);
  (state6 = 0);
}

// Prints the distance on the Serial Monitor
Serial.print("Distance2: ");
Serial.println(distance2);
}

}

if (digitalRead (BOTON) == LOW && digitalRead (BOTON2) == LOW && digitalRead (BOTON3) == HIGH && digitalRead (BOTONS) == HIGH && digitalRead (BOTONP) == LOW )
{
  val17 = digitalRead(BOTON3);
}
```

```
Archivo Edición Formato Ver Ayuda
{

  // lee el estado del Boton
  // chequear si el boton esta presionado o no
  if ((val7 == HIGH) && (old_val7 == LOW)) {

    state7 =1-state7;
    delay(10);
  }
  // val is now old, let's store it
  old_val7 = val7;
  if (state7==1){
    digitalWrite(trigpin, LOW);
    delayMicroseconds(3);

    // Sets the trigPin on HIGH state for 10 micro seconds
    digitalWrite(trigpin, HIGH);
    delayMicroseconds(11);
    digitalWrite(trigpin, LOW);
    digitalWrite(ledPinP, LOW);
    digitalWrite(ledPinS, HIGH);
    digitalWrite(ledPin3, HIGH);

    // Reads the echoPin, returns the sound wave travel time in microseconds
    duratioN = pulseIn(echopin, HIGH);

    // Calculating the distance
    distance2= duratioN*0.034/2;

    safetyDistance2 = distance2;
    if (safetyDistance2 <= 90){
      delay(1000);
      tiempo++;
    }
  }
  if(tiempo==30){
    digitalWrite(buzzer, HIGH);
    digitalWrite(ledPin, HIGH);
  }
}
```

```
Archivo Edición Formato Ver Ayuda
digitalWrite(ledPin3, HIGH);

// Reads the echoPin, returns the sound wave travel time in microseconds
duration = pulseIn(echopin, HIGH);

// Calculating the distance
distance2= duration*0.034/2;

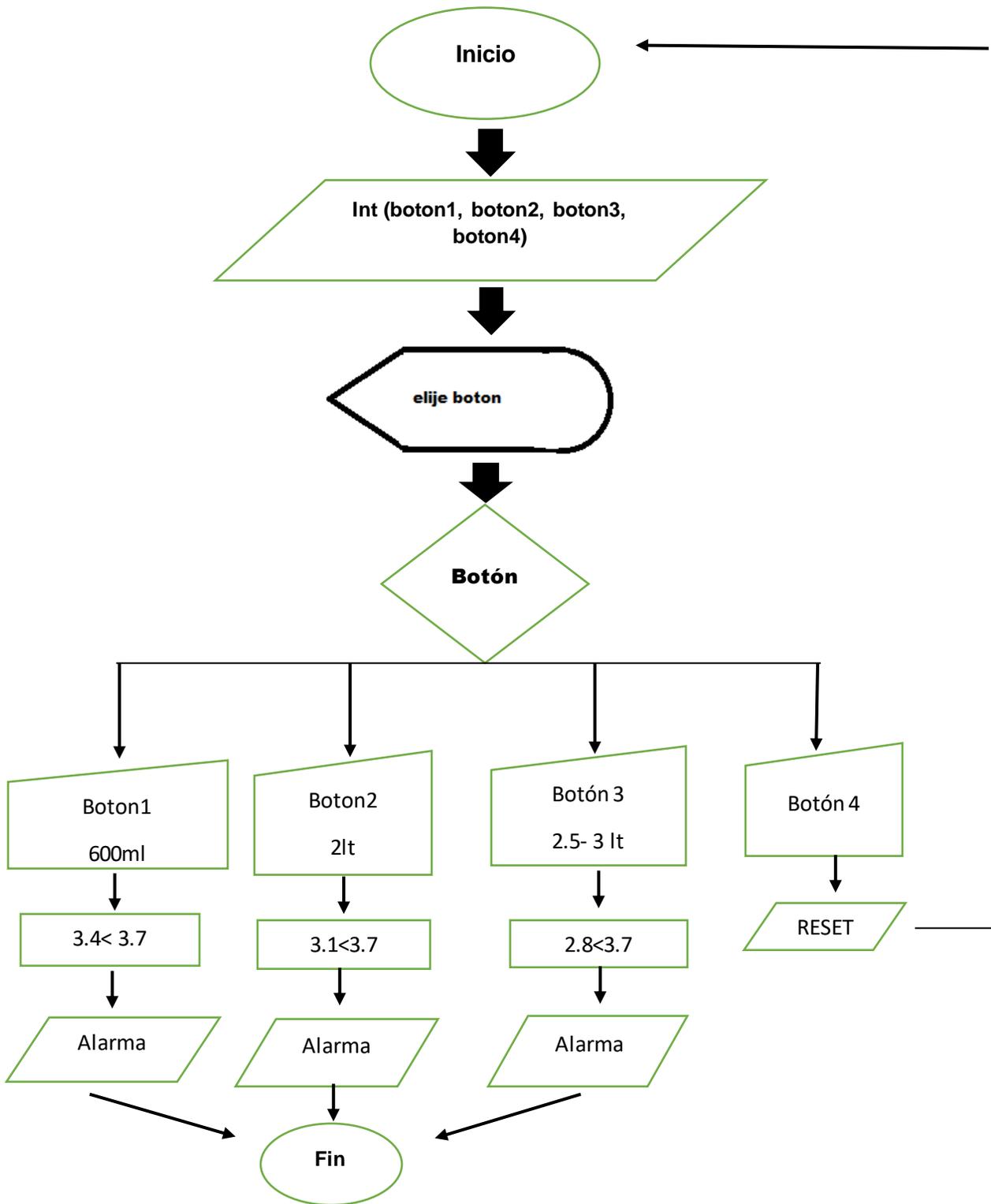
safetyDistance2 = distance2;
if (safetyDistance2 <= 90){
  delay(1000);
  tiempo++;
}
if(tiempo==30){
  digitalWrite(buzzer, HIGH);
  digitalWrite(ledPin, HIGH);
}
else if (digitalRead(BOTONP) == HIGH)
{
  digitalWrite(buzzer, LOW);
  digitalWrite(ledPin, LOW);
  digitalWrite(trigpin, LOW);
  digitalWrite(echopin, LOW);
  digitalWrite(ledPinP, LOW);

  (state7 = 0);
}

// Prints the distance on the Serial Monitor
Serial.print("Distance2: ");
Serial.println(distance2);
}
}
}
```

En la programación se hizo la estructura con la finalidad detectar diferentes distancias para dos sensores que se desea monitorear el control de llenado, ya que la empresa requiere que se controle distintas distancias para ambos silos que se encuentra teniendo en cuenta la misma estructura para los 2 silos.

6.3 Diagrama de flujo.



6.4 Símbolos del diagrama de flujo.



Inicio/ fin para indicar en donde empieza y termina el diagrama.



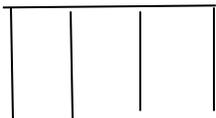
Entrada/salida, cualquier tipo de introducción de datos en la Memoria desde los periféricos “entrada” o registro de la información procesada en el periférico de “salida”.



proceso, operación para plantear instrucciones de asignación, tales como desarrollar una expresión o mover un dato de un lado.



Selección, múltiple en función del resultado de la comparación se asegura uno de los diferentes caminos de acuerdo con dicho resultado.



_____ Línea conectora, sirve de unión entre dos símbolos.



Teclado, entrada manual de datos desde un terminal o consola de computador.



Pantalla se utiliza en ocasiones en lugar de símbolos de ELS.



Figura 12 Caja de acrílico.

Aquí iniciamos con la fabricación del tablero, se hizo en material de acrílico por cuestión de inocuidad de la empresa.

Se aprecia 5 círculos donde ira colocado los botones, en la parte de un costado se encuentra un cuadro donde ira conectado el cable de salida.

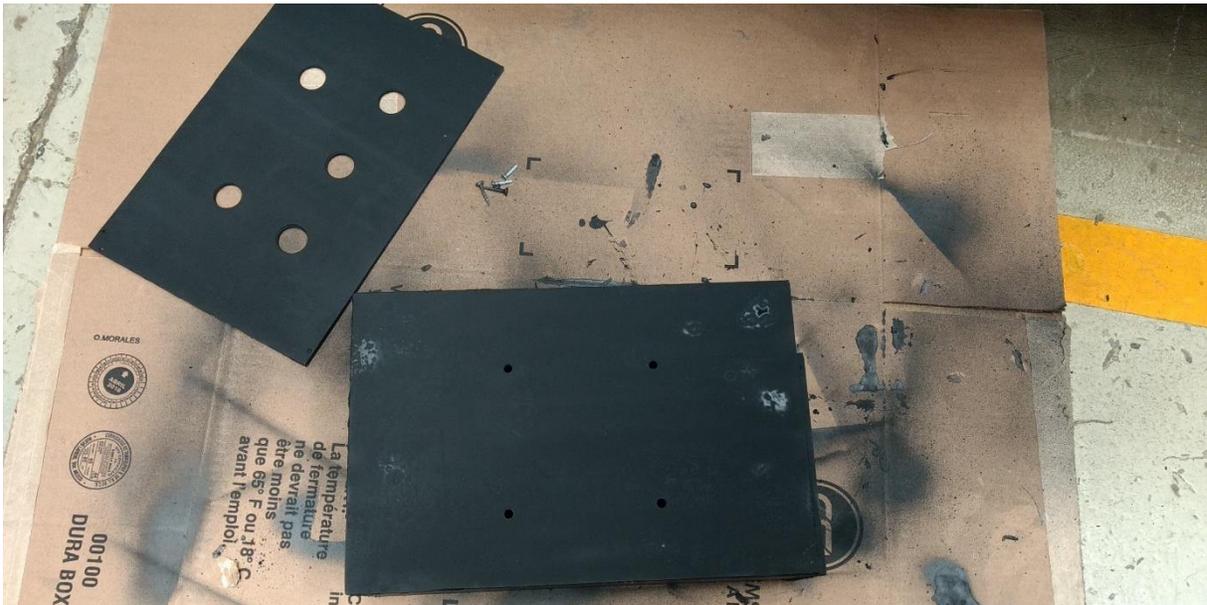


Figura 13 Se está pintando, para mejor estética.

Aquí se está pintando para mejor estética del proyecto.

6.5 Materiales que se utilizaron en la construcción del proyecto.



Figura 15 Sensor ultrasónico.



Figura 14 Relevador.



Figura 17 Conector rj45 hembra.



Figura 16 Resistencia 1/2 Kohm.



Figura 18 Cable utp.



Figura 19 Jumpers (macho).



Figura 20 Regulador de 5v.



Figura 21 Cargador de 9v.



Figura 23 Jumpers (hembra).



Figura 22 Plug hembra 3.5mm

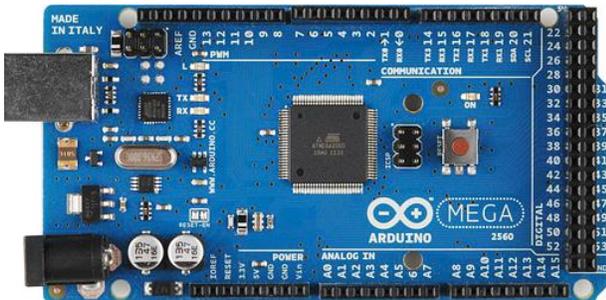


Figura 25 Placa arduino mega 250.



Figura 24 Torreta 12v.



Figura 27 Botones.



Figura 26 Sirena 12v



Figura 29 Plug macho 3mm.



Figura 28 Base de acrílico del sensor.



Figura 31 Conector Jack alimentación hembra.



Figura 30 Brazo o soporte del sensor ultrasónico.



Figura 32 Selector switch.

6.6 Esquema de la placa arduino mega.

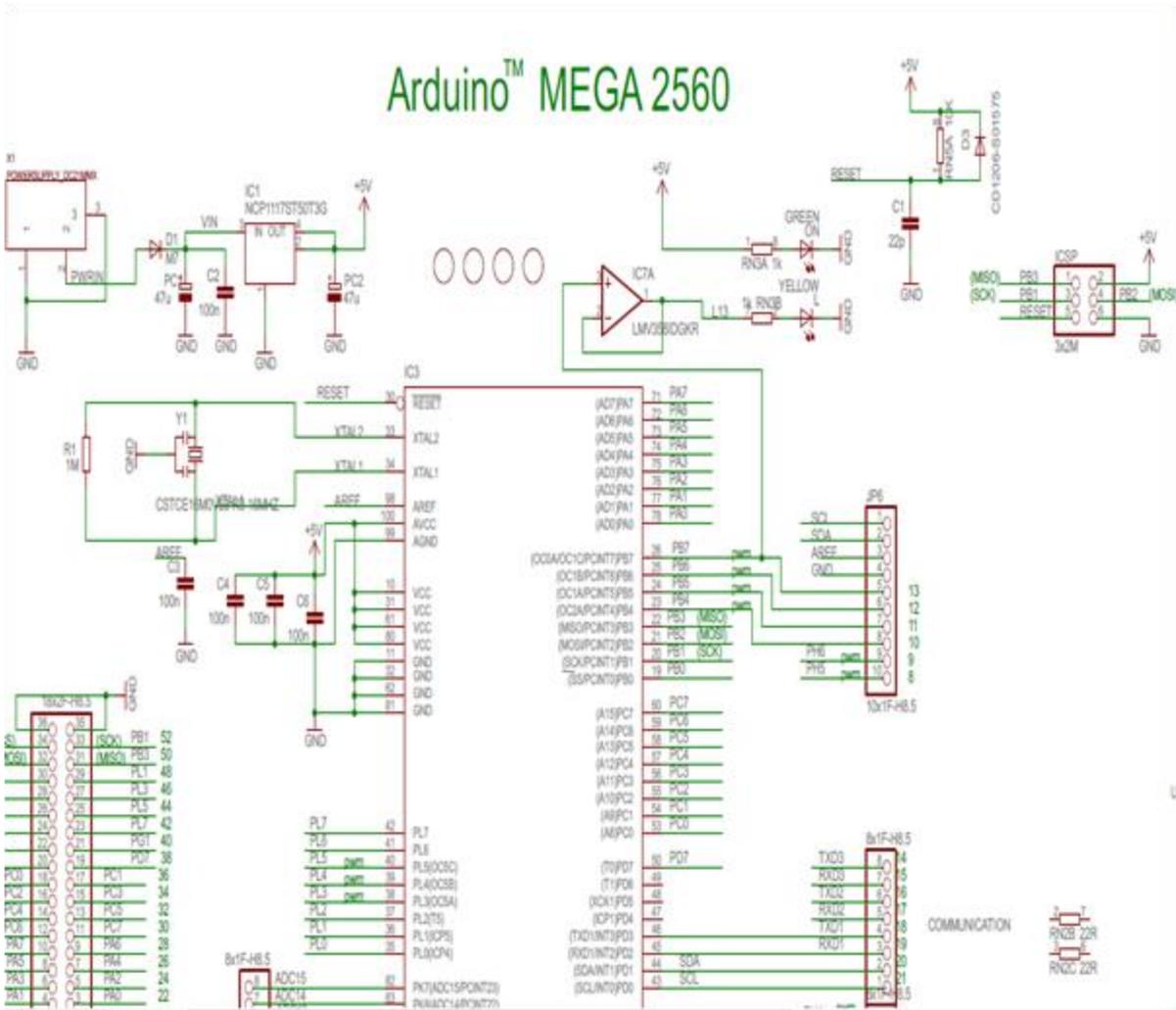


Figura 33 Esquema 1.

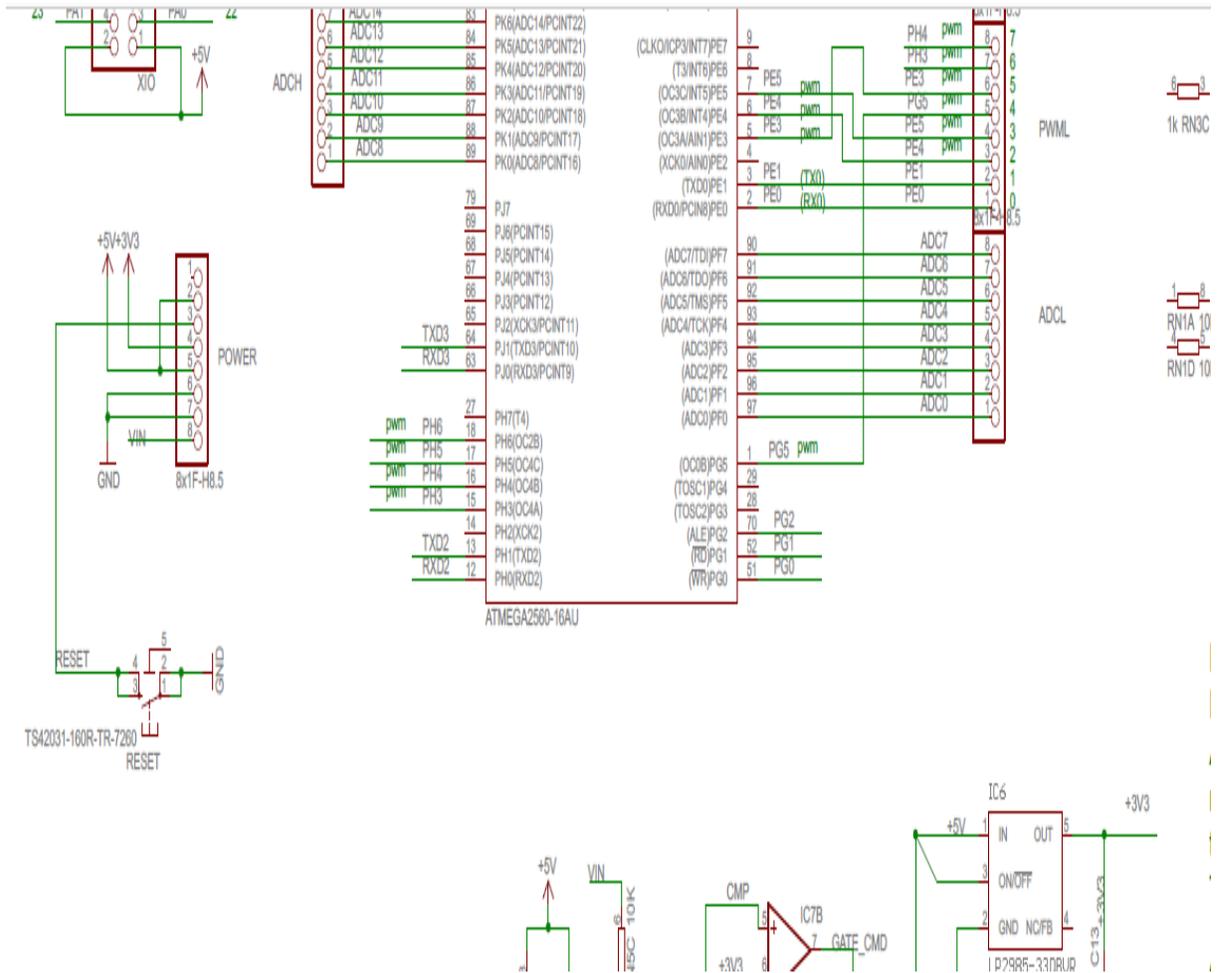


Figura 34 Esquema 2.

GND

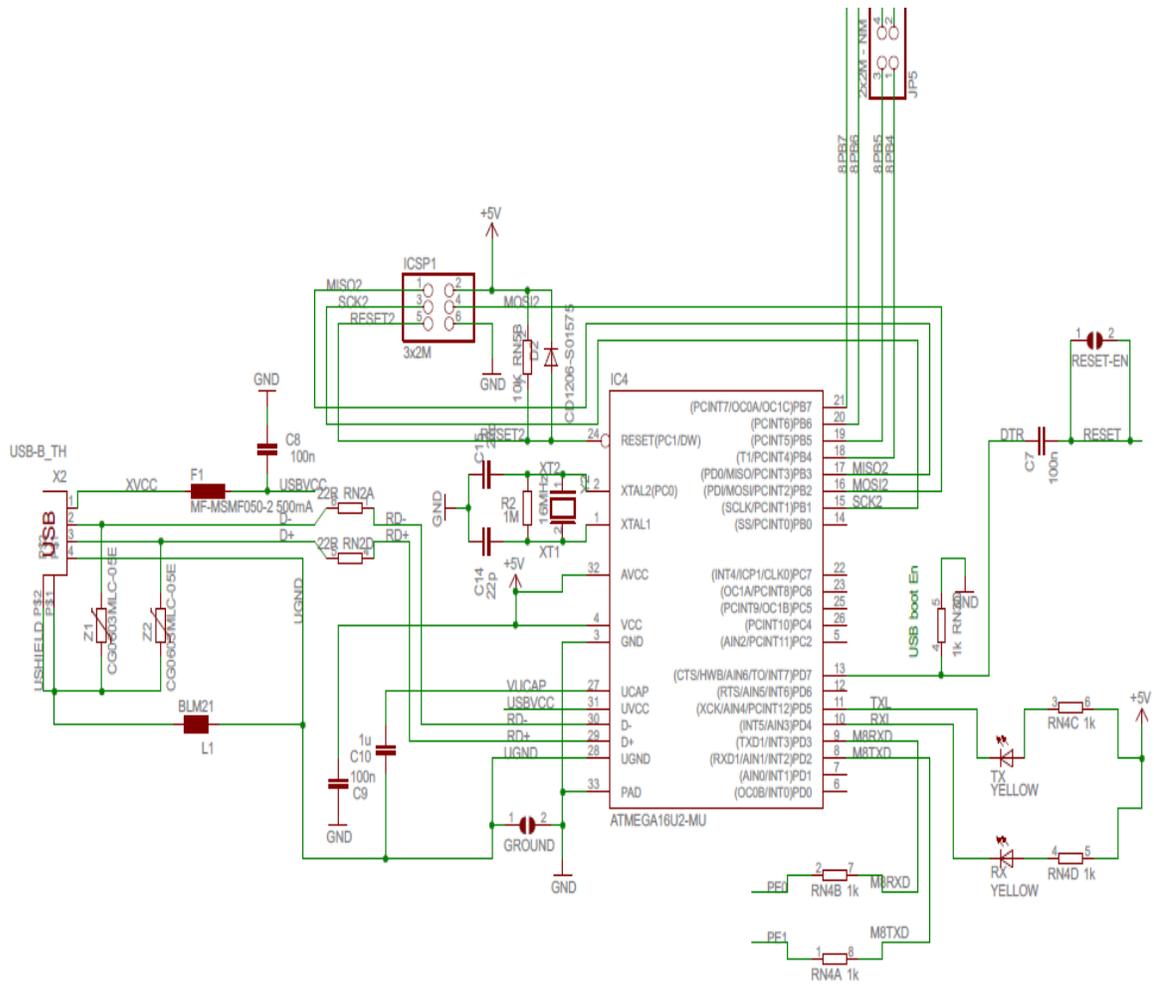


Figura 35 Esquema 3.

6.7 Diagrama eléctrico de conexiones.

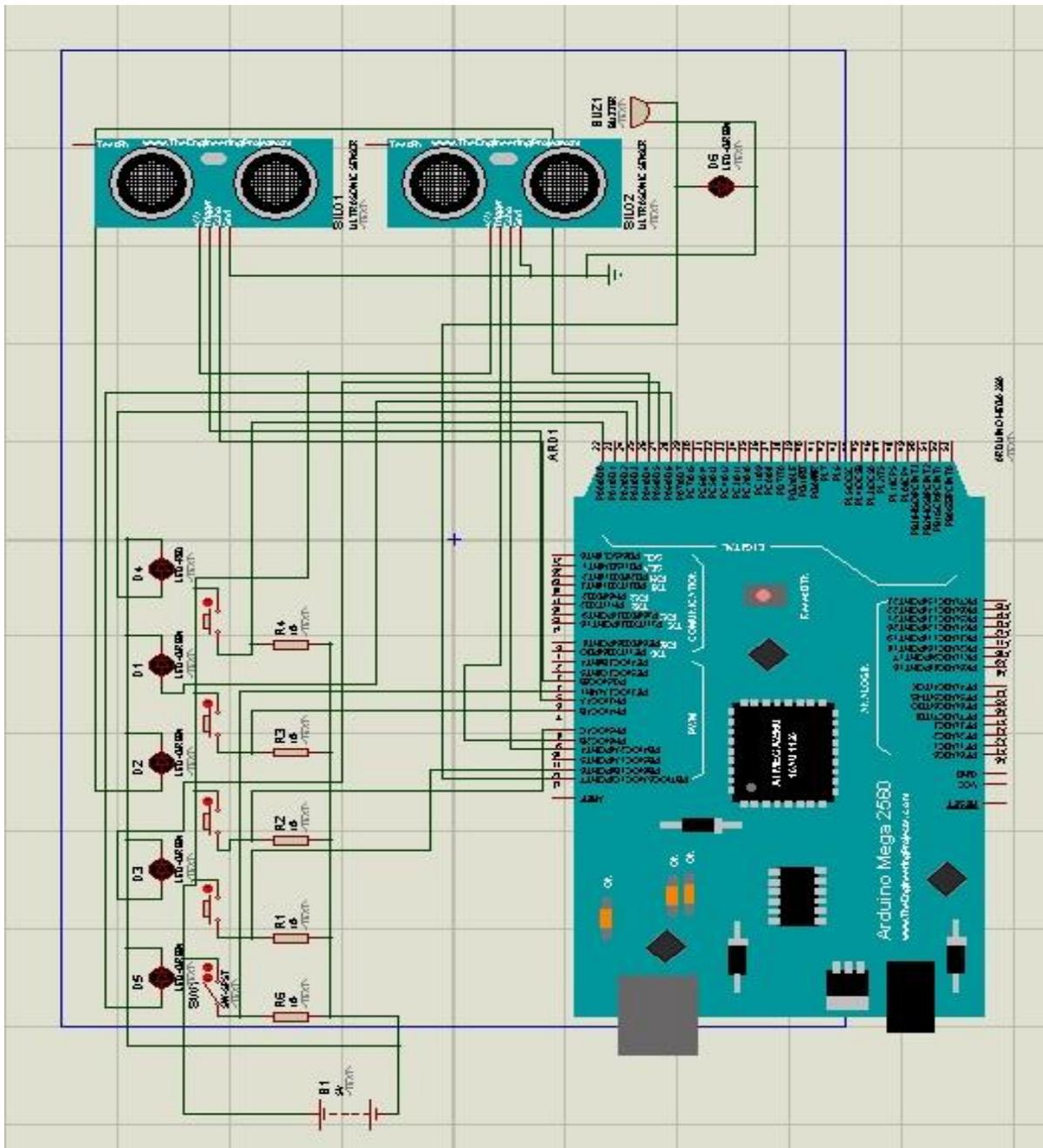


Figura 36 Diagrama de eléctrico de conexiones del proyecto.



Figura 37 Se están realizando pruebas.

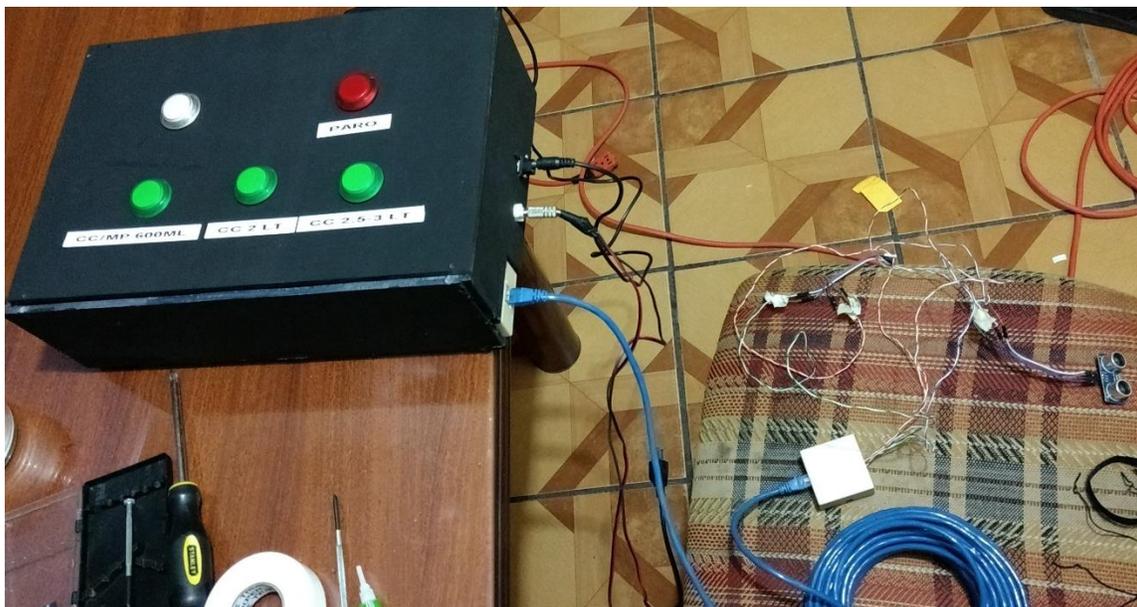


Figura 38 Aquí se realizan pruebas previas antes del montaje posterior.



Figura 39 Soporte colocado.



Figura 40 Ultrasonico unido al soporte.



Figura 41 Tablero del control de llenado de silos.

Está listo el tablero para usarse, se aprecias los botones verdes para la selección de acuerdo con la producción que está en corrida, también podemos ver que cuenta con un botón de paro y un selector para activar el silo un 1 o 2 según sea el caso.



Figura 42 En funcionamiento en corrida cc3.0 lt en posición silo 2.

Este etiquetado las conexiones externas para el uso del tablero.

6.8 Manual de uso de tablero de control de llenado de silos.

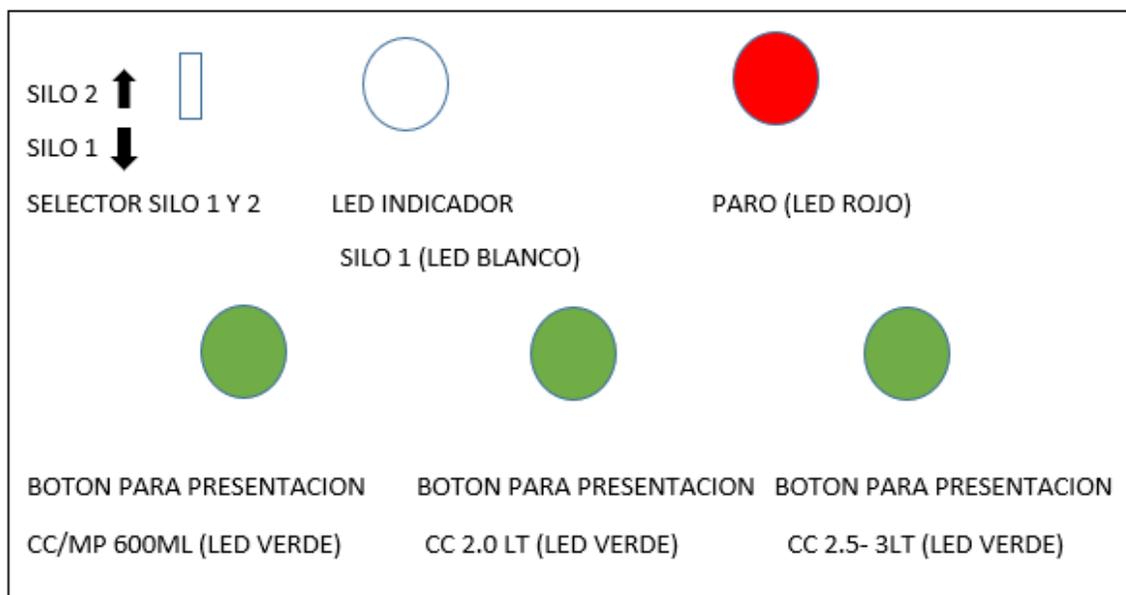


Figura 44 Manual de usuario del tablero de llenado de silos.

SELECTOR SILO 1 Y 2: ESTE SELECTOR NOS INDICA QUE SILO DESEAMOS DEJAR ACTIVADO, LO CUAL NOS INDICA QUE HACIA ARRIBA ES SILO 2 Y HACIA ABAJO SILO 1, TAMBIEN SE MENCIONA QUE AL ACTIVAR SILO 1 EL LED BLANCO QUEDA ACTIVADO Y TAMBIEN EL LED VERDE DE LA PRESENTACION QUE EL OPERADOR DESEA ELEGIR, EN CASO DE SILO 2 SOLO EL LED VERDE QUEDARA ENCENDIDO, PARA SU AVISO INMEDIATO CUANDO EL SILO SE ENCUENTRO LLENO.

PARO: ESTE BOTON SE UTILIZA PARA DETENER LA SIRENA Y TORRETA, PORQUE SE ALARMARÁ LA PRESENTACION QUE EL OPERADOR ELIGIO.

BOTONES PARA PRESENTACION (CC/MP 600ML, 2LT,2.5-3LT): ESTOS BOTONES NOS INDICA QUE PRESENTACION DESEA ACTIVAR DEACUERDO A LA PRODUCCION QUE ESTE EN CORRIDA.

NOTA: PARA CAMBIAR DE SILO 1 Y 2 PRIMERO SE PRESIONA EL BOTON DE PARO Y LUEGO EL SELECTOR SE MUEVE HACIA ARRIBA O HACIA ABAJO SEGÚN SEA EL CASO, PARA DESPUES ELEGIR CUALQUIERA DE LOS TRES BOTONES VERDES PARA LA SIGUIENTE PRESENTACION QUE REQUIERA MONITOREAR.

6.9 Cálculos de llenado de silos.

PRESENTACIONES PERSONALES	BOTELLAS MAXIMA POR SILO
MULTIPRODUCTO 600 ML	10,000
SPLASH 600 ML	8,000
SLIM 600 ML	5,000
OTG 600 ML	10,000

PRESENTACIONES PERSONALES	BOTELLAS MAXIMA POR SILO
CC 2000 ML	3,000
CC 2500 ML	3,000
CC 3000 ML	1,000

Figura 45 Análisis de llenado de diferentes presentaciones.

Aquí tenemos la cantidad máxima por silo, en las diferentes presentaciones que se aprecian.

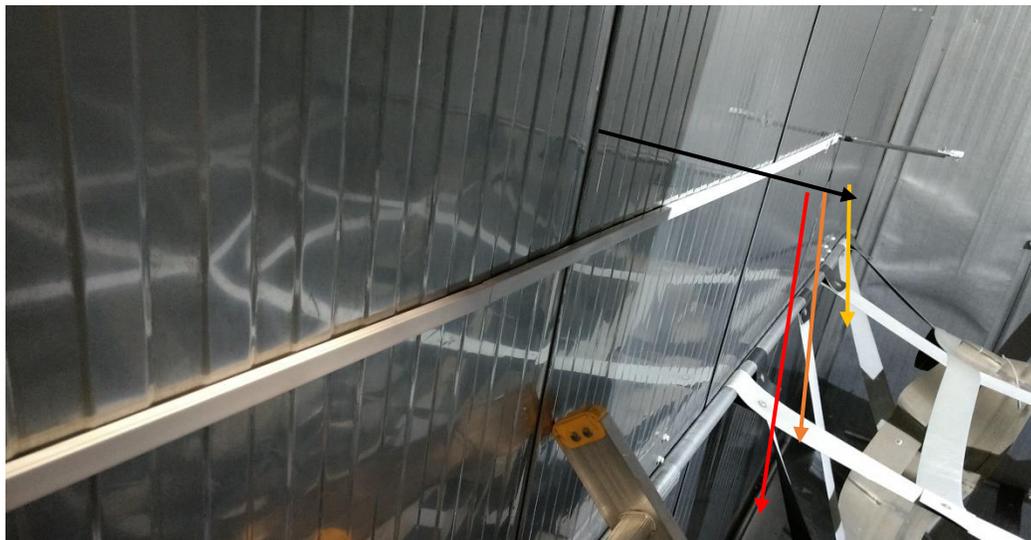


Figura 46 Las flechas de colores indica el nivel de llenado de cada presentación de botellas.

Para cada color son niveles de llenado de las diferentes tres presentaciones el color amarillo es cc/mp 600ml, color naranja cc2lt, color rojo cc2.5- 3lt.

Capítulo 7: Resultados.

7.1 Resultados de la plataforma labview.

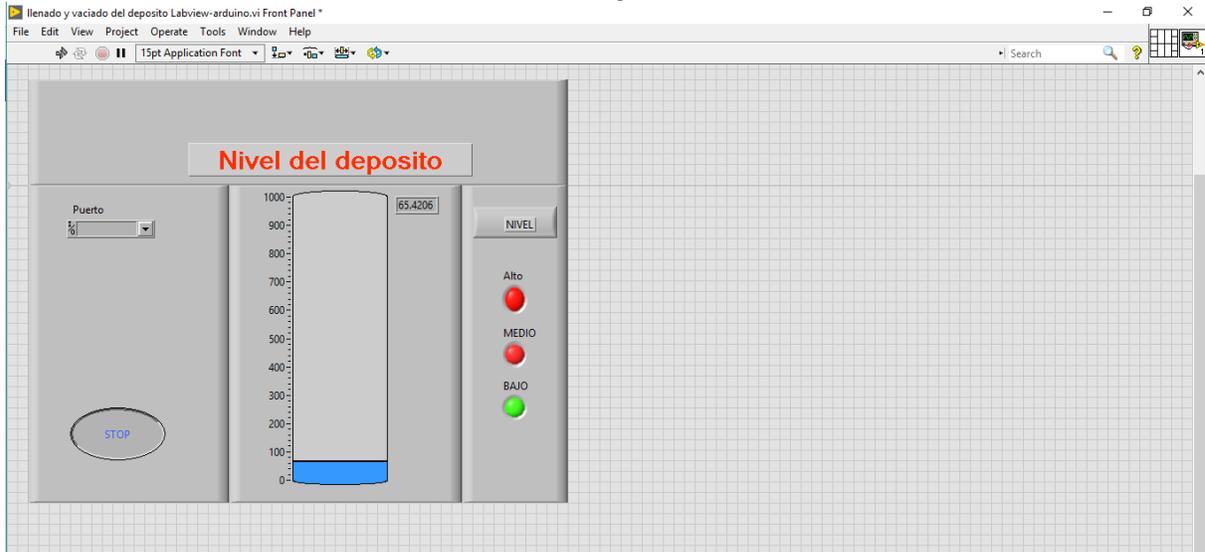


Figura 47 Nivel bajo de llenado.

Aquí vemos en el grafico del software como nos indica los tres niveles.

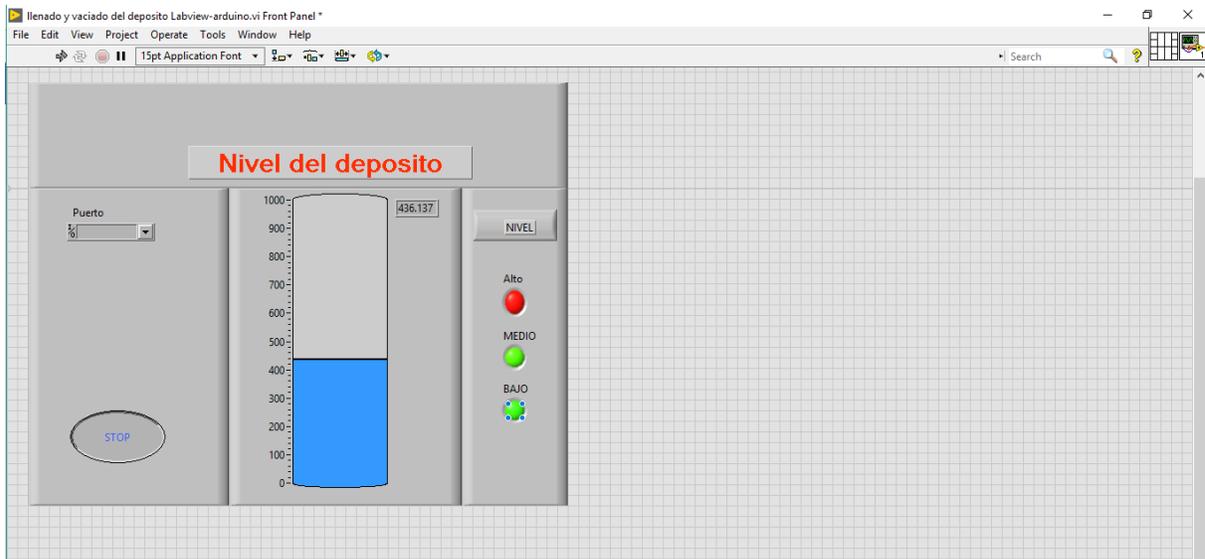


Figura 48 Nivel medio en un 50% de llenado.

Estamos monitoreando y vamos en un 50% de llenado en la visualización de software labview.

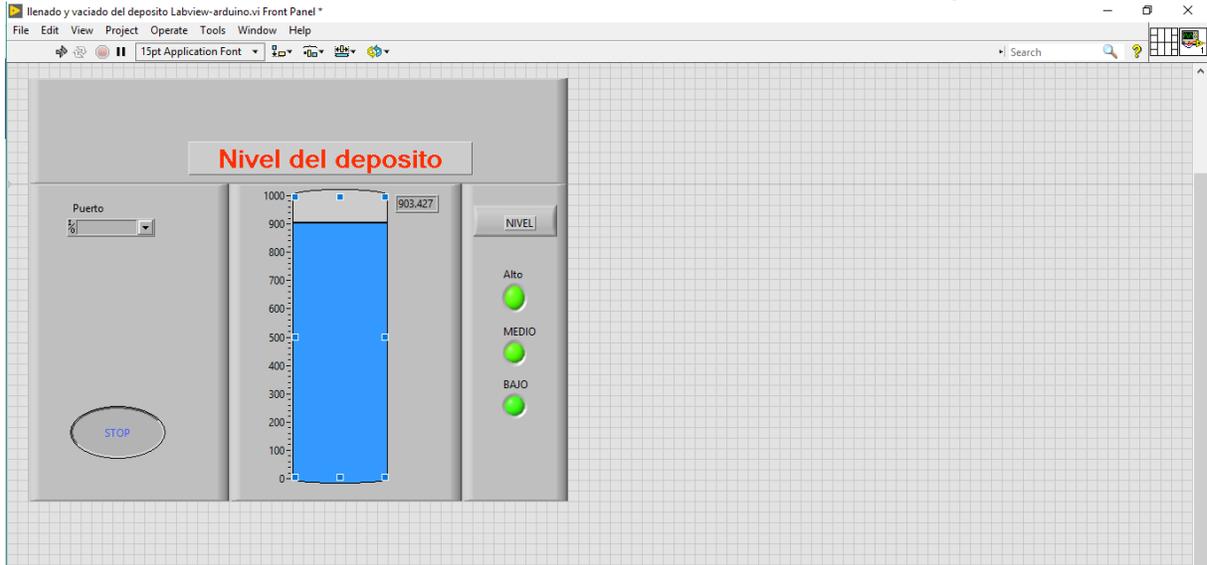


Figura 49 Nivel lleno de acuerdo con la presentación que eligió el operador.

Aquí podemos notar que los tres puntos están en color verde lo que indica nivel lleno, cuando esta en color rojo nos dice que todavía no ha llega al nivel deseado.

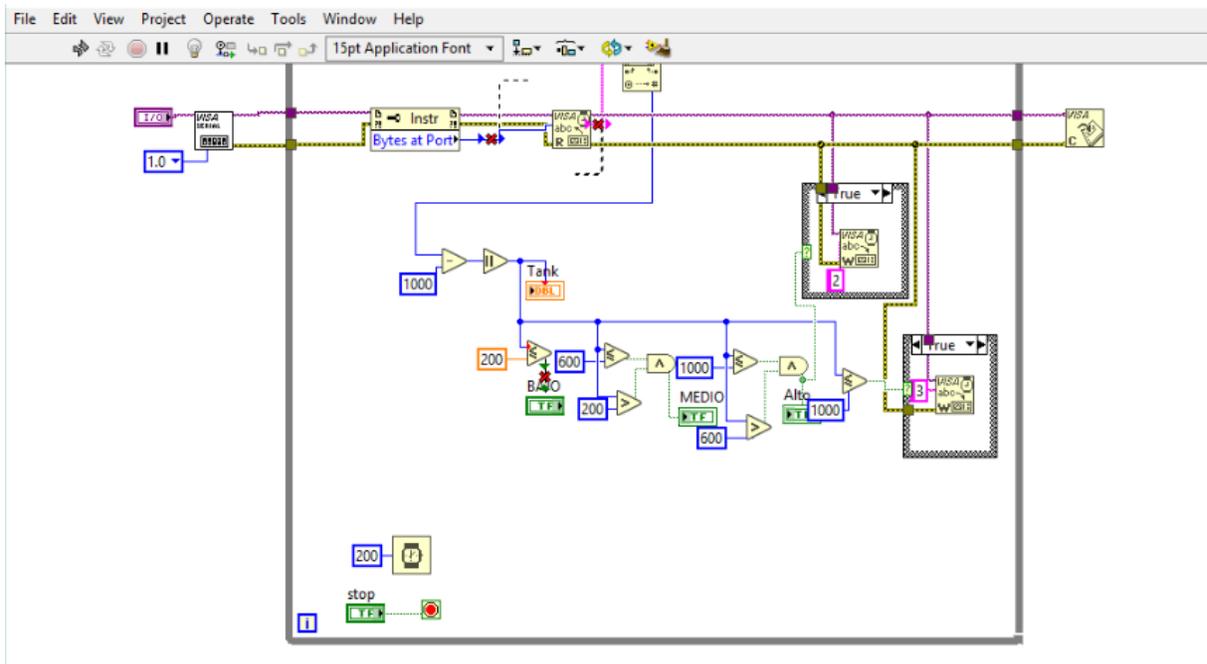


Figura 50 Diagrama de bloques de la plataforma labview.

Aquí vemos las conexiones y de los componentes que se agregaron para el diseño del circuito para que se pueda ver en pantalla.

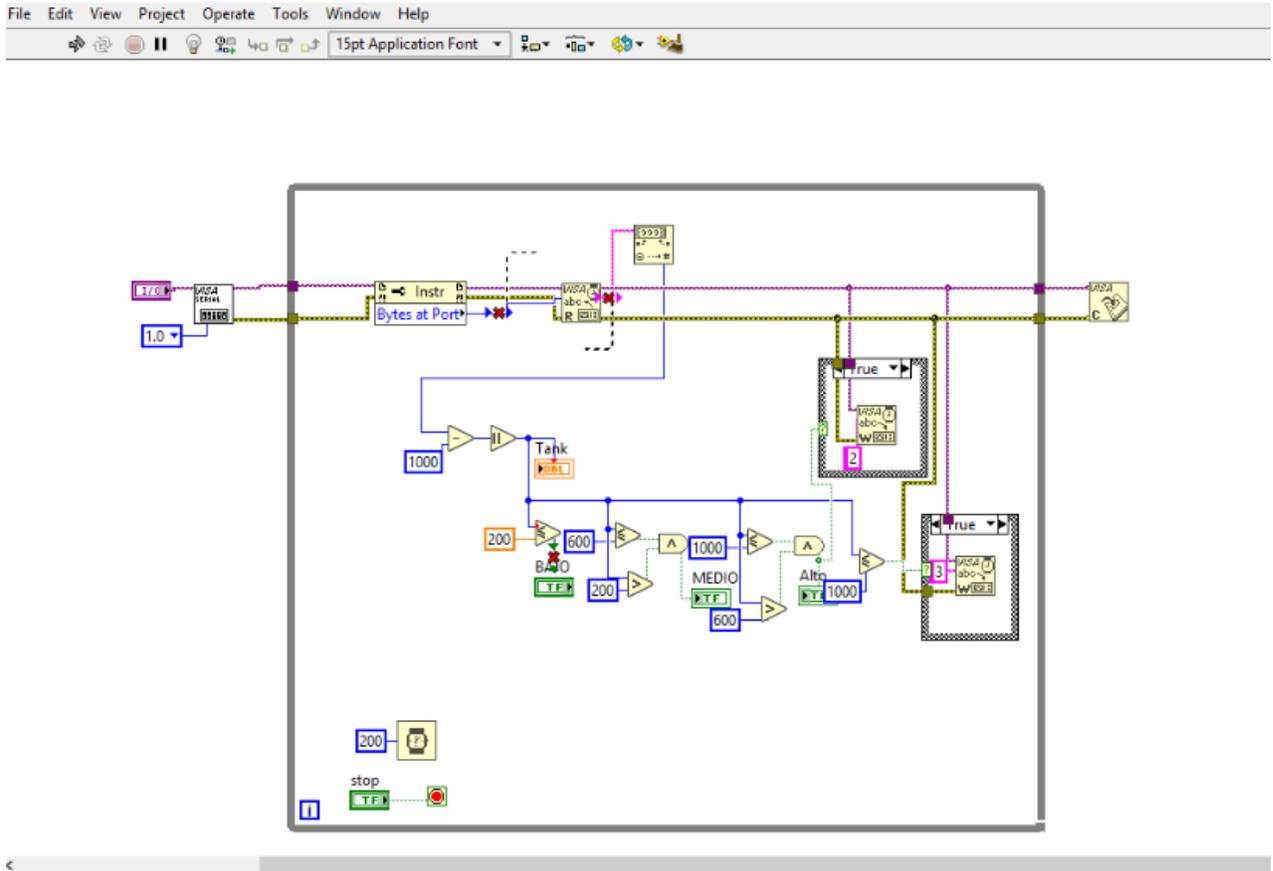


Figura 51 Esquema de diagrama de bloques.

En el diagrama de bloque podemos observar la construcción de los tres niveles de que desea monitorear.

Capítulo 8: Conclusiones y recomendaciones.

8.1 CONCLUSIONES.

Este proyecto a lo largo de toda esta investigación se logró desarrollar donde demuestra que el sistema automatizado de llenado de silos resulta ser efectivo en el abastecimiento de botellas para el cliente que lo requiere con mayor exactitud, así mismo que el operador le sea fácil manipular el sistema con solo pulsar el botón de la presentación que sea monitorear. Y en base a lo mostrado a lo largo del desarrollo del presente informe, podemos resaltar los siguientes puntos:

1. Que este tipo de proyecto puede ser aplicado en cualquier otra empresa que requiera de un sistema automatizado del llenado de contenedores y que requiera de aumentar su producción en menor tiempo y con mayor exactitud.
2. Con la implementación de los dispositivos propuestos para este proyecto como lo son sensores de nivel, torreta, sirena y el aviso automático cuando este en el nivel deseado, se garantiza que no habrá merma de botellas en la planta.
3. La ayuda y consejos de los asesores influyo a concluir satisfactoriamente el trabajo realizado.
4. El uso de las herramientas de diseño del software LabVIEW sistema scada, facilitó el buen desarrollo de las piezas mostradas en el apartado de resultados .

8.2 Recomendaciones.

- 1- Se recomienda el uso de dispositivos de hardware especializados como son los PLC (Controladores lógicos Programables), estos dispositivos pueden mejorar la adquisición de datos, puesto que en su arquitectura ellos manejan los datos en tiempo real y son aparatos exclusivamente dedicados, pero todo bajo su entorno de programación.
- 2- Para censar niveles de envases se recomienda utilizar un sensor de mayor jerarquía, estos sensores pueden ser: sensor de presión, sensor capacitivo, sensor laser, ellos tienen características importantes como es la precisión de los datos de los niveles de altura, pero se requiere de una inversión considerable.
- 3- Al realizar diseños de cualquier índole es importante prestar atención a los pequeños detalles, ya que en ocasiones son estos los que nos dan mayor problema.

Bibliografía.

1. Gassó, (. M. (s.f.). *Transductores de ultrasonidos (José Mº Lorente Gassó)*.
2. Jiménez Builes, J. A. (02 de junio de 2008). *Sistema Doble Umbral para el Incremento de la Directividad en un Sensor Ultrasónico*, 7.
3. Quiñones, C. (2010). *LabVIEW y la instrumentación virtual aplicados*, 7.
4. Wheat, D. (2011). *arduino internals*.
5. https://www.google.com.mx/search?q=torreta+roja&source=lnms&tbm=isch&sa=X&ved=2ahUKEwiKv_KZqeDbAhVbljQIHbpSCEkQ_AUoAXoECAEQAw&biw=1366&bih=637#imgrc=RzBjkMMpMG7W5M:
6. https://www.google.com.mx/search?rlz=1C1CHBD_esMX767MX767&biw=1366&bih=637&tbm=isch&sa=1&ei=Q0cpW-q3HoyT0PEP1Zq-mAM&q=sirena+de+12v&oq=sirena+d&gs_l=img.3.0.35i39k1j0i67k1j0l4j0i67k1j0l3.3554.5587.0.7457.2.2.0.0.0.368.536.0j1j0j1.2.0...0...1c.1.64.img..0.2.528...0.liWCqA75HCw#imgrc=vHkiDkf719EPXM:
7. https://www.google.com.mx/search?q=arduino+mega&sa=X&rlz=1C1CHBD_esMX767MX767&tbm=isch&tbo=u&source=univ&ved=2ahUKEwjBliv9qeDbAhWUCDQIHS5zD4UQsAR6BAgBEEI&biw=1366&bih=637#imgrc=JOuWaE2NyZl3nM:
8. https://www.researchgate.net/profile/Jovani_Jimenez_Builes/publication/220136828_Sistema_Doble_Umbral_para_el_Incremento_de_la_Directividad_en_un_Sensor_Ultrasonico/links/0046352ab8e049831e00000/Sistema-Doble-Umbral-para-el-Incremento-de-la-Directividad-en-un-Sensor-Ultrasonico.pdf
9. <https://www.robotshop.com/media/files/pdf/arduinomega2560datasheet.pdf>