UNIVERSIDAD TECNOLÓGICA ISRAEL

TRABAJO DE TITULACIÓN EN OPCIÓN AL TÍTULO DE:

“INGENIERA EN ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES”

TEMA:

PROTOTIPO DE DESPENSA INTELIGENTE PARA CONTROLAR EL PESO Y TIEMPO MÁXIMO DE CONSUMO DE VÍVERES EN UN SUPERMERCADO

AUTOR: WENDY VIVIANA ACHIG OCHOA

TUTOR: Mg. SILVIA DIANA MARTINEZ MOSQUERA

TUTOR TÉCNICO: Mg. FLAVIO DAVID MORALES ARÉVALO

Ph.D FIDEL DAVID PARRA BALZA

AÑO: 2017

DECLARACIÓN

Yo, Wendy Viviana Achig Ochoa, declaro que el trabajo aquí descrito es de mi autoría, que no ha sido previamente presentado por ningún grado o calificación profesional, y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Universidad Tecnológica Israel, puede hacer uso de los derechos correspondientes a este trabajo según lo establecido en su reglamento y por la normatividad institucional vigente.

_____

Wendy Viviana Achig Ochoa

UNIVERSIDAD TECNOLÓGICA ISRAEL

APROBACIÓN DEL AUTOR

En mi calidad de tutor del trabajo de titulación certifico:

Que el trabajo de graduación "PROTOTIPO DE DESPENSA INTELIGENTE PARA CONTROLAR EL PESO Y TIEMPO MÁXIMO DE CONSUMO DE VÍVERES EN UN SUPERMERCADO." presentado por la Srta. Wendy Viviana Achig Ochoa, estudiante de la Carrera de Electrónica Digital y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación

Quito D. M., agosto de 2017

TUTORA

_____

Ing. Silvia Diana Martinez Mosquera, Mg

AGRADECIMIENTO

Mi agradecimiento principal a Dios por ser la luz en mi camino, guiarme y darme la oportunidad de volver a retomar mis estudios a pesar de todos los inconvenientes que se presentaron, se que siempre está a mi lado para darme la mano en los momentos difíciles.

A mis padres porque han sido los ejes fundamentales de mi vida, que siempre han estado a mi lado para apoyarme en todo lo que han podido, brindándome su apoyo, comprensión e impulsándome a mejorar para ser mejor cada día.

A mi hermana Susana, que indirectamente a estado apoyándome, brindándome el cuidado de mis pequeños mientras duró este largo proceso de aprendizaje lo que me ayudó a estar tranquila con la confianza de que estaban en las mejores manos en mi ausencia.

A mis mejores amigas Jacqueline y Norma que son las personas que siempre estuvieron a mi lado apoyando, enseñándome y brindándome sus conocimientos para poder salir adelante en este reto que nos pusimos juntas y que ahora está dando frutos.

A mis profesores de la Universidad Tecnológica Israel que fueron la fuente principal de conocimientos, ya que supieron apoyar a sus estudiantes para que salgan adelante brindando sus mejores técnicas de enseñanza.

DEDICATORIA

Con todo mi corazón, dedico mi trabajo y logro alcanzado a mi familia que son: mis hijas Daniela y Camila, mis hijos Mario y David y a mi compañero de toda la vida Luis, quienes supieron apoyarme estando a mi lado en los buenos y malos momentos, siendo el pilar de mi vida y motivación para seguir adelante en los momentos más difíciles y que ahora los frutos de este esfuerzo los vamos a cosechar juntos cumpliendo una de las muchas metas trazadas. Quiero que sepan que todo esto nos brindará la oportunidad de obtener un mejor futuro para nuestra hermosa familia.

Tabla de Contenido

LISTA DE FIGURAS

LISTA DE TABLAS

LISTA DE ECUACIONES

RESUMEN

El presente documento trata de un prototipo de despensa inteligente que servirá para controlar el peso y tiempo máximo de consumo de víveres en un supermercado. La implementación está fundamentada en una combinación de *software y hardware* que interactúan entre sí.

Utilizando el método de análisis y síntesis, se detalla una fundamentación teórica en donde se dan a conocer las características principales de los dispositivos electrónicos utilizados para la elaboración del prototipo, la propuesta diseñada y la implementación realizada para alcanzar los objetivos propuestos.

Basado en el método Deductivo/ Inductivo se realiza el diseño del prototipo y adicionalmente, se explica la funcionalidad principal que es la generación de alertas visuales y auditivas generadas mediante dispositivos electrónicos utilizados como *hardware* y la creación de un *software*.

Para realizar las pruebas de verificación y funcionamiento se utiliza el método experimental con lo que se realiza la comprobación de la funcionalidad correcta del prototipo, dando a conocer las conclusiones en referencia a los objetivos planteados, además de las recomendaciones que podrían ayudar a la mejora del prototipo.

Palabras clave: configuración de nombres**,** registro del producto, tiempo de caducidad, alerta auditiva, alerta visual

ABSTRACT

This document deals with a prototype smart pantry that will serve to control the weight and maximum time of consumption of food in a supermarket. The implementation is based on a combination of software and hardware that interact with each other.

Using the method of analysis and synthesis, a theoretical foundation is detailed where the main characteristics of the electronic devices used for the prototype, the designed proposal and the implementation made to reach the proposed objectives are presented.

Based on the Deductive / Inductive method, the design of the prototype is performed and, additionally, explains the main functionality that is the generation of visual and auditory alerts generated by electronic devices used as hardware and the creation of software.

In order to carry out the verification and operation tests, the experimental method is used, which verifies the correct functionality of the prototype, making known the conclusions in reference to the proposed objectives, besides the recommendations that could help the improvement of the prototype.

Keywords: name setting, product registration, expiration time, auditory alert, visual alert

INTRODUCCIÓN

En la actualidad existen algunas soluciones para saber cuándo su despensa está desbastecida, como por ejemplo el fabricante Samsung que lanzó al mercado el Samsung Family Hub, un frigorífico inteligente con pantalla digital. Una de las funciones del frigorífico es verificar mediante cámaras ubicadas en el interior, los alimentos disponibles, con el fin de que el usuario pueda conocer el consumo de los mismos y en base a esta verificación crear una lista para realizar su abastecimiento. Sin embargo, esta solución no es aplicable a supermercados ni tampoco permite generar alertas automáticas al usuario cuando se han consumido los alimentos total o parcialmente.

La implementación de un prototipo de despensa inteligente ayuda a las personas que abastecen las despensas en los supermercados, a obtener alertas visuales y auditivas que les permita conocer cuándo y dónde colocar el producto desabastecido.

Planteamiento del problema

En la actualidad en los supermercados no cuentan con un sistema de almacenamiento que pueda generar alertas a las personas que realizan el abastecimiento de los productos en las perchas lo que puede ocasionar que en algún momento queden desabastecidas, generando malestar en los clientes y demoras en el surtimiento.

En este proyecto se implementará un prototipo de despensa inteligente que mediante ayudas visuales y auditivas ayudará de manera oportuna a los empleados que realizan el abastecimiento a saber que producto está agotado o a la vez, si su fecha de expiración esta caducada con lo que se evitara tener víveres en mal estado y adicionalmente se permitirá abastecer de manera inmediata los productos agotados.

La estructura de este prototipo será diseñado en madera de 25 cm de largo y 29 cm de ancho, permitiendo el almacenamiento de hasta cuatro productos similares o diferentes con un peso máximo de 20 kg.

Justificación:

El abastecimiento de los productos en los supermercados actualmente se los realiza verificando las perchas una vez que los productos se agotaron o mediante un chequeo por parte de las personas que realizan el suministro de los productos lo que puede provocar que se desabastezcan si la inspección no se la realizó por todas las áreas.

La elaboración del prototipo de despensa inteligente con alertas visuales y auditivas permitirá a las personas que distribuyen los productos en los supermercados saber exactamente el tipo de producto agotado, caducado y ubicación donde debe abastecerse permitiendo mejorar el suministro de productos al conocer exactamente los productos que se necesitan volver a colocar.

Este sistema de alerta ayudará a mejorar el surtimiento de productos evitando tener gavetas vacías o con producto caducado que provoque una mala atención al cliente final.

Objetivo general:

Implementar un prototipo de despensa inteligente para controlar la presencia del producto a ser almacenado y tiempo máximo de consumo de víveres en un supermercado.

Objetivos específicos:

1. Diseñar un prototipo de despensa inteligente para controlar el abastecimiento de víveres en un supermercado, mediante el control de las variables de peso y tiempo, a través del uso de celdas de carga, registros por lotes y una placa Arduino.
2. Implementar un *software* que permita visualizar las alertas del peso y el tiempo en un computador que estará conectado vía serial a la placa Arduino.
3. Construir el prototipo de despensa inteligente diseñado con comunicación a la aplicación para visualizar las alertas.
4. Realizar pruebas de validación y funcionamiento del prototipo.

Descripción de capítulos

El trabajo está estructurado de la siguiente manera. En el capítulo I se presenta el marco teórico donde se incluyen de cada uno de los componentes utilizados para la elaboración del prototipo los conceptos, funcionamiento y características. En el capítulo II se detalla, la propuesta a implementarse tanto a nivel de *hardware* como por ejemplo componentes, circuitos, diagramas de flujo y *software,* programa, funcionamiento. En el capítulo III, se detalla el desarrollo del prototipo placas, implementación, pruebas de funcionamiento y análisis de resultados obtenidos de las pruebas de funcionamiento realizadas.

Al final se detallan las conclusiones y recomendaciones obtenidas luego de la elaboración del presente prototipo.

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

1.1 Marco Teórico

Una despensa inteligente tiene la función de almacenar los productos y adicionalmente, contar con un sistema de alerta auditivo y visual que ayuda a la persona que realiza el abastecimiento de los alimentos a conocer cuando el producto está agotado o terminó su tiempo de almacenamiento.

Actualmente, en los supermercados, las despensas son utilizadas únicamente como medio para almacenar sus productos y ponerlos a disposición para los usuarios.

Para el diseño del presente prototipo se detallan materiales que son comúnmente utilizados en su fabricación:

- Celdas de carga.
- Trasmisor de celda de carga HX711.
- Placa Arduino Mega T2560.
- Diodo Led.
- Zumbador de 5V.

A continuación, se describen cada uno de los componentes mencionados:

1.1.1 Celdas de Carga

Son un tipo de transductor utilizado para convertir una fuerza en una señal eléctrica. La conversión se la realiza a partir de un dispositivo mecánico, es decir, la fuerza que se desea medir deforma una galga extensiométrica. (Haeussler, 2003).

La medición se realiza con patrones de resistencias pequeños que son usados como indicadores de tensión con eficiencia, llamados medidores. Estos están unidos a una estructura que se deforma cuando se aplica un peso, cuando varía el medidor de deformación la resistencia eléctrica cambia en proporción a la carga (Haeussler, 2003).

En la figura 1 se muestra un ejemplo de celda de carga de 5 Kg.

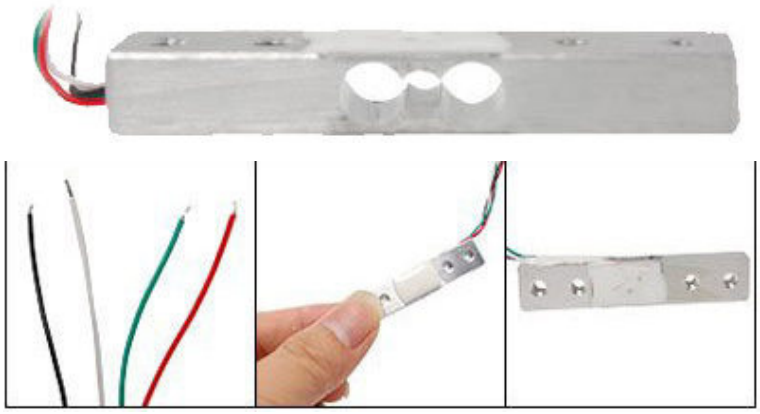


Figura 1. Celda de carga de 5 kg

Fuente: ((Mechatronics, 2012)

1.1.1.1 Características técnicas de las celdas de carga

En las tablas 1, 2 y 3 se muestran las características técnicas de las celdas de carga de 5 kg, 10 kg y 20 kg.

Dentro de las características indicadas para cada celda de carga se tienen:

- **Rango de carga:** Indica el valor hasta el cual puede sensar la celda (ROBOTSHOP, 2017).
- **Salida nominal:** Indica el valor máximo de voltaje suministrado a la salida (ROBOTSHOP, 2017).
- **Repetibilidad:** Indica la capacidad que tiene la celda para dar el mismo resultado con las mismas condiciones de entrada (ROBOTSHOP, 2017).
- **Efecto de la temperatura en la salida:** Es el cambio en el valor de salida cuando existe un cambio en la temperatura ambiente (ROBOTSHOP, 2017).
- **Efecto de la temperatura sobre cero:** Es el cambio en el valor 0 cuando existe un cambio en la temperatura ambiente (ROBOTSHOP, 2017).
- **Cero:** Indica la diferencia máxima entre los cables de salida +/- cuando no se aplica carga, ayuda a restablecer la condición a cero (ROBOTSHOP, 2017).
- **Impedancia de entrada:** Determina la potencia que será consumida por la celda de carga. Cuanto más bajo sea este número, más corriente será requerida, y más calentamiento ocurrirá cuando la celda de carga esté energizada (ROBOTSHOP, 2017).

- **Impedancia de salida:** Muestra el valor aproximado a la impedancia de entrada. Si la impedancia de salida es muy alta la medición en el puente puede distorsionar el resultado (ROBOTSHOP, 2017).
- **Resistencia de aislamiento:** Es el valor de resistencia eléctrica medida entre la estructura metálica de la célula de carga y el cableado (ROBOTSHOP, 2017).
- **Tasa Sobrecarga segura:** Indica el porcentaje de sobrecarga que puede soportar celda sin que provoque daños (ROBOTSHOP, 2017).
- **Tasa de sobrecarga final:** Es el porcentaje de sobrecarga máximo que puede soportar antes de su deformación (ROBOTSHOP, 2017).
- **Rango de temperatura:** Indica el valor máximo y mínimo de temperatura que puede soportar la celda para trabajar normalmente (ROBOTSHOP, 2017).
- **Voltaje de funcionamiento:** Muestra los valores de voltaje a los cuales puede trabajar normalmente (ROBOTSHOP, 2017).
- **Material:** Indica el tipo de material con el que está fabricada la celda (ROBOTSHOP, 2017).
- **Grado de protección:** Determina un rango de valor basado en una norma técnica que califica la protección de las personas contra el acceso a partes peligrosas y contra la penetración de cuerpos sólidos especiales. Y adicionalmente la protección del equipo en el interior de la envolvente contra los efectos por penetración de agua (ROBOTSHOP, 2017).

Para las celdas de carga se utiliza un sistema de protección IP (*Ingress Protection*), en la que se indica el grado de protección ante (Sánchez, 2013):

- Contacto y entrada de cuerpos sólidos extraños (Sánchez, 2013).
- Entrada de líquido (Sánchez, 2013).

La codificación es IP-NN, donde NN son 2 números que significan lo siguiente (Sánchez, 2013):

- Primer número, cuerpos extraños sólidos: 0 al 6 (Sánchez, 2013).
- Segundo número, entrada de líquidos: 0 al 8 (Sánchez, 2013).

- **Cable:** Da a conocer el diámetro del cable utilizado y el tamaño del mismo (ROBOTSHOP, 2017).
- **Tamaño:** Indica las dimensiones de largo, ancho y espesor que tiene la celda (ROBOTSHOP, 2017).
- **Full Scale (FS):** Indica un cambio en la salida cuando el sensor está completamente cargado (ROBOTSHOP, 2017).

Tabla 1: Características de celdas de carga de 5 kg

Descripción	Valor
Rango de carga :	5 kg
Salida nominal:	1,0 ± 0.15mV / V
Efecto de la temperatura sobre cero:	0.05% FS / ° C
Repetibilidad:	0,03 % FS
Efecto de la temperatura en la salida:	0,01% FS / °C
Cero:	± 0.1000 mV / V
Impedancia de entrada:	1115 + -10% Ω
Impedancia de salida:	1000 + - 10% Ω
Resistencia de aislamiento:	> = 1000 MΩ
Tasa Sobrecarga segura:	150% FS
Tasa de sobrecarga final:	200% FS
Rango de temperatura:	-20 - 60 ° C
Voltaje de funcionamiento:	3VDC ~ 14 VDC (corriente directa).
Material:	Aleación de aluminio
Tamaño:	4.5 x 0.9 x 0.6 cm
Grado de protección:	IP65
Cable:	Diámetro (Φ) 0.8 × 25 cm

Fuente: (Electrónicas, 2013)

Tabla 2 Características de celda de carga de 10 kg

Descripción	Valor
Rango de carga :	10 kg
Efecto de la temperatura sobre cero:	0.05% F.S / ° C

Salida nominal:	1,0 ± 0.15mV / V
Repetibilidad:	0,03% FS
Efecto de la temperatura en la salida:	0,01% F.S / °C
Impedancia de entrada:	1115 + -10% Ω
Impedancia de salida:	1000 + - 10% Ω
Cero:	± 0.1000 mV / V
Resistencia de aislamiento:	> = 1000 MΩ
Tasa Sobrecarga segura:	150% FS
Tasa de sobrecarga final:	200% FS
Rango de temperatura:	-20 - 60 ° C
Voltaje de funcionamiento:	3VDC ~ 14 VDC.
Material:	Aleación de aluminio
Tamaño:	4.5 x 0.9 x 0.6cm
Cable:	φ0.8 × 25 cm
Grado de protección:	IP65

Fuente: (BIGTRONICA, 2017)

Tabla 3: Características de celda de carga de 20 kg

Descripción	Valor
Rango de carga :	20 kg
Repetibilidad:	0,03% FS
Salida nominal:	1,0 ± 0.15mV / V
Efecto de la temperatura en la salida:	0,01% F.S / °C
Impedancia de entrada:	1115 + -10% Ω
Efecto de la temperatura sobre cero:	0.05% F.S / ° C
Cero:	± 0.1000 mV / V
Impedancia de salida:	1000 + - 10% Ω
Resistencia de aislamiento:	> = 1000 MΩ
Tasa Sobrecarga segura:	150% FS
Tasa de sobrecarga final:	200% FS
Rango de temperatura:	-20 - 60 ° C
Voltaje de funcionamiento:	3VDC ~ 14 VDC.

Grado de protección:	IP65
Material:	Aleación de aluminio
Cable:	φ0.8 × 25 cm
Tamaño:	4.5 x 0.9 x 0.6cm

Fuente: (Electrónicas, 2013)

1.1.1.2 Modo de Operación

Las celdas de carga permiten convertir una fuerza en señal eléctrica. Esto se logra por medio de un puente Wheastone indicado en la figura 1, este es utilizado para medir resistencias desconocidas mediante el equilibrio de "brazos" del puente formando un circuito cerrado por medio de 4 resistencias. En el caso de las celdas de carga las resistencias son los medidores de deformación (Haeussler, 2003).

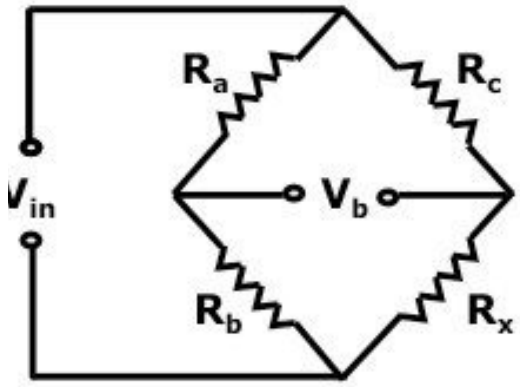


Figura 2. Puente Wheastone
Fuente: (TodoHard, 2007)

1.1.2. Trasmisor de celda de carga HX711

En la figura 3, se muestra un transmisor HX711 (Módulo amplificador de celdas de carga) que es un módulo utilizado como interfaz para medir peso/fuerza en las celdas de carga. Al conectar el amplificador al microcontrolador es posible medir los cambios en la resistencia de la celda de carga, con algunas calibraciones y cálculos, se puede obtener medidas bastante confiables (GeekFactory, 2017).

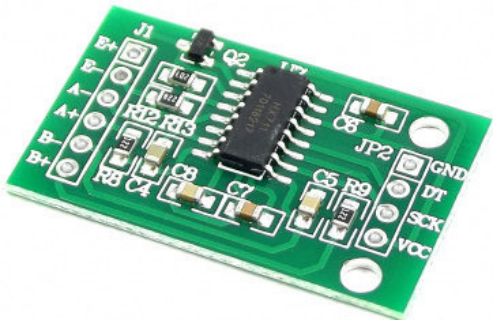


Figura 3. Trasmisor de celda de carga HX711

Fuente: (Mechatronics, 2012)

1.1.2.1 Características técnicas

En la tabla 4, se muestran las características principales del Transmisor de celda de carga HX711, como son:

- **Voltaje de operación**: Es el voltaje con el cual debe ser alimentado para su normal funcionamiento.
- **Consumo de corriente**: Indica el valor de corriente que consume la placa cuando está en funcionamiento
- **Voltaje de entrada diferencial**: Toma referencia cuando está en funcionamiento y es la diferencia entre los valores de voltaje de entrada entregados.
- **Dimensiones:** Son los valores de ancho, largo, y espesor de la placa.
- **Resolución a la conversión A/D**: Es el aumento que se le puede dar en bits cuando se realiza la conversión análoga a digital.
- **Frecuencia de refresco:** Es el valor de frecuencia al que se realiza una actualización de la información.

Tabla 4: Características de HX711

Características	Valor
Consumo de corriente	menor a 10 mA
Voltaje de Operación	5 V DC
Resolución conversión A/D	24 bit
Voltaje de entrada diferencial	±40 mV

Frecuencia de refresco	80 Hz
Dimensiones	10 mm x 38 mm x 21 mm

Fuente: (Mechatronics, 2012)

1.1.2.2 Modo de Operación

El modulo HX711 permite ser utilizada como una interfaz entre las celdas de carga y el microcontrolador, usa una interfaz de dos hilos que permiten realizar la comunicación entre dos pines de I/O (entrada y salida) de cualquier microcontrolador. Este modulo convierte la lectura analógica a digital con su conversor A/D interno de 24 bits internamente basado en la lectura del puente de Wheatstone formado por la celda de carga, (GeekFactory, 2017).

Adicionalmente, tiene varias librerías que permiten realizar la interfaz de este dispositivo con plataforma Arduino (GeekFactory, 2017).

1.1.3 Placa Arduino Mega 2560

Arduino es una compañía de *hardware* libre, que se dedica a la fabricación de placas que integran un microcontrolador y un entorno de desarrollo para facilitar el uso de la electrónica. Está conformada por un *hardware* que contiene un microcontrolador encargado de realizar procesos matemáticos y lógicos dentro de la placa, además permite gestionar y controlar los recursos de cada uno de los componentes conectados (Arduino, 2010).

Cuenta con entradas de pines digitales y analógicos con las que se puede integrar componentes externos como sensores de sonido o ultrasónicos, motores, seguidores de línea entre otros. Tiene puertos seriales de entrada/salida, que ayudan a conectarse por medio de un cable USB a la computadora para trabajar mediante *software* (Arduino, 2010).

El lenguaje puede ampliarse a través de librerías de C++, funciona en los sistemas operativos Windows, Linux entre otros ya que su entorno de programación es directo, simple y publicado bajo una licencia libre. (Arduino, 2010).

En la figura 4, se muestra el Arduino Mega 2560 que se utilizó para la elaboración del presente prototipo.

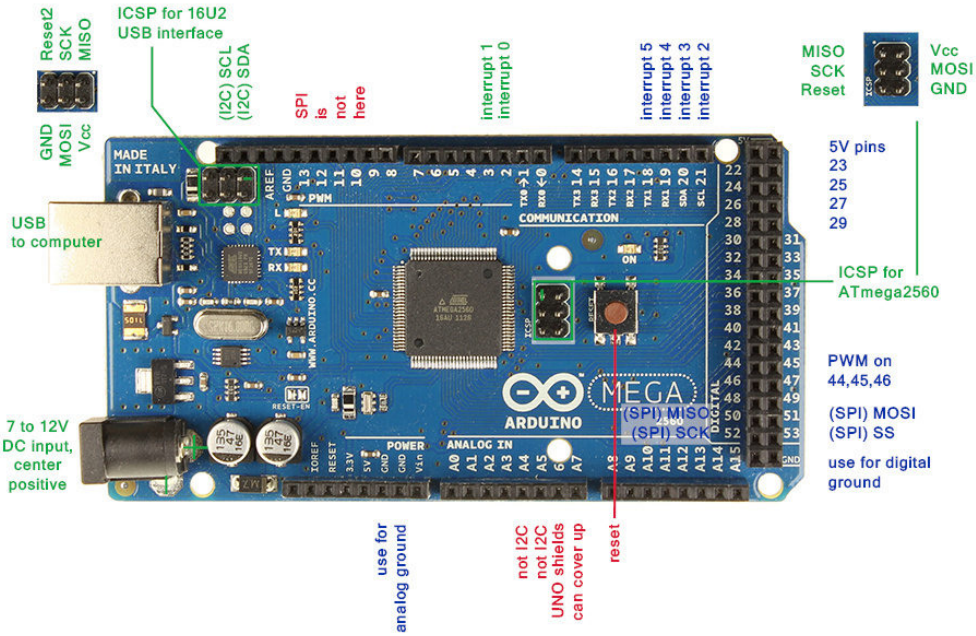


Figura 4: Arduino Mega

Fuente: (Arduino, 2010)

1.1.3.1 Características Técnicas de Arduino ATmega2560

En la tabla 5 se muestran las principales características técnicas de la placa Arduino ATMega2560 como son:

- **Tensión de funcionamiento**: Es el valor de voltaje al cual trabaja la tarjeta (Arduino, 2010).
- **Tensión de entrada (recomendado)**: Es el valor de voltaje de entrada que puede soportar para su correcto funcionamiento (Arduino, 2010).
- **Tensión de entrada (limite):** Define el valor de voltaje minino o máximo que puede soportar para trabajar normalmente (Arduino, 2010).
- **Pines de E / S digitales:** Son los pines designados para ser conectados como medios de ingreso o salida de datos los mismos que deben ser digitales (Arduino, 2010).

- **Pines de entrada analógica**: Son los pines designados para ser conectados como medios de ingreso de datos de señales analógicas (Arduino, 2010).
- **Corriente DC por pin de E / S:** Es la corriente continua que entrega cada PIN a la entrada o salida del mismo (Arduino, 2010).
- **Corriente DC para clavija:** Es la corriente continua que entregan las uniones de la placa (Arduino, 2010).
- **Memoria Flash:** Es la capacidad de almacenamiento de información que puede guardar cuando no está alimentado el circuito (Arduino, 2010).
- **SRAM:** Es el tamaño de memoria estática utilizada para refrescar los datos en un rango de 10 a 30 nanosegundos (Arduino, 2010).
- **EEPROM:** Es el tamaño dememoria de solo lectura que dispone la placa y esta es reprogramable (Arduino, 2010).
- **Velocidad del reloj:** Es la velocidad de frecuencia con la que trabaja la señal de reloj que posee la placa (Arduino, 2010).
- **Longitud:** Indica la longitud de la placa en milímetros (Arduino, 2010).
- **Ancho:** Indica el ancho de la placa en milímetros (Arduino, 2010).
- **Peso:** Indica el peso que tiene la placa (Arduino, 2010).

Tabla 5: Características de Arduino ATmega2560

Características	Valor
Tensión de entrada (límite)	6-20V
Tensión de entrada (recomendado)	7-12 V
Tensión de funcionamiento	5 V
Pines de entrada analógica	16
Pines de E/S (Entrada/salida) digitales	54 (de los cuales 15 proporcionan salida PWM (Modulación por ancho de pulso)
Corriente DC(por PIN de E/S)	20 mA
Corriente DC para clavija	3.3 V 50 mA
Memoria Flash	256 KB de los cuales 8 KB utilizados por *bootloader*
SRAM (Memoria estática de acceso aleatorio)	8 KB

EEPROM (Memoria de sólo lectura programable y borrable eléctricamente)	4 KB
Velocidad del reloj	16 MHz
Longitud	101.52 mm
Ancho	53.3 mm
Peso	37

Fuente: **(Arduino, 2010)**

1.1.3.2 Modo de Operación

La placa Arduino ATmega2560 cuenta con un oscilador de cristal de 16 MHz, un puerto USB de conexión, 4 UART(Transmisor-Receptor Asíncrono Universal) puertos de *hardware* de serie, 54 pines entradas/salidas digitales de los cuales 14 pueden ser utilizados como salidas PWM(Modulación por ancho de pulso) y 16 entradas analógicas, un botón de reinicio y un conector de alimentación, una cabecera de ICSP(Programación Serial en Circuito) (Arduino, 2010).

Puede alimentarse con una batería, conectándolo a un ordenador con cable USB, o con un adaptador AC/DC (Corriente directa a corriente continua). Adicionalmente, por ser un *software* libre permite integrar en sus librerías el funcionamiento para la placa HX711 para realizar el sensado del peso.

1.1.4 Diodo led

Los diodos led están fabricados con un compuesto semiconductor denominado arseniuro de galio (AsGa), se caracterizan porque emiten radiación en el espectro visible, sobre todo el infrarrojo, cuando se les polariza en forma directa (Carretero, 2009).

El diodo led al igual que un diodo normal, absorbe energía en la generación de pares electrón hueco, esta energía vuelve a ser emitida cuando los electrones se recombinan con los huecos. En la mayoría de los semiconductores esta energía se emite en forma de calor (Carretero, 2009).

Los diodos pueden ser utilizados como indicadores de presencia de tensión de circuitos, en la construcción de *displays*, etc. Pueden ser fabricados de diversos colores correspondiéndose de la longitud de onda (λ) de la radiación luminosa (roja, verde, ámbar, naranja, azul e infrarroja) que depende del tipo de dopaje añadido al cristal semiconductor. En la figura 5 se indica el esquema y símbolo que se utiliza para identificar a un diodo led (Carretero, 2009).

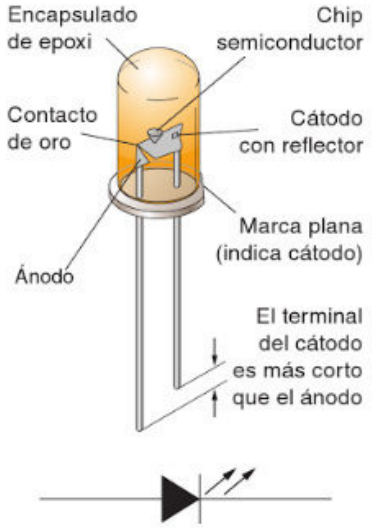


Figura 5. Esquema y símbolo del **Diodo led**

Fuente: (Carretero, 2009)

1.1.4.1 Características técnicas:

La tabla 6, muestra la tensión umbral típica que es el valor de voltaje al cual empiezan a conducir los diodos led de acuerdo a la radiación luminosa.

Tabla 6: Valores de Tensión umbral típica de acuerdo al color del led

Radiación Luminosa	**Infrarrojo**	**Rojo**	**Naranja**	**Ámbar**	**Verde**	**Azul**
Tensión umbral Típica	1,3 V	1,7 V	2,0 V	2,5 V	2,5 V	4, 0 V

Fuente: **(Carretero, 2009)**

1.1.4.2 Modo de Operación

El diodo led se polariza a la inversa como la mayoría de los diodos, no circulando por él corriente apreciable. Cuando tiene polarización directa, la tensión

umbral (Vu) varía entre 1,3 V y 4 V dependiendo de la radiación emitida (Carretero, 2009).

La intensidad mínima es de es de 4 mA y por precaución como máximo debe aplicarse 50 mA para que un diodo emita luz visible. El valor típico de intensidad que se utiliza frecuentemente es de 10 mA (Carretero, 2009).

1.1.5 Buzzer o Zumbador

Es un transductor electroacústico que produce un sonido a veces continuo o intermitente de un mismo tono que por lo general es agudo. Sirve como mecanismo de señalización o aviso y se utiliza en múltiples sistemas, como medio de señales auditivos, se lo puede encontrar en automóviles o en electrodomésticos, incluidos los despertadores (Electrónica A. , 2017).
En la figura 6 se muestra un *buzzer* de 5 VDC.

Figura 6. Buzzer de 5VDC
Fuente: (Electrónica A. , 2017)

1.1.5.1 Características técnicas del Zumbador.

En la tabla 7, se muestran las características técnicas del Zumbador de 5 VDC en las cuales se detalla lo siguiente:

- **Voltaje:** El valor de voltaje al cual trabaja el zumbador.
- **Rango de voltaje:** Son los valores de voltaje máximos y mínimos que puede soportar normalmente para trabajar sin generar daños a los componentes.

- **Corriente (mA):** Es el valor de corriente que consume un componente cuando está en funcionamiento.
- **Salida sonido:** Es el valor de potencia entregada a la salida cuando el componente está en funcionamiento.
- **Frecuencia de Resonancia:** Es la frecuencia en la cual trabaja el zumbador.
- **Rango temperatura:** Es la temperatura mínima y máxima que puede soportar el componente para trabajar normalmente.
- **Diámetro:** Es el diámetro que mide el componente
- **Alto:** Es el valor de altura que tiene el componente.

Tabla 7: Características del Zumbador de 5 VDC

Características	Valor
Voltaje	5 VDC
Rango de voltaje	4-7 VDC
Corriente (mA)	<=32 mA
Salida sonido	>= 85 dB
Frec. Resonancia	2300 Hz +-300
Rango temperatura	-20 ºC a 45 ºC
Diámetro	Φ 12 mm
Alto	9.5 mm

Fuente: (Electrónica A. , 2017)

1.1.5.2 Modo de Operación:

El zumbador, está compuesto de dos elementos, un electroimán o disco piezoeléctrico y una lámina metálica de acero que al ser conectado a circuitos integrados especiales emite distintos tonos. Se crea un campo magnético variable que hace vibrar la lámina de acero sobre la armadura cuando circula la corriente por la bobina del electroimán, o bien, la corriente pasa por el disco piezoeléctrico haciéndolo entrar en resonancia eléctrica y produciendo ultrasonidos que son amplificados por la lámina de acero (Electrónica A. , 2017).

1.2. Marco Conceptual:

A continuación se algunos de los temimos utilizados en el documento.

Transductor: Son dispositivos que permiten transformar o convertir una señal de un tipo de energía ya sea de naturaleza eléctrica, mecánica, acústica u otras en otro tipo de energía con el fin de obtener la misma información.

Transductor electroacústico: es aquel que permite transformar una señal eléctrica en sonora cuando existe una variación entre las placas de un condensador provocando una diferencia de voltaje.

Galga extensiométrica: es un tipo de transductor que convierte el desplazamiento o deformación en señales eléctricas cuando existe un esfuerzo mecánico.

Pin (Personal Identification Number las siglas en ingles): es un número de identificación personal que se utiliza en ciertos sistemas para identificar una salida. En el presente trabajo es utilizado para identificar las señales de salida y/o entrada de un circuito o componente electrónico.

Tensión umbral: es el valor de voltaje con la cual empieza a conducir un diodo y este depende del tipo de diodo que se esté utilizando.

IP65: es un código de identificación utilizado para indicar el grado de protección que tiene una envolvente contra el contacto, penetración de agua y suciedad.

Estándar IEEE 830: es una especificación de requisitos para la creación de software en la cual incluyen normas y procedimiento que deben ser utilizados.

Tara: se le llama al valor del peso de un vehículo o recipiente donde se transporta o contiene una mercancía.

CAPÍTULO II. PROPUESTA

2.1 Diseño

El diseño del prototipo se divide en dos etapas una de *software* y otra de *hardware* como se detalla a continuación:

2.1.1 Diseño de *Hardware*

2.1.1.1 Diagrama de bloques

En la figura 7, se muestra cómo está conformado el prototipo propuesto, el cual consta de una parte eléctrica que es la encargada de la alimentación del circuito electrónico y el *Hardware* conformado por la parte electrónica que está dividido en 3 etapas que son: una etapa de entrada, una etapa de control y una etapa de salida dando como resultado la despensa inteligente.

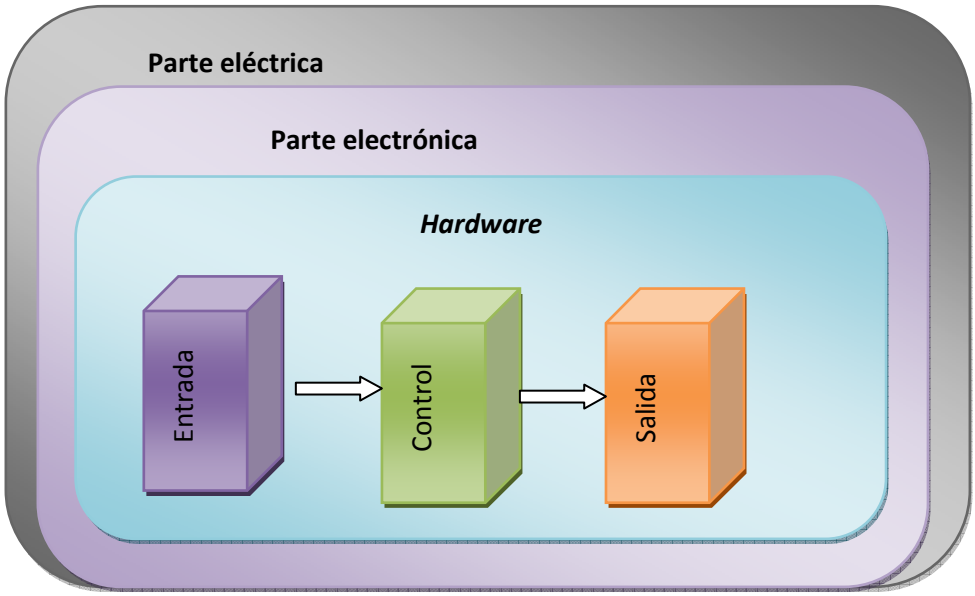


Figura 7. Diagrama de Bloques del Prototipo

Fuente: Elaborado por el autor

El prototipo se lo debe alimentar a través de la placa Arduino que puedes ser conectado a un cable USB conectado al computador o un adaptador de 5 VDC de acuerdo a lo indicado en las características técnicas del componente.

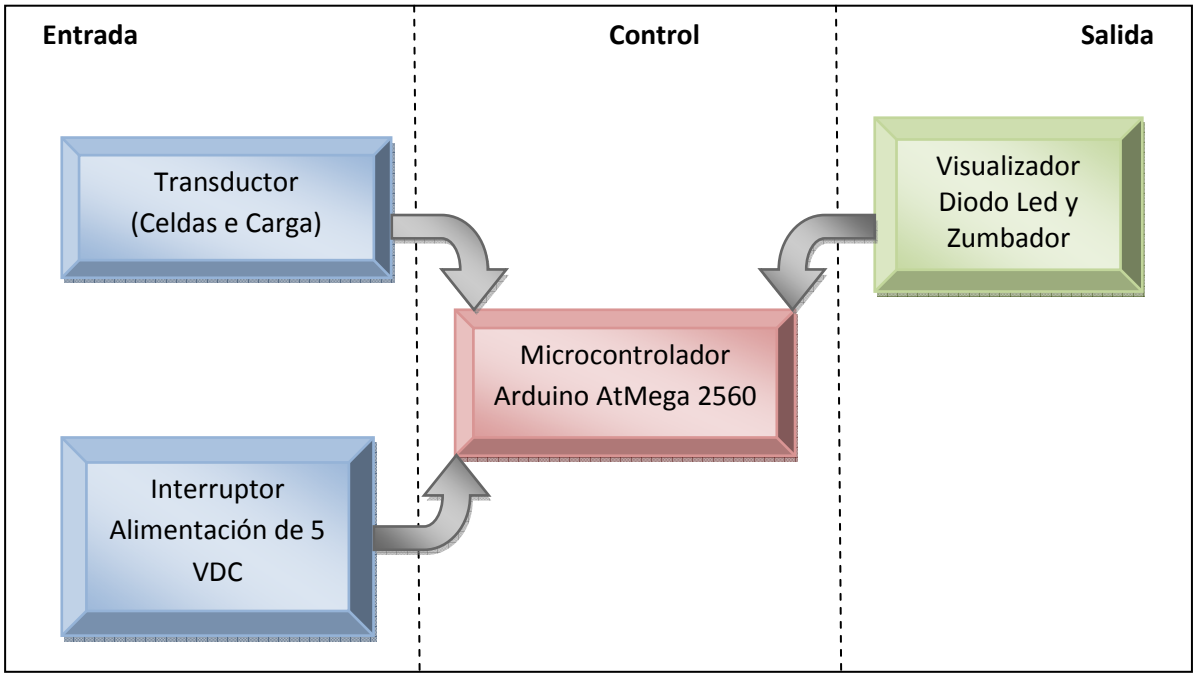


Figura 8. Diagrama de bloques del diseño electrónico

Fuente: Elaborado por el autor

En el gráfico 8, se indican las partes del *hardware* en sus diferentes etapas.

Etapa de entrada.

En la etapa de entrada está compuesta por un transductor que será utilizado para tomar el peso de los productos a ser almacenados, para esto se ocuparán celdas que carga que por su funcionalidad como transductor permitirán capturar el peso de los productos a través de los medidores de deformación que contienen internamente y convirtiendo el resultado de esta medición en señales eléctricas las cuales facilitaran él envió de información a la placa Arduino que se encuentra en la etapa de control.

Etapa de control

Para la etapa de control se utilizará una placa Arduino que contiene un microcontrolador ATMega 2560, al tener un microcontrolador permitirá grabar en su memoria un programa con instrucciones, procesar los datos entregados por las celdas de carga y enviar los resultados de las instrucciones a los visualizadores.

Etapa de salida

La etapa de salida está enfocada a generar alertas a las personas encargadas del abastecimiento cuando el producto de encuentra agotado o fuera del tiempo de almacenamiento mediante alertas auditivas y visuales. Para esto se utilizará un diodo led que debido a que trabaja con polarización inversa se encenderá cuando ya no se detecte peso sobre las celdas de carga, adicionalmente se utilizará un zumbador que al ser un transductor electroacústico emitirá un sonido generando la alerta sonora.

2.1.1.2 Diagrama Esquemático

En la figura 9, se indica el diagrama esquemático del prototipo que consta de las partes que se muestran a continuación:

A. La estructura de la gaveta para abastecimiento será elaborada en madera de 25 cm de largo y 29 cm de ancho ya que los productos a ser abastecidos no superan estas dimensiones, adicionalmente servirán para ubicar las celdas de carga que al trabajar con el estándar IP65 de acuerdo a las características técnicas especificadas por el fabricante, cuentan con las protecciones necesarias contra agua y polvo para su normal funcionamiento.

B. Indica la cavidad donde se colocarán los productos a ser pesados mediante celdas de carga que se instalarán al fondo tomando en cuenta el punto de apoyo que se debe generar para una correcta medición.

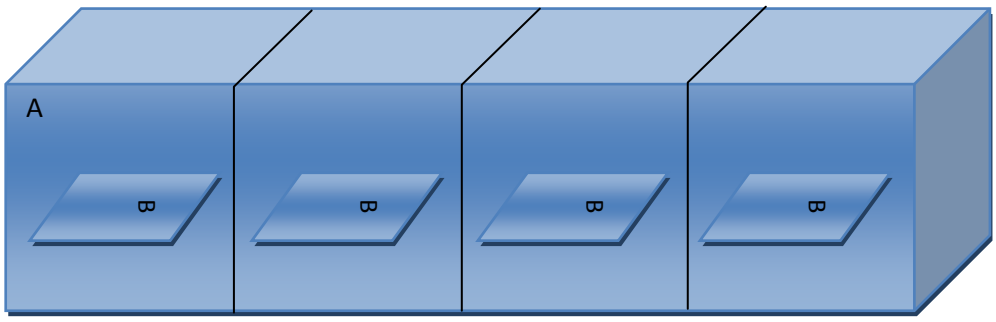


Figura 9. Diagrama esquemático del Prototipo

Fuente: Elaborado por el autor

2.1.2 Diseño del *Software*:

La elaboración de la propuesta se basará en el estándar IEEE 830, el aplicativo llevará el nombre "Despensa inteligente" y será el encargado de realizar un registro de la fecha de ingreso de los productos a ser almacenados, creará una estructura de código para la identificación del producto con la cual se podrá almacenar el dato del periodo máximo de consumo, se encargará también de asignar en el sistema el lugar donde será ubicado físicamente el producto. Este programa será capaz de interpretar los datos entregados por la placa Arduino y codificarlos para enviar las señales de alerta.

El programa será diseñado con el *software* Visual Studio 2013 el mismo que por sus características permite el desarrollo de aplicaciones compatibles con Windows, además por la ventaja que brinda esta plataforma se desarrollará un código que permita controlar los datos entregados por la placa Arduino.

2.1.2.1 Flujograma del Sistema

En las figuras 10 y 11, se detalla el flujograma del programa que será utilizado para la elaboración del prototipo.

El programa "Despensa Inteligente", inicia con la lectura de los puertos de comunicación disponibles en el computador instalado. Una vez detectados los puertos se deberá seleccionar el puerto correspondiente al Arduino, se dispondrá de un botón para abrirlo. Una vez abierto el puerto permitirá la comunicación entre el aplicativo y la placa Arduino dando paso a la siguiente etapa del proceso.

Cuando se envía la solicitud de búsqueda y no encuentra ningún puerto disponible el programa deberá seguir insistiendo hasta obtener respuesta a lo solicitado, una vez conseguido el resultado se continuará con el proceso.

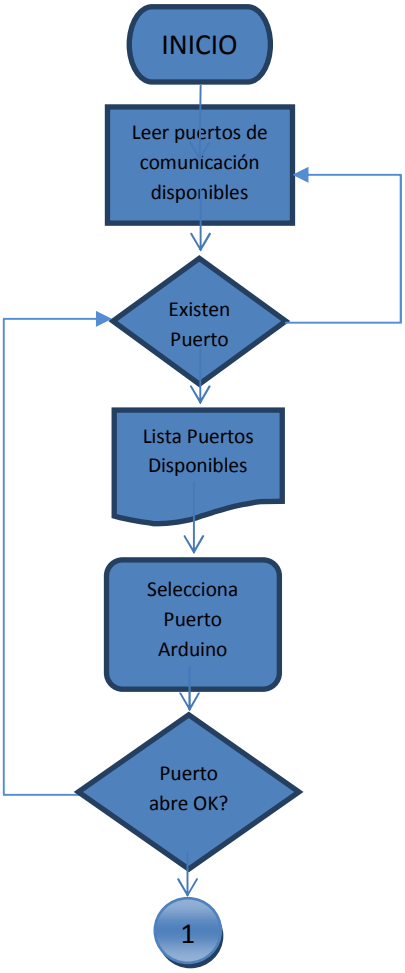


Figura 10. Flujograma del sistema parte I

Fuente: Elaborado por el autor

Cuando se establece la comunicación el programa enviará un comando para leer el peso en la placa Arduino, si encuentra la información solicitada muestra los valores, caso contrario sigue solicitando hasta obtener respuesta.

Una vez obtenido el valor del peso, realiza una comparación de que el valor obtenido no sea inferior al valor configurado como mínimo en cada una de las gavetas. Cuando se cumple la condición de que el peso es inferior se envía un comando para activar el led y zumbador.

Adicionalmente, realiza una comparación entre la fecha de ingreso y los días de almacenamiento configurados, si la fecha de ingreso supera el límite de días, envía un comando para la activación del led y zumbador. Tiene la opción de ingreso de nuevo producto con la que obtiene la fecha del sistema, calcula los días de validez, guarda los cambios y almacena un archivo con los datos obtenidos.

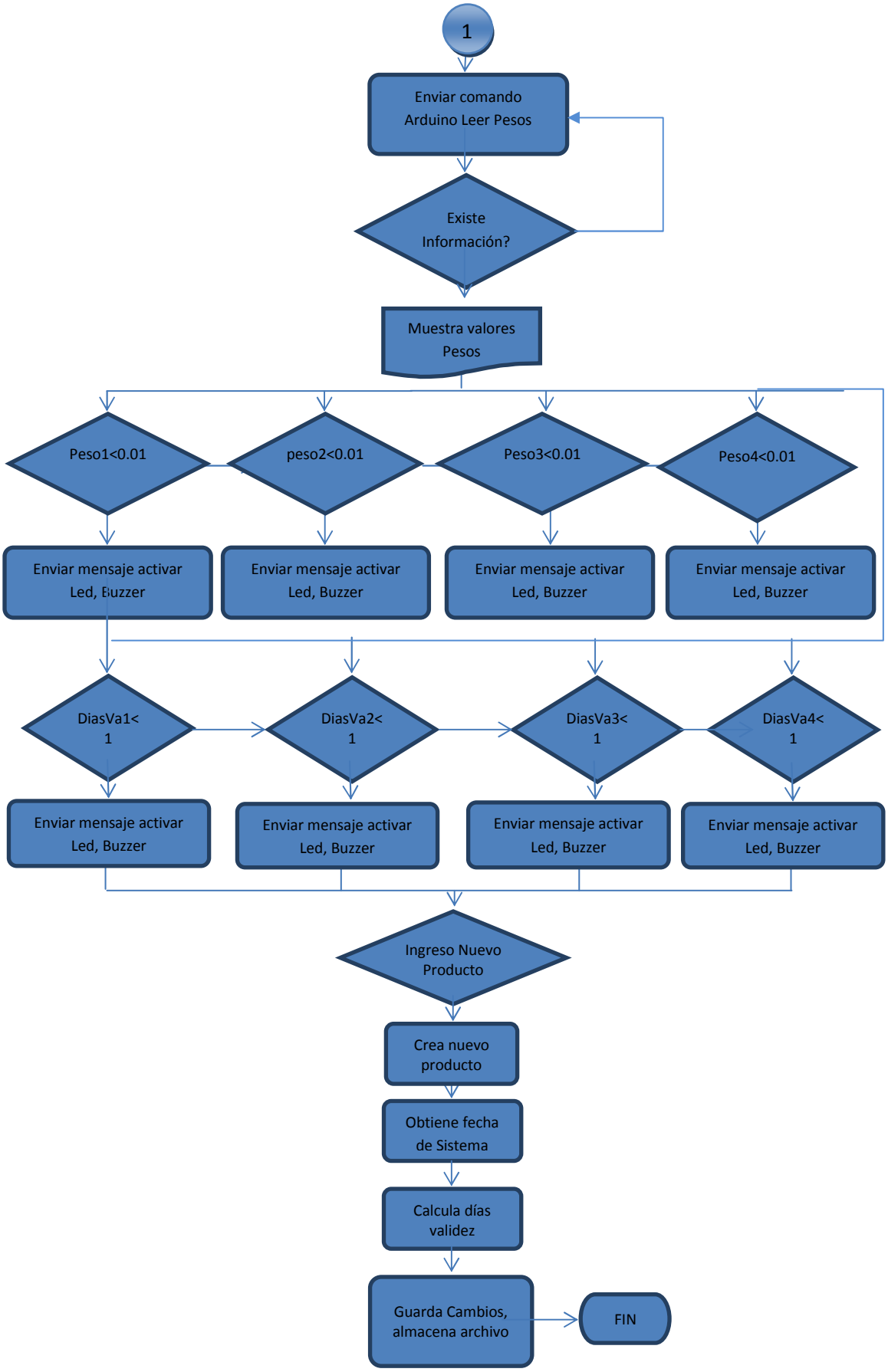


Figura 11. Flujograma del sistema parte II

Fuente: Elaborado por el autor

CAPÍTULO III ETAPA DE IMPLEMENTACIÓN

En este capítulo se detallarán las fases que se utilizaron para la implementación del prototipo.

3.1 Hardware

Para la elaboración del prototipo se necesita cumplir con 2 variables que son el control de peso y el tiempo máximo de consumo de los productos a ser almacenados. Para realizar la medición del peso se utilizaron celdas de carga ya que por su funcionalidad como transductor brindan la posibilidad de aplicar un peso y transformarlo a la salida en una señal eléctrica. De acuerdo a las características técnicas especificadas por el fabricante estas celdas de carga trabajan con un voltaje de 3 VDC hasta 14 VDC lo que permite ser acoplada a otros circuitos con similares características.

Adicionalmente, trabajan con un grado de protección IP65 basado en la norma CENELEC IEC 60529, que significa:

6: Protección completa contra contacto con las partes móviles, situadas dentro de la envolvente. Protección completa contra entrada de polvo
5: Protección contra mangueras de agua. El agua proyectada por una manguera desde cualquier dirección no causará efectos perjudiciales.

Lo que ayuda a evitar problemas de manipulación de las partes internas y desgaste o mal funcionamiento por efecto del agua.

El rango de carga que se necesita para realizar el prototipo va estar limitado hasta 20 kg que es el peso máximo que podría soportar la despensa cuando se encuentre en funcionamiento. Por lo que se colocó 4 gavetas de la siguiente manera: una de 5 kg, dos de 10 kg y una de 20 kg, dependiendo del producto que se necesite almacenar se lo deberá ubicar en las celdas correspondientes. Estos pesos son suficientes ya que no se almacenan productos con pesos superiores en una despensa.

En la figura 12 y 13 se muestran el diagrama circuital del prototipo de despensa inteligente.

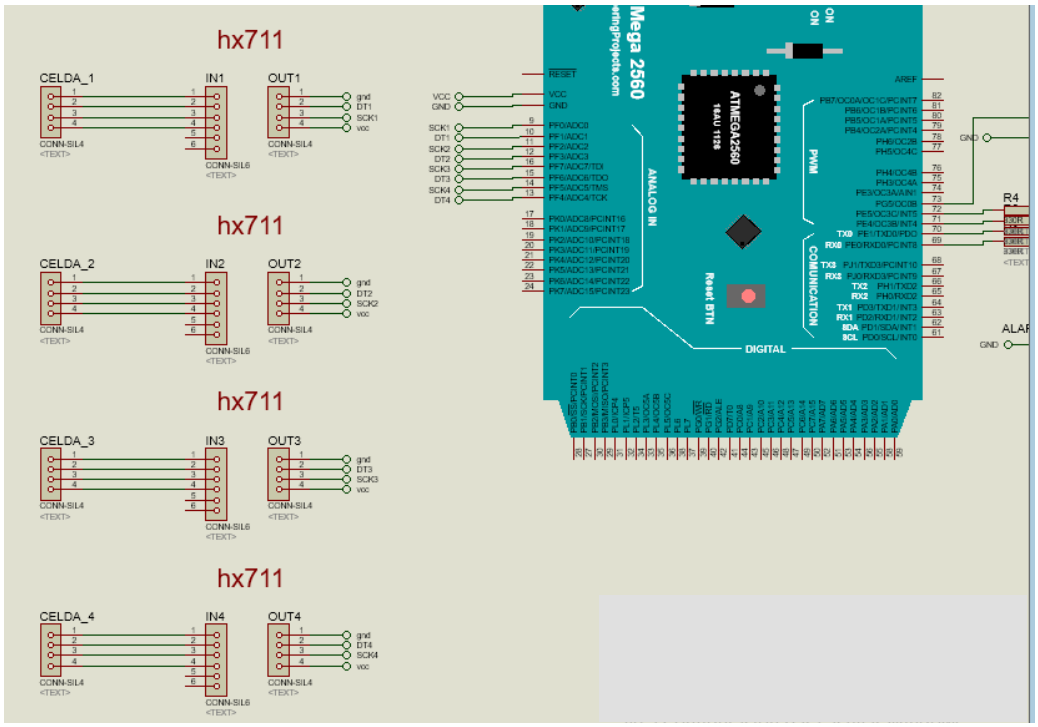


Figura 12. Diagrama Circuital de la etapa de control (parte 1)

Fuente: Elaborado por el autor

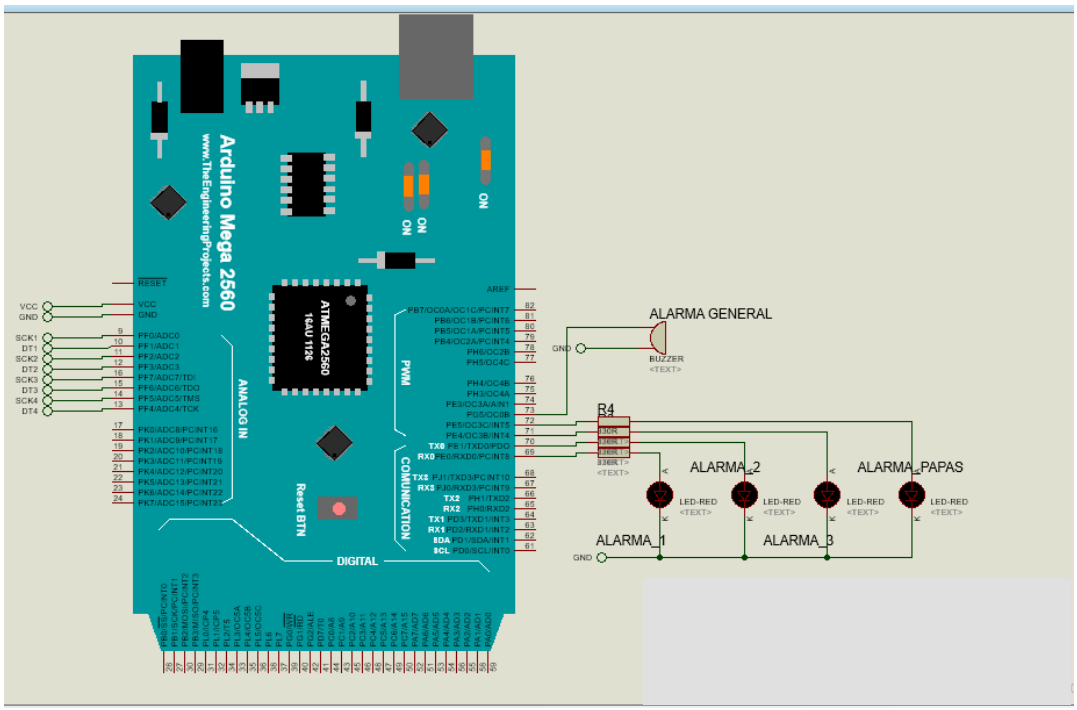


Figura 13. Diagrama Circuital de la etapa de control

Fuente: Elaborado por el autor

Las celdas de carga fueron conectadas a un HX711 (**t**rasmisor de celda de carga), se escogió este trasmisor ya que de acuerdo a las características técnicas indicadas por el fabricante posee internamente un conversor análogo a digital de 24 bits lo que permitirá ser conectado directamente a la salida de las celdas de carga y convertir el valor entregado a una señal digital la misma que podrá ser utilizada como entrada para la placa Arduino que se utilizará en el prototipo. Además, tiene una librería compatible con Arduino lo que ayuda a tener una programación más sencilla, su voltaje de operación es de 5 VDC lo que permite tener una referencia de voltaje igual en el circuito.

Para la etapa de control se utilizó una placa Arduino AtMega 2560, se escogió esta placa ya que facilita trabajar con un lenguaje de programación sencillo y adicionalmente, se ajusta a los componentes utilizados para la detección del peso. Lo importante también es que trabaja con un voltaje de 5 VDC lo que ayuda a tener una sola referencia de voltaje en todos los componentes utilizados, cuenta además con 14 entradas digitales lo que permite conectar directamente la señal digital entregada por el transmisor HX711. Cuenta con una librería compatible con la que se puede crear un código de programa donde se colocarán condiciones para obtener el resultado de control de peso, al disponer también de salidas digitales facilita trabajar con los led y zumbador utilizados como alertas visuales y auditivas. Además, la placa Arduino al trabajar con un puerto USB facilitó la integración con el *software* diseñado para el ingreso y control de productos.

3.1.1 Conexiones entre la Celda de carga, módulo HX711 y Arduino

De acuerdo al *datasheet* del Módulo HX711 y Arduino AtMega 2560, se realizarán las conexiones indicadas en las tablas 8 y 9.

En la tabla 8, se muestran las conexiones realizadas entre la celda de carga y el módulo HX711,

Los cables rojos de alimentación y cable negro de fuente negativa de las celdas de carga se conectaron a los pines excitación positivo y negativo del módulo HX711 respectivamente. Los cables verdes de señal positiva y cable blanco de señal negativa de la celda de carga se los conecto a los pines de amplificación positiva y negativa respectivamente.

Estas conexiones corresponden al puente de Wheastone que tiene la celda de carga y que ayudaran a sensar el peso de los productos.

Tabla 8: Conexión de la Celda de carga y el módulo HX711

Celda De Carga	Módulo HX711
Cable Blanco	PIN A+
Cable Negro	PIN E-
Cable Rojo	PIN E+
Cable Verde	PIN A-

Fuente: (Mechatronics, 2012)

En la tabla 9, se indican las conexiones que se deben realizar desde el módulo HX711 hacia la placa Arduino.

Los PINES de tierra (GND) de las 2 placas son conectados directamente, el PIN de salida de datos serial se conecta al PIN A1 que corresponde a uno de los pines de entrada análoga de la placa Arduino.

El PIN SCK que es el encargado del control de desconexión (activo alto) y entrada de reloj serie del módulo HX711 es conectado al PIN A0 de la placa Arduino como una entrada análoga.

El PIN VCC del módulo HX711 fue conectado al PIN 5 V del Arduino para alimentar el circuito formado por la celda de carga y módulo HX711.

Tabla 9: Conexión entre HX711 y Arduino

Módulo HX711	Arduino MEGA
PIN GND	PIN GND
PIN DT	PIN A1

PIN SCK	PIN A0
PIN VCC	PIN 5V

Fuente: (Mechatronics, 2012)

Para tener un control sobre los productos que se agotaron se realizará una medición del peso de cada producto colocado en cada gaveta con las que cuenta la despensa. Para poder realizar esta medición se instalaron en cada una de las gavetas celdas de carga de 5 kg, 10 kg y 20 kg debido a que por su funcionalidad como transductor ayudaron con la medición del peso de los productos y a la vez entregaron a la salida una señal de voltaje según lo indicado en las especificaciones técnicas del fabricante, las celdas de carga trabajan con un puente de Wheastone el mismo que puede ser acoplado con el módulo HX711 para convertir la señal análoga a digital.

En las figuras 14 y 15 se muestran la placa PCB (*Printed Circuit Board* con sus siglas Ingles o Placa de circuito impreso en español), en la cual se dibujaron las pistas que unen los diferentes componentes utilizados para el prototipo de acuerdo al diagrama circuital de las figuras 12 y 13. En esta placa se instalaron los 4 módulos HX711, la placa Arduino y resistencias de protección para los diodos led, también se colocaron los terminales que salen de los módulos para ser conectados con las celdas de carga los que ayudaran a medir el peso de los productos y además, los terminales que irán conectados en los diodos led que servirán de alertas visuales.

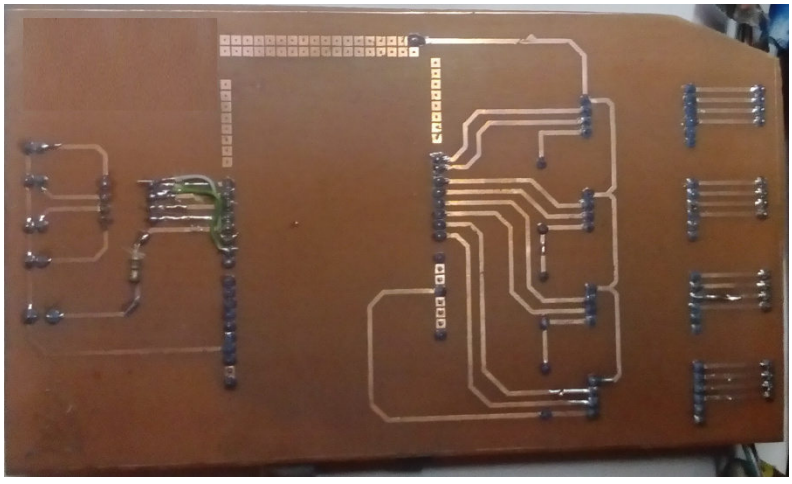


Figura 14. Placa inferior del circuito montada los componentes

Fuente: Elaborado por el autor

Para obtener las señales de alerta del sistema se instaló en la placa PCB un zumbador que sirvió de alerta sonora. Las alertas se activarán o desactivarán bajo las siguientes condiciones:

Se activa cuando:

1. La gaveta se encuentra con peso menor al configurado.
2. La fecha de ingreso del producto superó el límite de días a ser almacenados.

Se desactivan cuando:

1. La gaveta tiene un peso superior al configurado
2. Cuando se retiró el producto caducado, es decir se eliminó del sistema.

Cuando estén presente una de las 2 condiciones el sistema mantendrá las señales encendidas hasta que se abastezca de producto o a la vez se retire físicamente y del sistema el producto que se encuentra caducado.

Figura 15. Placa superior del circuito montada los componentes

Fuente: Elaborado por el autor

En la figura 16, se muestra la instalación de los diodos led en cada una de las gavetas que están destinados a servir como medios de alerta visual cuando se active una señal de alarma.

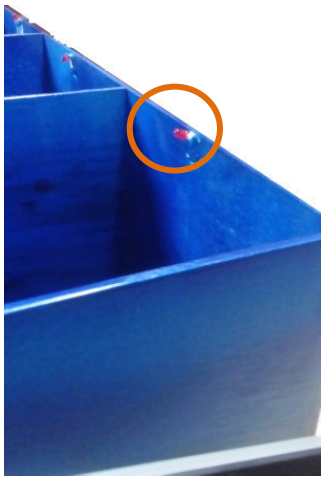


Figura 16. Led de Alerta

Fuente: Elaborado por el autor

En la figura 17, se muestra las gavetas donde se ubicarán los productos a ser almacenados, en este lugar se encuentran instaladas celdas de carga una de 5 kg, dos de 10 kg y una de 20 kg que ayudaran para obtener el peso.

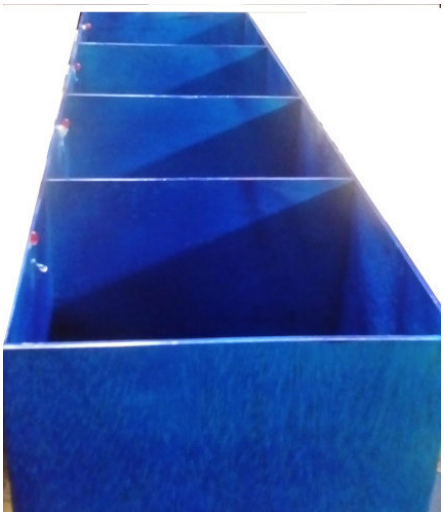


Figura 17. Etapa de sensado de Peso

Fuente: Elaborado por el autor

En la figura 18, se muestra las celdas de carga instaladas en cada una de las gavetas que de acuerdo a las especificaciones técnicas dadas por el proveedor indica que puede ser instalada en cualquier tipo de superficie que soporte el peso a ser medido, por lo que por ser un prototipo se las colocó sobre una superficie de madera.

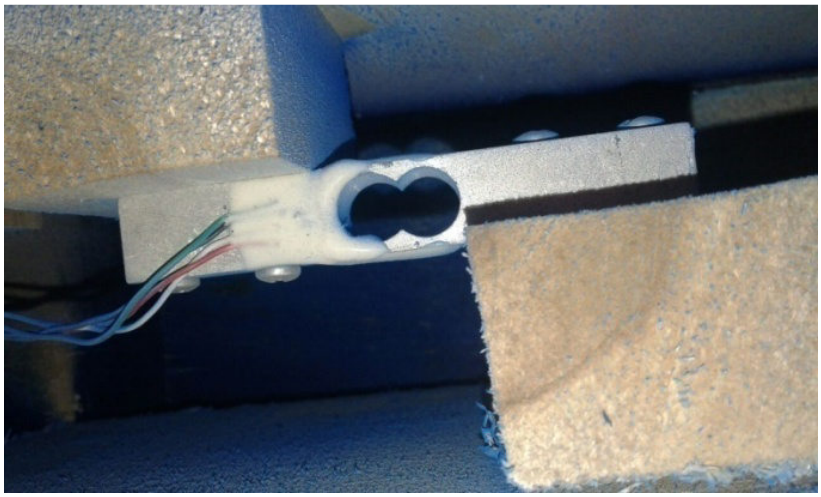


Figura 18. Celdas de carga instaladas en Prototipo
Fuente: Elaborado por el autor

3.1.2 Calibración de Celda de carga

Para calibrar las celdas de carga con Arduino y HX711, se utiliza la ecuación 1, para lo cual es necesario tener un peso conocido en la escala que se necesita medir sea esta en gramos, kilogramos, etc.

$$Escala: \frac{Valor\ de\ Lectura}{Peso\ Real}$$

Ecuación 1. Para calibrar la celda de carga

Para poder utilizar las celdas de carga se las debe calibrar corriendo un código de Arduino proporcionado por el fabricante del módulo HX711 con el que se podrá sacar el Tara que es el valor que permite conseguir la escala de cada celda como se muestra en la figura 19.

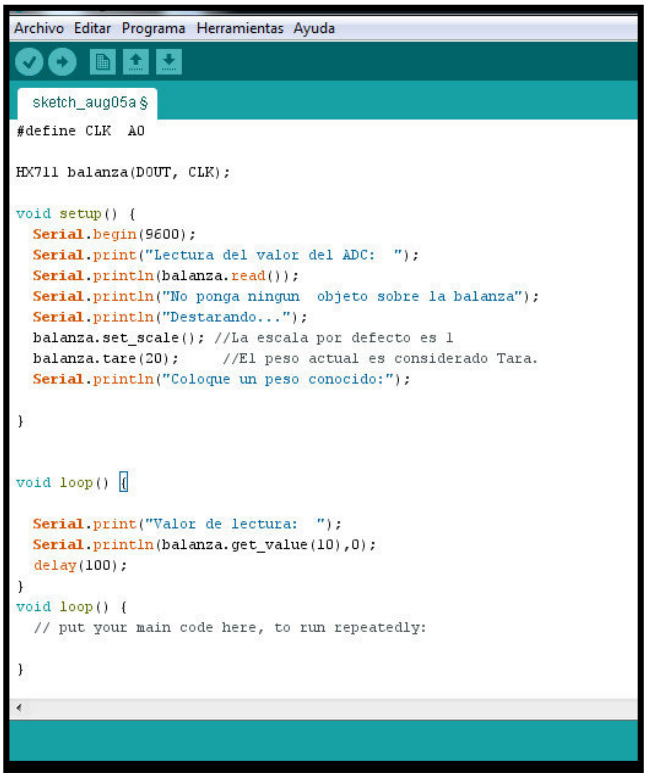


Figura 19. Código para obtener escala

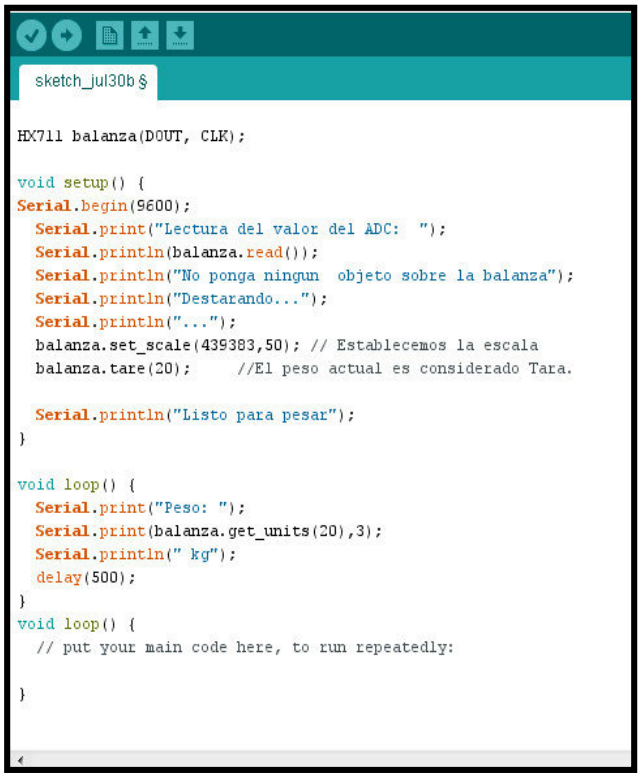
Fuente: Elaborado por el autor

Para la celda de carga de 5 kg tomando como valor de referencia un peso de 4 Kg y se utilizó la ecuación 1 para obtener el valor en kilogramos (kg).

$$Escala: \frac{1757534}{4\ kg}$$

$$Escala: 439383{,}5$$

En la figura 20 se muestra el código en Arduino utilizado para calibrar la celda de 5 kg utilizando el resultado anterior, con este valor se obtendrá la escala con la que se requiere trabajar para realizar las mediciones correspondientes en la celda de carga para el prototipo se realizaran las mediciones en kilogramos.



```
sketch_jul30b §

HX711 balanza(DOUT, CLK);

void setup() {
Serial.begin(9600);
  Serial.print("Lectura del valor del ADC:  ");
  Serial.println(balanza.read());
  Serial.println("No ponga ningun  objeto sobre la balanza");
  Serial.println("Destarando...");
  Serial.println("...");
  balanza.set_scale(439383,50); // Establecemos la escala
  balanza.tare(20);     //El peso actual es considerado Tara.

  Serial.println("Listo para pesar");
}

void loop() {
  Serial.print("Peso: ");
  Serial.print(balanza.get_units(20),3);
  Serial.println(" kg");
  delay(500);
}
void loop() {
  // put your main code here, to run repeatedly:

}
```

Figura 20. Código de Arduino para calibrar la celda de carga de 5 Kg.
Fuente: Elaborado por el autor

Para la celda de carga de 10 kg tomando como valor de referencia un peso de 4 Kg y se utilizó la ecuación 1 para obtener el valor en kilogramos (kg).

$$Escala: \frac{1757656}{4\ kg}$$

$$Escala: 439414$$

En la figura 21, se muestra el Código en Arduino utilizado para calibrar la celda de 10 kilogramos, con el resultado anterior se podrán obtener los valores que permitan visualizar las mediciones de peso que se realicen en la celda de carga

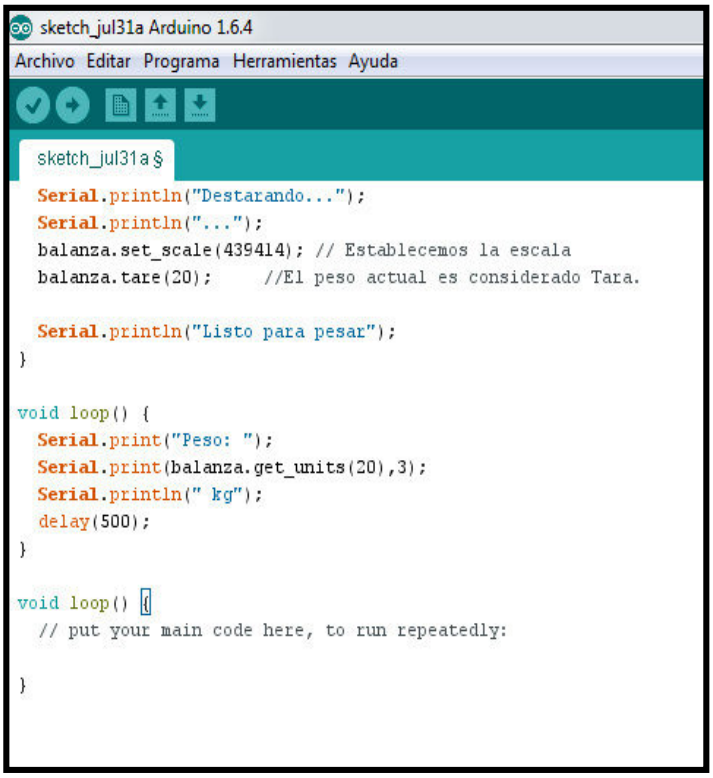


Figura 21. Código de Arduino para calibrar la celda de carga de 10 Kg

Fuente: Elaborado por el autor

Para la celda de carga de 20 kg tomando como valor de referencia un peso de 4 Kg y se utilizó la ecuación 1 para obtener el valor en kilogramos (kg).

$$Escala: \frac{1757662}{4\ kg}$$

$$Escala: 439415.5$$

En la figura 22, se muestra el Código en Arduino utilizado para calibrar la celda de 20 kilogramos, una vez configurado este valor se podrán realizar las mediciones de peso y visualizar los resultados.

.

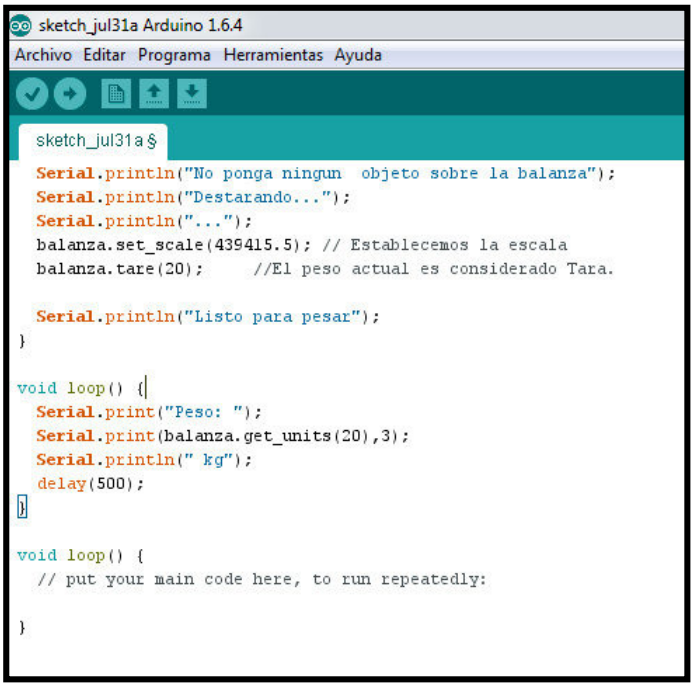


Figura 22. Código de Arduino para calibrar la celda de carga de 20 Kg

Fuente: Elaborado por el autor

3.1.3 Código de Arduino para la etapa de control

Para tener un control sobre las variables de peso se creó un código bajo plataforma Arduino el mismo que tiene como objetivo analizar la variable y activar cuando sean necesarias las señales de alerta. Este código permitirá además interactuar con el aplicativo de despensa inteligente creado como un control para el tiempo de caducidad de los productos.

En las figuras 23, 24 y 25 se indican el código del programa grabado en el microcontrolador de la placa Arduino, este código tendrá como entradas los datos proporcionados por el módulo HX711 el cual obtiene información de las celdas de carga, analiza los datos entregados y envía los resultados a los visualizadores y aplicativo de “Despensa inteligente”.

En la figura 23, se indica la definición de variables que se van a utilizar en cada una de las gavetas de la despensa. Estas gavetas contienen una celda de carga en cada una de ellas. Se disponen de dos celdas de 10 kg, una de 5 kg y una de 20 kg, que es el peso máximo que puede soportar cada una ellas.

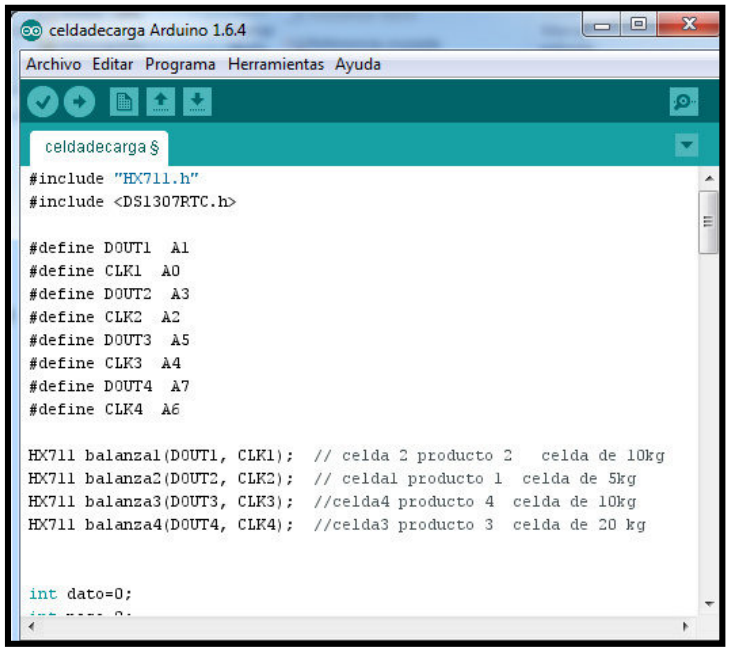


Figura 23. Código de Arduino para el control de las celdas de carga parte I

Fuente: Elaborado por el autor

En la figura 24, se muestra el código de programa para definir las señales de entrada y salida que serán utilizadas para obtener los valores de peso entregados por las celdas de carga.

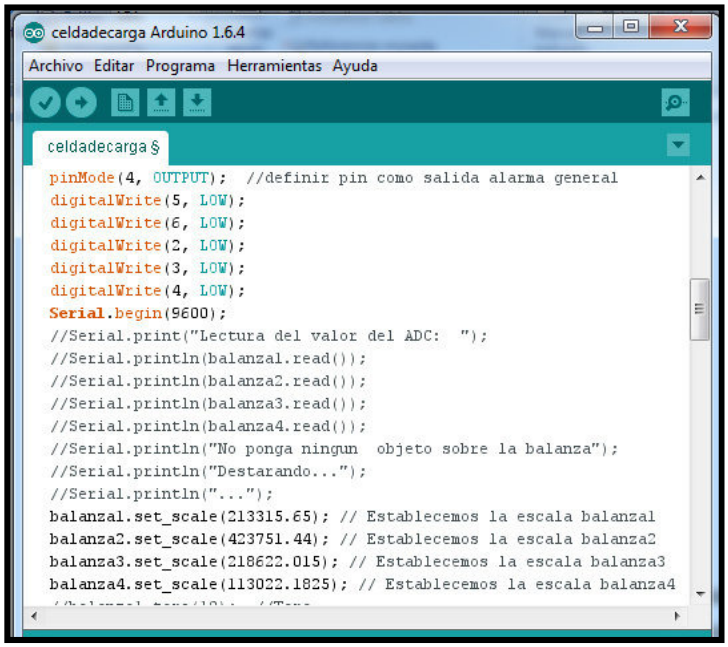


Figura 24. Código de Arduino para el control de las celdas de carga parte II

Fuente: Elaborado por el autor

En la figura 25, se muestra el código de Arduino utilizado para obtener el valor del Tara de cada celda de carga con lo que se podrá leer el valor del peso entregado.

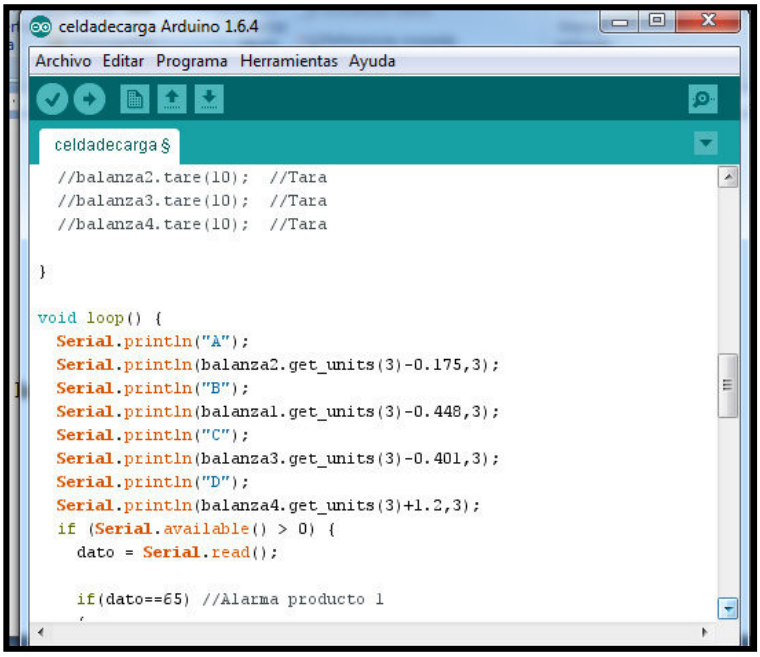


Figura 25. Código de Arduino para el control de las celdas de carga parte III

Fuente: Elaborado por el autor

3.2 Software

Para trabajar con la variante de tiempo máximo de almacenamiento se realizó el un aplicativo realizado en Visual Studio ya que es un *software* que permite trabajar bajo plataforma Windows.

El Aplicativo se lo llamará “Despensa inteligente” y será utilizado para determinar los días de vigencia que tiene el producto almacenado.

Permitirá asignar a cada gaveta un tipo de producto, el mismo que estará identificado con una serie por lotes la misma que se la registrará en el aplicativo de manera manual o con ayuda de un lector de un lector de barra.

Los códigos serán únicos y se le asignará a cada producto al momento de ingresarlo en el sistema. Estará compuesto por máximo dos letras y números que permitirán identificar el lote almacenarse.

El aplicativo consta de las siguientes partes:

1. Etapa de encendido y apagado
2. Configuración de nombres
3. Registro o eliminación de producto
4. Estado del producto

3.2.1 Etapa de encendido y apagado

El prototipo se lo controla a través del aplicativo despensa inteligente en el que como primer paso se debe detectar el puerto del Arduino disponible para poder encender la placa Arduino con la que se controla el sistema, de igual manera se desconecta el prototipo con el aplicativo.

En la figura 26, se indica el código del programa utilizado para buscar el puerto de Arduino disponible y seleccionarlo.

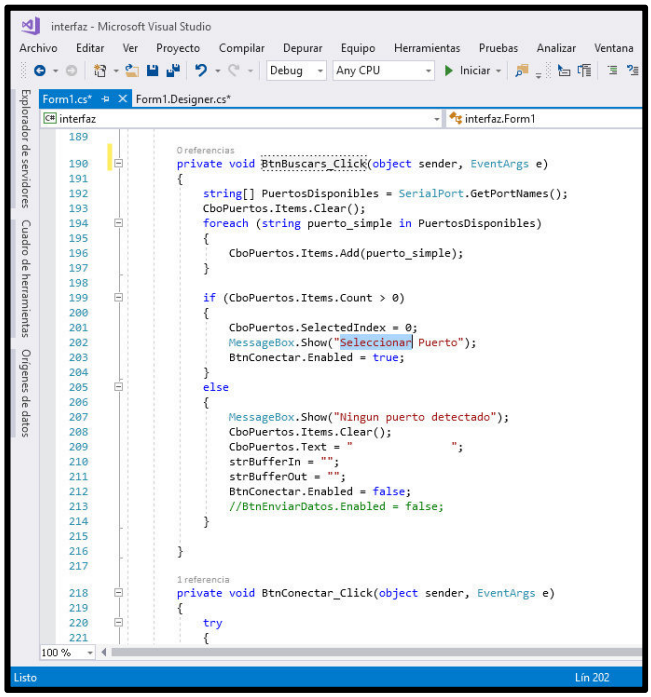


Figura 26. Código de programa para buscar puerto disponible

Fuente: Elaborado por el autor

En la figura 27, muestra la pantalla donde se puede buscar y seleccionar el puerto de Arduino disponible en el computador que se está trabajando.

Figura 27. Pantalla de búsqueda de puerto

Fuente: Elaborado por el autor

En la figura 28, se muestra el código del programa que se utilizo para permitir iniciar el uso del aplicativo, el botón se activa cuando previamente se seleccionó el puerto del Arduino, cuando el aplicativo ya se encuentra en funcionamiento se lo puede desconectar presionando el botón de “desactivar”.

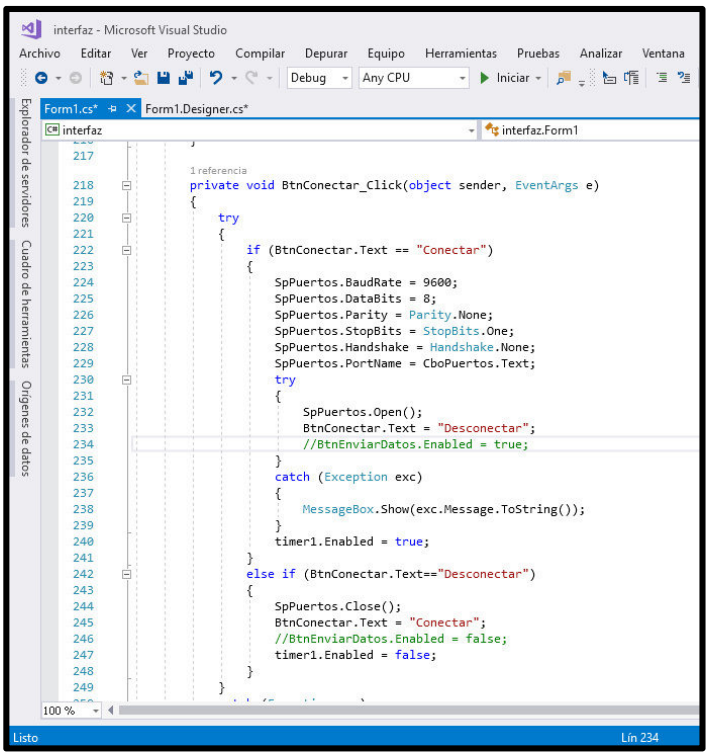


Figura 28. Código de programa para conectar y desconectar despensa

Fuente: Elaborado por el autor

En las figuras 29 y 30, se indican los botones para conectar y desconectar la placa Arduino, a través de estos se podrá poner en funcionamiento el aplicativo que interactuará con el *hardware* permitiendo controlar los sistemas de alarma.

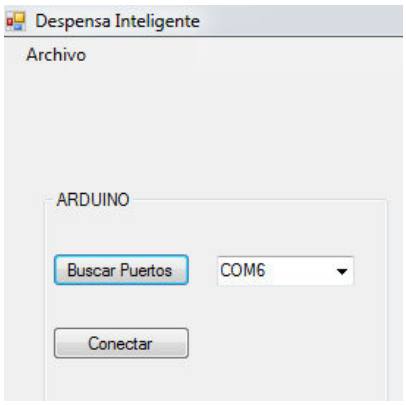


Figura 29. Pantalla de botón conectar

Fuente: Elaborado por el autor

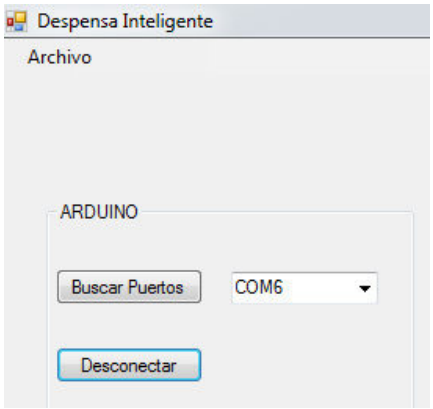


Figura 30. Pantalla de botón desconectar

Fuente: Elaborado por el autor

3.2.2 Configuración de nombres

En la figura 31, se detalla el código utilizado para la identificar los productos con un nombre único y serie alfanumérica por lotes, además la configuración del valor mínimo de peso que tendrá la despensa en cada una de las gavetas para generar la señal de alarma.

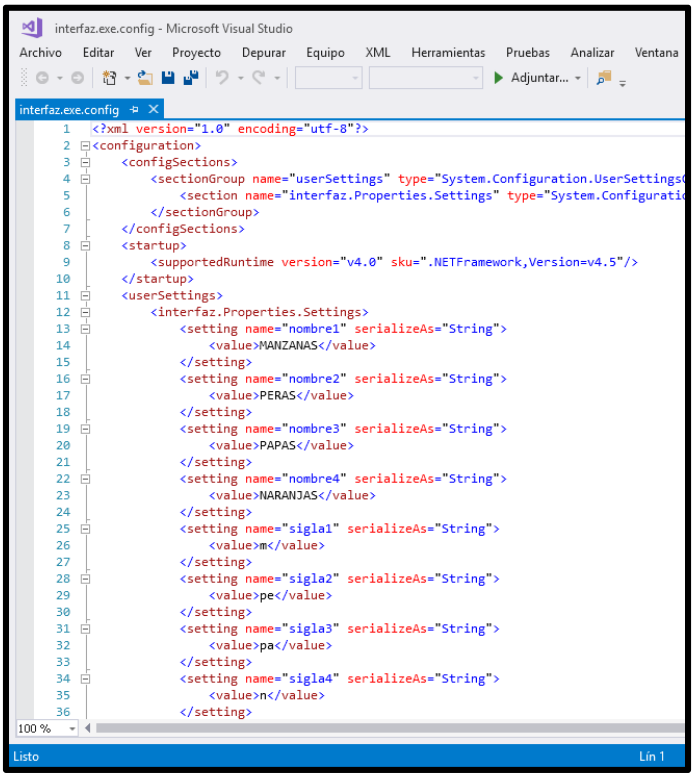


Figura 31. Código de programa para configurar nombre y serie de productos

Fuente: Elaborado por el autor

En la figura 32, se muestra la pantalla de configuración de nombres donde se asigna: el nombre del producto, identificación de código y peso mínimo para generar la señal de alarma y activar las señales sonoras y auditivas.

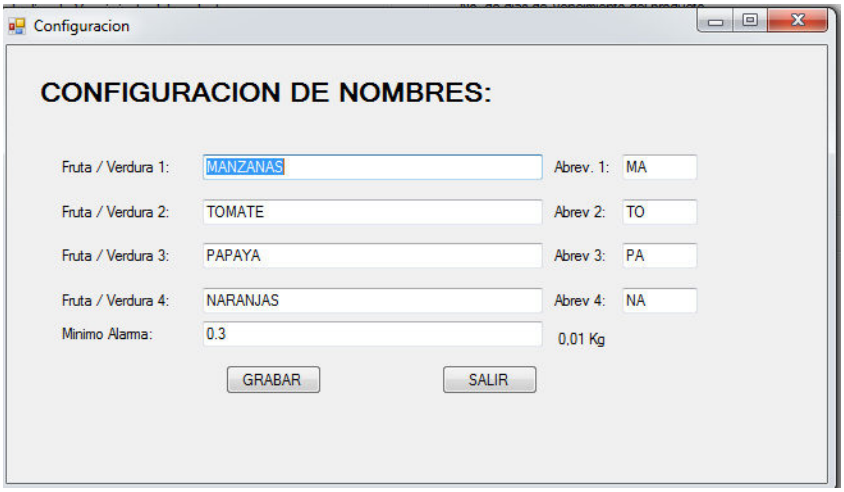


Figura 32. Pantalla de configuración de nombres

Fuente: Elaborado por el autor

3.2.3 Registro o eliminación de producto

A continuación en la figura 31, se indica el código del programa utilizado para generar el cuadro de dialogo donde indica al usuario si se quiere agregar o vender un producto basado en el serial del producto ingresado.

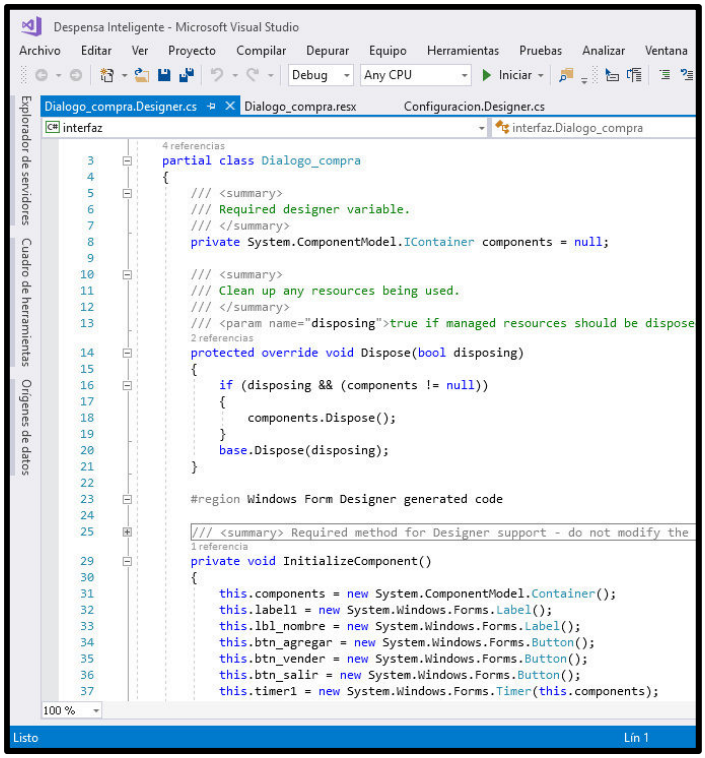


Figura 33. Código de programa de dialogo para agregar o vender un producto

Fuente: Elaborado por el autor

En la figura 34, se muestra el resultado del programa de diálogo indicado anteriormente, este cuadro permite al usuario determinar si el producto registrado va a ser agregado o vendido

Figura 34. Pantalla de dialogo para agregar o vender

Fuente: Elaborado por el autor

En la figura 35, se muestra el código del programa para agregar o eliminar un producto, cuando se agrega el producto se guarda el nombre, código, fecha de ingreso.

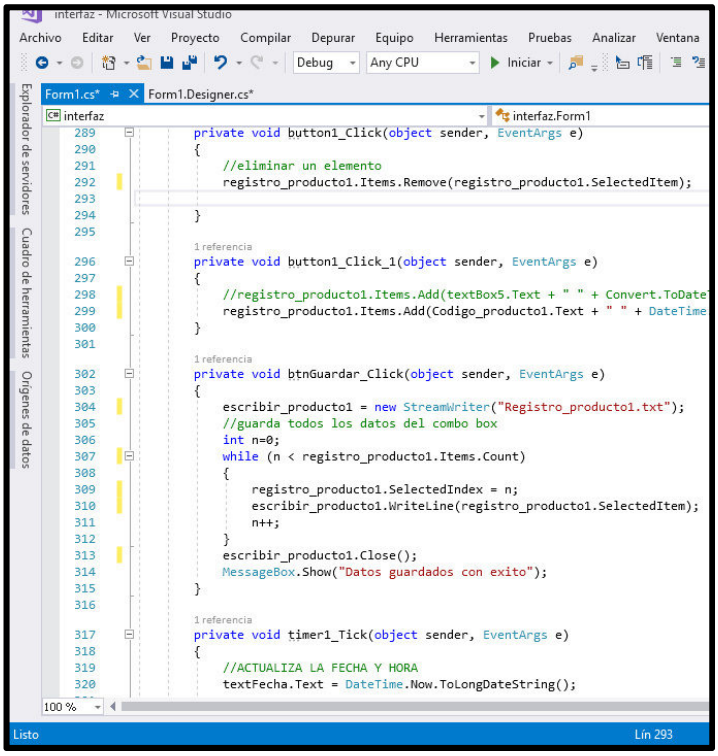


Figura 35. Código de programa para agregar o eliminar un producto

Fuente: Elaborado por el autor

En la figura 36, se muestra el código del programa donde se ingresa la serie del producto en el espacio código de barra, inmediatamente se presenta el cuadro de dialogo en el cual se debe indicar si se desea agregar o vender, una vez seleccionada la opción que se quiere realizar se guardará el registro del producto, fecha de ingreso, días de almacenamiento.

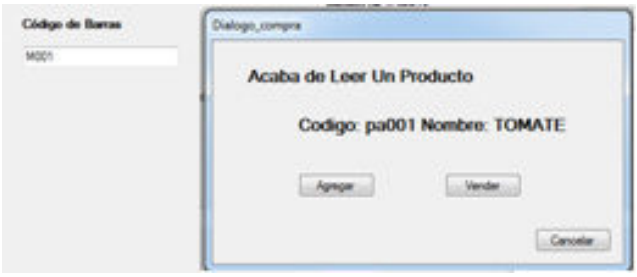


Figura 36. Cuadro de dialogo para ingreso y eliminación de un producto

Fuente: Elaborado por el autor

3.2.4 Estado del Producto

En la figura 37, se muestra el código del programa utilizado para verificar el estado del producto, este valor será controlado basado en los datos de fecha de ingreso, días de vencimiento y código de producto almacenado ingresado al momento de agregar el producto. Si detecta que el producto paso los días de almacenamiento genera una señal de "ALERTA" y activa las señales visuales y auditivas de la despensa.

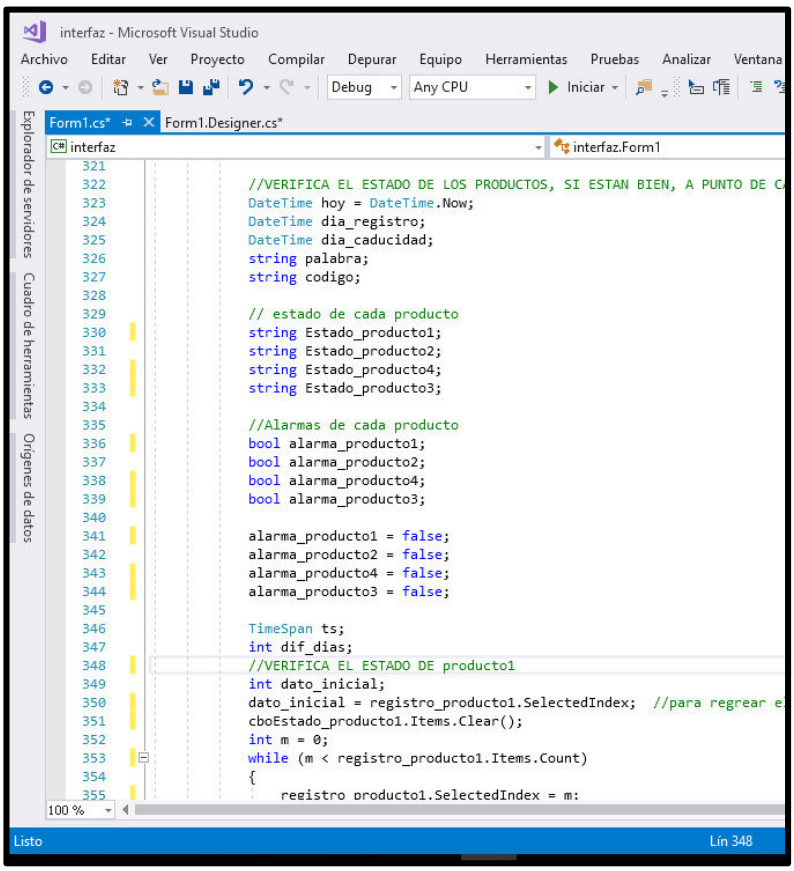


Figura 37. Código de programa para verificar el estado del producto

Fuente: Elaborado por el autor

La pantalla de estado de producto indicada en la figura 38, presenta el valor del peso que tiene el producto almacenado y los días de almacenamiento que le quedan de acuerdo a la fecha de ingreso. Si están con el peso y fecha correcta, se presentará un mensaje de "Normal", caso contrario presentara el mensaje "Alarma".

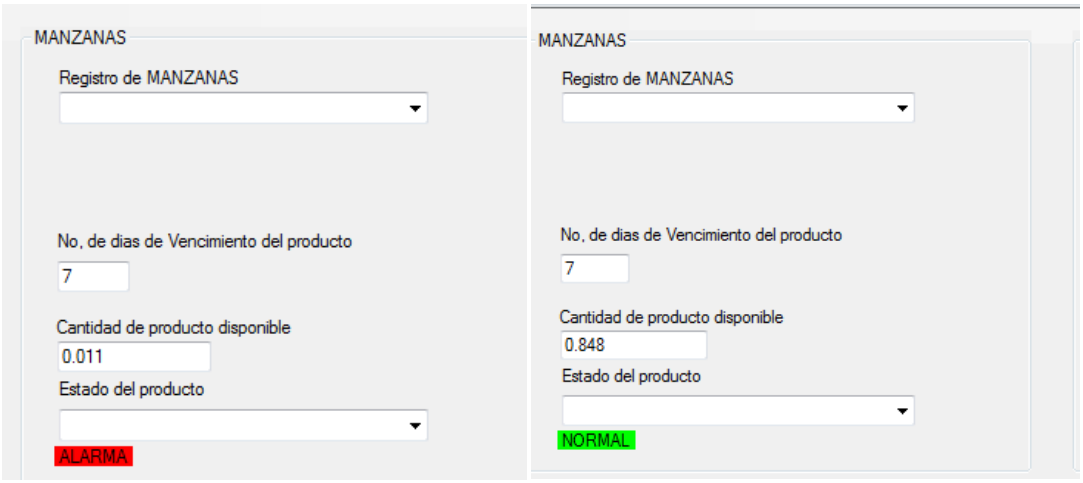


Figura 38. Pantalla de estado del producto

Fuente: Elaborado por el autor

3.3 Pruebas de funcionamiento

Con el fin de verificar el funcionamiento adecuado de la despensa se realizaron pruebas de funcionamiento tanto de medición de peso como de verificación de la fecha de caducidad del producto. Utilizando un peso conocido para cada una de las 4 gavetas disponibles. Además, se verificó que se registren adecuadamente los nombres de los productos y códigos.

Previamente, para realizar las pruebas primero se debe correr la hoja de puesta a punto detallada en el anexo b con el fin de garantizar las conexiones correctas del prototipo.

En la figura 39, se muestra la pantalla del aplicativo con la despensa vacía, en la que se presenta un mensaje de “ALARMA” en las 4 gavetas disponibles, paralelamente se encendieron los leds que sirven de alerta visual y el zumbador que sirve como alerta auditiva. Las alertas se apagan cuando se coloca un peso mayor al configurado realizando previamente una verificación del estado del producto.

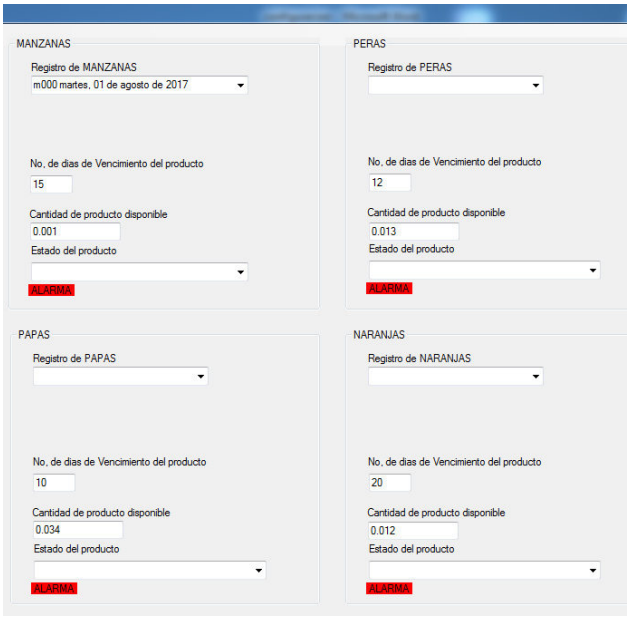


Figura 39 Pantalla con despena vacía

Fuente: Elaborado por el autor

La figura 40, indica la pantalla de configuración de nombres en la cual se asignó a la gaveta 1 el producto manzana, en la gaveta 2 el producto papaya, en la gaveta 3 el producto tomate y en la gaveta 4 el producto naranjas. También, se muestra las iníciales con la cuales se identificarán a cada uno de los productos mediante el código de barras de la siguiente manera: manzanas: MA, papaya: PA, tomate: TO, naranjas: NA.

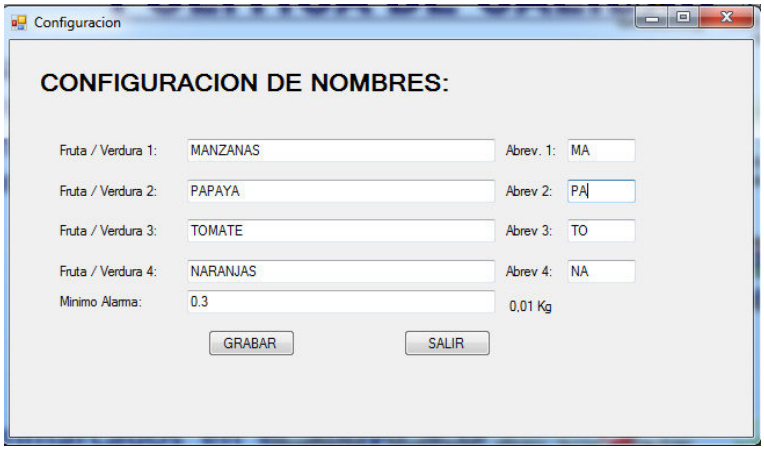


Figura 40. Pantalla de Configuración de nombres

Fuente: Elaborado por el autor

En la figura 41, se muestra la despensa colocada productos en cada una de sus gavetas.

Figura 41. Despensa con producto colocado en cada gaveta

Fuente: Elaborado por el autor

A continuación, en la figura 42, se muestra un cuadro de diálogo que servirá para realizar el ingreso o venta del producto de acuerdo al condigo configurado al inicio. En esta ventana se le asigna una gaveta a cada producto, adicionalmente se registran los parámetros iníciales como son: días de almacenamiento, fecha y lote de ingreso de acuerdo al código asignado.

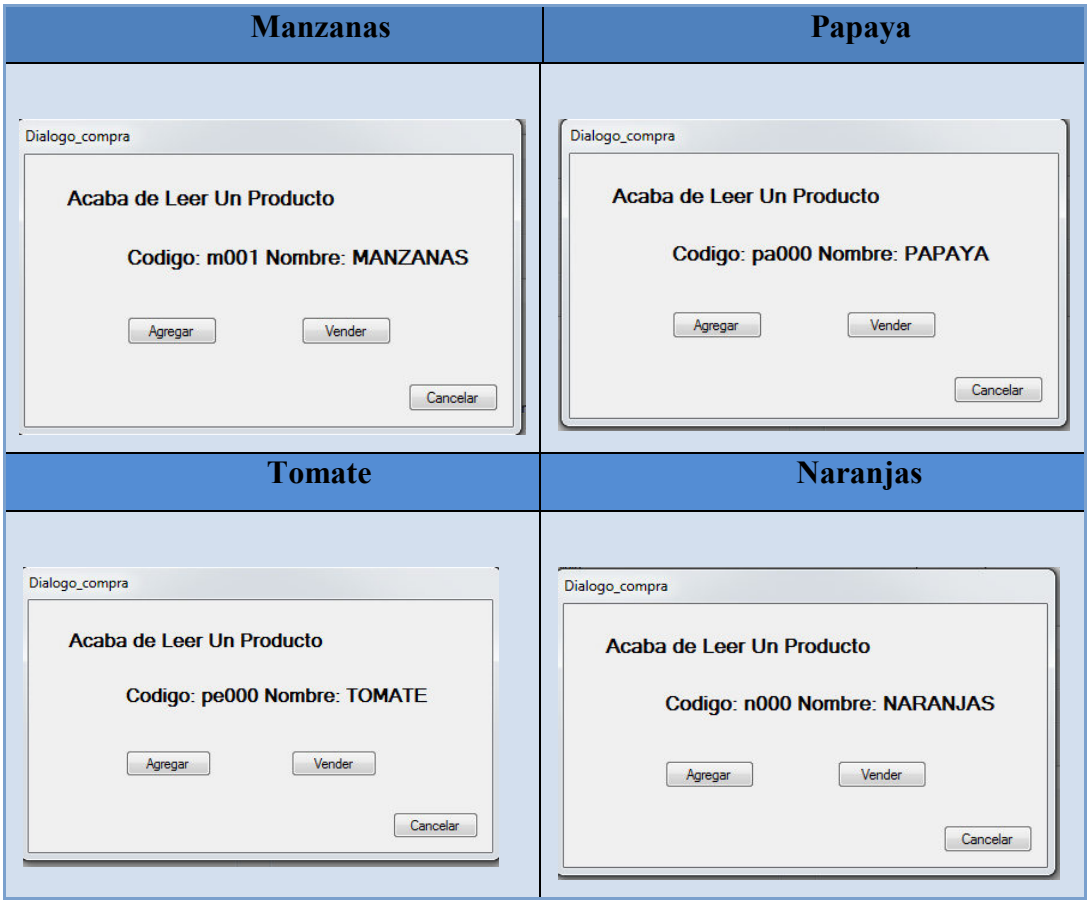


Figura 42. Pantalla de ingreso o venta del Producto a cada gaveta

Fuente: Elaborado por el autor

3.3.1 Pruebas de registro del tiempo máximo de consumo

En las figuras 43, 44, 45 y 46 se muestran los tiempos de almacenamiento máximo que le queda a los productos en cada gaveta basándose en el número de días configurados para almacenarse cuando se registro tomado como base la fecha de ingreso del producto.

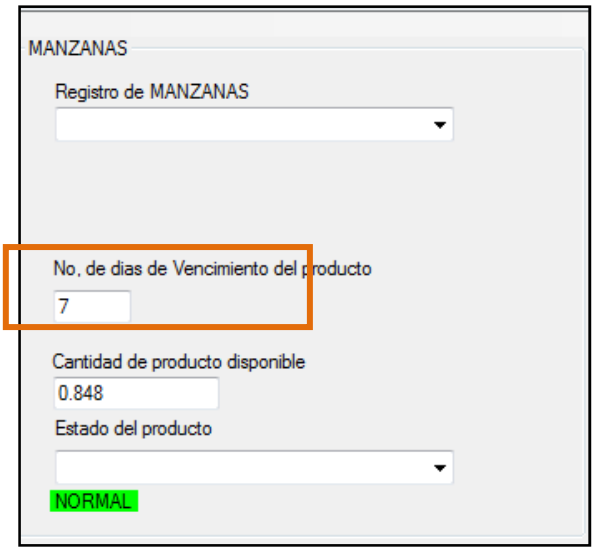


Figura 43 Pantalla de tiempo límite de almacenamiento de manzanas

Fuente: Elaborado por el autor

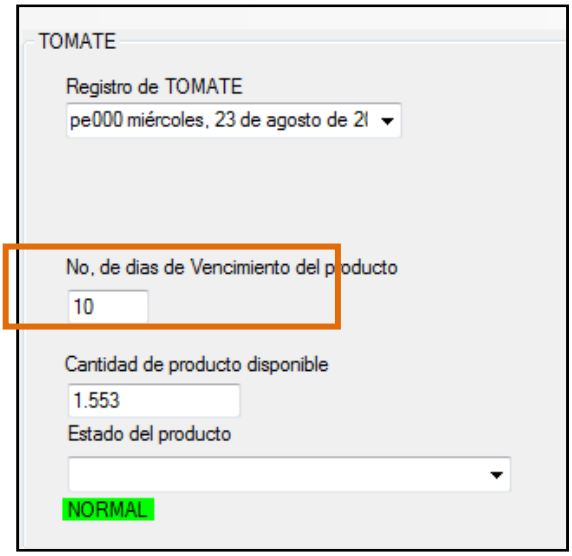


Figura 44. Pantalla de tiempo límite de almacenamiento de tomate

Fuente: Elaborado por el autor

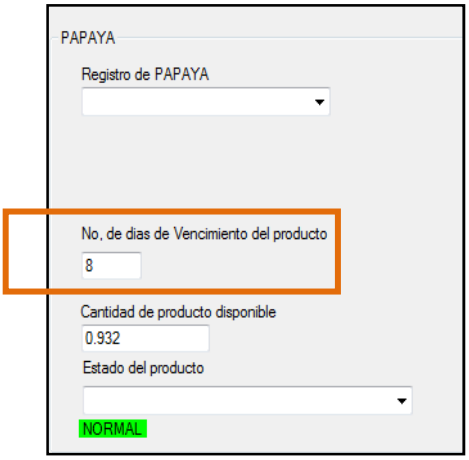


Figura 45. Pantalla de tiempo límite de almacenamiento de papaya

Fuente: Elaborado por el autor

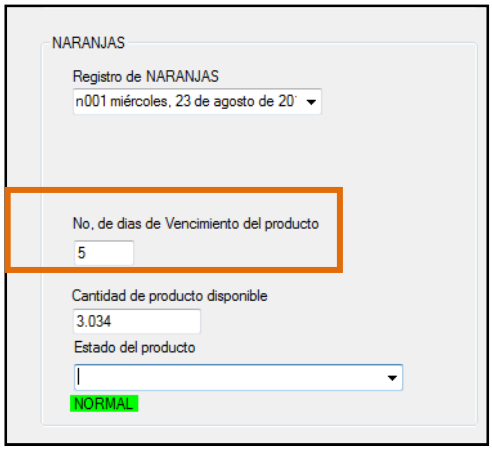


Figura 46. Pantalla de tiempo límite de almacenamiento de naranjas

Fuente: Elaborado por el autor

3.3.2 Pruebas de comprobación de peso

Se realizarán pruebas de comprobación de peso para verificar que las celdas de carga se encuentren calibradas y que no arrojen resultados erróneos en la medición que puedan influir en los resultados de las alertas entregadas por el prototipo.

En las figuras 47, 48, 49 y 50 se muestran cómo van variando los pesos hasta retirar todo el producto almacenado dejando la gaveta vacía, se verifica que se encienden los leds indicadores y el zumbador.

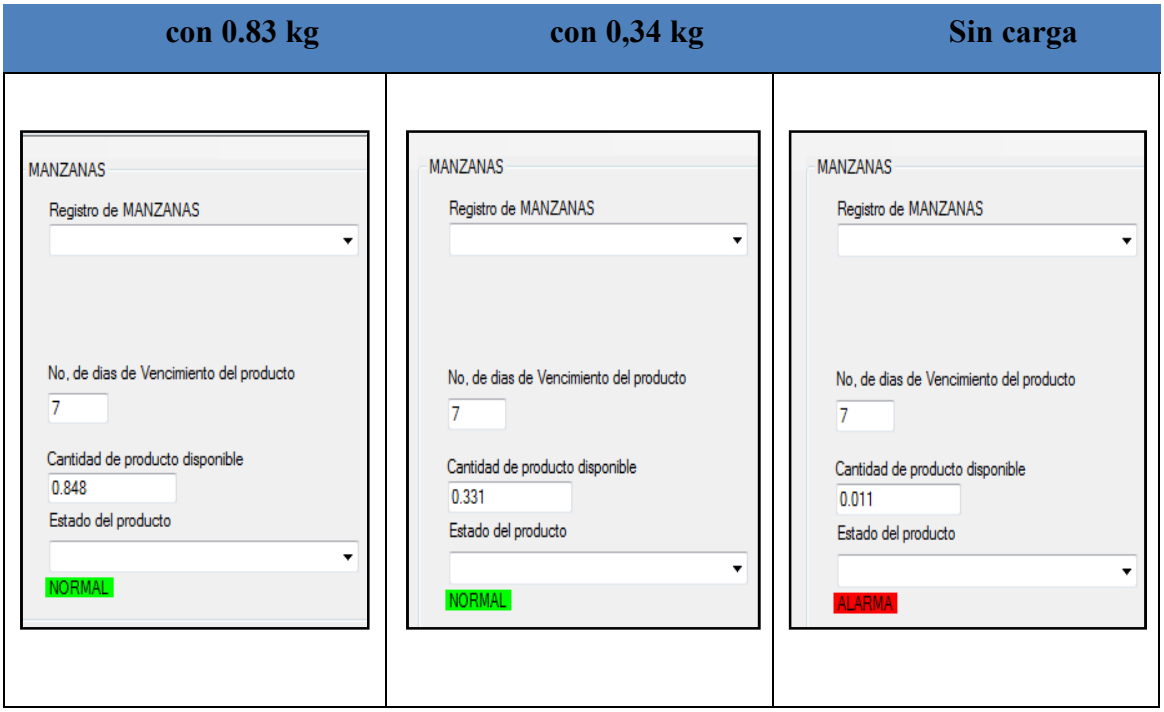


Figura 47. Pantallas de Pesos para el producto Manzanas

Fuente: Elaborado por el autor

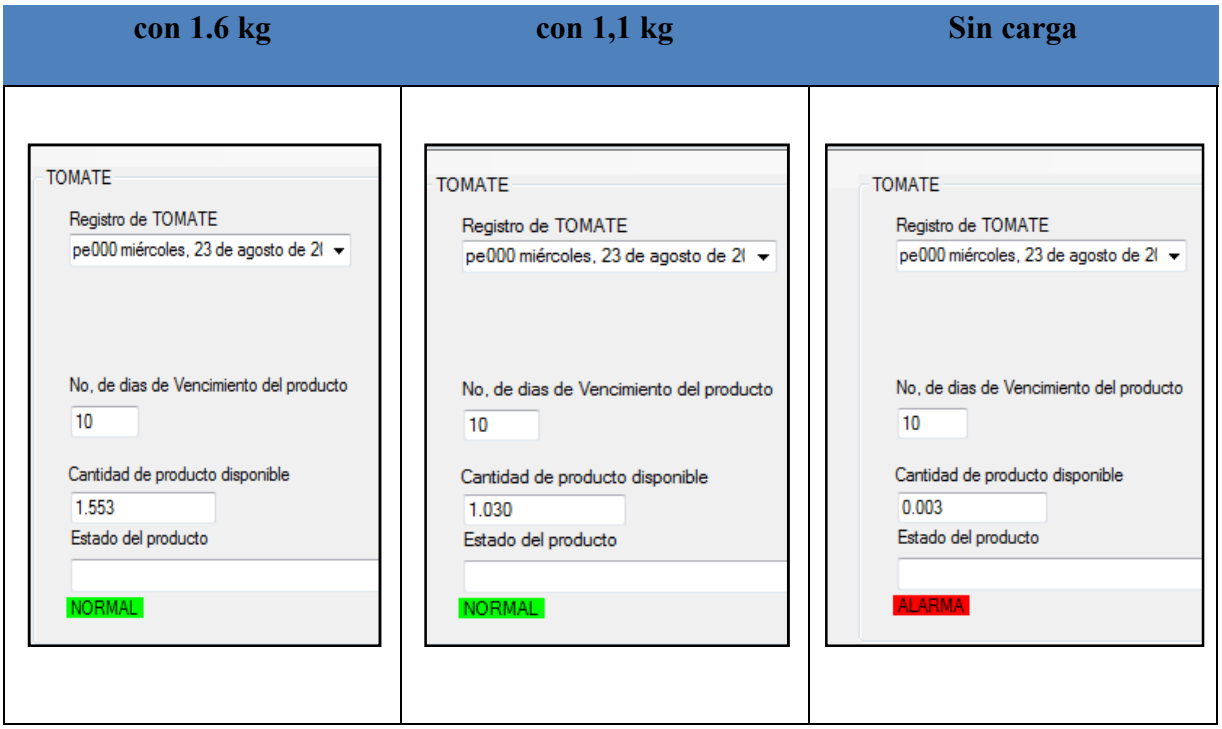


Figura 48. Pantalla de Pesos para el producto Tomate

Fuente: Elaborado por el autor

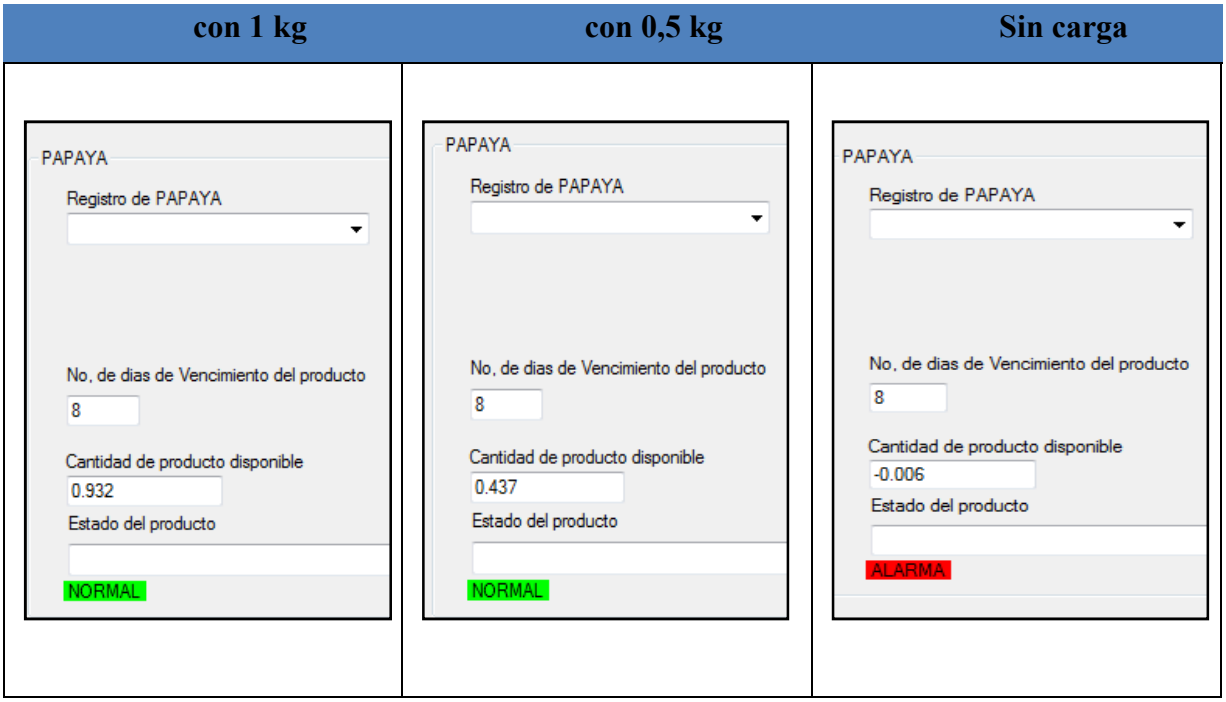


Figura 49: Pantalla de Pesos para el producto Papaya

Fuente: Elaborado por el autor

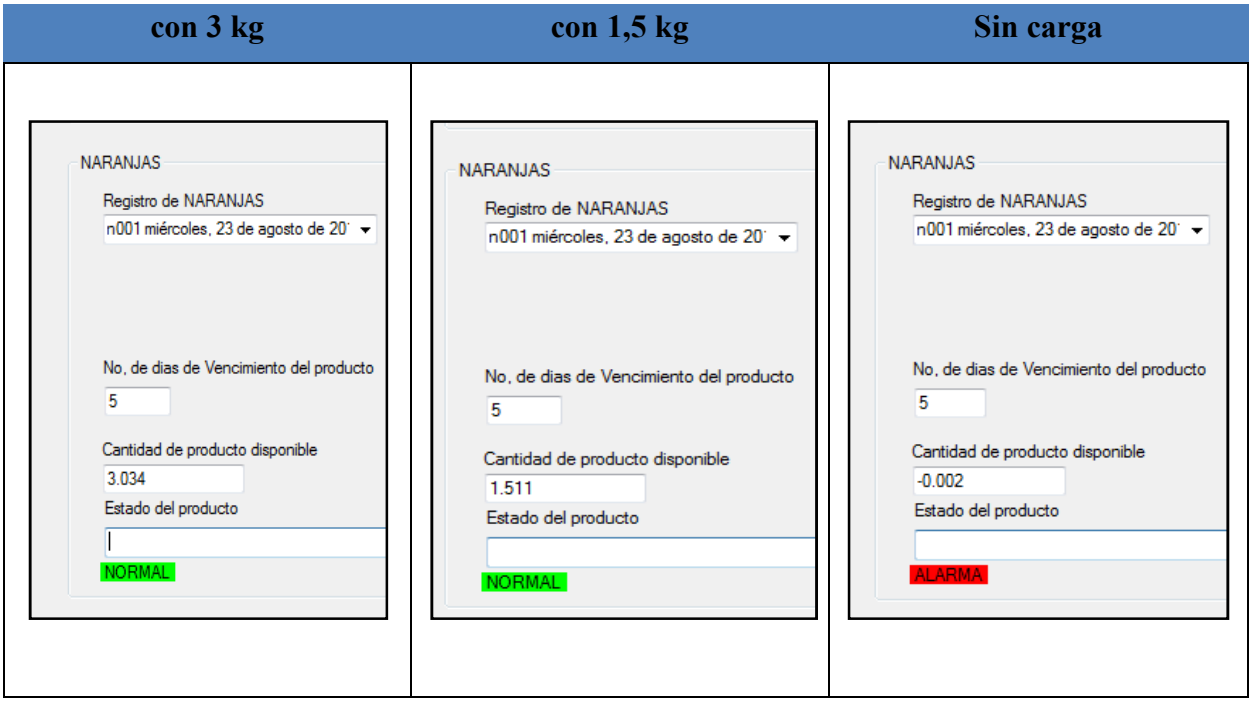


Figura 50: Pantalla de Pesos para el producto Naranja

Fuente: Elaborado por el autor

En la tabla 10, se indican los valores presentados al realizar las pruebas en los 4 productos que se colocaron en la despensa inteligente.

Tabla 10: Pruebas de medición de peso del producto

PRUEBA DE MEDICIÓN DE PESO DEL PRODUCTO EN kg			
Producto	**Peso 1**	**Peso 2**	**Vacío**
Manzanas	0.848	0.331	0.011
Tomate	1.553	1.03	0.003
Papaya	0.932	0.437	-0.006
Naranja	3.034	1.511	-0.002

Fuente: Elaborado por el autor

De acuerdo a los resultados obtenidos en las pruebas de registro de tiempo máximo de consumo y comprobación del peso, se observa que el prototipo cumple con lo propuesto con lo que respecta al desabastecimiento y fecha límite de consumo de los víveres almacenados de acuerdo a lo parametrizado.

3.3.3 Pruebas de caducidad del producto

Para realizar las pruebas de caducidad del producto se verificó la fecha de ingreso de los números de lotes que se encontraban registrados y dependiendo a los días de vigencia configurados arrojaron como resultado los lotes que superaron las fecha límite de almacenamiento como se muestra en la figura 51.

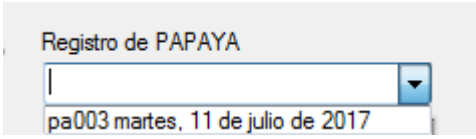


Figura 51. Numero de lote caducado

Fuente: Elaborado por el autor

Para la validación de la fecha límite de consumo toma como referencia la fecha configurada en el sistema del computador y toma el número de días establecidos en la

aplicación y que depende de cada uno de los productos. En este caso las pruebas se realizaron el 14 de agosto de 2017 y el producto se registró el 11 de julio de 2017 con una vigencia de 8 días con lo que se tiene 26 días de expiración del lote. Mientras en el aplicativo estuvieron registrados productos pasados los días límites de consumo, se activó el zumbador y se encendió el led rojo colocado en la gaveta con producto caducado.

En la figura 52, se muestra la señal de alarma activada en el aplicativo a pesar de tener en la gaveta un producto con peso, esto debido a que existe un lote que ya pasó su tiempo máximo de consumo lo que evita que la alerta sea desactivada hasta que se retire físicamente y del sistema el lote caducado. Los diodos led y el zumbador siguieron sonando mientras permaneció esta condición.

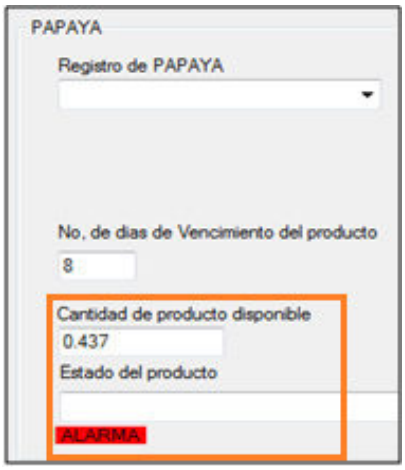


Figura 52. Despensa con producto caducado

Fuente: Elaborado por el autor

3.3.4 Pruebas de Operatividad

En la tabla 11, se indica el porcentaje de error que tiene cada gaveta aplicando la ecuación 2 indicada a continuación:

$$e_{abs} = fm - fr$$

Ecuación 2. Ecuación para calcular el error absoluto

Fuente: (Peña, 2007)

Dónde: fm = valor medido

fr = valor real

Tabla 11: Tabla de Error en la medición

Producto	V. Med 1	V. Real 1	Error 1	V. Med 2	V. Real 2	Error 2
Manzanas	0.848	0.830	0.018	0.331	0.340	-0.009
Tomate	1.553	1.600	-0.047	1.030	1.100	-0.07
Papaya	0.932	1.000	-0.068	0.437	0.500	-0.063
Pepino	3.034	3.000	0.034	1.511	1.500	0.011

Fuente: Elaborado por el autor

3.4 Análisis de resultados

El prototipo está diseñado para conocer cuando un producto se agotó, por lo que se utilizó la variable del peso con el que se comprueba que el producto este presente cuando tiene un valor superior a 0,2 kg, para realizar las pruebas se utilizo pesos conocidos de 0,83; 0,34; 1,6; 1,5; 1,1; 1; 0,5 y 3 todas las medidas en kilogramos, luego de realizadas las mediciones, el margen de error en la medición de acuerdo a los resultados obtenidos en la tabla 11 indican que el valor minino de error es de -0,068 kg y el valor máximo es de 0,034 kg lo que no afectaría el resultado ya que dependiendo del producto a ser colocado se configura el valor mínimo para generar las alertas.

En el prototipo se plantea implementar una alerta cuando el producto expiró, esto se controló con la variable tiempo que fue diseñada con el software “Despensa Inteligente “, en el que se registraron los lotes basados en una serie por cada producto indicando la fecha de ingreso y limite de consumo. De las pruebas realizadas se verifica que en el producto Papaya existió un lote ingresado el 11 de julio de 2017 con una vigencia de 8 días con lo que se tiene 26 días de expiración del lote corroborando que las alertas se activaron a pesar de tener el peso de 0,437 kg.

Tabla 12. Tabla de Productos Expirados.

LOTE	DÍAS DE VENCIMIENTO	FECHA DE INGRESO	FECHA MÁXIMA	PESO	ESTADO PRODUCTO
MA000	7	23-ago	30-ago	0.33	ALARMA
TO002	10	04-oct	14-oct	1.03	NORMAL
PA001	8	11-jul	19-jul	0.44	ALARMA
NA000	5	23-ago	28-ago	1.51	ALARMA

Fuente: Elaborado por el autor

Se comprobó adicionalmente, que cuando se retiró el producto caducado del sistema se desactivaron las alertas. Basados en los resultados anteriores se puede decir que el prototipo cumplió con lo planificado.

4. Conclusiones:

Luego de concluido el trabajo de titulación y previo a un análisis y discusión de los resultados, a continuación se presentan las conclusiones del presente estudio, las cuales permiten poner en realce los manifestaciones más importantes encontradas en la actual investigación. Las mismas se detallan con base en los objetivos específicos que se formularon para la investigación.

Con respecto al primer objetivo específico, que consistió en diseñar un prototipo de despensa inteligente para controlar el abastecimiento de víveres en un supermercado, mediante el control de las variables de peso y tiempo, a través del uso de celdas de carga, registros por lotes y una placa Arduino, se utilizaron celdas de carga en conjunto con un módulo HX711 los cuales permitieron verificar la presencia del producto realizando la medición del peso, adicionalmente se pudo comprobar la fecha de expiración con aplicativo creado para ingresar los productos, guardar su fecha máxima de consumo y enviar alerta cuando el límite de consumo caducó, tanto las partes de hardware como software fueron controladas por una placa Arduino.

Cabe destacar que la tarjeta Arduino es una herramienta muy buena ya que por su hardware permite incorporar todo tipo de señales ya sean estas análogas y digitales en una sola placa generando circuitos más simples al momento de la construcción y adicionalmente permitiendo trabajar con un software de fácil uso. Así mismo el uso de una librería compatible con la placa Arduino y el módulo HX711 permitieron aprovechar el uso del software Arduino facilitando la programación en conjunto, lo que además preparo la calibración de las celdas de carga.

También se utilizaron leds indicadores son visualizadores y en el caso de este prototipo ayudaron a generar una alerta para el abastecimiento del producto y un zumbador por su funcionalidad de emitir señales sonoras permitió generar alertas auditivas cuando alguna gaveta de la despensa se encontraba desabastecida, advirtiendo de esta manera a la persona que tenga que realizar la dotación de productos.

En relación con el segundo objetivo específico relativo a implementar un software que permita visualizar las alertas del peso y el tiempo en un computador que estará conectado vía serial a la placa Arduino, se creó un software que se llamó "Despensa Inteligente", el cual interactuó con las señales que entrega la placa Arduino permitiendo visualizar las alertas del peso y el tiempo en un computador conectado vía serial.

Los resultados de las pruebas realizadas revelan el funcionamiento adecuado del mismo en relación a los componentes del hardware utilizado ya que el acoplamiento entre estos y el software se realizó de manera exitosa, lo cual demuestra el cumplimiento de este objetivo. Con relación al tercer objetivo específico referente a construir el prototipo de despensa inteligente diseñado con comunicación a la aplicación para visualizar las alertas, se encontró que el mismo se pudo realizar de manera favorable, utilizando los materiales derivados del diseño y el software creado.

Finalmente, con relación al cuarto objetivo específico referente a realizar pruebas de validación y funcionamiento del prototipo, se observa que las mismas se pudieron verificar, ya que se registraron correctamente los productos, el peso entregado por la aplicación es el correcto con su margen de tolerancia y las alertas fueron

activadas cuando se cumplen las condiciones de ausencia de producto y fecha de expiración, confirmando su correcto funcionamiento con relación a lo planificado.

Es de hacer notar que basados en pesos conocidos se realizaron pruebas de comprobación obteniendo margen de un error minino de -0,068 kg y máximo de 0,034 kg en cada una de las gavetas lo que no afecta a los resultados esperados ya que para la activación de las alertas está determinado tener un valor inferior a 0,3 kg.

Los resultados mostrados presentan la importancia y funcionabilidad del prototipo desarrollado, ya que con la implementación del mismo, resuelve la problemática de la rotación de productos en un supermercado.

Adicionalmente, cabe indicar que el usuario puede utilizar una comunicación serial en lugar de una comunicación con Bluetooth o Wifi para transmitir los datos a dispositivos Android con la ventaja de que visualice las alertas en sus dispositivos

5. Recomendaciones

- Cuando se realice la instalación de las celdas de carga se debe tener en cuenta el lugar donde se debe colocar el elemento a ser medido para que puede realizar una lectura correcta, la mala ubicación puede causar una deformación en las celdas y arrojar datos erróneos.

- Las alertas entregadas por la despensa pueden ser conectadas a display o pantallas si el usuario lo requiere facilitando una mejor visualización de resultados. Se debe realizar previamente la configuración de los dispositivos a ser utilizados en el aplicativo y además verificar las características técnicas de alimentación de los componentes.

- Para un uso industrial de la despensa inteligente, pueden ser instaladas las celdas de carga en un material más robusto que se adapten al presupuesto y a las necesidades propias de cada usuario teniendo en cuenta las especificaciones del fabricante.

- El aplicativo se lo podría adaptar para generar una base de datos con los registros, con esta se puede obtener un listado de los productos agotados, próximos a caducarse, los que se encuentran caducados.

- Cuando se realiza una conversión de datos tipo doublé se debe verificar si el computador trabaja con el carácter de punto o con coma con el fin de evitar fallas en la aplicación cuando se realice la conversión de datos ya que utiliza directamente el código fuente.

- Cuando se realice el cambio de nombres en los productos, se debe verificar previamente que no exista ningún producto asignado a esa gaveta con el anterior nombre y código ya que esto puede generar que no se pueda eliminar el producto anteriormente configurado cuando sobrepase su fecha límite de almacenamiento lo que ocasionará que no se apaguen las alertas.

- Se debe tomar en cuenta el tiempo de vigencia que tendrá cada uno de los productos a ser almacenados y el mismo deberá ser previamente configurado antes de conectar el aplicativo para que puedan generarse las alarmas en base a la fecha proporcionada por el sistema del computador.

- Cada vez que se ponga a funcionar el prototipo se debe previamente verificar todo lo detallado en la hoja de puesta a punto del aplicativo indicada en el anexo b, la cual ayudará con la verificación de todas las conexiones de cables y dispositivos que influyen directamente en el funcionamiento

6. Bibliografía

- Arduino. (2010). *https://www.arduino.cc/*.
- BIGTRONICA. (2017). *BIGTRONICA*. Recuperado el 2017, de http://bigtronica.com/sensores/406-celda-de-carga-10kg-5053212004064.html
- Carretero. (2009). *Electricidad y Electrónica.*
- Dictionary, T. f. (2017). *The free dictionary*. Recuperado el 05 de 08 de 2017, de http://es.thefreedictionary.com/tara
- Electrónica. (2017). *Celdas de carga.* Obtenido de http://www.electrrio.com/Anuncios/SensoresyModulos/Peso/sensordepeso.pdf

- Electrónica, A. (2017). *ABC Electrónica*. Recuperado el 24 de Julio de 2017, de www.abcelectronica.net
- Electrónicas, D. (2013). *www.didacticaselectronicas.com*. Obtenido de www.didacticaselectronicas.com
- GeekFactory. (2017). *Geek Factory*. Recuperado el 05 de 08 de 2017, de https://www.geekfactory.mx/
- Granda, M. (2015). *Instrumentación Electrónica: Transductores y acondicionadores de señal.* Cantabria: de la Universidad de Cantabria.
- Haeussler. (2003). *Matemáticas para administración y economía.* México: PEARSON EDUCACION.
- Martin, G. (2016). *(Des)empleo y bienestar en la era digital.* Barcelona: Thinkstock.
- Mechatronics, N. (2012). *http://www.naylampmechatronics.co*.
- Moises, E. (Marzo de 1995). *Diseño y contrucción en celdas de carga.* Obtenido de http://cdigital.dgb.uanl.mx/te/1020074697.pdf
- Peña, A. G. (2007). *Introducción a errores en la medición.* Bogota: Instituto Tecnologico Metropolitano.
- Pérez, E. M. (2009). *Manual de prácticas de electrónica digital.*
- ROBOTSHOP. (2017). *ROBOTSHOP.* Obtenido de http://www.robotshop.com/media/files/PDF/datasheet-3133.pdf
- Rodriguez, M. A. (2017). *Personales Unican.* Recuperado el 31 de julio de 2017, de http://personales.unican.es/rodrigma/PDFs/Puente%20de%20Wheatstone.pdf
- Sánchez, A. (2006). *Instrumentación y control avanzado de procesos.* Madrid: Edigrafos, S.A.
- Sánchez, A. (2013). *Instrumentacion y control básico de procesos.* Madrid: Díaz de Santos, S.A.
- TodoHard. (2007). *TodoHard*. Obtenido de http://todohard.awardspace.com/calc/resist/wheatstone/wh.htm

7. Anexos

ANEXO A.
COSTO DEL PROTOTIPO

En la tabla 17, 18 y 19 se detallan los componentes utilizados para la realización del presente prototipo.

Tabla 13: PRESUPUESTO DE COMPONENTES ELECTRÓNICOS

COSTO ELECTRÓNICO			
DESCRIPCIÓN	**QTY**	**V. UNIT**	**V. TOTAL**
Resistencia	4	0.02	0.08
Zumbador	1	0.65	0.65
Diodo led	4	0.08	0.32
HX711 + celda de carga	4	19.00	76.00
Arduino Mega	1	18.00	18.00
Molex 4 Pines	8	0.65	5.20
Baquelita	1	10.00	10.00
TOTAL			**110.25**

Fuente: Elaborado por el autor

Tabla 14: PRESUPUESTO DE COMPONENTES MECÁNICOS

COSTO MECÁNICO			
DESCRIPCIÓN	**QTY**	**V. UNIT**	**V. TOTAL**
Maqueta	1	80.00	80.00
TOTAL			**80.00**

Fuente: Elaborado por el autor

Tabla 15: PRESUPUESTO TOTAL DEL PROTOTIPO

COSTO TOTAL			
DESCRIPCIÓN	**QTY**	**V. UNIT**	**V. TOTAL**
Electrónico	1	80.00	80.00
Mecánico	1	110.25	110.25
TOTAL			**350.25**

Fuente: Elaborado por el autor

ANEXO B. Hoja de chequeo utilizada para realizar las pruebas

HOJA DE CHEQUEO DEL PROTOTIPO

SUPERVISOR:_____

RESPONSABLE:_____

FECHA:_____

ITEM	DETALLE	FUNCIONAMIENTO	
		CORRECTO	INCORRECTO
	HARDWARE		
1	Placa Arduino encendida		
	SOFTWARE		
2	Aplicativo despensa inteligente abierto		
3	Puerto de Arduino detectado y abierto		
4	Tiempo de vigencia configurado		
	VISUALIZADORES		
5	Diodo led en cada una de las gavetas de la despensa		
6	Zumbador colocado en placa PCB		
	CONEXIONES		
7	Cable USB conectado a placa Arduino		
8	Cable USB conectado a computador		
9	Terminales de conexión de leds conectados a la placa		
10	Terminales de conexión de módulo HX711 conectados a la placa		

ANEXO C. Hoja de chequeo del prototipo

HOJA DE CHEQUEO DEL PROTOTIPO

RESPONSABLE:_____

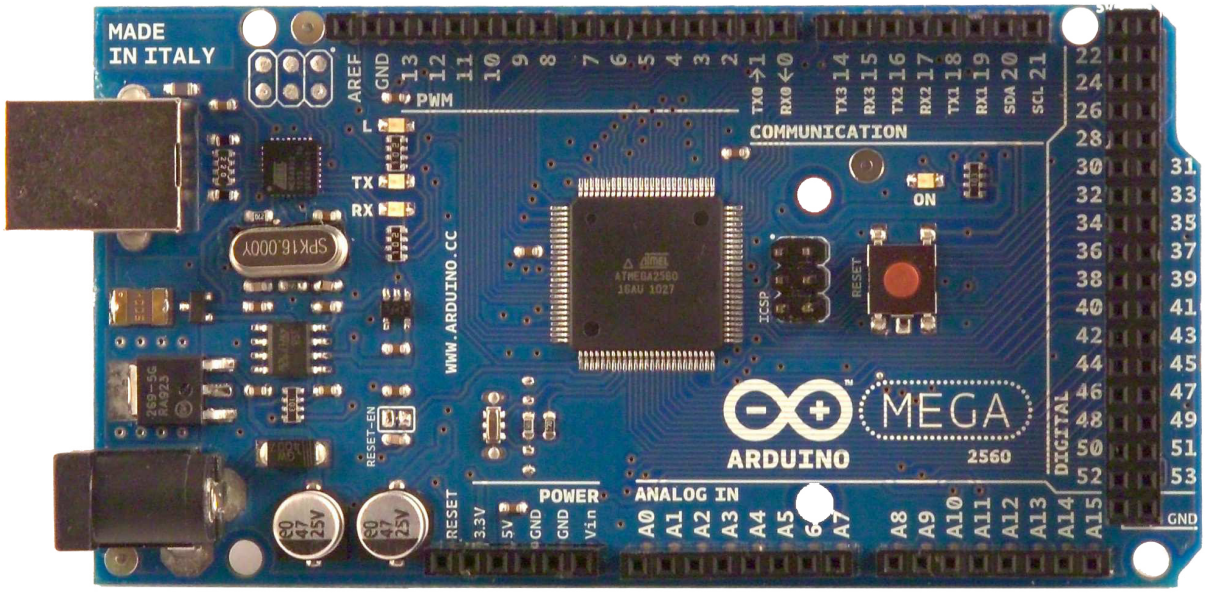
SUPERVISOR:_____

FECHA:_____

ITEM	DETALLE	FUNCIONAMIENTO	
		CORRECTO	INCORRECTO
	HARDWARE		
1	Placa Arduino encendida		
	SOFTWARE		
2	Aplicativo despensa inteligente abierto		
3	Puerto de Arduino detectado y abierto		
4	Tiempo de vigencia configurado		
	VISUALIZADORES		
5	Diodo led en cada una de las gavetas de la despensa		
6	Zumbador colocado en placa PCB		
	CONEXIONES		
7	Cable USB conectado a placa Arduino		
8	Cable USB conectado a computador		
9	Terminales de conexión de leds conectados a la placa		
10	Terminales de conexión de módulo HX711 conectados a la placa		

ANEXO D

Arduino MEGA 2560



Product Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 (datasheet). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Index

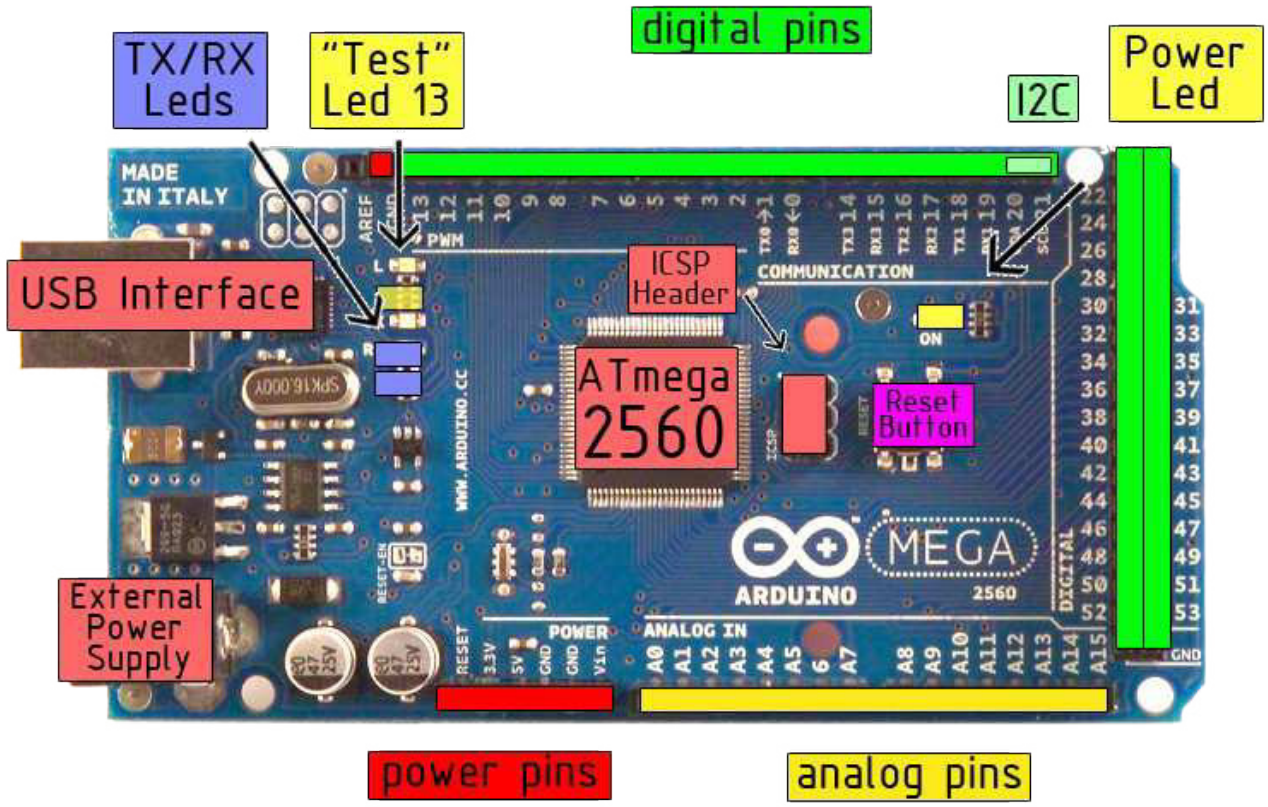
Technical Specification

EAGLE files: arduino-mega2560-reference-design.zip Schematic: arduino-mega2560-schematic.pdf

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

the board



Power

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the analogWrite() function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I^2C: 20 (SDA) and 21 (SCL).** Support I^2C (TWI) communication using the Wire library (documentation on the Wiring website). Note that these pins are not in the same location as the I^2C pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analogReference() function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analogReference().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Mega's digital pins.

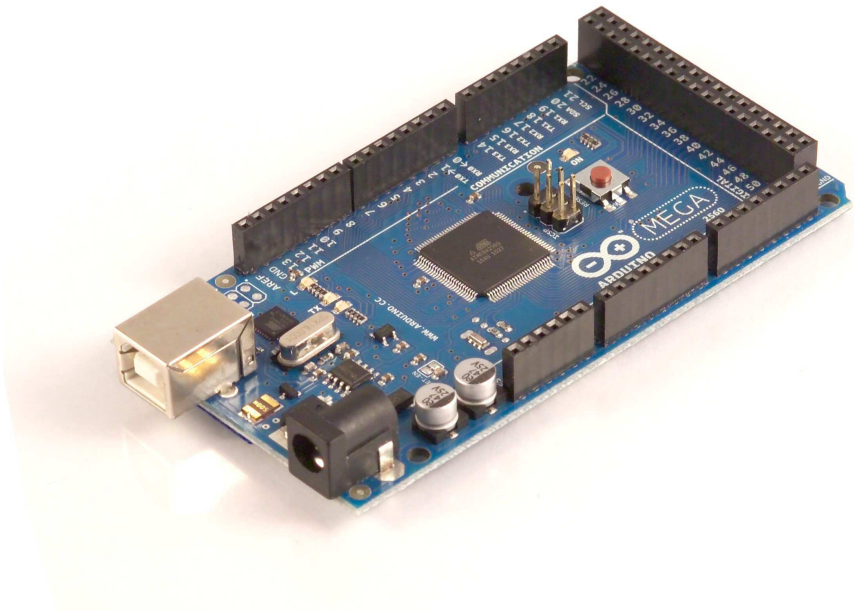
The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation on the Wiring website for details. To use the SPI communication, please see the ATmega2560 datasheet.

Programming

The Arduino Mega2560 can be programmed with the Arduino software (download). For details, see the reference and tutorials.

The Atmega2560 on the Arduino Mega comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.



Automatic (Software) Reset

Rather then requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

USB Overcurrent Protection

The Arduino Mega has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Diecimila. **Please note that I^2C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).**

How to use Arduino

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platoform program. You'll have to follow different instructions for your personal OS. Check on the Arduino site for the latest instructions. *http://arduino.cc/en/Guide/HomePage*

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

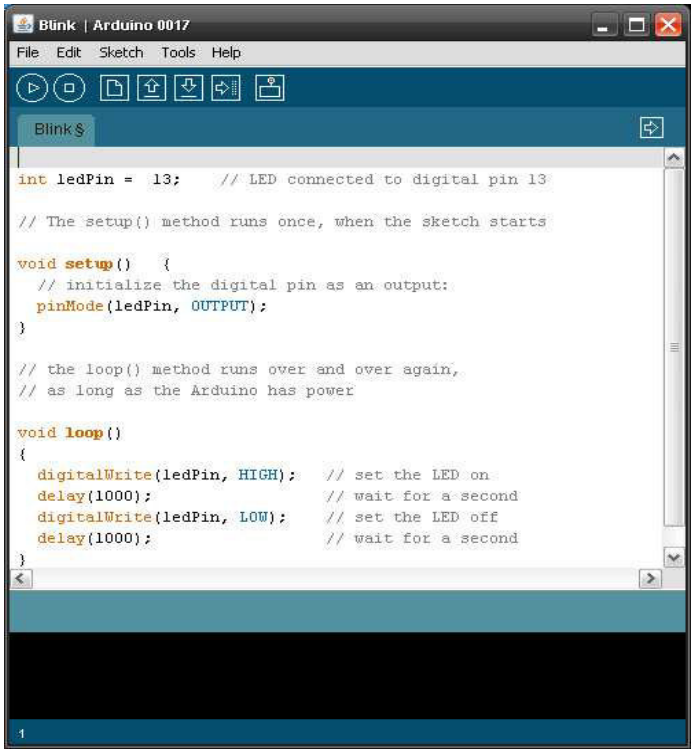
Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

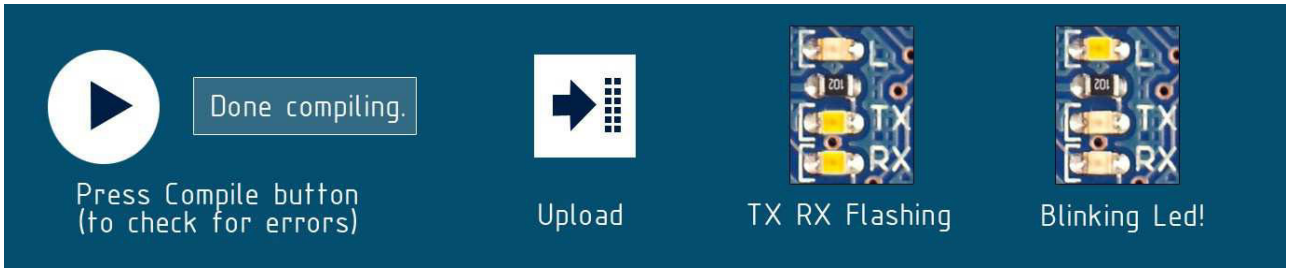
File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink

Once you have your skecth you'll see something very close to the screenshot on the right.

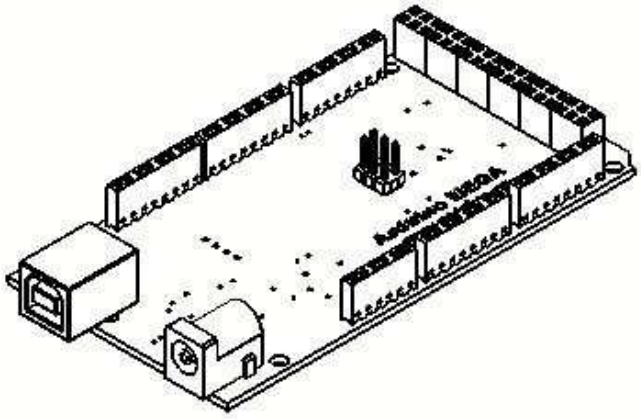
In **Tools>Board** select MEGA

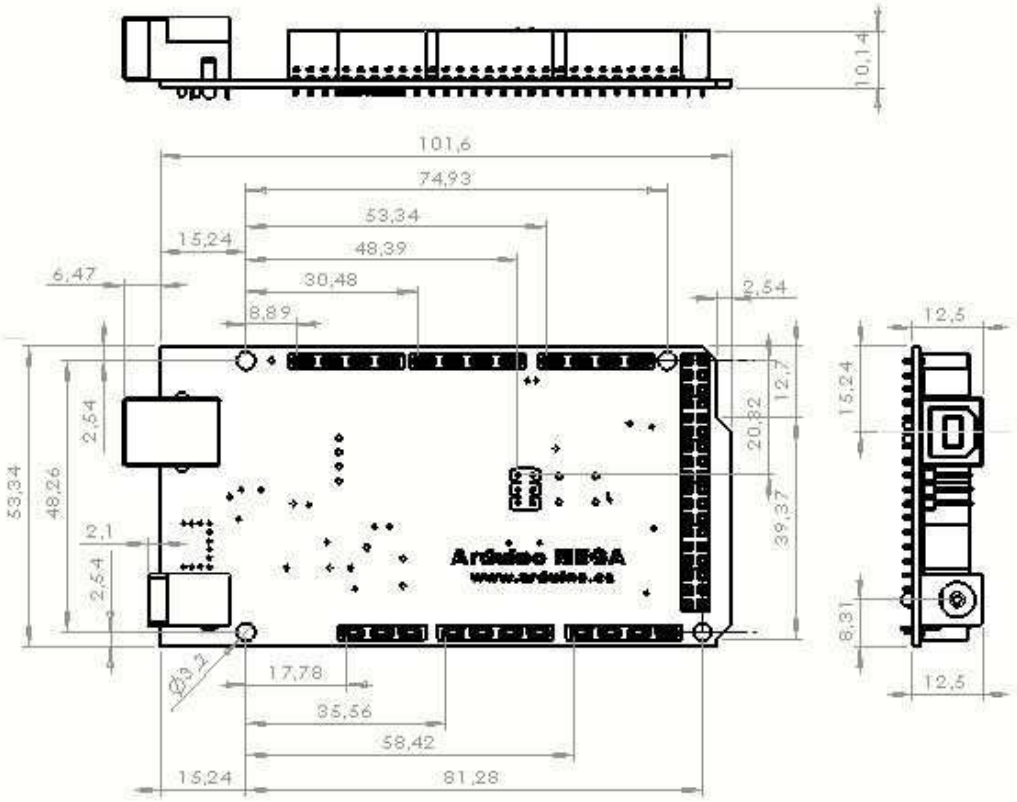
Now you have to go to
Tools>SerialPort
and select the right serial port, the one arduino is attached to.





Dimensioned Drawing





Terms & Conditions

1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

Enviromental Policies

The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.it. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.

ANEXO E

24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales

DESCRIPTION

Based on Avia Semiconductor's patented technology, HX711 is a precision 24-bit analog-to-digital converter (ADC) designed for weigh scales and industrial control applications to interface directly with a bridge sensor.

The input multiplexer selects either Channel A or B differential input to the low-noise programmable gain amplifier (PGA). Channel A can be programmed with a gain of 128 or 64, corresponding to a full-scale differential input voltage of ±20mV or ±40mV respectively, when a 5V supply is connected to AVDD analog power supply pin. Channel B has a fixed gain of 32. On-chip power supply regulator eliminates the need for an external supply regulator to provide analog power for the ADC and the sensor. Clock input is flexible. It can be from an external clock source, a crystal, or the on-chip oscillator that does not require any external component. On-chip power-on-reset circuitry simplifies digital interface initialization.

There is no programming needed for the internal registers. All controls to the HX711 are through the pins.

FEATURES

- **Two selectable differential input channels**
- **On-chip active low noise PGA with selectable gain of 32, 64 and 128**
- **On-chip power supply regulator for load-cell and ADC analog power supply**
- **On-chip oscillator requiring no external component with optional external crystal**
- **On-chip power-on-reset**
- **Simple digital control and serial interface: pin-driven controls, no programming needed**
- **Selectable 10SPS or 80SPS output data rate**
- **Simultaneous 50 and 60Hz supply rejection**
- **Current consumption including on-chip analog power supply regulator: normal operation < 1.5mA, power down < 1uA**
- **Operation supply voltage range: 2.6 ~ 5.5V**
- **Operation temperature range: -40 ~ +85℃**
- **16 pin SOP-16 package**

APPLICATIONS

- **Weigh Scales**
- **Industrial Process Control**

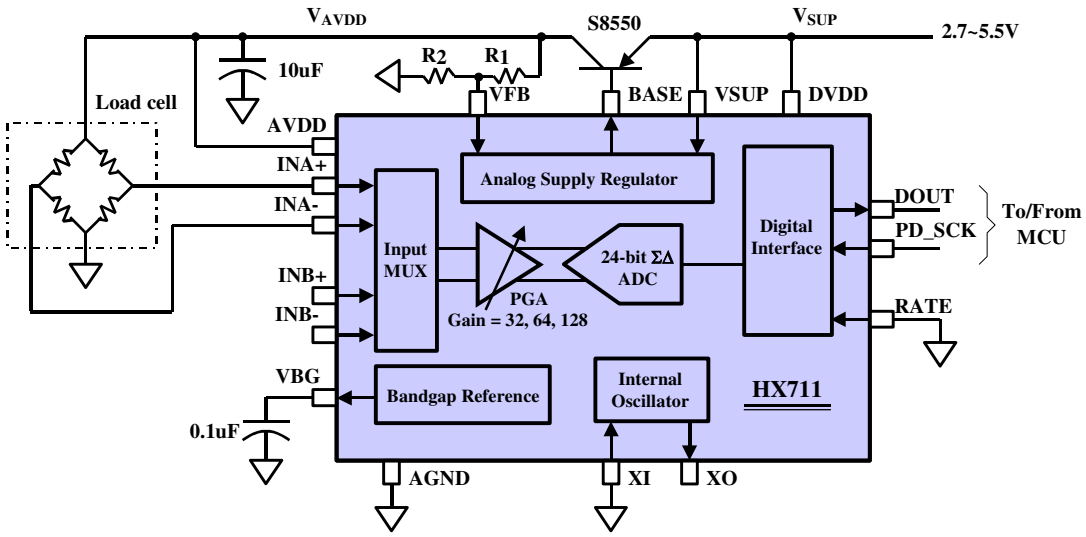
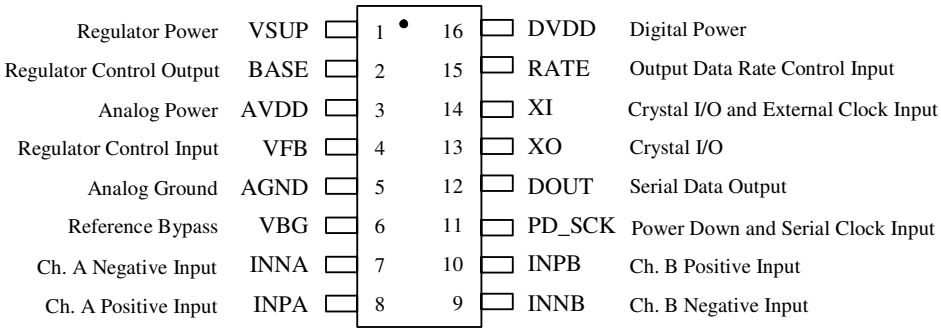


Fig. 1 Typical weigh scale application block diagram

Pin Description



SOP-16L Package

Pin #	Name	Function	Description
1	VSUP	Power	Regulator supply: 2.7 ~ 5.5V
2	BASE	Analog Output	Regulator control output（NC when not used）
3	AVDD	Power	Analog supply: 2.6 ~ 5.5V
4	VFB	Analog Input	Regulator control input（connect to AGND when not used）
5	AGND	Ground	Analog Ground
6	VBG	Analog Output	Reference bypass output
7	INA-	Analog Input	Channel A negative input
8	INA+	Analog Input	Channel A positive input
9	INB-	Analog Input	Channel B negative input
10	INB+	Analog Input	Channel B positive input
11	PD_SCK	Digital Input	Power down control (high active) and serial clock input
12	DOUT	Digital Output	Serial data output
13	XO	Digital I/O	Crystal I/O (NC when not used）
14	XI	Digital Input	Crystal I/O or external clock input, 0: use on-chip oscillator
15	RATE	Digital Input	Output data rate control, 0: 10Hz; 1: 80Hz
16	DVDD	Power	Digital supply: 2.6 ~ 5.5V

Table 1 Pin Description

KEY ELECTRICAL CHARACTERISTICS

Parameter	Notes	MIN	TYP	MAX	UNIT
Full scale differential input range	V(inp)-V(inn)		±0.5(AVDD/GAIN)		V
Common mode input		AGND+1.2		AVDD-1.3	V
Output data rate	Internal Oscillator, RATE = 0		10		Hz
	Internal Oscillator, RATE = DVDD		80		
	Crystal or external clock, RATE = 0		f_{clk}/1,105,920		
	Crystal or external clock, RATE = DVDD		f_{clk}/138,240		
Output data coding	2's complement	800000		7FFFFF	HEX
Output settling time (1)	RATE = 0		400		ms
	RATE = DVDD		50		
Input offset drift	Gain = 128		0.2		mV
	Gain = 64		0.4		
Input noise	Gain = 128， RATE = 0		50		nV(rms)
	Gain = 128， RATE = DVDD		90		
Temperature drift	Input offset（Gain = 128）		±6		nV/℃
	Gain（Gain = 128）		±5		ppm/℃
Input common mode rejection	Gain = 128， RATE = 0		100		dB
Power supply rejection	Gain = 128， RATE = 0		100		dB
Reference bypass （V_{BG}）			1.25		V
Crystal or external clock frequency		1	11.0592	20	MHz
Power supply voltage	DVDD	2.6		5.5	V
	AVDD，VSUP	2.6		5.5	
Analog supply current (including regulator)	Normal		1400		μA
	Power down		0.3		
Digital supply current	Normal		100		μA
	Power down		0.2		

（1）Settling time refers to the time from power up, reset, input channel change and gain change to valid stable output data.

Table 2 Key Electrical Characteristics

Analog Inputs

Channel A differential input is designed to interface directly with a bridge sensor's differential output. It can be programmed with a gain of 128 or 64. The large gains are needed to accommodate the small output signal from the sensor. When 5V supply is used at the AVDD pin, these gains correspond to a full-scale differential input voltage of ±20mV or ±40mV respectively.

Channel B differential input has a fixed gain of 32. The full-scale input voltage range is ±80mV, when 5V supply is used at the AVDD pin.

Power Supply Options

Digital power supply (DVDD) should be the same power supply as the MCU power supply.

When using internal analog supply regulator, the dropout voltage of the regulator depends on the external transistor used. The output voltage is equal to $V_{AVDD}=V_{BG}*(R1+R2)/ R1$ (Fig. 1). This voltage should be designed with a minimum of 100mV below VSUP voltage.

If the on-chip analog supply regulator is not used, the VSUP pin should be connected to either AVDD or DVDD, depending on which voltage is higher. Pin VFB should be connected to Ground and pin BASE becomes NC. The external 0.1uF bypass capacitor shown on Fig. 1 at the VBG output pin is then not needed.

Clock Source Options

By connecting pin XI to Ground, the on-chip oscillator is activated. The nominal output data rate when using the internal oscillator is 10 (RATE=0) or 80SPS (RATE=1).

If accurate output data rate is needed, crystal or external reference clock can be used. A crystal can be directly connected across XI and XO pins. An external clock can be connected to XI pin, through a 20pF ac coupled capacitor. This external clock is not required to be a square wave. It can come directly from the crystal output pin of the MCU chip, with amplitude as low as 150 mV.

When using a crystal or an external clock, the internal oscillator is automatically powered down.

Output Data Rate and Format

When using the on-chip oscillator, output data rate is typically 10 (RATE=0) or 80SPS (RATE=1).

When using external clock or crystal, output data rate is directly proportional to the clock or crystal frequency. Using 11.0592MHz clock or crystal results in an accurate 10 (RTE=0) or 80SPS (RATE=1) output data rate.

The output 24 bits of data is in 2's complement format. When input differential signal goes out of the 24 bit range, the output data will be saturated at 800000h (MIN) or 7FFFFFh (MAX), until the input signal comes back to the input range.

Serial Interface

Pin PD_SCK and DOUT are used for data retrieval, input selection, gain selection and power down controls.

When output data is not ready for retrieval, digital output pin DOUT is high. Serial clock input PD_SCK should be low. When DOUT goes to low, it indicates data is ready for retrieval. By applying 25~27 positive clock pulses at the PD_SCK pin, data is shifted out from the DOUT output pin. Each PD_SCK pulse shifts out one bit, starting with the MSB bit first, until all 24 bits are shifted out. The 25th pulse at PD_SCK input will pull DOUT pin back to high (Fig.2).

Input and gain selection is controlled by the number of the input PD_SCK pulses (Table 3). PD_SCK clock pulses should not be less than 25 or more than 27 within one conversion period, to avoid causing serial communication error.

PD_SCK Pulses	Input channel	Gain
25	A	128
26	B	32
27	A	64

Table 3 Input Channel and Gain Selection

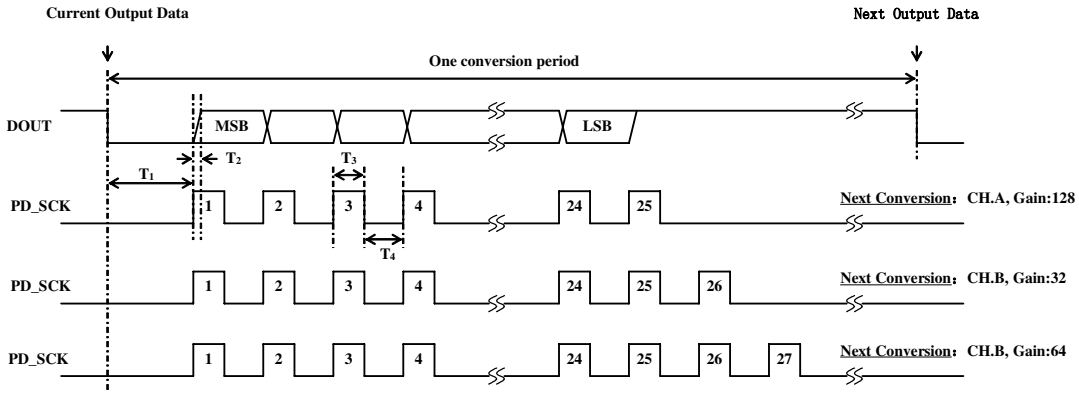


Fig.2 Data output, input and gain selection timing and control

Symbol	Note	MIN	TYP	MAX	Unit
T_1	DOUT falling edge to PD_SCK rising edge	0.1			μs
T_2	PD_SCK rising edge to DOUT data ready			0.1	μs
T_3	PD_SCK high time	0.2	1	50	μs
T_4	PD_SCK low time	0.2	1		μs

Reset and Power-Down

When chip is powered up, on-chip power on rest circuitry will reset the chip.

Pin PD_SCK input is used to power down the HX711. When PD_SCK Input is low, chip is in normal working mode.

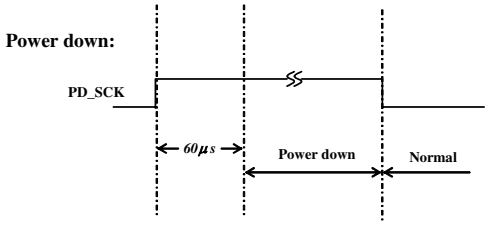


Fig.3 Power down control

When PD_SCK pin changes from low to high and stays at high for longer than 60μs, HX711 enters power down mode (Fig.3). When internal regulator is used for HX711 and the external transducer, both HX711 and the transducer will be powered down. When PD_SCK returns to low, chip will reset and enter normal operation mode.

After a reset or power-down event, input selection is default to Channel A with a gain of 128.

Application Example

Fig.1 is a typical weigh scale application using HX711. It uses on-chip oscillator (XI=0), 10Hz output data rate (RATE=0). A Single power supply （2.7～5.5V） comes directly from MCU power supply. Channel B can be used for battery level detection. The related circuitry is not shown on Fig. 1.

Reference PCB Board (Single Layer)

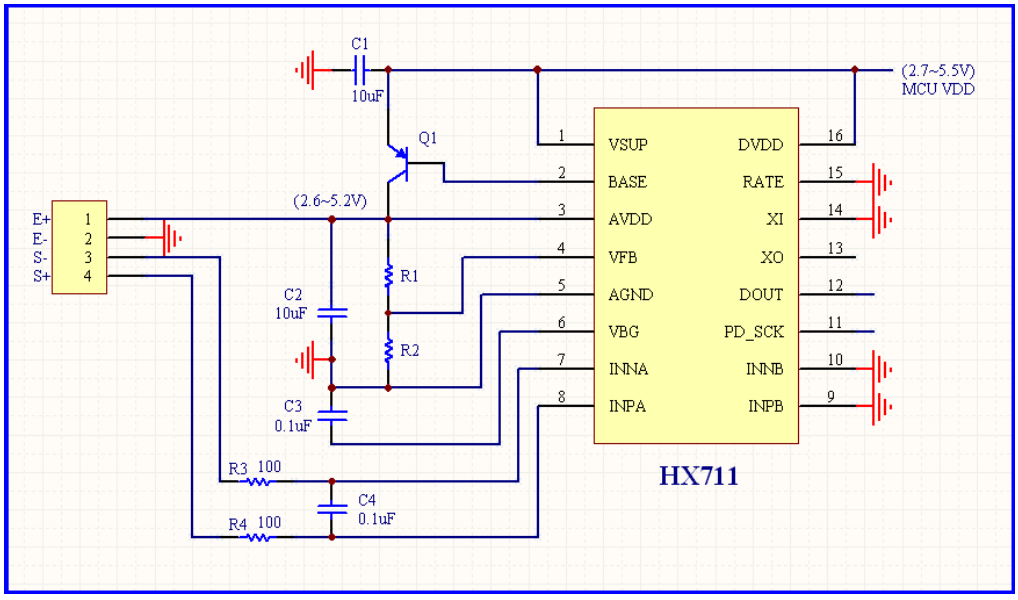


Fig.4 Reference PCB board schematic

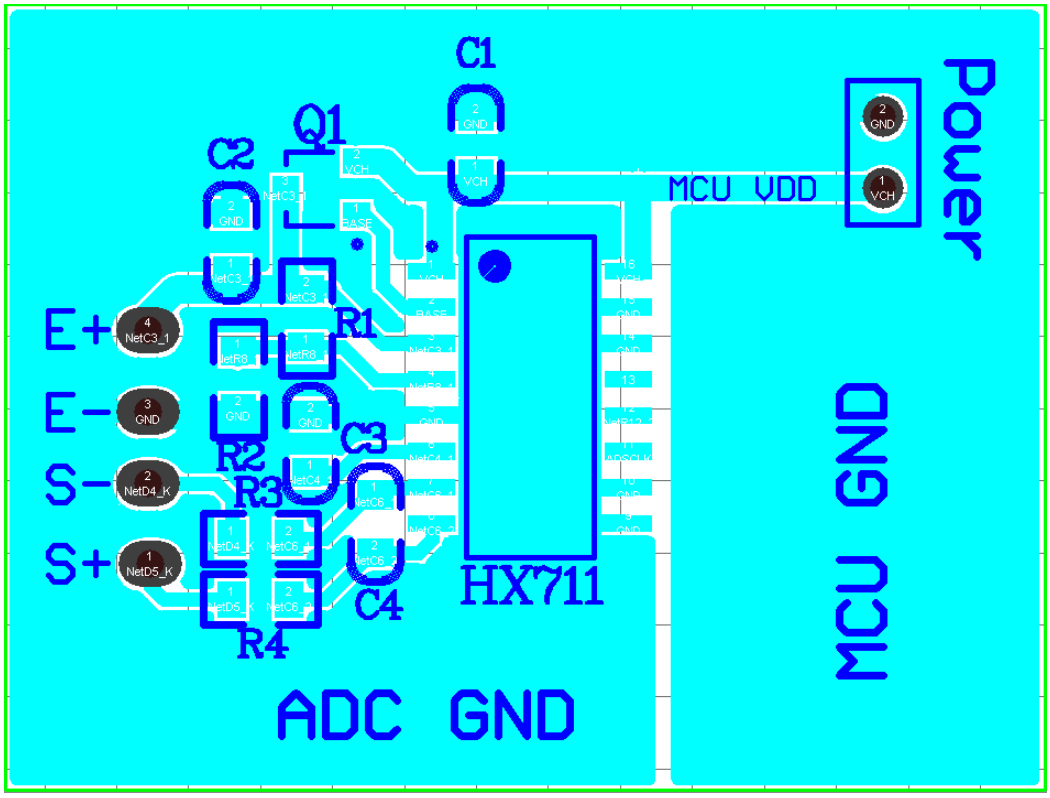


Fig.5 Reference PCB board layout

Reference Driver (Assembly)

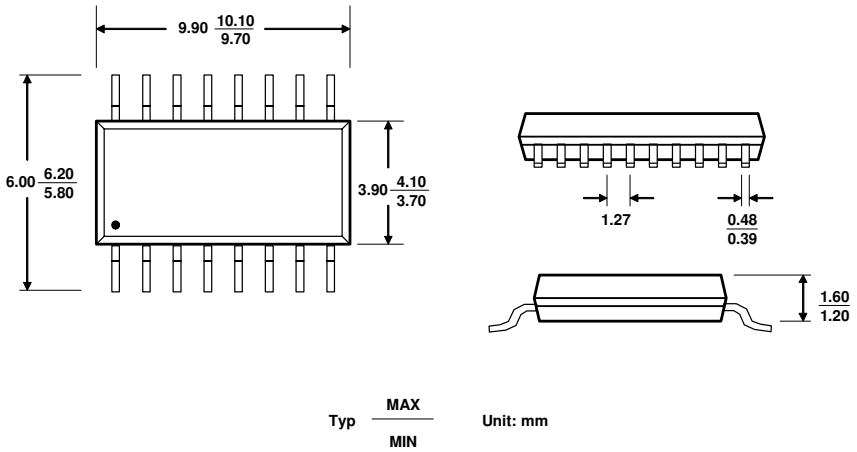
```
/*----------------------------------------------------------------------
Call from ASM:      LCALL   ReaAD
Call from C:    extern unsigned long ReadAD(void);
                    .
                    .
                  unsigned long data;
                  data=ReadAD();
                    .
                    .
----------------------------------------------------------------------*/
PUBLIC        ReadAD
HX711ROM      segment  code
rseg          HX711ROM

sbit           ADDO = P1.5;
sbit           ADSK = P0.0;
/*-------------------------------------------------
 OUT:   R4, R5, R6, R7   R7=>LSB
-------------------------------------------------*/
ReadAD:
    CLR    ADSK              //AD Enable (PD_SCK set low)
    SETB   ADDO              //Enable 51CPU I/O
    JB     ADDO,$            //AD conversion completed?
    MOV    R4,#24
ShiftOut:
    SETB   ADSK              //PD_SCK set high (positive pulse)
    NOP
    CLR    ADSK              //PD_SCK set low
    MOV    C,ADDO            //read on bit
    XCH    A,R7              //move data
    RLC    A
    XCH    A,R7
    XCH    A,R6
    RLC    A
    XCH    A,R6
    XCH    A,R5
    RLC    A
    XCH    A,R5
    DJNZ   R4,ShiftOut        //moved 24BIT?
    SETB   ADSK
    NOP
    CLR    ADSK
    RET
    END
```

Reference Driver（C）

```
//-----------------------------------------------------------------------
sbit   ADDO = P1^5;
sbit   ADSK = P0^0;
unsigned long ReadCount(void){
  unsigned long Count;
  unsigned char i;
  ADDO=1;
  ADSK=0;
  Count=0;
  while(ADDO);
  for (i=0;i<24;i++){
    ADSK=1;
    Count=Count<<1;
    ADSK=0;
    if(ADDO) Count++;
  }
  ADSK=1;
  Count=Count^0x800000;
  ADSK=0;
  return(Count);
}
```

Package Dimensions



SOP-16L Package