Sistemas operativos avanzados

MagicBox

Martes del 1er Cuatrimestre

2019

Docentes

- Lic. Graciela De Lucca
- Ing. Waldo Valiente
- Ing. Sebastián Barillaro
- Ing. Esteban Carnuccio
- Ing. Gerardo García
- Ing. Mariano Volker

Integrantes

Apellido	Nombre	DNI
Flores	Cristian	37787914
Mercado	Maximiliano	37250369
Maciel	Gabriel	37551121
Lorenz	Marcelo	38851212
Lorenz	Lautaro	37661245

Contenido

Presentación

¿Cuántas veces estamos tan ocupados que no podemos administrar el contenido de nuestro refrigerador?
Imagínese la respuesta de esa pregunta llevándola a los ajetreos diarios del ambiente industrial y comercial.

El equipo de *Magicbox* pretende con este proyecto plantear una solución a este tipo de inquietudes mediante un contenedor inteligente, capas de avisarle cuando se están por acabar sus provisiones con la suficiente anticipación para poder actuar de forma inteligente.

Características

El equipo viene preparado para cumplir las siguientes funcionalidades

- Medición de temperatura interna
- Control de temperatura interna, programable
- Medición de peso del producto almacenado
- Medición de volumen del producto almacenado
- Censado de puerta abierta o cerrada
- Aviso de puerta abierta
- Iluminación interna con encendido y apagado automático
- Conexión con aplicación Android
- Búsqueda de proveedores
- Comandos por voz

Puesta en marcha

1. coloque el equipo sobre una superficie plana, con el contenedor vacío y la puerta cerrada
2. conecte el cable de alimentación a la fuente
3. enchufe el cable de alimentación a una conexión eléctrica de 220v de corriente alterna
4. aguarde unos segundos hasta que el equipo termine de inicializarse
5. abra la puerta y coloque un producto dentro
6. inicialice la aplicación Android provista con el equipo, en un celular
7. utilizando la aplicación empareje el celular al bluetooth llamado HC-05 (o MagicBox según la versión que posea)
8. busque en el listado de productos de la aplicación, el producto que colocó en el contenedor de almacenamiento
9. opcional, puede asignar una temperatura de almacenamiento diferente a la que se asigna de forma automática

Prototipo final

Maqueta en tres dimensiones

Ilustración 1 presentación

Especificaciones

- **Dimensiones del recipiente contenedor:** ancho 20 [cm], alto 25 [cm], largo 18 [cm]
- **Peso máximo que puede medir:** 5 [kilogramos]
- **Temperatura mínima que puede mantener:** 17 +/- 2 [grados Celsius]
- **Temperatura máxima que puede mantener:** 24 +/- 1 [grados Celsius]
- **Dimensiones exteriores:** ancho 25 [cm], alto 35 [cm], largo 42.5 [cm]
- **Alimentación:** 220 [volt] en corriente alterna
- **Nombre del bluetooth:** HC-05
- **Clave del bluetooth:** 1234

Vistas de la estructura

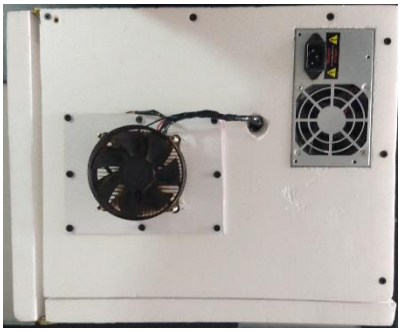


Ilustración 2 lateral izquierdo

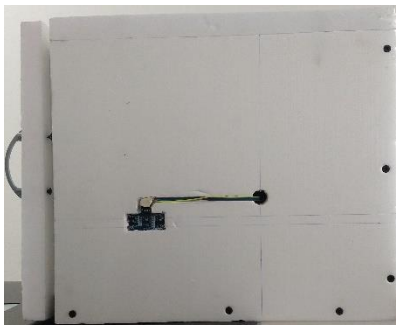


Ilustración 3 lateral derecho

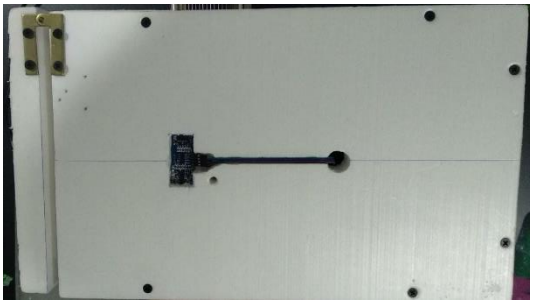


Ilustración 4 superior

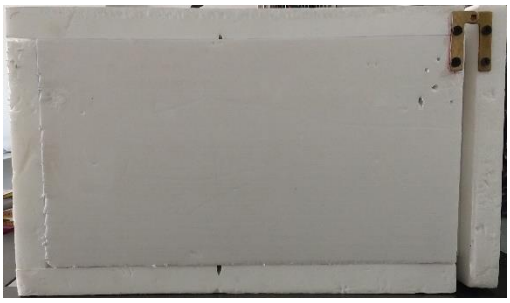


Ilustración 5 inferior

Ilustración 6 frente

Ilustración 7 atrás

Ilustración 8 interior

Sistema embebido

Diagrama de estados

En el diagrama se puede observar el estado del dispositivo en un momento dado, y las causas que pueden provocar los cambios de un estado a otro, según la entrada que reciba.
Por estado se debe entender como las diferentes combinaciones de información que la máquina puede mantener.

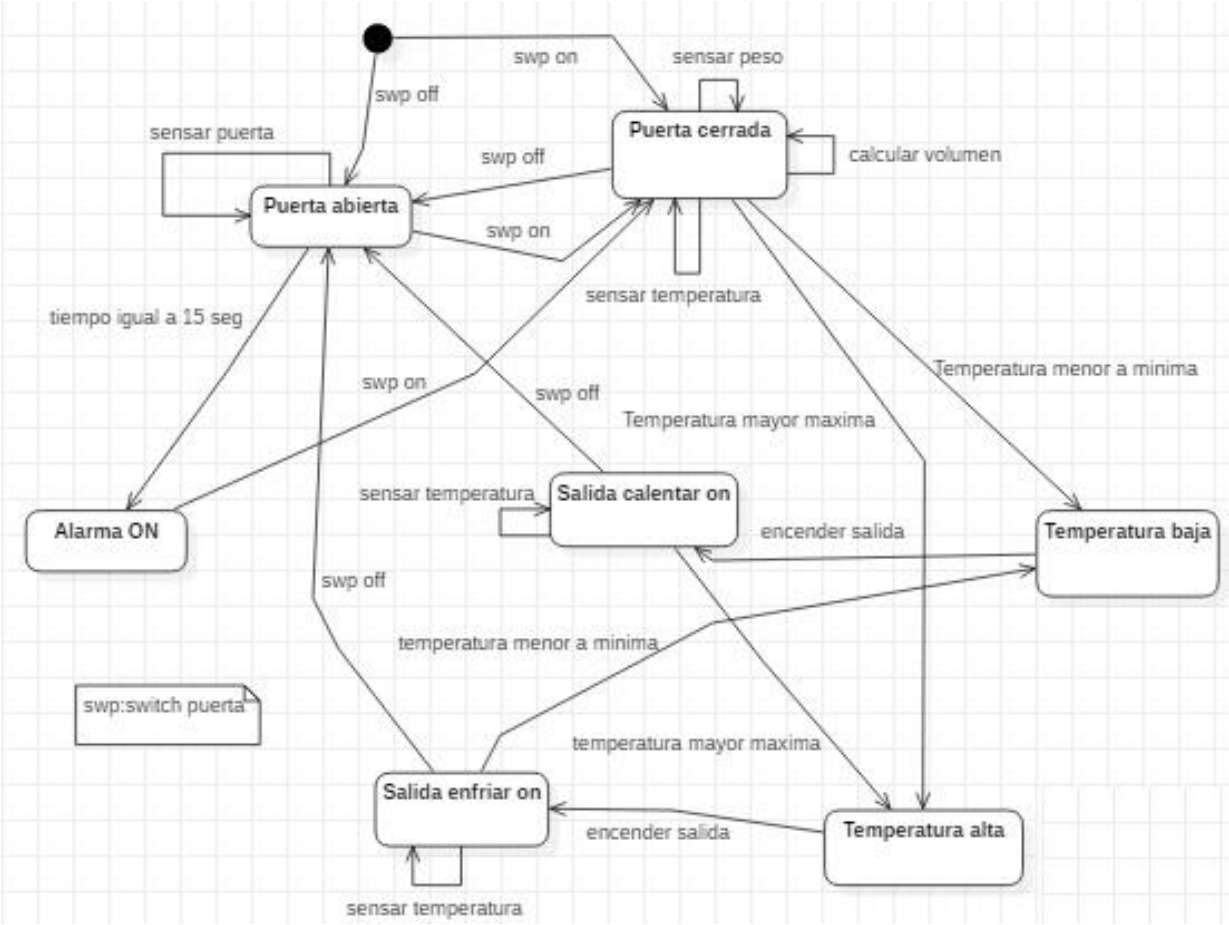


diagrama 1 estados

Diagrama de software

Como una representación visual del flujo de datos, el siguiente diagrama es útil para describir la lógica del programa. Puede ayudar a organizar una perspectiva general.

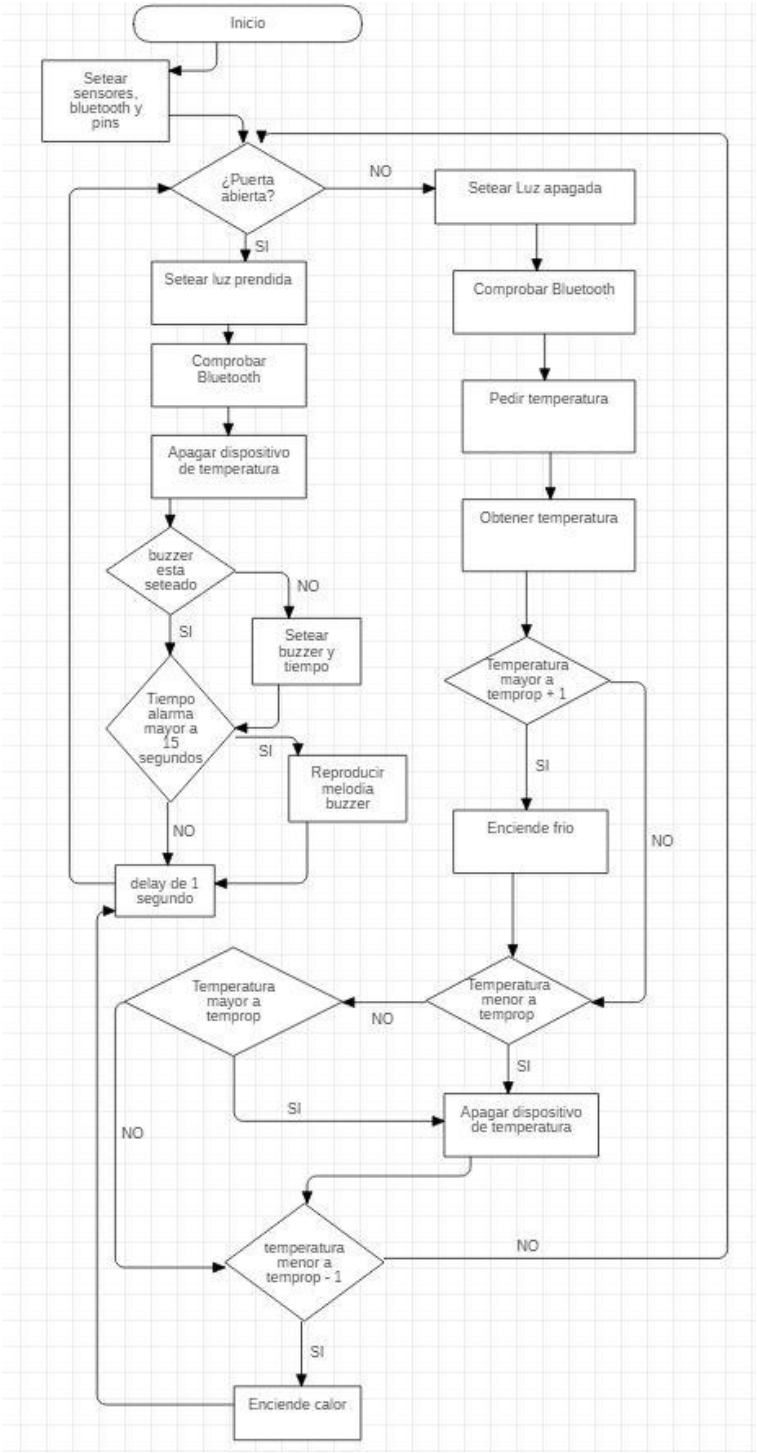


diagrama 2 flujo de software del sistema embebido

Diagrama de conexiones

Este diagrama muestra los diferentes componentes del circuito de manera simple y con pictogramas uniformes de acuerdo a normas, y las conexiones de alimentación y de señal entre los distintos dispositivos. El arreglo de los componentes e interconexiones en el esquema no corresponde a sus ubicaciones físicas en el dispositivo terminado.

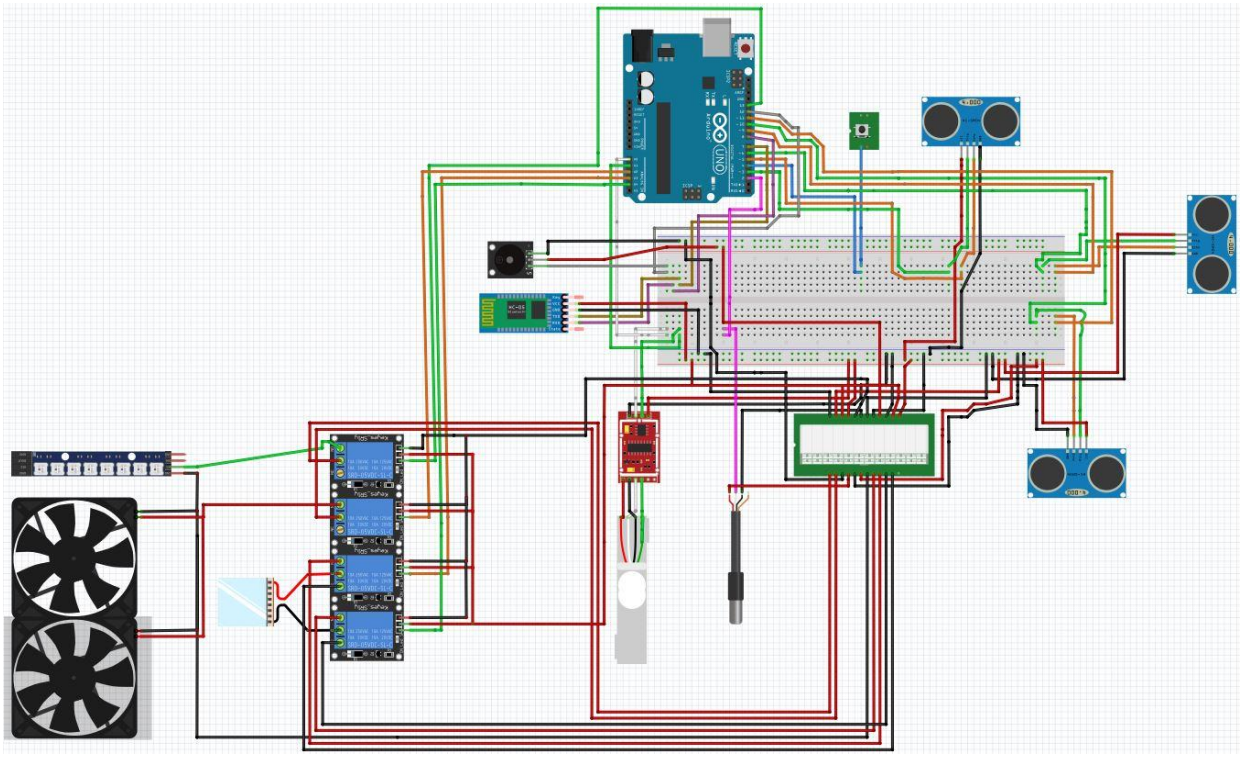


diagrama 3 conexiones

Pines utilizados

En la siguiente tabla se puede observar cuales pines de la placa Arduino fueron utilizados por los diferentes periféricos

PERIFÉRICO	PINES UTILIZADOS		
	DIGITAL	ANALÓGICO	PWM
Sensor temperatura	2		
Sensor peso		A0, A1	
Sensor switch	4		
Sensor ultrasonido X			3, 5
Sensor ultrasonido Y			6, 9
Sensor ultrasonido Z			10, 11
Actuador Buzzer	12		
Actuador temperatura calentar		A3	
Actuador temperatura enfriar		A4	
Actuador luz led interna	13		
Bluetooth	7, 8		
Coolers 1 y 2		A2	

Tabla 1 pines

Detalle de componentes

Arduino R1

El hardware de Arduino consiste en una placa con un microcontrolador generalmente Atmel con puertos de comunicación y puertos de entrada/salida.

Arduino Uno es una placa electrónica basada en el microcontrolador ATmega328. Cuenta con 14 entradas/salidas digitales, de las cuales 6 se pueden utilizar como salidas PWM (Modulación por ancho de pulsos) y otras 6 son entradas analógicas.

Bluetooth HC-05

El módulo de bluetooth HC-05 es un módulo Maestro-Esclavo, quiere decir que además de recibir conexiones desde un celular, PC, tablet, también es capaz de generar conexiones hacia otros dispositivos bluetooth.

Tiene un modo de comandos AT que debe activarse. Una vez que estamos en el modo de comandos AT, podemos configurar el módulo bluetooth y cambiar parámetros como el nombre del dispositivo, contraseña, modo maestro/esclavo, etc.

Las conexiones para realizar con Arduino son bastante sencillas. Solamente requerimos colocar como mínimo la alimentación y conectar los pines de transmisión y recepción serial (TX y RX). Hay que recordar que en este caso los pines se debe conectar cruzados TX Bluetooth -> RX de Arduino y RX Bluetooth -> TX de Arduino.

Distancia (sensor de Ultrasonido)

Los sensores ultrasónicos miden la distancia mediante el uso de ondas ultrasónicas. El cabezal emite una onda ultrasónica y recibe la onda reflejada que retorna desde el objeto. Los sensores ultrasónicos miden la distancia al objeto contando el tiempo entre la emisión y la recepción.

La distancia se puede calcular con la siguiente fórmula:

$$Distancia\ L\ =\ 1/2\ \times\ T\ \times\ C$$

Donde L es la distancia, T es el tiempo entre la emisión y la recepción, y C es la velocidad del sonido. (El valor se multiplica por 1/2 ya que T es el tiempo de recorrido de ida y vuelta).

Temperatura (sensor DS18B20)

El sensor DS18B20 puede medir temperaturas entre -55ºC y 125ºC.

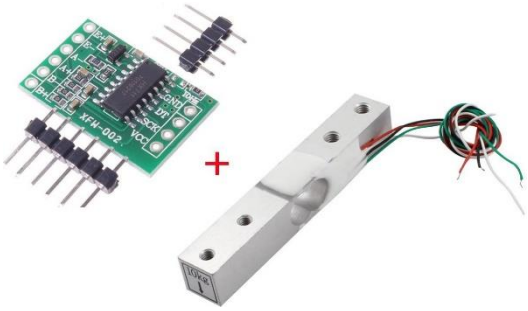
Para temperaturas entre -10ºC y 85ºC podemos tener ±0,5ºC. Para el resto de las temperaturas entre -55ºC y 125ºC el error es de ±2ºC.

Por defecto utiliza la resolución de 12-bit.

Para acceder al valor leído por el sensor debemos utilizar el protocolo 1-wire, es decir, en un único cable se pueden conectar más de uno de estos sensores.

Para utilizarlo con Arduino serán necesarias las librerías OneWire y DallasTemperature.

Celda de carga (sensor de peso) + Convertidor HX711



Transductor: Dispositivo que tiene la misión de recibir energía de una naturaleza eléctrica, mecánica, acústica, etc., y suministrar otra energía de diferente naturaleza, pero de características dependientes de la que recibió.

Galga: Instrumento de precisión para medir ángulos y longitudes muy pequeñas.

Puente Wheatstone: es un circuito eléctrico que se utiliza para medir resistencias desconocidas mediante el equilibrio de los brazos del puente. Estos están constituidos por cuatro resistencias que forman un circuito cerrado, siendo una de ellas la resistencia bajo medida.

Una celda de carga es un *transductor* capaz de convertir una fuerza en una señal eléctrica, esto la hace a través uno o más *galgas* internas que posee, configuradas en un *puente Wheatstone*.

El transmisor Hx711 es una interfaz entre las celdas de carga y el microcontrolador, permitiendo poder leer el peso de manera sencilla. Internamente se encarga de la lectura del puente Wheatstone formado por la celda de carga, convirtiendo la lectura analógica a digital con su conversor Analógico/Digital interno de 24 bits.

Al conectarlo con Arduino lo primero que se debe de hacer es calibrar, que es básicamente hallar el valor de la escala que se usará; es decir hallar el factor de conversión para convertir valor de lectura en un valor con unidades de peso. La escala es diferente para cada celda.

Primero necesitamos conseguir un objeto cuyo peso sea conocido. Se recomienda que el peso conocido sea cercano al valor máximo del rango de trabajo de la celda de carga.

$$Escala = \frac{valor\ de\ lectura}{peso\ real}$$

Buzzer pasivo (actuador de sonido)

Son dispositivos que generan un sonido a una frecuencia dada que puede ser variable, cuando son conectados a tensión.

Los buzzer pasivos necesitan recibir una onda de la frecuencia. Técnicamente tanto buzzer como altavoces son transductores electroacústicos, es decir, dispositivos que convierten señales eléctricas en sonido. La diferencia entre ambos es el fenómeno en el que basan su funcionamiento.

Los buzzer son transductores piezoeléctricos. Los materiales piezoeléctricos tienen la propiedad especial de variar su volumen al ser atravesados por corrientes eléctricas.

Un buzzer aprovecha este fenómeno para hacer vibrar una membrana al atravesar el material piezoeléctrico con una señal eléctrica.

Los buzzer son dispositivos pequeños y compactos, con alta durabilidad, y bajo consumo eléctrico. Por contra, la calidad de sonido es reducida.

Celda Peltier (actuador de temperatura)

El efecto Peltier se produce cuando hacemos pasar una corriente eléctrica continua por un circuito compuesto por dos materiales. Si bien sus uniones están a la misma temperatura, al paso de la corriente se produce el efecto termo eléctrico. Una absorbe calor y la otra lo desprende.

Si invertimos la polaridad de la corriente eléctrica, también se invierte la temperatura de las membranas.

Está compuesta por dos materiales semiconductores, uno es Bismuto tipo N y el otro Teluro tipo P para ser tipo P o N, buenos conductores de electricidad y malos conductores del calor unidos entre sí por una lámina de cobre.

Si en el lado del material N se aplica el polo positivo de una fuente de alimentación de corriente continua y en el lado del material P el polo negativo, la placa de cobre de la parte superior se enfría, mientras que la inferior se calienta. Si en esta misma celda, se invierte la polaridad de alimentación, se invierte el efecto de la temperatura. Los elementos de un módulo Peltier son bloques de 1 mm cúbico

Relé (actuador acoplador electromecánico)

Un relé es un dispositivo electromecánico que permite a un procesador como Arduino controlar cargas a un nivel tensión o intensidad muy superior a las que su electrónica puede soportar.

Por ejemplo, con una salida por relé podemos encender o apagar cargas de corriente alterna a 220V e intensidades de 10A, lo cual cubre la mayoría de los dispositivos domésticos que conectamos en casa a la red eléctrica.

Las salidas por relé son muy frecuentes en el campo de la automatización de procesos, y casi todos los autómatas incluyen salidas por relé para accionar cargas como motores, bombas, climatizadores, iluminación, o cualquier otro tipo de instalación o maquinaria.

Físicamente un relé se comporta como un interruptor “convencional” pero que, en lugar de accionarse manualmente, es activado de forma electrónica. Los relés son aptos para accionar cargas tanto de corriente alterna como continua.

El circuito primario se conecta con la electrónica de baja tensión, en nuestro caso Arduino, y recibe la señal de encendido y apagado.

El circuito secundario es el interruptor encargado de encender o apagar la carga.

Al ser dispositivos electromecánicos que requieren el movimiento de componentes interno para su funcionamiento el tiempo de conmutación de un relé es elevado, del orden de 10ms.

Como consecuencia los relés no pueden usarse con una señal PWM, ni otro tipo de señales de frecuencia media-alta.

Disipador de calor (actuador ventilación)

Un disipador es un instrumento que se utiliza para bajar la temperatura de algunos componentes electrónicos.

Su funcionamiento se basa en la ley cero de la termodinámica, transfiriendo el calor de la parte caliente que se desea disipar al aire. Este proceso se propicia aumentando la superficie de contacto con el aire, permitiendo una eliminación más rápida del calor excedente.

Fuente de tensión ATX

La fuente ATX es un dispositivo que se acopla internamente en el gabinete, la cual se encarga de transformar la corriente alterna de la línea eléctrica en corriente directa; así como reducir su voltaje. Esta corriente es utilizada por los elementos electrónicos y eléctricos. Otras funciones son las de suministrar la cantidad de corriente y voltaje que los dispositivos requieren, así como protegerlos de problemas en el suministro eléctrico como subidas de voltaje.

Repasando algunos términos de electricidad, recordemos que la electricidad no es otra cosa más que electrones circulando a través de un medio conductor. La potencia eléctrica de una fuente ATX se mide en Watts (W) y esta variable está en función de otros dos factores:

- El Voltaje: es la fuerza con la que son impulsados los electrones a través del circuito. Se mide en Volts (V) y varía acorde a la región.
- La Corriente: es la cantidad de electrones que circulan por un punto en específico del circuito cada segundo. Su unidad de medida es el Ampere (A).

Luz led 12v (actuador iluminación)

Un diodo LED se trata de componentes electrónicos que permiten el paso de la corriente en un solo sentido. La palabra viene del inglés Light Emitting Diode, que traducido al español es Diodo Emisor de Luz.

Cuando la electricidad pasa a través de un diodo, los átomos de uno de los materiales (contenido en un chip-reflector) son excitados a un mayor nivel. Los átomos en el primer material retienen mucha energía y requieren liberarla. Esta energía libera electrones al segundo material dentro del chip-reflector y, durante esta liberación, se produce la luz.

En otras palabras, la electroluminiscencia se da cuando, estimulados por un diferencial de voltaje, las cargas eléctricas negativas (electrones) y las cargas eléctricas positivas, al combinarse entre sí, dan como resultado la liberación de energía en forma de fotones.

Librerías

SoftwareSerial

Arduino viene construido para soportar comunicación serial en los pines 0 y 1 de manera nativa, esto lo hace mediante una pieza de hardware. Este hardware ayuda a que el procesador atmega pueda comunicarse vía serie incluso aunque este trabajando en otras tareas, para esto implementa un buffer de 64byte.

Para poder implementar comunicación serie en cualquier otro pin del Arduino se puede utilizar esta librería, cuya funcionalidad es simular una comunicación serie utilizando pines digitales.

El protocolo serie consiste en enviar señales de nivel alto y bajo (ceros y unos) a través de un pin emisor comúnmente conocido como TX y son recibidas por el pin receptor RX. Esta librería manipula los estados de los pines digitales que hayan sido configurados por el desarrollador del software para poder usarlos como emisor y receptor de los pines necesarios para implementar el protocolo serie.

Funciones utilizadas:

- SoftwareSerial BT(7,8) define una estructura de tipo SoftwareSerial y asigna los valores de los pines emisor y receptor.
- BT.begin(9600) define la velocidad de comunicación en baudios, del protocolo serial que será implementado, ambas partes deben manejar la misma velocidad.
- BT.available() verifica si los pines están disponibles para ser usados.
- BT.read() retorna los bytes del buffer serie que fueron recibidos.
- BT.write(mensaje) envía un mensaje a través del pin emisor.

OneWire

Con este sistema de conexionado logramos enviar y recibir datos por un único cable. Para esto se implementa el protocolo del mismo nombre de la librería “OneWire”. Como principal ventaja de este protocolo podemos mencionar que se puede conectar más de un dispositivo del mismo tipo utilizando un único cable de datos.

Funciones utilizadas:

- OneWire ourWire(2) establece el pin 2, como bus a través del cual se implementará el protocolo de comunicación OneWire.

DallasTemperature

Esta librería está pensada para trabajar en conjunto con la librería OneWire, y fue ideada para interpretar de forma sencilla los datos que se reciben por el bus de datos cuando el sensor que los envía es un sensor de temperatura DS18B20 (entre otros). Tiene como principales ventajas que no presenta limitaciones en cuanto a la cantidad de sensores que podemos conectar y además es simple de utilizar, ya que esta especialmente pensada para estos sensores.

Funciones utilizadas:

- DallasTemperature sensorTemp(&ourWire) se declara una estructura de tipo DallasTemperature y se asigna el bus de datos OneWire en cual están previamente conectados los sensores de temperatura.
- sensorTemp.begin() inicializa todos los sensores de temperatura que estén conectados al bus.
- sensorTemp.requestTemperatures() se transmite a través del bus el comando para que los sensores de temperatura envíen la temperatura medida a través del bus.
- sensorTemp.getTempCByIndex(0) recibe la medida de temperatura que fue transmitida por el sensor que está en la posición cero.

NewPing

Nos ayuda a medir la distancia utilizando sensores de ultrasonido de la serie HC-SR0X, en nuestro caso HC-SR04. Para esto se encarga de administrar los pines de echo y trigger que incorpora el sensor y devolvernos los resultados procesados de diferentes maneras.

Funciones utilizadas:

- NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE) define una estructura de tipo NewPing e inicializa los pines de trigger, echo y la máxima distancia que se puede medir.
- sonar.ping_median() realiza cinco mediciones de ping, es decir poner en alto el trigger y escuchar el echo, y luego retorna la media de tiempo en microsegundos que tardó el tren de pulsos de ultrasonido en salir y volver del sensor, en base a esta medida podemos calcular la distancia entre el sensor y el objeto.

hx711

Con esta librería podemos convertir las mediciones analógicas que son emitidas por la interfaz Hx711 a digitales (ADC) necesario para leer las mediciones de la celda de carga (sensor de peso) y aplicarles la escala de medida justa para interpretar la medida en una unidad de peso que sea conocida, en nuestro caso el kilogramo.

Funciones utilizadas:

- Hx711 scale(A1, A0) define una estructura de tipo Hx711 y asigna los valores de los pines DOUT y SCK, donde el pin DOUT es el pin de datos y SCK es el pin de clock.
- scale.getGram() retorna los gramos medidos por la celda de carga. Esta medida debe ser manipulada matemáticamente para ajustarse a la escala real. Para esto la balanza debe ser calibrada mediante la utilización de un peso que sea conocido.

String

Permite crear objetos de la clase String, útiles para simplificar el manejo de cadenas de texto.

Funciones utilizadas:

- definir objetos de la clase string
- comprar valores string utilizando los operadores “=”, “==”, “!=”, “>=” y “<=”
- charAt(pos) retorna el carácter que se encuentra en la posición “pos”

Android

Capturas de pantalla

Ilustración 9 inicio con bluetooth apagado

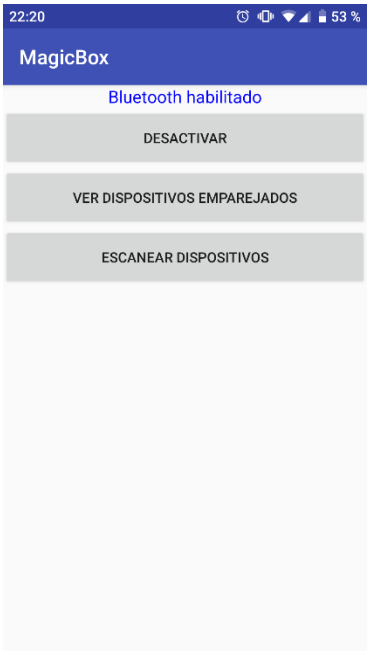


Ilustración 10 inicio con bluetooth encendido

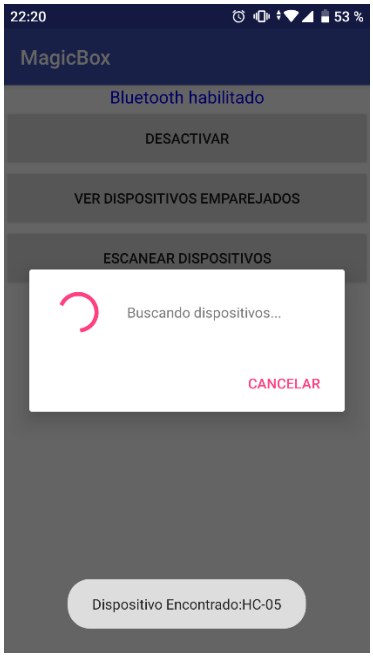


Ilustración 11 buscando dispositivos

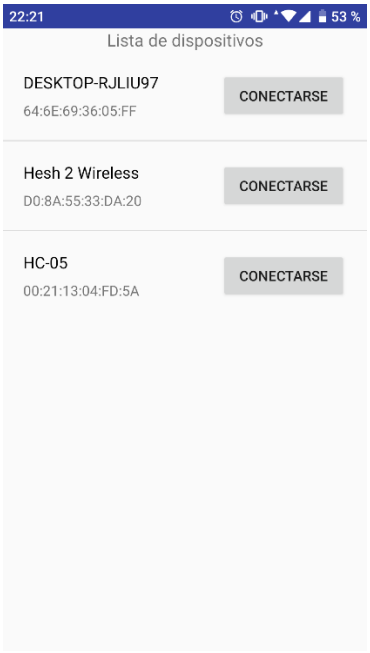


Ilustración 12 dispositivos encontrados

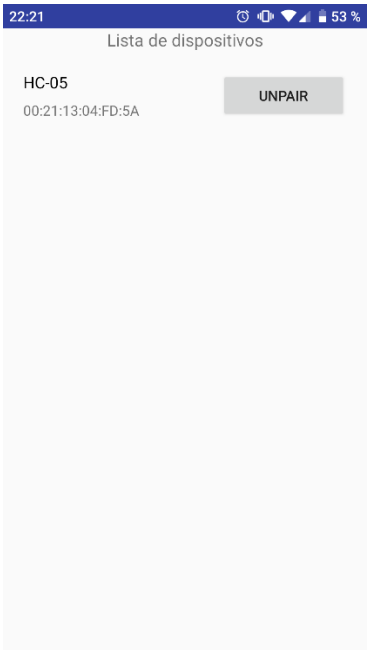


Ilustración 13 dispositivos emparejados

Ilustración 14 pantalla principal

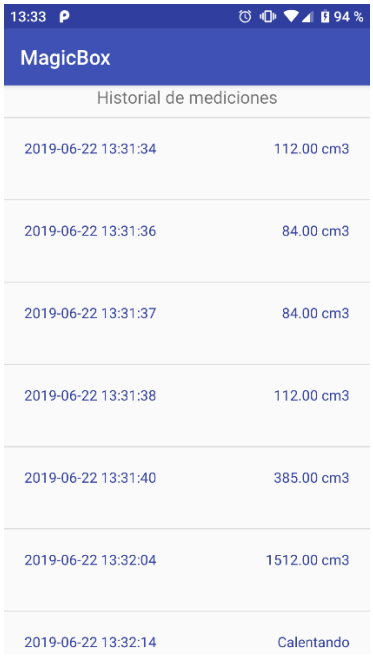


Ilustración 15 historial

Ilustración 16 proveedores

Ilustración 17 productos

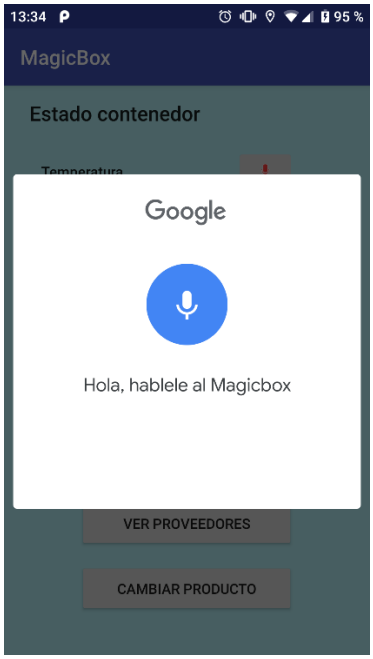


Ilustración 18 Comandos por voz

Diagrama de software

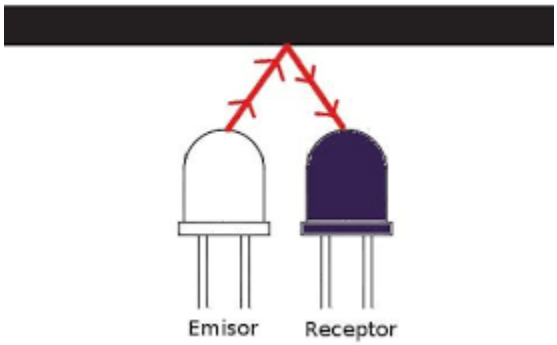
[diagrama de software lo hace marce]

Sensores utilizados

A continuación, se detalla cada uno de los sensores, del dispositivo con sistema operativo Android en el cual corre la aplicación MagicBox, que fueron utilizados para mantener actualizada la información acerca del estado del sistema embebido y cuya finalidad es mantener informado al usuario de la aplicación.

Proximidad

Este consta de dos elementos, un emisor de infrarrojos y el propio sensor que recibe este espectro no visible de la luz. Emisor y receptor detectan los objetos que hay próximos al actuar dichos objetos como un espejo. El emisor de infrarrojos emite luz dentro de ese espectro no visible, igual que hace el mando a distancia de una televisión; y el receptor capta la señal emitida, como haría la tele. Cuando la luz infrarroja rebota en un objeto la capta el emisor.

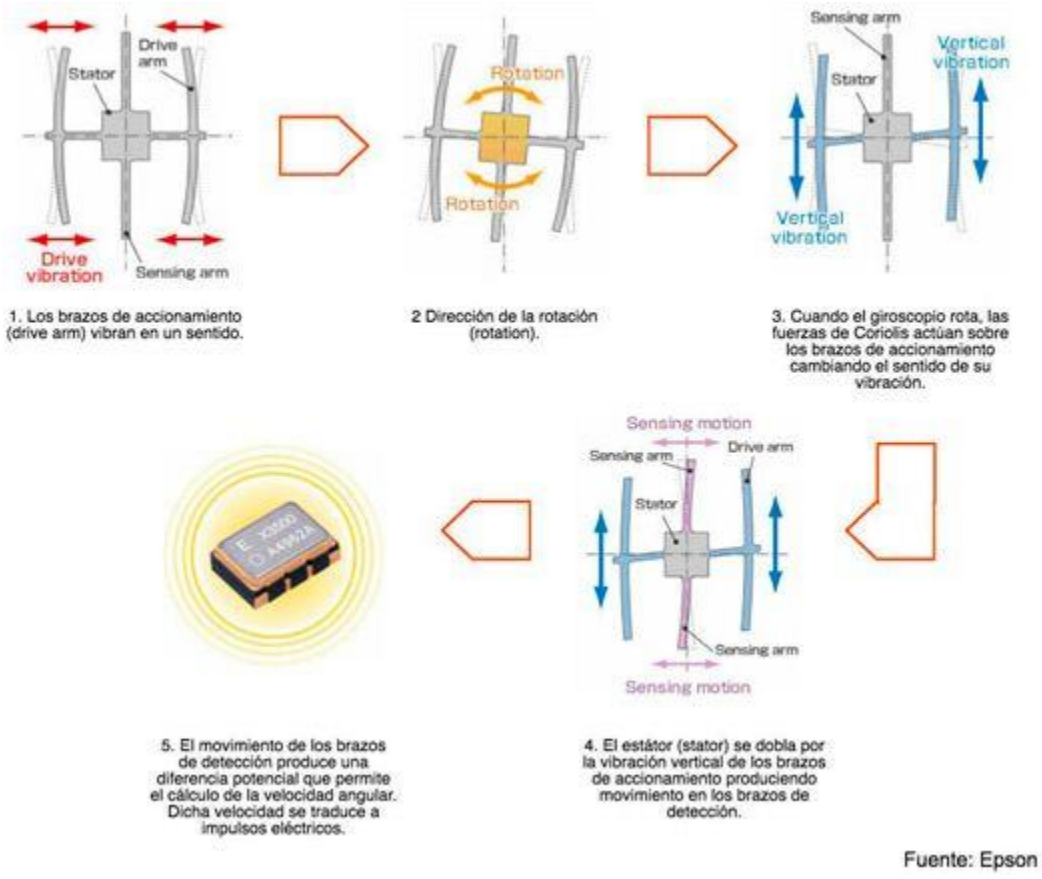


El mismo fue utilizado para enviar el comando “T” al sistema embebido, el cual responderá la temperatura actual en el recinto, dicha temperatura obtenida, es mostrada en la pantalla principal *(ilustración 14)*.

Giroscopio

Con el giroscopio el móvil sabe si está totalmente horizontal (se calibra de serie para que ésta sea la posición base), si lo movemos a la izquierda, a la derecha o si, por el contrario, le damos la vuelta. También es capaz de registrar la aceleración del movimiento.

Es el encargado de medir nuestra posición en el espacio, tarea que realiza mediante el movimiento de un brazo de accionamiento, que rota sobre un componente fijo llamado estátor, el cual a través de los principios del llamado "Efecto Coriolis" se dobla, lo que se traduce en movimiento en el brazo de detección.



Fuente https://tecnologia-facil.com/que-es/el-giroscopio-en-el-celular/

Este sensor fue utilizado para detectar el sentido de giro sobre el eje Z:

- Si gira en sentido horario, se envía al sistema embebido una "P", el sistema embebido nos responderá entonces el peso del contenedor, cuyo dato es utilizado para actualizar la información en la pantalla principal *(ilustración 14)*.
- Si gira en sentido antihorario, se envía al sistema embebido una "V", el sistema embebido nos responderá entonces el volumen ocupado en el contenedor, cuyo dato es utilizado para actualizar la información en la pantalla principal *(ilustración 14)*.

Micrófono

El dictado por voz de Google nos permite escribir con nuestra voz para que no tengamos que teclear nada. Solo tenemos que hablar para que nuestra voz se convierta en texto *(ilustración 18)*.

Dicho texto es interpretado por la aplicación buscando palabras clave las cuales ejecutarán comandos para ser enviados al sistema embebido:

- Al detectar la palabra "alarma", envía una "B", dicho comando apaga el aviso sonoro que es emitido por el Buzzer cuando la puerta está abierta.
- Al detectar la palabra "estado", envía una "E", dicho comando devuelve el estado del actuador de temperatura, el cual puede estar “Apagado” puede estar en “Calor” es decir aumentando la temperatura del contenedor o “Frio” es decir, está extrayendo el calor del contenedor para reducir la temperatura.
- Al detectar la palabra "puerta", envía una "Z", dicho comando devuelve el estado de la puerta, la cual puede estar “Abierta” o “Cerrada”.

La información obtenida por los comandos es utilizada para actualizar la información de la pantalla principal *(ilustración 14)*.

Protocolo de comunicación

La aplicación Android le envía al sistema embebido los siguientes comandos como caracteres, los cuales son interpretados para ejecutar una acción en el sistema:

si recibe	acción
E	devuelve estado del actuador de temperatura (Apagado, Frio o Calor)
B	apagar Buzzer si está sonando
d - } (caracteres ASCII 100 - 125)	configura la temperatura entre 0 y 25 grados Celsius según el carácter
V	devuelve el volumen ocupado en el recinto, en cm^3
P	devuelve el peso del objeto en el recinto, en kilogramos
T	devuelve la temperatura actual en el recinto
S	devuelve peso, volumen y temperatura
Z	devuelve el estado de la puerta "cerrada" o "abierta"

Tabla 2 protocolo

El sistema embebido recibe los caracteres desde la aplicación y los compara con los valores enteros que están asociados a cada acción en el software. Dichos valores enteros asociados a cada acción no están escritos explícitamente en el código, sino que están representados como datos de tipo string.

Para ejemplificar esto supongamos que la aplicación envía el comando “T”, entonces el sistema embebido hará la siguiente comparación (en pseudo código):

```
Si (comando_recibido == “84”)
    responder(temperatura_actual);
```

Obsérvese que el “84” es el valor del comando “T” que es capaz de entender el software, y “responder(temperatura_actual)” es la acción asociada a dicho comando.

Inconvenientes durante la etapa de desarrollo

Calibración de sensores

Un problema asociado con distintos sensores es que cada uno tiene una sensibilidad, un modo de medición, retrasos. Y de forma complementaria Arduino es una placa limitada que no admite concurrencia; es decir; no se pueden medir dos sensores y ejecutar otras acciones simultáneamente. Esta razón entre otras provoca que las lecturas de un sensor o la manipulación de los datos obtenidos de ellos desemboquen en errores, a veces muy difíciles de detectar.

Ruido

Una forma de arreglar errores en las mediciones a causa del ruido es la implementación de correcciones por software, por ejemplo, realizar las mediciones en bucle de manera tal que si la medición tomada es igual una cantidad prudente de veces seguidas, se asume la que medida es correcta, mientras que si la medida arroja valores diferentes en cada vez que es tomada, se asume que se está midiendo ruido, para que este tipo de código sea válido, las mediciones se deben tomar más rápido de lo que la medida realmente puede cambiar. Esto nos da pie a la siguiente forma de error.

Error de velocidad

Este tipo de error se da cuando la velocidad de respuesta del sistema es demasiado lenta respecto de la velocidad de cambio de la magnitud que se intenta medir, para cuando las mediciones son tomadas, estás ya deben ser descartadas porque están desactualizadas respecto de la medida real.

También puede darse si el sistema toma las diferentes mediciones con una velocidad mayor a la que el ruido realmente puede cambiar, en este caso, por más que se hagan mediciones en bucle, todas arrojaran el mismo valor de ruido, ya que el mismo recién desaparece luego de que el sistema haya tomado la medición como correcta.

Programación de alarma

Para poder reproducir la marcha imperial, cuando la puerta de contenedor es olvidada abierta, se tuvo que programar las diecinueve frecuencias de cada tono de forma manual, además de esto, mientras la canción se reproduce se puede recibir un comando por bluetooth en cualquier momento, por lo que tuvo que agregarse código para controlar si se recibió un comando, manteniendo así la respuesta en tiempo real aun cuando la marcha está sonando.

Nueva funcionalidad “control de temperatura” agregada de forma tardía

Inicialmente se ideo que la funcionalidad de control de temperatura sería teórica, la misma estaba pensada por tres estados (Enfriando, Calentando o Apagado) con dos diodos led uno para el estado Enfriando, uno para el estado Calentando, o ambos leds apagados para el estado Apagado.

En plena etapa de construcción se decidió que el control de temperatura sería implementado de forma real mediante un dispositivo como se muestra a continuación

El desafío que esto nos planteo fue resolver los nuevos tiempos que representaban la adquisición de materiales, la programación que hubo que agregar al software ya desarrollado y los cambios estructurales que se realizaron sobre el prototipo construido.

Conexión bluetooth con dispositivo de bajo poder de procesamiento

Aunque la conexión entre el sistema embebido Arduino y la aplicación que corre en Android pudo ser establecida con éxito, esto no fue todo, la comunicación es lenta en gran medida debida al bajo poder de procesamiento del procesador atmega328p con el que contamos.

Para resolver este inconveniente se tuvo que desarrollar un protocolo en donde la aplicación le envía al sistema embebido un único carácter y el mismo es interpretado por el software del sistema embebido para saber qué acción debe tomar *(ver Protocolo de comunicación)*.

Medición del volumen con los sensores disponibles

Actualmente el dispositivo utiliza un sensor de ultrasonido por eje X, Y, Z es decir tres sensores en total, para calcular la distancia que hay hasta el objeto del cual se debe tomar las dimensiones y en base a ello calcular el espacio que ocupa.

Esto da como resultado un error en la medición debido a que este tipo de sensores no es capas de detectar la forma real del objeto. Se puede mejorar la medición agregando más sensores de ultrasonido, por ejemplo, un total de seis sensores, ubicando dos por eje en forma contrapuesta, pero en este caso, no sería suficiente la cantidad de pines con las que se cuenta utilizando un Arduino R1, lo que provocaría un cambio de procesador, o el agregado de hardware adicional para implementar un nuevo circuito lógico de control de sensores, para poder medir de a uno por vez.

Cambiar producto en la aplicación

Al crear el activity de la pantalla principal por primera vez, se inicia un hilo cuya tarea es escuchar los mensajes que llegan a través del buffer del bluetooth, para con dicha información mantener los datos que se observan en la vista, actualizados.

Para que la información recibida se vea reflejada en la vista se utilizó un manejador, el cual apunta a cada elemento de la interfaz visual y permite cambiar el valor que está mostrando.

Cuando se cambia el producto, el manejador pierde la referencia que tiene de cada elemento de la interfaz.

La solución para este inconveniente fue volver a crear el manejador de elementos de la interfaz cuando se crea el activity de la pantalla principal y también actualizar el hilo para informarle cual es el nuevo manejador al que debe enviarle la información recibida.

Mejoras aplicables y nuevas funcionalidades

Medición del volumen

Agregar sensores que sean capaces de detectar la forma real del objeto para poder dar una medida más exacta del volumen ocupado en el contenedor.

Medición de temperatura

Agregar una matriz de sensores infrarrojos, capas de medir la temperatura del objeto en la parte exterior.

Control de temperatura

Modificar el módulo de control de temperatura aumentando la potencia calórica con más celdas Peltier, aprovechando la potencia que puede entregar la fuente de energía utilizada.

Cableado y circuitería

Se puede reducir la cantidad de cableado y con ellos optimizar el espacio ocupado por los circuitos desarrollando una plaqueta con pistas en donde los diferentes módulos puedan ser conectados.

Canción seleccionable

Consiste en agregar a la aplicación y al sistema embebido la lógica necesaria para poder seleccionar la canción que se reproduce cuando la puerta quedó abierta.

A demás se podría implementar una funcionalidad para controlar el tiempo con el cual se considera que la puerta está abierta (actualmente fijado en 15 segundos) o bien apagar está funcionalidad si no queremos que se nos avise cuando dejamos la puerta abierta.

Medición de humedad

A grandes rasgos este punto consiste en agregar un sensor de humedad al contenedor para informar al usuario la misma, mediante la interfaz de la pantalla principal de la aplicación.

Intensidad de la luz

Esta funcionalidad consiste en darle al usuario la posibilidad de controlar la intensidad lumínica cuando la puerta está abierta, además se podría mantener las luces siempre encendidas más allá del estado de la puerta, útil por ejemplo en el caso de los comercios que desean mantenerlas activadas durante los horarios de venta al público.

Un efecto secundario esta característica es el ahorro en el consumo que se puede conseguir si no queremos utilizar siempre las luces a la potencia máxima.

Fuentes consultadas

http://www.iescamp.es/miarduino/2016/01/21/placa-arduino-uno/

https://aprendiendoarduino.wordpress.com/2016/12/11/que-es-arduino-2/

https://www.geekfactory.mx/tutoriales/bluetooth-hc-05-y-hc-06-tutorial-de-configuracion/

https://www.keyence.com.mx/ss/products/sensor/sensorbasics/ultrasonic/info/index.jsp

https://programarfacil.com/blog/arduino-blog/ds18b20-sensor-temperatura-arduino/

https://naylampmechatronics.com/blog/25_tutorial-trasmisor-de-celda-de-carga-hx711-ba.html

https://www.luisllamas.es/reproducir-sonidos-arduino-buzzer-pasivo-altavoz/

https://javierona.net/ingenieria/peltier/

https://www.luisllamas.es/arduino-salida-rele/

https://es.wikipedia.org/wiki/Disipador

http://www.informaticamoderna.com/Fuente_ATX.htm

https://okdiario.com/curiosidades/como-funciona-led-450024

https://www.lucidchart.com/pages/es/que-es-un-diagrama-de-flujo

https://elandroidelibre.elespanol.com/2016/07/giroscopio-movil-android.html

http://blascarr.com/medir-errores-con-arduino/

https://www.milesburton.com/Dallas_Temperature_Control_Library#Introduction

https://playground.arduino.cc/Code/NewPing/