Curso de programación de placas robóticas

Centro de formación: CIFO La Violeta

Formador: Joan Masdemont i Fontás

Proyecto de final de curso: Construcción de un dispensador desinfectante de manos

Integrantes del equipo: Mikel Gurruchaga y Martín Marietta

Noviembre del 2020

Indice

Introducción

La actual situación planteada por la COVID-19 nos lleva a replantearnos ciertos hábitos que conciernen al distanciamiento social y a la higiene personal entre otros aspectos. La higiene y limpieza de manos se ha convertido en un hábito clave donde desde diferentes organizaciones privadas y públicas se incide sobre la población para lograr disminuir la propagación del virus SARS-COV2. Desde esta perspectiva los dispositivos accionados sin contacto directo resultan muy útiles para cumplir con las premisas planteadas desde estas organizaciones.

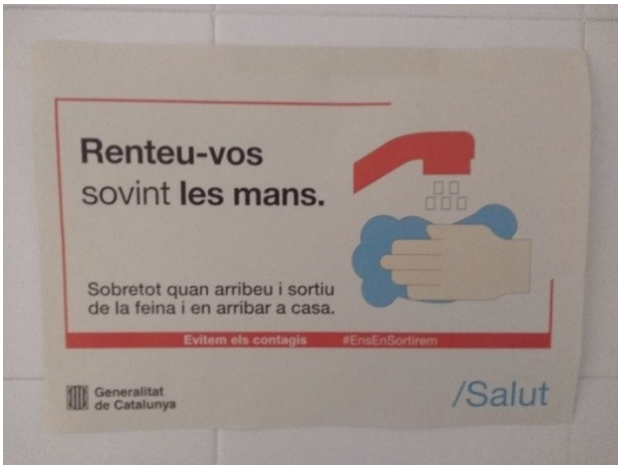

Fig. 1: Cartel de recomendación de hábitos de higiene de manos presente en un lavabo de CIFO La Violeta.

Motivación

Utilizar los conocimientos adquiridos en el curso para diseñar y construir de un dispensador de solución hidro-alcohólica accionado sin contacto directo por el usuario. De esta forma contribuimos al propósito de fomentar el uso de la solución desinfectante evitando el contacto para su dosificación. La intención es que el dispositivo pueda ser utilizado por el centro educativo para implementar las medidas higiénicas obligatorias para alumnado y el personal docente.

Alcance

El proyecto se basa en la construcción de un dispositivo electromecánico de pequeñas dimensiones controlado por una placa Arduino© o similar.

Descripción y características

Se plantea que el diseño del dispositivo incluya las siguientes funcionalidades y/o características particularmente distintivas respecto a otros dispositivos similares existentes en el mercado. Estas características se enumeran a continuación:

. Sistema recargable de solución hidro-alcohólica.

. Control visual lumínico del nivel de líquido para la oportuna reposición.

. Control de la cantidad de desinfectante a dosificar.

. Alimentación dual: mediante cargador enchufado a la red o a través de baterías recargables.

. La construcción del prototipo involucrará la utilización de componentes de bajo coste y de materiales reutilizados con el objeto de disminuir el impacto en el medio ambiente.

Diseño

Teniendo en cuentas estas funcionalidades el dispensador deberá poseer un deposito donde colocar la solución desinfectante en cuyo interior se colocará una mini-bomba sumergible y un sensor de nivel. Estos elementos, junto con los circuitos controladores estarán contenido en el cuerpo del dispositivo. La bomba expelerá el líquido desde el interior del depósito hasta la zona de dosificación a través de una manguera de PVC flexible. La zona de dosificación de desinfectante deberá estar separada físicamente del depósito de manera de disponer de un espacio libre donde un sensor detecte la presencia de la mano (obstáculo) y se realice la descarga desde un pico dosificador ubicado en el extremo de la manguera tal como lo indica la Fig.2.

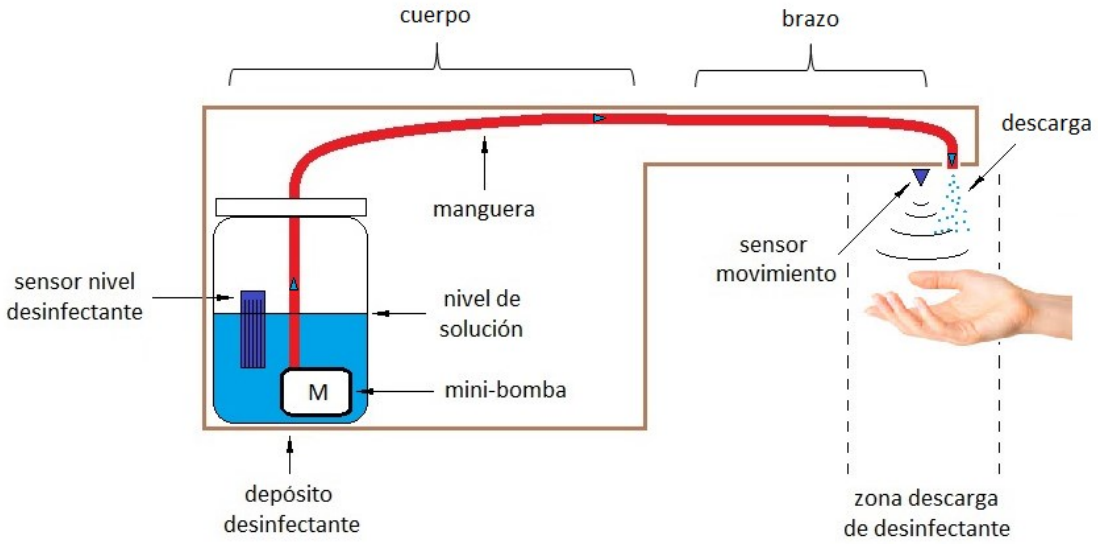

Fig.2: Esquema del diseño del prototipo del dispensador de solución hidro-alcohólica. Fuente: elaboración propia.

Con los conocimientos adquiridos en este curso de formación, se diseña una lógica de control de sensores y actuadores conectados a la placa Arduino UNO© y se simula en el programa TINKERCAD© (Fig.3).

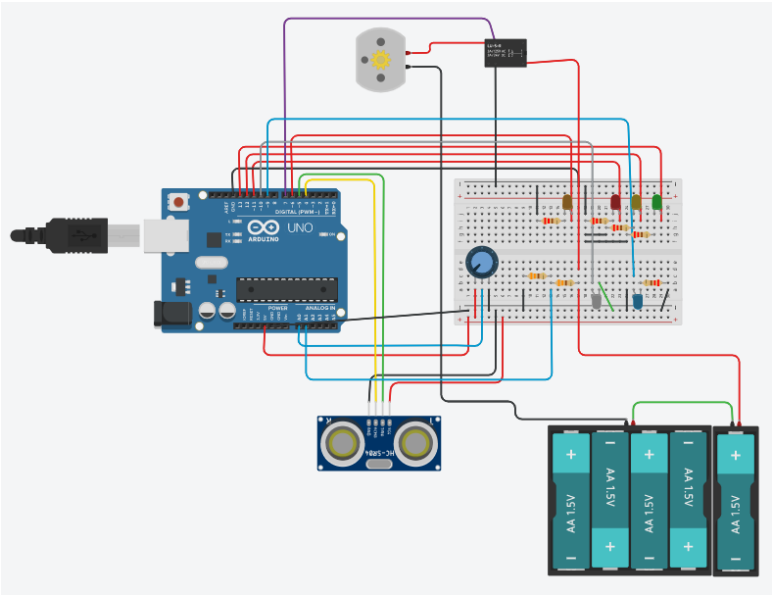

Fig. 3: Detalle del conexionado de sensores y actuadores del dispositivo simulado en TINKERCAD©.

Se obtiene después de varias pruebas una versión del código que hace que la lógica de control deseada para el dispositivo responda a las premisas de funcionamiento que se exponen en el apartado siguiente.

Funcionamiento

Cuando el usuario aproxima la mano a la zona donde descarga, un sensor ultrasónico la detecta y hace accionar la bomba realizando una descarga de desinfectante. La detección de la mano por el dispositivo activa una señal lumínica de color blanco ubicada en el extremo del brazo.

El nivel de carga del depósito es señalizado por medio del encendido de 3 leds de distintos colores. El nivel alto se señaliza con color verde, el nivel medio-bajo con color amarillo y el nivel bajo color rojo de señal intermitente.

El dispositivo puede ser alimentado con una fuente externa de 9 volts o bien de forma autónoma por pilas de recargables de litio-ion. En este caso, el dispositivo medirá el nivel de las baterías e indicará su carga mediante señales lumínicas: color azul para el nivel de carga apropiado para el funcionamiento del dispositivo y color naranja intermitente para indicar un nivel bajo de carga de las baterías.

Construcción del prototipo

Materiales

- 1 x placa Arduino UNO© y cable de conexión al PC.

- 1 x módulo de sensor de nivel de agua.

- 1 x módulo de sensor ultrasónico de distancia HC-SR04.

- 1 x mini-bomba de agua sumergible 100 l/h.

- 1 x mini-relé 5 V 10A.

- 1 x módulo de alimentación externa de 5V 1A.

- 6 x leds (verde, amarillo, rojo, blanco, azul, naranja).

- 6 x resistencias de 220 ohm.

- 1 x fuente de alimentación externa de 9V, 1 A.

- Hilo de estaño y barra de silicona.

Materiales reutilizados de otros proyectos

- 1 x frasco de vidrio con tapa hermética.

- 0,5 m. de manguera de PVC cristal flexible.

- 2 x baterías de ion-litio NCR 18650.

- 1 x interruptor unipolar.

- Cajas de cartón.

- 5 cm de cierre de gancho y bucle (velcro©).

- Soportes plásticos para placas y relé.

- Boquilla dosificadora de desinfectante.

- Botella de PET.

- Cables y pines de conexionado interno.

Herramientas necesarias

- Pistola de silicona.

- Alicate y pinza.

- Soldador de estaño.

- Cúter y tijera.

- Multímetro.

- Regla métrica.

Proceso constructivo paso a paso (Make up)

Se reutiliza un frasco para construir el depósito de solución hidro-alcohólica y en su interior se introduce el sensor de nivel y la mini-bomba. A modo de implementar un sistema recargable se adosa a la tapa del depósito el pico de una botella de PET tal como se muestra en la Fig. 4.

Fig. 4: Adaptación de pico de botella PET a la tapa del depósito.

También se efectúan dos agujeros a la tapa del frasco para pasar la manguera de PVC que conecta el depósito con la zona de descarga y los cables que conectan el sensor de nivel con la placa Arduino UNO© tal como muestra la flecha azul y roja en la Fig.5. respectivamente.

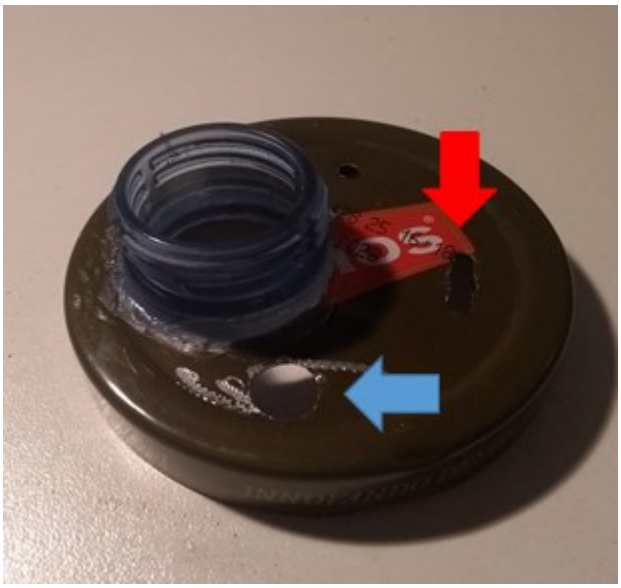

Fig. 5: Agujeros de paso de tubo PVC (flecha azul) y cables del sensor de nivel (rojo) en la tapa del depósito.

Tanto el sensor de nivel (Fig.6) como la mini-bomba se pegan al costado y al fondo del depósito tal como se indica respectivamente en la Fig. 7.

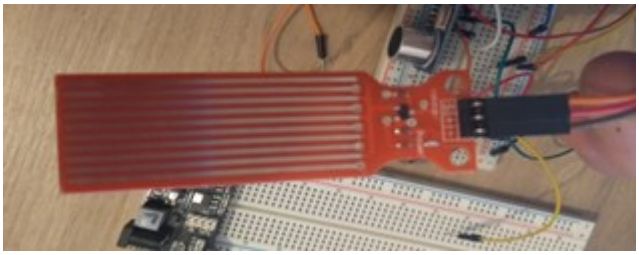

Fig. 6 – Sensor de nivel utilizado en el montaje.

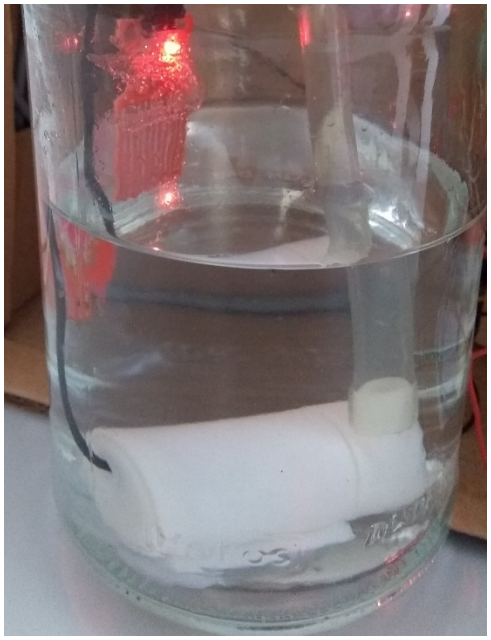

Fig. 7 – Mini-bomba y sensor de nivel adosados al depósito.

La manguera de PVC se conecta a la bomba y se dispone a través de la tapa metálica por el agujero practicado anteriormente tal como se muestra en la Fig. 8.

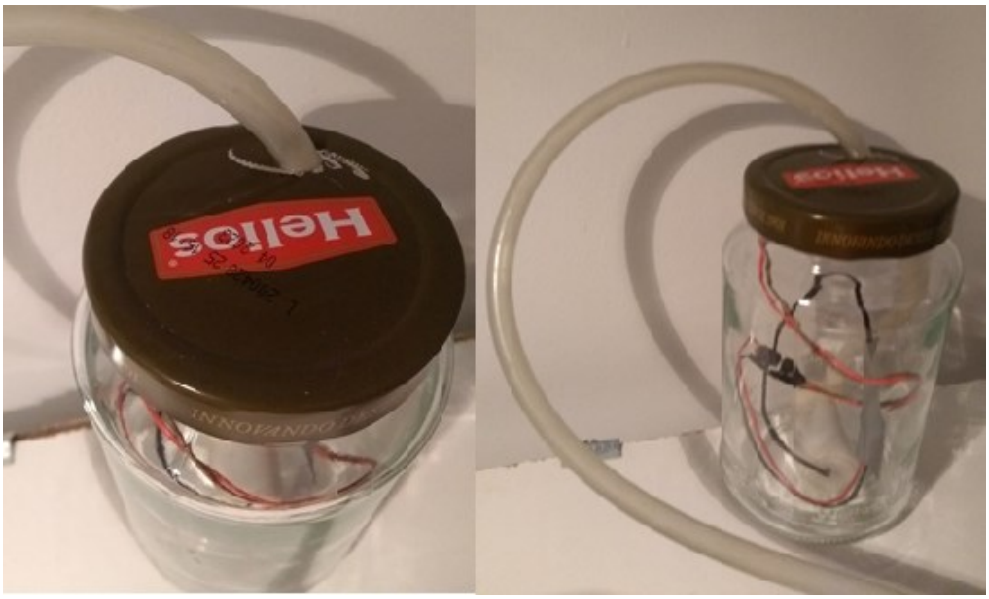

Fig. 8: Detalle del conexionado de la manguera de PVC cristal flexible.

A posteriori, se comienza la etapa de construcción física del sistema de control sobre una protoboard en la cual se montan todos los sensores y actuadores. A partir de este primer prototipo se ensaya el

funcionamiento de la mini-bomba y el sensor de nivel en situación de experimentación real utilizando agua tal como se muestra en la Fig. 9.

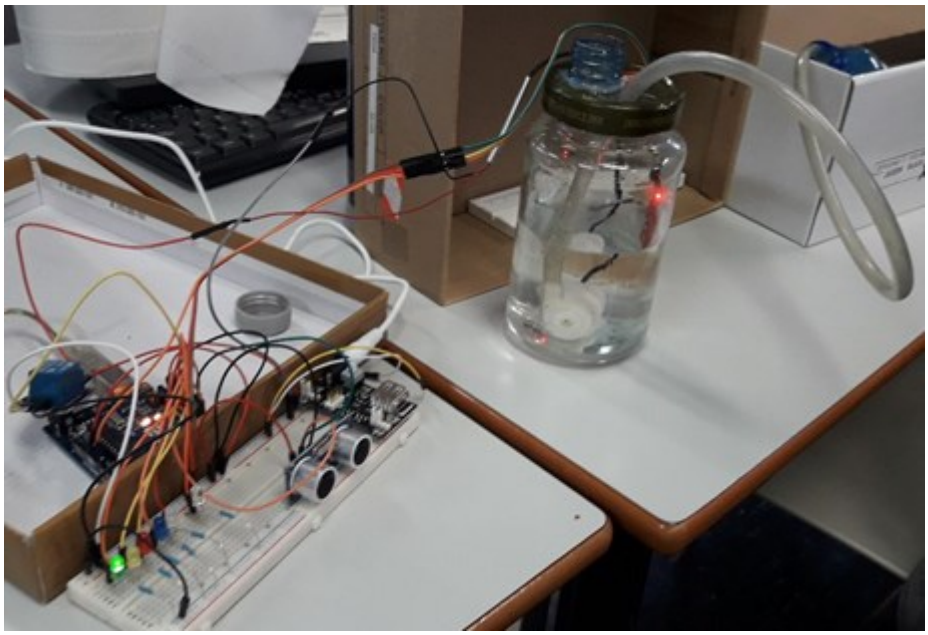

Fig. 9: Pruebas de funcionamiento de la mini-bomba y sensor de nivel realizada con la primera versión del prototipo.

Del mismo modo se realizan pruebas de alimentación autónoma del prototipo con baterías de ion-litio NCR1650 (Fig.10).

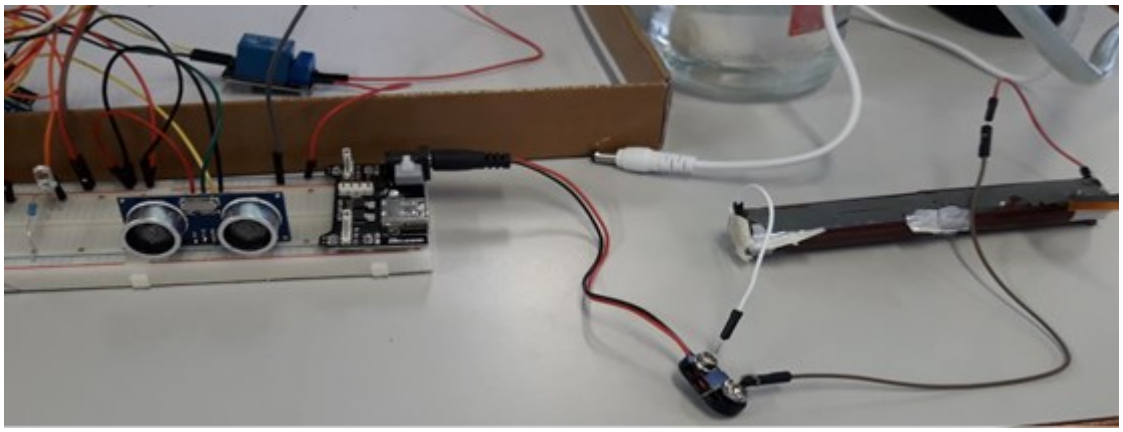

Fig. 10: Pruebas alimentación con baterías de la primera versión del prototipo.

Terminada exitosamente la etapa de pruebas se comienza la construcción física de la carcasa externa del cuerpo y del brazo del dispensador reutilizando cartón de distintos embalajes tal como se aprecia en la Fig.11 y 12.

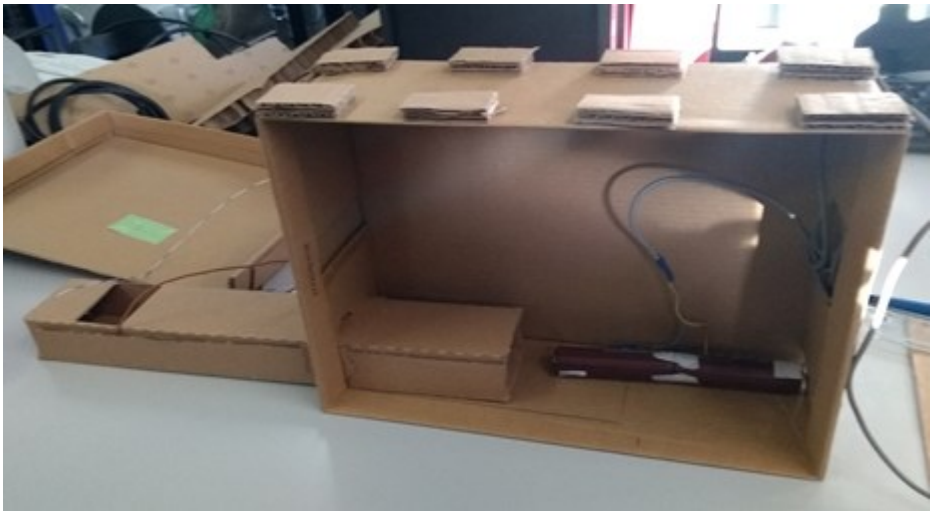

Fig.11: Vista lateral izquierda del dispositivo dispensador realizada en cartón.

Fig.12: Vista lateral derecha del dispositivo dispensador realizada en cartón.

Una vez concluida esta etapa se fijan a la estructura de cartón los distintos elementos constitutivos del diseño:

. Tarjeta Arduino UNO©: internamente sobre la parte frontal (Fig. 13).

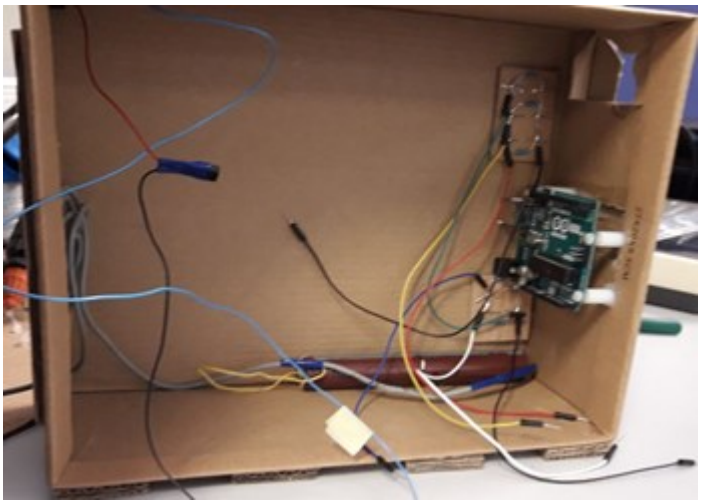

Fig.13: Tarjeta de control ubicada internamente en la parte frontal de la estructura.

. Indicadores luminosos de nivel y de carga de la batería (Fig. 14).

Fig.14: Leds de distintos colores ubicados en el panel lateral derecho del cuerpo del dispositivo.

. El módulo de alimentación externa fue adaptado a una base de cartón para fijarlo posteriormente en la parte interior trasera del cuerpo del dispensador (Fig. 15).

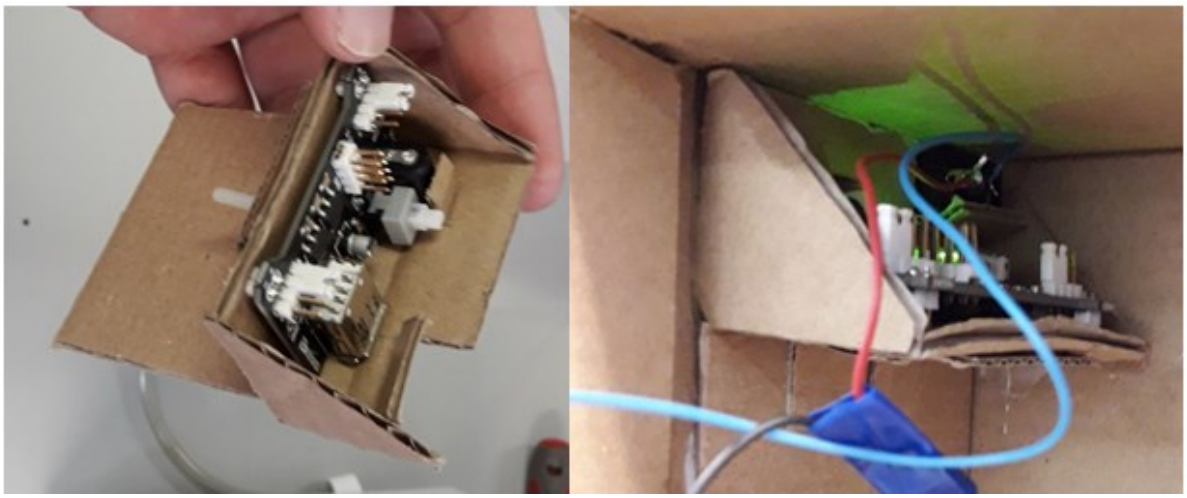

Fig.15: Alimentación externa montada sobre soporte de cartón y luego fijada al dispositivo.

. En la parte exterior trasera se coloca un interruptor para el encendido - apagado del dispositivo y una perforación para la alimentación de la fuente (Fig. 16).

Fig.16: Interruptor ON/OFF y perforación en la parte trasera para la alimentación de la fuente.

. En la parte interior también se fija el módulo de relé para el comando de la mini-bomba (Fig.17) dado que su consumo (>100 mA) excede la intensidad de corriente máxima permitida por el puerto de salida digital de la placa controladora.

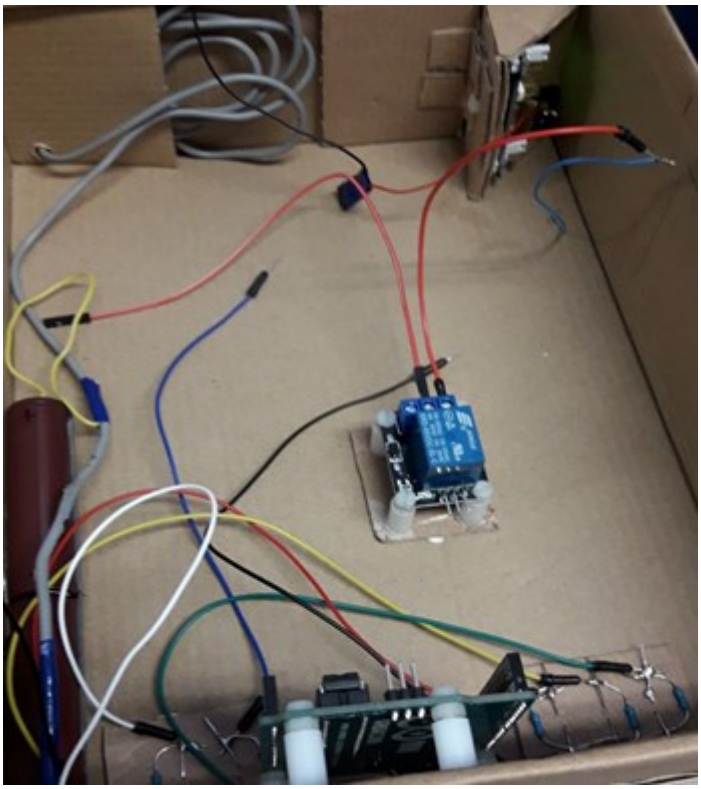

Fig.17: Relé de comando de la mini-bomba de solución desinfectante.

. Para la correcta dosificación de la solución en forma de spray, se reutiliza la boquilla de pulverización de un envase de solución desinfectante en desuso (Fig. 18).

Fig.18: Adaptación de la boquilla pulverizadora al extremo de la manguera de PVC.

Se coloca la manguera flexible de PVC, el sensor de distancia y el led indicador de activación de la descarga dentro del brazo de dispositivo (Fig. 19).

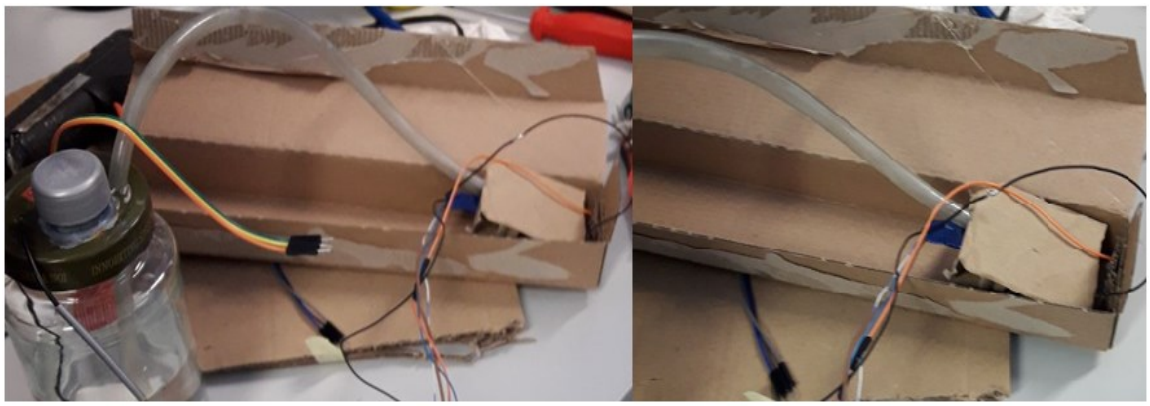

Fig. 19 – Detalle de la disposición de sensores y actuadores sobre el brazo del dispositivo.

La luz indicadora de la descarga se coloca en la parte frontal del brazo. Además, para la correcta detección de la mano se realiza una incisión rectangular en la parte inferior del brazo tal como se aprecia en la Fig.20.

Fig. 20 – Detalle de la disposición del led frontal y la incisión rectangular sobre el brazo del dispositivo.

El montaje del resto de los componentes se realiza a partir del correcto etiquetado del cableado y la disposición del depósito en su ubicación definitiva (Fig. 21 a 22).

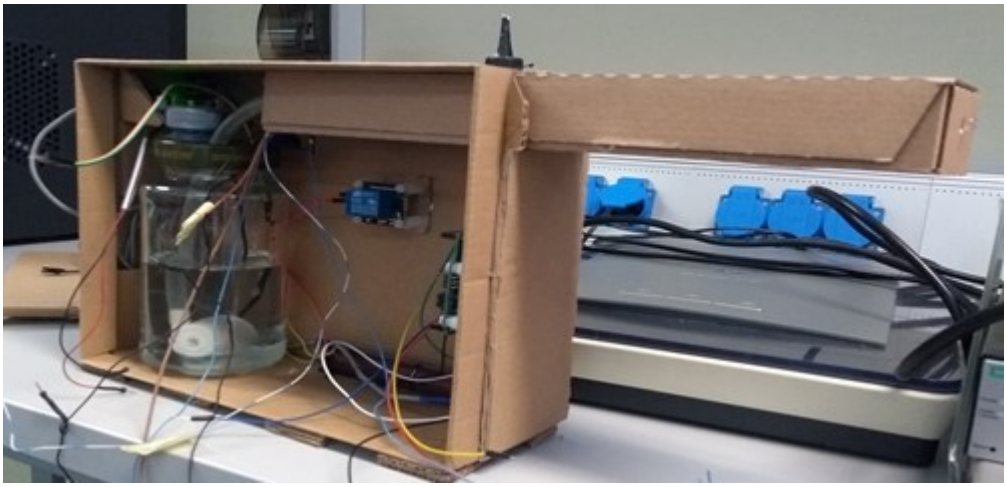

Fig. 21 – Disposición final de todos los componentes en el cuerpo del dispositivo – Vista frontal – lateral

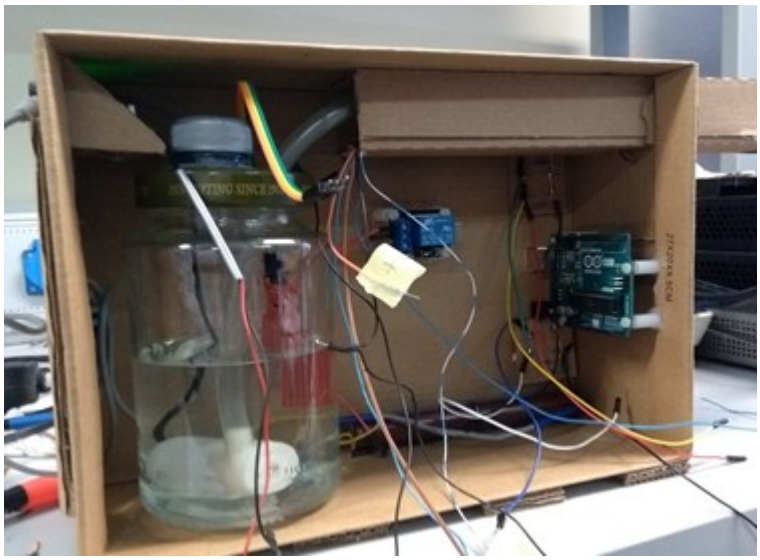

Fig. 22 – Disposición final de todos los componentes en el cuerpo del dispositivo – Vista anterior - lateral

Luego se adapta y se fija una tapa al cuerpo mediante el uso de cierre de gancho y bucle de manera de hacerla removible. Esto permite la verificación del dispositivo en caso de fallas, la reposición de baterías y reposición de desinfectante (Fig. 23).

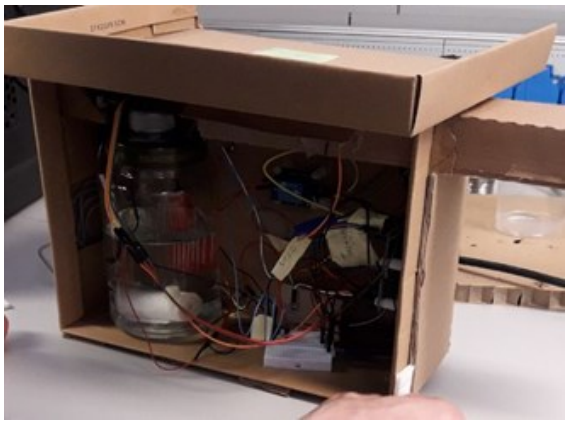

Fig. 23 – Adaptación de una tapa al cuerpo del dispositivo.

Finalmente se identifican los leds de nivel de líquido, estado de la batería y el conector de entrada a la placa controladora mediante carteles indicativos (Fig.24).

Fig. 24 – Letreros indicativos sobre un lateral del dispositivo

Código para ARDUINO UNO©

Primera parte: Definición de variables

```
// By Mikel Gurruchaga and Martin Marietta, noviembre 18 2020

//Definicion de variables
int LlevelPin = A0;             //pin analógico de entrada del nivel del líquido
int Llevel = 0;                 //variable indicadora del nivel actual de líquido [div]
int Lv_low = 50;                //variable indicadora de nivel bajo de líquido [div]
int Llow = 250;                 //variable indicadora de nivel medio de líquido [div]
int pin_red = 11;               //pin digital de salida del led rojo correspondiente al nivel muy bajo de líquido
int pin_yellow = 12;            //pin digital de salida del led amarillo correspondiente al nivel bajo del líquido
int pin_green = 13;             //pin digital de salida del led verde correspondiente al nivel alto del líquido
int TRIG = 5;                   //pin digital de salida del trigger del sensor ultrasónico
int ECCO = 4;                   //pin digital de entrada del ecco del sensor ultrasónico
int pin_orange = 6;             //pin digital de salida del led de actuación del sensor ultrasónico
int DURACION;                   //variable que almacena la duración del pulso del sensor ultrasónico [mseg]
unsigned int DISTANCIA;         //variable que almacena la distancia calculada entre objeto y sensor [cm]
int pin_pump = 7;               //pin digital de salida de activación del relé que comanda la mini-bomba de liquido
int dMAX = 10;                  //variable que almacena el valor máximo de detección del sensor ultrasónico [cm]
int carga = A1;                 //pin analógico de entrada del voltaje de la batería
int pin_azul = 9;               //pin digital de salida del led azul correspondiente al voltaje alto de la batería
int pin_blanc = 8;              //pin digital de salida del led blanco correspondiente al voltaje bajo de la batería
float vmedio = 675;             //variable indicadora del voltaje medio de la batería [div]
float vmini = 614;              //variable indicadora del voltaje mínimo de la batería [div]
float voltaje = 0;              //variable indicadora del voltaje actual de la batería [V]

// parpadeo de los leds
unsigned long tpast_red = millis();
long led_interval = 100;
int red_state = LOW;
unsigned long tpast_white = millis();
int white_state = LOW;

//delay de la duración de la activación de la bomba
unsigned long milis_comienzo_pump = millis();
int tpump = 1700;               //variable que indica el tiempo de actuación de la mini-bomba [mseg]
int KEY;                        //variable auxiliar que permite una descarga mientras se cumple con la condición de distancia
int start = 0;

//delay para frenar la reactivacion de la bomba
unsigned long milis_pausa_pump = millis();
int ppump = 3000;               //variable que indica una pausa de activación de la mini-bomba
int pausa = 0;                  //variable auxiliar
```

En esta parte, se definen las variables a utilizar. Como dato a resaltar, se declara la variable “DISTANCIA” del tipo “unsigned long” dado que en ocasiones daba números negativos y no se ejecutaba correctamente

la rutina LOOP. Otro dato a destacar es que para las variables “medio”, “mini” y “voltaje” se las declara del tipo “float” con el objeto de obtener valores más exactos de la tensión de las baterías.
Se ha de destacar que los valores de las variables “Lv_low”: nivel mínimo y “Llow”: nivel bajo de desinfectante en el depósito han sido determinadas mediante la calibración *in situ* de los niveles necesarios para la activación de los sensores luminosos: rojo intermitente (nivel bajo) y amarillo no parpadeante (nivel medio de desinfectante) correspondientemente.

Adicionalmente la variable “vmini”: nivel mínimo de voltaje de la batería de Litio-ion ha sido determinado como el número de divisiones correspondiente al valor 6 volts (3 volt por cada batería conectada en serie) correspondiente al voltaje mínimo a partir del cual la fuente de alimentación se vuelve inestable y no puede asegurar los 5 volts continuos y estables en la salida.

Ambos números de divisiones han sido calculadas a partir del voltaje Vout correspondiente a la terminología indicada en el divisor de tensión de la Fig. 26. Dado que se utiliza dos resistencias de 330 Ω el valor de Vout correspondiente a vmini es de 3 volts.

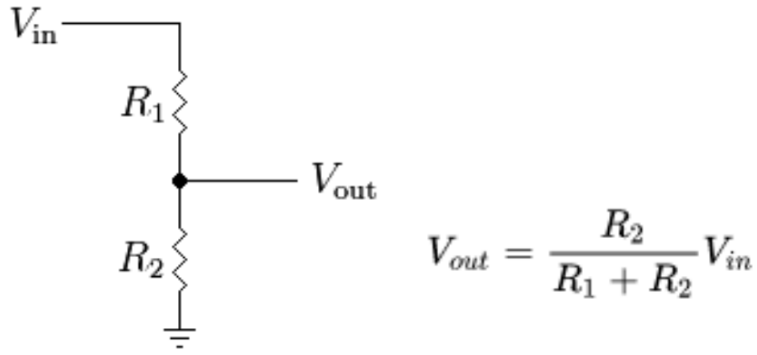

Fig. 26 – Divisor de tensión utilizado para el cálculo del número de divisiones.

Luego el número de divisiones correspondiente se calcula por regla de 3 simple:

$$\text{Vout= 3 volts} \rightarrow Nro.div. = \frac{3x1023}{5} = 613{,}8 \cong 614.$$

Así mismo el valor de la variable “vmedio”: nivel medio de voltaje ha sido fijado como el número de divisiones correspondiente al valor 3,3 volts que es el valor medio entre el nivel mínimo y el voltaje nominal de la batería (3,6 volts). El número de divisiones se calcula de forma similar al caso anterior:

$$\text{Vout= 3,3 volts} \rightarrow Nro.div. = \frac{3{,}3x1023}{5} = 675{,}2 \cong 675.$$

También se define la función “millis” y así crear el efecto de parpadeo de los leds de color rojo y blanco sin detener el funcionamiento del programa.

Segunda parte: Configuración

```
//Configuracion
void setup()
 {
 Serial.begin(9600);
 pinMode(pin_red, OUTPUT);                  //declara el pin_red como salida
 pinMode(pin_yellow, OUTPUT);               //declara el pin_yellow como salida
 pinMode(pin_green, OUTPUT);                //declara el pin_green como salida
 pinMode(pin_orange, OUTPUT);               //declara el pin_orange como salida
 pinMode(TRIG, OUTPUT);                     //declara el pin TRIG como salida
 pinMode(ECCO, INPUT);                      //declara el pin ECCO como entrada
 pinMode(pin_pump, OUTPUT);                 //declara el pin_pump como salida
 pinMode(pin_azul,OUTPUT);                  //declara pin azul como salida
```

```
 pinMode(pin_blanc,OUTPUT);                    //declara el pin blanco como salida
 }
```

Tercera parte: Ciclo repetitivo

```
//Ciclo repetitivo
void loop()
 {
  if (start == 0){
  Serial.print("start \n");
  start = 1;
  }
 //Sensor de nivel
 Llevel = analogRead(LlevelPin);               //lee la entrada analógica del pin de nivel de líquido
 Serial.print("Nivel: ");
 Serial.println(Llevel);

  if  (Llevel < Lv_low)                        //si el nivel de líquido es muy bajo y es obligada su reposición
  {
  digitalWrite(pin_green,LOW);
  digitalWrite(pin_yellow,LOW);

  //parpadeo led rojo
  unsigned long tnow = millis();
  if(tnow - tpast_red > led_interval)
   {
   tpast_red = tnow;
   if (red_state == HIGH)
    {
    red_state = LOW;
    }  else  {
    red_state = HIGH;
    }
   }
   digitalWrite(pin_red,red_state);
  }  else  {
    if  (Llevel < Llow)                        //si el nivel de líquido bajo, y es sugerida su reposición
     {
     digitalWrite(pin_red,LOW);
     digitalWrite(pin_yellow,HIGH);
     digitalWrite(pin_green,LOW);
     }  else  {
     digitalWrite(pin_yellow,LOW);             //se activa el led de nivel adecuado de líquido
     digitalWrite(pin_green,HIGH);
     }
  //delay(20);//  debug value
  }

 //Sensor de proximidad
  digitalWrite(TRIG,HIGH);                     //envía un pulso de 1ms por el pin TRIGGER
  digitalWrite(TRIG,LOW);
  DURACION = pulseIn(ECCO,HIGH);               //mide el tiempo que tarda el pin ECCO en ponerse en HIGH y
                                               almacena el valor [us]
  DISTANCIA = DURACION / 58.4;                 //calcula la distancia entre el objeto y el sensor [cm]

 //Serial.print("Distancia: ");
// Serial.println(DISTANCIA);
  //Serial.println(KEY);
  if (DISTANCIA > dMAX)                        //sin proximidad suficiente para activarse
   {
   KEY = 0;
   }
  if (DISTANCIA <= dMAX && DISTANCIA > 0 && KEY == 0 && pausa == 0)        //si la distancia de activación es menor o
                                                                           igual a la indicada y la mini-bomba no
```

```
                                                    está activo y no hay confirmación de la
                                                    pausa
   {
   Serial.println("yes");
   digitalWrite(pin_orange,HIGH);          //enciende la luz blanca
   digitalWrite(pin_pump,HIGH);            //enciende la bomba
   milis_comienzo_pump = millis();         //registra el tiempo actual de comienzo de operación de la bomba
   KEY = 1;                                //setea la variable KEY en 1 para que no siga operando la bomba por más que la
                                           distancia sea la apropiada para el disparo
   pausa = 1;                              //setea la variable pausa en 1 para activar la pausa del sistema y evitar múltiples
                                           disparos
   }
  unsigned long milisactual = millis();
  if(milisactual - milis_comienzo_pump > tpump)        //delay de uso de la mini-bomba, de forma de limitar la cantidad de
                                                       líquido que se dispensa
   {
   milis_comienzo_pump = milisactual;
   digitalWrite(pin_orange,LOW);           //apaga la luz naranja
   digitalWrite(pin_pump,LOW);             //se detiene la bomba
   }

   //pausa de funcionamiento de la bomba para evitar disparos múltiples
   if(milisactual - milis_pausa_pump > ppump)
   {
    milis_pausa_pump = milisactual;
    pausa = 0;
   }
 int bateria = analogRead(carga);
 //Serial.print("voltaje: ");
 //Serial.println(bateria);
 //delay(100);
  if (bateria > medio){
 digitalWrite(pin_azul,HIGH);
 digitalWrite(pin_blanc,LOW);
 }
 if (bateria < medio){
  digitalWrite(pin_blanc,HIGH);
  digitalWrite(pin_azul,LOW);
 }
  if (bateria < mini){
  digitalWrite(pin_blanc,HIGH);
  digitalWrite(pin_azul,LOW);
  unsigned long ahora = millis();
   if(ahora - tpast_white > 200)          //delay para el parpadeo del led blanco
   {
   tpast_white = ahora;
   digitalWrite(pin_blanc,LOW);
 }   }   }
```

Al inicio de la rutina “Loop” se describe el funcionamiento de los 3 indicadores luminosos (verde, amarillo y rojo) para indicar el nivel de líquido desinfectante y se implementa la función “millis” para crear el parpadeo del led rojo cuando el nivel de líquido es inferior al necesario para el funcionamiento adecuado de la bomba.

Posteriormente se describe el código que permite el funcionamiento del sensor de distancia y a su vez activar la bomba para activar la descarga de solución hidro-alcohólica y el led que indica dicha descarga. También se utiliza la función “millis” para dosificar la cantidad de líquido a suministrar y una pausa en la cual no se volverá a activar la mini-bomba.

Por último, se encuentra la rutina de activación de los indicadores luminosos del voltaje de la batería. Según sea el valor de dicho voltaje se encenderá un led azul o blanco parpadeante correspondiente a nivel alto o nivel bajo de la carga de la batería.

Dificultades encontradas por el camino y retos a superar

Todo proyecto conlleva sus retos y dificultades inesperadas y este proyecto no fue la excepción. El primer reto que a enfrentar era hacer que todos los sensores y componentes funcionaran al mismo tiempo, se descubrió que usando la función “ping” con el sensor ultrasonido para hacer el cálculo de la distancia se creaba un conflicto con la operación de la mini-bomba, por lo que se optó por usar la librería “newping”. Desgraciadamente esta librería traía aparejados conflictos con las funciones de “delay” por lo que finalmente se optó por la función “millis” y se decidió hacer la operación del sensor de ultrasonido de forma manual, configurando la placa controladora para mandar y recibir un pulso. Con este valor se calcula la distancia al obstáculo.

Otro desafío que se presentó fue cambiar las funciones “delay” por “millis” para que el programa no se detuviera cada vez que se acciona la mini bomba. Realizar la función de parpadeo de los leds usando la función “millis” no traería aparejada ninguna dificultad al ser un bucle fijo, pero al tener 2 pausas diferentes (una que limita la cantidad de solución que dispensa la mini-bomba y otra para frenar la reactivación de la misma) no se podía utilizar la misma función que en los indicadores luminosos, así que se decidió agregar la variable “pausa” en forma de contador para poder obtener ambas pausas.

Además de las dificultades a solventar en el montaje físico del proyecto, también el montaje nos propuso retos a los que hubo que darle solución. Uno de estos problemas fue relativo a las fuentes de alimentación: al hacer las pruebas con el prototipo conectado a una fuente de alimentación externa el prototipo funcionaba de acuerdo a lo previsto, pero al hacerlo funcionar con la batería existía un fallo de funcionamiento. Con la realización de pruebas se descubre que con la batería alimentando el prototipo se reiniciaba la placa controladora, causando el mal funcionamiento del programa subido a la misma. Finalmente se determinó que el problema se debía a la variación de tensión en bornes de la batería al activarse la mini bomba, así que se decidió conectar la batería a la fuente de alimentación externa que al funcionar como regulador de voltaje y proveía de tensión estable y continua de 5 volts a la placa controladora.

Trabajo futuro

Esta primera versión del prototipo puede ser mejorada en muchos aspectos. Por ejemplo, se podría incorporar la posibilidad de regular la cantidad de solución a dosificar mediante un potenciómetro.
Otra funcionalidad que se podría implementar referente al funcionamiento autónomo, sería la de carga de baterías incorporada al dispositivo.
En lo referente al aspecto exterior, se podría construir una cobertura plástica adaptada a su uso y condiciones de operación. También es posible dotar al dispensador de conectividad a internet, de manera de poder visualizar vía web el nivel de líquido remanente en el depósito. Esta funcionalidad cobra sentido en el caso de simplificar la logística de reposición de solución hidro-alcohólica en un establecimiento donde múltiples dispensadores proporcionen servicio simultáneo.
Se propone al docente formador, realizar ambos proyectos en cursos posteriores de impresión 3D y de IOT (Internet of Things) que se dicten posteriormente en este centro de formación.

Conclusiones

Este proyecto se diseñó con la finalidad de dar servicio de higiene de manos a los asistentes del curso de Placas Robóticas que a su vez lo dispuso como la etapa final y aplicativa de todo el conocimiento adquirido. El prototipo se construyó íntegramente en clase utilizando en la manera de lo posible la reutilización de materiales como medida de cuidado del medio ambiente. Su puesta en marcha y funcionamiento no estuvieron exentos de contratiempos y resolución de problemas técnicos, los cuales fueron resueltos exitosamente. Al día de la redacción de esta memoria el dispositivo se encuentra en su cuarto día de servicio ininterrumpido sin presentar problemas importantes en su operación.