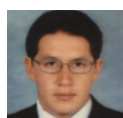




Dispensador automático de alimento para mascotas

Automatic food dispenser for pets



Icaza Daniel

Docente de la Universidad Católica de Cuenca. Director de Carrera de Ing. Eléctrica.



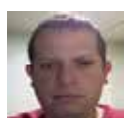
Gabino Tito

Estudiante de la Universidad Católica de Cuenca. Carrera de Ingeniería Eléctrica.



Padilla Wilma

Estudiante de la Universidad Católica de Cuenca. Carrera de Ingeniería Eléctrica.



Pesantez Geovanny

Estudiante de la Universidad Católica de Cuenca. Carrera de Ingeniería Eléctrica.



Pozo Freddy

Estudiante de la Universidad Católica de Cuenca. Carrera de Ingeniería Eléctrica.

Resumen

Este equipo tiene como finalidad ayudarnos en la alimentación de nuestra mascota en horarios definidos, basta con una simple configuración inicial y trabajará de forma continua indefinidamente, con respaldo del suministro de energía eléctrica. Con el contenedor lleno de alimento utilizándolo para razas grandes tendríamos 6 días de alimentación continua, teniendo en cuenta de la cantidad de alimento que contiene el dispensador es de 5kg, los días podrían variar dependiendo de la cantidad de alimento que se desee despachar a su mascota. Este equipo puede racionalizar la comida de su mascota y evitar que los dueños deban estar permanentemente alimentándolo, es fácil de transportarlo de un lugar a otro y resulta muy útil cuando se deba dejar su hogar por varios días y a su mascota no le faltará su alimento.

Palabras clave: dispensador; arduino; automático; alimentación de mascotas; microcontrolador

Abstract

This equipment has as a purpose we help in the feeding of our pet in defined hours, it is enough with a simple initial configuration and we work of continuous form indefinitely, with backing of the electric power supply. With the full container of food used for large breeds 6 days of continuous feed, taking into account the amount of food contained in the 5kg dispenser, the days depend on the amount of food that you want to dispense your pet. This equipment you can rationalize your pet's food and prevent owners from being permanently fed, it is easy to transport from one place to another and is very useful when you have to leave your home for several days and your pet will not miss its food.

Keywords: dispenser; arduino; automatic; pet feed; microcontroller

Introducción

En los hogares el cuidado nutricional de los animales es muy importante en horas específicas, más aún si por algún motivo el animal tiene que pasar algún tiempo solo sin quien pueda cuidar su nutrición, en vista de ello se pensó en el diseño de un dispensador de comida para mascotas dado que hoy en día en casa no es muy común en las familias hayan personas permanentes en el hogar y cuiden sus mascotas (Estévez P., 2005; Prudente Tixteco, 2014; Grimaldi D., 2014).

Son aproximadamente 13.000 toneladas anuales de alimento balanceado para mascotas que se venden en el Ecuador; la mayoría de ellas para consumo canino, ya que, en ciudades grandes del país como Quito, Guayaquil y Cuenca que poseen entre de 0.8 y 2.5 millones de habitantes respectivamente, existe un 20% de población canina en cada ciudad. Esto quiere decir que por ejemplo, si en Quito habitan hoy en día 2 millones de personas, existe una población aproximada de 400 mil canes, de los cuales 300 mil pueden ser estimados para estadísticas y como mercado potencial para un posible negocio relacionado con estos animales. (Hervás V., 2010; Estévez P., 2005; Prudente Tixteco, 2014).

Dado que en las familias ecuatorianas la mascota preferida es el perro nos propusimos fabricar un dispensador de comida para perros sin que ello implique que lo pudiéramos utilizar con el propósito de alimentar otra mascota, el cual dispone de autonomía completa, gracias a que puede funcionar con suministro de energía continuo y en caso de no disponer de energía eléctrica funcionaria mediante sus dos baterías como respaldo. Se podrá configurar la cantidad de alimento, dependiendo del tipo de raza, ya sea grande, mediana o pequeña, y podrá escoger la hora que desea suministrar el alimento dos veces al día (Navajas G. et al., 2015; Hervás V., 2010).

El prototipo diseñado cuenta con una especie de contenedor de almacenamiento del alimento, una compuerta que abre el paso a la comida, un circuito electrónico y el arduino que actúa para enviar una señal y abastecer de alimento al equipo cuando previamente ha consumido su mascota.

El circuito electrónico consta de temporizadores para la apertura de la puerta, con eso se calcula la cantidad necesaria para cada raza. Si el animal es grande, requiere más alimentación, entonces el temporizador mantendrá la puerta abierta por más tiempo; pero si se trata de una mascota pequeña, la compuerta se cerrará más rápido. Para la apertura

y cierre de la compuerta utilizaremos un tornillo sin fin accionado por un motor de plumas de vehículo.

Objetivos

Objetivo General.

- Diseñar y construir un dispensador de alimento para mascotas.

Objetivos Específicos.

- Lograr que nuestro dispositivo sea programable a las necesidades del usuario con flexibilidad en el horario.
- Dispensar la cantidad de alimento necesario de acuerdo al tamaño de la mascota.

Materiales y métodos

PARTE FISICA:

- Recipiente plástico.
- Tubos de PVC de 2 pulg. 1m.
- Una "T" de PVC de 2 pulg.
- Codo de PVC 2 pulg.
- 30 cm de grilon para construcción del tornillo sin fin.
- 4 llantas giratorias.
- Caja de protección.
- 40 pernos
- Un yarda de plástico
- Amarras plásticas.

PARTE DE CONTROL

- Motor de plumas para vehículo 12V CD.
- 2 baterías de 12V dc
- Un arduino Mega 2560.
- Display 16X2
- Adaptador 120V CA -12V CD.
- 5 pulsantes
- Resistencias 1.2 KΩ
- Adaptador de 12-24 a 5 V
- Relé 12V
- 1 interruptor
- Sensor de distancia ultrasonido HC-SR04
- Reloj de tiempo real (RTC).



En medida de las posibilidades se ha buscado que varios elementos que conforman este equipo sean reciclados, tales como el recipiente plástico, el tubo y accesorios de PVC, el adaptador puede ser de celular. Los demás elementos deben presentar muy buenas características constructivas por lo que no pueden ser reciclados.

Marco teórico.

La alimentación del perro: Los perros, como todos los seres vivos, deben seguir una correcta alimentación para poder realizar sus actividades diarias. En general, el tamaño y la actividad física, dicta el requerimiento calórico del perro. Para conocer cómo se puede satisfacer dicho requerimiento calórico, es necesario conocer los nutrientes principales que se encuentra en todos los alimentos; a dichos nutrientes, se les conoce como macronutrientes (Navajas G. et al, 2015)

Macronutrientes: La dieta de un perro debe de cubrir sus requisitos energéticos. Para lograr lo anterior, la dieta debe de estar basada en los 3 macronutrientes esenciales; los carbohidratos, las proteínas y los lípidos. Los carbohidratos son la fuente de energía más importante para cualquier ser vivo. Desempeñan una función importante para la estructura y el funcionamiento de los tejidos y los órganos, y son particularmente importantes para el correcto funcionamiento del cerebro y del sistema nervioso. Existen distintos tipos de carbohidratos, pero en general, el contenido total de energía es de 4.15 kcal/g. (Navajas G. et al, 2015; Hervás V., 2010).

En la gran mayoría de los casos se ha acostumbrado a los perros a que se alimenten, ya sea con comida preparada en casa o con algún balanceado existente en el mercado, de un plato o recipiente en el cual previamente se ha dosificado de alguna forma cierta cantidad de alimento. Si se han fijado en este hecho, es decir, en el momento justo en el que el perro se está alimentando de dicho plato o recipiente, se puede observar que el perro se adapta a la manera de alimentarse que se le ofrece y que a simple vista no se necesita nada más que este plato o recipiente con la cantidad, apropiada o no, de alimento balanceado que se le está proporcionando, ya que en muchos casos se desconocen las cantidades requeridas por los perros según su raza y edad principalmente, para satisfacer la necesidad de alimentación de las mascotas. (Franco G., 2010; Isaza-Echavarría C. & Zapata-Álvarez P. A., 2014; León Quenguan J. D. & Rueda Almario D., 2016; Torres Huerta L. A. 2016).

Un comedero es un recipiente en donde se vierte la comida para los animales de cría o domésticos.

Los comederos suelen consistir en cuencos de metal, plástico o cerámica en donde se coloca el alimento de forma periódica para la alimentación del animal. Existen diversos tipos de comederos en función del tamaño y características de la mascota.

Los comederos con tolva permiten alimentar a la mascota sin necesidad de la presencia del dueño. Consisten de un depósito en donde se almacena el alimento que se va dispensando por su base sobre un cuenco. De este modo, el animal dispone de una cantidad cierta de alimento pero no necesita ser dispensado por la mano humana (Navajas G. et al, 2015).

Algunos comederos cuentan con sistemas para limitar la cantidad ingerida por la mascota. En general, consisten en obstáculos en el interior del plato que obligan al animal a buscar la comida alargando el período de ingestión y favoreciendo la sensación de hartazgo. Algunos tienen un núcleo central rígido haciendo que la comida se mueva alrededor (Franco G., 2010; Isaza-Echavarría C. & Zapata-Álvarez P. A., 2014; León Quenguan J. D. & Rueda Almario D., 2016).

Entre los accesorios asociados a los comederos figuran:

- Soportes para dos cuencos: comedero y bebedero
- Soportes elevados para uno o dos cuencos de modo que el animal no tenga que agacharse.
- Bases de realce para elevar el comedero unos centímetros sobre el suelo

Desarrollo del equipo

Arduino es una plataforma física computacional open-hardware basada en una sencilla placa con entradas y salidas (E/S), analógicas y digitales, y en un entorno de desarrollo que implementa el lenguaje Processing/Wiring. Arduino puede utilizarse en el desarrollo de objetos interactivos autónomos o puede conectarse a un PC a través del puerto serie utilizando lenguajes como Flash, Processing, MaxMSP, etc.

Los motores absorben energía eléctrica de la red, nos aportan energía mecánica en su eje el mismo que acoplado al tornillo sin fin nos da un movimiento lineal capaz de trasladar el alimento de mascota de forma progresiva en la cantidad programada. No se necesita invertir el sentido de giro de los motores, salvo por situaciones de mantenimiento (Hervás V., 2010).

Un tornillo sin fin puede ser construido utilizando las maquinas herramientas en el torno paralelo.

Se realizan varias medidas en el envase plástico, para acoplar los elementos al recipiente, recortamos el espacio para la caja de proyectos y para el tubo de 2".



Ilustración 1.- Recipientes donde irán montados todos los elementos mecánicos y electrónicos.

Se diseño el tornillo sin fin en AutoCAD para proceder a construirlo en material grillón, su eje es de 2cm de diámetro interior, la altura de los filetes para que circule la comida es de 1.5cm y el paso es de 2cm, su giro es hacia la izquierda con una entrada.

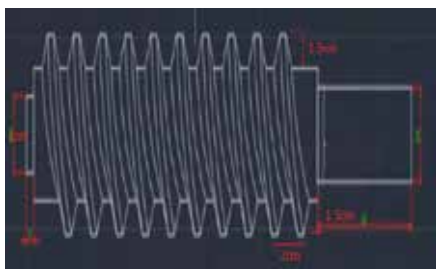


Ilustración 2. Construcción del tornillo sin fin.

Realizado un diseño del tornillo sin fin de Arquímedes, procedemos a construirlo con 30cm de Grillón, el cual tomara la forma deseada en el torno paralelo. Es necesario tener en cuenta el sentido de giro del mandril que se utiliza ya que de este dependerá el sentido de rosca del tornillo sin fin. En nuestro caso el mandril girará a la derecha lo cual determina que el tornillo sin fin tenga rosca izquierda.



Ilustración 3.- Construcción del tornillo sin fin en el torno paralelo.

Al cumplir el mecanizado,acoplamos el eje de grillón con el motor y con la tubería de PVC en el recipiente plástico, fijamos las llantas en el exterior y las baterías de 12V a 1.5A conectadas en paralelo entre si y paralelo a nuestra fuente de 120V CA a 12VCC. Dirigimos todos los conductores hacia la posición de la caja de proyectos (Gimaldi D., 2014).

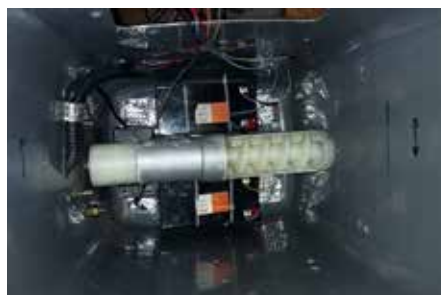


Ilustración 4.- Vista superior donde se aprecia el montaje mecánico y electrónico.

Colocamos el plástico PVC antideslizante transparente en el interior del recipiente con el fin de que las croquetas no caigan hacia las instalaciones mecánicas y de esta forma dirigiendo las croquetas solo hacia el tornillo sin fin. De esta manera se concluye la construcción mecánica y eléctrica de nuestro proyecto.



Ilustración 5.- Ajustes del tornillo sin fin.

Para hacer funcionar el dispensador automático de alimento para mascotas es necesario configurar la hora por lo que se utilizara un RTC Reloj de Tiempo Real con el integrado ds-1307.

Para poder visualizar las diferentes configuraciones de la parte de control es necesario un display de 16x2 y para poder controlar estas configuraciones se utilizara 5 pulsantes el cual estará distribuido de esta manera un pulsante para Menú, un pulsante para Enter, uno para escape, y los dos últimos para aumentar y disminuir los valores de las variables.

Para medir la cantidad de contenido es necesario un sensor de distancia en este caso es el HC-SR04.

La parte de control tiene la siguiente disposición de los diferentes componentes conectados a nuestra placa de Arduino 2560 (Navajas G. et all, 2015).

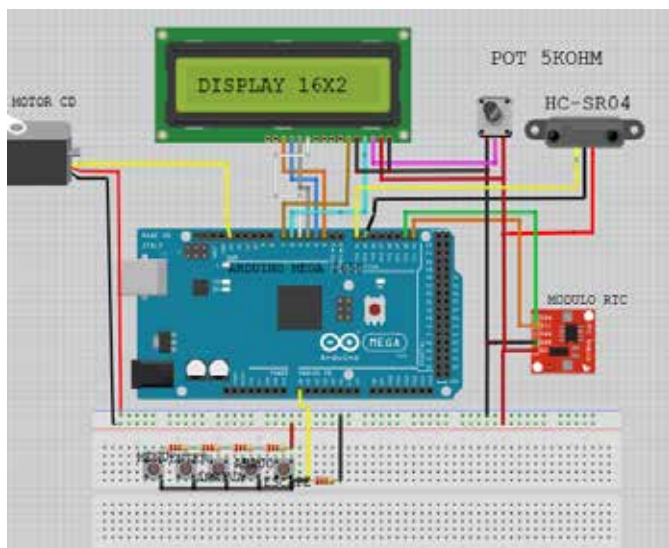


Ilustración 6.- disposición de los diferentes circuitos integrados en la placa de arduino 2650

En la gráfica se puede observar que los pulsantes están conectado con las resistencias en serie, al pulsar cada una produce un diferente potencial por lo cual esta se conecta al puerto analógico (A0).

El shield RTC de arduino consta de 4 pines de los cuales dos son pines de alimentación y dos son pines para la comunicación de estos dos dispositivos, por lo que es una comunicación I2R y estan conectados a los pines 20 (SDA) y 21 (SCL) (Navajas G. et all, 2015).

El sensor de distancia ultrasónico consta 4 pines: "VCC" conectado a la salida de 5V de la placa, "Trig" conectado al pin digital 14 de la placa encargado de enviar el pulso ultrasónico, "Echo" al pin de entrada digital 13 que recibirá el eco de dicho pulso y "GND" a tierra.

El display 16x2 tiene la siguiente conexión.
 LCD RS pin to digital pin 7 del arduino.
 LCD Enable pin to digital pin 6 del arduino.
 LCD D4 pin to digital pin 5 del arduino.
 LCD D5 pin to digital pin 4 del arduino.

LCD D6 pin to digital pin 3 del arduino.
 LCD D7 pin to digital pin 2 del arduino.

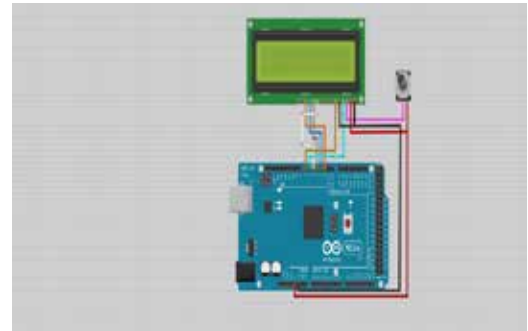


Ilustración 7.- Conexión del LCD con el Arduino Mega 2650

Y por último el pin 13 del arduino está conectado a la señal del relé que controla el movimiento del motor.

Una vez conocido como está organizado el circuito de control de este proyecto, se procede a armar en la caja de proyectos.



Ilustración 8.- Montaje del sistema electrónico.

Una vez montado el circuito de control en la caja de proyecto.



Ilustración 9.- Circuito de control en la caja de proyecto

Se procede con la programación del arduino para lo cual se utiliza el IDE propio de arduino.



Ilustración 10 Página de descarga del arduino compiler.

En la programación declaramos las librerías que controlan cada uno de los componentes del circuito de control.

```
#include <LiquidCrystal.h> // DLL PARA CONTROLAR EL LCD 16X2
#include <EEPROM.h> // DLL CONTROLA LA EEPROM DEL ARDUINO
#include <Wire.h> // PUERTO DE COMUNICACION I2C
#include "RTClib.h" // CONTROLA EL RTC
RTC_DS1307 RTC; // CONTROLADOR DEL RTC
LiquidCrystal lcd(7, 6, 5, 4, 3, 2); // CONFIGURACION DE LOS PUERTOS DEL ARDUINO CON EL LCD
```

Variables usadas en el programa.

El programa consta de las siguientes subsunciones.

```
const int analogInPin = A0; // Pin Analógico Botones del Menu
int Hora_Uno=0; // Almacena la Variable de la hora uno
int Min_Uno=30; // Almacena las de los minutos de la primera hora
int Hora_Dos=16; // Almacena la Variable de la hora uno
int Min_Dos=30; // Almacena las de los minutos de la segunda hora
int Nivel_Max=30; // Almacena el nivel max de alimento
int Nivel_Min=10; // Almacena el nivel min de alimento
int Raza_Grande=20; // Almacena el tiempo de duración de giro de motor para la raza grande
int Raza_Mediana=20; // Almacena el tiempo de duración de giro de motor para la raza mediana
int Raza_pequena=20; // Almacena el tiempo de duración de giro de motor para la raza pequeña
int Tip_raza=1; // Guarda la conf del tipo de raza
#define Fecho 14 // pin que recibe la señal del hc-sr04
#define Ptring 15 // pin que envia la señal del hc-sr04
long duracion, distancia; // calculo de la distancia del hc-sr04
int motor=13; //señal del rele para mover el motor
int aux1=1; //auxiliar uno
int aux=0; // auxiliar 2
```

Menú() corresponde al control del menú para las determinadas configuraciones (Prudente Tixeco, 2014)

```
void menu()
{
  lcd.clear();
  lcd.setCursor(0,0);lcd.print("Conf Tip Raza");delay(1000);
  TipodeRaza(); //tipo de raza
  lcd.setCursor(0,0);lcd.print("Conf Tiemp X raza");delay(1000);lcd.clear();
  Conf_Tiemp(); //configura el tiempo para activar la comida
  lcd.setCursor(0,0);lcd.print("Conf Nivel Alimento");delay(1000);lcd.clear();
  Conf_nivel(); // nivel de alimento
  lcd.setCursor(0,0);lcd.print("Hora de Alimentación");delay(1000);lcd.clear();
  Conf_hora();
  lcd.setCursor(0,0);lcd.print("Salir Menu");delay(1000);lcd.clear();
}
```

Tipo de Raza() esta función permite seleccionar el tipo de raza y lo guarda en la memoria eeprom.

```
void TipodeRaza()
{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("ELIJA RAZA :");
  lcd.setCursor(0,1);
  while(botonas !=1)
  {
    if(botonas !=2)
    {
      aux1=aux1+1;
      delay(500);
      if(aux1>3){aux1=1;}
    }

    lcd.setCursor(0,1);
    lcd.print(aux1);
    switch (aux1) {
      case 1: lcd.setCursor(0,1); lcd.print("RAZA GRANDE"); break;
      case 2: lcd.setCursor(0,1); lcd.print("RAZA MEDIANA"); break;
      case 3: lcd.setCursor(0,1); lcd.print("RAZA PEQUEÑA"); break;
    }
  }

  lcd.clear();
  switch (aux1)
  {
    case 1: lcd.setCursor(0,0); lcd.print("RAZA GRANDE"); break;
    case 2: lcd.setCursor(0,0); lcd.print("RAZA MEDIANA"); break;
    case 3: lcd.setCursor(0,0); lcd.print("RAZA PEQUEÑA"); break;
  }

  EEPROM.write(10,aux1);
  lcd.setCursor(0,1); lcd.print("GUARDADO");
}
```

Distanciometro() esta función envía el pulso al hc-sr04 y recibe la señal del mismo por lo cual realiza los cálculos necesarios y devuelve un valor entero este valor está en cm

```
int Distanciometro()
{
  digitalWrite(Ptring, LOW);
  delayMicroseconds(2);
  digitalWrite(Ptring, HIGH); // genera el pulso de trigger por 10ms
  delayMicroseconds(10);
  digitalWrite(Ptring, LOW);
  duracion = pulseIn(Pecho, HIGH);
  distancia = (duracion/2) / 29; // calcula la distancia en centimetros
  return distancia;
}
```

Hora() esta función crea un objeto nuevo y permite leer la hora del RTC visualiza en el display, también compara las horas programadas para la alimentación para mandar la señal al motor.

```
void hora()
{
  DateTime now = RTC.now();
  DateTime future (now.year()+ 1 * 86400 + 30);
  lcd.setCursor(0, 1); lcd.print("Hora: ");
  lcd.print(future.hour(), 2); lcd.print(":");
  lcd.print(future.minute(), 2); lcd.print(":");
  lcd.print(future.second(), 2); lcd.print(":");
  lcd.print(future.month(), 2);
  if((future.hour() != EEPROM.read(1)) || (future.minute() != EEPROM.read(2)))
  {
    switch (EEPROM.read(10))
    {
      case 1: if(EEPROM.read(7) > future.second()) digitalWrite(motor, HIGH); else digitalWrite(motor, LOW); break;
      case 2: if(EEPROM.read(8) > future.second()) digitalWrite(motor, HIGH); else digitalWrite(motor, LOW); break;
      case 3: if(EEPROM.read(9) > future.second()) digitalWrite(motor, HIGH); else digitalWrite(motor, LOW); break;
    }
  }
  if((future.hour() != EEPROM.read(3)) || (future.minute() != EEPROM.read(4)))
  {
    switch (EEPROM.read(10))
    {
      case 1: if(EEPROM.read(7) > future.second()) digitalWrite(motor, HIGH); else digitalWrite(motor, LOW); break;
      case 2: if(EEPROM.read(8) > future.second()) digitalWrite(motor, HIGH); else digitalWrite(motor, LOW); break;
      case 3: if(EEPROM.read(9) > future.second()) digitalWrite(motor, HIGH); else digitalWrite(motor, LOW); break;
    }
  }
}
```

Conf_Tiemp() esta función permite dar un tiempo determinado para que gire el motor dependiendo el tipo de raza, este valor configurado se guarda en la eeprom.

```
void Conf_Tiemp()
{
  Raza_Grande= EEPROM.read(7); //Memoria 7 max tiempo COMIDA PAZA GRANDE
  Raza_Mediana= EEPROM.read(8); //Memoria 8 MAXIMO TIEMPO PAZA MEDIANA
  Raza_pequena= EEPROM.read(9); //Memoria 9 MAXIMO TIEMPO PAZA PEQUENA

  lcd.setCursor(0,0); lcd.print("RAZA GRANDE"); lcd.setCursor(0,1); lcd.print(Raza_Grande);
  Raza_Grande=cambiar_numeros(Raza_Grande,2); EEPROM.write(7,Raza_Grande); lcd.clear();
  lcd.print("Dato guardado"); delay(1000);

  lcd.setCursor(0,0); lcd.print("RAZA MEDIANA"); lcd.setCursor(0,1); lcd.print(Raza_Mediana);
  Raza_Mediana=cambiar_numeros(Raza_Mediana,2); EEPROM.write(8,Raza_Mediana); lcd.clear();
  lcd.print("Dato guardado"); delay(1000);

  lcd.setCursor(0,0); lcd.print("RAZA PEQUEÑA"); lcd.setCursor(0,1); lcd.print(Raza_pequena);
  Raza_pequena=cambiar_numeros(Raza_pequena,2); EEPROM.write(9,Raza_pequena); lcd.clear();
  lcd.print("Dato guardado"); delay(1000);
}
```

Conf_nivel() esta función permite configurar el nivel max y el nivel mínimo de contenido de alimento, lo guarda dentro de la memoria eeprom del arduino.

```
void Conf_nivel()
{ Nivel_Max=EEPROM.read(5); // Memoria 5 nivel max de alimento para canes
  Nivel_Min=EEPROM.read(6); // Memoria 6 nivel min de alimento para canes
  lcd.setCursor(0,0); lcd.print("NIVEL MAX ALIM:");
  lcd.setCursor(0,1); lcd.print(Nivel_Max);lcd.print(" ca");
  while(botones() !=1)
  {
    if (botones()==3) (Nivel_Max=Nivel_Max+1; delay(200));
    if (botones()==4) (Nivel_Max=Nivel_Max-1; delay(200));
    lcd.setCursor(0,1);lcd.print(Nivel_Max);
  }
  EEPROM.write(5,Nivel_Max); lcd.clear();
  lcd.print("Dato guardado");
  delay(1000);
  lcd.clear();
  lcd.setCursor(0,0); lcd.print("NIVEL MIN ALIM:");
  lcd.setCursor(0,1); lcd.print(Nivel_Min);lcd.print(" ca");
  while (botones() !=1)
  {
    if (botones()==3) (Nivel_Min=Nivel_Min+1; delay(200));
    if (botones()==4) (Nivel_Min=Nivel_Min-1; delay(200));
    lcd.setCursor(0,1);lcd.print(Nivel_Min);
  }
  EEPROM.write(6,Nivel_Min); lcd.clear();
  lcd.print("Dato guardado");
  delay(1000);
}
```

Conf_hora() esta función permite el ingreso de las diferentes horas y minutos para que se mueva el motor, estas se guardan en variables y a su vez en la memoria eeprom del arduino.

```
void Conf_hora()
{ Hora_Dia=EEPROM.read(1); // Memoria 1 la primera hora de alimentacion
  Min_Dia=EEPROM.read(2); // Memoria 2 las primeros minutos de alimentacion
  Hora_Noche=EEPROM.read(3); // Memoria 3 la segunda hora de alimentacion
  Min_Noche=EEPROM.read(4); // Memoria 4 segundos minutos de alimentacion
  lcd.clear();
  //CAMBIA LA HORA DE LA PRIMERA COMIDA DEL PERRITO//
  lcd.setCursor(0,0); lcd.print("H 1er ALIMENTO"); Hora_Dia= cambiar_numero(Hora_Dia,0);
  EEPROM.write(1, Hora_Dia); // Memoria 1 la primera hora de alimentacion
  lcd.clear(); lcd.setCursor(0,0); lcd.print("Minutos de Comida"); lcd.setCursor(0,1); lcd.print( EEPROM.read(2)); delay(1000); lcd.clear();
  //CAMBIA LA LOS MINUTOS DE LA PRIMERA COMIDA DEL PERRITO//
  lcd.setCursor(0,0); lcd.print("Min de ALIMENTO"); Min_Dia= cambiar_numero(Min_Dia,1);
  EEPROM.write(2, Min_Dia); // Memoria 2 la primera hora de alimentacion
  lcd.clear(); lcd.setCursor(0,0); lcd.print("Segundo Comida"); lcd.setCursor(0,1); lcd.print( EEPROM.read(3)); delay(1000); lcd.clear();
  //CAMBIA LA HORA DE LA SEGUNDA COMIDA DEL PERRITO//
  lcd.setCursor(0,0); lcd.print("H 2o ALIMENTO"); Hora_Noche= cambiar_numero(Hora_Noche,0);
  EEPROM.write(3, Hora_Noche); // Memoria 3 la primera hora de alimentacion
  lcd.clear(); lcd.setCursor(0,0); lcd.print("Minutos de Comida"); lcd.setCursor(0,1); lcd.print( EEPROM.read(4)); delay(1000); lcd.clear();
  //CAMBIA LA LOS MINUTOS DE LA SEGUNDA COMIDA DEL PERRITO//
  lcd.setCursor(0,0); lcd.print("Min 2o ALIMENTO"); Min_Noche= cambiar_numero(Min_Noche,1);
  EEPROM.write(4, Min_Noche); // Memoria 4 la primera hora de alimentacion
  lcd.clear(); lcd.setCursor(0,0); lcd.print("Segundo Comida"); lcd.setCursor(0,1); lcd.print( EEPROM.read(4)); delay(1000);
}
```

botones() esta función lee el puerto analógico A0 del arduino, en el cual estan colocados pulsantes con resistencia en serie, identifica la diferencia de potencial y va asignando un rango mínimo y máximo para cada botón, esta función devuelve el número del botón que se esta pulsando.

```
int botones()
{
  int Tecla=0;
  Tecla = analogRead(analogInPin);
  if ((Tecla<1030)&&(Tecla>600))
  { return 1; }
  if ((Tecla<559)&&(Tecla>400))
  { return 3; }
  if ((Tecla<400)&&(Tecla>330))
  { return 2; }
  if ((Tecla<329)&&(Tecla>230))
  { return 5; }
  if ((Tecla<229)&&(Tecla>100))
  { return 4; }
  return 0;
}
```

Este dispensador automático está facultado para funcionar conectado a un toma de 120V y mediante un adaptador el ingreso de voltaje es de 12V, en caso de falta de suministro de energía utilizará sus baterías de 12V a 1.5A conectadas en paralelo, lo cual le otorga una autonomía de funcionamiento de largo tiempo hasta que retorne el servicio de energía de la red pública.



Ilustración 11.- Vista principal del dispensador automatico de comida para mascota.

Nuestro equipo tiene un peso de 11.8kg, puede almacenar 5kg de alimento.

Resultados y discusión.

El proyecto “dispensador automático de alimento para mascotas ” desarrollo un prototipo que pueda dosificar el alimento para mascotas de manera programada dependiendo del tamaño de la mascota. El diseño y la estructura mecánica del equipo consta de un tanque de almacenamiento y un sistema dosificador. El tanque o tolva es de plástico y sus propiedades físicas cumplen con los requerimientos para el almacenamiento de comida y es un material liviano que permite el fácil transporte del usuario. El sistema dosificador funciona con base en un motor DC que regula la dosificación del alimento activando un tornillo sin fin para recorrer internamente el alimento.

A continuación se da a conocer los resultados y programación básica para su funcionamiento:

Tabla 1.- Alimentación según tamaño de mascotas.

Alimentación diaria				
Raza	Tiempo por porción (seg)	Kg	Porciones por día	Días de autonomía
Grande	59	0.37	2	6
Mediana	30	0.2	2	12
Pequeña	15	0.11	2	22

Este proyecto comparado con otros existentes son de fácil programación, robusto no se vira fácilmente ya que en muchas ocasiones productos de similares características, pueden ser derribados cuando mascotas de gran tamaño se acerca.

El dispensador para mascotas provee el alimento suficiente dependiendo de cada raza y así se evitan desperdicios y contaminación del ambiente donde se encuentra la mascota.

Los dueños de las mascotas pueden sentirse seguros en que la alimentación será permanente a pesar que exista corte de energía el equipo seguirá funcionando de manera autónoma.

Conclusiones

Este equipo es de gran importancia y ayuda para personas que poseen mascotas, se garantiza la disponibilidad de una porción de alimento dependiendo del tamaño de la mascota, sin embargo hay que considerar los días de autonomía que tiene el equipo para abastecer nuevamente de una buena cantidad de alimento.

El dispensador automático de comida para mascotas permite configurar dos horarios al día, elegir el tamaño de la raza del perro, conocer el nivel de alimento disponible, la visualización en el display podrá variar desde 4cm (lleno) hasta 34cm (vacío), nuestro equipo brinda confiabilidad y tranquilidad de que nuestras mascotas siempre dispondrán de alimento en los horarios deseados, gracias a un banco de baterías que respalda la energía.

Este dispensador automático también aporta a que no existan desperdicios de alimento y también a proteger nuestro medio ambiente del derrame de alimento por diversos lugares del hogar.

Se sugiere profundizar los estudios para proveer alimentación a animales de mayor tamaño y con me-

canismos distintos a tornillos sin fin.

Recomendaciones

Aunque se disponga de este dispensador automático en su hogar, es de suma importancia recordar el compartir con su mascota momentos de esparcimiento y cuidado, ya que este equipo brindará alimento permanente pero no amor.

Referencias bibliográficas

Estévez P. (2005) Manual básico para la educación del cachorro, p. 1-36.

Franco G., Galicia J., & Ostria D. (2010). Desarrollo de un sistema de dosificación automático de alimentos para equinos. Instituto Politécnico Nacional, México D.F.

Grimaldi D. (2014) Proyecto de factibilidad para el ensamble y comercialización de un dispensador de alimentos para mascotas, en el área metropolitana de San Salvador, El Salvador (Doctoral dissertation, Universidad de El Salvador).

Hervás V. (2010). Motores de corriente continua. Obtenido de Motores de corriente continua: <https://iesvillalbahervastecnologia.files.wordpress.com/2010/01/motores-electricos-par-te-i1.pdf>

Isaza-Echavarría C. & Zapata-Álvarez P. A. (2014). Intervención remota a un espacio domotizado para mascotas (Doctoral dissertation, Biomédica, Mecatrónica y Mecánica).

León Quenguan J. D. & Rueda Almario D. (2016). Dispensador automático de comida para mascotas, programable y controlado remotamente [recurso electrónico] (Doctoral dissertation).

Navajas G., Cabezas J. S. R., Ávila K. G. & Prada S. R. (2015). Diseño y construcción de un dosificador automático de alimento para perros de raza media. Memorias.

Prudente Tixteco (2014) Diseño e implementación de un sistema dispensador de alimento para mascota controlado vía ethernet.

Torres Huerta L. A. (2016). Desarrollar un modelo de negocio de productos de perfumería, accesorios y farmacia para el cuidado de mascotas en Chile.

Universidad Católica de Cuenca. Carrera de Ingeniería Eléctrica.