

PET FEEDER

PROYECTO COMEDERO PARA MASCOTAS



1 DE ENERO DE 2020

CANCLINI, CHIERCHIE & LAURENZI
Instituto Técnico La Piedad



INDICE:

¿Qué es Pet Feeder?.....	2
¿Como surge?:.....	2
Funcionamiento:.....	3
Circuito:.....	4
Funcionamiento Lógico (Programación).....	6
Desarrollo de la página WEB:.....	6
HTML:	8
CSS:.....	10
Componentes electrónicos necesarios:.....	10
COMPONENTES.....	10
PRECIO	10



¿Qué es Pet Feeder?

Pet Feeder es un comedero de mascotas programable, este mismo puede ser programado por su dueño fácilmente desde la página web del dispositivo a través de internet (WiFi).

Lo que el dueño puede modificar es:

- ❖ El tiempo que deposita el alimento, con esto se puede calcular la cantidad que quiera servirle a la mascota, y dependiendo de las necesidades del animal, puede variar.
- ❖ Hora en la que el dueño quiera que se sirva la ración, esta configurada por defecto que se servirá cada día.
- ❖ Estado del motor, para saber si el motor está en funcionamiento.

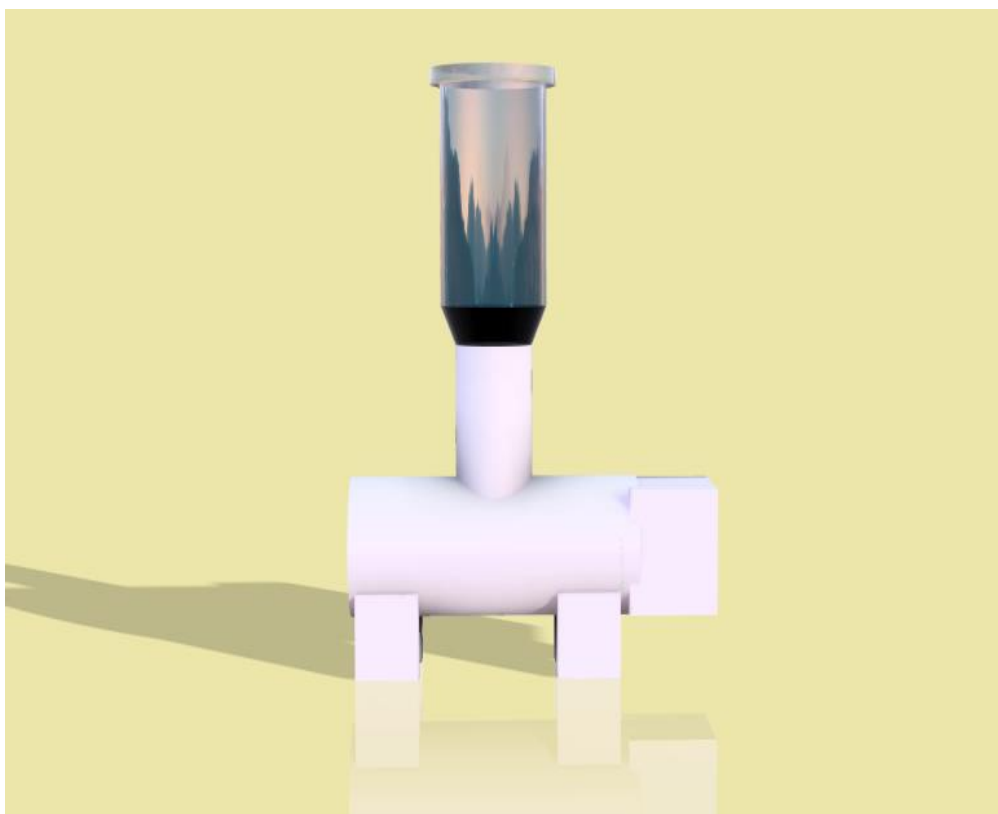
Todas configuraciones básicas para un agradable ambiente y uso sencillo de la aplicación para aquellas personas que no sean interesados en el lado logístico de la aplicación y poder estar conformes. Básicamente al ser automático, facilita en gran medida la tarea de administrar a la mascota en lo que a alimentos respecta

El diseño del dispositivo está hecho y pensado para plástico PLA, el cual es biodegradable y es el que comúnmente utilizan las impresoras 3D. Se compone de 5 partes: Por un lado, esta el tubo principal de 3 entradas, el cajón donde se guarda la parte electrónica, el recipiente del alimento, la hélice que lo deposita y los soportes de la carcasa.

¿Como surge?:

En la búsqueda por automatizar una tarea cotidiana, con nuestro equipo pensamos en las tareas que necesitan cierto tiempo para ser administradas, eso sumado a la necesidad por hacerlo completamente personalizado y a medida para cada uno, es que nace esta idea.

C



Funcionamiento:

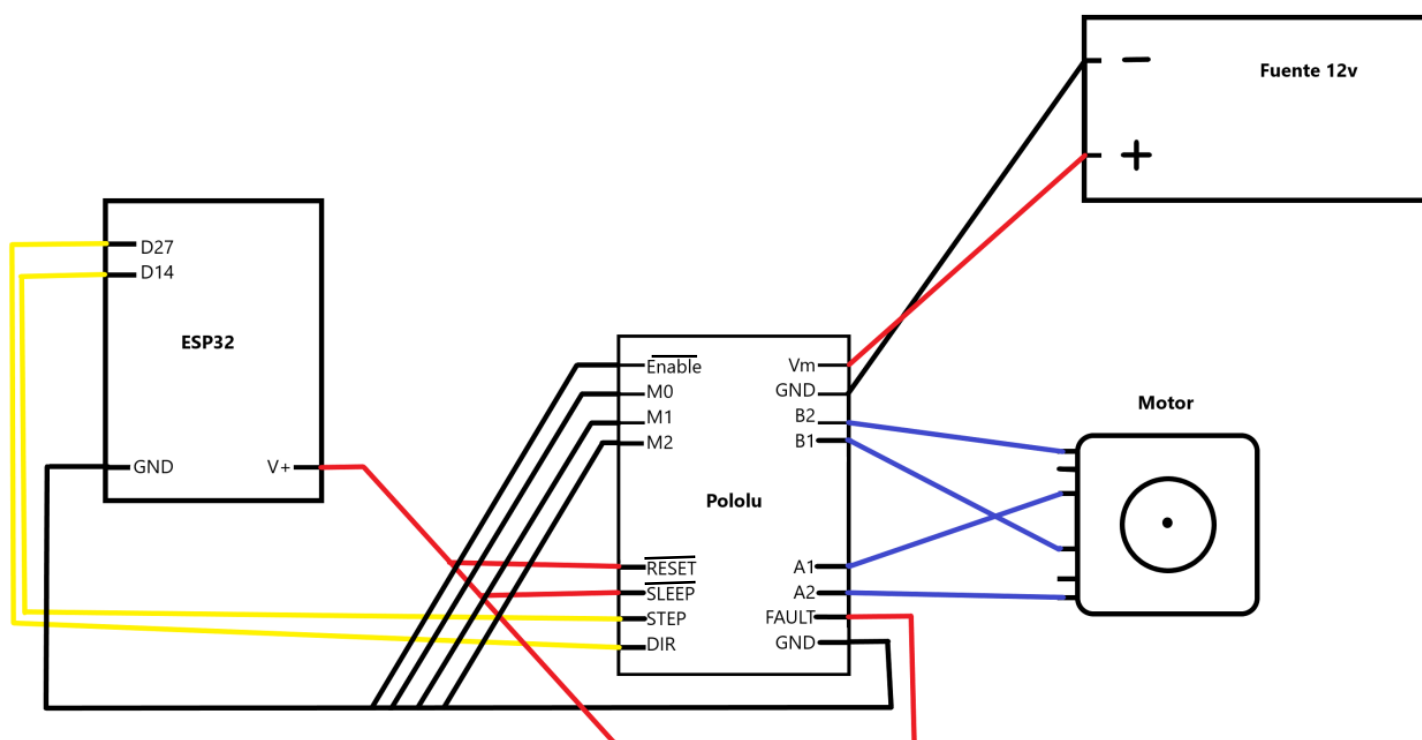
Mediante la utilización del motor Nema 17, el ESP32 le envía una señal para que rote, este se encuentra conectado a una espiral o hélice, la cual desplaza el alimento hacia el recipiente que se encuentra en el extremo del tubo. El ensamble es muy sencillo, la parte principal es el dichoso tubo y hélice. Esta última se encuentra dentro del mismo y está conectada, en uno de sus extremos, al motor que se encuentra en el cajón de la electrónica, donde además reside, obviamente el Esp32.

Después de que pasa la cantidad de tiempo programada, el motor se detendrá en la posición donde se encuentre, esto no es un problema, debido a que, al utilizar la forma de una hélice, el alimento, sale gradualmente.



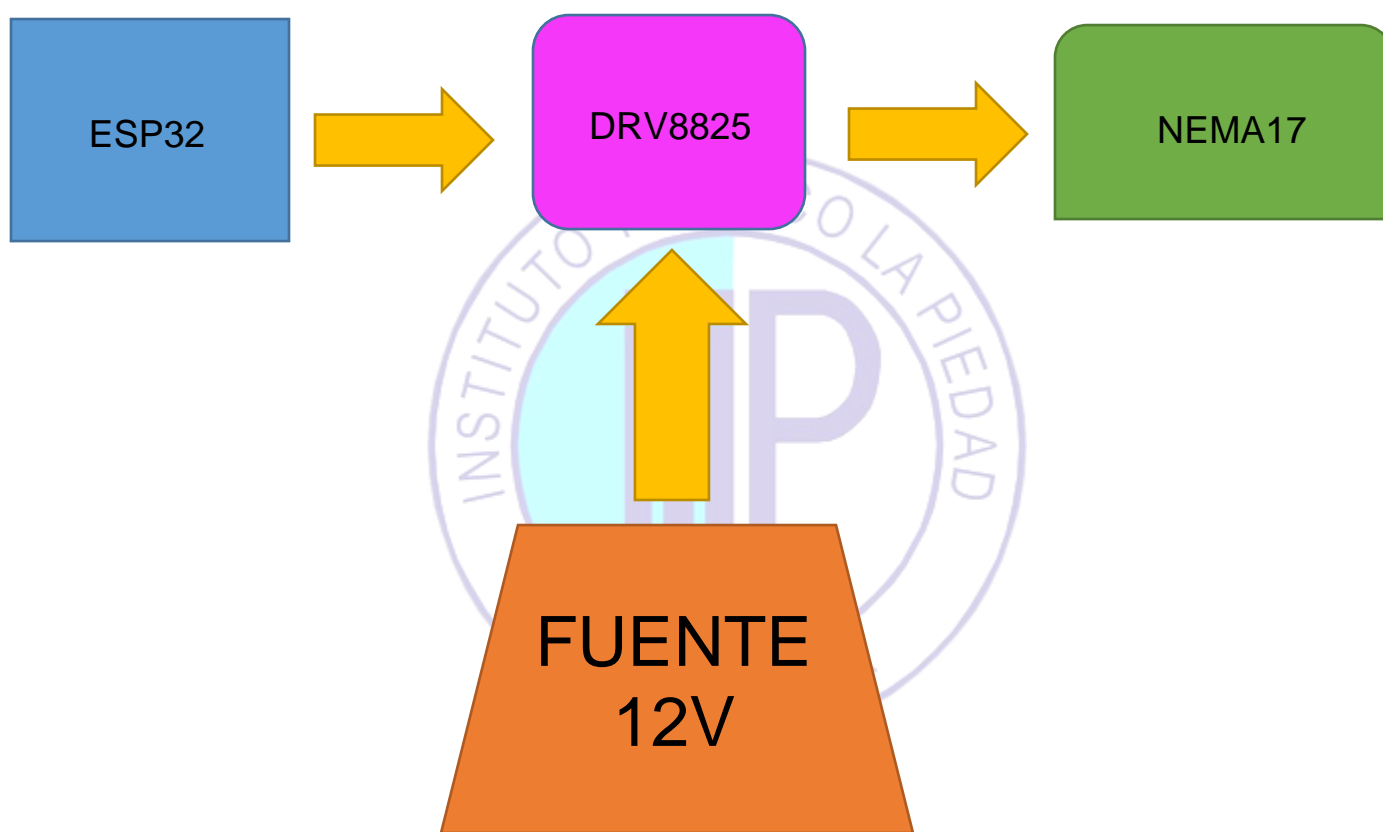
Círculo:

Todos los dispositivos de este proyecto, están escondidos en una caja que se encuentra en uno de los extremos del tubo y se encuentran conectados con el siguiente diseño, para poseer una mejor gestión de cables y evitar cualquier desconexión del circuito.



Como se poder ver en la circuitería, dentro de la caja encontraremos: el esp32, el controlador driver pololu drv8825, una fuente de 12v, el ya mencionado motor y los cables que los interconectan.

Para simplificar la explicación podemos ver el siguiente diagrama:



Básicamente todo el circuito se conecta a través de el driver drv8825 o “pololu”. Este cumple la función de “manejar” el motor, pero para poder hacerlo requiere de las correspondientes instrucciones, en este caso del esp32, el cual le indica cuanto girara el motor y en qué dirección. No obstante, necesita de un suministro de energía adicional para poder lograr mover el motor, es por eso que aparece una fuente de 12v.

Funcionamiento Lógico (Programación)

Para realizar la programación del esp32, utilizamos el IDE de Arduino, que, mediante el uso de las librerías correspondientes, permite configurar y programar el esp32 como si fuera un Arduino común.

Dentro de la programación podemos distinguir entre 5 partes principales: Para el manejo y administración de datos utilizamos una función y librería de Esp32 como lo es el SPPIFS (SPI Flash File System), el cual nos permite hacer uso de la memoria flash de 4 Mb que posee el Esp32, mediante este almacenamos la página web (HTML, CSS Y fotos) y las variables que guardan las configuraciones. La principal ventaja de esto es que al ser una memoria flash, al apagar el dispositivo las configuraciones pasadas no se pierden.

También se utilizamos las librerías de wifi que permiten al esp32, conectarse a una red Wifi, montar un servidor y administrarlo, es gracias a esto que podemos abrir un pagina HTML, lo que nos lleva a nuestro tercer elemento. Mediante las librerías de WebServer podemos, como dijimos, administrar el server y realizar las HTML request, GET and POST, que básicamente son las funciones que permiten al pagina interactuar o intercambiar datos con el usuario. Es gracias a la utilización y combinación de estos 2 sistemas que la pagina almacena, administra y controla el dispositivo.

Por si fuera poco, también el dispositivo mediante las ya mencionadas, librerías wifi, puede conectarse a un servidor externo, y pedir los datos horarios y un calendario, todo esto se almacena en variables.

Por ultimo y como quinto elemento tenemos las funciones del motor mismo, las que realizan, al fin y al cabo, “acciones reales”. Es aquí donde finalmente se utilizan las variables o configuraciones almacenadas para proceder con el depósito de alimento.

Para terminar de entender el funcionamiento de este algoritmo veamos el siguiente diagrama:

Se cargan las librerías
(WIFI-SERVIDOR NTP-WEBSERVER-SPIFFS)



Se declaran Variables y
Funciones



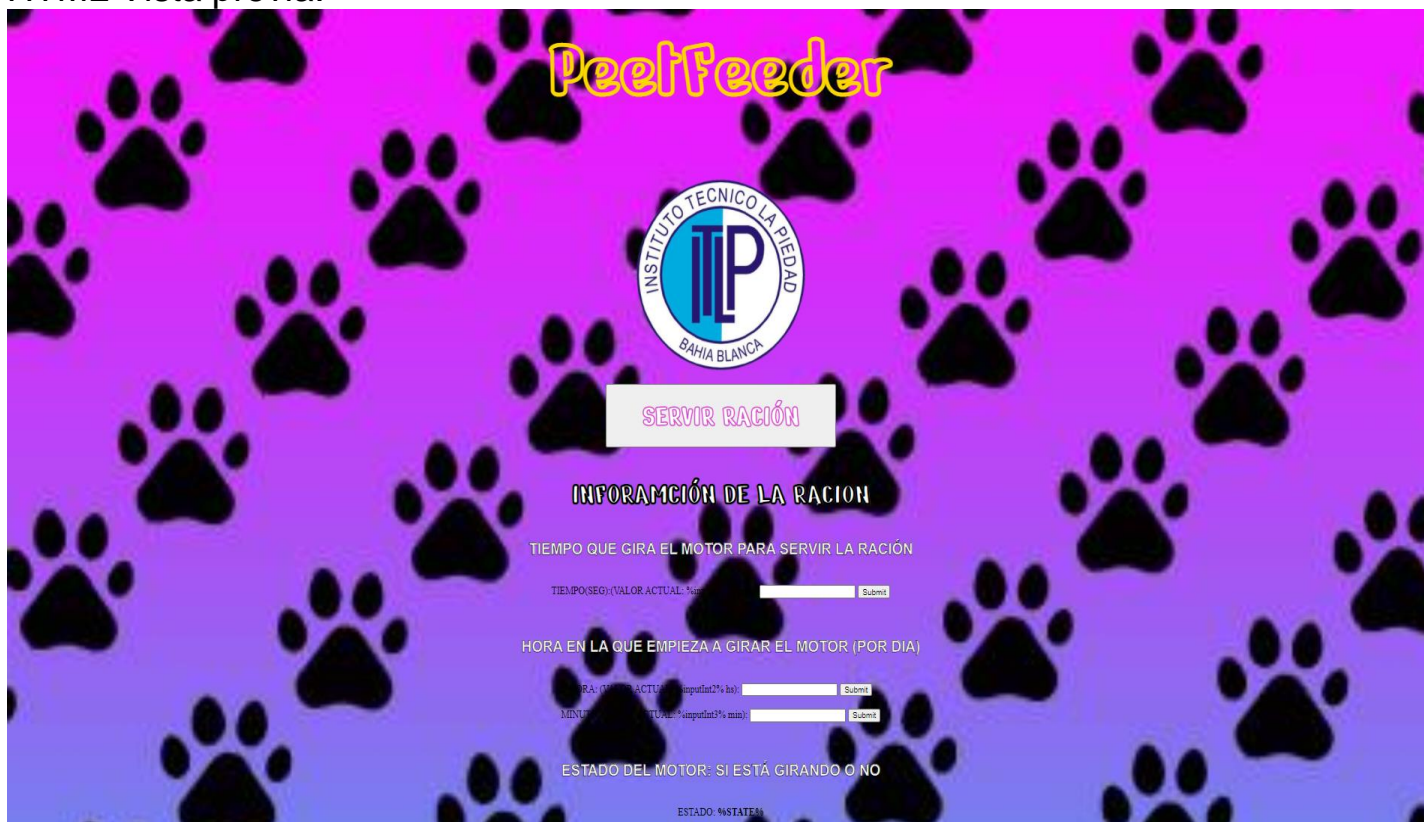
Se incian Librerias y servicios



Espera ordenes y actualiza
constantemente variables

Desarrollo de la página WEB:

HTML Vista previa:



En el desarrollo web nos encontraremos con 4 opciones: por un lado, 3 formularios para enviarle al Esp32 las correspondientes configuraciones o variables que guardara mediante el SPIFFS. Como podemos ver en los formularios se nos permite ingresar la el tiempo que girara la el motor, con esto podemos determinar la cantidad de las raciones. El valor se envía, se almacena por SPIFFS y se envía a la función que le indica al motor que gire ya que esta condiciona su tiempo. También aparecen los formularios del horario en que se depositara la comida, estos también funcionan como el anterior, la diferencia es que en vez de llamar a la función del motor inmediatamente, espera y compara mediante el servidor NTP, el cual nos brinda la hora por un servidor de terceros y la compara con la ingresada constantemente hasta que coincidan, una vez que eso sucede se realiza el llamado.

Por el otro lado tenemos un botón que nos permitirá servir la ración inmediatamente y con las configuraciones guardadas. El funcionamiento consiste básicamente, en que al

momento de pulsarlo se envía una señal al WebServer HTML, el cual la procesa y llama a la función que acciona al motor.

SERVIR RACIÓN

INFORMACIÓN DE LA RACIÓN

TIEMPO QUE GIRA EL MOTOR PARA SERVIR LA RACIÓN

TIEMPO(SEG):(VALOR ACTUAL: 10 segs):

HORA EN LA QUE EMPIEZA A GIRAR EL MOTOR (POR DIA)

HORA: (VALOR ACTUAL: 3 hs):

MINUTO: (VALOR ACTUAL: 45 min):

ESTADO DEL MOTOR: SI ESTÁ GIRANDO O NO

ESTADO:

CSS:

```

1  @import url('https://fonts.googleapis.com/css2?family=Nerko+One&display=swap');
2  @import url('https://fonts.googleapis.com/css2?family=Amatic+SC:wght@700&family=Nerko+One&display=swap');
3  @import url('https://fonts.googleapis.com/css2?family=Amatic+SC:wght@700&family=Nerko+One&display=swap');
4
5  body{
6    background-size: 175%;
7  }
8  h1{
9    font-family: 'Nerko One', cursive;
10   font-size: 135px;
11   -webkit-text-stroke: 2px gold;
12   color: transparent;
13 }
14 h2{color: white;
15   font-family: 'Nerko One', cursive;
16   -webkit-text-stroke: 2px black;
17   font-size: 50px;
18 }
19 h5{color: white;
20   font-family: 'Oswald', sans-serif;
21   font-size: 25px;
22   -webkit-text-stroke: 1px black;
23 }
24 button{color: white;
25   background-image:"comida.jpg";
26   font-family: 'Nerko One', cursive;
27   -webkit-text-stroke: 2px #dd48d6;
28   font-size: 50px;
29   height: 100px;
30   width: 410px;
31 }
32
33 }
```

Componentes electrónicos necesarios:

COMPONENTES	PRECIO
ESP 32 NODEMCU WiFi	\$1400
POLOLU Driver Motor DRV8825	\$480
PROTOBOARD Breadboard 830	\$308
KIT 40 CABLES Protoboard 10cm	\$293