

Alimentador Automático para Mascotas

Leiva Alvarez, Edwin Salvador
ITCA FEPAD E Santa Tecla
Hardware Computacional
121917

Lopez Gonzalez, Marlon Ulises
ITCA FEPAD E Santa Tecla
Hardware Computacional
023917

Trigueros Alvarez, Javier
Edgardo
ITCA FEPAD E Santa Tecla
Hardware Computacional
160417

Abstract—Este documento muestra cada uno de los pasos necesarios para crear un alimentador automático para mascotas utilizando la placa de desarrollo Raspberry Pi 3B+, enfocándose en el funcionamiento de este y también en la importancia que este dispositivo tiene para el cuidado adecuado de las mascotas en El Salvador

Keywords—*component, formatting, style, styling, insert (key words)*

I. INTRODUCCION

La falta de alimentación en las mascotas puede deteriorar la salud y el estado de ánimo de estas en los hogares, generando enfermedades y mal comportamiento en ellos. Estos casos se suscitan muy a menudo en la actualidad, debido a que sus dueños deben dejarlos solos durante el día mientras realizan sus actividades, esto conlleva a un desequilibrio en la cantidad de veces que se da alimento a la mascota por tal motivo se decidió crear un alimentador automático para mascotas para que diera solución a este problema así las personas podrían hacer sus actividades sin preocupación ya que este dispositivo envía una notificación al correo si es que la comida se ha acabado, este dispositivo es controlado por la Raspberry pi 3B+ que tiene todas las funcionalidades necesarias para este proyecto.

II. COMPONENTES

- A. un sensor ultrasonico HC-SR04
- B. una Raspberry pi 3B+
- C. un servomotor MG90s con pines metalicos
- D. cableas para pin header macho-hembra y hembra - hembra
- E. pin header macho – macho resistencias de 330 y 470 ohmios

III. DESCRIPCION DE COMPONENTES

sensor ultrasónico HC- SR04 El HC-SR04 es un sensor de distancias por ultrasonidos capaz de detectar objetos y calcular la distancia a la que se encuentra en un rango de 2 a 450 cm. El sensor funciona por ultrasonidos y contiene toda la electrónica encargada de hacer la medición. Su uso es tan sencillo como enviar el pulso de arranque y medir la anchura del pulso de retorno. De muy pequeño tamaño, el HC-SR04 se destaca por su bajo consumo, gran precisión y bajo precio por lo que está reemplazando a los sensores polaroid en los robots más recientes. Su voltaje de alimentación es de 5 V y su frecuencia de trabajo es de 40Khz.

servomotor MG90s con pines metálicos es un micro servo reforzado que destaca por su gran torque (en relación a su tamaño), engranajes metálicos y gran robustez. Funciona con la mayoría de tarjetas electrónicas con microcontroladores y además con la mayoría de los sistemas de radio control comerciales. Es utilizado principalmente en proyectos de robótica y modelismo de pequeña escala. Los cables en el conector están distribuidos de la siguiente forma: Rojo =

Alimentación (+), Cafe = Alimentación (-) o tierra, Naranja = Señal PWM.

Raspberry pi 3B+ tiene un procesador de 1.4GHz quad-core ARM Cortex A53 de 64-bit,. 1GB de memoria RAM soldada a la placa e incluso los puertos: HDMI de tamaño completo, 4 puertos USB 2.0, MicroSD para memoria interna, conector CSI camera y DSI .

IV. DISEÑO MECANICO

Para el diseño del comedero automático para mascotas se fabrico con un tubo pvc de tres pulgadas con un adaptador para pcv a dos pulgadas el tubo es de aproximadamente 50cm de alto y en la parte inferior se le coloco el servo motor MG90s que es el que gira de izquierda a derecha haciendo que se abra y se cierre el agujero por donde sale la comida, para colocar el sensor ultrasónico HC-SR04 se hicieron perforaciones en el tubo pvc del tamaño de las celdas del sensor ultrasónico esto para que el sensor fuera capaz de verificar por dentro el estado en el que esta el comedero automático para mascotas y si este tenia comida o no.

V. PRIMERAS PRUEBAS

Despues de haber concluido el montaje se procede a realizar las diferentes pruebas, los resultados obtenidos fueron los esperados al menos con el servomotor sin embargo con el sensor ultrasónico se tenia problemas al recibir los datos ya estaba generando mucho ruido y fallaba en el diagnostico del estado del comedero pero esto se solvento usando el sensor en el pvc directamente. Una vez realizadas todas las pruebas se decidio a intentar cumplir cada uno de los alcances y metas planteados

VI. ALCANCES Y METAS

se puede decir que se cumplieron cada una de las metas planteadas con el alimentador automatico para mascotas las metas mencionadas son

- A. Automatización del proceso de alimentación de los perros y gatos de los hogares en El Salvador para mejorar la salud física y mental de las mascotas que se quedan solas todo el día por las diversas actividades realizadas por sus dueños cuando salen de casa.
- B. Construcción del prototipo tomando como fecha límite para terminarlo el 30 de octubre de 2018 tomando en cuenta no sobrepasar el presupuesto estimado del proyecto.
- C. Hacer pruebas del prototipo luego de su construcción en un plazo máximo de 1 semana para poder arreglar posibles fallos y errores del producto

Tambien se cumplieron los alcances establecidos con anterioridad los cuales son.

- A. El presente proyecto investigará la forma adecuada de implementar un comedero de perros y gatos automatizado con materiales y elementos que cualquier persona de El Salvador pueda adquirir con sus propios medios.
- B. El proyecto solo abarcará a la alimentación automatizada a perros y gatos en el entorno doméstico de El Salvador.
- C. La aplicación de comunicación vía Wifi del dispositivo solo será para proveer de alimento de forma remota.

VII. CODIGOS DE IMPLEMENTACION

Codigo para sensor ultrasonico

```
#Libraries
import RPi.GPIO as GPIO
import time
import smtplib
```

```
#GPIO Mode (BOARD / BCM)
GPIO.setmode(GPIO.BCM)
```

```
#set GPIO Pins CAMBIAR PINES!
GPIO_TRIGGER = 18
GPIO_ECHO = 24
```

```
#set GPIO direction (IN / OUT)
GPIO.setup(GPIO_TRIGGER, GPIO.OUT)
GPIO.setup(GPIO_ECHO, GPIO.IN)
```

```
#Email Configuration
smtpUser= 'alimentadorautomatico2018@gmail.com'
smtpPass= 'VigiliA27'
```

```
toAdd= 'jedgardo.trigueros@gmail.com'
fromAdd= smtpUser
```

```
subject= 'AVISO IMPORTANTE'
header= 'To: ' + toAdd + '\n' + 'From: ' + fromAdd + '\n' +
'Subject: ' + subject
body= 'EL ALIMENTADOR DE TU MASCOTA SE HA
QUEDADO SIN COMIDA. Por favor, deposita comida en el
de nuevo, para mantener a tu mascota saludable.'
```

```
x=0
y=0
```

```
def distance():
    # set Trigger to HIGH
    GPIO.output(GPIO_TRIGGER, True)

    # set Trigger after 0.01ms to LOW
    time.sleep(0.00001)
    GPIO.output(GPIO_TRIGGER, False)
```

```

StartTime = time.time()
StopTime = time.time()

# save StartTime
while GPIO.input(GPIO_ECHO) == 0:
    StartTime = time.time()

# save time of arrival
while GPIO.input(GPIO_ECHO) == 1:
    StopTime = time.time()

# time difference between start and arrival
TimeElapsed = StopTime - StartTime
# multiply with the sonic speed (34300 cm/s)
# and divide by 2, because there and back
distance = (TimeElapsed * 34300) / 2

return distance

if __name__ == '__main__':
    try:
        while (x<=2):
            dist = distance()
            #print ("Measured Distance = %.1f cm" % dist)
            time.sleep(1)
            if(dist <= 5.0):

                print("Lleno")
                x+=1
            else:

                if(y==0):
                    #Notification email send
                    print (header + '\n' + body)
                    s = smtplib.SMTP('smtp.gmail.com',587)

                    s.ehlo()
                    s.starttls()
                    s.ehlo()

                    s.login(smtpUser, smtpPass)
                    s.sendmail(fromAdd, toAdd, header + '\n\n' +
body)

                    s.quit()
                    x=3
                # Reset by pressing CTRL + C
        except KeyboardInterrupt:
            print("Measurement stopped by User")
            GPIO.cleanup()

```

Código para servomotor

```

import RPi.GPIO as GPIO
import time

servo_pin = 17

```

```

# Tweak these values to get full range of servo movement
deg_0_pulse = 0.5 #ms
deg_180_pulse = 2.5 #ms

f = 50.0 #50Hz = 20ms between pulses

# Do some calculations on the pulse width parameters
period = 1000 / f # 20ms
k = 100 / period #duty 0..100 over 20ms
deg_0_duty = deg_0_pulse * k
pulse_range = deg_180_pulse - deg_0_pulse
duty_range = pulse_range * k

# Initialize the GPIO pin
GPIO.setmode(GPIO.BCM)
GPIO.setup(servo_pin, GPIO.OUT)
pwm = GPIO.PWM(servo_pin, f)
pwm.start(0)

def set_angle(angle):
    duty = deg_0_duty + (angle / 180.0) * duty_range
    pwm.ChangeDutyCycle(duty)

try:
    #set_angle(28)
    #time.sleep(1)
    set_angle(38)
    time.sleep(0.5)
    set_angle(3)
    time.sleep(1)

finally:
    print("Cleaning up")
    GPIO.cleanup()

```

VIII. DIAGRAMA DE CONEXION

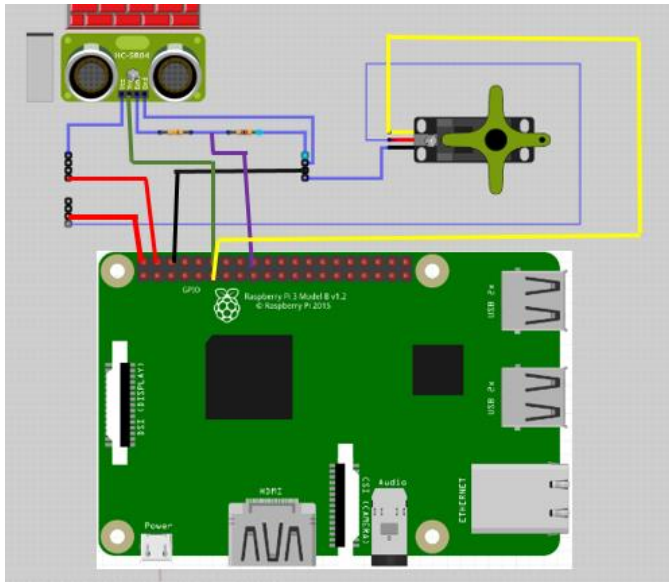


Fig 1. Ejemplo de conexion

Este es el circuito. El pin VCC del sensor irá conectado al Pin 2, GND a GND, Trig irá conectado directamente al Pin 18, y Echo al Pin 24. El servo motor irá conectado a los pines vcc y GND el único pin por el único cable que debemos preocuparnos es el de control (amarillo) que irá conectado al pin 17 de la raspberry pie

IX. RESULTADOS FINALES

Por ultimo hicimos pruebas con el tanque lleno de comida esto con el fin de probar el funcionamiento del comedero automático para mascotas con los resultados finales Podemos decir que fueron exitosos ya que cuando la comida se estaba agotando en el recipiente este enviaba una notificación al correo electrónico de la persona encargada pero si el comedero tenía comida este daba de comer a la mascota a la hora establecida por los desarrolladores una de las cosas que tuvimos que mejorar con la experimentación es como proteger el sensor ultrasónico ya que este mostraba datos erróneos si la comida estaba demasiado cerca, por lo que se decidió poner una .

a.)

REFERENCES

- [1] Comederos automaticos para perros 2018 . <https://planetaperruno.com/comederos-automaticos-para-perros/>
- [2] Escuela politecnica del ejercito, Jaime Velasco trabajo de investigacion R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.
- [3] Zapata, P.A.Z Roles, Diseño y simulacion de un comedor automatico para perros de compañía, 42.