



TECNOLÓGICO DE COSTA RICA

INGENIERÍA EN COMPUTADORES

INTRODUCCIÓN A LOS SISTEMAS EMBEBIDOS

Smart Feeder

Estudiantes:

Malcolm DAVIS

David MONESTEL

Fabian SOLANO

Profesor:

Ing. Jeferson GONZÁLEZ

17 de noviembre de 2018

Índice

1. Descripción del Sistema	2
1.1. Servidor Web	3
1.2. Biblioteca GPIO	3
1.3. Descripción de métodos de la Biblioteca GPIO	4
1.4. Maqueta/Dispositivos	4
1.5. Descripción de las Bibliotecas	4
1.6. Requisitos del Sistema	5
2. Metodología de diseño	6
2.1. Análisis del Problema	6
2.2. Propuestas de Diseño	7
2.3. Comparación de las propuestas	7
3. Herramientas de Ingeniería	9
3.1. RaspberryPi	9
3.2. ATMEGA328	9
3.3. Corte Láser	9
3.4. Yocto	9
3.5. Android Studio	10
3.6. Impresión 3d	10

1. Descripción del Sistema

Este documento contiene el proceso detallado del diseño y desarrollo de un dispensador de alimento de mascotas automatizado que, se logró con el uso de un sistema a la medida implementado en la placa de desarrollo Raspberry Pi 2.

Para proponer el prototipo del sistema desarrollo en proyecto se utilizó la metodología modular, se decidió de este modo para así facilitar el diseño individual de cada uno de los componentes o soluciones de subproblemas con menor complejidad que al final se unen para componer el sistema completo [1]. La subdivisión de estos problemas se hace en subsistemas individuales que cumplen con los criterios de desarrollo y pueden ser utilizados para crear el sistema completo, más adelante se explica cada uno de ellos.

Además se utiliza la arquitectura cliente servidor para poder realizar la comunicación entre el dispositivo móvil del usuario, en este caso un celular inteligente con un sistema operativo Android 5.0+ y el controlador principal de la casa que se encuentra conectado a la red.

Por otra parte, y tomando en cuenta que el sistema operativo que se carga en la placa de desarrollo RaspberryPI es un sistema a la medida, se debe de ejecutar una compilación cruzada desde un computador para poder obtener los binarios que el controlador de la casa va a utilizar. El RaspberryPi funciona tanto como servidor como punto de contacto para los dispositivos que se van a controlar utilizando los pines generales de entrada y salida que la placa provee.

Como se mencionó anteriormente, la Raspberry tiene cargada una imagen de un sistema operativo hecho a la medida utilizando el Proyecto Yocto, con el mismo se logró crear una imagen básica con los componentes del sistema mínimos necesarios para ejecutar la aplicación del servidor. A esta imagen se le agregan varias capas para garantizar el funcionamiento de la aplicación, entre ellos, cabe destacar la capa `.alsa` que es la que permite reproducir audio, esta es la capa más compleja que se agregó, ya que la implementación de esa funcionalidad es más compleja y se aleja del alcance del proyecto. En general se intentó mantener el sistema lo más simple posible para simular el uso de un dispositivo de bajos recursos.

Cuadro 1: Protocolo de Mensajes

Comando	Token	Argumento	Acción
1	000	Credenciales	Login
2	000	Datetime	Agendar Alimentación
3	000	N/A	Alimentar ahora
4	000	Audio.wav	Envía Audio

1.1. Servidor Web

La implementación del servidor se realizó con el lenguaje de programación C con la biblioteca de sockets del sistema. Así como otras estándares del sistema operativo para poder coordinar los mensajes y generar los tokens aleatorios.

El servidor debe poder autenticar usuarios, para esto se definieron usuarios y contraseñas por defecto que se le brindan a los usuarios para acceder a su cuenta. Una vez el usuario envía las credenciales validas, se le asigna un token que se utilizará para validar las instrucciones enviadas por la aplicación, y así evitar posibles accesos no deseados a funcionalidades de sistema.

Se definió un protocolo de mensajes que utiliza tanto el servidor como los clientes que se conecten al sistema así como usuarios y contraseñas por defecto. En el cuadro 1 se puede observar la definición de los mensajes que puede enviar o recibir el servidor junto con su significado, o la acción que invoca.

El servidor se ejecuta sobre la placa de desarrollo Rapsberry Pi V2 B, aprovechando esto, se puede utilizar los pines de propósito general para controlar el actuador que permite el paso de la comida.

Las señales de configuración, y del actuador s reciben en el servidor y se envían a los componentes mediante los pines de propósito general que posee la placa de desarrollo, para lograr esta comunicación se implementó una biblioteca que modifica los registros encargados del funcionamiento de estos pines, esta biblioteca está implementada en C, y se encarga de la definición de los pines utilizados, revisar o modificar el estado y liberarlos en el momento en que se dejan de utilizar para que otra aplicación los pueda usar.

1.2. Biblioteca GPIO

Para poder utilizar la funcionalidad de entrada y salida general (GPIO) del Raspberry Pi, se implementó una biblioteca en C que accede y modifica

a los registros necesarios para interactuar con los pines de entrada y salida para así activar o desactivar dispositivos conectados. Esta modificación de registros en Linux funciona como la escritura o lectura de un archivo [3].

1.3. Descripción de métodos de la Biblioteca GPIO

A continuación se enlistan los metodos implementados en la biblioteca:

- **exportGPIO:** Le dice al sistema que se va a utilizar el pin ingresado.
- **unexportGPIO:** Hace lo contrario del anterior, libera el pin.
- **clean:** Libera todos los pines que estén siendo utilizados mediante la biblioteca.
- **pinMode:** Define el tipo de modo en el cual se utilizará el pin.
- **digitalRead:** Se utiliza para conocer el estado de un pin.
- **digitalWrite:** Se utiliza para modificar el estado de un pin.
- **blink:** Método que alterna el estado de un pin.

Y utilizando un diagrama de distribución de pines para la placa de desarrollo se pueden mapear los pines conectados a los dispositivos con el software (ver figura 1).

1.4. Maqueta/Dispositivos

La maqueta fue elaborada con una combinación de cortes láser sobre una lámina de MDF y cartón. Se utilizó aproximadamente $1m^2$ de cartón para la representación de la estructura. Además, se utilizó un recipiente de cocina para contener el alimento a dispensar y otra más pequeña para contener el alimento dispensado. Para poder hacer estos cortes en la máquina se requiere un diseño bidimensional de los cortes que debe de hacer la máquina.

1.5. Descripción de las Bibliotecas

A continuación se describen las bibliotecas utilizadas:

- **stdio** Biblioteca que provee las operaciones de entrada y salida.

Cuadro 2: Cumplimiento de los Requerimientos del Proyecto

Código	Estado
R0	Cumple
R1	Cumple
R2	Cumple
R3	Cumple
R4	Cumple
R5	Cumple

2. Metodología de diseño

A continuación se encuentra una descripción de la metodología de diseño utilizada para el proyecto, la cual involucra el análisis del problema, la investigación respectiva, y las propuestas de diseño.

La metodología de diseño modular como se mencionó anteriormente, y de igual forma se describió. Consiste en la división en módulos(compilación, ejecución e interfaz con el usuario), dónde el comportamiento del sistema deseado se aprecia al unir los módulos[1].

2.1. Análisis del Problema

El principal objetivo de la computación ubicua es la creación de productos inteligentes conectados que tengan una alta disponibilidad, y que hagan la comunicación más fácil[7]. Estos dispositivos son utilizados para integrar elementos del entorno del ser humano facilitando su quehacer diario.

En los últimos años, el desarrollo de los dispositivos embebidos ha tenido un gran incremento gracias a su bajo consumo de energía y espacio. La tecnología más reciente pretende crear cada vez más dispositivos con más funciones y que utilicen la menor cantidad de recursos energéticos o de espacio. Para lograr esto se debe utilizar menos componentes y/o componentes más modestos.

Este proyecto está enfocado en desarrollar un prototipo funcional de un dispensador de comida para mascotas. Por los puntos descritos anteriormente, y con el objetivo de consumir la menor cantidad de recursos y mantener la simplicidad del sistema, se diseña para contener los componentes mínimos que permitan la creación del mismo.

Para desarrollar el prototipo del sistema desarrollado en proyecto se utilizó la metodología modular, se decidió de este modo para así facilitar el diseño individual de cada uno de los componentes o soluciones de subproblemas con menor complejidad que al final se unen para componer el sistema completo [1]. La subdivisión de estos problemas se hace en subsistemas individuales que cumplen con los criterios de desarrollo y pueden ser utilizados para crear el sistema, en este documento se explica cada uno de ellos.

Otro de los retos propuestos con este proyecto, es que se debe de proveer una interfaz al usuario para poder modificar los valores del sistema, realizando acciones como agendar la dispensación de alimento, enviar audios a la mascota y alimentar inmediatamente.

2.2. Propuestas de Diseño

Seguidamente, se muestran las propuestas de diseño planteadas para el sistema a desarrollar, cada una difiere de la otra según la forma en que describe el comportamiento que el sistema debería tener para cumplir con las funciones requeridas. Se planearon 3 propuestas, todas enfocadas en la forma en que funciona el actuador, ya que las demás funcionalidades del sistema se han estudiado e implementado en otros proyectos del curso y se reutilizaron para facilitar la implementación.

Propuesta 1: Para la primera propuesta, plantea una compuerta sencilla que se abre un tiempo determinado para dejar pasar el alimento.

Propuesta 2: La segunda, en una estructura en forma de cruz extruida que rota sobre un eje dejando pasar una cantidad de alimento por cada cavidad que posee, se puede observar el modelado de esta estructura en la imagen 2.

Propuesta 3: La tercera y última propuesta consiste en utilizar un cilindro con una cavidad que rota sobre un eje y deja pasar a su interior una cantidad de alimento cuando la cavidad está orientada hacia arriba, para luego dejar que se dispense cuando la cavidad está orientada hacia abajo. Se puede observar el modelado de esta estructura en la imagen 3

2.3. Comparación de las propuestas

En esta sección se profundiza en las razones por las cuales se eligió la propuesta número 3, señalando los puntos fuertes y débiles de cada una. La primera propuesta es la más sencilla de implementar de las tres, y esto

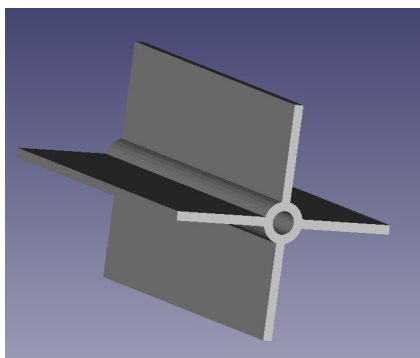


Figura 2: Propuesta 2

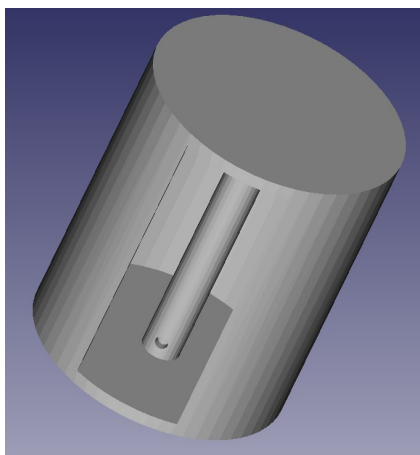


Figura 3: Propuesta 3

le da ventaja, pero se pierde la funcionalidad esperada de poder calcular la cantidad de alimento a dispensar, por esto se descartó esta.

La siguiente propuesta, es un poco más compleja de implementar y cumple con la funcionalidad de calcular la cantidad de alimento a dispensar, pero al hacer pruebas esta propuesta dió problemas al atascarse con la comida y las paredes del prototipo, por lo cual también se descartó.

Por último, la propuesta del cilindro provee un funcionamiento estable y cumple con las características planeadas para el dispositivo, su forma cilíndrica evita que se atasque con las paredes del dispositivo si se interpone el alimento.

3. Herramientas de Ingeniería

3.1. RaspberryPi

Esta herramienta es la base del proyecto, con ella se controla el actuador que permite dispensar la comida y también se reproduce el audio. Además es donde corre el servidor implementado y el sistema operativo hecho a la medida con el Proyecto Yocto.

3.2. ATMEGA328

Este microcontrolador tiene pines PWM que se utilizan para controlar el movimiento de los servos de las puertas.

3.3. Corte Láser

Para la elaboración del modelo físico se utilizó la cortadora láser que el Tecnológico de Costa Rica provee a los estudiantes. Con esta cortadora se elaboraron las divisiones de la casa y luego la misma se armó.

3.4. Yocto

El proyecto Yocto provee una herramienta para la creación de una imagen mínima de GNU/Linux para así poder crear un sistema a la medida. El proyecto Yocto está compuesto de varias herramientas como poky que hacen esto posible.

3.5. Android Studio

Android Studio es un IDE creado por Google para el desarrollo de Android, fue una herramienta de gran ayuda para probar la aplicación.

3.6. Impresión 3d

La pieza que se mueve por el actuador fue diseñada e impresa en una impresora 3d del Tec.

Referencias

- [1] J. Leiva, "*Diseño de Algoritmos*", Lcc.uma.es. [Online]. Available: <http://www.lcc.uma.es/~jlleivao/algoritmos/t2.pdf>. [Accessed: 21- Sep- 2018].
- [2] "*How Much Does It Cost To Install A Home Automation System*", homeadvisor.com. [Online]. Available: <https://www.homeadvisor.com/cost/electrical/install-or-repair-a-home-automation-system/>. [Accessed: 21- Sep- 2018].
- [3] Aguilar, M, 2009. "*Tutorial Linux*". [Online] Available: TEC Digital
- [4] Aguilar, M, 2009. "*Tutorial GCC*". [Online] Available: TEC Digital
- [5] Aguilar, M. 2009. "*Tutorial Make*". [Online] Available: TEC Digital
- [6] González, J. (2018). "*Tutorial Yocto*". [Online] Available: TEC Digital
- [7] "*What is Ubiquitous Computing? - Definition from Techopedia*", Techopedia.com, 2018. [Online]. Available: <https://www.techopedia.com/definition/22702/ubiquitous-computing>. [Accessed: 21 - Sep- 2018].