Robótica y Mecatrónica, Capítulo 4, pp. 37-46.
ISBN: 978-607-9394-14-1.

Prototipo de Asistente Virtual para su Aplicación en Edificios Inteligentes

Martínez Guapo Luis Alberto, Pedraza Ortega Jesús Carlos, Ramos Arreguín Juan Manuel, Vargas Soto José Emilio y Aceves Fernández Marco Antonio

Facultad de Informática ([1]), Facultad de Ingeniería ([2 - 5]).
Universidad Autónoma de Querétaro.

Resumen

Se propone la aplicación de asistentes virtuales en edificios inteligentes, en edificios inmóticos y domóticos, siendo el desarrollo de este prototipo un desarrollo completamente OpenSource. Se implementó un asistente virtual con interacción por comandos de voz en una tarjeta Raspberry Pi 3 haciendo uso de los servicios de voz a texto (STT) y texto a voz(TTS) que proporciona Google Cloud Platform, utilizando lenguaje Python. Se implementó y se incorporó un protocolo UART para comunicación serial entre el asistente virtual (tarjeta Raspberry Pi) y una tarjeta de desarrollo Arduino, para el control de actuadores y el monitoreo de sensores de interés.

Palabras clave: Asistente Virtual, Voz a Texto(STT), Texto a Voz (TTS), Motor Lógico (Logic Engine), Automatización de tareas, Edificios Inteligentes.

1. Introducción

Los asistentes virtuales no son tan comunes, sin embargo, son muy novedosos y muy interesantes por la capacidad y cantidad de desarrollo de aplicaciones que se les puede dar. Los asistentes virtuales que existen en el mercado se encuentran disponibles en su mayoría para computadores, dispositivos móviles y/o usables (wearables), algunos alojados como servicios en la nube (cloud) como lo es el caso de Alexa de Amazon y el dispositivo echo dot, el cual se puede adquirir a un precio de $49.99 USD. El echo dot permite controlar directamente dispositivos creados por Amazon, y algunos otros dispositivos de fabricantes como Philips y TP-LINK, esta alternativa es costosa, pues se basa en la adquisición de dispositivos que interactúan directamente con echo dot, pero brinda la facilidad para controlar dispositivos, pues estos ya tienen una interfaz implementada.

En algunos otros casos, el funcionamiento de los asistentes virtuales está limitado por las empresas que los crearon, pues no hacen nada más fuera de su entorno de ejecución (interacción física).

Convencionalmente los asistentes virtuales no tienen interacción física con dispositivos de hardware, lo cual es un área de oportunidad en cuanto a explotación tecnológica para ampliar las aplicaciones que se le pueden dar a un asistente virtual principalmente en edificios inteligentes [1]. Se busca generar sinergia, innovación y generación de conocimiento al hacer convergencia entre distintas áreas tecnológicas como el software embebido [2], inteligencia artificial, interfaces de hardware, etc...

El objetivo de este proyecto es desarrollar un asistente virtual con reconocimiento de voz implementado en la tarjera Raspberry Pi 3 utilizando librerías OpenSource y mediante interfaces con tarjetas basadas en microcontroladores, se puedan ejecutar tareas de automatización simples indicadas por el usuario.

2. Asistentes virtuales

La llegada de los asistentes virtuales ha tenido un importante evento en la historia de la computación. Los asistentes virtuales son muy útiles ayudando a usuarios de sistemas computacionales automatizando y realizando tareas con la mínima interacción hombre-maquina.

La interacción entre un asistente virtual y un usuario debe ser natural, el usuario se comunica usando la voz y el asistente virtual lo procesa e interpreta y responde de la misma manera [3].

2.1 Arquitectura básica de un asistente virtual

Un asistente virtual convencional consiste en tres componentes principales, voz a texto (STT), motor lógico (Logic Engine) y texto a voz (TTS).

STT es el componente que convierte las instrucciones de voz del usuario a una cadena de texto que puede ser procesada por el Logic Engine. Esto involucra grabar la instrucción de voz dada por el usuario y utilizar un procesador de lenguaje natural para convertir la grabación a una cadena de texto. Como procesador de lenguaje natural se hace uso de la API SpeechRecognition de Google.

Se importa el módulo de reconocimiento de voz, se obtiene una instancia de reconocimiento, se obtiene una referencia del micrófono, se calibra el micrófono, se realiza la grabación de audio y se realiza el reconocimiento de voz en lenguaje español, utilizando el siguiente bloque de código:

```
import speech_recognition as sr
r=sr.Recognizer()

with sr.Microphone() as source:
    r.adjust_for_ambient_noise(source)
    audio=r.listen(source, timeout=None, phrase_time_limit=1)
    text=r.recognize_google(audio, language='es-MX')
```

Logic Engine es el componente que recibe la cadena de texto que devuelve STT para validar la instrucción dada por el usuario y de esta manera proporcionar una respuesta al TTS. Logic Engine puede ser considerado el cerebro del asistente virtual, maneja las consultas del usuario por medio de una serie de cláusulas programadas. El asistente decide que debe darse de como respuesta ante una instrucción específica. En este proyecto este componente es modificado para que valide también instrucciones de tareas de automatización, que es la característica que lo diferenciará de los asistentes virtuales convencionales.

TTS es el componente que recibe la salida o respuesta del Logic Engine y sintetiza la cadena de texto a audio para completar la interacción con el usuario. Es una parte de suma importancia, pues le da al asistente virtual un toque más humano. Para la síntesis de texto a voz se hace uso de la API de Google TTS.

Se importa el módulo de síntesis de voz, se importa el módulo para interactuar con el sistema operativo, se realiza la síntesis de texto a voz en lenguaje español, se guarda el audio recuperado en un archivo con formato mp3 y se reproduce el archivo guardado, utilizando el siguiente bloque de código:

```
from gtts import gTTS
import os

tts=gTTS('cadena de texto', lang='es')
tts.save("audio.mp3")
os.system("mpg321 audio.mp3")
```

2.2 Recursos para el desarrollo del asistente virtual

En recursos de hardware es necesario una Tarjeta Raspberry Pi 3, al menos una tarjeta Arduino, un micrófono compatible con sistemas operativos Linux, teclado, mouse y monitor para conectar la tarjeta Raspberry.

En cuanto software se refiere, se requiere instalar Raspbian en la tarjeta Raspberry, instalar Python en su versión 2.7, el gestor de paquetes pip, instalar PortAudio, Python-dev y también el IDE de Arduino. Por medio de comandos pip instalar las siguientes librerías:

1. PyAudio
2. gTTS
3. SpeechRecognition

Es de suma importancia contar con una conexión a internet bastante veloz, con 10Mbps el tiempo de respuesta es muy bueno.

2.3 Interacción del asistente virtual con dispositivos basados en microcontroladores

La necesidad de que el asistente virtual tenga interacción con dispositivos basados en microcontroladores se debe a que es necesario delegar funcionalidades, pues así el sistema es escalable, mantenible, modular y descentralizado. Se delegan tareas como el monitoreo y control de sensores y actuadores [4] a la tarjeta Arduino, para que la tarjeta Raspberry (el asistente virtual) se encargue solamente del procesamiento de voz, síntesis de voz y toma de decisiones y que de esta manera el desgaste a nivel de hardware sea menor, algo que es muy benéfico al performance (desempeño) del sistema.

Para realizar esta delegación de tareas se hace uso de comunicación serie entre la tarjeta Raspberry y la tarjeta Arduino utilizando un protocolo UART [5] en su modalidad **full-duplex** operando a 9600 baudios.

En Python se hace uso de la librería serial, la cual provee la clase y métodos necesarios para implementar el protocolo UART. Dado que la tarjeta Raspberry no está basada en microcontrolador, no es posible implementar interrupciones para las recepciones de datos por puerto serie, por lo que se implementa un **Thread** (Hilo de ejecución) para mantener un proceso independiente del programa principal, el cual actuará como un ciclo de escucha para la recepción de datos en el puerto serie.

En el caso de la tarjeta Arduino es diferente, sólo es necesario inicializar el puerto serie mínimo, indicando la velocidad de transmisión de datos (en este caso 9600 baudios). La tarjeta Arduino al ser basada en microcontrolador permite el uso de interrupciones, de forma más concreta **eventos en el puerto serie**, que es la forma por la cual se recuperan las instrucciones provenientes del asistente virtual (tarjeta Rapsberry) para su validación y ejecución.

3. Flujos de operación del asistente virtual

A diferencia de los asistentes virtuales convencionales, este asistente tiene la capacidad de interactuar con dispositivos basados en microcontroladores, de forma más general con cualquier dispositivo que tenga una interfaz de comunicación serie, característica que vuelve muy escalable. Incorporada esta característica, se han planteado tres flujos de operación diferentes para este asistente virtual: El flujo de operación de un asistente virtual convencional, el flujo de automatización y el flujo de monitoreo.

3.1 Flujo de operación de un asistente virtual convencional

En este flujo de operación, un usuario pregunta algo al asistente virtual, el asistente valida la petición, realiza toma de decisiones y sintetiza un audio con la respuesta para el usuario, Figura 1.

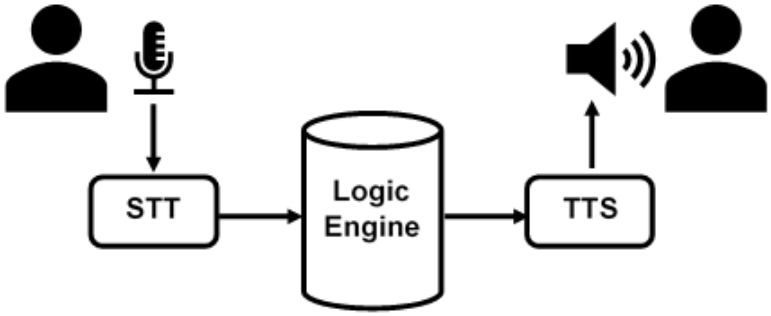

Figura 1. Flujo de operación de un asistente virtual convencional.

En este flujo es posible validar tantas instrucciones como funcionalidades sean implementadas e incorporadas al asistente virtual, tareas desde agenda de tareas con recordatorio por síntesis te voz, consulta de clima, fechas, reproducción de música. Para un ámbito más académico, es posible incorporar al asistente virtual conexión a bases de datos de tipo enciclopedia, para que el mismo asistente redacte la información solicitada al usuario, y con expectativas más altas, que el asistente virtual sea capaz de enseñar a un usuario temas de interés.

3.2 Flujo de automatización del asistente virtual

La implementación de un protocolo de comunicación serie en el asistente virtual, permite que este flujo de operación pueda llevarse a cabo, es en este proceso donde el usuario indicará al asistente virtual ejecutar alguna tarea, por ejemplo: Abrir una puerta o bloquear algún acceso, encender o apagar luces, encender el aire acondicionado e indicar la temperatura que se desea mantener todo por medio de comandos de voz, Figura 2.

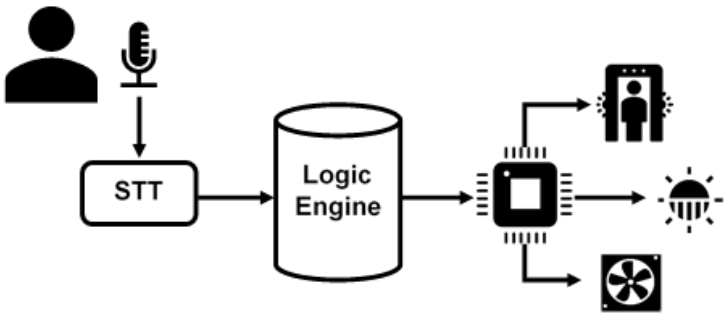

Figura 2. Flujo de automatización.

Incursionando más en temas de seguridad es posible que el asistente virtual sea capaz de controlar cámaras, incorporando también visión por computadora. Es en este flujo donde se puede hacer uso de gran variedad de actuadores, tantos como tareas se quieran o se necesiten automatizar.

Robótica y Mecatrónica, Capítulo 4, pp. 37-46.
ISBN: 978-607-9394-14-1.

3.3 Flujo de monitoreo del asistente virtual

Este flujo se creó pensando en la capacidad del asistente virtual para alertar o reportar al usuario automáticamente sobre el estado de ciertas variables físicas o el posible caso de que ocurran ciertos eventos como que se detecte una fuga de gas, temperaturas demasiado altas en alguna zona del edificio, que una persona ajena al edificio intente ingresar por la fuerza. Este flujo comprende básicamente el monitoreo de sensores, aunque también es posible dotar al asistente virtual con la capacidad de realizar alguna acción específica cuando algún sensor detecte alguna anomalía en el entorno como se muestra en la Figura 3.

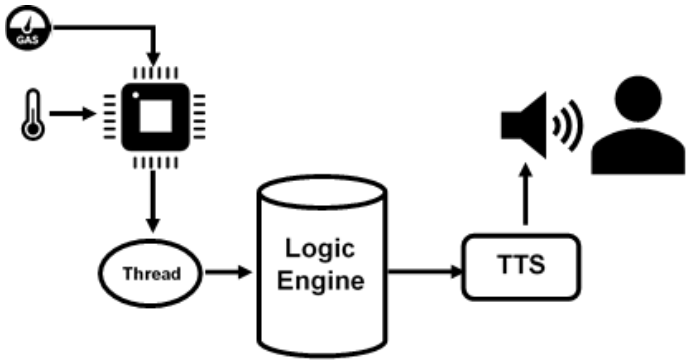

Figura 3. Flujo de monitoreo.

En este flujo de operación, es posible que el asistente virtual se comunique con cualquier tipo de dispositivo que tenga integrada una interfaz de comunicación serie. **El asistente virtual no está limitado a comunicarse sólo con microcontroladores que monitorean sensores.**

3.4 Integración de los flujos de operación

Integrados todos los flujos de operación el asistente virtual, valida instrucciones indicadas por el usuario y/o alertas provenientes de algún dispositivo que se comunique por interfaz serial. La validación de instrucciones no importando de donde provienen, todas se realizan en el componente Logic Engine, este componente esta implementado para realizar una correcta toma de decisiones de acuerdo con las instrucciones que recibe, es aquí donde se valida a que flujo de operación corresponde la instrucción recibida.

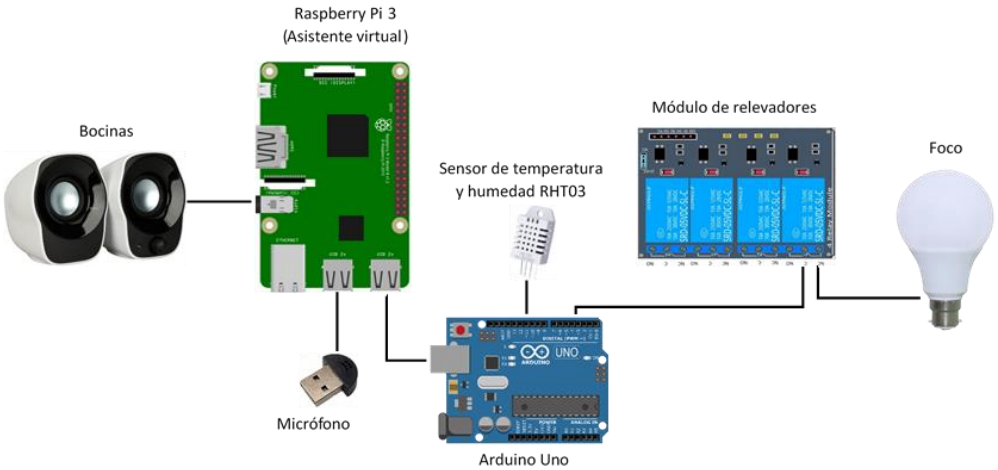

Figura 4. Arquitectura de hardware del prototipo.

La Figura 4 muestra de forma gráfica la arquitectura del prototipo, mostrando en forma general los componentes de hardware que lo conforman.

4. Resultados y discusión

Se realizaron pruebas utilizando algunos sensores digitales y analógicos, relevadores y algunas pruebas simples de consulta. Durante el desarrollo del prototipo y la fase de pruebas se detectaron varias limitaciones las cuales no son críticas, pero el solventarlas hace que el asistente virtual pueda tener un mayor y mejor rendimiento, lo suficiente como para tratarlo como una versión beta.

4.1 Limitaciones

La principal limitación que tiene este prototipo es la dependencia de internet, pues para su funcionamiento es indispensable, además que requiere de una conexión veloz, otro punto que se debe destacar es la importancia de adquirir un micrófono con muy buena calidad y de gran alcance.

El servicio de síntesis de voz tiene un inconveniente, pues mientras más grande es la cadena de texto que se quiere sintetizar a voz más tardará el asistente virtual en dar una respuesta al usuario, dado que el servicio que se consume devuelve un stream de audio, más no se sintetiza el texto letra por letra.

Las limitaciones detectadas pueden ser mitigadas, para remover la dependencia de internet es necesario implementar un procesador de lenguaje natural y un sintetizador de texto a voz, lo cual requiere tiempo de investigación. Para adquirir un micrófono de calidad sólo se requiere contar con el presupuesto necesario.

4.2 Condiciones de pruebas

Las pruebas se realizaron en ambientes de poco ruido. La primera prueba se realizó con un sensor DHT03 (sensor digital, Figura 5) de temperatura y humedad para el flujo de monitoreo. La prueba consistió en lo siguiente: En cuanto la tarjeta Arduino obtuviera una lectura de temperatura o humedad en un rango especificado, se envía al asistente virtual por puerto serie una trama de datos indicando si se trataba de humedad o temperatura, y el valor que se estaba obteniendo del sensor. La trama proveniente de la tarjeta Arduino es recibida por el asistente virtual y este sintetiza un audio con indicando al usuario “Se ha detectado una temperatura de x grados centígrados”, y en el caso de humedad “Se ha detectado una humedad de x %”.

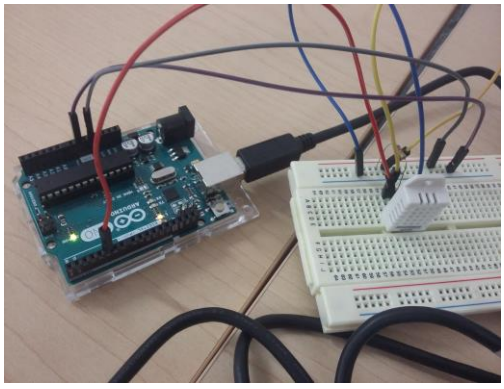

Figura 5. Sensor DHT03 conectado a la tarjeta Arduino Uno.

El asistente virtual recibe la trama con el formato **“xn\n”**, donde **x** indica la variable física que se está monitoreando, **n** es el valor de la variable física y **\n** es el carácter de fin de línea (EOL) necesario

para que la trama pueda ser leída del puerto serie. En la Figura 6, se muestra la trama enviada por la tarjeta Arduino y la cadena de texto que será sintetizada a voz.

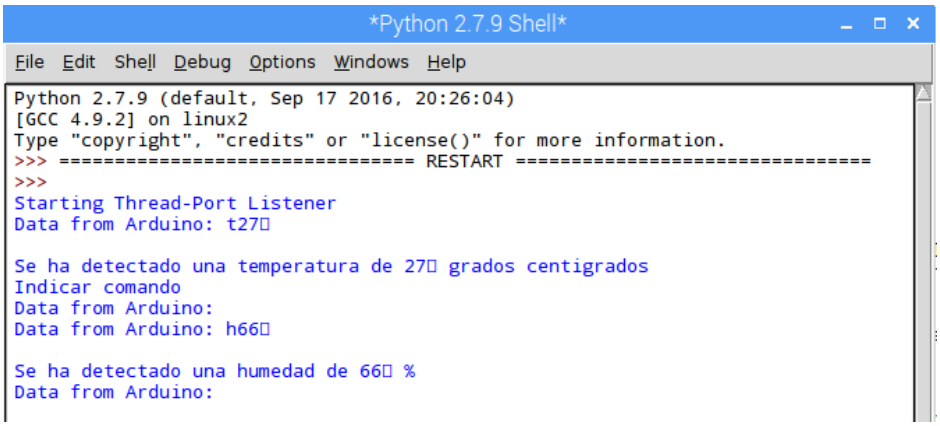

Figura 6. El asistente virtual recibe tramas con valores de temperatura y humedad.

Para la segunda prueba se utilizó un sensor analógico para medir distancia Sharp GP2Y0A41SK0F, la prueba fue de la siguiente manera: Se validaba la aproximación de algún objeto hacia este sensor (Figura 7), en el momento en que la tarjeta Arduino obtiene una lectura que indica que un objeto se encuentra a una cierta distancia, se envía al asistente virtual una trama de datos indicando la distancia a la que se detecto el objeto. Al igual que en la primera prueba, el asistente virtual recibe la trama de datos y sintetiza un audio indicando al usuario “Se detectó un objeto a x cm de distancia” del sensor (colocado en una zona de interés).

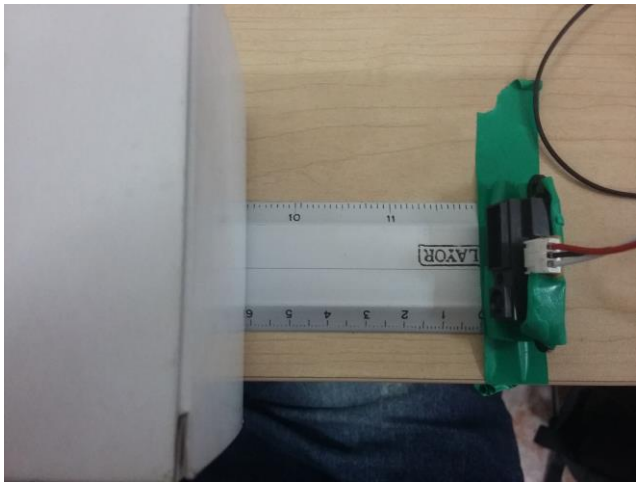

Figura 7. Objeto colocado aproximadamente a 6cm del sensor.

Al igual que en la primera prueba, se recibe una trama con el formato **“xn\n”**, Figura 8.

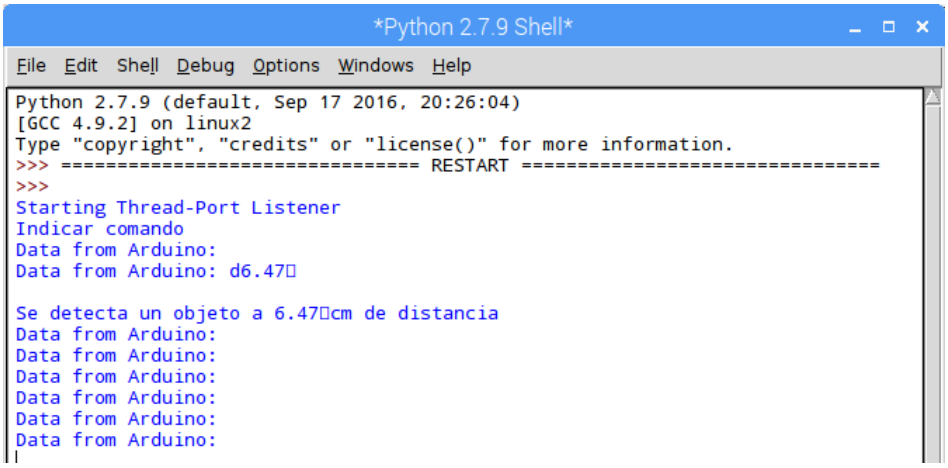

Figura 8. El asistente virtual recibe trama con valor de distancia.

En la tercera prueba, se hace uso de un módulo de 4 relevadores, esto para encender un foco conectado a una toma de corriente alterna. Un usuario dicta la instrucción “luces” y el asistente virtual la valida y envía a la tarjeta Arduino un código para indicarle que active el relevador que permitirá que el foco sea encendido y para apagarlo se dicta también la misma instrucción.

En la Figura 9 se muestra las instrucciones reconocidas por el asistente virtual para encender y apagar el foco.

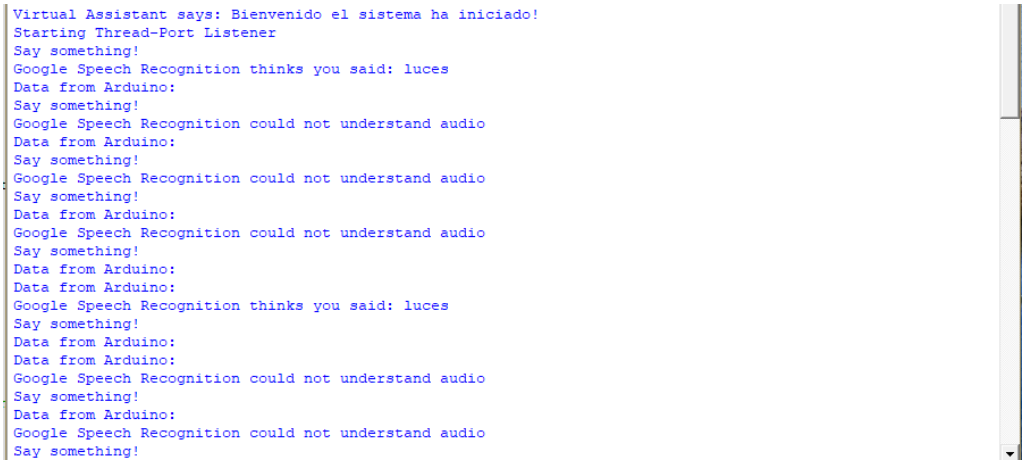

Figura 9. Instrucción luces para encender y apagar foco.

En las Figuras 10 a y b, se muestra el foco encendido y apagado después de validar la instrucción luces.

Como cuarta prueba se hace la consulta del concepto de electrónica para que el asistente proporcione la respuesta, indicando el siguiente comando: **“define concepto_de_interes”**. En la Figura 11 se muestra la instrucción recibida por el asistente virtual y la respuesta que dictará al usuario.

Robótica y Mecatrónica, Capítulo 4, pp. 37-46.
ISBN: 978-607-9394-14-1.

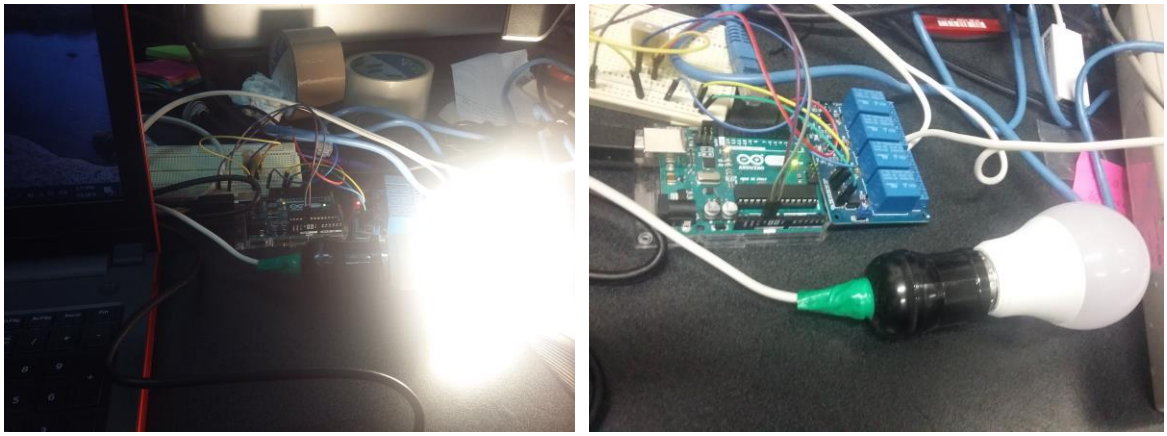

Figura 10. a) Foco encendido al validar instrucción luces, b) foco apagado al validar de nuevo la instrucción luces.

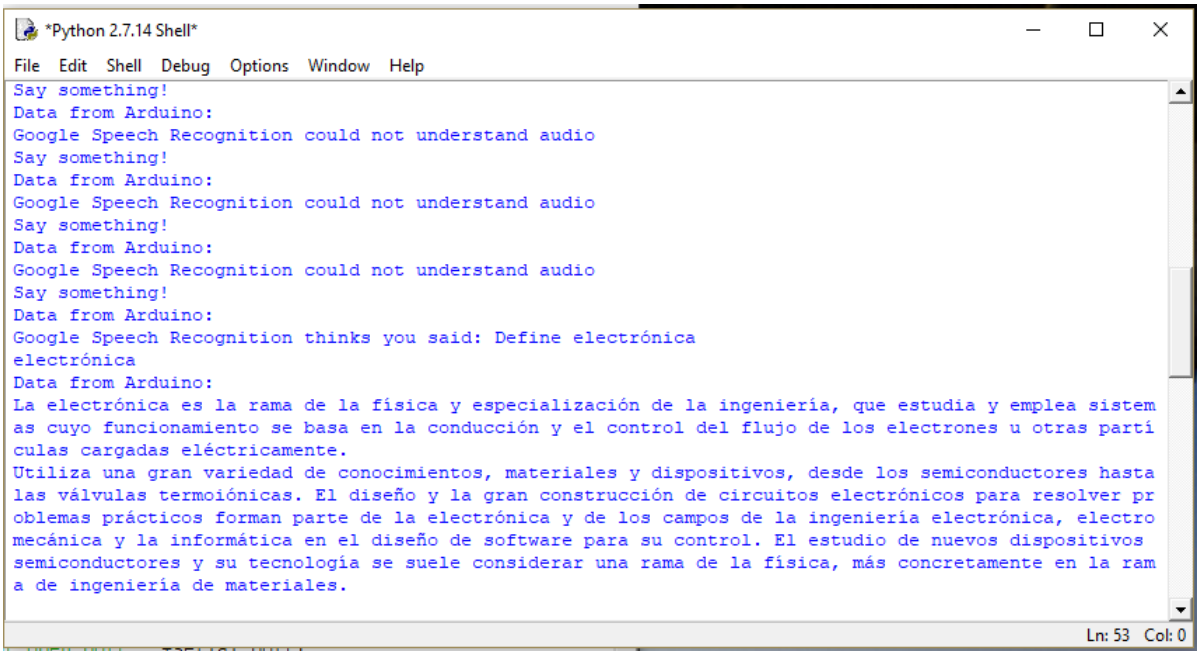

Figura 11. El asistente virtual recibe la instrucción Define electrónica.

La tercera y cuarta prueba se realizaron en el sistema operativo Windows, por la limitación del poco alcance del micrófono en la tarjeta Raspberry.

Las pruebas uno y dos comprenden el flujo de operación de monitoreo, la prueba tres comprende el flujo de automatización y la prueba cuatro comprende el flujo de un asistente convencional.

5. Conclusiones

Estamos en una época en la que hay gran diversidad de tecnologías, que por si solas son sorprendentes, novedosas y muy interesantes, pero ¿Qué pasa cuando convergen dos o más

tecnologías?, el resultado, sinergia tecnológica. Esta sinergia de la que se habla está al alcance de cualquier persona con interés y ganas por crear e innovar. En nuestros días existen muchas tecnologías OpenSource, las cuales podemos tomar, modificar y/o adaptar a nuestras necesidades para crear nuevas cosas. Estas tecnologías aguardan a ser explotadas para mejorar nuestro entorno. La propuesta que se presenta aquí, en un futuro será algo muy común.

Se pretende a futuro, implementar un procesador de lenguaje natural y un sintetizador de texto a voz, para eliminar la dependencia de internet y aumentar los tiempos de respuesta del asistente virtual. Se tiene planeado incorporarle al asistente virtual la capacidad de aprendizaje, esto utilizando técnicas de inteligencia artificial. Se busca además sustituir el protocolo de comunicación serie por un protocolo TCP/IP para montar al asistente virtual en una red.

Los asistentes virtuales pueden ser implementados en cualquier lugar, en casas, oficinas, hospitales, departamentos de policías, escuelas, etc. Entre las aplicaciones de gran aporte que se prevén están: como medio de monitoreo y cuidado para pacientes y/o personas de la tercera edad, también como medio de ayuda en vías públicas como paradas de autobuses y en escuelas como una herramienta de enseñanza.

Referencias

[1] Romero C, Vázquez F & Castro C. *"Domótica e inmótica". Alfaomega, México, 3ra Edición, 2011.*
[2] Aceves M & Ramos J. *"Fundamentos de Sistemas Embebidos".* Asociación Mexicana de Mecatrónica A.C., México, 2012.
[3] Pant T. *"Building a Virtual Assistant for Raspberry Pi"*. Ghaziabad, Uttar Pradesh, India, Apress, 2016.
[4] Corona L, Abarca G & Mares J. *"Sensores y actuadores".* Larousse – Grupo Editorial Patria, México, 2014.
[5] García E. *"Compilador C CCS y simulador PROTEUS para microcontroladores PIC".* Alfaomega, México, Primera edición, 2008.
[6] Mechling L, Gast D & Seid N. *"Using a personal digital assistant to increast independent task completion by students with autism spectrum disorder"*. Journal of autism and developmental disorders, 2009.
[7] Hauswald J, Laurenzano M, Zhang Y, Li C, Rovinski A, Khurana A & Mars J. *"Sirius: An open end-to-end voice and vision personal assistant and it's implications for future warehouse sacale computers"*. In ACM SIGPLAN notices (Vol. 50, No. 4, pp 223-238). ACM, 2015.
[8] Hernández O. *"Edificios inteligentes y sostenibles: arquitectura de percepción y control para la gestion de energía".* Madrid, 2016.
[9] Moumtadi F, Delgado J & Granados F. *"Activación de funciones en edificios inteligentes utilizando comandos de voz desde dispositivos móviles"*. Ingeniería Investigación y Tecnología. (Vol. 15, No 2, pp 175-186), 2014.