UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Introducción al uso de Flite

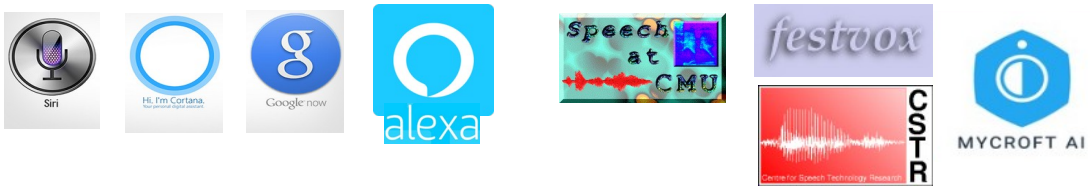
Apellidos, nombre	**Agustí Melchor, Manuel** (magusti@disca.upv.es)
Departamento	**Departamento de Ingeniería de Sistemas y Computadores**
Centro	Escola Tècnica Superior d'Enginyeria Informàtica Universitat Politècnica de València

1 Resumen de las ideas clave

En este artículo vamos a presentar una pequeña librería pasa síntesis de voz que se llama *Flite* y que por su pequeño tamaño está indicada en sistemas empotrados, videoconsolas o móviles que no quieran hacer uso de la red (o no puedan) para obtener el sonido asociado con un texto.

Las novelas y, sobre todo, el cine nos han enseñado ya muchas situaciones en las que la interacción entre el hombre y la máquina se hacen a través de la voz. Seguro que el lector conocerá ya algunos ejemplos como el de: "Una odisea en el espacio" (en la que el computador Hal 9000[1] tiene sesudas conversaciones con los tripulantes de la nave espacial en se que dirigen de camino a Saturno en busca del origen de una señal extraterrestre); o el de Jarvis[2] (el "asistente" de Ironman).

Es un tema en el que existen ya muchas soluciones propietarias y cerradas (Figura 1a) como Bixby, (Samsung), Alexa (Amazon), Google now (Google), Cortana (Microsoft) o Siri (Apple). Dados sus intereses, no es de extrañar que [1] las grandes empresas registren y almacenen de forma rutinaria los datos de voz, que luego se utilicen para segmentar audiencias, perfilar personas y hacerles llegar publicidad. Dada la adopción generalizada de la comprensión del lenguaje natural como interfaz de usuario y las posibles intrusiones de privacidad de las soluciones propietarias, es importante tener soluciones de código abierto (Figura 1b).



a) *b)*

Figura 1: Ejemplos de aplicaciones actuales que usan la voz como medio de comunicación entre el hombre y la máquina: (a) propietarios y (b) de código abierto.

En este artículo nos ocupamos sólo de la generación de voz por parte del computador, esto es la síntesis de voz o *Text To Speech* (TTS), buscando soluciones multiplataforma y de carácter abierto, que nos permitan incorporar este medio de comunicación a nuestras aplicaciones.

2 Objetivos

Una vez que el lector haya leído con detenimiento este documento, será capaz de:

- Instalar las dependencias y compilar el trabajo.

[1] Puede ver más al respecto en <https://es.wikipedia.org/wiki/HAL_9000>.

[2] J.A.R.V.I.S. <https://marvelcinematicuniverse.fandom.com/es/wiki/J.A.R.V.I.S.>.

- Añadir a sus aplicaciones operaciones que hagan uso de la voz para comunicarse con el usuario a través del interfaz de Flite, tanto "en directo", como guardando la salida de audio en un fichero.
- Escoger, de entre las disponibles, la voz que va a "leer" un texto.

Para abordar estos objetivos, en ese documento se está trabajando sobre la distribución de Linux Ubuntu 20.04, pero la explicación es trasladable a otras plataformas. Así como las referencias le proporcionarán caminos para ampliar las experiencias aquí descritas.

Se sugiere al lector que siga los comentarios de este documento comprobándolos en su propio equipo al tiempo que lee estos contenidos.

3 Introducción

Vamos a centrarnos en desarrollar un ejemplo que ponga a vista los principales métodos de la librería escogida. Como ya hemos mencionado, el interés de Flite [2] está en que:

- Es posible utilizarlo en dispositivos embebidos como, p. ej., la *Raspberry Pi*[3].
- Ha sido portado a plataformas móviles como Android[4].
- Ha servido de base de partida a otros motores de síntesis como Mimic[5], el motor de TTS de Mycroft AI.

Flite, está basado en el trabajo de Festival (el motor de TTS del *Centre for Speech Technology Research* de la universisd de Edinburgh), las voces del proyecto FestVox[6] y es un desarrollo del *Speech Group* de la Universidad Carnegie Mellon. En su actual versión, CMU Flite 2.1.0, ofrece soporte para escoger la voz a usar (en tiempo de de ejecución) de entre [3]:

- 18 voces (denominadas ARTIC[7]) inglesas (de origen americano y referenciadas como slt y clb -de mujer. y bdl o rms – de hombre-), canadiense (jmk), escocés (awb) e indú (ksp).
  De las que destacan por su calidad awb y rms, posiblemente[8] basadas en las de Alan Black y Richard Stallman.
- 13 voces del continente indio (denominadas INDIC[9]).

Aproveche para descargarlas en un directorio *voices*, las utilizaremos. La parte que necesita ser mejorada de Flite, como otros proyectos abiertos, es la de la documentación. Por eso aquí queremos colaborar un poco ofreciendo un ejemplo comentado con las

[3] Véase <https://www.raspberrypi.org/>.

[4] Véase *Flite TTS Engine for Android* <https://github.com/happyalu/Flite-TTS-Engine-for-Android>.

[5] Mimic. <https://mycroft-ai.gitbook.io/docs/mycroft-technologies/mimic-overview>.

[6] FestVox es también un proyecto de grupo que desarrolla Flite en CMU <http://festvox.org/>.

[7] Véase <http://festvox.org/cmu_arctic/>.

[8] Según <https://www.raspberrypi.org/forums/viewtopic.php?t=161762>.

[9] Véase <http://festvox.org/cmu_indic/>.

operaciones que habitualmente hemos necesitado de este motor. Para ello hemos utilizado la documentación [4] y también hemos tenido que bucear el propio código [5], porque la lista de opciones que ofrece es mayor que lo que cuenta la documentación.

```
$ sudo apt-get update
$ apt-cache search flite
$ sudo apt-get install libflite1-dev flite
```

Figura 2: Instalación de flite.

En el momento de redactar este documento, en Ubuntu 20.04 está disponible la versión 2.1 de la librería de desarrollo y el ejecutable. Para instalar con rapidez los paquetes necesarios se pueden utilizar las órdenes de la Figura 2: la primera permite actualizar la lista de paquetes, la segunda nos muestra las que tienen algo que ver con "flite" y en la tercera escogemos el paquete de desarrollador y la aplicación ya creada para poder empezar a experimentar.

4 Desarrollo

Si ha seguido los pasos de instalación, tendrá en su equipo instalada la librería junto a los ficheros de cabeceras necesario para incluir en código, sobre lenguaje C, las funciones del API de esta librería.

```
$ flite –version
$ flite --help
$ flite -lv
$ flite_-f contamCoses.txt
$ flite "Hola, Manolo"
$ flite "Hola, Manolo" -o audio.wav
```

Figura 3: Algunas posibilidades de uso de flite.

Además, habrá instalado un ejecutable con el que puede comprobar que se ha instalado correctamente y hacer ya algunas pruebas de síntesis de voz como las sugeridas en la Figura 3. Ahí podrá comprobar la versión instalada, el formato de la orden (y su gran complejidad), cómo puede leer el contenido de un archivo de texto (con "-f"), cómo se le puede pasar una cadena de caracteres para que la lea y cómo poder guardar en un fichero WAVE el resultado de la síntesis. Ya que no se puede incluir aquí el resultado sonoro, si que se puede ver como Audacity lo muestra (véase la Figura 4). No se lo crea y compruébelo: ¡Audacity lo puede reproducir!.

Como lo que nos interesa es el punto de vista del desarrollador, vamos a probar un primer ejemplo que viene con la documentación, denominado *flite_test.c*. La Figura 5 lo muestra y, es importante, cabe destacar que es necesario retocar la línea 3 del original, puesto que da un error en tiempo de ejecución.

Como se puede ver, el trabajo es relativamente sencillo:

- Se declara una voz a usar.

- Se lanzan peticiones de síntesis de voz asociadas a cadenas de caracteres contenidas en el fichero que se pasa como parámetro.

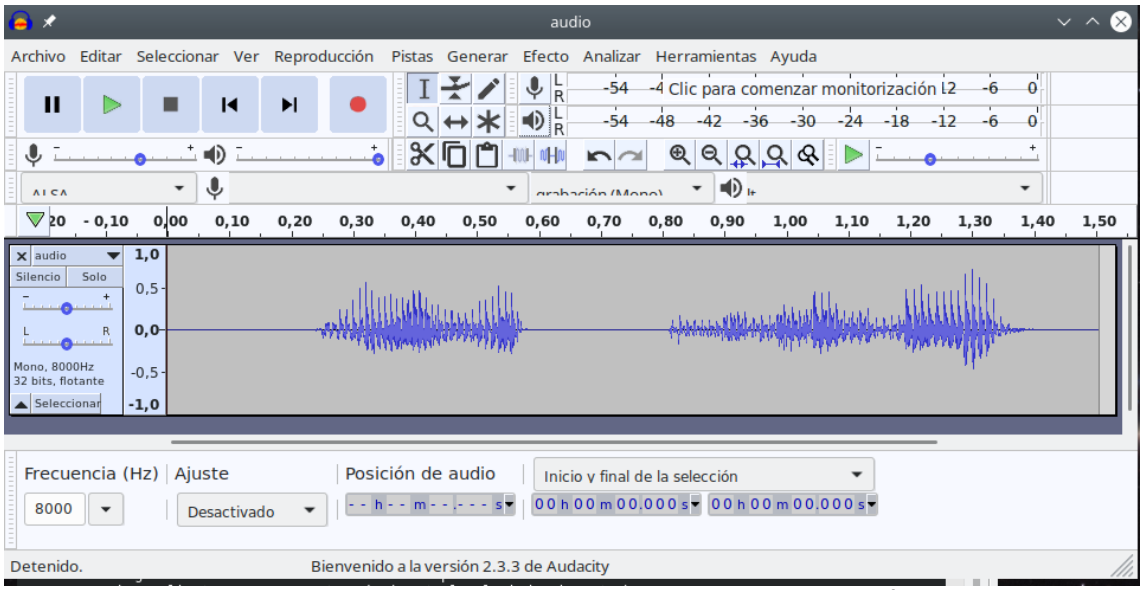


Figura 4: Contenido del fichero audio.wav resultado de la ejecución de la orden flite "Hola, Manolo" -o audio.wav.

```
1.  #include "flite.h"
2.  // Es importante el cambio: register_cmu_us_kal();
3.  cst_voice *register_cmu_us_kal();
4.
5.  int main(int argc, char **argv) {
6.      cst_voice *v;
7.      if (argc != 2)     {
8.          fprintf(stderr,"usage: flite_test FILE\n");
9.          exit(-1);
10.     }
11. flite_init();
12. v = register_cmu_us_kal(NULL);
13. flite_file_to_speech(argv[1],v,"play");
14. }
```

Figura 5: Código fuente de flite_test [4].

Las órdenes para compilarlo y ejecutarlo se muestran en la Figura 6 . Si no ha habido error al compilar, la segunda orden lo ejecuta pasándole un nombre de fichero como argumento (*contamCoses.txt*), de este obtendrá el texto para el que generar el audio. Pruebe a crearlo y, si decide cambiarle el nombre, hágalo en la orden.

```
$  gcc -o flite_test flite_test.c\ -I/usr/include/flite/
-L/usr/lib/x86_64-linux-gnu/ -lflite_cmu_us_kal \
-lflite_usenglish -lflite_cmulex -lflite -lm
$ cat << FI > contamCoses.txt
Hola, mon. From Flite
¿Que me cuentas?
¡Cuenta, cuenta!
2 + 2 = 4
3 / 3 = 1
Què me contes?
0123456789
Carrer del treball, número 10
FI
$ flite_test contamCoses.txt
```

Figura 6: Compilación y ejecución de flite_test.

Es importante insistir en comprobar (escuchar) lo que resulta de los ejemplos escritos en contamCoses.txt. Observe que la pronunciación es inglesa, así que atención a lo que escribe y que no hay una interpretación especial para cantidades numéricas o números de teléfono.

4.1 Ampliando el ejemplo

El ejemplo que se muestra en el Listado 1 denominado *flite_test_ampliado.c* es una versión ampliada del original [4], visto en la Figura 5, para incorporar mayor control y flexibilidad en la síntesis de voz. Hay partes de otros componentes del código fuente y, mayormente, del código de la aplicación filte [5]. Por ello, en este ejemplo se han mantenido los comentarios del código original donde los había y se han añadido algunas ampliaciones.

El código del ejemplo se ha repartido entre dos listados en este documento para facilitar la maquetación y separar dos bloques. Por una parte, el Listado 1 contiene las declaraciones globales. Se pueden destacar las siguientes instrucciones:

- Entre las líneas 4 y 5 se han declarado dos funciones más que se necesitan para declarar las voces.
- Las líneas 6 a la 16 recogen el listado de voces que se podrán utilizar en este ejemplo. En el ejemplo actual solo hemos incluido los nombres de las 18 de tipo ARTIC por brevedad de la exposición; pero, al utilizar las funciones de las líneas 4 y 5, se puede completar la lista con las INDIC [3] y el resto del código seguirá operativo.

```
1.  #include <stdio.h>
2.  #include <flite.h>
3.
4.  void usenglish_init(cst_voice *v);
5.  cst_lexicon *cmulex_init(void);
6.  #define NVEUS 18
7.  static char idVeus[NVEUS][100] = {
8.  "voices/cmu_us_aew.flitevox", "voices/cmu_us_ahw.flitevox",
9.  "voices/cmu_us_aup.flitevox", "voices/cmu_us_awb.flitevox",
10. "voices/cmu_us_axb.flitevox", "voices/cmu_us_bdl.flitevox",
11. "voices/cmu_us_clb.flitevox", "voices/cmu_us_eey.flitevox",
12. "voices/cmu_us_fem.flitevox", "voices/cmu_us_gka.flitevox",
13. "voices/cmu_us_jmk.flitevox", "voices/cmu_us_ksp.flitevox",
14. "voices/cmu_us_ljm.flitevox", "voices/cmu_us_lnh.flitevox",
15. "voices/cmu_us_rms.flitevox", "voices/cmu_us_rxr.flitevox",
16. "voices/cmu_us_slp.flitevox", "voices/cmu_us_slt.flitevox" };
17.
```

Listado 1: Listado de flite_test_ampliado.c (parte 1).

El resto del código se encuentra en el Listado 2 y podemos allí distinguir:

- Entre las líneas 8 y 11 el mensaje que recordará cómo se ha de escribir la orden, esto es, qué argumentos espera esta aplicación. Hemos ampliado el ejemplo original con:
  - La posibilidad de indicar la ruta de una de las voces que hemos guardado en el directorio *voices* para poder escogerla en tiempo de ejecución.
  - Y una cadena de caracteres, para poder indicar directamente un texto a reproducir.
- La líneas 15 a la 24 son la inicialización del motor de *flite*, el idioma y la voz.
- La línea 25 nos da un ejemplo de síntesis de voz a partir de una cadena de texto, con la voz que hemos inicializado.
- Las líneas 27 y 28 muestran que es posible generar el audio y decidir cuándo lo reproduces. Para lo cual la línea 29 es la manera de hacerlo.
- Y, como se ve en la línea 30, si se ha generado así el audio, también es posible exportarlo a un fichero externo.

```
18. int main(int argc, char **argv) {
19.     const char *voicedir = NULL;
20.     cst_voice *v;
21.     cst_wave *w;
22.     cst_utterance *u;
23.     const char *voice_pathname. *text, *outfile;
24.
25.     if (argc != 4)     {
26.    fprintf(stderr,"usage: VOICE.flitevox TEXT WAVEFILE\n");
27.    return 1;
28.     }
29.     voice_pathname = argv[1]; // pathname to .flitevox file
30.     text = argv[2];            // text to be synthesized
31.     outfile = argv[3];        // output file (or "play" or "none"
32.     /* Initialize Flite, and set up language and lexicon */
33.     flite_init();
34.     flite_add_lang("eng",usenglish_init,cmulex_init);
35.     flite_add_lang("usenglish",usenglish_init,cmulex_init);
36.     /* Load and select voice */
37.     v = flite_voice_select(voice_pathname);
38.     if (v == NULL)     {
39.    fprintf(stderr,"failed to load voice: %s\n",voice_pathname);
40.         return 1;
41.     }
42.    flite_text_to_speech("Hola, Manolo!", v, "play");
43.
44.     u = flite_synth_text(text,v);
45.     w = utt_wave(u);
46.     play_wave(w);
47.     cst_wave_save_riff(w,outfile);
48.     delete_utterance(u); /* will delete w too */
49.     return 0;
50. }
```

Listado 2: Listado de flite_test_ampliado.c (segunda parte).

Será necesario, de nuevo, que el lector pruebe lo que está leyendo para convencerse de lo sencillo que es, al menos en número de líneas de código, generar una salida de voz audible en tiempo de ejecución en su propia aplicación.

5 Conclusiones y cierre

El lector, siguiendo el contenido de este documento, habrá podido instalar la aplicación *flite* para comprobar que se puede sintetizar voz y la librería *libflite* para desarrollar un ejemplo en el que probar las funciones básicas del API de síntesis de voz que ofrece este SDK para TTS. También habrá podido comprobar que se puede escoger la voz que utilizará el sintetizador y que cada una tiene su propio acento y, la mayoría, son de una calidad aceptable.

Es el momento de buscar algunas frases y la voz que le resulte más interesante y modificar los ejemplos proporcionados para proponerse que su equipo le comunique algunas cosas de forma sonora ¡¡ÁNIMO!!

6 Bibliografía

[1] K. Reid. (2019). Challenges in open source voice interfaces. URL: <https://opensource.com/article/19/1/open-source-voice-interfaces>.

[2] CMU Flite: a small, fast run time synthesis engine. <http://www.festvox.org/flite/>.

[3] Voces del proyecto FestVox para Flite 2.1. <http://www.festvox.org/flite/packed/flite-2.1/voices/>.

[4] Documentación de Flite <http://www.festvox.org/flite/doc/index.html>.

[5] Github de Flite. <https://github.com/festvox/flite>.