ISBN 978-987-46297-6-0

Desarrollo de dispositivos electronicos para entrenamiento deportivo

Ramiro Romero Dopazo
Laboratorio de Investigación en Nuevas Tecnologías Informáticas
La Plata, Argentina
ramiroromero94@gmail.com

Laura A. Fava
Laboratorio de Investigación en Nuevas Tecnologías Informáticas
La Plata, Argentina
lfava@info.unlp.edu.ar

Diego Vilches Antao, Javier F. Díaz
Laboratorio de Investigación en Nuevas Tecnologías Informáticas
La Plata, Argentina
dvilches, jdiaz@info.unlp.edu.ar

***Resumen*—En este trabajo se describe el proceso de desarrollo de un dispositivo electrónico, basado en sensores y LEDs, que se utilizan para hacer entrenamiento cognitivo de deportistas. El proceso tuvo tres iteraciones que incluyeron diferentes arquitecturas, dispositivos, protocolos y software.**

***Index Terms*—sistemas embebidos, dispositivo electrónico, eda, printed circuit board, diseño de PCB, entrenamiento cognitivo-deportivo.**

I. Introducción

La ciencia está descubriendo que si bien toda la actividad física tiene un efecto positivo en el cerebro, el ejercicio que combina el uso de múltiples sentidos con movimientos corporales completos, desafía al cerebro a niveles más altos, que requieren una función cognitiva más compleja para tomar decisiones y ejecutar habilidades. En consecuencia, la práctica de estas actividades mejora la toma de decisiones y el rendimiento físico, sensorial y neurológico de los deportistas [1].

Conscientes de estas prácticas deportivas emergentes y de la evolución de las tecnologías, en el Laboratorio de Nuevas Tecnologías (LINTI) se comenzó a trabajar, entre otras cosas, en el desarrollo de dispositivos para deporte basados en sensores y LEDs para mejorar el entrenamiento y rendimiento de los deportistas de élite [2]. El dispositivo consiste en un conjunto de luces LEDs que se prenden según cierta configuración de rutinas y se apagan cuando el deportista pasa sobre ellas, lo cual es detectado por un sensor de proximidad. Las rutinas pueden usar grupos de dispositivos para lograr actividades mas atractivas.

En este artículo, se sintetiza el proceso de desarrollo de estos dispositivos que involucró tres iteraciones, cada una arrojando un prototipo evolucionado. Para cada iteración se detalla la arquitectura, junto con el protocolo, software y hardware involucrado para la creación del producto.

II. Desarrollo de los prototipos

El proceso de desarrollo del producto actual, comenzó con una exploración de componentes, luego diseño y desarrollo de un primer sistema (prototipo 2017) que sirvió para realizar pruebas de conceptos. En una segunda iteración se realizaron cambios al diseño de la placa y al software, principalmente para añadir nuevas funcionalidades (prototipo 2018) y por último, en la iteración actual (prototipo 2019), se está realizando un nuevo diseño del sistema, se actualizaron algunos componentes y se está desarrollando una nueva versión del software. Para cualquiera de sus versiones, los prototipos incluyeron uno o varios dispositivos o nodos y un dispositivo controlador, encargado de manejar el estado de cada nodo.

Todas las iteraciones incluyen:

- El protocolo de comunicación nodo-controlador. Varía entre los prototipos.
- El software controlador de nodos. Varía entre los prototipos.
- El firmware del microcontrolador embebido en el hardware del nodo. Se encarga del control de las luces y de la lectura de los sensores en cada nodo, además de implementar el protocolo de comunicación.

III. Fases de desarrollo del dispositivo electrónico

En todos los prototipos se utilizó el SoC[1] ESP8266EX de Espressif como microcontrolador para el hardware del nodo. El chip ESP8266EX combina un microcontrolador Tensillica Xtensa L106, un RISC de 32 bits corriendo a 80 Mhz, con hardware para la funcionalidad Wi-Fi [5].

III-A. Prototipo 2017 (Golfina)

Para este prototipo se diseñó un circuito impreso de capa simple con los siguientes componentes: Una placa *Adafruit Feather HUZZAH* que combina un módulo ESP12E con regulación de tensión, un circuito de cargado de baterías de polímero de litio y un puerto micro-USB por el cual se programa el firmware y se carga la batería. El módulo ESP12E le provee al chip memoria flash y una antena de radiofrecuencia para la comunicación WiFi, un sensor de proximidad IR-08H, un *circuito driver* para alimentar los LEDs y un interruptor *ON-OFF*.

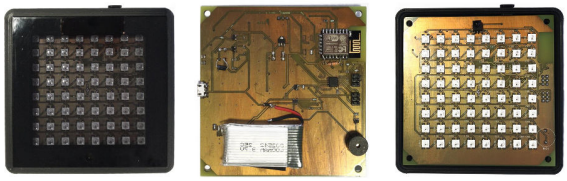
En esta primera iteración se cubrió la funcionalidad básica del sistema: un dispositivo que se conecta a una red Wi-Fi, recibe comandos que «activan» encienden las luces y reportan eventos de proximidad para el posterior cómputo de estadísticas de los tiempos de reacción.

[1]System on a Chip, en un sólo chip que integran el procesador, memoria, almacenamiento, y controladores de entrada/salida.

ISBN 978-987-46297-6-0

(a) Carcasa (b) Cabecera (c) Interior

Figura 1: Prototipo 2017



(a) Carcasa (b) Dorso de PCB (c) Frente de PCB

Figura 2: Prototipo 2018

La fabricación del circuito impreso se realizó mediante el método de transferencia de tóner, dado que se fabricaron únicamente seis unidades, y la simpleza del circuito permitía tener trazas con margen de error relativamente alto. El Firmware fue escrito en C, utilizando el SDK provisto por Espressif, que provee funcionalidad para controlar las conexiones TCP y los paquetes UDP.

Se diseñó e implemementó un protocolo de aplicación binario específico para este sistema. Su unidad de dato es un paquete de longitud fija de 16 bytes. Estos paquetes se encapsulan en datagramas UDP o se transmiten a través de conexiones TCP, dependiendo de su tipo y función. La aplicación móvil de administración de las rutinas incluye el módulo controlador.

III-B. Prototipo 2018 (Prieta)

Para el prototipo 2018, se optó por diseñar una placa impresa doble capa que integre directamente el módulo ESP-12F, en vez de tener el módulo HUZZAH, como es en el caso del prototipo 2017. Esta decisión de diseño permitió tener control sobre aspectos que antes eran manejados por el circuito dentro del módulo HUZZAH. Además, se cambiaron los LEDs que previamente eran controlados como una sola unidad y se los reemplazaron por una matriz de LEDs RGB direccionables.

Se añadió un circuito de control de prendido y apagado del tipo «press ON, hold OFF», con posibilidad de autoapagado, se implementó un circuito para la medición (aproximada) de batería, se hizo uso extenso de componentes de montaje superficial, para mayor manufacturabilidad, se recreó el circuito para el sensado de proximidad (que previamente formaba parte del módulo IR-08H) de forma que la sensibilidad pueda ser ajustable y se utilice un sensor IR con filtro pasabanda centrado en 56 Khz sin control automático de ganancia y se añadió un buzzer para emitir pitidos.

El controlador fue movido de la aplicación Android a una Raspberry Pi 3B+. Se mantuvo la interfaz de usuario en el dispositivo móvil (celular o tablet) pero se delegó el control de los nodos al programa controlador residente en la Raspberry. Se hizo esto a causa de una limitación en Android que hacía que resulte más difícil mantener procesos en segundo plano sin que el sistema operativo lo termine [7]. En el caso de iOS, habría resultado imposible ejecutar las rutinas desde el dispositivo móvil, ya que iOS tiene políticas estrictas de uso de CPU en segundo plano [8].

Para esta iteración se decidió utilizar MQTT [4], con el objeto de usar un protocolo bien establecido en IoT. Sin embargo, los *payloads* de los mensajes continuaron siendo estructuras binarias.

III-C. Prototipo 2019 (Laúd)

Para el prototipo actual, 2019, se conservará el hardware del nodo y se realizarán mejoras en cuanto al software del controlador y de la aplicación móvil. El controlador se desarrolla en lenguaje de programación Go y para la aplicación de administración se utiliza el *framework* React Native (para producir una versión para Android y otra para iOS).

En cuanto al protocolo de comunicación, se está evaluando volver a un protocolo propio, es decir, similar a la primera versión, un binario sobre TCP/UDP modificado, ya que el uso de MQTT, si bien es un estándar para IoT, no ha mejorado la implementación de la comunicación en ningún aspecto.

IV. Resultados

En la primera iteración, se obtuvo como resultado un sistema de entrenamiento enteramente funcional formado por seis prototipos de nodos y una aplicación Android para controlarlos. Este sistema sirvió para hacer pruebas con equipos de fútbol y de hockey de clubes de La Plata. A partir de estos resultados, se comenzó con la segunda iteración, en la cual se obtuvieron dos prototipos con placas de mayor manufacturabilidad y un programa Java que se ejecuta en una Raspberry Pi y controla los nodos.

Se continúa trabajando en la iteración actual en base a la experiencia adquirida en los prototipos anteriores. El objetivo actual es realizar modificaciones mínimas a la placa del prototipo 2018, utilizando una implementación modificada del protocolo de la primera iteración, y agilizar el proceso de fabricación.

Cuadro I: Componentes y protocolos de cada iteración

	2017	**2018**	**2019**
Nodo	PCB capa simple	PCB capa doble	PCB capa doble
Controlador	Biblioteca Java usada en Android	Aplicación Standalone Java en Raspberry Pi	Go
Protocolo	Binario sobre TCP/UDP	Binario MQTT	Binario sobre TCP/UDP modificado
Software usuario	Aplicación Android	Aplicación Android	Aplicación React Native

ISBN 978-987-46297-6-0

REFERENCIAS

[1] Cathi Lamberti, ”Brain Training Enhancing Athletic Performance”, SMARTfit Multisensory Fitness. https://www.ibm.com/developerworks/ssa/library/iot-mqtt-why-good-for-iot/index.html, Inc., Febrero 2018.

[2] Fava, L., Vilches, D., Romero Dapozo, R, and Díaz, J.. ”Tecnología aplicada al deporte de alto rendimiento”. XX Workshop de Investigadores en Ciencias de la Computación, Corrientes, Argentina. ISBN 978-987-3619-27-4. Abril, 2018.

[3] Roy T. Fielding, James Gettys and Jeffrey C. Mogul, Henrik Frystyk Nielsen, Larry Masinter, Paul J. Leach, Tim Berners-Lee, “Hypertext Transfer protocol – HTTP/1.1” Internet Request for Comments, no. 2616, Junio 1999.

[4] Kraijak, S., and Tuwanut, P. “A survey on internet of things architecture, protocols, possible applications, security, privacy, real-world implementation and future trends”. In Communication Technology (ICCT), 2015 IEEE 16th International Conference on (pp. 26-31). Octubre, 2015.

[5] Espressif “ESP8266 Technical Reference”, https://www.espressif.com/sites/default/files/documentation/esp8266-technical_reference_en.pdf. Accedido en abril, 2019.

[6] Espressif “ESP8266 NON-OS SDK Reference ”, https://www.espressif.com/sites/default/files/documentation/2c-esp8266_non_os_sdk_api_reference_en.pdf. Accedido en abril, 2019.

[7] Google Developers. “Services Overview“. https://developer.android.com/guide/components/services. Accedido en abril, 2019.

[8] Apple. “App Programming Guide for iOS“. https://developer.apple.com/library/archive/documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/BackgroundExecution/BackgroundExecution.html. Accedido en abril, 2019.