

# Uso del Internet de las Cosas como estrategia de aprendizaje en asignaturas de Electrónica

M. Perales Esteve, F. Barrero García, S. Toral Marín  
Dept. Ingeniería Electrónica.

Escuela Superior de Ingeniería. Universidad de Sevilla  
mperales@us.es

**Abstract—** Es fácil constatar el creciente desinterés de los alumnos de ingeniería por la electrónica. La novedad que para la generación anterior supuso la electrónica ha pasado y es fácil presumir su dificultad para *engancharse* a las asignaturas de electrónica analógica y digital. Por el contrario, el concepto de *Internet de las Cosas* resulta, por cercanía y novedad, muy atractivo para los alumnos, permitiéndoles controlar circuitos electrónicos básicos desde un navegador web estándar, fijo o móvil, siendo por ello una útil herramienta para reactivar el interés de los alumnos por los circuitos y sistemas electrónicos. En este trabajo se presenta una manera sencilla de introducir el concepto del Internet de las Cosas en una asignatura optativa de electrónica.

**Keywords—** *Electrónica digital, ESP8266, Internet de las Cosas (Internet of Things).*

## I. INTRODUCCIÓN

Indudablemente, el concepto de Internet de las Cosas (Internet of Things, IoT por sus siglas en inglés) [1] es un factor motivante para los alumnos, al ser un concepto relativamente novedoso y que les resulta atractivo por muchas causas, como puede ser la posibilidad de control o monitorización de variables físicas remotamente, desde el móvil o cualquier dispositivo conectado. Su uso en la enseñanza está justificado, sobre todo como factor de enganche de los estudiantes a la materia [2][3].

En este trabajo se muestra la implementación sencilla que se puede realizar del Internet de las cosas o Internet of Things (IoT)[1] en una asignatura optativa (Electrónica de Consumo) que se oferta a estudiantes de distintas titulaciones (Grados de Ingeniería en Tecnologías Industriales, en Tecnologías de las Telecomunicaciones y en Organización Industrial). Dichos alumnos tienen formación previa muy diversa, y es necesario adaptar las prácticas a esta realidad. Es importante, sobre todo, despertar el interés de los alumnos por la electrónica, más que realizar un desarrollo consistente de una aplicación completa. Por tanto, se hará un uso muy instrumental de la conexión inalámbrica y de los protocolos de red, sin entrar a valorar mejores opciones. Se trata, fundamentalmente, de hacer las clásicas prácticas de electrónica con microcontroladores (leer variables analógicas o digitales y actuar sobre periféricos digitales o pines de entrada/ salida), pero usando un microcontrolador que pueda conectarse a internet de manera

sencilla y mostrar los resultados del proceso en una página web. La experiencia se puede extrapolar a otras asignaturas básicas de sistemas digitales, así como servir de puente hacia un desarrollo más electrónico de asignaturas centradas en las redes o en los protocolos de internet.

En el presente trabajo se definirá el contexto docente en el que se desarrollará la experiencia. Tras esto, se plantearán los objetivos de la misma, fijando qué se quiere lograr y dónde poner el acento. Posteriormente se expondrán las diferentes herramientas hardware y software disponibles, se justificará la elección de una concreta para la experiencia, y se mostrarán a continuación las diferentes prácticas que se pueden realizar con la misma. Finalmente, se presentarán algunas conclusiones preliminares de la experiencia, que fue puesta en marcha el curso pasado.

## II. CONTEXTO DOCENTE

La asignatura de Electrónica de Consumo es una optativa de centro, lo que significa que se oferta como optativa a alumnos de varias titulaciones del centro. En concreto, actualmente se oferta a alumnos de los grados de Ingeniería de las Tecnologías Industriales, de Ingeniería en Tecnologías de las Telecomunicaciones, de Ingeniería en Organización Industrial y de Ingeniería Aeroespacial. Se sitúa en el segundo cuatrimestre de cuarto curso, al final de la vida académica de los alumnos. El hecho de tener alumnos de distintas titulaciones de procedencia provoca una gran dispersión en los conocimientos previos y, sobre todo, en el interés que la electrónica despierta en los alumnos. Hay que contar con que un cierto porcentaje de ellos se matriculan de esta asignatura simplemente porque no tiene *exámenes*, sino que se evalúa normalmente a través de la realización de trabajos.

Al ser una asignatura optativa, no está sujeta a cubrir ninguna competencia específica de los alumnos, teniendo bastante libertad a la hora de diseñar el proyecto docente de la misma. En los años que se lleva cursando, de hecho, ha ido cambiando su temario adecuándose al perfil de los alumnos y a los cambios tecnológicos que se han ido produciendo. Actualmente, la asignatura tiene tres bloques diferenciados, para cubrir aspectos diferentes de la materia:

- Electrónica analógica de audio. Se estudian circuitos de audio analógicos de uso en electrónica de consumo (amplificadores, previos, altavoces, sistemas de medida...)

- Instrumentación y normativas de electrónica de consumo. Se estudian circuitos de instrumentación electrónica de uso común y normativas a cumplir para dispositivos de electrónica de consumo.
- Electrónica digital de consumo. Se estudian elementos y sistemas digitales típicos de la electrónica de consumo (pantallas táctiles, sistemas bluetooth, dispositivos IoT, Sistemas embebidos...)

Es en esta tercera parte en la que se ha implementado la experiencia de introducir el concepto de IoT como factor de creación de interés en el alumnado. Hay que tener en cuenta que se trata de un tercio de la asignatura (más bien dos quintos, atendiendo al número de horas de la misma).

Por último, otra variable que se debe tener en cuenta, a la hora de diseñar unas prácticas o una actividad docente para esta asignatura es la variabilidad en el número de matriculados, que ha oscilado entre 10 y 30 en anteriores años. Esto lleva a que resulte difícil hacer acopio de material para la misma, siendo complicado conocer de antemano cuántas placas o cuántos sensores se van a necesitar. Como consecuencia, resulta conveniente diseñar actividades docentes flexibles y, sobre todo, económicas.

### III. OBJETIVOS PERSEGUIDOS CON LA EXPERIENCIA

El principal objetivo, como se ha comentado ya anteriormente, es despertar el interés de los alumnos por los sistemas electrónicos digitales, usando como *cebo* la conexión de los mismos a internet para el control y visualización de los sistemas a través de una página web, por ejemplo. Como resultado, el alumno debe ser capaz de realizar el control de un periférico del sistema microcontrolador (un pin de E/S, un pwm, o cualquier otro periférico) de manera remota desde una interfaz web, y de la misma manera debe poder representar en dicha interfaz algún valor interno del sistema microcontrolador. No se plantean otras cuestiones de vital importancia a la hora del diseño de sistemas de consumo IoT como pueden ser la seguridad de la comunicación, la encriptación de los datos, el control de accesos, contraseñas, etc. Todo esto se obvia, de manera intencionada, para centrar el foco en la implementación electrónica del sistema.

Para poder lograrlo, se necesitará disponer de una serie de elementos que faciliten el desarrollo del mismo:

- Disponer de un sistema microcontrolador de bajo coste, con conexión a internet, preferiblemente inalámbrica, que tenga accesibles entradas analógicas, entradas y salidas digitales y periféricos clásicos (uart, spi, i2c, pwm...)
- Entorno de programación asequible y de fácil instalación, con idea de que los alumnos puedan trabajar de forma autónoma
- Base de conocimientos accesible, con ejemplos de manejo del sistema y con plantillas para la creación de páginas web o manejo de protocolos estándar de comunicaciones.

En función del objetivo principal que se quiere conseguir, y teniendo en cuenta los medios necesarios para ello, se pueden

definir de manera sucinta una serie de objetivos secundarios, que se han ido estableciendo a lo largo del desarrollo:

- Elegir una plataforma hardware con suficiente trayectoria pero que no esté obsoleta.
- Elegir un entorno de programación sencillo de arrancar, y que resulte adecuado a alumnos de cuarto curso de ingenierías.
- Desarrollar un conjunto mínimo de demostradores de las diferentes opciones disponibles, que sirva de base para el desarrollo de los proyectos de los alumnos.

Dado que los alumnos deberán hacer al final un trabajo de tema libre usando los conocimientos adquiridos en la asignatura, y en ésta se van a tratar aspectos y herramientas muy diferentes, una manera de evaluar el éxito de la experiencia será observar cuántos alumnos deciden hacer alguna implementación de dispositivos IoT basándose en lo tratado en la asignatura.

### IV. SELECCIÓN DE LA PLATAFORMA

El primer paso a dar será la elección de la plataforma hardware y software a utilizar en las prácticas de la asignatura. Deberá ser, según se ha dicho, una plataforma barata, fácil de manejar, robusta y con suficiente base de conocimiento accesible. Hay gran cantidad de sistemas que cumplen con esta especificación, por lo que habría que realizar un estudio comparativo y elegir una de entre todas las posibilidades disponibles. Se encuentran en el mercado plataformas basadas en microcontroladores de propósito general, con conexión Ethernet. Igualmente, se encuentran numerosos desarrollos de sistemas basados en microcontroladores específicos de comunicaciones. Por último, también hay disponibles gran cantidad de sistemas que implementan soluciones basadas en sistemas embebidos, con un sistema operativo empotrado en la propia placa (*System on Chip*)

En la Figura 1 se muestran algunas de las placas analizadas. Las dos primeras (a y b) son microcontroladores ARM Cortex M4F a 100MHz, con potencia y memoria suficiente para gestionar las tareas del sistema microcontrolador y a la vez servir una página web, usando el cable Ethernet que llevan incorporado. La primera de ellas, basada en un microcontrolador de Texas Instruments [4], cuenta con la ventaja añadida de que los alumnos que vengan de algunas titulaciones (en concreto del grado de Ingeniería en Tecnologías Industriales) han visto algún microcontrolador similar en asignaturas anteriores, mientras que la segunda, de NXP [5] tiene un tamaño más comedido. Ambas cuentan con el handicap de implementar únicamente una conexión cableada, no pudiendo usarse de manera inalámbrica.

Las dos siguientes, de izquierda a derecha, son las basadas en chips de la compañía Espressif. Ambos son sistemas que integran la pila TCP-IP, la conexión wi-fi y un microcontrolador de bajo coste. El primero de ellos (c) es la placa Feather Huzzah ESP8266 [6], comercializada por Adafruit. Es un sistema con una cierta trayectoria y del que hay disponible bastante información. El segundo (d) está basado en el ESP32 [7], sucesor del anterior y bastante más potente, pero con menos base de desarrollos disponibles aún.

Los dos sistemas representados más a la derecha son dos sistemas empotrados que incorporan un sistema Linux embebido. De las dos, la primera (e) es una placa Omega2+ desarrollada por Onion [8], que dispone de una CPU a 580MHz, 128Mb de memoria DDR2 y 32Mb de memoria Flash para el almacenaje, aparte de disponer de tarjeta microsd. El segundo es la conocida RaspberryPi 3 [9], un pequeño ordenador monotarjeta con Linux (o Windows 10) como sistema operativo, a 1.2GHz de frecuencia de reloj y 1GB de memoria Ram disponible. Ambas tarjetas tienen la posibilidad de conectarse a internet inalámbricamente.

#### A. Plataforma hardware elegida

La plataforma que se eligió para esta experiencia fue la mostrada en tercer lugar, la ESP8266. Los motivos para elegir este sistema frente a los otros son diversos. Se citan a continuación los más significativos:

- Las tarjetas basadas en microcontroladores Cortex están pensadas para una conexión cableada a una red Ethernet, y aunque probablemente sean las más robustas de todas las probadas, resultan menos atractivas al no permitir la conexión desde dispositivos móviles (a no ser que se disponga de un router inalámbrico)
- Las tarjetas basadas en sistemas embebidos con Linux parecen demasiado *pesadas* para tareas sencillas, perdiendo además la capacidad de reacción en tiempo real, fundamental para sistemas de control. Por otra parte, ambas carecen de convertidor analógico/digital, lo que limita mucho el tipo de prácticas que se puedan hacer con ellas.
- De las dos tarjetas basadas en sistemas ESP, se ha elegido la primera por la gran base de desarrollos existentes, que facilita a los alumnos consultar fuentes diversas.

Aparte de los motivos citados, cabe destacar que es la más barata de todas ellas, teniendo en cuenta las posibilidades que

tiene. Algunas de sus características más destacadas son:

Procesador ESP8266 a 80MHz, aunque con posibilidad de ejecutar a 160MHz.

- 4MBytes de memoria Flash
- Regulador de corriente y cargador de batería LiPo
- 9 Pines de E/S con funciones extra (pwm, i2c, spi)
- Una entrada analógica (1V máximo)
- Dos leds en la placa

Como se ha comentado previamente, el sistema dispone de manera nativa de un módulo wi-fi integrado, que puede funcionar tanto como interfaz como de punto de acceso, dando mucha versatilidad a la hora de usarse. Por lo tanto, se dispone de todos los elementos necesarios para realizar prácticas de sistemas electrónicos digitales, pudiendo conectar a la misma numerosos periféricos externos, como sensores i2c, pantallas spi o cualquier otro sistema que use estos estándares de comunicación entre circuitos.

#### B. Sistema de desarrollo

Una vez seleccionada la plataforma hardware, es necesario elegir el sistema de desarrollo a utilizar. De nuevo, la selección no es inmediata dado que se dispone de varios sistemas de desarrollo, cada uno con sus ventajas e inconvenientes.

La primera posibilidad es usar el lenguaje de intérprete de comandos que viene cargado: NodeMCU [10], basado en el lenguaje Lua [11]. Este lenguaje tiene una cierta trayectoria dentro de los sistemas embebidos, pero resulta poco conocido para los alumnos de la Escuela, y no tiene (al menos, de momento) mucha presencia en ámbitos académicos.

La segunda posibilidad que se presenta es cambiar el Firmware e implantarle un intérprete de Arduino [12]. Esta opción cuenta con la ventaja de que es un lenguaje que resulta familiar para algunos de los alumnos de la asignatura, aunque no se use en ninguna materia previa. Esto se debe al rápido crecimiento de la plataforma Arduino y al abaratamiento de las

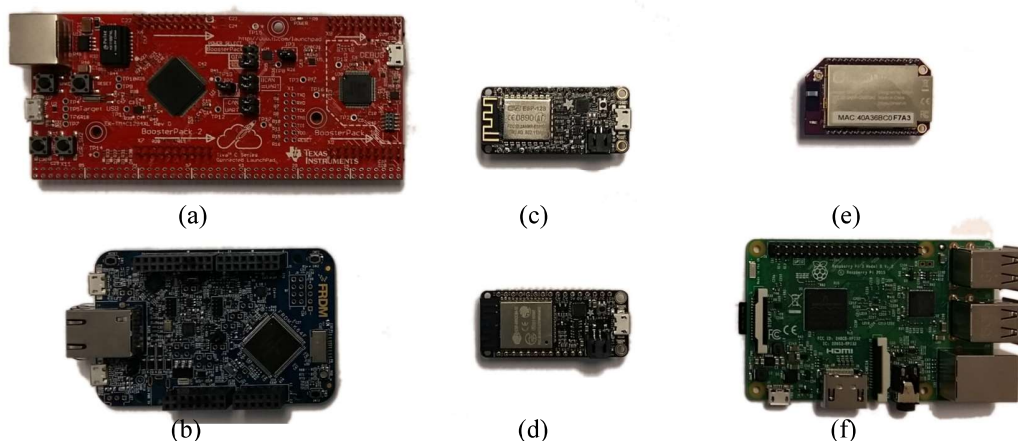


Fig. 1. Tarjetas disponibles para la implementación. (a) TIVA Connected Launchpad; (b) Freedom K64F de NXP, (c) Feather Huzzah ESP8266; (d) Feather Huzzah ESP32; (e) Omega2+, (f) Raspberry Pi 3

placas que ha llevado a una gran proliferación de cursos, presenciales y online, que tratan este entorno de desarrollo.

Por último, otra posibilidad es programar la tarjeta usando un firmware alternativo que implementa un intérprete de MicroPython [13]. Esta posibilidad resulta muy interesante, al ser Python un lenguaje de programación de mucha presencia en el mundo académico y científico, siendo actualmente el primero en el ranking que publica la página web de IEEE Spectrum [14].

Esta será la opción elegida, basándonos más en el interés del lenguaje en sí que en el conocimiento que puedan tener los alumnos de él. Una de las ventajas adicionales del lenguaje Python, que está presente también en MicroPython es la posibilidad de ir probando el código línea a línea o por bloques, antes de generar un código completo. Esto resulta muy práctico cuando no se tiene todavía un dominio del lenguaje.

En cuanto al entorno de desarrollo implementado, se puede usar simplemente un terminal de puerto serie (como TeraTerm o Putty), junto con la aplicación Ampy [15]. Esta era, de hecho, la única opción disponible el curso pasado y fue la que se implementó. El ciclo de trabajo resulta un tanto farragoso, ya que hay que:

- Editar un fichero Python en un editor de texto, por ejemplo el Notepad++ que incorpora ayudas de coloración de sintaxis
- Pasar el fichero con la aplicación Ampy
- Abrir una sesión con algún programa terminal (Putty)
- Ejecutar desde esta ventana, en la línea de comandos de Python, el script subido
- Volver a iterar en caso de fallo.

Desde hace muy poco (octubre de 2017) está disponible el entorno integrado EsPy, desarrollado por Jung Ervin, que se ofrece de manera gratuita [16] y que permite editar código Python con resalte de sintaxis y, simultáneamente, pasar dicho código al microcontrolador, bien de manera permanente o bien ejecutándolo por bloques en la línea de comandos de Python, observando la salida de la consola de Python, como se muestra en la Fig. 2

Fig. 2. Entorno de desarrollo EsPy

## V. DEMOSTRADORES DE LA TECNOLOGÍA

Una vez elegido el sistema electrónico y el entorno de desarrollo para la experiencia, sólo queda cubrir el último de los objetivos previstos, que era contar con una serie mínima de demostradores que sirvan a los alumnos como punto de partida para el desarrollo de proyectos de IoT basados en la plataforma elegida. Para ello, se han desarrollado un par de aplicaciones básicas en MicroPython que hacen uso de las entradas y salidas digitales, así como de la entrada analógica. En la entrada analógica se conectó un divisor resistivo con una LDR, para medir la luminosidad, ajustando los valores para que no llegase a saturar en condiciones de máxima luminosidad.

La primera de ellas es un servidor web mínimo, que crea una página web, accesible desde cualquier sistema conectado a la misma red inalámbrica. En esta página se puede actuar sobre uno de los led de la placa, encendiéndolo o apagándolo a voluntad. Igualmente, se representa en la página web el valor medido por el convertidor analógico-digital, con un botón de refresco para recargar el valor del mismo.

La segunda aplicación desarrollada se basa en un ejemplo suministrado por Adafruit, para hacer uso de su plataforma online de desarrollo de proyectos IoT, llamada Adafruit.io [17]. Esta aplicación hace uso del protocolo MQTT (Message Queue Telemetry Transport) para enviar periódicamente (cada 5s) datos de la luminosidad medida en la placa. Igual que en el caso anterior, se dispone de un botón en la interfaz web que permite encender o apagar el led de la placa.

En la Fig. 3 se muestra el aspecto de ambas aplicaciones, abiertas en un navegador web. Los códigos de las aplicaciones se ofrecen a los alumnos libremente, y pueden ser consultados en el repositorio GitHub de la asignatura [18]

## VI. EVALUACIÓN INICIAL Y CONCLUSIONES

La experiencia se puso en práctica el curso pasado, por lo que sólo se tienen datos de un curso académico. Además, hay que tener en cuenta que, en concreto, el curso pasado hubo una gran cantidad de alumnos que provenían de titulaciones poco relacionadas con la electrónica (como Ingeniería de Organización Industrial).

Lo que resulta innegable es que los alumnos mostraron interés por la experiencia, y fueron todos ellos capaces de crear sus propias aplicaciones web. Sin embargo, a la hora de elegir temática para el trabajo de la asignatura, ninguno de ellos eligió hacerlo sobre esta plataforma. Hubo varios de ellos que usaron la Raspberry Pi, que también había sido utilizada en las prácticas, realizando aplicaciones de IoT, pero basadas en la Raspberry Pi. Esto es explicable por la mayor facilidad que supone programar en el propio dispositivo y tener a su disposición la potencia y los periféricos de un ordenador completo.

Los resultados de la Encuesta de Satisfacción desarrollada por la propia universidad son bastante positivos, obteniendo una puntuación de 4.35 (sobre 5). Resulta significativo que la valoración media de la titulación se queda en el 3.92 mientras que la media del área de conocimiento (tecnología electrónica) baja hasta el 3.86. Estos resultados, aunque referidos a un solo

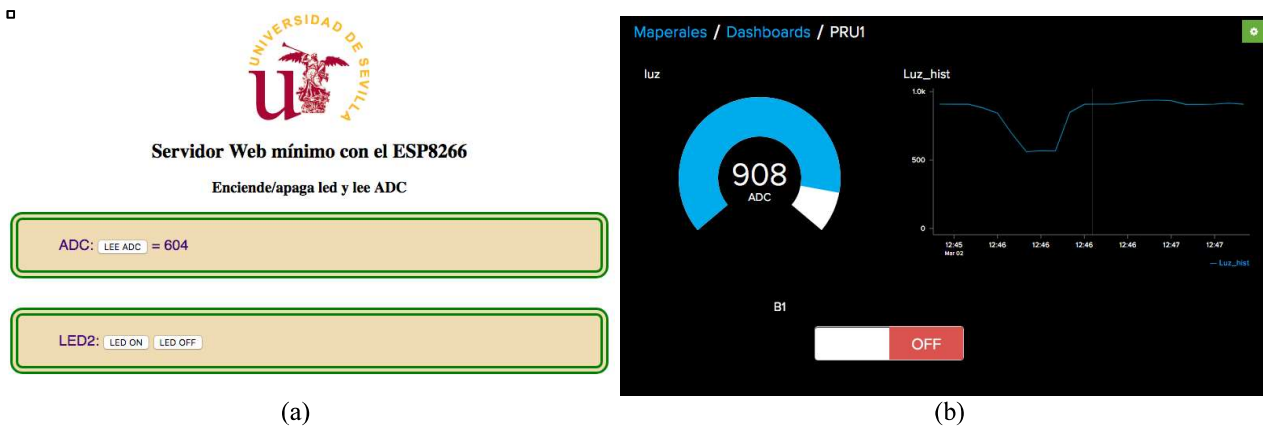


Fig. 3. Visualización de las aplicaciones de ejemplo desarrolladas. (a) página web estática con pulsadores para actuar sobre la placa; (b) página web dinámica creada sobre la plataforma Adafruit.io

curso académico, muestran que la asignatura resulta de interés para los alumnos, y nos animan a avanzar en esta línea.

Como primera conclusión, se deduce de la experiencia que los alumnos ven el Internet de las Cosas como algo interesante, que les motiva, y entienden el potencial de esta herramienta para el desarrollo de proyectos de electrónica de consumo. Sin embargo, parece que la plataforma elegida no ha resultado fácil o intuitiva para el desarrollo de aplicaciones.

Se propone, para el siguiente curso, hacer más hincapié en la descripción de la tarjeta y realizar más ejemplos de uso de la misma. Por otro lado, se estima que al disponer del entorno integrado EsPy los alumnos tendrán menos dificultades para realizar pruebas y alcanzar resultados.

Por último, se dispondrán de equipos suficientes como para poder cederlos en préstamo a los alumnos para que desarrollen el trabajo en modo no presencial, en sus domicilios o la biblioteca, por ejemplo.

#### REFERENCIAS

[1] Kopetz H. (2011) Internet of Things. In: Real-Time Systems. Real-Time Systems Series. Springer, Boston, MA

- [2] J. Gomez , Juan F. Huete, O. Hoyos, L. Perez, D. Grigori "Interaction System Based on Internet of Things as Support for Education" *Procedia Computer Science* Vol. 21, 2013, pp 132-139
- [3] M. G. Domingo and J. A. M. Forner, "Expanding the Learning Environment: Combining Physicality and Virtuality - The Internet of Things for eLearning," 2010 10th IEEE International Conference on Advanced Learning Technologies, Sousse, 2010, pp. 730-731.
- [4] <http://www.ti.com/tool/EK-TM4C1294XL>
- [5] <https://www.nxp.com/products/processors-and-microcontrollers/arm-based-processors-and-mcus/kinetis-cortex-m-mcus/k-series/performance4/k2x-usb/freedom-development-platform-for-kinetis-k64-k63-and-k24-mcus:FRDM-K64F>
- [6] <https://www.adafruit.com/product/2821>
- [7] <https://www.adafruit.com/product/3405>
- [8] <https://onion.io/omega2/>
- [9] <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- [10] <https://nodemcu.readthedocs.io/en/master/>
- [11] <http://www.lua.org/>
- [12] <https://github.com/esp8266/Arduino>
- [13] <http://docs.micropython.org/en/latest/pyboard/>
- [14] <https://spectrum.ieee.org/computing/software/the-2017-top-programming-languages>
- [15] <https://github.com/adafruit/ampy>
- [16] <https://github.com/jungervin/EsPy>
- [17] <https://io.adafruit.com/>
- [18] <https://github.com/maperales/ECONS>