elektron ámbar

Original Article

Revista elektron, Vol. 6, No. 1, pp. 41-45 (2022)
https://doi.org/10.37537/rev.elektron.6.1.142.2022

Análisis de la capacidad de la placa ESP32 para integrar sistemas IoT descentralizados

Analysis of the ESP32 board to integrate decentralized IoT systems

Ciro Edgardo Romero[†1], and Alejandro Elustondo[*2]

[†]*Dpto. de I + D + i, C&S Informatica S.A.*
Buenos Aires, Argentina
[1]cromero@cys.com.ar
[*]*XDK2MAM for IOTA Foundation*
Buenos Aires, Argentina
[2]alejandro.elustondo@ext.iota.org

Recibido: 26/02/22; Aceptado: 04/05/22

***Abstract*—Some years ago, articles and commercial products started using ESP32 as its core. This tendency has helped cyber-physic systems to start evolving into more complex models. Among them, one can quote those used in distributed computing models. Nevertheless, those kinds of systems face the difficulties of software development and also the uncertainty caused by the use of new technologies. In this article one can read the experience of developing a system capable of collecting environmental variables integrated to decentralized network. The objectives proposed in this article are: to develop a proof of concept where one can see the problems that may appear when integrating a project of this nature with a decentralized network; to use different open-source technologies that are enabled for decentralized environments and to analyse the viability for productive developments.**

Keywords: ESP32; IOTA; Internet of things; Blockchain.

***Resumen*— En los últimos años comenzaron a verse artículos y productos comerciales que emplean la placa ESP32 como núcleo. Esta tendencia ha servido para que los sistemas ciber-físicos comiencen a evolucionar a modelos más complejos. Dentro de estos, se hallan los utilizados en el modelo de la computación distribuida. No obstante, este tipo de sistemas poseen problemáticas propias del desarrollo de software, sumadas a la incertidumbre deveniente del uso de las nuevas tecnologías. En este trabajo se detalla la experiencia al desarrollar un sistema capaz de recolectar variables ambientales integrado con una red descentralizada. Los objetivos del mismo son: realizar una prueba de concepto donde se vean las problemáticas a resolver al momento de integrar un proyecto de la naturaleza descripta anteriormente en una red descentralizada; utilizar las diferentes tecnologías abiertas, disponibles para entornos descentralizados, y analizar la viabilidad para desarrollos productivos.**

Palabras clave: ESP32; IOTA; Internet of things; Blockchain.

I. Introducción

Los sistemas ciber-físicos se vieron favorecidos por el avance en el poder de cómputo, la miniaturización electrónica y las interconexiones de redes, en un amplio espectro de nuevas capacidades que antes no eran posibles [1]. Estos sistemas, utilizan componentes informáticos digitales que interactúan directamente con el mundo a su alrededor. En este tipo de desarrollos la arquitectura del sistema define la ruta a través de la cual un dispositivo se conecta a otro; a su vez, esta ruta determina cuán flexible puede ser el sistema, al mismo tiempo que establece cuán receptivo y confiable puede llegar a ser [2]. Desde hace algunos años, apareció un concepto conocido como Internet de las Cosas o IoT (por sus siglas en inglés, *Internet of Things*) que utiliza las tecnologías que se ven involucradas en los sistemas de esta naturaleza. Un sistema IoT involucra elementos electrónicos que interactúan con servicios informáticos. Este sistema puede ser planteado como una arquitectura cliente-servidor con el propósito de que los profesionales especializados en cada elemento, pueden centrar esfuerzos en su campo de expertise. No obstante, la arquitectura centralizada enfrenta varios problemas.

A. Caso de estudio y propuesta de solucion

En el presente articulo, se plantea un sistema mínimo que incursiona en la integración de un dispositivo con conexión a internet (propio de los sistemas IoT) con una red descentralizada. Desde esta experiencia se abordan algunas problemáticas de las antes mencionadas y se proponen soluciones para las mismas. El objetivo principal de realizar este trabajo, es dejar registro sobre el proceso de desarrollo de un sistema de las características citadas anteriormente, como una base para implementaciones similares en diversas aplicaciones.

II. Problemática de seguridad en sistemas ciber-físicos

Según MITRE ATT&CK, una de las problemáticas de los sistemas clientes-servidor es que, todas las operaciones informáticas de cada elemento de la red se llevan a cabo utilizando un único servidor. Esto crea un punto crítico central donde una falla provoca que todo el sistema esté no disponible. Además, la arquitectura centralizada es un objetivo fácil de varios tipos de ataques de seguridad y privacidad, ya que todos los datos de IoT recopilados desde diferentes dispositivos

están bajo la autoridad total de un único servidor [3]. Las debilidades y vulnerabilidades del IoT pueden ser mitigadas utilizando tecnología descentralizada, también conocida como DLT (por sus siglas en ingles, *Distributed Ledger Technology*).

A. Empleo de tecnologías descentralizada

Las tecnologías descentralizadas resuelven las líneas de fallas de seguridad presentes en los entornos públicos sin confianza, donde la mayoría de los dispositivos IoT están conectados entre sí [4]. Dichos entornos, representan las redes existentes a las cuales los dispositivos se conectan para poder tener acceso a internet. La capacidad de mantener la integridad de las transacciones al descentralizar la comunicación entre los nodos participantes en la red, elimina la necesidad de una autoridad central. Además, su naturaleza distribuida *peer-to-peer* [1] puede abordar las deficiencias de los modelos cliente-servidor en las soluciones de la nube. Si bien la convergencia de IoT y DLT puede potenciar las implementaciones sobre estas tecnologías. Su adopción todavía presenta varios problemas, tales como la escalabilidad, algoritmos de consenso, privacidad de datos , eficiencia, disponibilidad, almacenamiento, interoperabilidad, estandarización, entre otros. Además, no hay consenso hacia ningún modelo de referencia o mejores prácticas que especifiquen cómo se deben integrar las tecnologías correspondientes a cada campo [5].

III. Entorno distribuido

Desde un punto de vista teórico, un sistema ciberfísico podría integrar objetos inteligentes con seres humanos de forma estandarizada. El paradigma IoT buscar conectar entre sí a operadores humanos y/o consumidores del sistema, a través de un canal seguro y confiable. Esto implica la implementación de una red distribuida que demuestre confianza en su diseño, al mismo tiempo que pueda ser integrada a otras redes [6]. En este punto, el salto tecnológico necesario para lograrlo es alcanzado por la tecnología blockchain la cual proporciona una red de registro único, consensuado y distribuido [7]. Los gestores que intervienen, obran como agentes de confianza al verificar la identidad y las credenciales de la red. Dicha red utiliza el formato de cadena de bloques, vinculados de manera segura, para que la información pueda ser rastreada [8].

El empleo de la tecnología blockchain posee sus propias problemáticas al momento de su implementación. El costo asociado para cada transacción es un punto de análisis a tener en cuenta, ya que este puede ser elevado según el tráfico que se necesite en la red.

[1]red en la que todos, o algunos, nodos funcionan sin clientes ni servidores fijos; comportándose igualitariamente entre sí. Ver: AUTORES, VARIOS (2015). "10.Peer-To-Peer". Distributed Systems, Concepts and Design (en inglés) (5ª edición). Pearson. pp. 423-461. ISBN 978-01-3214-301-1

A. IOTA como red distribuida

En una red blockchain, cada nodo tiene la necesidad de que se llegue a un consenso, antes de lanzar un nuevo bloque. Si no se mantiene esta sincronicidad, obtenemos bloques "huérfanos", los cuales comprometen el rendimiento general de la red [9]. Esta problemática se ve resuelta con la propuesta innovadora de IOTA Foundation, de utilizar una arquitectura propia conocida como *The Tangle*. La misma está basada en un concepto matemático llamado Grafo Acíclico Dirigido (DAG). Dicha arquitectura es un libro mayor distribuido abierto, gratuito y escalable, diseñado para admitir la transferencia de datos y valor sin problemas. En esta estructura, los bloques "huérfanos" se tienen presentes por ser una consecuencia inevitable de una alta tasa de transacciones. A diferencia de los sistemas blockchain, estos se fusionan nuevamente en el sistema, sin generar desperdicios [10]. Todas las transacciones en blockchain deben esperar hasta que se incluyan en un bloque. Debido a las limitaciones en el tamaño del bloque y al tiempo de producción de este, se crea congestión y tiempos de espera para las transacciones. En el caso de *The Tangle* esto no ocurre, ya que al ser un grafo acíclico dirigido, cada transacción se adjunta a dos transacciones anteriores, es por esto que el protocolo puede procesar varias cantidades de transacciones en paralelo. En la figura 1 se muestra la comparativa del cuello de botella del tráfico de bloques producido por una arquitectura blockchain y por la arquitectura Tangle

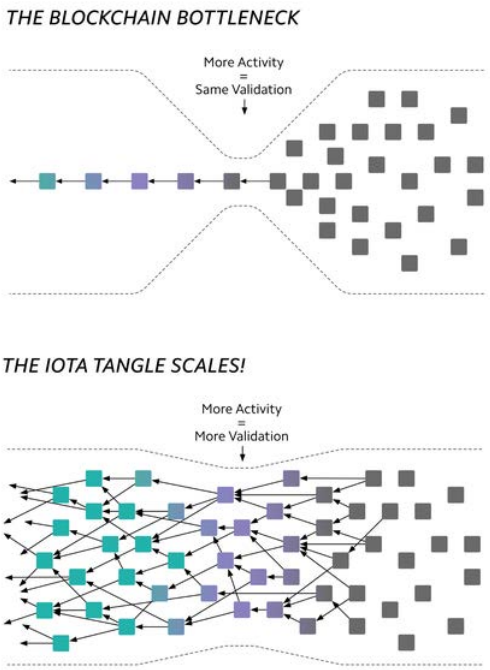

Fig. 1. Cuello de botella de Blockchain vs Tangle [2].

[2]Imagen tomada de https://commons.wikimedia.org/wiki/File:Blockchain_vs_tangle_bottleneck.png

En esta red no hay bloques y no existen los mineros. Cuando una transacción es enviada, valida simultáneamente otras dos transacciones. Esto permite que IOTA supere las limitaciones de costo y escalabilidad de blockchain sumado a que garantiza que la información sea confiable y no pueda ser manipulada ni destruida. La *IOTA Foundation*, describe su red como una tecnología de registro de datos distribuida y de código abierto que permite de forma segura el intercambio de información y valor, dado que soporta micro-transacciones gratuitas con bajos recursos de hardware [11].

IV. Adquisición de datos de forma distribuida

El esquema de un sistema ciber-físico, de arquitectura distribuida, supone una interacción de diferentes elementos comunicándose entre sí de manera independiente [12]. Basado en este concepto, se puede interpretar que existirán dispositivos que realicen mediciones y las envíen a una base de consulta colectiva. Por otro lado, habrá dispositivos que funcionarán como actuadores, y que realizarán acciones sobre el sistema basados en la mediciones anteriores. En la figura 2 se muestra un esquema de posibles elementos que formarían parte de un sistema distribuido.

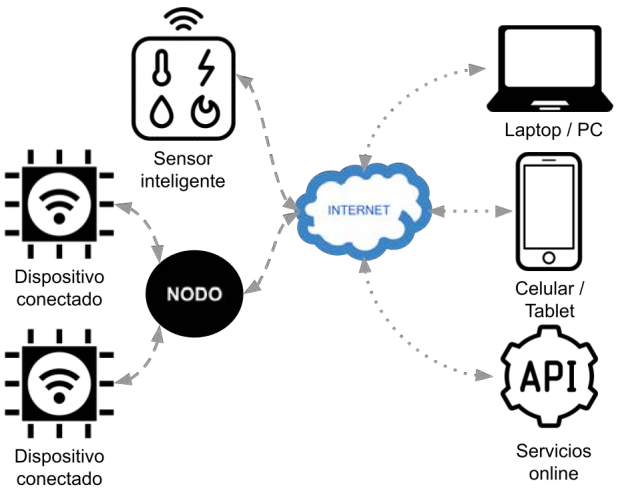

Fig. 2. Ejemplo de sistema distribuido

A. Sistemas IoT con ESP32

En nuestra experiencia, la placa ESP32 es una buena alternativa para el desarrollo y mantenimiento de sistemas. La misma es autoría de la empresa Espressif Systems, y representa una serie de microcontroladores de bajo costo. Se adecúa a diversos tipos de implementaciones por su bajo consumo de energía, sus múltiples entornos de código abierto y sus bibliotecas [13]. Al mismo tiempo, posee una extensa documentación y varias comunidades activas con ejemplos para programarla en lenguaje C/C++ y Python, entre otros.

V. Desarrollo de prueba de concepto

Al plantear un entorno distribuido se pueden presentar diversos elementos que conforman la totalidad del sistema de una manera compleja. Desde este punto, es interesante realizar una prueba de concepto simplificando la interacción que existirá en el sistema utilizando dispositivos que tendrán un comportamiento elemental. El entorno distribuido puede ser descripto como un sistema IOT, desde su entendimiento más sencillo. Sobre esta idea, se tienen que abordar las problemáticas de:

- Complejidad del sistema general
- Capacidades limitadas en los dispositivos
- Diversidad de tecnologías coexistiendo
- Seguridad informática
- Costos
- Falta de rigurosidad en el tratamiento de datos
- Decisiones que comprometen la eficiencia

Desde el Departamento de Investigación, Desarrollo e Innovación de la empresa C & S Informática S.A, se realizó una prueba de concepto como experiencia para abordar algunas de las problemáticas anteriores [14]. Sobre estas pruebas se logra la comunicación entre un nodo sensor con una red distribuida. Los datos recolectados por un nodo basado en la placa ESP32, son enviados a la red IOTA (elegida como red distribuida), con la intención de simular la comunicación típica de un sensor inteligente dentro de un sistema no centralizado. Al mismo tiempo, se intenta desarrollar un programa fácil de mantener desde la perspectiva del dispositivo, a través de un lenguaje de programación de fácil lectura.

A. *Código* bare-metal

Según Tollervey, autor del libro *Programming with MicroPython*, Python es un lenguaje de programación de código interpretado fácil de aprender y legible [15]. Este lenguaje multiparadigma es capaz de soportar programación imperativa, programación funcional y, parcialmente, orientada a objetos. Además, es un lenguaje dinámico y compatible con diversas plataformas. Basado en el mismo lenguaje existe Micropython, que es una implementación sencilla y eficiente con todas las funciones del Python 3, salvo algunas excepciones. Incluye un subconjunto pequeño de la biblioteca estándar, reimplementada y optimizada para ejecutarse en microcontroladores. Micropython se ejecuta de tal forma que las operaciones y servicios sean manejados como sistema operativo [16].

B. Nodo sensor basado en ESP32

El trabajo descrito en este articulo utilizó un dispositivo que funciona como un nodo recolectar de variables ambientales. El nodo cuenta con un módulo BMP180 [17], basado en el sensor homónimo, para medir presión, temperatura y humedad de la placa de desarrollo. Dicha placa, ejecuta la configuración propia de sus componentes y los servicios desarrollados para cada propósito. También se conecta un LED tipo RGB y se configuran diferentes colores, asociados al estado de la placa. De esta manera se puede chequear la situación del dispositivo una vez desconectado de la PC.

En la figura 3 se muestran los componentes del nodo sensor basado en la ESP32

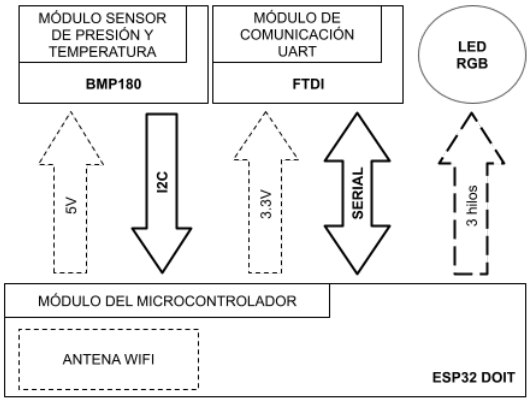

Fig. 3. Elementos que componen el nodo sensor

El dispositivo se conecta a internet y realiza una configuración del reloj interno, realizado una petición a un servidor público NTP [3]. El comportamiento general comienza cuando se comunica con el sensor para realizar la medición. Los datos de esta, se guardan en un objeto dentro del programa ejecutado en la ESP32. A este objeto, se le suman el UUID (identificador único de dispositivo), hora y fecha de la medición (en formato *UNIX timestamp*). Antes de enviar la información, esta debe comprimirse a través de una función criptográfica hash; o simplemente conocida como *hash*. Este es un algoritmo que transforma cualquier bloque de datos en una nueva serie de caracteres con una longitud fija. Independientemente de la longitud de los datos de entrada, el valor de salida tendrá siempre la misma longitud [18]. Por ultimo, se realiza el envío de los datos en formato JSON a través de una *request*. Mediante de esta prueba de concepto, se recrea el comportamiento básico de un sistema de recolección y distribución de datos. Al tener un sensor que funcione como nodo conectado a internet, se logra convertir una variable física en una variable utilizable por un sistema informático. Posteriormente, este dato es enviado a la red distribuida IOTA para que pueda ser consumido por otros sistemas.

C. Red distribuida

Se buscó generar una red de topología colectiva, donde los nodos se vinculen unos a otros de modo que ninguno de ellos tenga poder de filtro sobre la información que se transmite. En una primera versión, para comunicarse con *The Tangle* se utilizó un nodo IRI [4]. Esta versión tuvo que ser descartada por quedar obsoleta. Siguiendo las propias recomendaciones de la *IOTA Foundation*, se migró el desarrollo a IOTA 1.5, el cual representa una evolución del mismo protocolo. Este nuevo protocolo brinda especificaciones especiales que dicta cómo deben comportarse los nodos de IOTA. Desde este concepto se desprende Hornet, una implementación escrita en Go, que proporciona capacidades de nodo e incluye el soporte completo de la actualización de la red [19]. Los nodos se presentan como dos interfaces para desarrollar e integrar las aplicaciones. Estas funcionan como APIs de bajo nivel permitiendo la comunicación entre dos aplicaciones

- API REST: permite a los clientes interactuar con *The Tangle* y comunicarse con los nodos. De esta manera puede comunicarse con la red a través de mensajes (enviar y recibir) y realizar pruebas de trabajo.
- Eventos de la API: permite a los clientes sondear nodos en busca de nuevos mensajes y otros eventos. Este tipo de API es útil para crear aplicaciones como billeteras digitales (conocidas como *wallets*) que necesitan monitorear *The Tangle* para actualizaciones de los saldos de ciertas direcciones.

VI. Comunicación en la red

Cada nodo se puede identificar de forma individual mediante un identificador único (del inglés *peer identity*), también llamado PeerId. Las conexiones con un nodo IOTA, se establecen a través de un protocolo conocido como *peer discovery* que se utiliza para exponer una interfaz que proporcione una lista de nodos verificados. La implementación del protocolo proporciona una lista codificada y confiable, administrada por la *IOTA Foundation* y sus colaborares [20] . Estos protocolos se identifican de manera única, y autentican los paquetes enviados, utilizando criptografía basada en clave pública, o PKC (por sus siglas en inglés, *public Key-based cryptography*). En esta versión del proyecto, se utilizó un nodo público provisto por la misma comunidad de IOTA [5]. Los dispositivos se comunicaban con el nodo, enviando *requests* a la ruta /api/v1/messages, con la información recolectada a través de sus sensores. Esta era enviada en formato JSON con la siguiente estructura de ejemplo:

```
{
  "networkId": "9466822412763346725",
  "parentMessageIds": [
    "222e88a63e5aca8ef4(...)",
    "a22137ebe61435c6d0(...)",
    "a6db9d0b3ecb274d90(...)",
    "fd31d9c926b5d97ae0(...)"
  ],
  "payload": {
    "type": 2,
    "index": "68656c6c6f20776f726c64",
    "data": "5370616d66(...)"
  },
  "nonce": "2102864"
}
```

Al enviar el mensaje, el nodo intentará autocompletar los campos networkId, parentMessageIds y nonce si

[3]Protocolo de tiempo de red , utilizado para sincronizar los relojes de los dispositivos con alguna referencia de tiempo. Ver: http://www.ntp.org/ntpfaq/NTP-s-def.htm

[4]Ver: https://github.com/iotaledger/iri

[5]Ver lista de nodos. Link: https://thetangle.org/nodes

estos faltasen. Si la comunicación es exitosa, el mensaje se almacenará en *The Tangle*. Este punto final devolverá el identificador del mensaje creado. Si faltara *payload*, el mensaje se generará igualmente pero vacío. Cuando la transacción es exitosa, se devuelve un *hash* de referencia que representa el identificador de la transacción. Este será utilizado por otros elementos dentro del sistema para localizar y decodificar la información insertada.

En la figura 4 se muestra un diagrama para ilustrar el esquema de comunicación.

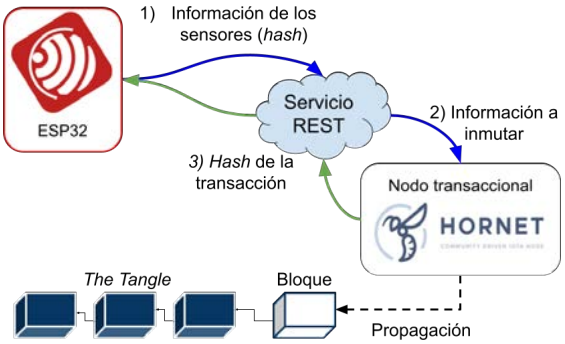

Fig. 4. Comunicación entre elementos del sistema

VII. Conclusiones

La curva de aprendizaje de Micropython fue un factor clave para el desarrollo de la prueba de concepto. Sumado a esto, la practicidad para adquirir la placa ESP32 en el mercado, vuelve a este conjunto de opciones una buena alternativa para el desarrollo de diversos sistemas embebidos.

En la implementación de proyectos críticos, integrar servicios para la transferencia y encriptación de datos provee una capa de seguridad interesante. Adicionalmente, los sistemas ciber-físicos pueden ver mejorada la confianza en sus agentes intervinientes. Por otro lado, la red IOTA reduce significativamente los costos asociados a la implementación de una red. Todo esto favorece a los usuarios de placas como la ESP32 y otras de similares prestaciones.

El código para comunicarse con la red descentralizada puede ser integrado con un sistema de medición inteligente de cualquier naturaleza. Sus métodos pueden ser perfectamente migrados a otros lenguajes propios de otros microprocesadores. Es decir, podría descentralizarse un sistema productivo, ya instalado, realizando pequeñas incorporaciones de hardware y software.

El resultado final del trabajo realizado es un sistema mínimo perfectamente funcional, capaz de ser escalado en una implementación que requiera una medición ambiental, sin depender de un servidor central.

VIII. Trabajo futuro

Al momento de escribir este artículo Espressif lanzó la versión ESP32-S3 mejorada. Este nuevo kit de desarrollo muestra más y mejores prestaciones que sus predecesores. Según el fabricante, es un microcontrolador ideal para aplicaciones IoT. Por otra parte, la *IOTA Foundation* se encuentra en proceso de lanzar una actualización de la red para mejorar la integración. Sin embargo, hasta el momento la misma se encuentra en una etapa experimental y ninguno de los parámetros está finalizado. Por tal motivo, se seguirán realizando experimentos para generar soluciones con la última versión de todos sus elementos y componentes.

References

[1] K. Rose, S. Eldridge, and L. Chapin, "The internet of things: An overview," *The internet society (ISOC)*, vol. 80, pp. 1–50, 2015.

[2] C. Rowland, E. Goodman, M. Charlier, A. Light, and A. Lui, *Designing connected products: UX for the consumer Internet of Things*. " O Reilly Media, Inc.", 2015.

[3] M. ATT&CK, "Mitre att&ck," *URL: https://attack. mitre. org*, 2021.

[4] B. Farahani, F. Firouzi, and M. Luecking, "The convergence of iot and distributed ledger technologies (dlt): Opportunities, challenges, and solutions," *Journal of Network and Computer Applications*, vol. 177, p. 102936, 2021.

[5] H. F. Atlam and G. B. Wills, "Intersections between iot and distributed ledger," in *Advances in Computers*. Elsevier, 2019, vol. 115, pp. 73–113.

[6] R. Buyya and A. V. Dastjerdi, *Internet of Things: Principles and paradigms*. Elsevier, 2016.

[7] J. Kehrli, "Blockchain explained," *Netguardians [en línia].[Data de consulta: 25 de juny de 2017]¡ https://www. netguardians. ch/news/2016/11/17/blockchain-explained-part-1*, 2016.

[8] V. Gisbert Soler and A. I. Pérez Molina, "Blockchain vs iso 9001: 2015," *3C Tecnología*, vol. 8, no. 2, pp. 37–48, 2019.

[9] I. Fundation, "Coordinator. part 2: Iota is a dag, not a blockchain," https://blog.iota.org/coordinator-part-2-iota-is-a-dag-not-a-blockchain-2df8ec85200f/, 11 2018.

[10] ——, "The transparency compendium," https://blog.iota.org/the-transparency-compendium-26aa5bb8e260/, 06 2017.

[11] W. F. Silvano and R. Marcelino, "Iota tangle: A cryptocurrency to communicate internet-of-things data," *Future Generation Computer Systems*, vol. 112, pp. 307–319, 2020.

[12] R. Hadidi, J. Cao, M. S. Ryoo, and H. Kim, "Robustly executing dnns in iot systems using coded distributed computing," in *Proceedings of the 56th Annual Design Automation Conference 2019*, 2019, pp. 1–2.

[13] A. Maier, A. Sharp, and Y. Vagapov, "Comparative analysis and practical implementation of the esp32 microcontroller module for the internet of things," in *2017 Internet Technologies and Applications (ITA)*. IEEE, 2017, pp. 143–148.

[14] C. E. Romero, A. M. Elustondo, R. K. Der Boghosian, and M. C. Fontela, "Nodo experimental de registro e inmutabilidad de variables ambientales," in *III Simposio Argentino de Informática Industrial e Investigación Operativa (SIIIO 2020)-JAIIO 49 (Modalidad virtual)*, 2020.

[15] V. Frittelli, D. Serrano, R. Teicher, F. Steffolani, M. Tartabini, J. Fernández, and G. Bett, "Uso de python como lenguaje inicial en asignaturas de programación," *Editor Responsable*, vol. 132, 2013.

[16] N. H. Tollervey, *Programming with MicroPython: embedded programming with microcontrollers and Python*. " O'Reilly Media, Inc.", 2017.

[17] Bosch, "Bmp180," https://ae-bst.resource.bosch.com/media/_tech/media/product_flyer/BST-BMP180-FL000.pdf, 04 2013.

[18] B. Preneel, "Cryptographic hash functions," *European Transactions on Telecommunications*, vol. 5, no. 4, pp. 431–448, 1994.

[19] I. Fundation, "Hornet. community driven iota node," https://wiki.iota.org/hornet/welcome, 11 2021.

[20] T. Neudecker and H. Hartenstein, "Network layer aspects of permissionless blockchains," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 838–857, 2018.