

Ingenieros industriales desarrollando proyectos con tecnologías IOT y Arduino – ¿Qué se necesita, cómo iniciar?

Industrial engineers developing projects with IOT and Arduino technologies - What do you need, how to start?

Jorge Eduardo Royero Vergara*

* Magister en Gestión Logística. Ingeniero Electrónico. Docente escuela de ciencias básicas, tecnologías e ingenierías. Universidad Nacional Abierta y a Distancia UNAD, Barranquilla. jeroyero@gmail.com

Fecha de recepción: 17 de abril de 2021

Fecha de aceptación: 13 de junio de 2021

Citación:

Royero Vergara, J. E. (2021). Ingenieros industriales desarrollando proyectos con tecnologías IOT y Arduino – ¿Que se necesita, como iniciar? Gestión, Competitividad e innovación(Enero-Junio 2021), 1-10.

RESUMEN

Las tecnologías de automatización, robótica y domótica, generalmente son competencias de programas relacionados con la programación como la ingeniería de sistemas o la electrónica como la ingeniería electrónica, y no es común ver a ingenieros industriales participando directamente en estas, pero hoy en día ya existe la necesidad de que los ingenieros industriales reconozcan estas competencias y que estén en capacidad de interactuar con ella de manera mas cercana, hay debilidades, pero también hay fortalezas como diseñador de modelos CAD y los procesos de fabricación. La automatización de procesos, la robótica y domótica están presente hoy en día en casi todos los aspectos de nuestra vida cotidiana que hace necesario introducir más a los ingenieros industriales desde fases más tempranas que incluyan la programación y electrónica dirigida a este tipo de proyectos tecnológicos. En este documento describimos la importancia de desarrollar e incorporar estas competencias en el proceso de formación de los estudiantes de ingeniería industrial y de cómo adquirirlas en el desarrollo profesional, y se propone la ruta a seguir para el aprendizaje y desarrollo de estas a partir de la experiencia personal recorrida en este camino.

Palabras claves: Robótica, domótica, internet de las cosas, Arduino, placas esp8266, Arduino uno, Arduino nano, esp32, automatización, industria 4.0, ingeniería industrial, educación disruptiva, STEM.

ABSTRACT

Automation, robotics and home automation technologies are generally competencies of programs related to programming such as systems engineering or electronics such as electronic engineering, and it is not common to see industrial engineers participating directly in these, but today it already exists the need for industrial engineers to recognize these competencies and to be able to interact with them more closely, there are weaknesses, but there are also strengths as a designer of CAD models and manufacturing processes. Process automation, robotics and home automation are present today in almost all aspects of our daily lives that make it necessary to introduce more industrial engineers from earlier phases that include programming and electronics aimed at this type of technological projects. In this document we describe the importance of developing and incorporating these competencies in the training process of industrial engineering students and how to acquire them in professional development, and we propose the route to follow for their learning and development based on the personal experience traveled on this path.

Keywords: Robotics, home automation, internet of things, Arduino, esp8266 boards, Arduino uno, Arduino nano, esp32, automation, industry 4.0, industrial engineering, disruptive education, STEM.

1. Introducción

Los constantes avances tecnológicos y la creciente masificación de la automatización, inteligencia artificial y la domótica están obligando a que los actuales y futuros ingenieros industriales tengan esas competencias que les permitan diseñar procesos integrados a estas tecnologías.

Los hogares inteligentes, las industrias inteligentes incorporan estas tecnologías que permiten la automatización de tareas y la integración de los equipos, electrodomésticos, luces, tomacorrientes, etc.

Las industrias cada vez más están integrando sus procesos a la automatización y control remoto donde el internet es la pieza clave y los ingenieros tienen que ser capaces de entender esta integración, los estudiantes de ingeniería industrial dentro de su plan de formación aprenden competencias relacionadas con estas tecnologías como lo son el diseño 3d, electricidad y electrónica básica, automatización y algoritmos básicos, pero no son capaces de lograr la integración de ellos y es por ello que se plantea la necesidad de incorporarla acorde a las nuevas tecnologías disponibles.

Hay diferentes niveles de automatización, unos más complejos que otros, y actualmente existen muchas tecnologías disponibles en el mercado que permiten hacer esta integración de manera mucho más simple para el hogar como por ejemplo sistemas inteligentes de Google y Amazon, ¿pero que hay disponible si queremos desarrollar nuestros propios proyectos, si tenemos necesidades diferentes en cuanto a demanda de corriente, número de salidas o entradas, número de sensores, necesidad de conectores específicos?. Es aquí donde se hace necesario tener los conocimientos en el diseño y construcción de prototipos que den solución a las necesidades requeridas.

Existe la necesidad de que los ingenieros industriales tengan la capacidad y habilidad de diseñar y construir prototipos funcionales adaptados a estas y que comprendan el funcionamiento y la integración del internet de las cosas con la automatización de equipos.

2. Metodología

La investigación se realizó partiendo de la experiencia en diseño y desarrollo de múltiples proyectos de automatización, robóticos y domóticos, y de mis competencias como ingeniero industrial, en la cual encontré la ruta más favorable para el aprendizaje y desarrollo de las habilidades requeridas.

3. Resultados

No es común ver ingenieros industriales desarrollando proyectos, programas o prototipos basados en tecnologías IOT, este tipo de actividades se acercan más a las ingenierías de sistemas, electrónica, robótica, o similares, entonces ¿es posible que un ingeniero industrial pueda competir en este segmento?

Soy ingeniero industrial con más de 5 años desde que inicie en el desarrollo de proyectos IOT utilizando tecnología Arduino, realmente no ha sido fácil, pero tampoco imposible, se necesita mucha dedicación, investigación y tiempo para aprender a programar y a utilizar de forma correcta muchos componentes como son sensores, actuadores, y elementos

electrónicos como las resistencias, condensadores, reguladores de voltaje, opto acopladores, opto triac, diodos, transistores, mosfet, etc.

El desarrollo de prototipos que integran componentes electrónicos, micro procesadores, sensores, actuadores y partes mecánicas implica una integración de habilidades que se adquieren gradualmente iniciando por el reconocimiento de las diferentes tecnologías de impresión 3d, suministros, configuración y parámetros de calidad de estos, acompañado por habilidades de diseño 3d que se requieren como punto de partida para la elaboración de las piezas, componentes fijos o partes móviles, dependiendo de las características del proyecto.

El diseño 3d es el proceso mediante el cual convertimos una idea de proyecto en componentes o partes requeridas según las necesidades o características deseadas para luego ser impresas y validar su funcionalidad según lo esperado, es por esto la importancia de esta habilidad en un ingeniero industrial.

Por otro lado este tipo de proyectos requiere la integración de elementos electrónicos básicos que obliga al ingeniero industrial a entender el funcionamiento de ellos como son las resistencias, condensadores, reguladores de voltaje, transistores, triacs, opto triacs, opto acopladores, diodos, entre otros que serán necesarios en la parte de control, y aquí observamos otra necesidad de habilidades y/o competencias en electricidad y electrónica básica que en la mayoría de los casos se integran en el plan de estudio de programa.

Los proyectos robóticos o domóticos o inteligentes integran componentes capaces de ser programados y reprogramados desde diferentes lenguajes según la tecnología utilizada, que son la parte lógica y que es capaz de realizar acciones o ejecutar instrucciones según condiciones establecidas, por ejemplo, encender una luz cada 2 horas, o encender el sistema de riego a las 9 pm cada día, lo que evidencia la necesidad de estas habilidades o competencias en los ingenieros industriales.

Aunque es cierto que la ingeniería con el perfil de programación es la ingeniería de sistemas no podemos desconocer la creciente necesidad de que los ingenieros industriales conozcan y adquieran competencias de programación básicas que les permitan comprender, y desarrollar algoritmos de programación de forma más amigable y fluida.

Por último y no menos importante el ingeniero industrial debe reconocer las diferentes tecnologías de entrada y salida, su funcionamiento básico y posibles campos de aplicación en proyectos robóticos y/o domóticos como son sensores, motores, servos, circuitos, receptores y emisores de frecuencia, dispositivos bluetooth, wifi, y muchos otros utilizados en el desarrollo de estos proyectos.

La realidad de hoy es que muchos ingenieros no cuentan con todas o algunas de las anteriores competencias, y muchos de los futuros ingenieros tampoco, pero no significa que sea imposible, como cualquier competencia esta puede ser adquirida con investigación, estudio, práctica y disciplina.

Existen diferentes fases a recorrer en este camino, y aun que no hay un camino claro o único a seguir, desde mi experiencia y en base al camino recorrido para iniciar en el desarrollo de modelos o proyectos o prototipos robóticos, IOT, domóticos, puedo sugerir el siguiente:

Nivel 1: desarrollo de proyectos básicos con la placa Arduino uno R3

En este nivel se da inicio al desarrollo de programas básicos con la placa Arduino uno R3, es necesario hacer un reconocimiento de la placa, sus partes, identificar los diferentes pines diferenciando los digitales de los analógicos.

Se sugiere iniciar con programas básicos para encender y apagar leds con la ayuda de una resistencia, incluir tiempos de encendido y apagado y alternancia de pines como salidas digitales, encender led utilizando pines de salida analógicos, y utilizar pulsadores en pines configurados como entrada para encender leds.

Para el desarrollo de estos programas básicos se recomienda inicialmente el uso de entornos de programación en bloques y seguidamente en entorno IDE de Arduino, de esta manera se adquieren los conocimientos de ambas plataformas en la cual el entorno en bloques generalmente usa el lenguaje Python o, mientras que el IDE de Arduino utiliza un lenguaje de programación de alto nivel Processing, lo que significa que es similar a C++.

Dentro de los entornos de programación en bloque podemos usar Mblock (<https://mblock.makeblock.com/en-us/>). MBlock es un entorno gráfico de programación basado en el editor Scratch 3.0 para introducir de forma sencilla la programación. Otras plataformas de programación en bloque son Minecraft Code Builder, Microsoft Makecode, Code.org, Tynker, Microbit.

En esta etapa es recomendable experimentar la programación de componentes individuales realizando variaciones en la programación, dentro de los componentes se recomienda trabajar con leds, servos, módulos relés, sensores de ultra sonido, sensores infra rojos, módulos bluetooth, módulos lcd, botones pulsadores, sensor de humedad y temperatura, sensor de ruido, sensor de movimiento, controlador de motores, potenciómetro, etc.

Es importante iniciar la programación de estos componentes básicos en módulos integrados. Para entender mejor a que me refiero con módulos integrados pondré como ejemplo la programación de relés, en el mercado existen relés y módulos relés, la diferencia radica en que el módulo relé trae ya integrado unos componentes electrónicos de control como son resistencia, transistor, led, opto acoplador y pines o terminales para conexiones, y la conexión a una placa Arduino se hace solamente a través de 3 cables, mientras que realizar la conexión del relé individual implica la necesidad de utilizar un protoboard al cual conectamos el transistor que tiene 3 pines, una resistencia con dos pines, un opto acoplador con 4 o 6 pines, un diodo con dos pines, y terminales conectados a las salidas del relé, lo que finalmente se traduce en muchos cables y conocimientos mayores en conexión y utilización de estos además de un circuito inestable con alto riesgo de mala conexión y de una desconexión en cualquier momento.

Para la construcción de prototipos más avanzados y profesionales son necesarios estos componentes sueltos (sin módulos) y sus conexiones se hacen montados sobre un PCB prediseñado, para lo cual ya estamos hablando de niveles de conocimiento mayores por lo que no aplican en este primer nivel.

Nivel 2: desarrollo de proyectos básicos con la placa Esp8266 y/o familias.

En el nivel 1 se propone trabajar con placa Arduino uno o similares las cuales son muy básicas y requieren instrucciones muy sencillas de programación, además de que se integran fácilmente al entorno de programación Arduino.

La placa ESP8266 fue desarrollada y producida por el fabricante chino Espressif Systems, esta placa contiene un MCU (Micro Controller Unit) muy eficiente y que se integra con tecnología Wifi lo que la hace ideal para el desarrollo de proyectos domóticos y otros que requieran conectividad wifi.

Esta placa permite desarrollar los mismos programas que con la placa Arduino uno R3 o la Arduino nano, además de otros programas y proyectos domóticos que son imposible con la anterior sin componentes adicionales.

La empresa Espressif Systems dispone de una amplia variedad de placas basadas en la familia de la ESP8266 y otras mucho más dotadas de mayor velocidad, mayor número de entradas y salidas digitales, conectividad mejorada, y muchas ventajas que son muy interesantes para programadores con más experiencia en el desarrollo de programas y/o prototipos más complejos para los cuales la esp8266 se queda corte precisamente por las limitantes anteriores.

Se recomienda el aprendizaje de esta placa directamente en el IDE de Arduino por la gran cantidad de librerías existentes para esta y que solo se podrían usar en este entorno de programación.

Una librería en Arduino es un programa diseñado específicamente para que el entorno de programación Arduino reconozca instrucciones y simplifique los comandos de programación de este componente. Una librería es un archivo que contiene códigos de programa hechos por terceros específicamente para hacer funcionar sensores, placas, actuadores de manera mucho más sencilla y que solo se deben integrar al IDE de Arduino para ser utilizadas.

Los primeros programas a realizar en esta placa son la base para entender el funcionamiento de esta dentro de los cuales podrían iniciar con:

- Encender y apagar un led vía wifi desde el celular con una aplicación como por ejemplo Blink.
- Encender y apagar un led desde una página web como por ejemplo Adafruit (<https://io.adafruit.com>)
- Encender y apagar un led utilizando el asistente de Google
- Luego es recomendable iniciar programas que permitan accionar relés, controlar motores DC, mover servos y más.

Las placas de la familia ESP8266 son ideales para el desarrollo de proyectos o prototipos domóticos o de automatización y control de circuitos, aparatos, electrodomésticos y otros vía wifi, lo que permite el control y automatización a distancia desde cualquier lugar del mundo. Hay una variedad de placas pertenecientes a esta familia y que se diferencian en cuanto al número de entradas y salidas digitales y análogas y tamaño y la elección de una de estas depende de las características del proyecto a desarrollar, por ejemplo para la construcción de un switch domótico para el control de una luz de un departamento sería suficiente y practico utilizar una placa Esp01, o la Esp12f la cual es muy pequeña y se podría construir un circuito para ser instalado dentro del interruptor de pared, o si por ejemplo lo que quiero es controlar un grupo de 4 luces o tomacorrientes ya en este caso necesitaría una placa D1mini, o la tradicional Esp8266.

En esta fase es importante explorar y utilizar cada una de estas variaciones de placas con el propósito de entender sus ventajas y desventajas, diferencias y mejor aplicabilidad según la necesidad, disponibilidad y el proyecto a realizar.

La programación de placas de esta familia implica también el conocimiento de las diferentes plataformas tecnológicas que brindan soporte y conectividad a estas, entre estas plataformas tenemos IFTTT, Adafruit, Google Cloud, Zetta, Blynk, OpenRemote, y muchas más. Estas plataformas permiten la utilización de diferentes protocolos de comunicación entre dispositivos, estas plataformas son las que permiten la verdadera conexión IOT enlazando los dispositivos permitiendo el control de estos desde dispositivos móviles, pc, Tablet, etc.

La recomendación en este caso es iniciar con la utilización de una de estas plataformas, investigar y aprender cómo utilizarlas desde Arduino con la placa ESP8266, iniciar con programación básica para encender y apagar un led desde una app o desde la web directamente, luego cambiar el led por un módulo relé, incrementar a dos leds y/o dos relés, e ir avanzando poco a poco hasta dominar su funcionamiento, configuración, parametrización y aplicación. Las aplicaciones para esta tecnología y posibles proyectos son muchísimos.

Nivel 3: desarrollo de proyectos básicos con la placa Esp32

Después de adquirir habilidades básicas en la programación y aplicación de las tarjetas de la familia Arduino y esp8266 es hora de incursionar en el desarrollo de proyectos mas complejos que integren la domótica, automatización y control de dispositivos mediante el internet (IOT), aplicaciones móviles, bluetooth, y procesos mas complejos que requieran mayor capacidad de procesamiento de datos y tareas multifunción o tareas paralelas.

La esp32 es una placa fabricada por la misma empresa que fabrica las esp8266 pero con grandes cambios referente a estas por el incremento notorio de pines, versatilidad, adelantos tecnológicos, capacidad de procesamiento entre otros. Los proyectos recomendados a trabajar inicialmente serian los mismos trabajados con las placas esp8266, d1 mini, y las demás de la familia con el propósito de dominar las diferencias en el manejo de pines y librerías de control, luego se recomienda el inicio de proyectos más avanzados relacionados con sistemas integrados de control y automatización.

Nivel 4: Desarrollo de prototipos en PCB - (Printed Circuit Board)

Como último nivel es necesario pasar del montaje superficial de componentes sobre las diferentes placas y/o protoboards a PCB o baquelitas comúnmente llamadas.

Un PCB es una tarjeta o superficie dotada de pistas o caminos o hilos de cobre y agujeros que permiten la conexión de múltiples componentes electrónicos cuyo objetivo principal es la reducción del espacio requerido en un prototipo gracias a la eliminación de los cables conectores.

Hay diferentes formas y procesos para realizar un PCB, hay métodos manuales, semi automatizados y automatizados. Para iniciar una opción es utilizar procesos manuales que son muy sencillos, pero con baja calidad y fiabilidad en estos, dentro de estos tenemos el método del planchado, dibujo a mano, los cuales requieren un proceso de desgaste del cobre mediante agitación con cloruro férrico.

Otra opción y es la que recomiendo es realizar el diseño del PCB utilizando programas especializados y gratuitos como son el Kitcat, Pcbwizard, Fritzing, EasyEDA, entre otros para luego generar un archivo especializado del tipo Gerber con el cual pueden con tratar una empresa especializada de forma online para su fabricación y envió en tiempos muy cortos a un bajo costo, algunas de estas empresas son Pcbway, Jlcpcb, etc

La calidad de estos circuitos es la más alta posible con los mejores acabados y la posibilidad de error se limita solamente a las fallas en el diseño original enviado. Adicional a las dos opciones existe una tercera y es la fabricación de los PCB utilizando maquinas CNC o mini CNC en la cual se fabrican las pistas y los agujeros de forma precisa con procesos de desbaste y taladrado controlado por computador.

Esta opción es muy recomendada ya que permite la creación de los circuitos (prototipos) de manera muy rápida con una calidad media y facilita la validación de este realizando el montaje de prueba y que es funcional pero no profesional para luego de validado proceder al envío de la fabricación en empresas especializadas.

¿Qué fortalezas tienen los ingenieros industriales para el desarrollo de estos proyectos?

Los ingenieros industriales durante su formación adquieren muchas habilidades y potencializan otras, en este caso puntual podemos destacar:

- Habilidad para el pensamiento computacional lo que facilita entender y programar con mucha facilidad los diferentes lenguajes.
- Habilidad para el diseño de modelos 3d lo que permite el desarrollo de prototipos con más facilidad.
- Conocimientos básicos de electricidad y electrónica lo que facilita la construcción e integración de proyectos y prototipos.
- Habilidad en resolución de problemas, desarrollo de ideas, dirección de proyectos, desarrollo de experimentos, validación de datos, análisis de información, desarrollo de planes de trabajo etc.

¿Qué debilidades tienen los ingenieros industriales para el desarrollo de estos proyectos?

Se encuentran en desventaja frente a ingenieros de sistemas los cuales son expertos en programación y requieren más dedicación y estudio para el dominio de estos lenguajes. El ingeniero industrial reconoce elementos básicos de electrónica, pero no sabe aplicarlos en diseño de prototipos por lo que necesita reforzar y aprender a utilizarlos de manera correcta. Los diferentes prototipos robóticos y/o domóticos requieren el uso de resistencias, condensadores, transistores y otros componentes que permitan regular la corriente, controlar entradas, salidas, proteger los circuitos, etc.

Referencias

Castro G, Isabel R. (2021). La educación disruptiva.
<https://hdl.handle.net/20.500.11777/4923>

- CLAYTON M. CHRISTENSEN, MICHAEL B. HORN y CURTIS W. JOHNSON (2008) Clase disruptiva: cómo La innovación disruptiva cambiará la forma en que el mundo aprende. McGraw-Hill
- Ocampo-Eyzaguirre, D. ., Sucari, W. ., Anaya, J. ., Medina, R. ., & Zuñiga-Sánchez, H. . (2021). Educación disruptiva: Nuevos desafíos en la formación de investigadores sociales en tiempos de pandemia y distanciamiento social. *Apuntes Universitarios*, 12(1). <https://doi.org/10.17162/au.v11i5.916>
- Lozano, R. (2011). Las ‘TIC/TAC’: de las tecnologías de la información y comunicación a las tecnologías del aprendizaje y del conocimiento. *Anuario ThinkEPI*, 5.
- Cortés-González, P., Leite-Méndez, A. E., & Rivas-Flores, J. I. (2020). Una experiencia disruptiva en educación secundaria: El caso del IES Esmeralda. Recuperado de <https://repositorio.uam.es/handle/10486/689683>
- Aikenhead G.S. (2006) *Science Education for Everyday Life. Evidence-Based Practice*. New York: Teachers College Press.
- Akerson V.L., Burgess A., Gerber A., Guo M. (2018) Disentangling the meaning of STEM: Implications for Science Education and Science Teacher Education. *Journal of Science Teacher Education*, 29(1), 1-8.
- Bonney R., Cooper C. B., Dickinson J., Kelling S., Phillips T., Rosenberg K.V. y Shirk J. (2009) Citizen Science: A Developing Tool for Expanding Science Knowledge and Scientific Literacy. *BioScience*, 59 (11), 977–984.
- Bybee R.W. (2010) What is STEM Education? *Science*, 329 (5995), 996.
- Windschitl M., Thompson J. y Braaten M. (2008) Beyond the scientific method: Model-based inquiry as a new paradigm of preference for school science investigations. *Science Education*, 92(5), 941-967.
- Wing J.M. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33–35.
- Barker, B., & Ansorge, J. (2007). Robotics as means to increase achievement scores in an informal learning environment. *Journal of Research on Technology in Education*, 39(3), 229-243. doi: <https://doi.org/10.1080/15391523.2007.10782481>
- Benitti, F. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education*, 58(3), 978-988. doi: <https://www.sciencedirect.com/science/article/abs/pii/S0360131511002508>
- Celis, D. (2020). Aporte de la metodología STEAM en los procesos curriculares. Universidad Antonio Nariño. Doi: <http://repositorio.uan.edu.co/handle/123456789/2889>
- Zollman, A. (2012). Learning for STEM Literacy: STEM Literacy for Learning. *School Science and Mathematics*, 112(1), 12–19. doi: <https://doi.org/10.1111/j.1949-8594.2012.00101.x>
- Thibaut L, Ceuppens S, De Loof H, et al. Educación STEM integrada: una revisión sistemática de las prácticas de instrucción en la educación secundaria. *Revista europea de educación STEM* . 2018; 3 (1), 02. <https://doi.org/10.20897/ejsteme/85525>

- Melo Niño, D. S. (2020). Integración de las ciencias básicas en educación media con enfoque STEM en robótica comparada con una metodología tradicional de enseñanza. (Tesis de maestría). Universidad Pedagógica y Tecnológica de Colombia, Dutama. <http://repositorio.uptc.edu.co/handle/001/3202>
- Santillán Aguirre, J. P., Cadena Vaca, V. del C., & Cadena Vaca, M. (2019). Educación Steam: entrada a la sociedad del conocimiento. *Ciencia Digital*, 3(3.4.), 212-227. <https://doi.org/10.33262/cienciadigital.v3i3.4.847>
- Jimenez Leon, R., Magaña Medina , D. E., & Aquino Zúñiga, S. P. (2021). Gestión de tendencias STEM en educación superior y su impacto en la industria 4.0. *Journal of the Academy*, (5), 99-121. <https://doi.org/10.47058/joa5.7>
- Penagos, L., Fonseca, S. y Cruz, A. (2020). Implementación del método STEM con base en el currículo de EngageK12 en un colegio distrital de la ciudad de Bogotá. [Tesis de pregrado, Universidad Cooperativa de Colombia]. Repositorio Institucional UCC. <https://repository.ucc.edu.co/handle/20.500.12494/19846>