UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE CUENCA

CARRERA INGENIERÍA ELECTRÓNICA

CONTROL DE TEMPERATURA DE UN PROCESO TÉRMICO UTILIZANDO EL MICROCONTROLADOR STM32F407 BAJO PROGRAMACIÓN E INTERACCIÓN CON SIMULINK DE MATLAB

Trabajo de titulación previo a la obtención
del título de Ingeniero Electrónico

AUTORES: CARLOS ANDRÉS CUJI CÁCERES
PAUL ESTEBAN JARA BARRETO
TUTOR: ING. JULIO CÉSAR ZAMBRANO ABAD, Ph.D.

Cuenca - Ecuador
2023

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Carlos Andrés Cuji Cáceres con documento de identificación N°0302241880 y Paul Esteban Jara Barreto con documento de identificación N°0105826416; manifestamos que:

Somos autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Cuenca, 18 de abril del 2023

Atentamente,

Carlos Andrés Cuji Cáceres
0302241880

Paul Esteban Jara Barreto
0105826416

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, Carlos Andrés Cuji Cáceres con documento de identificación N°0302241880 y Paul Esteban Jara Barreto con documento de identificación N°0105826416; expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto técnico: "Control de temperatura de un proceso térmico utilizando el microcontrolador STM32F407 bajo programación e interacción con Simulink de Matlab", el cual ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, 18 de abril del 2023

Atentamente,

Carlos Andrés Cuji Cáceres
0302241880

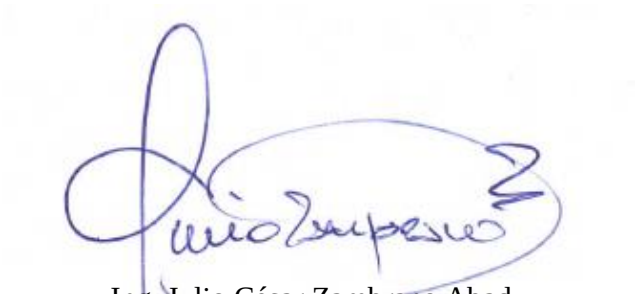
Paul Esteban Jara Barreto
0105826416

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACION

Yo, Julio César Zambrano Abad con documento de identificación N° 0301489696, docente de la Universidad Politécnica Salesiana, declaro que bajo mi autoría fue desarrollado el trabajo de titulación: CONTROL DE TEMPERATURA DE UN PROCESO TÉRMICO UTILIZANDO EL MICROCONTROLADOR STM32F407 BAJO PROGRAMACIÓN E INTERACCIÓN CON SIMULINK DE MATLAB, realizado por Carlos Andrés Cuji Cáceres con documento de identificación N° 0302241880 y por Paul Esteban Jara Barreto con documento de identificación N° 0105826416, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca, 18 de abril del 2023

Atentamente,



Ing. Julio César Zambrano Abad
0301489696

ÍNDICE GENERAL

ÍNDICE FIGURAS

ÍNDICE DE TABLAS

AGRADECIMIENTOS

Agradezco a Dios por bendecirme día a día con vida y salud, por darme la fuerza necesaria para seguir adelante y poder culminar mi ingeniería.

Gracias a mi esposa Lisseth y a mi hijo Christopher que son el motor de mi vida y fueron el impulso que me motivo a seguir luchando para ser un profesional y por el amor, paciencia que día a día me brindan los amo.

A mis padres Nelson y Lucia que me enseñaron los valores de una familia, fueron son y serán mi ejemplo para seguir les agradezco por su compresión, apoyo, paciencia para culminar con mi carrera y sobre todo por el amor que me brindan día a día.

A mis hermanos Priscila, Juan (+), Dayana, Nelson, Max por el apoyo que me brindaron en momentos difíciles y nunca dejaron de confiar en mí.

Agradezco al Ing. Julio Zambrano por brindarnos de sus conocimientos y ser guía en este trabajo de titulación

Finalmente agradezco a toda mi familia, amigos y compañero de trabajo Carlos Cuji que supieron confiar en mí y apoyarme siendo fundamentales en mi vida universitaria.

Paul Esteban Jara Barreto

DEDICATORIA

El presente trabajo de titulación va dedicado a mis padres Nelson y Lucia, a mi esposa Lisseth y a mi hijo Christopher que pusieron su apoyo, confianza y amor a lo largo de mis estudios, nunca dejaron de confiar en mí y fueron fundamentales para poder culminar mi vida universitaria.

De manera especial agradezco a Lucia y Lisseth porque día a día me enseñan que el amor una madre y una esposa es fundamental, les agradezco por su apoyo fundamental y por no dejarme solo nunca las amo.

Christopher hijo mío, va dedicado para ti también no sabes cuanto me ayudaste fuiste, eres y serás fundamental en mi vida te amo.

Paul Esteban Jara Barreto

AGRADECIMIENTOS

Agradezco a mi madre por ser mi ejemplo de trabajo constante y generosidad. A mi hermana por siempre brindarme calidez y ser ejemplo de fe. A mis tíos/tías por siempre contar con su apoyo y consejos. A mis primos por ser uno de los motivos de mi inspiración. A mis abuelitos por siempre ver por el bienestar de la familia. A mi esposa por todos los momentos que hemos compartido día a día y por brindarme su apoyo siempre.
Agradezco a nuestro tutor el Ing. Julio Zambrano por brindarnos su tiempo al orientarnos en este proyecto.

Carlos Andrés Cuji Cáceres

DEDICATORIA

Dedicado para mis personas amadas:

María Cáceres, Mercedes Cáceres, Victoria Guamán (+), Julio Enrique Cáceres, Diana Curillo.

Carlos Andrés Cuji Cáceres

RESUMEN

En el presente trabajo de investigación se plantea el diseño y la implementación de un sistema de control PID y un sistema de control en espacio de estados para controlar un proceso térmico de laboratorio. La novedad del trabajo radica en que los controladores son implementados en una tarjeta STM32F407, la cual se programa con Simulink de Matlab.

De esta manera se aprovecha toda la potencialidad de Simulink para poder programar la tarjeta y ejecutar los controladores en tiempo real. Para poder programar la tarjeta se utiliza la librería Waijung. Además se crea un interfaz de operación y monitoreo dentro de Simulink para poder observar el desempeño del controlador.

En primera instancia para poder familiarizarnos con la programación de la tarjeta se realiza una programación sencilla para controlar las entradas y salidas. Una vez conocido el proceso se procede con la implementación de los controladores.

Se debe indicar que el diseño de los controladores inicia con la obtención del modelo del proceso térmico. Para esto se hace una captura de datos del sistema en lazo abierto y se utiliza una técnica de identificación para obtener la función de transferencia que representa la dinámica del sistema sobre un punto de operación.

A partir de la función de transferencia se diseñan los controladores utilizando algunas herramientas y comandos de Matlab. A continuación, los controladores son implementados y evaluados.

PALABRAS CLAVES:

VARIABLE MANIPULADA: Es la variable o condición de la planta que se modifica a final del influir sobre la variable de control a través de la dinámica de la planta.

PLANTA: Es un equipo o simplemente un conjunto o subsistema de una maquina o planta térmica o química que puede ser objeto de nuestro control.

PROCESO: Normalmente se orienta esta denominación a reacciones químicas u operaciones físicas industriales que pueden ser controladas.

PID: Controlador proporcional, integral y derivativo.

CAD: Convertidor analógico-digital.

CDA: convertidor digital-analógico.

FT: Función de transferencia

MCU: Unidad de control principal

DC: Corriente Continua

(ADC): convertidor analógico- digital

(DAC): convertidor digital-analógico

ABSTRACT

In the present research work, the design and implementation of a PID control system and a state space control system to control a laboratory thermal process are proposed. The novelty of the work lies in the fact that the controllers are implemented in an STM32F407 card, which is programmed with Simulink from Matlab.

In this way, the full potential of Simulink is used to be able to program the card and execute the controllers in real time. To be able to program the card, the Waijung library is used. In addition, an operation and monitoring interface is created within Simulink to be able to observe the performance of the controller.

In the first instance, in order to familiarize ourselves with the programming of the card, a simple programming is carried out to control the entrances and exits. Once the process is known, we proceed with the implementation of the controllers.

It should be noted that the design of the controllers begins with obtaining the model of the thermal process. For this, an open-loop system data capture is made and an identification technique is used to obtain the transfer function that represents the dynamics of the system on an operating point.

From the transfer function, the controllers are designed using some Matlab tools and commands. The controllers are then implemented and evaluated.

KEYWORDS:

MANIPULATED VARIABLE: It is the variable or condition of the plant that is modified at the end of influencing the control variable through the dynamics of the plant.

PLANT: It is a piece of equipment or simply a set or subsystem of a thermal or chemical machine or plant that may be subject to our control.

PROCESS: Normally this denomination is oriented to chemical reactions or industrial physical operations that can be controlled.

PID: Proportional, integral, and derivative controller.

CAD: Analog-digital converter.

DAC: Digital-analog converter.

FT: Transfer Function

MCU: Main Control Unit

DC: Direct Current

(ADC): Analog-digital converter

(DAC): Digital-analog converter

Capítulo 1

1 Introducción

1.1 Antecedentes

En la actualidad, los sistemas de control juegan un papel fundamental a nivel industrial. En la mayoría de los casos estos sistemas se implementan sobre plataformas especializadas o sobre controladores lógicos programables. No obstante, debido al gran desarrollo de los microprocesadores, en estos últimos años se han reportado en la literatura varios casos de implementación de esquemas de control sobre sistemas embebidos o plataformas alternativas (Espinel Cangui, 2012).

Actualmente se puede encontrar en el mercado una gran variedad de sistemas embebidos, entre los cuales destacan sistemas como: Arduino, Raspberry PI, NEXYS 4 DDR, entre otros. El presente proyecto se basa en la utilización del sistema embebido **SMT32F407 Discovery** (Silva-Díaz et al., 2019), el cual tiene ciertas características interesantes. Una de las características más importantes de este sistema embebido es que gracias a su librería **Waijung** puede interconectarse con Simulink de MATLAB. Bajo este contexto, en este proyecto de implementación se plantea el desarrollo de esquemas de control automático programados de una manera fácil y amigable, explotando todas las bondades de Simulink de Matlab (Proaño, 2016; Taipe, 2019).

Para el desarrollo de esta investigación se cuenta con una planta térmica de laboratorio (véase en la figura 1.1). En este sentido, se plantea la implementación de un control PID y un control por retroalimentación de estados para controlar la temperatura del proceso.

Figura 1.1 Planta térmica para control de temperatura [(Ilustración propia de los autores)].

1.2 Justificación e importancia

El proyecto inicia con una investigación de la tarjeta STM32F407, analizando sus entradas, salidas, métodos de interconexión, compatibilidad con software de programación de alto de nivel como lo es Simulink. De esta manera se creará una base sobre la cual se puede investigar y desarrollar a futuro los esquemas de control planteados.

Con este proyecto se pretende dotar a los laboratorios de una herramienta innovadora para implementar controladores, ya que se utilizará un software muy sofisticado y poderoso como Simulink para programar un controlador dentro de un sistema embebido. De esta manera se puede aprovechar todas las facilidades de programación que brinda Simulink y el potencial del procesador de la tarjeta STM32F407.

Simulink de Matlab es una herramienta de programación de alto nivel. Se trata de un lenguaje de programación gráfico, bastante amigable y muy potente, donde existe muchos módulos y bloques que traen sus algoritmos embebidos. La idea de este proyecto es construir los controladores utilizando estos módulos y bloques y descargar los programas creados sobre el sistema embebido.

Con el desarrollo de este proyecto se pretende introducir e iniciar una línea de investigación para el desarrollo de controladores utilizando sistemas embebidos de bajo costo. Este tipo de sistemas embebidos, por su tamaño pueden ser utilizados en diversas aplicaciones, como por ejemplo en la robótica y en la navegación de vehículos aéreos.

De igual manera, el hecho de poder programar la tarjeta con Simulink, abre varios frentes de investigación que se podrían centrar en el análisis de desempeño de la tarjeta para otro tipo de aplicaciones como por ejemplo las redes neuronales. En este caso, se podría desarrollar y entrenar una red neuronal en Matlab y se podría descargar el programa sobre la tarjeta.

1.3 Objetivos

OBJETIVO GENERAL

Desarrollar un controlador en el sistema microprocesador STM32F407 bajo programación en Matlab Simulink para regular la temperatura de un proceso termino didáctico.

OBJETIVOS ESPECÍFICOS

- Crear una interfaz de comunicación entre el sistema microprocesador STM32F407 y el software Matlab Simulink.
- Analizar la factibilidad de la tarjeta STM32F407 para implementar controladores PID y en Espacios de Estados.
- Implementar los esquemas de control correspondiente para regular la temperatura en un proceso térmico didáctico.
- Validar el desempeño de los controladores mediante pruebas de campo.

GRUPO OBJETIVO (BENEFICIARIOS)

Este proyecto tiene como beneficiarios a todos los estudiantes de las carreras de Ingeniería Electrónica y de Electrónica y Automatización de la UNIVERSIDAD POLITECNICA SALESIANA, que se encargarían de seguir con la investigación y desarrollo del proyecto.

1.4 Arquitectura del sistema

A través del presente trabajo de investigación se pretende implementar dos tipos de controladores, sobre una tarjeta STM32F407, aprovechando la potencialidad y las facilidades de programación que brinda Matlab de Simulink. En la Figura 2 se presenta la arquitectura del sistema. Como elemento principal se tiene la tarjeta STM32F407 sobre la cual se ejecutará el controlador. En adición, esta tarjeta también se encargará de la adquisición de datos del proceso y también tendrá conexión con un computador con Matlab. Cabe indicar que el computador no es un elemento necesario para que funcione el controlador. En este caso, se utilizará el computador únicamente como instrumento de programación y visualización de las variables del controlador (véase en la figura 1.2).

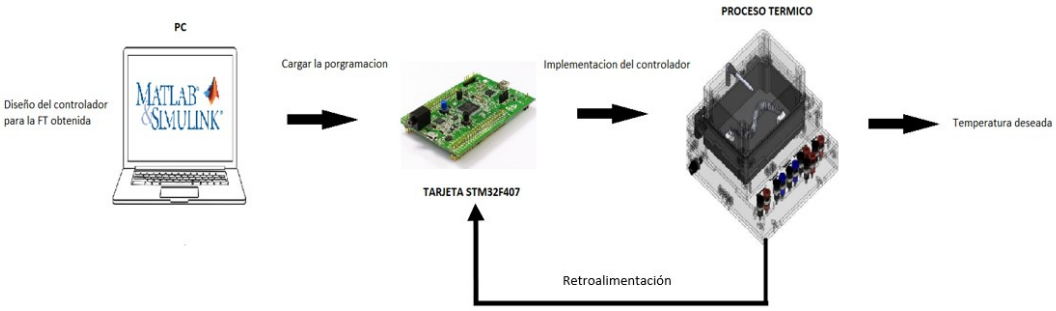


Figura 2.2 Diagrama de la implementación del controlador [(Ilustración por Autores)].

El proceso para controlar es un proceso térmico basado en una niquelina. Este proceso cuenta con un elemento de medición de temperatura con su respectivo transmisor. El transmisor ha sido configurado para que entregue un voltaje entre 0 y 10 voltios. De igual manera la señal de control debe ingresar a la planta bajo el mismo estándar. Por consiguiente, la arquitectura mostrada en la Figura 2 también incluye un sistema electrónico de adaptación de las señales.

1.5 Organización del manuscrito

De aquí en adelante el manuscrito está organizado de la siguiente manera. En el Capítulo 2 se presenta un abordaje de los componentes tecnológicos utilizados en el proyecto. En este capítulo también se presenta el procedimiento que se debe seguir para programar la tarjeta STM32F407 utilizando Simulink de Matlab. Para propósitos de demostración se realiza una pequeña aplicación para manejar entradas y salidas de la tarjeta. En el Capítulo 3 se presenta el diseño e implementación de los controladores. Finalmente, en el Capítulo 4 se reportan algunas conclusiones y posibles trabajos futuros.

Capítulo 2

2 Componentes tecnológicos

2.1 Tarjeta STM32F407 Discovery.

La STM32F4 es un microcontrolador electrónico perteneciente a la familia de dispositivos STM32. Este dispositivo de alta gama tiene múltiples características que lo convierten en una opción ideal para diversas aplicaciones en distintos ámbitos de la electrónica. Su procesador ARM Cortex-M4 de 32 bits brinda alto rendimiento y consume muy poca energía. La STM32F4 también dispone de una memoria flash integrada que permite guardar programas y datos, y una memoria RAM que ofrece espacio para la ejecución de los programas [(*STM32 F4 Cortex™-M4 MCUs - STMicro | Mouser*, s. f.)].

La STM32F4 es una tarjeta que se destaca por su alta velocidad de procesamiento, capacidad de manejar múltiples tareas simultáneamente y amplia variedad de periféricos integrados. Entre estos periféricos se incluyen puertos USB, CAN, Ethernet, SPI e I2C, lo que la hace una opción versátil para una amplia variedad de aplicaciones. [(*O'Connor, D. et al (2016), Universality, Integr... |OECD*, s. f.)].

En la figura 2.1 se muestra la tarjeta STM32F4 Discovery, la cual dispone de periféricos de entrada y salida para la adquisición y entrega de datos, respectivamente. Estos periféricos permiten que el microcontrolador interactúe con otros sistemas de manera eficiente y efectiva. (Proaño, 2016). La STM32F4 Discovery requiere de herramientas específicas para que tanto los usuarios principiantes como los expertos puedan reconocer todos los elementos que se encuentran incorporados en ella. Esta tarjeta también cuenta con opciones de lenguaje de programación en C/C++, así como la posibilidad de programación por bloques en MATLAB mediante la compatibilidad con la librería Waijung. Esto permite a los usuarios programar en el lenguaje que prefieran y utilizar herramientas que se adapten a sus necesidades específicas. (véase en la figura 2.1) (Ramos et al., 2020).

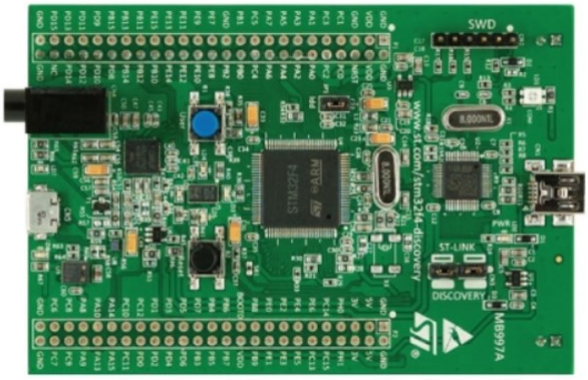


Figura 3.1 Microcontrolador STM32F4DISCOVERY [(Ramos et al., 2020)].

Características.

- Interfaz de comunicación serie: la tarjeta STM32F4 está equipada con múltiples periféricos de comunicación serie, tales como USART, UART, SPI e I2C. Estos periféricos permiten la comunicación con otros dispositivos electrónicos, como sensores, actuadores y otros microcontroladores. De esta manera, se puede lograr una integración efectiva de múltiples componentes electrónicos en una sola plataforma, lo que amplía su rango de aplicaciones y posibilidades de uso [(*STM32 Microcontrollers - STMicro | Mouser*, s. f.)].
- Interfaz USB: La tarjeta STM32F4 posee una interfaz USB que facilita la comunicación con un PC u otro dispositivo que cuente con un puerto USB. Gracias a esta interfaz, es posible programar la tarjeta STM32F4 mediante la conexión USB, lo que permite una actualización sencilla del firmware y la transferencia de datos de manera rápida y eficiente. De esta forma, se garantiza una comunicación fluida y una gestión eficaz de los datos en la tarjeta STM32F4 [(*STM32 Microcontrollers - STMicro | Mouser*, s. f.)].
- Convertidor analógico-digital (ADC): la tarjeta STM32F4 está equipada con múltiples periféricos ADC (Conversor Analógico-Digital), que posibilitan la medición de señales analógicas y su conversión en valores digitales, los cuales pueden ser procesados por el microcontrolador. Gracias a estos periféricos ADC, la tarjeta STM32F4 es capaz de adquirir y procesar señales analógicas, lo que la hace una herramienta valiosa en aplicaciones que requieren el procesamiento de señales eléctricas analógicas, como en sistemas de control y monitoreo de procesos industriales [(*STM32 Microcontrollers - STMicro | Mouser*, s. f.)]
- Temporizadores: La tarjeta STM32F4 posee diversos periféricos de temporización, los cuales permiten la medición del tiempo y la generación de señales de temporización. Estos periféricos son de gran utilidad en aplicaciones que requieren el control preciso de tiempos, como en sistemas de control de motores y en la automatización de procesos industriales. La tarjeta STM32F4, gracias a estos periféricos, es capaz de proporcionar

una gestión de tiempo eficiente y precisa en sus aplicaciones [(*STM32 Microcontrollers - STMicro | Mouser*, s. f.)].

- Interfaz Ethernet: La tarjeta STM32F4 cuenta con una interfaz Ethernet que facilita la conexión a una red Ethernet. Esta característica resulta muy útil en aplicaciones que requieren conectividad a Internet o comunicación en red. La presencia de la interfaz Ethernet permite a la tarjeta STM32F4 ser una excelente opción para proyectos que necesitan una interacción eficiente y segura con dispositivos o sistemas externos a través de una red Ethernet [(*STM32 Microcontrollers - STMicro | Mouser*, s. f.)].
- Interfaz CAN: La tarjeta STM32F4 posee una interfaz CAN, la cual facilita la comunicación con otros dispositivos electrónicos que utilizan el protocolo CAN. Esta característica resulta muy útil en aplicaciones relacionadas con la industria automotriz, en particular en el control de motores. La capacidad de comunicación con otros dispositivos mediante CAN, permite que la tarjeta STM32F4 sea una excelente opción para proyectos que requieren una interacción eficiente y precisa con dispositivos electrónicos externos. [(*STM32 Microcontrollers - STMicro | Mouser*, s. f.)].

La tarjeta STM32F4 es una herramienta extremadamente valiosa para diseñadores y desarrolladores de sistemas electrónicos, ya que permite la construcción de soluciones personalizadas para diversas aplicaciones. Por ejemplo, la tarjeta STM32F4 es adecuada para aplicaciones de control automático, procesamiento de señales de audio, control de sistemas de iluminación, y muchas otras aplicaciones más. Con la capacidad de programar la tarjeta STM32F4 utilizando diferentes lenguajes de programación, como C/C++ y MATLAB, los diseñadores y desarrolladores tienen una amplia gama de opciones para construir soluciones personalizadas que satisfagan sus necesidades específicas. [(Bharadwaj et al., 2016)].

Periféricos.

La tarjeta tiene algunos periféricos que se indican a continuación. De igual manera, para un mayor detalle puede verse la Figura 2.2 [(Lab, 2019)].

- Puerto de audio
- USB
- Consumo actual
- Acelerómetro
- Puerto de depuración
- Botón e indicadores

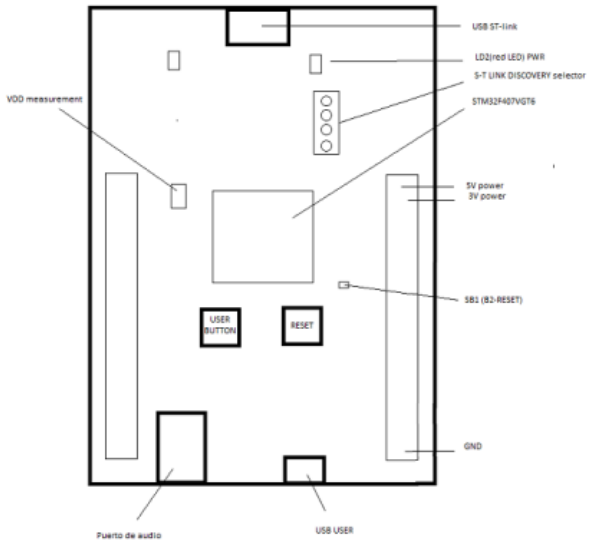


Figura 4.2 Periféricos de STM32F4DISCOVERY [(Lab, 2019)].

2.2 Librería Waijung.

Waijung es una librería gratuita perteneciente que fue creada en Tailandia por la compañía Aimagin (Proaño, 2016). La creación de la librería se enfocó en brindar soporte a diferentes familias de microcontroladores y simplificar su programación mediante bloques de Simulink. Esto permite que el proceso de programación sea más accesible y fácil para los programadores. La librería es compatible con toda la familia de microcontroladores STM32F4 que están disponibles a nivel mundial, ofreciendo así una amplia variedad de opciones para los diseñadores y desarrolladores que buscan soluciones personalizadas para sus proyectos de sistemas electrónicos. Con el uso de esta librería, los programadores pueden interactuar con el microcontrolador de una forma más sencilla y amigable, sin la necesidad de tener conocimientos avanzados de programación, lo que hace que el proceso de desarrollo sea más eficiente y menos complicado.(Mejia, s. f.).

En esta sección se describirá de una manera muy detallada la instalación de la librería, ya que existen diferentes tipos de errores al momento de realizar la compilación con Matlab. Estos errores se fueron corrigiendo en el trascurso de su instalación con el fin de que la librería Waijung pueda ser utilizada adecuadamente.

La descarga de la librería de Waijung se realiza de manera gratuita de la página Aimagin HELPDESK: https://support.aimagin.com/projects/support/wiki/Waijung_2_installation (véase la Figura 2.3)

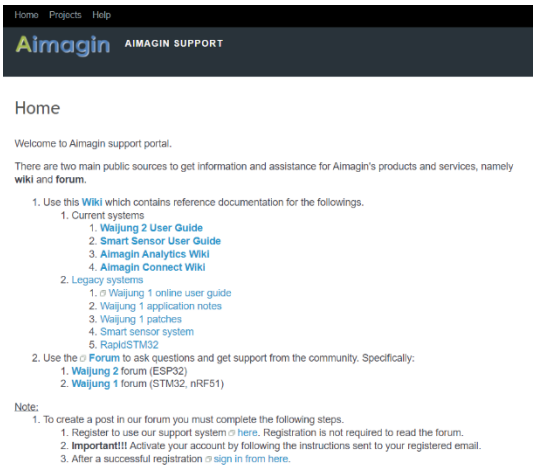


Figura 5.3 Entorno de la página Aimagin [(Aimagin Support, s. f.)].

Una vez dentro de la página se debe buscar la librería Waijung y tras la descarga se debe realizar la instalación utilizando el archivo waijung17_03a (véase en la Figura 2.4).

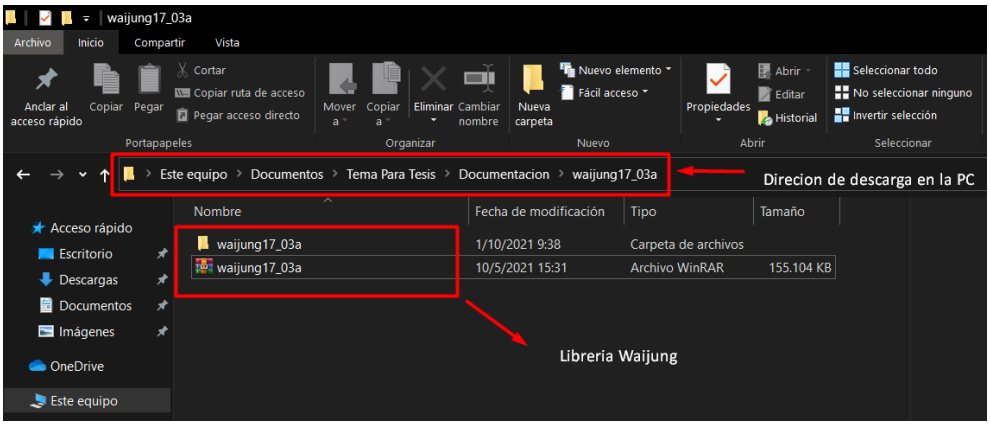


Figura 6.4 Localización de la librería [(Ilustración por Autores)].

2.2.1 Instalación de la librería Waijung en MATLAB R2020b.

Una vez descargada la librería se debe abrir Matlab. Para este caso se ha utilizado la versión R2020b. Este programa se encuentra en la página de MathWorks que es su página ofician (link para la descarga https://la.mathworks.com/products/matlab.html).

Una vez que se ha abierto Matlab se debe direccionar la carpeta donde se encuentran los archivos de la librería que fueron previamente descargados. (véase en la figura 2.5).

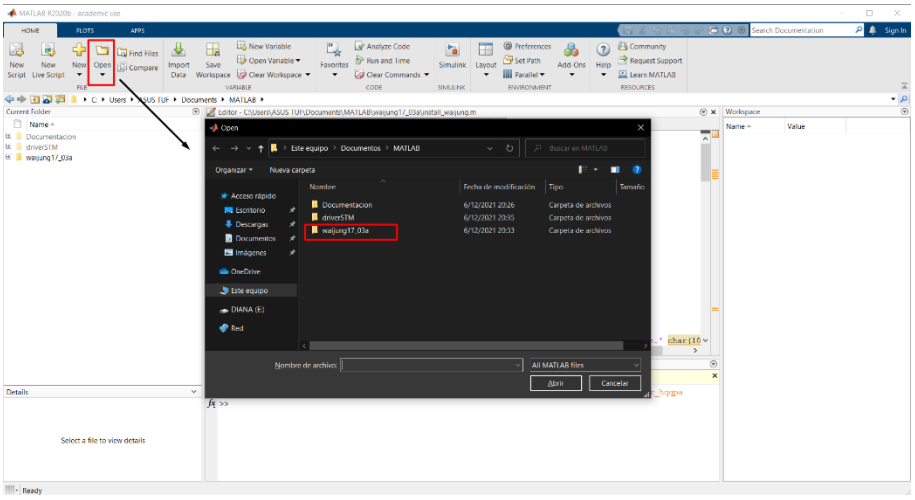


Figura 7.5 Localización de librería en software Matlab [(Ilustración por Autores)].

Dentro de la carpeta waijung17_03 reposará el instalador denominado install_waijung.m. Este archivo posee el código de ejecución para instalar la librería con Matlab (véase las Figura 2.6 y 2.7).

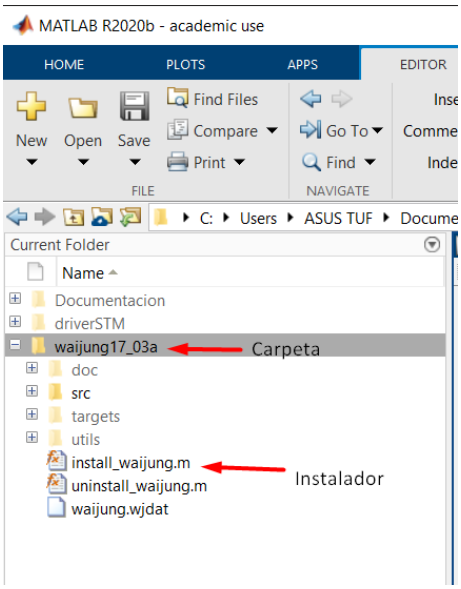


Figura 8.6 Carpeteas en el entorno Matlab [(Ilustración por Autores)].

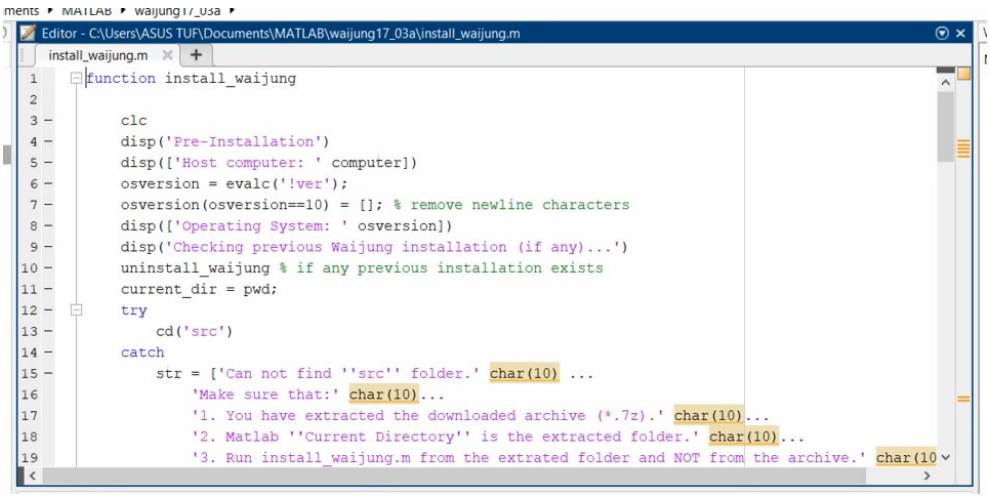


Figura 9.7 Instalación de librería Waijung [(Ilustración por Autores)].

Una vez ejecutado el código se comienza a realizar la instalación de todos los paquetes descomprimidos. Este proceso se podrá observar en la ventana de comandos de Matlab (Véase la Figura 2.8).

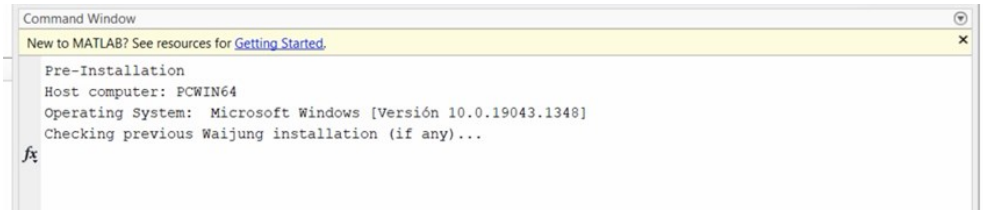


Figura 10.8 Instalación compilada [(Ilustración por Autores)].

Cabe recalcar que en la instalación pueden existir errores. La mayoría de estos errores se solucionan comentando comandos o líneas de código del archivo de instalación. Por ejemplo, el error que se muestra en la Figura 2.9 se soluciona comentando la línea 62. Generalmente, estos son errores de refrescamiento del servidor. La omisión de estas líneas de código no altera el desempeño de la librería (véase la Figura 2.10).

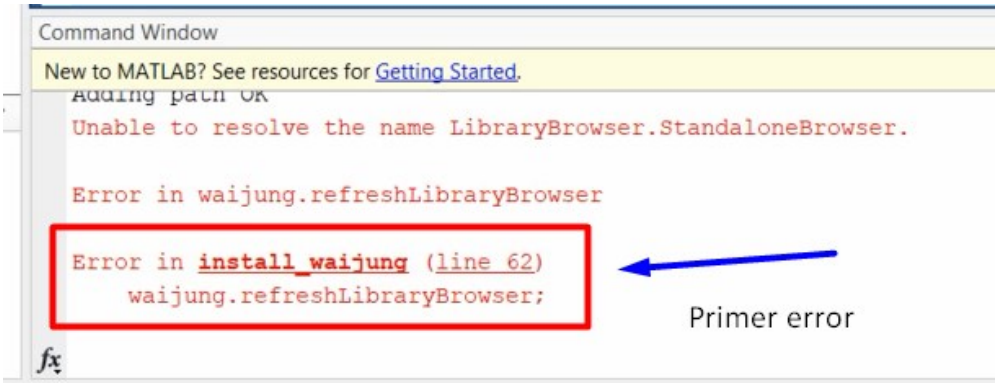


Figura 11.9 Primer Error de compilación [(Ilustración por Autores)].

Una vez corregido un error se debe volver a ejecutar el archivo. En el caso de que vayan apareciendo más errores se debe seguir el mismo procedimiento. Por ejemplo, en la Figura 2.11 se ilustra la corrección de un nuevo error, el cual se soluciona comentando la línea 21 de código.

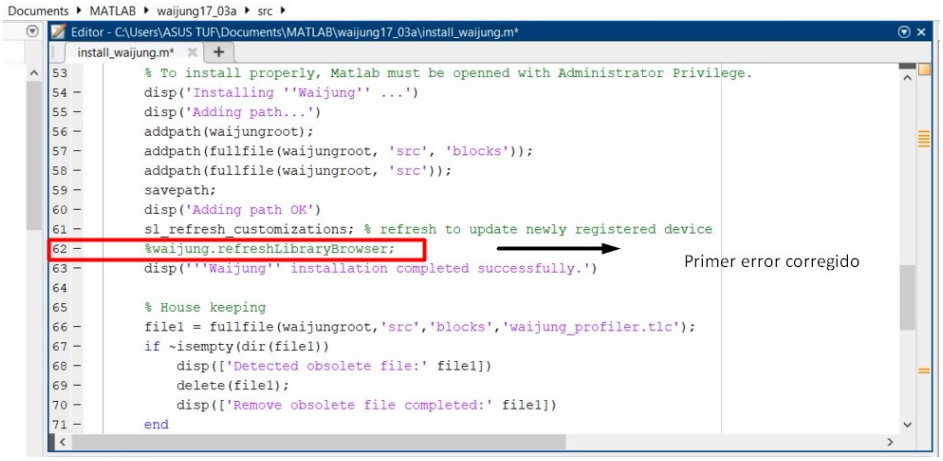


Figura 12.10 Corrección del primer error [(Ilustración por Autores)].

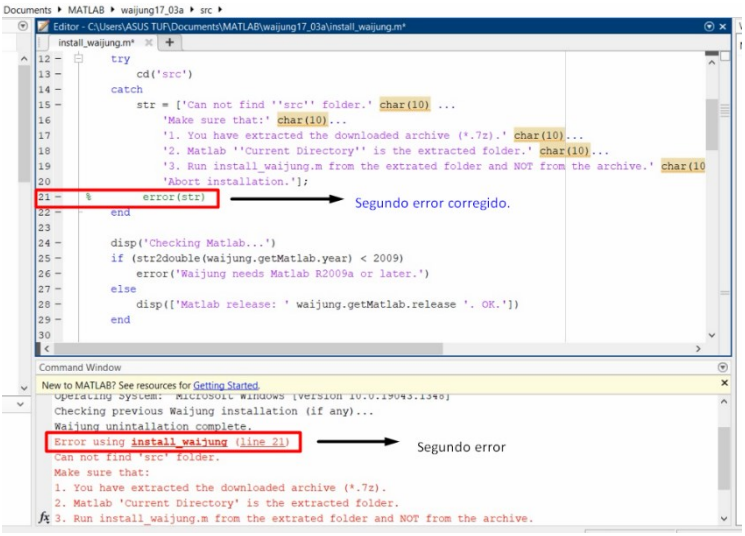


Figura 13.11 Segundo error de compilación y corregido [(Ilustración por Autores)].

A manera de ilustración, en las Figuras 2.11 y 2.12 se muestra la corrección de los siguientes errores que aparecieron en la instalación.

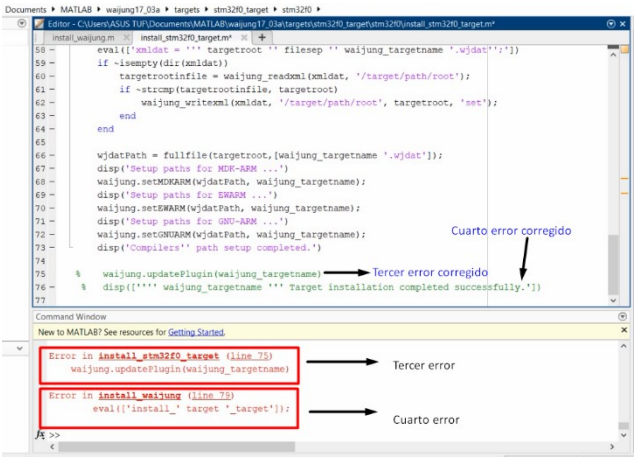


Figura 14.12 Tercer y cuarto error de compilación y corrección [(Ilustración por Autores)].

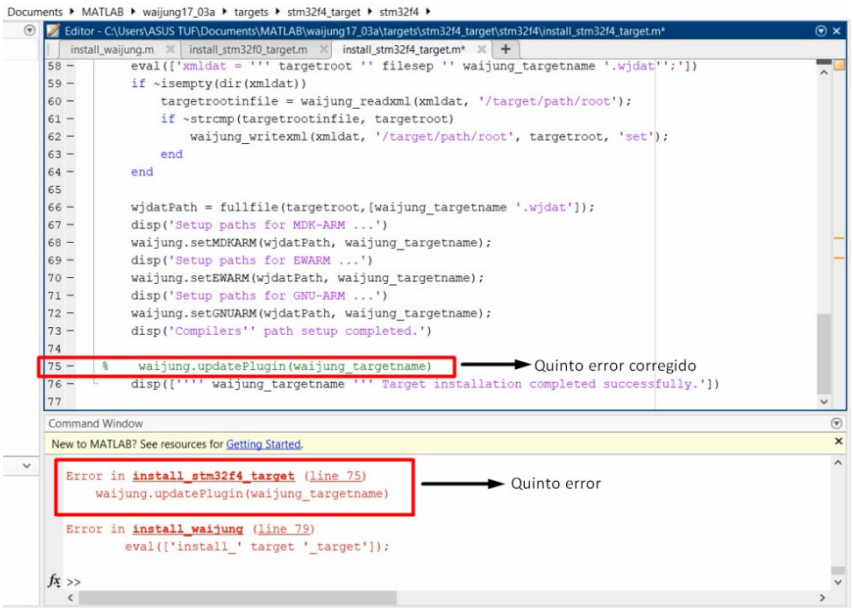


Figura 15.13 Quinto error de compilación [(Ilustración por Autores)].

En la Figura 2.14 se puede observar el mensaje que debe salir una vez que se ha culminado con la instalación de la librería, tras haber depurado todos los errores existentes.

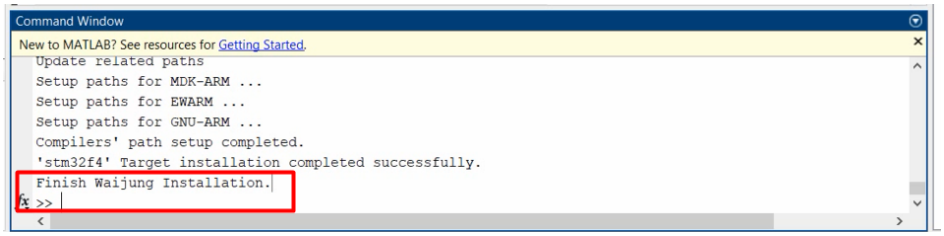


Figura 16.14 Instalación finalizada [(Ilustración por Autores)].

2.2.2 Comprobación de instalación de funcionamiento de la librería Waijung con Simulink Matlab.

Para verificar el funcionamiento de la librería se debe abrir Simulink y se debe crear un nuevo modelo (véase en la figura 2.15).

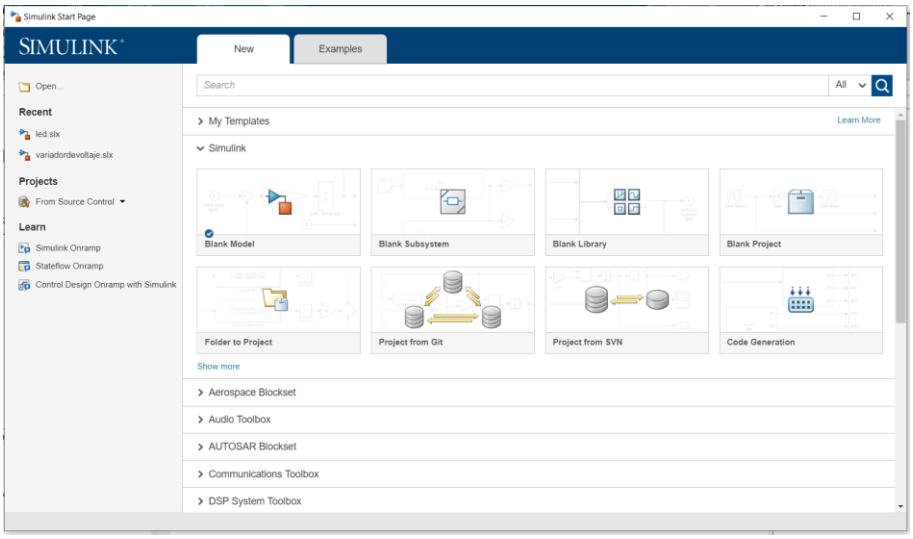


Figura 17.15 Entorno Simulink de Matlab [(Ilustración por Autores)].

Una vez creado el modelo se debe abrir la ventana de "Library Browser". Aquí se mostrará las librerías de Simulink y también estará disponible la opción para agregar nuevas librerías que han sido instaladas anteriormente (véase en la figura 2.16).

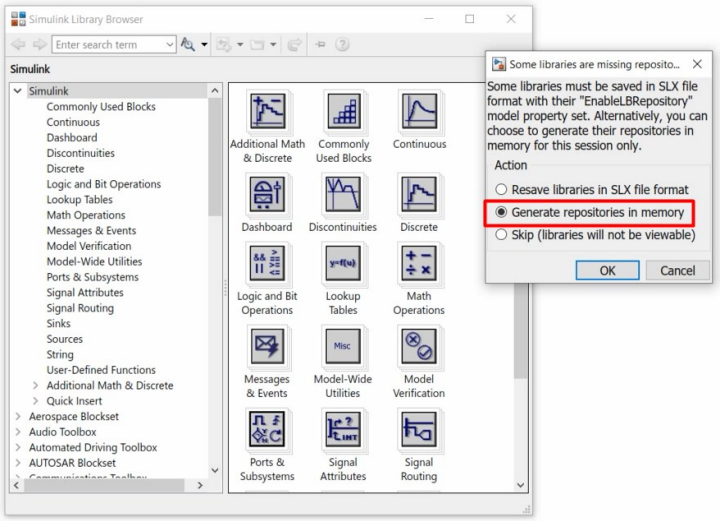


Figura 18.16 Actualización de librería [(Ilustración por Autores)].

En la Figura 2.17 se puede visualizar cómo se ha instalado de manera correcta la Librería Waijung para la tarjeta STM32F407.

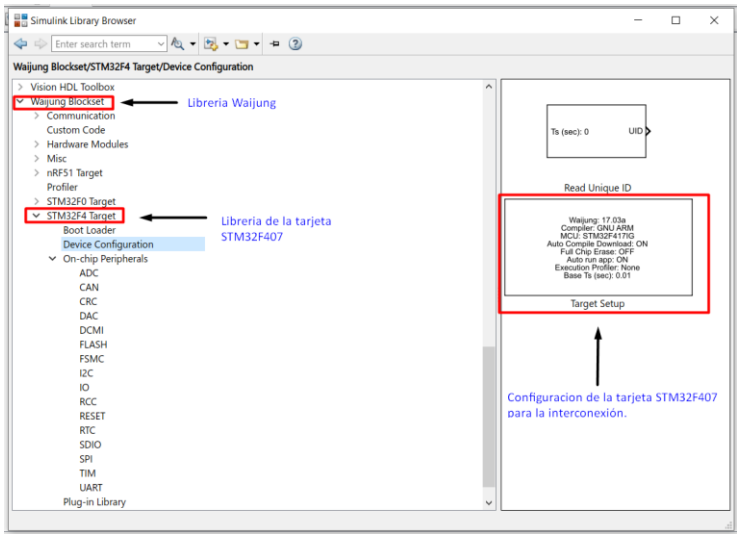


Figura 19.17 Librería de bloques de Simulink (Waijung) [(Ilustración por Autores)].

2.3 Reconocimiento y pruebas de la Tarjeta

Antes de empezar con el desarrollo de los controladores, en este apartado se describe el proceso de verificación de la funcionalidad de la librería. En este caso se realizará una pequeña aplicación para realizar el parpadeo de algunos leds existentes en la tarjeta. Para empezar, se necesita conectar la tarjeta y comprobar que el puerto de la tarjeta esté conectado correctamente. Para esto, es necesario dirigirnos al administrador de dispositivos y verificar el apartado de puertos (COM Y LPT) (véase en la figura 2.18).

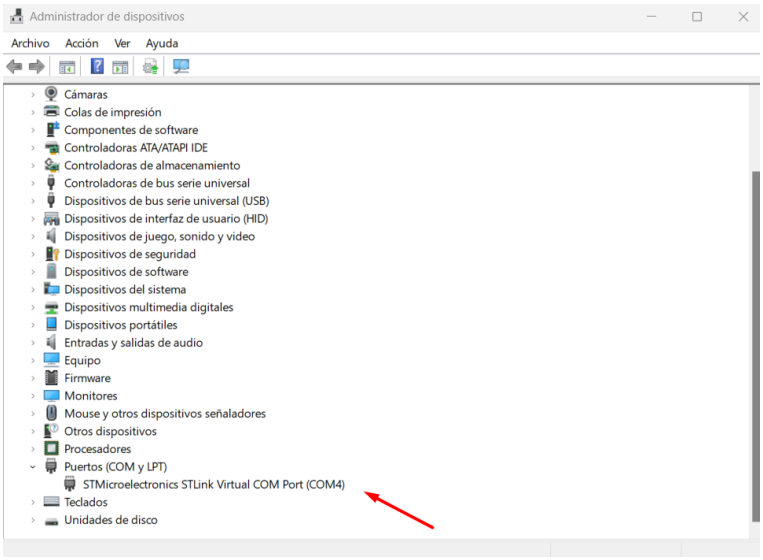


Figura 20.18 Conexión del puerto de la Tarjeta [(Ilustración por Autores)].

Una vez comprobada la conexión abrimos Matlab y cargamos Simulink para empezar con la configuración. A continuación, se debe abrir el apartado de las librerías de simulink y se debe

seleccionar la librería Waijung Blockset (véase en la figura 2.19), aquí encontraremos los bloques de configuración de la tarjeta.

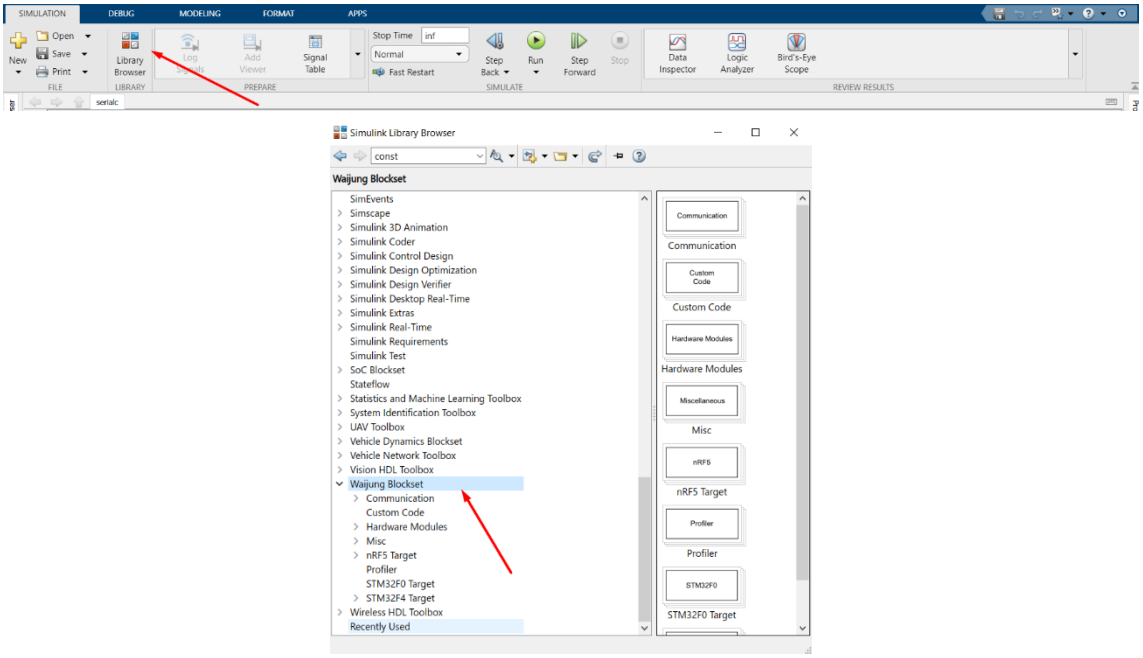


Figura 21.19 Waijung Blockset [(Ilustración por Autores)].

A continuación, se debe seleccionar la tarjeta STM32F4 como se puede ver en la figura 2.20. Cabe indicar que la librería puede también ser utilizada para otra familia de tarjetas.

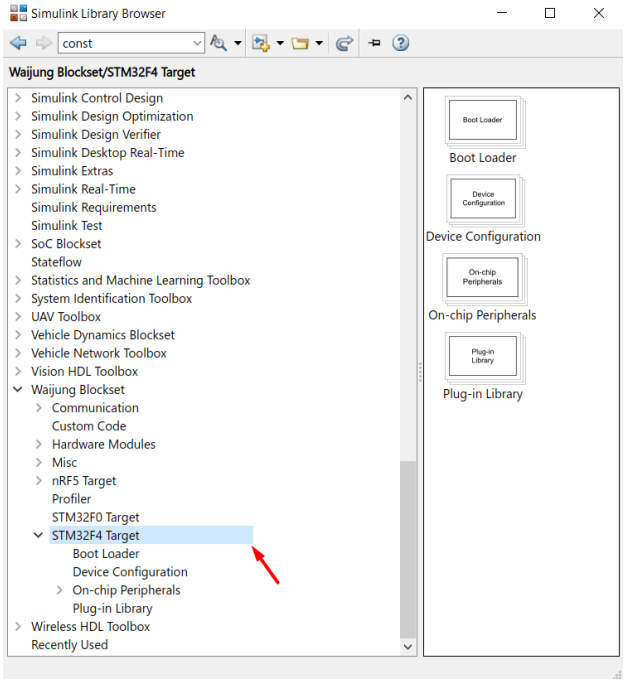


Figura 22.20 Librería STM32F4 [(Ilustración por Autores)].

A continuación, en la pestaña Device Configuration se debe seleccionar el Target Setup, el cual sirve para poder generar el código y grabarlo en la tarjeta (véase en la figura 2.21).

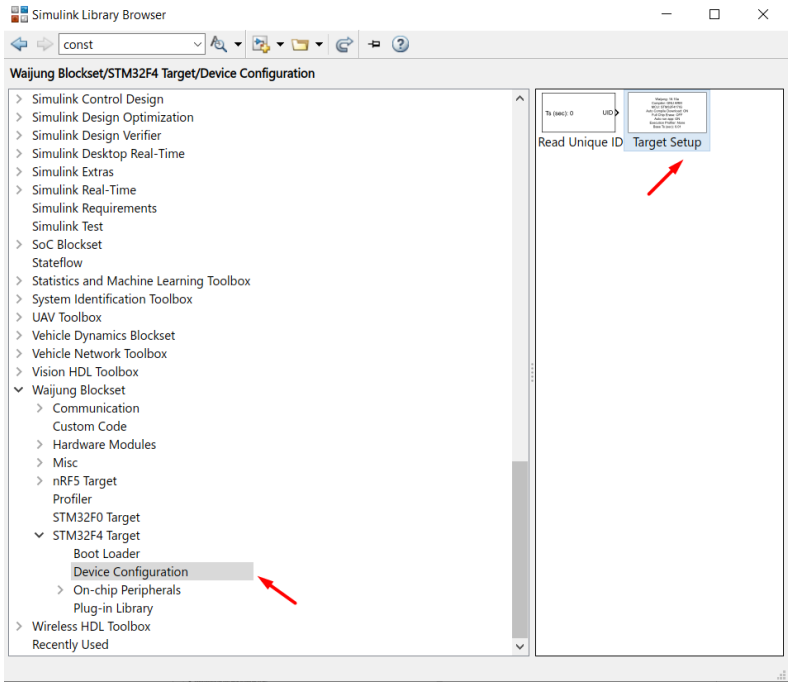


Figura 23.21 Device Configuration [(Ilustración por Autores)].

Una vez elegido el bloque se debe configurar los parámetros eligiendo el tipo de tarjeta y el tipo de compilación siendo lo más importante en este bloque. Esta configuración se ilustra de mejor manera en la Figura 2.22.

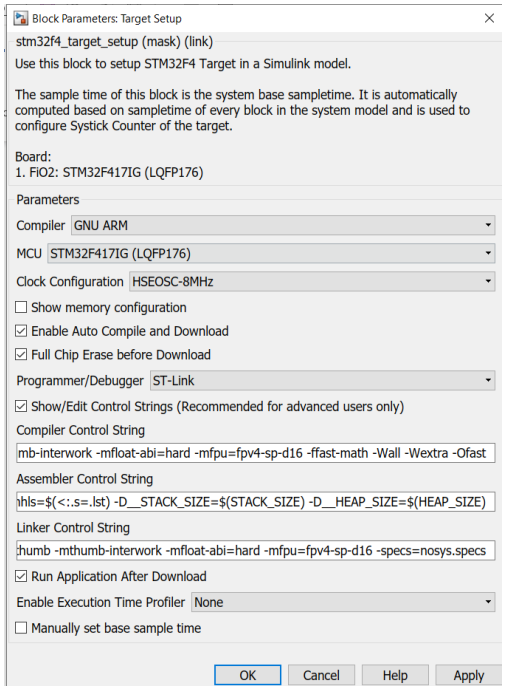


Figura 24.22 Parámetros del bloque Device Configuration [(Ilustración por Autores)].

Para realizar la aplicación de este ejemplo, en la parte de periféricos se debe seleccionar y arrastrar los bloques de entradas y salidas digitales, tal como se muestra en la Figura 2.23.

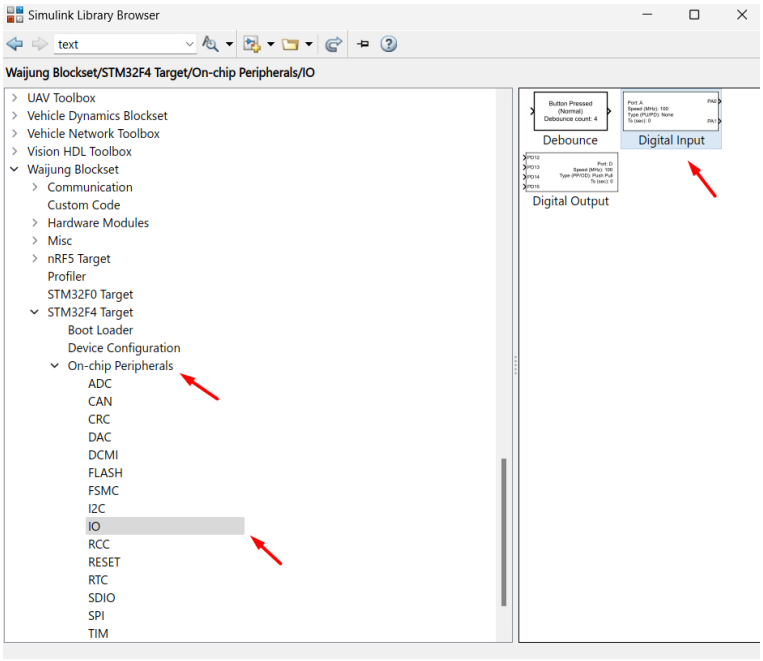


Figura 25.23 Parámetros del bloque Device Configuration [(Ilustración por Autores)].

Para utilizar los leds que vienen integrados en la tarjeta se debe utilizar las salidas PD12, PD13, PD14 y PD15. De igual manera se utilizará el botón integrado de la tarjeta que se encuentra en la entrada PA0 (véase en la figura 2.24).

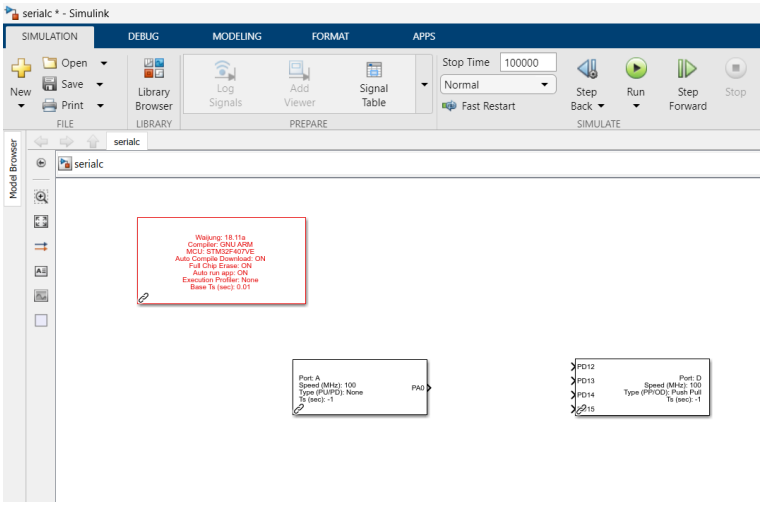


Figura 26.24 Configuración de entradas y salidas de la tarjeta [(Ilustración por Autores)].

Para cumplir con el propósito planteado en este ejemplo, se ha agregado una comparación de igual o diferente de cero para el encendido de los leds de la tarjeta. Esto se ilustra de mejor manera en la figura 2.25.

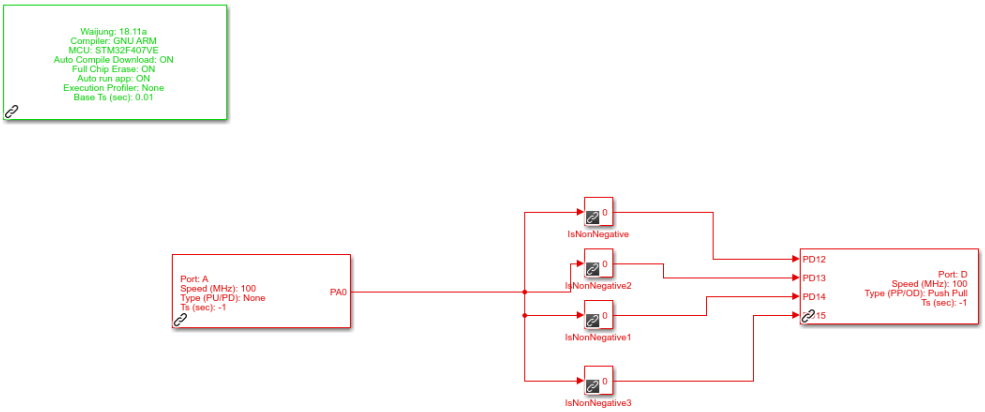


Figura 27.25 Configuración del encendido de los leds [(Ilustración por Autores)].

Una vez terminado el diseño es necesario dirigirse al apartado de apps de Simulink y dar click en la opción Embedded coder, tal como se puede ver en la figura 2.26.

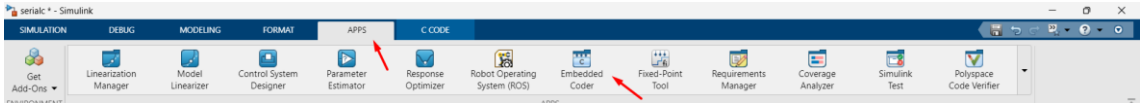


Figura 28.26 Configuración para cargar el programa [(Ilustración por Autores)].

A continuación, se abrirá la pestaña de C CODE. Aquí se debe dar clic en BUILD para cagar el programa diseñado (véase en la figura 2.27).

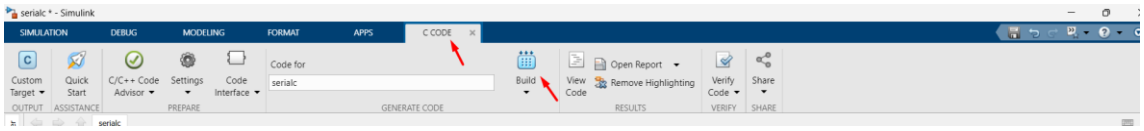


Figura 29.27 Carga del programa diseñado [(Ilustración por Autores)].

Así iniciará un proceso de construcción por etapas, donde todas las etapas deberán estar en color verde como prueba de que no existe ningún error. Este proceso se ilustra de mejor manera en la Figura 2.28.

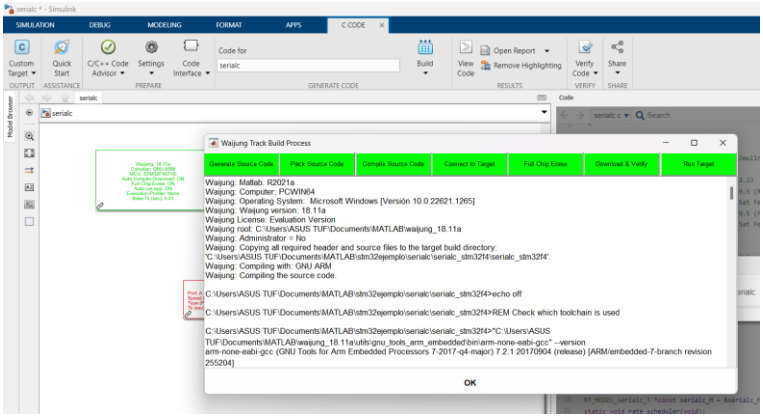


Figura 30.28 Proceso de construcción y cargado del programa diseñado [(Ilustración por Autores)].

Finalmente, se puede probar el funcionamiento de la programación, tal como se ve en la Figura 2.29.

Figura 31.29 Proceso de construcción y cargado del programa diseñado [(Ilustración por Autores)].

Ahora se probará el funcionamiento del ADC de la tarjeta. Para esto se realizará el control de encendido de los leds utilizando una señal de entrada analógica. Para esta aplicación se debe seguir el mismo procedimiento que se realizó anteriormente, únicamente la diferencia radica en que ahora se debe incorporar al proyecto el ADC (véase en la figura 2.30).

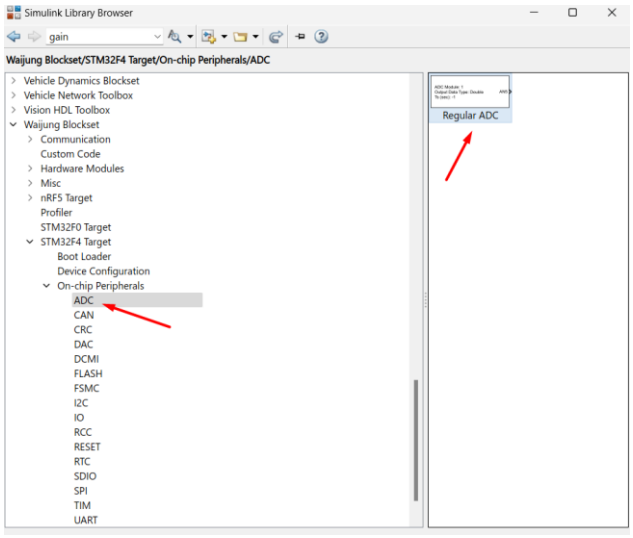


Figura 32.30 Parámetros del bloque ADC – REGULAR ADC [(Ilustración por Autores)].

Al igual que en la aplicación anterior se utilizarán las salidas PD12, PD13, PD14 y PD15. Por otra parte, para el control de encendido de la secuencia de luces led se hace uso del ADC. En complemento, también se agrega una ganancia ya que la tarjeta trabaja a 3.3V y finalmente se agrega unos comparadores a 0.5V a 1V, 1.5V y 2V (véase en la figura 2.31).

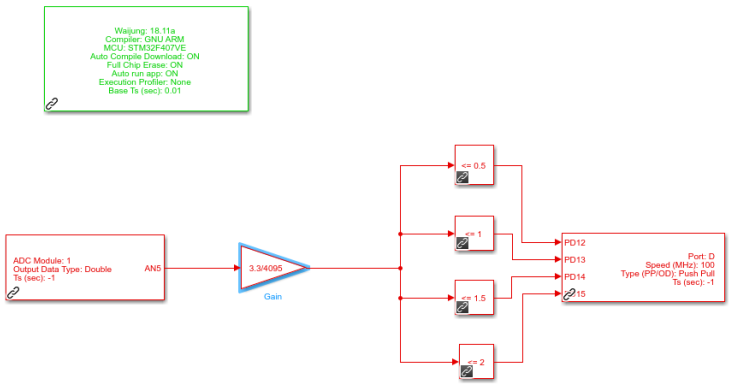


Figura 33.31 Parámetros del bloque ADC – REGULAR ADC [(Ilustración por Autores)].

Una vez terminado el diseño en el apartado de apps de Simulink damos clic en la opción Embedded coder y a continuación en la pestaña C CODE seleccionamos Build para cargar el programa, tal como se explicó en la aplicación anterior.

El funcionamiento del ADC se comprobará mediante un potenciómetro, el cual está conectado al PIN AN5 (véase la Figura 2.32). El funcionamiento del circuito depende del voltaje que proporcione el potenciómetro de 0v a 3.3v. De pendiendo de este voltaje se irán encendiendo los leds correspondientes de acuerdo a la comparación.

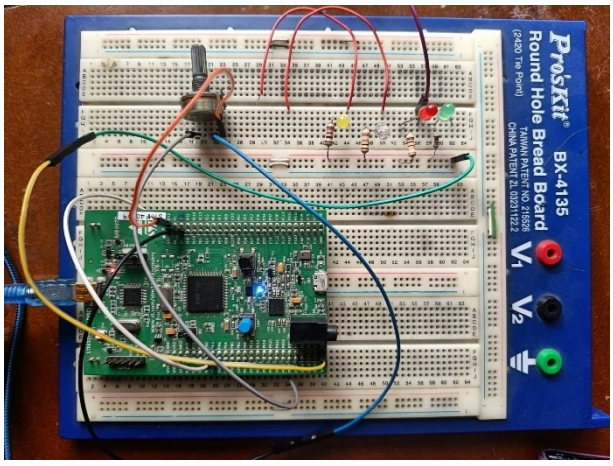


Figura 34.32 Funcionamiento de circuito con el ADC [(Ilustración por Autores)].

Capítulo 3

3 Diseño e Implantación de los controladores.

3.1 Introducción.

En el presente capítulo se presenta el diseño y la implementación de los controladores PID y espacio de estados. El diseño parte de la obtención del modelo del proceso térmico. Para este caso, para la obtención del modelo se ha utilizado un sistema de adquisición de datos mediante la tarjeta STM32F407 la cual opera a través de una interfaz de conexión con Simulink de Matlab.

3.2 Adquisición de datos con la tarjeta Discovery STM32F407.

Para la adquisición de datos del proceso se ha creado una interfaz de conexión en simulink juntamente con la librería Waijung. A través de este instrumento virtual se ha logrado la adquisición de la temperatura de la cámara, para poder obtener la función de transferencia del sistema.

Para el diseño de esta interfaz (véase la Figura 3.1) se han utilizado los bloques de entradas (ADC) y salida analógica (DAC), para obtener la temperatura de la cámara y también para enviar las señales de actuación hacia el proceso. Además, en el diseño existen diferentes bloques matemáticos los cuales sirven para hacer la transformación a unidades de ingeniería de los voltajes que suministran los transmisores de temperatura.

También se han utilizado bloques para construir la parte gráfica del diseño. Para la obtención de datos se utilizó los bloques de configuración del puerto serial (TTL), bloque de configuración UART TX y bloque configuración Host Serial RX. Por otra parte, los datos del proceso son almacenados mediante el bloque To Workspace. Dado que el diseño trabajará en tiempo real, se ha configurado el tiempo de muestreo a 0.01s tanto en los bloques de configuración del sistema y los parámetros de simulación, así obteniendo la interfaz de comunicación donde al momento de la simulación aparecerá la gráfica de la señal de excitación y la temperatura de la cámara. Cabe mencionar que los bloques To Workspace enviarán los datos del proceso directamente al workspace de Matlab.

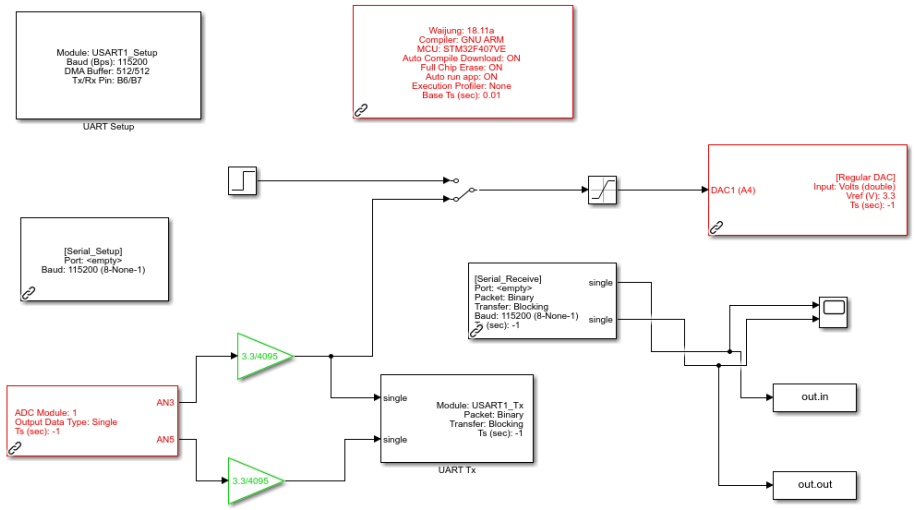


Figura 35.1 Interfaz para la obtención de datos del proceso [(Ilustración por Autores)].

3.2.1 Obtención y simulación de la función de transferencia del proceso térmico.

Una vez implementada la interfaz para la adquisición, los datos son capturados y guardados directamente en el Workspace de Matlab. Por esta razón se debe entrar al command Window y con la ayuda de los comandos, que se pueden apreciar en la figura 3.2. Se procede a guardar los datos para la estimación de la función de transferencia del proceso.

```
tiempo=out.simentrada.Time
entrada=out.simentrada.Data
salida=out.simsalida.Data
save disPID2 tiempo entrada salida
```

Figura 36.2 Almacenamiento de datos para la identificación del sistema. [(Ilustración por Autores)].

Una vez almacenado los datos se puede realizar la identificación del modelo. En este caso se utilizará Matlab para obtener la función de transferencia del proceso térmico.

Para realizar la identificación se ha llevado el proceso a un punto de operación inyectando una señal de voltaje de 0.75 V. Con este voltaje la temperatura se ha elevado a 18.75°C, hasta que el sistema alcance el régimen permanente en donde se ha suspendido la captura de datos. En la Figura 3.3 se muestra los datos de entrada y salida que han sido capturados y graficados con Matlab. Cabe indicar que el voltaje enviado al proceso actúa sobre la niquelina, lo cual hace que se eleve la temperatura. También se debe indicar que la identificación de datos se realiza considerando una señal de voltaje en la salida del proceso. Este es el voltaje que entrega el sensor de temperatura. La constante de conversión de voltaje a temperatura es de 10.

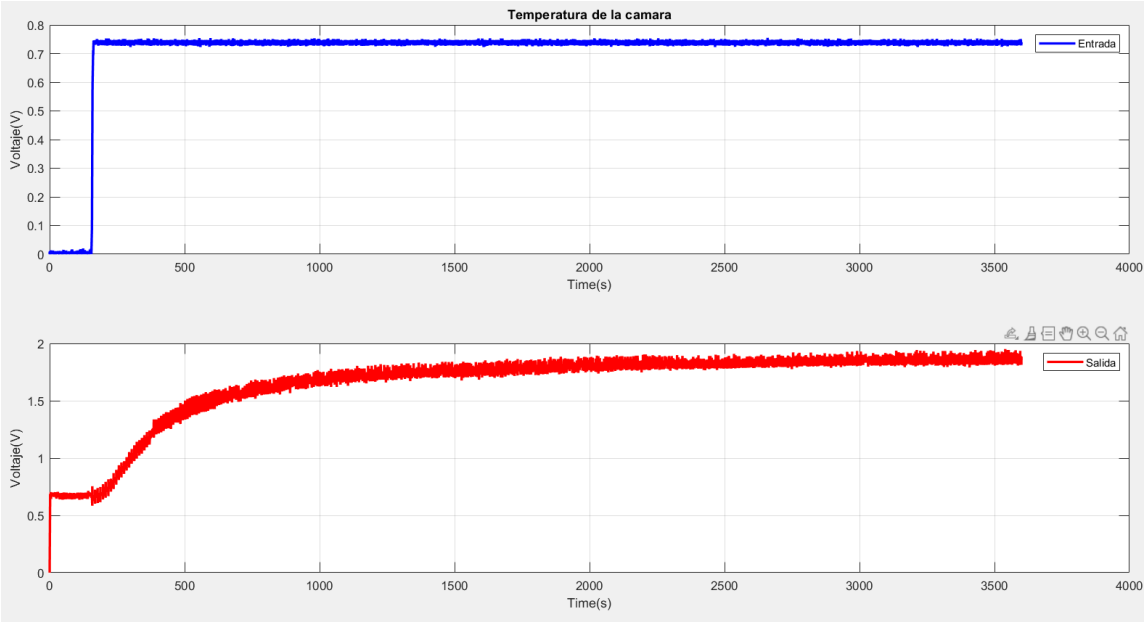


Figura 37.3 Datos obtenidos para la identificación del modelo del proceso térmico [(Ilustración por Autores)].

Para identificar el modelo se selecciona los datos eliminando las 1000 primeras muestras (véase en la figura 3.4).

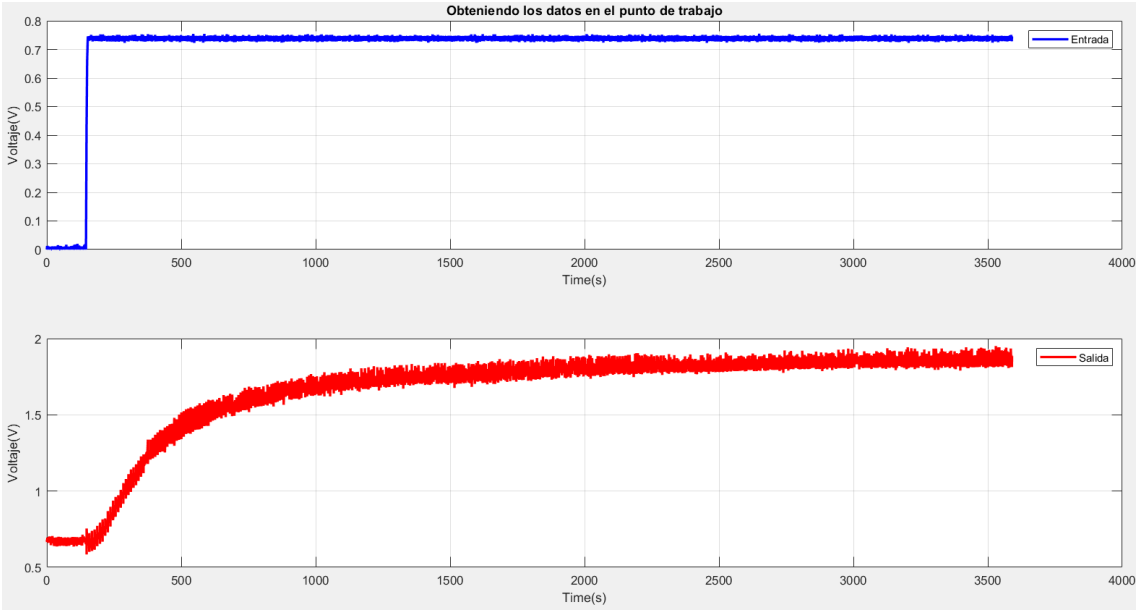


Figura 38.4 Selección de datos para la identificación [(Ilustración por Autores)].

El siguiente paso por seguir, es eliminar el offset ya que lo importante aquí seria la parte dinámica del sistema (véase en la figura 3.5).

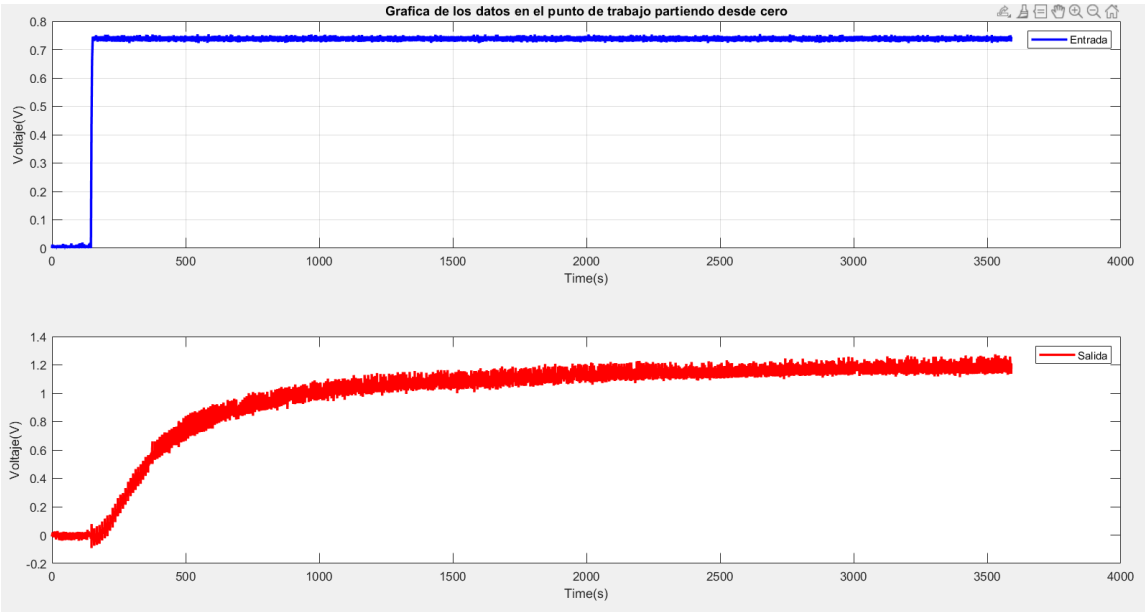


Figura 39.5 Gráfica de datos eliminado el offset [(Ilustración por Autores)].

Con los datos de la Figura 3.6 se realizará la identificación del sistema. Para este caso se hará una estimación con modelos de un polo, dos polos, dos polos y un cero y tres polos y un cero con el objetivo de seleccionar el modelo que más similitud tenga con la planta (véase en la figura 3.6).

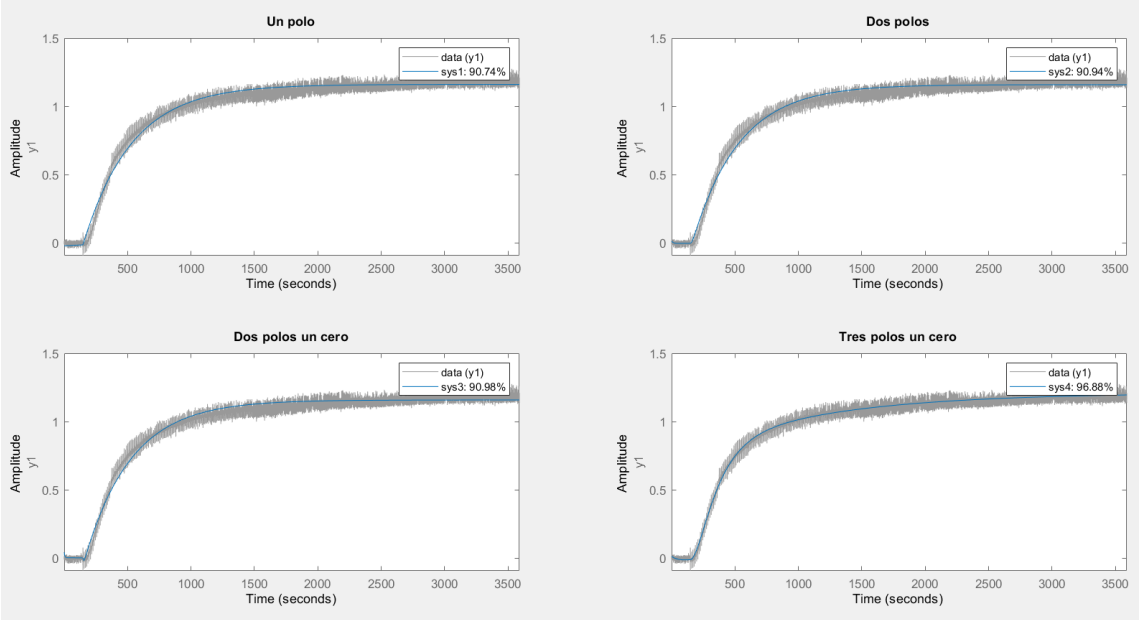


Figura 40.6 Selección del modelo con más similitud con la planta [(Ilustración por Autores)].

En la Figura 3.6 se puede observar que el sistema que tiene más porcentaje de similitud con la planta es el de tres polos y un cero. La respuesta al escalón del sistema y la distribución de polos y ceros se muestra en la Figura 3.7.

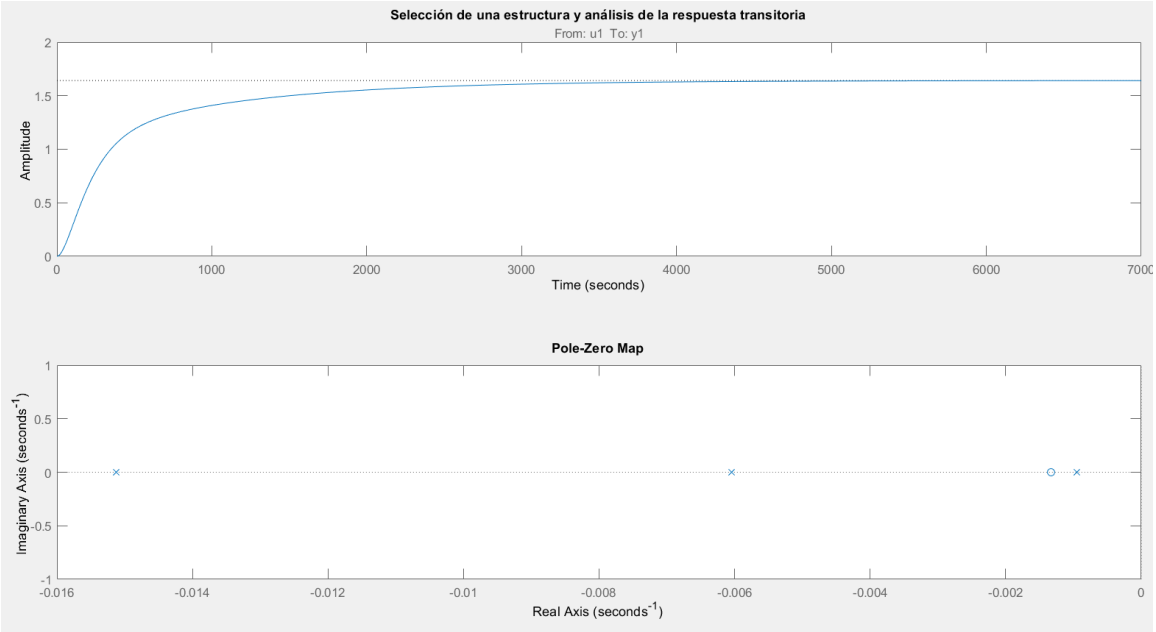


Figura 41.7 Análisis a la respuesta transitoria y ubicación de los polos y ceros [(Ilustración por Autores)].

Los parámetros característicos del sistema se muestran en la Figura 3.8.

```
struct with fields:

      RiseTime: 1.2804e+03
  SettlingTime: 3.0369e+03
   SettlingMin: 1.4781
   SettlingMax: 1.6416
     Overshoot: 0
    Undershoot: 0
          Peak: 1.6416
      PeakTime: 8.0328e+03
```

Figura 42.8 Parámetros característicos de nuestro sistema [(Ilustración por Autores)].

La función de transferencia que se ajusta más a la planta del proceso térmico es la siguiente:

$$sys4 = \frac{0.0001074s + 1.429e^{-07}}{s^3 + 0.02212s^2 + 0.0001116s + 8.703e^{-08}} \quad (1)$$

3.3 Diseño del controlador PID

El diseño del controlador PID parte del modelo de tres polos y un cero obtenido en el apartado anterior. Para este caso se utilizará la herramienta de sintonía de controladores PID que posee Matlab. Cabe indicar que el controlador se implementará en tiempo discreto utilizando la siguiente ecuación:

$$\frac{u(z)}{e(z)} = \frac{1}{1-z^{-1}} * \left(kp + ki * T + \frac{Kd}{T}\right) + \frac{z^{-1}}{1-z^{-1}} * \left(-2\frac{Kd}{T} - kp\right) + \frac{z^{-2}}{1-z^{-1}} * \left(\frac{Kd}{T}\right)$$

(2)

La ecuación discreta del PID se divide en la parte proporcional, la parte integral y la parte derivativa, las cuales se multiplican a las ganancias K1, K2 y K3, respectivamente. En (2) z representa el operador en tiempo discreto discreto. La parte proporcional (Kp) del controlador se especifica a través de (3), mientras que la parte integral (Ki) y la parte derivativa (Kd) se especifican a través de (4) y (5), respectivamente.

$$\frac{\mathbf{1}}{\mathbf{1}-\boldsymbol{z}^{-\mathbf{1}}} * \left(kp + ki * T + \frac{Kd}{T}\right) \qquad K1 = \left(kp + ki * T + \frac{Kd}{T}\right) \tag{3}$$

$$\frac{\boldsymbol{z}^{-\mathbf{1}}}{\mathbf{1}-\boldsymbol{z}^{-\mathbf{1}}} * \left(-2\frac{Kd}{T} - kp\right) \qquad K2 = \left(-2\frac{Kd}{T} - kp\right) \tag{4}$$

$$\frac{\boldsymbol{z}^{-\mathbf{2}}}{\mathbf{1}-\boldsymbol{z}^{-\mathbf{1}}} * \left(\frac{Kd}{T}\right) \qquad K3 = \left(\frac{Kd}{T}\right) \tag{5}$$

El controlador se ha diseñado para cumplir con las especificaciones que se muestran la Tabla 1.

Tabla 1. Parámetros de la respuesta al impulso de la función de transferencia

PARAMETRO	VALOR
Tiempo de subida (Tu)	367 s
Tiempo de establecimiento (Ts)	<600 s
Pico máximo (Mp)	1.02

Nota Fuente: Table por Autores.

Se hace uso de la aplicación del Matlab (PID Tuner) para la sintonía del controlador, donde se pueden observar las restricciones temporales que impone las condiciones de diseño.

Cabe mencionar que para el diseño del controlador se eligió un controlador PI, La acción derivativa no será efectiva debido a diversos factores. En primer lugar, el proceso térmico tiene cierta cantidad de ruido y la acción derivativa siempre aumenta el ruido ya que es tomado como error. Además, el proceso térmico es un sistema dinámico lento, lo que hace que el uso de la acción derivativa no sea factible. Si el sistema se mueve hacia el punto de referencia a una velocidad considerable, la inercia del sistema puede causar que se pase de largo sin poder detenerse en el objetivo deseado. En resumen, debido a la presencia de ruido y la dinámica del sistema térmico, la acción derivativa no es una opción viable para el control de este proceso.

En la Tabla 2 se indican los valores que nos brinda la sintonía del controlador mediante el PI tuner.

Tabla 2. Constantes obtenidas para el controlador PI

PARAMETRO	VALOR
Ki	0.0033156
Kp	0.78727

Nota Fuente: Table por Autores.

Con los valores de las constantes ya calculadas se puede desarrollar en simulink el controlador PI. Para el diseño empezamos con los bloques de configuración de la tarjeta STM32F4 haciendo uso del bloque Target Setup, el cual ayuda a elegir la serie de la tarjeta que se utilizará. También se utilizarán los bloques: UART SETUP, HOST SERIAL PORT, UART Tx y UART Rx, los cuales sirven para la conexión del puerto USB y obtención de datos para poder graficar las variables de interés. En la figura 3.9 se puede ver el diagrama de conexión del controlador PID. También se ha agregado el ADC para el control de la referencia y alimentación de la resistencia calefactora. Además, se ha agregado el bloque Discrete Controller PID en el cual ingresaremos los valores calculados de Kp y Ki.

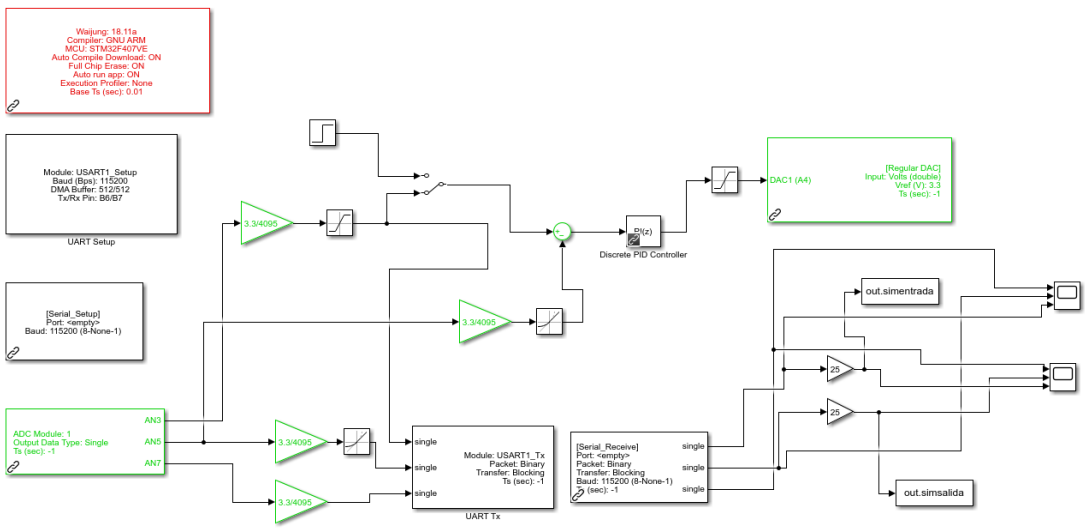


Figura 43.9 Controlador PID [(Ilustración por Autores)].

Una vez desarrollada la interfaz del controlador PI, nos dirigimos a las aplicaciones de Matlab y buscamos el icono de C CODE. A continuación, seleccionamos la opción de build para generar el código que se grabará en la tarjeta (véase en la figura 3.10).

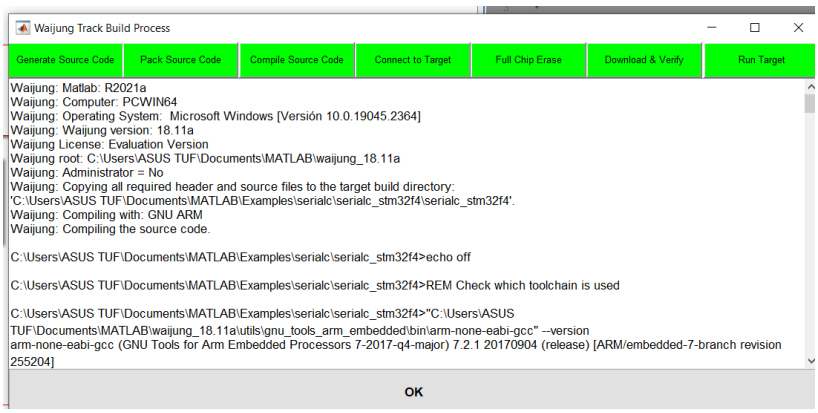


Figura 44.10 Generación del código para la tarjeta STM [(Ilustración por Autores)].

Para la prueba se ha establecido una temperatura de referencia (set point) de 22°C hasta que llegue al tiempo de establecimiento. Una vez alcanzada la referencia se le cambia el punto de ajuste (set point) a 25°C, alejándonos un poco del punto de operación donde fue linealizada y estimada la función de transferencia. En la Figura 3.11 se muestra el comportamiento del sistema reflejado a través de la señal controlada y la acción de control. Aquí vale la pena destacar que gracias a la variedad de periféricos que tiene la tarjeta, el punto de ajuste puede ser modificado durante el funcionamiento del sistema mediante un ADC.

En lazo abierto el sistema térmico se estabilizaba en un tiempo de 3036.9s equivalente a 50.615 minutos. Gracias al controlador se mejora bastante este tiempo y ahora el tiempo se reduce 600s equivalente a 10 minutos.

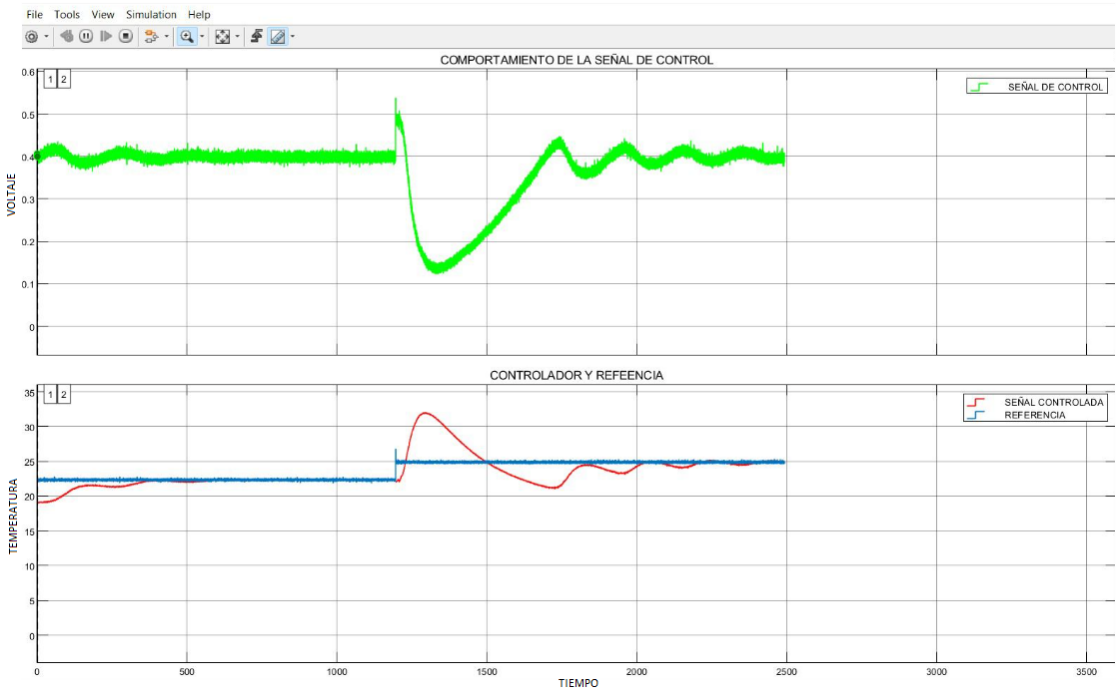


Figura 45.11 Proceso térmico a 22°C y cambio a 25°C [(Ilustración por Autores)].

En el segundo escalón se logra evidenciar los efectos significativos que tiene la no linealidad del proceso. A medida que nos alejamos del punto de operación el controlador empeora notándose una oscilación considerable. Esto deja ver que la ganancia estática del sistema es mayor mientras sube la temperatura. Esta ganancia estática se multiplica con la ganancia proporcional del controlador y hace que este sea muy agresivo.

3.4 Diseño del controlador por Espacio de Estados

Al igual que en el caso del controlador PID, el controlador en espacio de estados se diseña a partir del modelo identificado del proceso, considerando que se debe discretizar para la implementación.

Entonces se partirá de la función de transferencia calculada y la declararemos con los comandos de la figura 3.12.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Ingreso de la funcion de transferecia
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

NUM=[0.0003113];
DEN=[1 0.0759 0.0001981];
planta=tf(NUM,DEN);
```

Figura 46.12 Función de trasferencia [(Ilustración por Autores)].

A continuación, para tener una referencia del sistema en lazo abierto, se obtienen los valores característicos de la FT en el dominio del tiempo continuo (véase en la figura 3.14).

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% obtencion tiempo de muestreo
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[wn,z,p]=damp(planta);%obtener la frecuencianatural
Wmax=max(wn);
aa=wn;
w=(10*Wmax)/(2*pi);%para pasar lafrecuencia natural en Hz
Ts=1/w;
f=1/Ts;

roots(DEN);
```

Figura 47.13 Valores FT en dominio [(Ilustración por Autores)].

Para iniciar con el diseño del controlador en espacio de estados, se obtienen las matrices A, B, C, D del sistema en la forma canónica controlable. (véase en la figura 3.14).

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%sistema en la forma canónica controlable
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[NUM,DEN]=tfdata(planta,'v');
[A,B,C,D]=tf2ss(NUM,DEN);
A=rot90(A,2); %esto lleva a la forma canonica controlable
B=rot90(B,2);
C=rot90(C,2);
D=rot90(D,2);
```

Figura 48.14 Matrices en dominio S [(Ilustración por Autores)].

Con la forma canónica controlable se podrá hacer el análisis de la Controlabilidad y Observabilidad. (véase en la figura 3.15).

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%analisis de controlabilidad y observalidad
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Ct=ctrb(A,B);
if rank(Ct)==size(A,2)   %size es el tamaño de la matriz
                         %matriz A es de 3x3 el sist es de 3er orden
    disp('El sistema es controlable')
else
    disp('El sistema no es controlable')
end
Ob=obsv(A,C);
if rank(Ob)==size(A,2)   %size es el tamaño de la matriz
                         %matriz A es de 3x3 el sist es de 3er orden
    disp('El sistema es observable')
else
    disp('El sistema no es observable')
end
```

Figura 49.15 Matrices en dominio S [(Ilustración por Autores)].

El controlador se ha diseñado para cumplir con las especificaciones que se muestran la Tabla 3. Cabe indicar que no existe el propósito de comparar el desempeño de los controladores PID y espacio de estados. Por esta razón no se asumen condiciones de diseño iguales. La intención principal de esta investigación es analizar la factibilidad de implementación de ambos controladores sobre la tarjeta.

Tabla 3. Parámetros de la respuesta al impulso de la función de transferencia

PARAMETRO	VALOR
Factor de amortiguamiento (ζ)	1
Tiempo de establecimiento (Ts)	1000 s

Nota Fuente: Table por Autores.

A continuación, declaramos los polos del observador, integrador y polos deseados para la respuesta del sistema (véase en la figura 3.16).

```
P1=[(exp(-0.0016*Ts)) (exp(-0.0011*Ts))];% polos observador

P2=[(exp((-0.465+j*0.4730)*Ts)) (exp((-0.465-j*0.4730)*Ts)) exp(-50*Ts)];% polos integrador

P3= -0.0280; ;% Polos deseados 1
P4= -0.010 ;% Polos deseados 2

pz=[exp(Ts*P3) exp(Ts*P4)]; % polos plano Z
```

Figura 50.16 Declaración de los polos deseados para la respuesta del sistema [(Ilustración por Autores)].

El siguiente paso será discretizar nuestra función de transferencia, se puede apreciar en la figura 3.17.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Discretizacion de la planata y calculos de matrices aumentadas
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[A,B,C,D]=tf2ss(NUM,DEN); % Matrices de la F.T.
[G,H,C1,J]=c2dm (A,B,C,D,Ts,'tustin');  % Matrices de la F.T.D.
Gi=[G H; zeros(1,3)];  % Matriz G  Aumentada.
Hi=[0;0;1]; % Matriz H  Aumentada.
```

Figura 51.17 Discretización de la planta y obtención de matrices aumentadas [(Ilustración por Autores)].

Seguidamente se calcula el valor de la matriz de ganancias K utilizando el método de Ackerman. Este método se puede utilizar en MATLAB directamente con el comando acker (véase en la figura 3.18). Los valores obtenidos son:

K= (-0.0323 ,0.0001).

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculo de constante k
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('k=');
k=acker(G,H,pz); % ktte. controlada
disp(k);
```

Figura 52.18 Plano Z [(Ilustración por Autores)].

Con el cálculo de la matriz de ganancia k se puede graficar la función de transferencia en lazo abierto y lazo cerrado como se puede ver en las figuras 3.19 y 3.20. Esta comparación se ha realizado para demostrar que el controlador mejorará la respuesta del sistema en lazo cerrado. Como se puede observar en la figura 3.20, el sistema con controlador no tiene error en régimen permanente y el tiempo de establecimiento es mucho menor. Es decir, la respuesta del sistema será más rápida.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Grafica en lazo abierto y cerrado
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Alc=A-B*k;
figure(1)
step(Alc,B,C,D),hold on
step(A,B,C,D),hold on
```

Figura 53.19 Comandos para graficar la función de transferencia

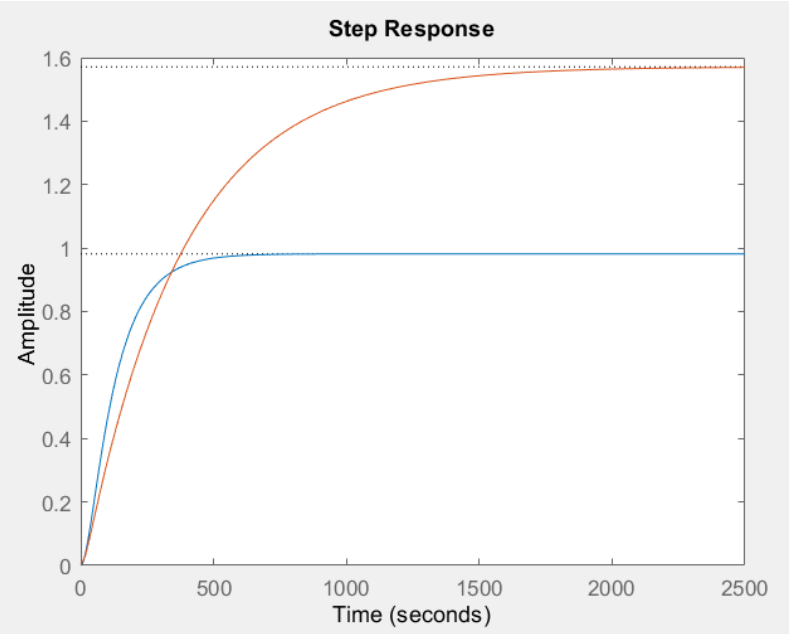


Figura 54.20 Función de transferencia.

Finalmente se realiza el cálculo de las constantes del observador (Ke1 y K1) (véase en las figuras 3.21 y 3.22). Los valores que se obtuvieron son los siguientes:

Ke1= (-0.073; -0.0002)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculo de constante del Observador
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Ke= acker(G,H,P1); % ktte. Observador
disp('Ke=');
Ke1=Ke'; % ktte. Observador
disp(Ke1);
```

Figura 55.21 Constante Ke1.

K1= (0.1951 0.0271 59.2708)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculo de constante del Seguidor
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Ki=acker(Gi,Hi,P2); %ktte. seguidor
disp('K1=');
K1=[Ki+[0 0 1]]*[G-eye(2) H;C*G C*H]^(-1);%ktte. seguidor
disp(K1);
```

Figura 56.22 Constante K1.

La Figura 3.23 muestra el diagrama del controlador en espacio de estados. Los bloques de configuración, adquisición y transferencia de datos son los mismos que se han utilizado en el controlador PID. En los bloques de ganancia del controlador en espacio de estados se han ingresado los valores que se obtuvieron en el diseño.

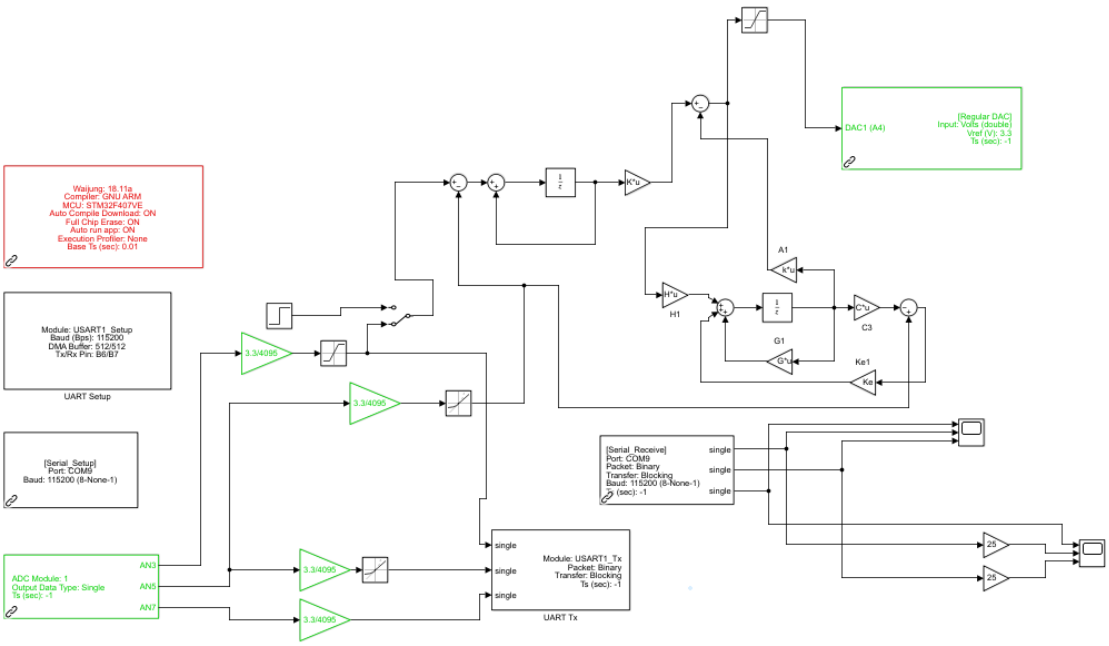


Figura 57.23 Interfaz gráfica del controlador por espacio de estados con observador [(Ilustración por Autores)].

Una vez realizado el diagrama de bloques en Simulink se procede con la generación de código para descargar el controlador a la tarjeta. Para esto nos dirigimos a apps y buscamos el icono de C CODE y le damos a build (véase la figura 3.24).

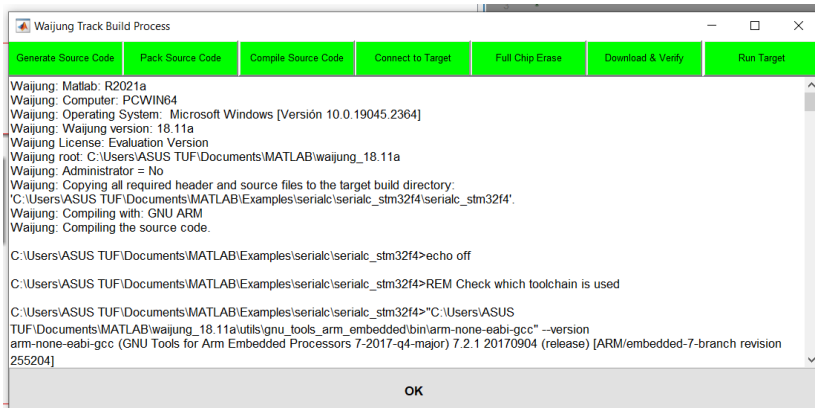


Figura 58.24 Generación del código del controlador en espacio de estados [(Ilustración por Autores)].

En la comprobación del funcionamiento del controlador se establece una temperatura de referencia (set point) a 20°C. En este valor fue donde se identificó el modelo, es decir nos encontramos sobre el punto de operación por lo tanto el desempeño del controlador debe cumplir

con las condiciones de diseño. Como se puede observar en la Figura 3.25, el tiempo de establecimiento es aproximadamente de 1000 segundos. Después de que el sistema alcanza el régimen permanente se inyecta un segundo escalón a 23°C, donde nos alejamos del punto de operación y el tiempo de establecimiento cambia por la no linealidad del sistema. Al igual que en las pruebas del controlador PID, la ganancia estática del sistema es mucho más grande, la cual se multiplica con la ganancia del controlador por lo que tiene una respuesta oscilatoria debido a la agresividad del controlador. Finalmente, para concluir con la comprobación se baja la temperatura a 20°C nuevamente. Como se puede observar, el sistema se estabiliza en un tiempo aceptable y el desempeño del controlador es mucho mejor. Esto se debe a que estamos regresando a la zona de operación donde el controlador fue diseñado.

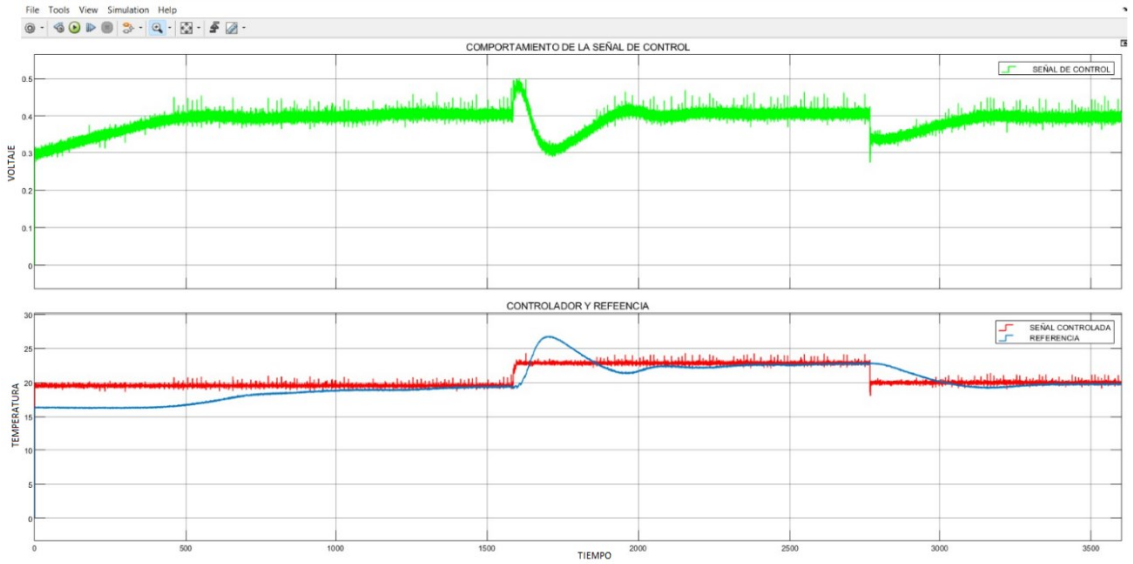


Figura 59.25 Proceso térmico a 24°C [(Ilustración por Autores)].

Capítulo 4

Conclusiones y Trabajos Futuros

- En este trabajo se realizó el diseño e implementación de dos controladores: PID y control por espacio de estados, mediante la tarjeta STM32F407 Discovery, creando una interfaz de comunicación entre el sistema microprocesador STM32F407 y el software Matlab Simulink mediante la librería Waijung, siendo esta una herramienta muy poderosa para el manejo de la tarjeta.

- Analizando las características de la tarjeta se tuvo que realizar un acoplamiento de voltajes mediante un divisor de tensión y un circuito de potencia debido a los niveles de voltaje en los que trabaja. Generalmente estos acoplamientos se basaron en partidores de tensión y en amplificadores, por lo tanto, su implementación fue sencilla. Estos aspectos deben ser considerados con alta importancia, ya que un mal manejo de los niveles de voltaje o corriente pueden dañar la tarjeta.

- Finalmente para el desarrollo de los controladores se realizó el diseño de los esquemas correspondientes para el control PID y el control por espacio de estados que se encuentran en el capítulo 3,en donde se controla la variable de temperatura del proceso térmico, realizando pruebas de campo y obteniendo resultados satisfactorios, los controladores trabajaron en diferentes valores de temperatura, mejorando el tiempo de estabilización de la planta y ofreciendo una interfaz gráfica de fácil uso para el usuario.
- Dentro del desarrollo de los controladores surgieron algunos problemas que fueron resueltos oportunamente con la documentación de la tarjeta y la librería. Sin embargo, se tuvo una complicación que no fue posible solucionarla. Esta complicación se debe a que no fue posible cambiar el tiempo de muestreo de la tarjeta. Únicamente se pudo asignar el tiempo de 0.01s, lo cual, a nuestro criterio y según la lógica, es un valor demasiado pequeño para un proceso térmico. Al respecto, se debe indicar que la velocidad de la tarjeta pudo responder satisfactoriamente, sin embargo, es un tiempo innecesario para un proceso térmico.
- Los controladores propuestos se pudieron implementar de manera correcta. Más allá de comparar su desempeño, se pudo verificar que la tarjeta puede aprovechar todo el potencial de simulink para generar esquemas de control complejos. Por ejemplo, el bloque PID discreto, internamente es un bloque complejo, del cual el compilador generó

el código y se pudo descargar y procesar en la tarjeta sin problemas. De igual manera, el controlador en espacio de estados involucra operaciones matriciales, las cuales también fueron resueltas por la tarjeta sin inconvenientes.

- El hecho de haber podido implementar estos controladores abre las puertas para poder implementar otros esquemas de control más avanzados, que seguramente podrán abordar de mejor manera el problema de la no linealidad de la planta. En este sentido se podría implementar esquemas de control basados en lógica difusa, controladores adaptativos o predictivos. Incluso se plantea como desafío el hecho de poder entrenar y programar una red neuronal en la tarjeta, lo cual podría no solo ser usado para resolver problemas de control sino otro tipo de problemas de ingeniería.
- También se plantea el hecho de utilizar otro tipo de tarjetas más avanzadas y del mismo fabricante, de manera que se pueda abrir una línea de investigación para desarrollar otro tipo de aplicaciones que incluyan mayores funcionalidades de Simulink.

REFERENCIAS BIBLIOGRÁFICAS

1. *Aimagin Support*. (s. f.). Recuperado 26 de mayo de 2022, de https://support.aimagin.com/
2. Andino Alberca, C. A., & Rodríguez Sánchez, D. X. (2016). *Diseño e implementación de algoritmos de control inteligente para un robot Phoenix tipo hexápodo mediante la tarjeta STM32F4 discovery y simulink de matlab* [BachelorThesis, Universidad de las Fuerzas Armadas ESPE. Carrera de Ingeniería en Electrónica, Automatización y Control.]. http://repositorio.espe.edu.ec/jspui/handle/21000/11839
3. Bharadwaj, R., Cunningham, K. M., Zhang, K., & Lloyd, T. E. (2016). FIG4 regulates lysosome membrane homeostasis independent of phosphatase function. *Human Molecular Genetics*, *25*(4), 681-692. https://doi.org/10.1093/hmg/ddv505
4. Espinel Cangui, M. A. (2012). *Artículo Científico—Diseño de un sistema de control en tiempo real para el Kernel del Sistema Operativo utilizando Matlab-Simulink.* http://repositorio.espe.edu.ec/jspui/handle/21000/5995
5. Goranov, A. E. (s. f.). *DISEÑO E IMPLEMENTACIÓN DE UN CUADRICÓPTERO BASADO EN EL MICROCONTROLADOR STM32F4*.
6. Granda Ortega, D. E., & Jadán Campoverde, A. P. (2020). *Diseño y construcción de un prototipo de proceso térmico didáctico orientado al análisis y control de sistemas dinámicos*. http://dspace.ups.edu.ec/handle/123456789/19467
7. Lab, M. (2019, diciembre 27). *STM32F4 Discovery Board Pinout, Features and Examples*. Microcontrollers Lab. https://microcontrollerslab.com/stm32f4-discovery-board-pinout-features-examples/
8. Mejia, B. P. B. (s. f.). *DISEÑAR Y CONSTRUIR UN PROTOTIPO DE SONÓMETRO DIGITAL CON PONDERACIÓN DE FRECUENCIA A.* 129.
9. *NI myRIO-1900 User Guide and Specifications*. (s. f.).
10. *O'Connor, D. et al (2016), Universality, Integr... |OECD*. (s. f.). Recuperado 24 de febrero de 2023, de https://community.oecd.org/docs/DOC-104443

11. Ogata, K. (1996). *Sistemas de control en tiempo discreto*. Pearson Educación.
12. Proaño, I. (2016). *TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO ELECTRÓNICO, AUTOMATIZACIÓN Y CONTROL*. 110.
13. Ramos, J. J. M., Garrido, R., & Zavala, G. C. (2020). *Real-time control prototype and its use in Automatic Control courses.* 6.
14. Saeteros Ortiz, C. M. (2018). *Implementación de un control PID Fuzzy la regulación de nivel del módulo MPS PA Compact workstation mediante la tarjeta STM32F4 Discovery* [BachelorThesis]. http://dspace.ups.edu.ec/handle/123456789/15353
15. Silva-Díaz, L. J., Morejón-Mesa, Y., Silva-Díaz, L. J., & Morejón-Mesa, Y. (2019). Sistemas Embebidos: Una alternativa para la automatización de la agroindustria cubana. *Revista Ciencias Técnicas Agropecuarias*, *28*(3). http://scielo.sld.cu/scielo.php?script=sci_abstract&pid=S2071-00542019000300008&lng=es&nrm=iso&tlng=es
16. *STM32 F4 Cortex™-M4 MCUs—STMicro | Mouser*. (s. f.). Recuperado 24 de febrero de 2023, de https://www.mouser.ec/new/stmicroelectronics/stm32f4/
17. *STM32 Microcontrollers—STMicro | Mouser*. (s. f.). Recuperado 24 de febrero de 2023, de https://www.mouser.ec/new/stmicroelectronics/stm32/
18. Taipe, C. (2019). Aplicación del software MATLAB en el aprendizaje de la cinemática lineal de una partícula en estudiantes universitarios de ingeniería. *Revista Innova Educación*, *1*(3), Art. 3. https://doi.org/10.35622/j.rie.2019.03.002
19. *UNIVERSIDAD POLITÉCNICA SALESIANA SEDE QUITO CARRERA: INGENIERÍA ELECTRÓNICA. Trabajo de titulación previo a la obtención del título de: - PDF Free Download*. (s. f.). Recuperado 17 de febrero de 2023, de https://docplayer.es/204192284-Universidad-politecnica-salesiana-sede-quito-carrera-ingenieria-electronica-trabajo-de-titulacion-previo-a-la-obtencion-del-titulo-de.html