



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**  
FACULTAD DE EDUCACIÓN TÉCNICA  
PARA EL DESARROLLO  
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

TEMA:

**Desarrollo de un algoritmo de control PID utilizando  
microcontroladores STM32 Cortex 4 para Robot Balancín  
mediante comunicación inalámbrica WiFi**

AUTOR:

Serrano Zambrano, Robert Ramon

Componente práctico del examen complejo previo a la  
obtención del título de **INGENIERO EN  
TELECOMUNICACIONES**

REVISOR:

M. Sc. Pacheco Bohórquez, Héctor Ignacio

Guayaquil, Ecuador

16 de septiembre del 2019



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**  
FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO  
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

## **CERTIFICACIÓN**

Certificamos que el presente **componente práctico del examen complejo**, fue realizado en su totalidad por **Serrano Zambrano, Robert Ramon** como requerimiento para la obtención del título de **INGENIERO EN TELECOMUNICACIONES**.

REVISOR

---

M. Sc. Pacheco Bohórquez, Héctor Ignacio

DIRECTOR DE CARRERA

---

M. Sc. Heras Sánchez, Miguel Armando

Guayaquil, 16 de septiembre del 2019



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**  
FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO  
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

### **DECLARACIÓN DE RESPONSABILIDAD**

Yo, **Serrano Zambrano, Robert Ramon**

#### **DECLARÓ QUE:**

**El componente práctico del examen complejo: Desarrollo de un algoritmo de control PID utilizando microcontroladores STM32 Cortex 4 para Robot Balancín mediante comunicación inalámbrica WiFi**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, 16 de septiembre del 2019

EL AUTOR

---

SERRANO ZAMBRANO, ROBERT RAMON



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**  
FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO  
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

### **AUTORIZACIÓN**

Yo, **Serrano Zambrano, Robert Ramon**

Autorizó a la Universidad Católica de Santiago de Guayaquil a la **publicación** en la biblioteca de la institución del **componente práctico del examen complejo: Desarrollo de un algoritmo de control PID utilizando microcontroladores STM32 Cortex 4 para Robot Balancín mediante comunicación inalámbrica WiFi**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, 16 de septiembre del 2019

EL AUTOR

---

SERRANO ZAMBRANO, ROBERT RAMON

# REPORTE DE URKUND

URKUND ★ Probar 7 Urkund Bloques Fernando Palacios Meléndez (edwin\_palacios)

**Documento** [Robert\\_Serrano\\_EC\\_2019A.docx](#) (D55758955)  
**Presentado** 2019-09-19 01:32 (-05:00)  
**Presentado por** fernandopm23@hotmail.com  
**Recibido** edwin.palacios.ucsg@analysis.orkund.com  
**Mensaje** Revisión EC Robert Serrano [Mostrar el mensale completo](#)  
1% de estas 8 páginas, se componen de texto presente en 1 fuentes.

Categoría	Enlace/nombre de archivo
>	<a href="#">Complejivo genesis BALANCE.docx</a>
	<a href="#">DOCUMENTO-VEGA.docx</a>
	<a href="#">Componente practico complexivo - vega marzo.docx</a>
<b>Fuentes alternativas</b>	
	Complejivo genesis BALANCE terminado.docx
	<a href="#">Complejivo genesis BALANCE.docx</a>

0 Advertencias. Reiniciar Exportar Compartir

UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

TEMA:

Desarrollo de un algoritmo de control PID

utilizando microcontroladores STM32 Cortex 4 para Robot Balancin mediante comunicación Inalámbrica WIFI

AUTOR: Serrano Zambrano, Robert Ramon

Componente práctico del examen complexivo previo a la obtención del título de INGENIERO EN TELECOMUNICACIONES

## **DEDICATORIA**

Este trabajo es dedicado a mi familia que con mucho esfuerzo me ayudaron a poder cumplir esta meta estudiantil.

**EL AUTOR**

**SERRANO ZAMBRANO, ROBERT RAMON**

## **AGRADECIMIENTO**

Agradecimiento a mis profesores por haber instruido en todo este periodo y poder inculcarme todos sus conocimientos.

EL AUTOR

SERRANO ZAMBRANO, ROBERT RAMON



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**  
FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO  
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

**TRIBUNAL DE SUSTENTACIÓN**

f. \_\_\_\_\_  
**M. Sc. ROMERO PAZ, MANUEL DE JESÚS**  
DECANO

f. \_\_\_\_\_  
**M. Sc. HERAS SÁNCHEZ, MIGUEL ARMANDO**  
DIRECTOR DE CARRERA

f. \_\_\_\_\_  
**M. Sc. PALACIOS MELÉNDEZ, EDWIN FERNANDO**  
OPONENTE



## ÍNDICE GENERAL

ÍNDICE DE FIGURAS.....	XI
ÍNDICE DE TABLAS .....	XII
Resumen .....	XIII
CAPÍTULO 1: DESCRIPCIÓN DEL COMPONENTE PRÁCTICO .....	2
1.1. Introducción.....	2
1.2. Objetivo General. ....	3
1.3. Objetivos Específicos. ....	3
CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA .....	4
2.1. Robots de equilibrio o balancín. ....	4
2.2. Las características de controladores para robots balance. ....	5
CAPÍTULO 3: DESARROLLO DEL COMPONENTE PRÁCTICO.....	7
3.1. Diagrama de control del robot. ....	7
3.2. Tarjeta controladora STM32F105.....	9
3.3. Controlador de motor LV8731V.....	10
3.4. Giroscopio y acelerómetro MP6050. ....	11
3.5. STM32CUBE MX. ....	11
3.6. Creación del proyecto stm32cube. ....	12
3.7. Configuración de protocolos del proyecto stm32cube. ....	12
3.8. Configuración del reloj del proyecto stm32cube.....	13
3.9. Generar proyecto stm32cube. ....	14
3.10. Ide de programación eclipse System Workbench. ....	15
3.11. Programación eclipse System workbench del prototipo. ....	15
3.12. Pruebas del robot. ....	18

Conclusiones .....	19
Recomendaciones .....	20
Referencias bibliográficas .....	21

## ÍNDICE DE FIGURAS

### Capítulo 2:

Figura 2. 1: Punto de ajuste y ángulo de inclinación real del robot. ....	4
Figura 2. 2: Robot comercial de equilibrio SEGWAY. ....	5

### Capítulo 3:

Figura 3. 1: Diagrama de control.....	7
Figura 3. 2: Estructura del robot balancín .....	8
Figura 3. 3: Tarjeta Stm32F105 .....	8
Figura 3. 4: Tarjeta actuadora.....	9
Figura 3. 5: Tarjeta wifi con host usb. ....	9
Figura 3. 6: Pin planner del microcontrolador STM32F105.....	10
Figura 3. 7: Diagrama esquemático del controlador LV8731V .....	10
Figura 3. 8: Giroscopio MPU 6050.....	11
Figura 3. 9: Giroscopio MPU 6050.....	11
Figura 3. 10: Desarrollo de nuevo proyecto .....	12
Figura 3. 11: Desarrollo de nuevo proyecto .....	12
Figura 3. 12: Desarrollo de nuevo proyecto .....	13
Figura 3. 13: Configuraciones en el cubemax para puertos seriales, cristal del microcontrolador .....	13
Figura 3. 14: Configuración de reloj de nuevo proyecto.....	14
Figura 3. 15: Configuración de reloj de nuevo proyecto.....	14
Figura 3. 16: System Workbench.....	15
Figura 3. 16: Cálculo del controlador proporcional-integral-derivativo (PID)	15
Figura 3. 18: Declaraciones de las variables de set point del ángulo de inclinación del robot balancín.....	16
Figura 3. 19: Métodos para la comunicación serial y USART .....	16
Figura 3. 20: Posición angular de referencia para el controlador PID.....	17
Figura 3. 21: System Workbench.....	17

## ÍNDICE DE TABLAS

### Capítulo 2:

Tabla 2. 1: Efecto de la sintonización PID.....	6
---	---

## Resumen

El documento del componente práctico del examen complejo se encarga de desarrollar un algoritmo de control PID utilizando microcontroladores STM32 Cortex 4 para Robot Balancín mediante comunicación inalámbrica WiFi. Las aplicaciones del robot balance son diversas y en este caso fue de encontrar la ecuación del controlador PID que permita la estabilidad y equilibrio de sistema y para ello fue necesario el microcontrolador STM32. En el capítulo 1, se describe una introducción relacionada a la robótica en la sociedad, así como, el objetivo general y los objetivos específicos del componente práctico. En el capítulo 2, se describe brevemente el estado del arte del robot balance con sus respectivas características de controladores PID. Finalmente, en el capítulo 3, se presenta el robot balance junto con los elementos que lo compone, también se muestra el desarrollo del algoritmo de equilibrio utilizando el controlador PID y su funcionalidad empleando el microcontrolador STM32 Cortex 4.

**Palabras claves:** CONTROLADORES, STM32, PROPORCIONAL, COMUNICACIÓN, INALÁMBRICA, PID

## **CAPÍTULO 1: DESCRIPCIÓN DEL COMPONENTE PRÁCTICO**

### **1.1. Introducción.**

En la sociedad moderna, las personas usan robots en su vida cada vez más. Los robots se usan de muchas maneras, como mejorar la experiencia de vida de las personas o ayudar a las personas a resolver problemas. Los robots pueden diseñarse para un propósito específico, por lo que resuelven problemas que la gente quiere que resuelvan. Los robots están diseñados por muchas personas y contruidos por todo tipo de materiales, para que puedan superar algunas de las limitaciones humanas como el tamaño, la velocidad, la capacidad de detección o la capacidad de procesamiento.

Las formaciones y diseños de los robots están cambiando con la historia. Interactúan cada vez más con las personas y ayudan a las personas cada vez más. Los robots son tan importantes que ayudan a la sociedad tanto y tan ampliamente que todos necesitan y reciben ayuda de algún tipo de robots en su vida diaria. Sin robots, muchos de los grandes logros no podrían hacerse tan pronto y de manera ordenada.

Hay innumerables categorías y estilos de robots que se diseñan y utilizan todos los días. Todos tienen su importancia y propósitos. Uno de los propósitos más importantes para usar robots es en misiones de rescate. Durante las misiones de rescate, el medio ambiente puede ser muy peligroso y difícil para todos. Podría ser demasiado difícil o imposible para las personas ayudar a otros en condiciones extremadamente difíciles. La eficiencia y el porcentaje de éxito podrían ser bajos. Lo peor de todo es que las personas pueden lastimarse o incluso morir. Por lo tanto, para evitar que ocurra este tipo de tragedia y aumentar las posibilidades de salvar personas, los robots están diseñados y contruidos para ayudarlos. En algunas situaciones, el rescatador debe pasar un puente estrecho para ayudar a las personas o dar suministros, pero el puente puede ser demasiado estrecho o demasiado peligroso para que las personas puedan pasar. En este tipo de situación, un pequeño robot balancín podría ayudar.

Este componente práctico estudia el robot balancín o de equilibrio que se puede utilizar en muchas situaciones de rescate. Hay cientos de miles de robots que se utilizan en misiones de rescate todos los días y todo el tiempo. Todos tienen sus funciones y misiones especiales. Un robot de equilibrio significa un robot con una o dos ruedas en contacto con el suelo y necesita algún tipo de método de equilibrio para controlar el ángulo del cuerpo y evitar que falle en el ángulo de inclinación o balanceo. Si bien el ángulo del cuerpo está controlado y se mantiene erguido, la posición de las ruedas también debe controlarse en ciertas situaciones. Controlar el ángulo de las ruedas significa mantener la rueda en la posición o ángulo que deseamos. Un robot balancín necesita equilibrar el ángulo de su cuerpo para mantenerse en pie. Al mismo tiempo, la posición de las ruedas debe permanecer en el mismo lugar. Además, el robot de equilibrio debe poder moverse hacia adelante, atrás, derecha e izquierda. Después de todos estos movimientos se han hecho, el robot balancín se inclina hacia adelante y la rueda se mantiene en un rango requerido. El componente práctica va utilizar un robot balancín con regulador cuadrático lineal (LQR) utilizando el controlador proporcional-integral-derivativo (PID). Los controladores están estabilizando el sistema y le piden que vaya a una posición o ángulo específico. Los trabajos se realizan en simulación y hardware.

## **1.2. Objetivo General.**

Desarrollar un algoritmo de control PID utilizando microcontroladores STM32 Cortex 4 para Robot Balancín mediante comunicación inalámbrica WiFi

## **1.3. Objetivos Específicos.**

- ✓ Describir el estado del arte de los robots balancín o de equilibrio.
- ✓ Diseñar el algoritmo del controlador PID utilizando el microcontrolador STM32 Cortex 4 para el robot balancín.
- ✓ Evaluar la funcionalidad y rendimiento del balance del robot de equilibrio utilizando control PID.

## CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA

### 2.1. Robots de equilibrio o balancín.

El problema del péndulo invertido no es infrecuente en el campo de la ingeniería de control. La singularidad y la amplia aplicación de la tecnología derivada de este sistema inestable ha despertado el interés de muchas investigaciones y entusiastas de la robótica en todo el mundo. En los últimos años, los investigadores han aplicado la idea de un modelo de péndulo invertido móvil a varios problemas, como el diseño de pasos para caminar para robots humanoides, sillas de ruedas robóticas y sistemas de transporte personal.

Los investigadores Grasser, D'Arrigo, Colombi, & Rufer, (2002) pertenecientes al laboratorio de electrónica industrial del Instituto Federal Suizo de Tecnología desarrollaron en el 2002 un prototipo reducido de un vehículo de dos ruedas controlado por un procesador de señal digital basado en el péndulo invertido con pesas unidas al sistema para simular un conductor humano, tal como se muestra en la figura 2.1. Se utiliza un controlador de espacio de estado lineal que utiliza información sensorial de un giroscopio y codificadores de motor para estabilizar este sistema.



Figura 2. 1: Punto de ajuste y ángulo de inclinación real del robot.

Fuente: (Grasser et al., 2002)

Dean Kamen, quien posee más de 150 patentes estadounidenses y extranjeras relacionadas con dispositivos médicos, sistemas de control climático y diseño de helicópteros, ha inventado un sistema similar y



disponible comercialmente, "SEGWAY HT" que puede equilibrar una posición humana en su plataforma mientras el usuario atraviesa el terreno con él. Esta innovación utiliza cinco giroscopios y una colección de otros sensores de inclinación para mantenerse en posición vertical. Solo se necesitan tres giroscopios para todo el sistema, los sensores adicionales se incluyen como medida de seguridad.



Figura 2. 2: Robot comercial de equilibrio SEGWAY.  
Elaborado por: Autor.

## **2.2. Las características de controladores para robots balance.**

Para mantener el robot en posición vertical, los controladores más utilizados son los Proporcionales-Integrales-Derivativos (PID) y el regulador cuadrático lineal (LQR). Otras investigaciones también han explorado el uso del control lineal-gaussiano (LQG), de lógica difusa y la colocación de polos; sin embargo, en algunos casos nunca se implementaron en un robot y solo se experimentaron en simulaciones. En este trabajo de titulación también se controla el desplazamiento del robot, para lo cual se utiliza LQR o una combinación de controladores. Por ejemplo, LQR para mantener el robot en posición vertical y PID para controlar el desplazamiento, o un controlador PID en cascada.

PID y LQR son los controladores explorados en este componente práctico, por lo tanto, solo se proporcionarán más detalles para ellos. PID es quizás el controlador más utilizado, como lo afirma VanDoren "más de 60 años después de la introducción de controladores proporcionales-integrales-derivativos, siguen siendo el caballo de batalla del control de procesos industriales". El algoritmo se describe mediante la siguiente ecuación:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{d}{dt} e(t)$$

Donde,  $u(t)$  es la salida del controlador,  $e(t)$  es el error y  $K_p$ ,  $K_i$  y  $K_d$  son los parámetros de ajuste del controlador PID. Esto es relativamente fácil de implementar y no requiere un modelo del sistema. El ajuste de los parámetros se puede hacer con prueba y error. Mediante este método simple, se ha controlado que los robots permanezcan en posición vertical. La tabla 2.1 muestra los efectos de sintonización en controladores PID.

Tabla 2. 1: Efecto de la sintonización PID

Respuesta	Incremento de tiempo	Incremento	Tiempo de estabilización	Estado estable error
<b>Kp</b>	Disminuye	Incrementa	Cambio menor	Disminuye
<b>Ki</b>	Disminuye	Incrementa	Disminuye	Eliminar
<b>Kd</b>	Cambio Menor	Disminuye	Disminuye	Sin cambio

Fuente: (Hormaza R., 2018)

Hay que tener en cuenta que estas correlaciones pueden no ser exactamente precisas porque  $K_p$ ,  $K_i$  y  $K_d$  dependen uno del otro. De hecho, cambiar una de estas variables puede cambiar el efecto de las otras dos.

## CAPÍTULO 3: DESARROLLO DEL COMPONENTE PRÁCTICO.

### 3.1. Diagrama de control del robot.

El diagrama de control a implementar para el robot, tiene como actuadores para el control de posición dos motores de paso de 12v(voltios), 500 rpm(revoluciones por minuto) con un torque de 320 Nm(Newton metro), como retroalimentación para la estabilización un giroscopio mp5060, una tarjeta controladora en tiempo real y de alta prestaciones stm32f105 con un córtex M4, modulo inalámbrico wifi como servidor host usb conectado físicamente a un shield stm32, siguiendo la figura 3.1 donde se muestra el diagrama de control.

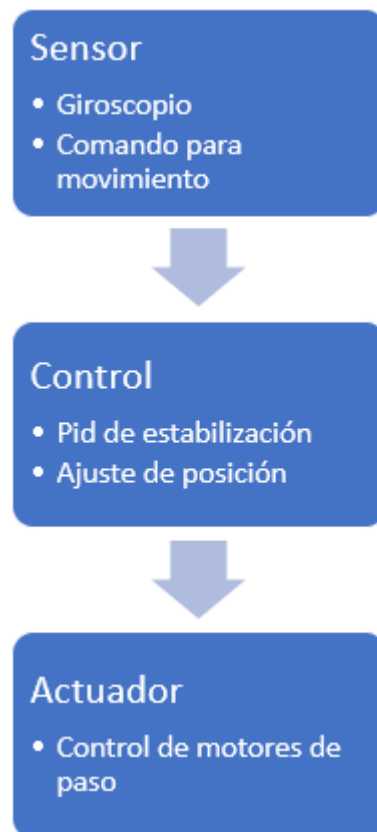


Figura 3. 1: Diagrama de control.  
Elaborado por: Autor.

En la figura 3.2 se muestra el prototipo acoplado con una cámara con la cual se visualiza el video mediante un aplicativo el cual nos sirve como un control en tiempo real para evitar colisiones y logrando un mayor plus de control al momento de pruebas de campo.

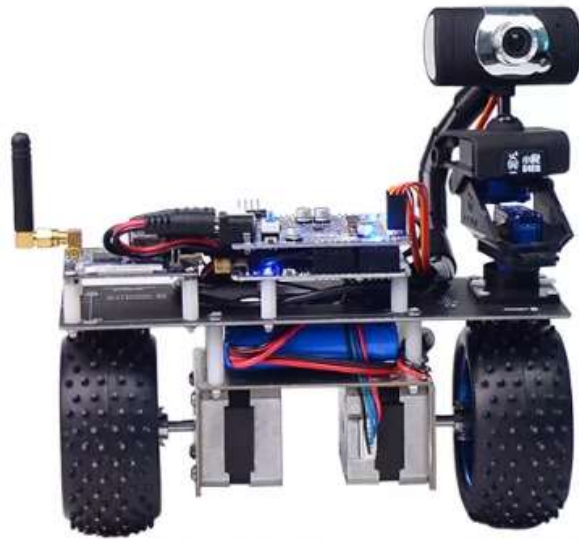


Figura 3. 2: Estructura del robot balancín  
Elaborado por: Autor.

La figura 3.3 muestra la tarjeta de desarrollo la cual es basada en stm32F105, que posee un microcontrolador de alta gama permitiendo la implementación de sistemas multitarea para las ejecuciones simultaneas de diferentes aplicaciones, superando a otros microcontroladores convencionales.



Figura 3. 3: Tarjeta Stm32F105  
Elaborado por: Autor.

La tarjeta actuadora se muestra en la figura 3.4, esta tarjeta posee el controlador (driver) para los motores de paso LV871 permitiendo un soporte de corriente de 3 amperios para controlar el sentido del motor y también las conexiones usb.

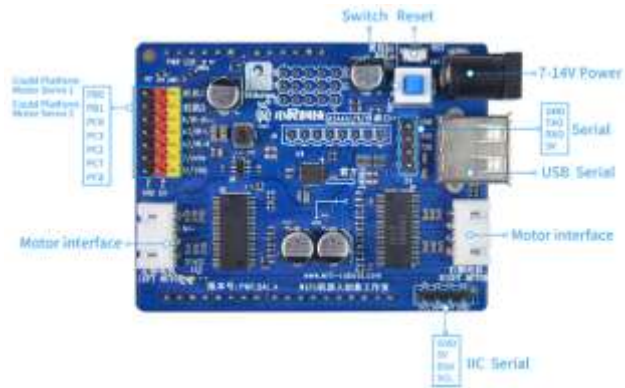


Figura 3. 4: Tarjeta actuadora  
Elaborado por: Autor.

La tarjeta inalámbrica donde se implementa el servidor para la conexión desde un computador o dispositivo móvil se muestra en la Figura 3.5 donde al iniciar se crea una red WiFi donde se conectará el dispositivo maestro que enviara los comandos.



Figura 3. 5: Tarjeta wifi con host usb.  
Elaborado por: Autor

### 3.2. Tarjeta controladora STM32F105.

Es un microcontrolador Flash de 32 bits desarrollada por ST Microelectronics el cual consta con una arquitectura que tiene como núcleo el procesador ARM® Cortex®, teniendo un rendimiento muy alto, capacidades en tiempo real, procesamiento de señales digitales y funcionamiento de baja potencia y bajo voltaje, el cual se muestra en la figura 3.6.

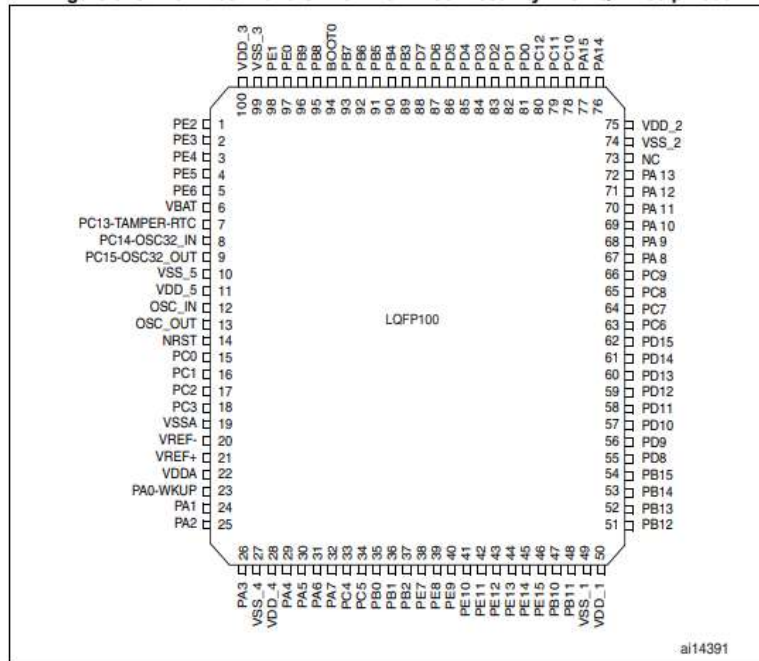


Figura 3. 6: Pin planner del microcontrolador STM32F105.  
Elaborado por: Autor

### 3.3. Controlador de motor LV8731V.

El controlador de motor que muestra en la figura 3.7 está basada en un tipo de controlador de modulación de pulsos continuos para un motor de paso, teniendo protecciones para los cambios de giro y paros de emergencias evitando que los picos de voltajes causen algún desperfecto a la parte electrónica y maximizando el tiempo de vida útil del robot.

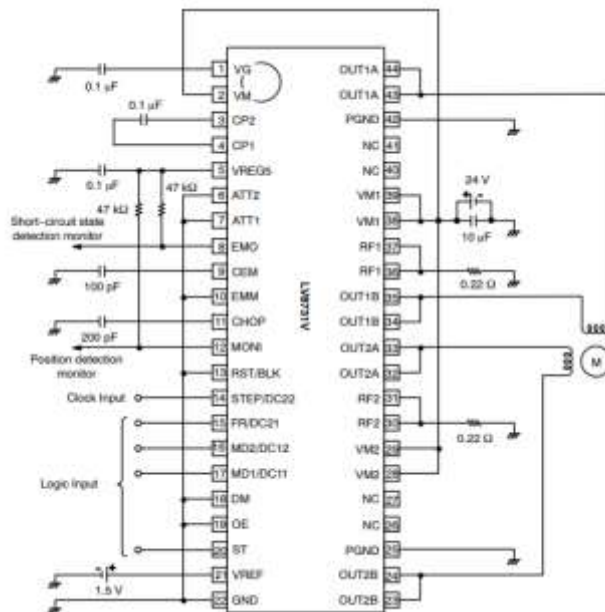


Figura 3. 7: Diagrama esquemático del controlador LV8731V  
Elaborado por: Autor



### 3.4. Giroscopio y acelerómetro MP6050.

El giroscopio es un sensor el cual nos envía mediante protocolo de comunicación i2c las posiciones en los ejes en el espacio x, y, z también como el ángulo de inclinación de este que sirve como retroalimentación del robot balancín para la estabilización de este. En la figura 3.8 se muestra el módulo en montaje superficial integrado a la placa actuadora.

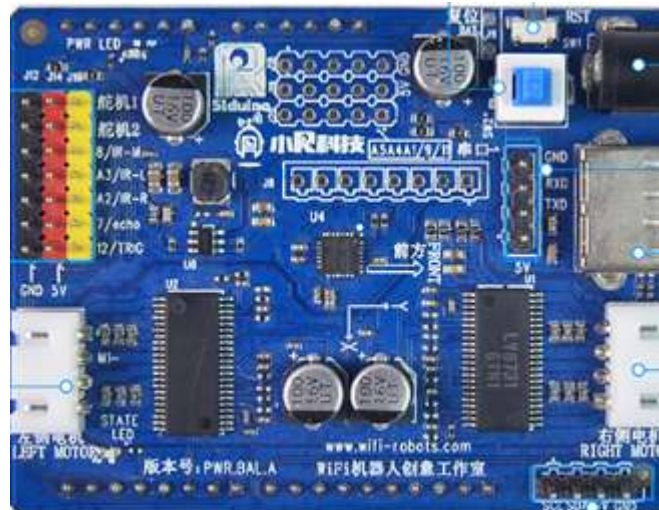


Figura 3. 8: Giroscopio MPU 6050  
Elaborado por: Autor

### 3.5. STM32CUBE MX.

STMCube™ de ST Microelectronics está diseñado para facilitar la vida de los desarrolladores mediante la reducción de los esfuerzos de desarrollo, el tiempo y el costo, el STM32Cube incluye el STM32CubeMX, que es una herramienta de configuración de software gráfico que permite generar código de inicialización C utilizando asistentes gráficos, que se muestra en la Figura 3.9.

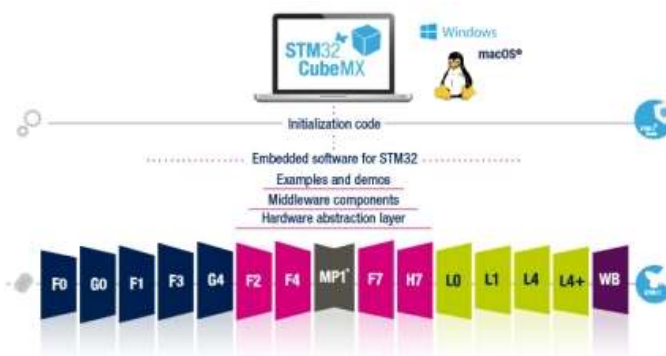


Figura 3. 9: Giroscopio MPU 6050  
Elaborado: Autor.

### 3.6. Creación del proyecto stm32cube.

STMCube™ se configura los pines de entrada y salida, protocolos de comunicación y demás periféricos a usarse, permitiendo desarrollar una arquitectura propia definida por el usuario y dando altas prestaciones en el tiempo de ejecución. La figura 3.10 muestra cómo se creará el proyecto y se dará paso a la configuración de cada uno de los pines.



Figura 3. 10: Desarrollo de nuevo proyecto  
Elaborado por: Autor

Al crear el proyecto se muestra una ventana donde se selecciona el controlador, el tipo de familia y el corte al que pertenece como se muestra en la figura 3.11.

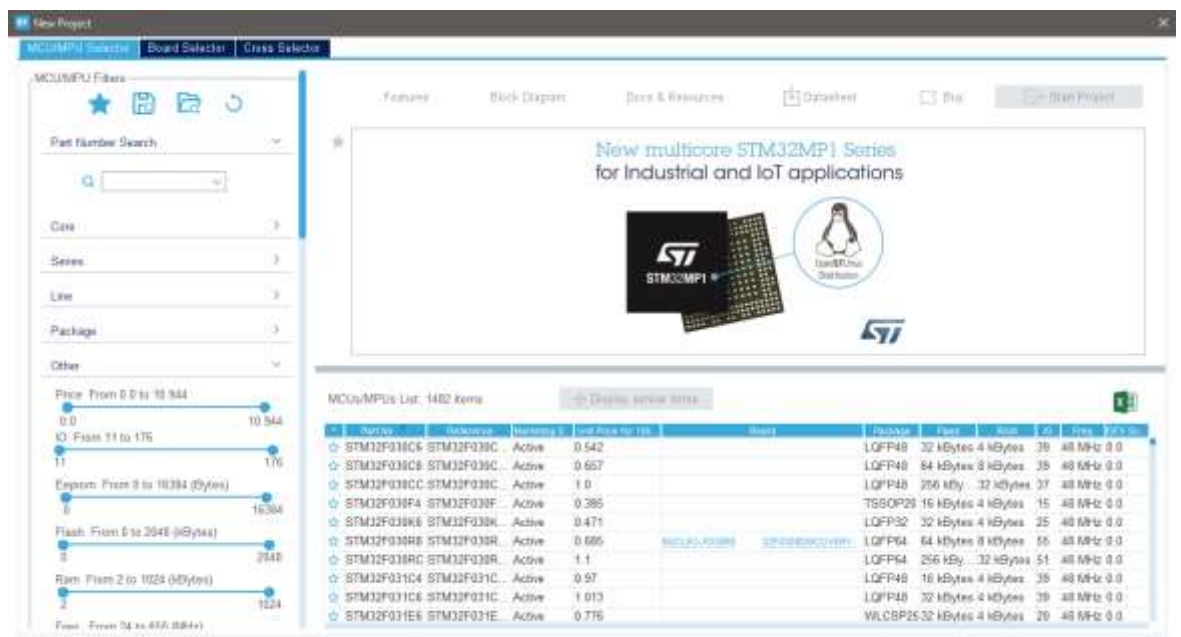


Figura 3. 11: Desarrollo de nuevo proyecto  
Elaborado por: Autor

### 3.7. Configuración de protocolos del proyecto stm32cube.

En la figura 3.12 se muestra el controlador seleccionado, al momento se configura los protocolos mediante la barra izquierda del ide donde se selecciona el tipo de elementos a configurar como el reloj del sistema,



protocolos, interrupciones, sistemas en tiempo real, puertos gpio y comunicaciones.



Figura 3. 12: Desarrollo de nuevo proyecto  
Elaborado por: Autor

En la figura 3.13 se muestra la configuración del serial, el puerto usb y la frecuencia del cristal del microcontrolador.

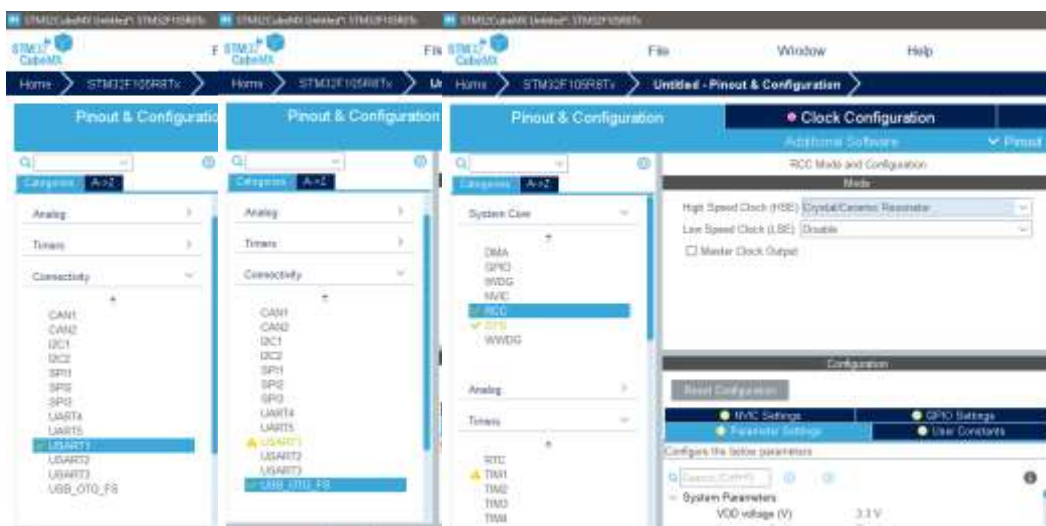


Figura 3. 13: Configuraciones en el cubemax para puertos seriales, cristal del microcontrolador  
Elaborado por: Autor

### 3.8. Configuración del reloj del proyecto stm32cube.

En la figura 3.14 se muestra la configuración interna de las velocidades del procesador interno para el proyecto y que la ejecución no tenga ningún retardo en la adquisición y procesamiento de datos.

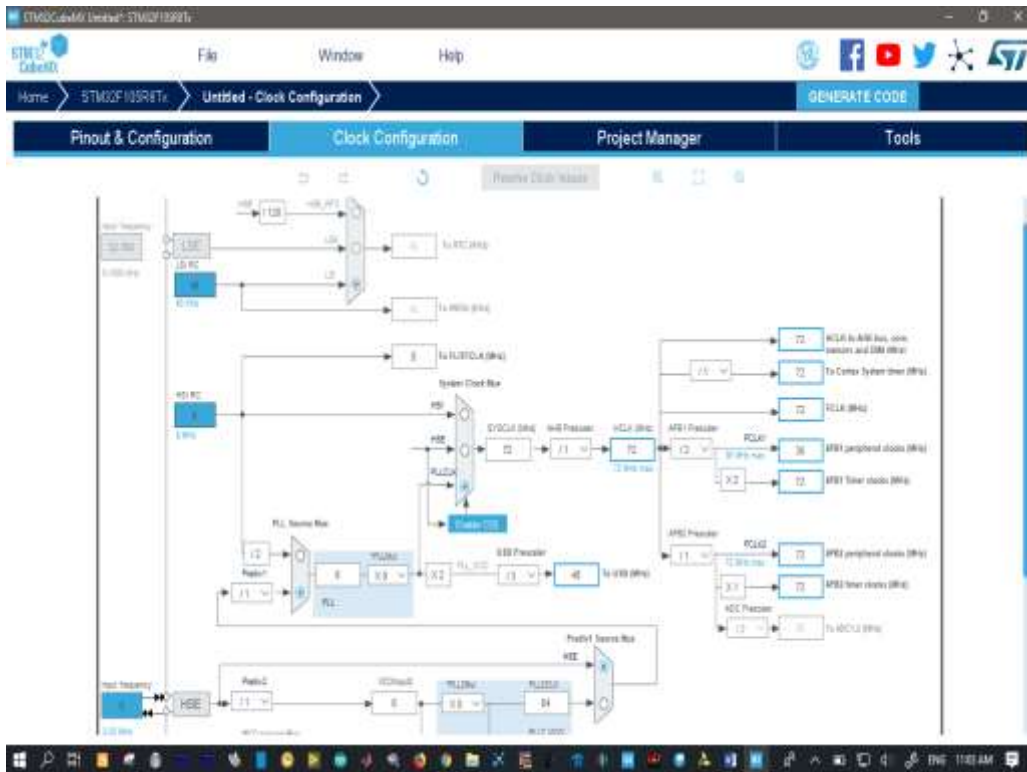


Figura 3. 14: Configuración de reloj de nuevo proyecto  
Elaborado por: Autor

### 3.9. Generar proyecto stm32cube.

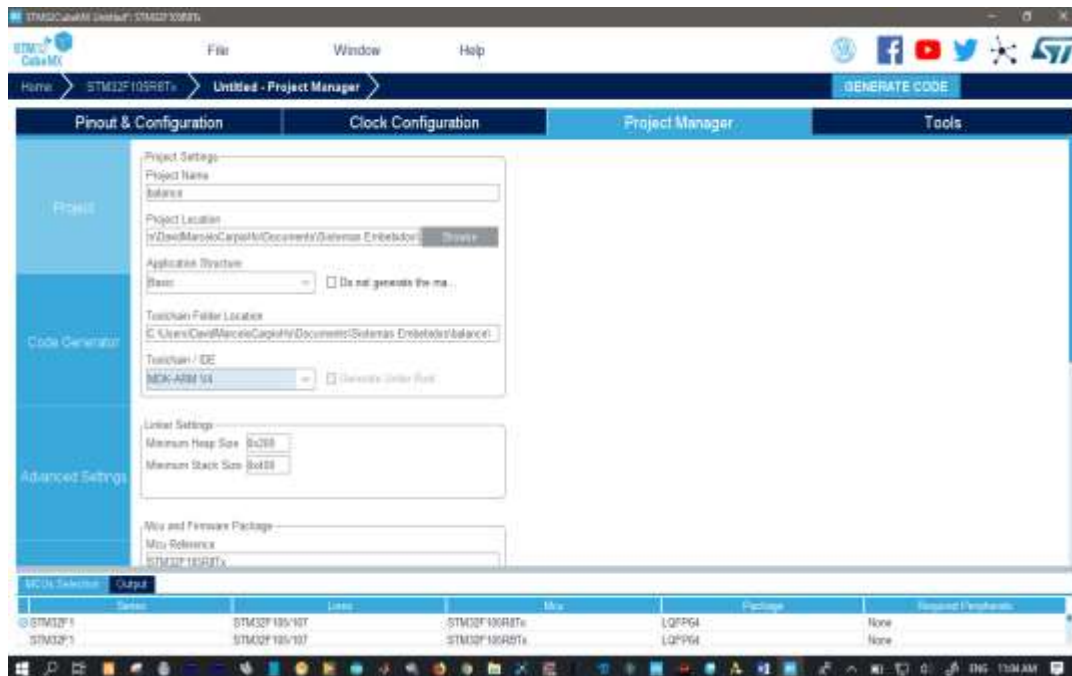


Figura 3. 15: Configuración de reloj de nuevo proyecto  
Elaborado por: Autor

En la figura 3.15 se muestra la generación del proyecto en el entorno de programación a usarse basado en eclipse, al momento de programar en el

entorno basado en eclipse ayudando al momento de las compilaciones al microcontrolador.

### 3.10. Ide de programación eclipse System Workbench.

La cadena de herramientas de System Workbench, llamada SW4STM32, es un entorno de desarrollo de software de múltiples sistemas operativos gratuito basado en Eclipse, que admite la gama completa de microcontroladores STM32 y placas asociadas (ver figura 3.16).



Figura 3. 16: System Workbench  
Elaborado por: Autor

### 3.11. Programación eclipse System workbench del prototipo.

En esta sección se definen los métodos para el cálculo del valor del controlador proporcional-integral-derivativo (PID) para corregir el valor dado y calcular el actuador entre cada elemento, tomando los datos de los sensores, y controlando los actuadores para la estabilización (ver la figura 3.17).

```
/*.....*/
float PidControl(float iAngle, float isetPoint )
{
    void Now;
    Now = micros();
    if (first) {
        first = false;
        Last_time = Now;
        integrated_error = 0;
    }
    timeChange = (Now - Last_time) ;
    timeChange = timeChange / 1000.0; // in millisekunden
    error = isetPoint - iAngle;
    if ( timeChange != 0 ) {
        dTerm = 1000.0 * Kd * (error - Last_error) / timeChange ;
    }
    integrated_error = integrated_error + ( error * timeChange );
    iTerm = Ki * integrated_error / 1000.0;
    pTerm = Kp * error + ( Ka * integrated_error ); // modifying Kp
    // Compute PID Output in Steps per second
    eSpeed = Kd * (pTerm + iTerm + dTerm) ;
    /*Remember something*/
    Last_time = Now;
    Last_error = error;
    if (eSpeed > 0 ) {
        DirForward = true ;
    } else {
        DirForward = false;
    }
    return eSpeed; // Steps per Second
}
/*.....*/
```

Figura 3. 17: Cálculo del controlador proporcional-integral-derivativo (PID)  
Elaborado por: Autor

Se declara las variables a usar en el código para el calculo de los valores del PID, referencias, datos de sensores y valores de los actuadores, tal como se muestra en la figura 3.18.

```

main.c
/
// -----
// Declaration
// -----
int          LoopTimeMsec = 12;
float        LoopTimeMicrosec = LoopTimeMsec * 1000;
unsigned long ProgStartTime; //general Start Time
const int    StartDelay = 20000; // msec
unsigned long CheckTimeStart;
int          CheckTimeEnd;

float        TuningValue;
float        SetpointA = 0;
float        SetpointB = 0;
float        setPoint = 0;
float        Angle = 0; // Sensor Aquisition
float        Calibration = -3.2;
float        Voltage;
float        ERROR ;
float        average;

```

Figura 3. 18: Declaraciones de las variables de set point del ángulo de inclinación del robot balancín.

Elaborado por: Autor

Se crean los métodos para el protocolo de comunicación serial el cual nos servirá para la lectura de los datos al microcontrolador que se muestra en la figura 3.19.

```

main.c
/
static void MX_USART1_UART_Init(void)
{
    /* USER CODE BEGIN USART1_Init 0 */

    /* USER CODE END USART1_Init 0 */

    /* USER CODE BEGIN USART1_Init 1 */

    /* USER CODE END USART1_Init 1 */
    huart1.Instance = USART1;
    huart1.Init.BaudRate = 115200;
    huart1.Init.WordLength = UART_WORDLENGTH_8B;
    huart1.Init.StopBits = UART_STOPBITS_1;
    huart1.Init.Parity = UART_PARITY_NONE;
    huart1.Init.Mode = UART_MODE_TX_RX;
    huart1.Init.HwFlowCtl = UART_HWCONTROL_NONE;
    huart1.Init.OverSampling = UART_OVERSAMPLING_16;
    if (HAL_UART_Init(&huart1) != HAL_OK)
    {
        Error_Handler();
    }
    /* USER CODE BEGIN USART1_Init 2 */

    /* USER CODE END USART1_Init 2 */

}

```

Figura 3. 19: Métodos para la comunicación serial y USART

Elaborado por: Autor

En la figura 3.20 se muestra el cálculo y lectura de los valores PID para el correcto funcionamiento del robot mediante los comandos enviados por el serial.

```

main.c
/* USER CODE END Header_StartTask02 */
void StartTask02(void const * argument)
{
    /* USER CODE BEGIN StartTask02 */
    /* Infinite loop */
    for(;;)
    {
        accX = (i2cData << 8) | i2cData-1;
        accY = (i2cData << 8) | i2cData-2;
        accZ = (i2cData << 8) | i2cData-3;
#ifdef RESTRICT_PITCH // Eq. 25 and 26
        double roll = atan2(accY, accZ) * RAD_TO_DEG;
        double pitch = atan(-accX / sqrt(accY * accY + accZ * accZ)) * RAD_TO_DEG;
#else // Eq. 28 and 29
        double roll = atan(accY / sqrt(accX * accX + accZ * accZ)) * RAD_TO_DEG;
        double pitch = atan2(-accX, accZ) * RAD_TO_DEG;
#endif
    }
    osDelay(1);
}
/* USER CODE END StartTask02 */
}

```

Figura 3. 20: Posición angular de referencia para el controlador PID.  
Elaborado por: Autor

En la figura 3.21 se muestra la corrección del controlador PID y de la actualización de datos en tiempo real.

```

main.c
* @retval None
*/
/* USER CODE END Header_StartTask03 */
void StartTask03(void const * argument)
{
    int SerialPacket,KA_P,KA_D,KP_I,K_Base,KP_P,SavingData ;
    for(;;){
        MySerialEvent();
        if(SerialPacket == true)
        {
            SerialPacket = false;
            switch(SerialPacket)
            {
                case 0x01:break;
                case 0x02: UpdatePID(); break;
                case 0x03: CarDirection();break;
                case 0x04: SendPID();break;
                case 0x05:
                    SavingData = KA_P;
                    SavingData = KA_D;
                    SavingData = KP_P;
                    SavingData = KP_I;
                    SavingData = K_Base;
                    WritePIDintoEEPROM(&SavingData);
                    break;
                case 0x06: break;
                case 0x07:break;
                default:break;
            }
        }
    }
}

```

Figura 3. 21: Corrección del controlador PID.  
Elaborado por: Autor

### 3.12. Pruebas del robot.

La parte experimental del robot balance fue mediante participación en diferentes concursos y exhibiciones. A nivel nacional, se desarrollan concursos de robótica donde existe la categoría de robot balance, que consiste en cumplir un determinado recorrido en el menor tiempo posible ya sea con o sin obstáculos. La figura 3.22 muestra al robot balance en pruebas de rendimiento. Durante las participaciones el robot logró cubrir una distancia de 3 m en 9 segundos.



Figura 3. 22: Funcionamiento del robot balance.  
Elaborado por: Autor

## **Conclusiones**

Al comienzo del proyecto, las expectativas eran bastante sustanciales. El objetivo fue desarrollar un robot que fuera tan robusto que pudiera recuperarse de una inclinación significativa e inducida deliberadamente. A medida que los problemas y las dificultades comenzaron a hacerse evidentes en la programación, las expectativas se redujeron a proporcionar simplemente un robot que pudiera mantener la estabilidad. La finalización del programa ha resultado ser muy buena, aunque se puede seguir mejorando la estabilidad del robot balance.

El lado del software y la programación fue decepcionante al inicio, así como la mayoría de las dificultades encontradas en estas áreas. Esto puede superarse con el tiempo, pero desafortunadamente, el tiempo no era abundante y a pesar de aquello el propósito fue cumplido, mantener el balance del robot mediante controladores PID.

## **Recomendaciones.**

Las siguientes recomendaciones proporcionan dirección para el trabajo futuro y áreas sugeridas de investigación adicional.

1. La estabilización proporcionada por la rueda de reacción está limitada por el par proporcionado por el motor de la rueda de reacción. El plan posterior es utilizar un disco giratorio y su precesión giroscópica para el equilibrio. Esto proporcionaría un diseño más estable capaz de proporcionar un par de restauración más alto. En tal caso, se debe prestar especial atención a los ejes rotativos, su alineación y cómo se fijan al modelo, a la posición y alineación de los soportes, y a El montaje y fijación de cualquier acoplamiento flexible.
2. Además de esto, el controlador de lógica difusa también se puede implementar para proporcionar flexibilidad y precisión en el control.



## Referencias bibliográficas

- Cedeño, & Verdezoto. (2016). Implementación de dos robots autónomos y uno controlado mediante tecnología bluetooth para las categorías seguidor de línea con obstáculos, laberinto y balancín. (UCSG). Recuperado de <http://repositorio.ucsg.edu.ec/handle/3317/6607>
- Grasser, F., D'Arrigo, A., Colombi, S., & Rufer, A. C. (2002). JOE: A mobile, inverted pendulum. *IEEE Transactions on Industrial Electronics*, 49(1), 107–114. <https://doi.org/10.1109/41.982254>
- Hormaza R., G. (2018). Diseño e implementación de un algoritmo con control PID y comunicación Bluetooth utilizando un Arduino Mega para el funcionamiento de un robot balance a distancia. (Trabajo de Titulación de Grado, Universidad Católica de Santiago de Guayaquil). Recuperado de <http://repositorio.ucsg.edu.ec/handle/3317/10232>
- Huang, A. S., & Rudolph, L. (2007). *Bluetooth Essentials for Programmers*. Cambridge University Press.
- Pérez, F. E. V., & Areny, R. P. (2007). *Microcontroladores: Fundamentos y aplicaciones con PIC*. Marcombo.



Presidencia  
de la República  
del Ecuador



Plan Nacional  
de Ciencia, Tecnología,  
Innovación y Saberes



SENESCYT  
Secretaría Nacional de Educación Superior,  
Ciencia, Tecnología e Innovación

## DECLARACIÓN Y AUTORIZACIÓN

Yo, **Serrano Zambrano, Rober Ramon** con C.C: # 092341968-3 autor del Trabajo de **componente práctico del examen complejo**: Desarrollo de un algoritmo de control PID utilizando microcontroladores STM32 Cortex 4 para Robot Balancín mediante comunicación inalámbrica WiFi, previo a la obtención del título de INGENIERO EN TELECOMUNICACIONES en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 16 de septiembre del 2019

f. \_\_\_\_\_

Nombre: SERRANO ZABRANO, ROBERT RAMON

C.C: 092341968-3



## REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA

### FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN

<b>TÍTULO Y SUBTÍTULO:</b>	Desarrollo de un algoritmo de control PID utilizando microcontroladores STM32 Cortex 4 para Robot Balancín mediante comunicación inalámbrica WiFi.		
<b>AUTOR(ES)</b>	SERRANO ZAMBRANO, ROBERT RAMON		
<b>REVISOR(ES)/TUTOR(ES)</b>	M. Sc. PACHECO BOHÓRQUEZ, HÉCTOR IGNACIO		
<b>INSTITUCIÓN:</b>	Universidad Católica de Santiago de Guayaquil		
<b>FACULTAD:</b>	Facultad de Educación Técnica para el Desarrollo		
<b>CARRERA:</b>	Ingeniería en Telecomunicaciones		
<b>TITULO OBTENIDO:</b>	Ingeniero en Telecomunicaciones		
<b>FECHA DE PUBLICACIÓN:</b>	16 de septiembre del 2019	<b>No. DE PÁGINAS:</b>	33
<b>ÁREAS TEMÁTICAS:</b>	Electrónica, Comunicaciones Inalámbricas, Microcontroladores		
<b>PALABRAS CLAVES/ KEYWORDS:</b>	CONTROLADORES, STM32, PROPORCIONAL, COMUNICACIÓN, INALÁMBRICA, PID		
<b>RESUMEN/ABSTRACT:</b>	<p>El documento del componente práctico del examen complejo se encarga de desarrollar un algoritmo de control PID utilizando microcontroladores STM32 Cortex 4 para Robot Balancín mediante comunicación inalámbrica WiFi. Las aplicaciones del robot balance son diversas y en este caso fue de encontrar la ecuación del controlador PID que permita la estabilidad y equilibrio de sistema y para ello fue necesario el microcontrolador STM32. En el capítulo 1, se describe una introducción relacionada a la robótica en la sociedad, así como, el objetivo general y los objetivos específicos del componente práctico. En el capítulo 2, se describe brevemente el estado del arte del robot balance con sus respectivas características de controladores PID. Finalmente, en el capítulo 3, se presenta el robot balance junto con los elementos que lo compone, también se muestra el desarrollo del algoritmo de equilibrio utilizando el controlador PID y su funcionalidad empleando el microcontrolador STM32 Cortex 4.</p>		
<b>ADJUNTO PDF:</b>	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
<b>CONTACTO CON AUTOR/ES:</b>	<b>Teléfono:</b> +593-9-81074718	<b>E-mail:</b> <a href="mailto:serranozambranorbert@gmail.com">serranozambranorbert@gmail.com</a>	
<b>CONTACTO CON LA INSTITUCIÓN: COORDINADOR DEL PROCESO DE UTE</b>	<b>Nombre:</b> Palacios Meléndez Edwin Fernando <b>Teléfono:</b> +593-9-68366762 <b>E-mail:</b> <a href="mailto:edwin.palacios@cu.ucsg.edu.ec">edwin.palacios@cu.ucsg.edu.ec</a>		
<b>SECCIÓN PARA USO DE BIBLIOTECA</b>			
<b>No. DE REGISTRO (en base a datos):</b>			
<b>No. DE CLASIFICACIÓN:</b>			
<b>DIRECCIÓN URL (tesis en la web):</b>			