

Instituto Tecnológico de Costa Rica
Área Académica de Ingeniería en Computadores

Programa de Licenciatura de Ingeniería en Computadores



Proyecto de Diseño de Ingeniería en Computadores

Verificación del Firmware del Drone Crazyflie 2.0

Proyecto de Diseño

Diseño de pruebas en un entorno aleatorio para exploración
por medio de drones con inteligencia de enjambre

Elaborado por

Jose Ignacio Granados Marín
Ingeniería en Computadores
No. Carné: 2018319698

Supervisado por: MSc. Luis Alberto Chavarría Zamora

1. Modelo del drone

El modelo considerado para el proyecto en cuestión proviene del fabricante Bitcraze, el cual se dedica, actualmente, al desarrollo y fabricación de drones, así como la constante implementación de infraestructuras de apoyo para sus diversos clientes, decks de expansión, depuradores, entornos de desarrollo y herramientas para que los usuarios finales puedan realizar modificaciones que deseen a los diversos productos que hayan adquirido [1].

Ahora bien, el drone Crazyflie consiste en un pequeño quadcopter que comenzó con la simple idea de desarrollar un tablero electrónico volador con la menor cantidad de piezas mecánicas posibles y que el mismo fuera capaz de utilizarse en entornos interiores de tamaños variados. Luego múltiples estudios, diseños y pruebas, el resultado fue el drone en cuestión, el cual se encuentra construido mediante una placa de circuito impreso (PCB), como marco mecánico principal, y cuatro motores que permiten movilizar al dispositivos según se requiera [1].

Por su parte, para el proyecto en cuestión, el modelo específico del drone a disposición el dispositivo Crazyflie 2.0.

2. Características del drone Crazyflie 2.0

Dentro de las principales características del drone Crazyflie 2.0 en cuestión, se encuentran las siguientes [2]:

- Puede ser utilizado como controlador de otros dispositivos.
- Se encuentra equipado con baja latencia, largo alcance radio y Bluetooth LE.
- Dispone de su propia aplicación e interfaz para dispositivos móviles, ordenadores y laptops.
- Se carga a través de USB.

- Dispone de una memoria EEPROM para almacenar parámetros de configuración.
- Dispone de IMU 10-DOF con acelerómetro, giroscopio, magnetómetro y sensor de presión de alta precisión.
- Su firmware está hecho para ser modificado.
- Dispone de una interfaz de expansión flexible, la cual permite diversas cubiertas, tanto en la parte superior como en la inferior del drone.
- Dispone de diversos buses UART, I2C y SPI, así como PWM, entrada / salida analógica y GPIO.

Por otra parte, el fabricante brinda y facilita las siguientes facilidades y especificaciones [2]:

- Trabajo continuo en el firmware, software, documentación y nuevas formas de uso del sistema.
- Diseño de nuevas plataformas de expansión que agregan nuevas funcionalidades, y posibilidades, las cuales permiten expandir el ecosistema del modelo Crazyflie 2.0.
- La plataforma soporta firmware inalámbrico y actualizaciones a través de radio y Bluetooth LE, por lo que cuando se lanza un nuevo firmware, su actualización es fácil de implementar.
- El Crazyflie 2.0 es un proyecto de código abierto, con código fuente y diseño de hardware documentado y disponible.
- La codificación del drone no es compatible con el sistema operativo de iOS.
- Existe una amplia serie de APIs compatibles escritas en Java, Ruby, C/C++, C# y Javascript.
- La API de cliente está escrita en Python.

- Existen muchas implementaciones, realizadas por diversos clientes, disponibles en la plataforma de GitHub, escritas en Ruby, C#, C/C++, JavaScript, NodeJS, CylonJS y Java.
- El procesador STM32F405 tiene la suficiente potencia computacional para llevar a cabo diversos experimentos, realizar mejoras y agregar nuevas funcionalidades al drone.
- Las plataformas de expansión permiten experimentar, crear prototipos y diseñar hardware propio.

Adicionalmente, algunas de las características de la interfaz de uso, son las siguientes [2]:

- El software, firmware y utilidades tienen funcionalidades como registro, configuración de parámetros en tiempo real y actualizaciones de firmware inalámbricas.
- El entorno de desarrollo completo, para la mayoría de los subsistemas, está disponible en una máquina virtual, por lo que no es necesario instalar ninguna cadena de herramientas para comenzar a desarrollar.
- Se encuentra disponible un kit de adaptador de desarrollo que admite fácilmente conexiones JTAG/SWD.

3. Propuestas de implementación de algoritmo PSO

Dadas cada una de características, funcionalidades, facilidades y beneficios que dispone el modelo del drone Crazyflie 2.0 en particular, es posible implementar el algoritmo de inteligencia de enjambre de optimización de enjambre de partículas (PSO), previamente desarrollado, para que el dispositivo en cuestión, disponga del comportamiento simulado en el motor de física de Pybullet. Ahora bien, con base en la especificaciones, documentación y repositorios oficiales del fabricante Bitcraze, se proponen las siguientes cuatro alternativas de diseño:

3.1. Personalización del firmware con Kbuild

El firmware del drone Crazyflie 2.0, en particular, es el sistema encargado de manejar y administrar el piloto automático del dispositivo en cuestión, a través del componente integrado STM32F4. Dicho sistema tiene la capacidad de controlar el movimiento del drone, según las instrucciones que el dispositivo reciba. A su vez, el firmware predeterminado depende de una serie de herramientas especializadas junto con los debidos procesos de instalación y compilación del mismo [3]. Sin embargo, el fabricante Bitcraze, en una sección particular de la documentación del drone, menciona que los usuarios tienen la capacidad de modificar el firmware predeterminado, a través de la realización de ciertos cambios en el Kbuild (Kernel build) junto con el establecimiento de algunos enlaces de Python. Dicho proceso, involucra una codificación tanto de alto como de bajo nivel, dado que es necesaria la interacción de los lenguajes de programación de C y Python.

Dadas las posibilidades anteriores, la propuesta consiste en integrar el algoritmo de inteligencia de enjambre desarrollado, a través de dichos enlaces de Python junto con la creación de una interfaz entre C y el lenguaje antes mencionado, para que que el Kbuild sea capaz de interpretar el algoritmo de optimización de enjambre de partículas (PSO).

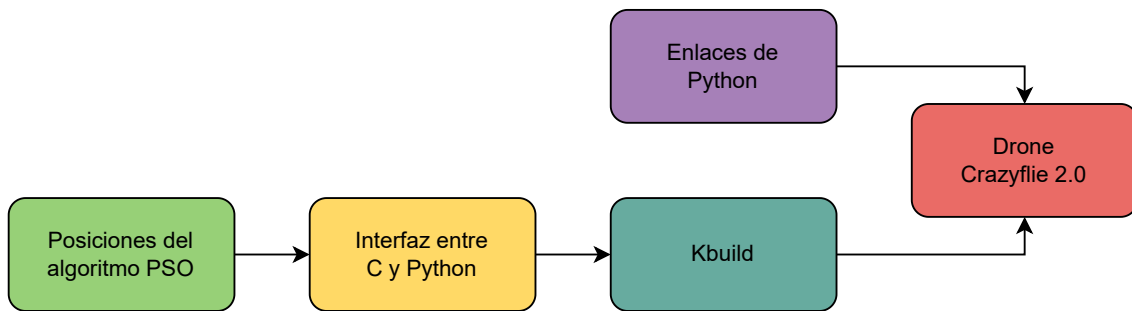


Figura 1: Diagrama de la primera propuesta de implementación.

3.2. Uso del Toolbelt

El Toolbelt del fabricante Bitcraze, consiste en una utilidad que permite ejecutar herramientas para probar y construir diversos módulos de software personalizados. Dicha tecnología, se ejecuta en un entorno de Docker, donde las cadenas de herramientas y frameworks se encuentran instalados previamente y, por lo tanto, eliminan la necesidad de disponer de compiladores en el sistema u ordenador local. En otras palabras, la idea básica detrás del Toolbelt consiste en proporcionar una herramienta de fácil acceso que ayude al usuario a realizar tareas comunes en sus respectivos proyectos, sin el arduo proceso de instalar múltiples cadenas de herramientas especiales, bibliotecas o programas [4].

Dada la serie de herramientas anteriores, la propuesta consiste en utilizar las funcionalidades necesarias que provea el Toolbelt para integrar el algoritmo de inteligencia de enjambre desarrollado, ya que el mismo consiste en un script de Python, el cual es corrido como un contenedor, de tal manera que, el mismo facilitará la interacción con el firmware del drone Crazyflie 2.0.

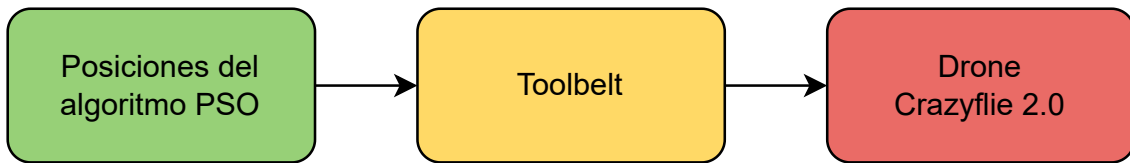


Figura 2: Diagrama de la segunda propuesta de implementación.

3.3. Adición de una nueva tarea al sistema

Una tarea FreeRTOS, dentro del firmware del drone Crazyflie 2.0, es similar a un subproceso en un sistema operativo estándar, ya que el mismo tiene su propia pila de llamadas de función. Una nueva tarea es, comúnmente, necesaria cuando se agrega un subsistema, fundamentalmente nuevo, al firmware del dispositivo en cuestión. Por lo general, si el nuevo subsistema necesita hacer algún cálculo significativo en respuesta a ciertas entradas, el mismo debe recibir dichos valores en una cola y llevar a cabo el respectivo proceso matemático dentro de su propia tarea [5].

Dada la alternativa anterior, la propuesta consiste en implementar el algoritmo de inteligencia de enjambre desarrollado, como una nueva tarea FreeRTOS del firmware del drone Crazyflie 2.0, en la será necesaria la configuración del entorno de desarrollo, constantes y variables para que el dispositivo puede ser controlado por el algoritmo de optimización de enjambre de partículas (PSO).



Figura 3: Diagrama de la tercera propuesta de implementación.

3.4. Creación de plataforma propia

La creación de una plataforma propia o personalizada puede ser una buena manera de ahorrar trabajo cuando se requiera adaptar una configuración específica para el drone Crazyflie 2.0, asegurar que se dispone de los valores predeterminados razonables para los parámetros necesario o confirmar que la inicialización del dispositivo se realiza correctamente. Lo anterior, involucra la adición de la plataforma propia al Kconfig (Kernel config), el archivo de origen de inicio al Kbuild (Kernel build), los valores de parámetros predeterminados, la configuración predeterminada, entre otras modificaciones en el firmware del dispositivo en cuestión [6].

Dada la opción anterior, la propuesta consiste en implementar el algoritmo de optimización de enjambre de partículas (PSO) desarrollado, como una nueva plataforma propia o personalizada para el drone Crazyflie 2.0, de manera que, el mismo pueda cumplir con los objetivos de exploración de territorios aleatorios, por medio de inteligencia de enjambre.

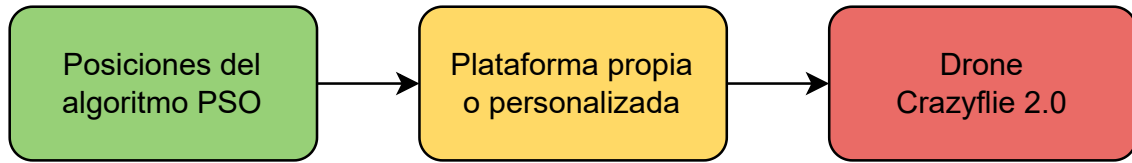


Figura 4: Diagrama de la cuarta propuesta de implementación.

Referencias

- [1] Bitcraze About. (s.f.). Home Bitcraze.
<https://www.bitcraze.io/about/bitcraze/>
- [2] Crazyflie 2.0 Bitcraze. (s.f.). Home Bitcraze.
<https://www.bitcraze.io/products/old-products/crazyflie-2-0/>
- [3] Building and Flashing Bitcraze. (s.f.). Home Bitcraze.
<https://www.bitcraze.io/documentation/repository/crazyflie-firmware/master/building-and-flashing/build/>
- [4] Bitcraze LPS Node Firmware. (s.f.). GitHub.
<https://github.com/bitcraze/lps-node-firmware>
- [5] Adding a new system task Bitcraze. (s.f.). Home Bitcraze.
<https://www.bitcraze.io/documentation/repository/crazyflie-firmware/master/development/systemtask/>
- [6] Creating Your Own Platform Bitcraze. (s.f.). Home Bitcraze.
https://www.bitcraze.io/documentation/repository/crazyflie-firmware/master/development/create_platform/