

TABLA DE CONTENIDOS

	página
Dedicatoria	I
Agradecimientos	II
Tabla de Contenidos	III
Índice de Figuras	VIII
Índice de Tablas	IX
Resumen	x
1. Introducción	11
1.1. Descripción del Contexto	11
1.2. Introducción al Problema	12
1.3. Objetivos	12
1.3.1. Objetivo General	12
1.3.2. Objetivos Específicos	13
1.4. Alcances del Proyecto	13
1.5. Resumen	13
2. Marco Teórico	15
2.1. Vehículo Aéreo no Tripulado (UAV)	15
2.1.1. Controladores de Vuelo	15
2.1.1.1. MultiWii	15
2.1.1.2. PixHawk	16
2.1.2. Microcomputador Beaglebone Black	17
2.1.3. Arquitectura	18
2.1.3.1. Centralizada	19
2.2. Capa de Abstracción	20
2.2.1. USAL	20
2.3. Interfaz de Programación de Aplicaciones (API)	22

2.3.1.	Diseño de un API	22
2.3.2.	Patrón de Diseño SOLID	23
2.3.2.1.	Principio de Única Responsabilidad (SRP)	24
2.3.2.2.	Principio Abierto/Cerrado (OCP)	24
2.3.2.3.	Principio de Sustitución de Liskov (LSP)	24
2.3.2.4.	Principio de Segregación de la Interfaz (ISP)	24
2.3.2.5.	Principio de Inversión de Dependencia (DIP)	25
2.3.3.	Lenguajes de Programación	25
2.3.3.1.	Lenguaje C	25
2.3.3.2.	Lenguaje C++	27
2.3.3.3.	Comparación	28
2.4.	Comunicación con Controladores de Vuelo	29
2.4.1.	Comunicación Serial	29
2.4.2.	Protocolo MAVLink	30
2.5.	Sistema Operativo para Robots (ROS)	31
2.6.	Resumen	33
3.	Análisis del Problema	34
3.1.	El Problema	34
3.1.1.	Subproblema 1: Comunicación entre BeagleBone Black y Controladores de Vuelo	35
3.1.2.	Subproblema 2: API	36
3.2.	Diseño General	37
3.2.1.	Diseño de Solución para Subproblema 1	38
3.2.2.	Diseño de Solución para Subproblema 2	38
3.2.3.	Definición de Pruebas	39
3.2.4.	Resumen	40
4.	Desarrollo	41
4.1.	Configuraciones	41
4.1.1.	Configuración Pixhawk	41
4.1.2.	Configuración MultiWii	43
4.2.	Desarrollo API para Pixhawk	46
4.2.1.	MAVProxy	46

4.2.2.	Cliente Pixhawk	47
4.2.3.	Funcionalidades	48
4.2.4.	Agregar nuevas funcionalidades	50
4.3.	Desarrollo API para MultiWii	53
4.3.1.	API MultiWii	54
4.3.2.	Funcionalidades	54
4.3.3.	Agregar nuevas funcionalidades	55
4.4.	Desarrollo API Controlador	59
4.4.1.	API Controlador y su relación con Pixhawk y MultiWii	60
4.4.2.	Funcionalidades	61
4.4.3.	Agregar nuevas funcionalidades	61
4.5.	API General	62
4.6.	Resumen	63
5.	Pruebas	65
5.1.	Pruebas en MultiWii	65
5.1.1.	Prueba 1	65
5.1.2.	Prueba 2	67
5.2.	Pruebas en Pixhawk	69
5.2.1.	Prueba 1	69
5.2.2.	Prueba 2	70
5.3.	Resumen	71
6.	Conclusión	73
6.1.	Del estudio teórico	73
6.2.	Del desarrollo	74
6.3.	De los objetivos trazados	75
6.4.	Del aprendizaje personal	75
6.5.	De las contribuciones	76
6.6.	Del trabajo futuro	76
	Glosario	78
	Bibliografía	80

Anexos

A: API Pixhawk	84
A.1. Funcionalidades	84
A.1.1. initConnection	84
A.1.2. endConnection	84
A.1.3. error	85
A.1.4. connecting	85
A.1.5. getAlt	85
A.1.6. setAlt	86
A.1.7. setVelocity	87
A.1.8. wpClear	87
A.1.9. wpList	87
A.1.10. wpLoad	88
A.1.11. wpUpdate	89
A.1.12. wpSave	89
A.1.13. wpRemove	90
A.1.14. getBat	90
A.2. Tabla de Funcionalidades Resumida	91
B: API MultiWii	92
B.1. Funcionalidades	92
B.1.1. getRC	92
B.1.2. getImu	93
B.1.3. getAtt	94
B.1.4. getAlt	95
B.1.5. getWP	95
B.1.6. getIdent	97
B.1.7. getNavWP	98
B.1.8. sendRC	100
B.1.9. setWP	101
B.1.10. setNavWP	102
B.2. Tabla de Funcionalidades Resumida	104

C: API Controlador	105
C.1. Funcionalidades	105
C.1.1. addWP	105
C.1.2. getAltitude	106
C.1.3. init	106
C.1.4. clearWP	107
C.1.5. close	107
C.2. Tabla de Funcionalidades Resumida	108

ÍNDICE DE FIGURAS

	página
2.1. MultiWiiPro, extraída desde [4].	16
2.2. PixHawk, extraída desde [10].	17
2.3. Evolución de BeagleBone, extraída desde [35].	18
2.4. Arquitectura con supervisión humana, extraída desde [19].	19
2.5. Control cíclico jerárquico para un UAV, extraída desde [19].	19
2.6. Arquitectura basada en servicios USAL, extraída desde [31].	21
2.7. ISP, extraída desde [26].	25
2.8. Estructura de comunicación serial, extraída desde [36].	29
2.9. Cabecera paquete MAVLink de 8 bytes, extraída desde [7].	31
2.10. Sistema integrado ROS MAVLink, extraída desde [18].	32
3.1. Problema.	35
3.2. Subproblema 1.	36
3.3. Diseño general.	37
3.4. Diseño general del Subproblema 1.	38
3.5. Diseño general del Subproblema 2.	39
4.1. Pines Beaglebone Black, extraída desde [14].	42
4.2. Pines para conexión serial MultiWii, extraída desde [12].	45
4.3. Conexión entre MultiWii y Beaglebone Black.	46
4.4. getAlt.	47
5.1. Resultado obtenido tras la ejecución del script MultiWii.	66
5.2. Resultado obtenido por MultiWii Win Gui.	67
5.3. Resultado obtenido tras la ejecución del script Controlador MultiWii.	68
5.4. Resultado obtenido por MultiWii Win Gui.	69
5.5. Resultado obtenido tras la ejecución del script Pixhawk.	70
5.6. Resultado obtenido por Mission Planner.	70
5.7. Resultado obtenido tras la ejecución del script Controlador Pixhawk.	71
5.8. Resultado obtenido por Mission Planner.	71
B.1. Esquema de la estructura	93

ÍNDICE DE TABLAS

	página
2.1. Comparación entre C, Basic y Pascal, extraída desde [20].	27
4.1. Mapa con puertos UART Beaglebone, extraída desde [2].	42
4.2. Pines seriales 4/5 en Pixhawk, extraído desde [10].	43
4.3. Lista de funcionalidades Pixhawk.	49
4.4. Lista de funcionalidades MAVProxy.	51
4.5. Lista de funcionalidades MAVProxy Continuación.	52
4.6. Lista de funcionalidades MultiWii.	55
4.7. Lista de funcionalidades Controlador.	61
4.8. Características del API General [17].	63
A.1. Lista de funcionalidades Pixhawk Anexo.	91
B.1. Atributos de un Punto de Ruta.	99
B.2. Lista de funcionalidades MultiWii Anexo.	104
C.1. Lista de funcionalidades Controlador Anexo.	108