

Índice general

1. INTRODUCCIÓN	1
1.1. Objetivos.	1
1.2. Justificación del proyecto.	1
1.2.1. Vehículos aéreos no tripulados.	1
1.2.2. Principios de control del vuelo de un helicóptero.	2
1.2.3. El proyecto Hermes.	3
1.3. Alcance del proyecto.	4
2. ARQUITECTURA DEL SISTEMA.	7
2.1. Contexto tecnológico.	7
2.1.1. Helicóptero de radiocontrol.	7
2.1.2. Actuadores.	8
2.1.3. Computador de vuelo.	8
2.1.4. Sistema Operativo.	10
2.1.5. Lenguaje de programación en la aplicación del computador de vuelo.	11
2.1.6. Compilador del computador de vuelo.	12
2.1.7. Sensores.	13
2.1.8. Puente inalámbrico.	15
2.1.9. Batería LiPo	15
2.1.10. Plataforma de vuelo.	16
2.1.11. Caja aviónica	16
2.1.12. Ordenador de tierra	17
2.2. Aplicación en tiempo real en el computador de vuelo.	17
2.2.1. Requisitos.	17
2.2.2. Captura de datos.	18
2.2.3. Comunicación con tierra.	18
3. ESTRUCTURA DEL SOFTWARE EN LA HÉRCULES.	21
3.1. Funciones principales de la aplicación.	21
3.2. Estructura general de la aplicación.	21
3.2.1. Arquitectura del proceso en hilos.	22
3.2.2. Rutas de ejecución en el bucle principal.	23
3.2.3. Interfaz de comunicaciones helicóptero-tierra.	24
4. DESARROLLO DE LA APLICACIÓN DE TIERRA.	27
4.1. MATLAB: Graphical User Interfaces GUIs.	28
4.2. Módulo de inicialización.	30
4.2.1. Código C en computador de vuelo.	30
4.2.2. Código Matlab en ordenador de tierra.	32

4.2.3.	Modificaciones en el código del computador de vuelo.	39
4.3.	Módulo de configuración de datos de red.	41
4.4.	Módulo de realización de experimentos.	43
4.4.1.	Código C en computador de vuelo	43
4.4.2.	Código Matlab en ordenador de tierra.	49
4.4.3.	Modificaciones en el código del computador de vuelo.	53
4.5.	Módulo de gestión de datos y archivos registrados.	66
4.5.1.	Código C en computador de vuelo	66
4.5.2.	Código Matlab en ordenador de tierra	69
4.5.3.	Modificaciones en el código del computador de vuelo.	74
4.6.	Módulo de representación de datos importados.	80
4.6.1.	Representación gráfica de los resultados.	81
4.6.2.	Exportación de datos a una hoja de cálculo.	86
4.7.	Módulo de recepción de datos en tiempo real.	87
4.7.1.	Código C en el computador de vuelo.	87
4.7.2.	Código Matlab en ordenador de tierra.	89
4.8.	Otras funcionalidades.	91
4.8.1.	Vuelta al estado inicial del sistema.	91
4.8.2.	Monitorización de eventos en el helicóptero.	94
4.8.3.	Envío de un informe de errores.	95
5.	Pruebas en laboratorio.	105
5.1.	Conexión en laboratorio de la caja aviónica.	105
5.2.	Fiabilidad en la medida de los sensores.	107
5.2.1.	Sensores de altura.	107
5.2.2.	Unidad de medida inercial.	109
5.2.3.	Calibración de la unidad de medida inercial ante interferencias magnéticas.	112
6.	Servicio de red	117
6.1.	Sistema de arranque de Debian GNU/Linux	117
6.2.	Crear un servicio de red	118
6.3.	Implementación	118
6.3.1.	Inicio con el sistema del programa principal.	118
6.3.2.	Sistema de monitorización del programa.	122
7.	Conclusiones y aplicaciones futuras.	127
7.1.	Resumen y conclusiones.	127
7.2.	Ampliaciones futuras.	130
	Bibliografía	132

Índice de figuras

1.1. Actitud de la aeronave: pitch, roll y yaw.	2
1.2. Detalle del cíclico y el colectivo.	3
1.3. Esquema del sistema de control a implementar en el helicóptero.	5
2.1. Esquema de las tecnologías utilizadas en el proyecto.	7
2.2. Helicóptero X-Cell-Spectra sin carenado.	8
2.3. Detalles sobre los actuadores que controlan un helicóptero.	9
2.4. Placa Hércules EBX.	9
2.5. Esquema de compilación.	12
2.6. Unidad de medida inercial 3DM-GX.	13
2.7. Vistas del potenciómetro dentro de su carcasa.	14
2.8. Sensor de ultrasonido en su carcasa.	14
2.9. Puente ethernet inalámbrico.	15
2.10. Estructura de la plataforma, situada en la azotea del edificio L1 de los laboratorios de la Escuela Superior de Ingenieros de Sevilla.	16
2.11. Vuelo de un helicóptero de radiocontrol sobre una plataforma de pruebas.	17
2.12. Situación de los sensores de altura dentro de la plataforma de pruebas.	18
2.13. Caja aviónica con las conexiones a la izquierda y sin las conexiones a la derecha.	19
2.14. Esquema en Solid Edge de la distribución de los componentes más importantes dentro de la caja.	20
2.15. Esquema en Solid Edge de la fijación de la caja aviónica en el brazo de la plataforma de vuelo.	20
2.16. Protocolos de nivel de enlace para el tramo helicóptero-router y el tramo router-tierra.	20
3.1. Esquema de la estructura en hilos de la aplicación en el computador de vuelo.	25
3.2. Esquema de las distintas funciones que se ejecutan en el bucle principal de la aplicación en el computador de vuelo.	26
4.1. Módulos que conforman la aplicación gráfica de tierra.	29
4.2. Ventana para configuración de parámetros de inicio en la captura de datos de los sensores.	35
4.3. Ventana para configuración de parámetros de los actuadores del sistema.	37
4.4. Ventana gráfica para visualización y modificación de parámetros de conexión (Conf.conexion.fig).	41
4.5. Cuadro para gestión de la realización de los experimentos.	49
4.6. Al iniciar las interrupciones con éxito en el computador de vuelo se desactiva el cuadro de envío de tramas de inicialización.	51
4.7. Elementos del módulo de gestión de datos y archivos.	69
4.8. Elementos del módulo de representación de datos.	80
4.9. Ventana para representación gráfica de resultados de los experimentos.	82

4.10. Representación de datos a escala 1:1, en el rango de valores de los datos importados.	85
4.11. Representación de datos a escala 4:1. Aparecen las barras de desplazamiento que nos permiten desplazarnos a lo largo de todo el rango de valores de los datos importados.	85
4.12. Exportación a hoja de cálculo de datos de un experimento.	86
4.13. Monitorización de los datos del potenciómetro y sensor de ultrasonidos mientras los datos de la UMI son registrados en la cola.	92
4.14. Ejemplo de un informe de errores importado desde el helicóptero.	103
5.1. Detalle de la parte frontal y la trasera de la caja aviónica.	105
5.2. Vistas de la caja aviónica sin la carcasa superior.	106
5.3. Conexión de los sensores de altura a la caja aviónica.	106
5.4. Conexión de resto de periféricos a la caja aviónica.	107
5.5. Comprobación de recepción correcta de datos en tierra.	108
5.6. Sistema de poleas para medición de alturas con el sensor de ultrasonidos.	108
5.7. Movimiento manual de la palanca de giro del potenciómetro para medidas de altura.	109
5.8. Rango completo de alturas medido con el giro manual de la palanca del potenciómetro.	109
5.9. Ejes en torno a los cuales toma la UMI las medidas de pitch, roll y yaw.	110
5.10. En la parte derecha de la gráfica se observa el giro de -90° y $+90^\circ$ en torno al eje y.	110
5.11. Giro de -90° y $+90^\circ$ en torno al eje x.	111
5.12. Giro de -225° aprox. para comprobar el salto de rango $-180^\circ \rightarrow +180^\circ$.	111
5.13. Giro de 180° en torno al eje z obteniéndose valores del yaw inconsistentes.	112
5.14. Giro de -180° en torno al eje z obteniéndose valores del yaw también inconsistentes.	112
5.15. Conexión de la UMI a PC para calibración de campo dentro de la caja aviónica.	114
5.16. Giro de $\pm 100^\circ$ en torno al eje z obteniéndose valores correctos del yaw.	114
5.17. Programa de calibración del software fabricante para la UMI 3DM-GX.	115
6.1. Ventana 'supervision.fig' para gestión del servicio de red.	125
7.1. Interfaz gráfica Matlab en control de tierra.	128
7.2. Ejemplo de cómo sería un sistema de indicadores de los ángulos de Euler para monitorización en tierra de la actitud del helicóptero.	130
7.3. Esquema sobre el intercambio de información que existiría entre tierra, la Hércules, la radio y los actuadores a través del receptor.	131

Lista de códigos fuente

4.1. Estructura principal de parámetros de inicialización en C.	30
4.2. Estructuras de sensores y actuadores en C.	30
4.3. Código del hilo de recepción de trama de inicialización en C.	32
4.4. Código Matlab de definición de estructuras de inicialización.	34
4.5. Función Matlab actualizatrama() para construcción de la trama de inicialización. . .	36
4.6. Envío en Matlab de las estructuras para inicialización de experimentos en el helicóptero.	38
4.7. Indicación a la aplicación de tierra de recepción correcta de la trama de inicialización.	38
4.8. Código adicional C en Hilo21 para la recepción de la trama de actuadores e introduc- ción en estructura 'datos'.	39
4.9. Código adicional C en Hilo21 para el envío de una confirmación de recepción de trama de inicio.	40
4.10. Código Matlab para la carga inicial de los parámetros principales de conexión.	42
4.11. Código Matlab para guardar modificaciones en los parámetros de conexión.	43
4.12. Carga en Matlab de los parámetros de conexión antes de salir de la aplicación.	43
4.13. Código adicional C en Hilo21 para el envío de una confirmación de recepción de trama de inicio.	44
4.14. Código del Hilo22 para recepción en C de directivas mediante n_control.	46
4.15. Código C al inicio de main() en el cual se configuran los parámetros para la captura A/D en la placa.	47
4.16. Código C de la función ad1() de captura de datos por la entrada analógica (potenció- metro).	48
4.17. Código Matlab para inicio de interrupciones en la placa Hércules.	50
4.18. Código Matlab para inicio de introducción de datos de sensores en cola.	52
4.19. Código Matlab para detención de interrupciones en la placa Hércules.	53
4.20. Código C adicional en Hilo22 para envío de confirmaciones en la gestión de los ex- perimentos.	56
4.21. Cambios en el código C de main() para configuración de los parámetros de conversión A/D en la placa Hércules.	57
4.22. Código adicional C en main() para establecer el punto de altura inicial del potenció- metro en la plataforma.	58
4.23. Código C adicional en main() para realización de la autocalibración del sensor de ultrasonidos.	59
4.24. Código C de la función ad1() de adquisición de alturas desde el potenciómetro y el sensor de ultrasonidos.	60
4.25. Función 'medida_potenc()' en C para el cálculo de la altura medida por el potenciómetro.	62
4.26. Función 'medida_us()' en C para el cálculo de la altura medida por el sensor de ultra- sonidos.	66
4.27. Opciones en el bucle principal para la gestión de los datos registrados.	66
4.28. Hilo 'd_disco.c' para paso de datos desde la cola al disco duro.	68

4.29. Código Matlab para paso de los datos registrados en cola al disco duro del computador de vuelo.	69
4.30. Código Matlab para borrado del archivo de datos registrados en la placa Hércules. . .	70
4.31. Código Matlab para recepción del archivo de datos registrados de los sensores. . . .	74
4.32. Modificaciones en el hilo 'd_disco.c' para paso de datos desde la cola al disco duro. . .	76
4.33. Función env_archivo() en C para el envío a tierra de 'datos.dat' a través del puerto tcp/ip 4954.	79
4.34. Código C adicional en Hilo22 para la cancelación del envío del archivo de datos registrados.	80
4.35. Código Matlab de carga e inicialización de variables al abrir la ventana graficas_archivo.fig.	81
4.36. Código Matlab de representación de los datos en sus ejes.	84
4.37. Código Matlab para ejecutar un zoom sobre las gráficas de datos.	85
4.38. Código Matlab para exportación de datos a una hoja de cálculo.	87
4.39. Código C de la función env_d_red() encargada de enviar los datos registrados de los sensores en tiempo real.	89
4.40. Código Matlab para monitorización de datos de sensores en tiempo real.	91
4.41. Código Matlab para recuperación de condiciones iniciales en la aplicación.	94
4.42. Código C de la función env_errores() que crea un informe detallado de errores ocurridos durante un experimento.	99
4.43. Código C de la función env_errores() que envía a tierra por el puerto tcp/ip 4954 el informe detallado de errores ocurridos durante un experimento.	101
4.44. Código Matlab para recepción de un informe de errores creado por la aplicación de vuelo.	103
6.1. Código del archivo <i>/etc/init.d/hercules</i>	122
6.2. Órdenes para el enlazado del archivo 'hercules'.	122
6.3. Definición del nuevo servicio <i>hercules</i> en <i>/etc/services</i>	123
6.4. Inclusión del servicio <i>hercules</i> en <i>/etc/inetd.conf</i>	123
6.5. Código de la función de supervisión 'supervisor.sh' creada para el servicio <i>hercules</i> . .	124
6.6. Código Matlab para envío de mensajes al servicio de red 'hercules'.	126