



INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y
ELÉCTRICA

“ICE – ELECTRÓNICA”

“DISEÑO Y PROTOTIPO DE UN CUADRICÓPTERO DE
EXPLORACIÓN USANDO
EL MICROCONTROLADOR ATMEGA 328”

QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMUNICACIONES Y ELECTRÓNICA

PRESENTAN:

LORENZO CALDERÓN JORGE EDUARDO

JIMÉNEZ PEÑA JAVIER

ASESOR TÉCNICO:

ING. HUGO JORGE MACÍAS PALACIOS

ASESOR METODOLÓGICO:

DR. JESUS ENRIQUE URBANO NORIEGA

MÉXICO D.F. 2015



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA
UNIDAD PROFESIONAL "ADOLFO LÓPEZ MATEOS"

TEMA DE TESIS

**QUE PARA OBTENER EL TÍTULO DE
POR LA OPCIÓN DE TITULACIÓN
DEBERA(N) DESARROLLAR**

**INGENIERO EN COMUNICACIONES Y ELECTRÓNICA
TESIS COLECTIVA Y EXÁMEN ORAL INDIVIDUAL
C. JAVIER JIMENEZ PEÑA
C. JORGE EDUARDO LORENZO CALDERON**

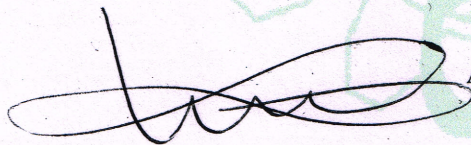
**"DISEÑO Y PROTOTIPO DE UN CUADRICÓPTERO DE EXPLORACIÓN USANDO EL MICROCONTROLADOR
ATMEGA 328"**

**DISEÑAR Y CONSTRUIR UN CUADRICÓPTERO PARA EXPLORAR ZONAS BOSCOSAS O ABIERTAS PARA
LOCALIZACIÓN DE PERSONAS EXTRAVIADAS**

- ❖ OBJETIVO
- ❖ JUSTIFICACIÓN
- ❖ RESUMEN
- ❖ ABSTRAC
- ❖ ÍNDICE CAPITULADO
- ❖ ÍNDICE DE FIGURAS, TABLAS Y GRAFICAS
- ❖ PRESENTACIÓN
- ❖ CAP. 1 HISTORIA DEL CUADRICÓPTERO

- CAP. 2 ANÁLISIS Y SELECCIÓN DE ELEMENTOS
- CAP. 3 DISEÑO Y ENSAMBLAJE DE CUADRICÓPTERO
- CAP. 4 VIABILIDAD DEL PROYECTO
- CONCLUSIONES DE TESIS
- BIOGRAFÍAS
- GLOSARIO
- ANEXOS

MÉXICO, D.F., A 02 DE FEBRERO DE 2016.

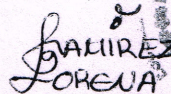


ASESORES

M. EN C. JESUS ENRIQUE URBANO NORIEGA



ING. HUGO JORGE MACÍAS PALACIOS



**ING. PATRICIA LORENA RAMÍREZ RANGEL
JEFE DEL DEPARTAMENTO ACADÉMICO DE
COMUNICACIONES Y ELECTRÓNICA**



OBJETIVO

Diseñar y construir un cuadricóptero para explorar zonas boscosas o abiertas para localización de personas extraviadas.

JUSTIFICACION

No existe un equipo menor de rastreo para reducir el alto índice de extravíos en zonas boscosas o abiertas, donde el equipo ayudara a explorar y buscar a personas extraviadas.

RESUMEN

En el presente trabajo de tesis se desarrolla el diseño y ensamblaje y control de un cuadricóptero el cual deberá cumplir la función de explorador de zonas boscosas o abiertas.

En el capítulo 1 se darán las generalidades de los **UAV**, así como una breve reseña histórica de su evolución y desarrollo, las características principales del cuadricóptero comercial que son utilizados como plataforma para investigación y los fundamentos básicos de los dispositivos usados para el control. De igual forma se hará una reseña de la búsqueda de personas extraviadas en las zonas boscosas y en zonas abiertas así como la prevención de extravió en estos lugares.

En el capítulo 2 se muestra la forma de comunicar el cuadricóptero con una estación fija para realizar el control del mismo de forma remota, los algoritmos matemáticos y su implementación de control y la ubicación del mismo que veremos más adelante en el capítulo 3.

En el capítulo 3 se detalla el diseño del software y el diseño electrónico del cuadricóptero, así como el acondicionamiento de la instrumentación, además se muestra las partes constitutivas del mismo.

En el capítulo 4 se muestra las diferentes pruebas y simulaciones realizadas para comprobar el correcto funcionamiento de exploración, de maniobrabilidad y control sobre el cuadricóptero, de tal forma su desarrollo y desempeño que tendrá en un futuro.

ABSTRACT

In this thesis the design, assembly and control a quadricopter which should play the role of explorer wooded or open areas develops.

In Chapter 1 an overview of the UAV will be given as well as a brief history of its evolution and development, the main characteristics of commercial quadricopter that are used as a platform for research and the basics of the devices used for control. Similarly there will be a review of finding people lost in the wooded areas and open areas and preventing lost in these places.

Chapter 2 describes how to communicate with a fixed station quadricopter for control of it remotely, mathematical algorithms and implementation of control and location that will see later in Chapter 3 is shown.

In Chapter 3 the design of software and electronic design quadricopter and design of instrumentation is detailed further the constituent parts of miso shown.

In Chapter 4 the different tests and simulations to verify the correct operation of exploration, maneuverability and control over the quadricopter, so its development and performance will be shown in the future.

INDICE GENERAL

Objetivo	I
Justificación	II
Resumen	III
Abstract	IV
Índice capitulado	V
Índice de figuras, tablas y graficas	VIII
Presentación	XI
Capítulo 1.	
Historia del cuadricóptero	1
1.1 Vehículo aéreo no tripulado	1
1.1.1 Definición	1
1.1.2 Historia	2
1.1.3 Características	4
1.1.4 Descripción	4
1.1.5 Clasificación	5
1.2 Cuadricóptero o Quadrotror	7
1.2.1 Definición	7
1.2.2 Historia del cuadricóptero	8
1.2.3 Principales características	10
1.2.4 Principio de funcionamiento	11
1.2.5 Arquitectura de un cuadricóptero	13
1.2.5.1 Sensores	13
1.2.5.2 Unidad de control	13
1.2.5.3 Etapa de potencia	14
1.3 Tipos de cuadricópteros usados como plataforma de investigación	15
1.3.1 Cuadricóptero Hummingbird de Ascending Technologies	15
1.3.2 Draganflyer X4	16
1.3.3 Diy drones arducopter	16
1.4 Búsqueda de personas extraviadas	18
1.4.1 Principio de búsqueda	18
1.4.2 Estrategias de búsqueda	18
1.4.3 Composición de un equipo de búsqueda	19
1.4.4 Tipos de búsqueda	20
1.4.5 Métodos de rescate	20

Capítulo 2.

Análisis y selección de elementos

25

2.1 Generalidades de los principales dispositivos electrónicos a emplearse	25
2.1.1 Dinámica	25
2.1.1.1 Movimiento de guiñada (yaw)	26
2.1.1.2 Movimiento de inclinación (pitch)	26
2.1.1.3 Movimiento de bamboleo (roll)	27
2.1.1.4 Movimiento vertical	28
2.2.1 Microcontroladores	29
2.2.1.1 Características de microcontrolador atmega 328	30
2.2.1.2 Módulo Bluetooth HC-05	31
2.2 Partes constitutivas del cuadricóptero	32
2.2.1 Motores	32
2.2.2 Etapa potencia / Driver	34
2.2.3 Alimentación	35
2.2.4 Cámara frontal	37
2.2.5 Hélices	38
2.3 Unidad de control	39
2.3.1 Sensores	40
2.3.1.1 Giroscopio	40
2.3.1.2 Acelerómetro	41
2.3.2 ESC: Electronic Speed Controler	41

Capítulo 3.

Diseño y ensamblaje de cuadricóptero

43

3.1 Estructura	43
3.2 Motores	45
3.3 Aspas	46
3.4 ESC	47
3.5 Bateria	48
3.6 Unidad de control	49
3.6.1 Microcontrolador	49
3.6.2 Sensores	49
3.6.3 Modulo Bluetooth	50
3.7 Peso total y por partes del cuadricóptero	51
3.8 Cálculos	51
3.8.1 Cálculos circuito	53
3.9 Esquemático y PCB del proyecto	66
3.9.1 Esquemático	66
3.9.2 PCB	66
3.10 Programa de control	67
3.11 Aplicación con Android	72

3.12 Controlador PID	74
3.13 Desarrollo	75
Capítulo 4.	
Viabilidad del proyecto	80
4.1 Precios de mercado	80
4.2 Costos del proyecto	80
4.2.1 Precio de venta	83
4.2.2 Servicio de mantenimiento y refacciones	84
4.3 Adaptaciones especiales a necesidades del cliente	84
4.4 Mejoras y proyectos futuros	85
4.4.1 Mejoras	85
4.4.2 proyectos futuros	86
Conclusiones de tesis	88
Bibliografías	89
Glosario	91
Anexos	95

INDICE FIGURAS

CAPITULO 1

1.1	Tipos de drones	2
1.2	Breguet de Autogiro 1907	3
1.3	Univ. 2011 Gamera	3
1.4	UAV micro	4
1.5	Sistema Quadrotor	8
1.6	Vista superior del cuadricóptero	8
1.7	Aparato de los hermanos Richet	8
1.8	Cuadricóptero diseñado por George De Bothezat	9
1.9	Convertawings Modelo A	9
1.10	Quad Tiltrotor de la compañía BELL	9
1.11	Moller Skycar	10
1.12	Ángulos de yaw, pitch y roll de un cuadricóptero común	11
1.13	Comportamiento de rotores	12
1.14	Arquitectura genérica del cuadricóptero	13
1.15	Arquitectura genérica del cuadricóptero	14
1.16	AscTec Hummingbird	15
1.17	Draganflyer X4	16
1.18	ArduCopter	17
1.19	Búsqueda de personas extraviadas	19
1.20	Búsqueda por espiral	21
1.21	Búsqueda por rumbo invertido	22
1.22	Búsqueda por abanico	23

CAPITULO 2

2.1	Arquitectura genérica del cuadricóptero	25
2.2	Grados de libertad de un cuadricóptero	26
2.3	Movimiento de yaw	26
2.4	Movimiento de pitch	27
2.5	Movimiento de roll	27
2.6	Movimiento vertical	28
2.7	Esquema de un microcontrolador	29
2.8	Esquema de ATMEGA 328	30
2.9	Placa de Arduino Uno con microcontrolador ATMEGA 328	31
2.10	Bluetooth HC-05	32
2.11	Motor Brushless	33
2.12	Motor DC	34
2.13	Señal PWM	35
2.14	Batería LiPo	37
2.15	Paso y Diámetro de una hélice	38
2.16	Hélice del tipo normal	39
2.17	Hélice del tipo pusher	39
2.18	Giroscopio	40
2.19	Ejemplo ESC	42

CAPITULO 3

3.1	Esqueleto cuadricóptero	45
3.2	Motor brushless	46
3.3	Aspas	47
3.4	ESC para control	48
3.5	Batería LiPo	48
3.6	ATmega 328	49
3.7	Sensor MPU 6050	50
3.8	Modulo bluetooth HC-05	51
3.9	Diodo zener como regulador de voltaje 5.1 volts	54
3.10	Diodo zener como regulador de voltaje 3.3 volts	55
3.11	Función cuadricóptero	57
3.12	PCB	66
3.13	Presentación de aplicación	73
3.14	Programa de control remoto del cuadricóptero	74
3.15	Esquema PID	75
3.16	Simulación estabilidad 1	75
3.17	Simulación estabilidad 2	76
3.18	Prueba estabilidad motores	76
3.19	Prueba estabilidad motores con balancín	77
3.20	Armado 1	77
3.21	Armado 2	77
3.22	Conexión ESC	78
3.23	ESC con placa de alimentación	78
3.24	Termino de armado	79
3.25	Prueba de motor sin hélices	79
3.26	Prueba de motor con hélices	79

CAPITULO 4

4.1	Equipo y refacciones de un cuadricóptero	84
4.2	Cuadricóptero adaptado	85
4.3	Cámaras Go Pro	85
4.4	Módulo Xbee	86
4.5	Cuadricóptero a futuro	87

INDICE TABLAS

CAPITULO 1

Tabla 1.1	Clasificación de los UAV	6
Tabla 1.2	Clasificación de los UAV's según el tipo de despegue	7
Tabla 1.3	Cuadro comparativo entre cuadricópteros comerciales	17

CAPITULO 2

Tabla 2.1	Características ATMEGA 328	31
Tabla 2.2	Comparativa baterías	37

CAPITULO 3

Tabla 3.1 Pesos del cuadricóptero	51
Tabla 3.2 Calibre cable	64
Tabla 3.3 Anchos de pista	64

CAPITULO 4

Tabla 4.1 Gastos unitarios	81
Tabla 4.2 Gasto por pieza	82
Tabla 4.3 Envíos de material	83
Tabla 4.4 Gastos indirectos	83

INDICE DIAGRAMAS

Diagrama N°1 Partes del cuadricóptero	43
Diagrama N°2 Esquemático de placa	66

INDICE FORMULAS

CAPITULO 2

Formula 2.1	35
-------------	----

CAPITULO 3

Formula 3.1	52
Formula 3.2	52
Formula 3.3	52
Formula 3.4	52
Formula 3.5	52
Formula 3.6	53
Formula 3.7	53
Formula 3.8	53
Formula 3.9	54
Formula 3.10	54
Formula 3.11	54
Formula 3.12	55
Formula 3.13	55
Formula 3.14	56
Formula 3.15	56
Formula 3.16	56

INDICE GRAFICAS

Grafica 1	58
Grafica 2	59
Grafica 3	60
Grafica 4	61
Grafica 5	62
Grafica 6	63

PRESENTACIÓN

En estos años el cuadricóptero se ha convertido en la plataforma estándar para la investigación de micro vehículos aéreos y proyectos de robots aéreos. Esto se ha logrado gracias a la simplicidad y efectividad del diseño de un cuadricóptero, el abaratamiento de costos e incrementos en el desempeño y capacidades de los componentes necesarios para su construcción.

Un cuadricóptero es un helicóptero pequeño, generalmente a escala de 30 cm a 1 m de diámetro, con 4 propulsores ordenados en pares de configuración de cruz (+) o de equis (x).

Estas aeronaves pueden ser autónomas o remotamente operadas, una computadora o micro controlador a bordo y una variedad de sensores que incluyen acelerómetros, giroscopios y magnetómetros son los encargados de mantener la estabilidad del cuadricóptero del aire.

Es por esto que se ha escogido este vehículo aéreo, para implementar una forma de exploración segura y eficaz en zonas boscosas o abiertas en las que son difíciles de explorar.

CAPITULO I

HISTORIA DEL CUADRICOPTERO

En este capítulo se hará un resumen de lo que son las naves aéreas no tripuladas, así como también se hablará del cuadricóptero, acerca de lo que son, sus orígenes como su historia, el desarrollo tecnológico que ha tenido a lo largo de los años, su funcionamiento las partes que los conforman a grandes rasgos.

1.1 VEHÍCULO AÉREO NO TRIPULADO

1.1.1 DEFINICION

Un vehículo aéreo es aquel equipo mecánico que se puede mover sobre diferentes altitudes de la superficie terrestre y que pueda portar o transportar un objeto o sistema. De igual forma podemos decir que es una aeronave que vuela sin tripulación humana a bordo, que pueden ser controlados desde una estación base o tener un funcionamiento autónomo mediante un algoritmo preestablecido [1].

Un avión no tripulado es definido con los términos siguientes: vehículo aéreo no tripulado (UAV), avión operado de forma remota (ROA), y vehículo piloteado de forma remota (RPV).

Este tipo de vehículos han sido utilizados en aplicaciones militares tales como reconocimiento de terreno y ataque. En la actualidad nos podemos encontrar con situaciones en las que la visión aérea sería de gran ayuda, por ejemplo en la detección de incendios, control policial en situaciones de riesgo, reconocimiento de desastres naturales [2].

Como todos los campos, se pueden clasificar los UAVs utilizando multitud de características. Tamaño, autonomía, forma de sustentación o utilidad serían las más acertadas [3]. La figura 1.1 muestra algunos UAVs los cuales son desde un precio muy accesible como pequeños juguetes de radio control, hasta grandes proyectos militares.

[1] Monitoreo y control en tiempo real de un vehículo aéreo no tripulado, "Adriana Pérez Pedraza, Miguel Alejandro Ruiz de los Santos"

[2] <http://eleconomista.com.mx/seguridad-publica/2011/03/16/que-son-drones-o-aviones-no-tripulados>

[3] Desarrollo de un cuadricóptero operado por ROS, Iván Monzón, Proyecto de fin de Carrera



Figura 1.1 Tipos de drones

1.1.2 HISTORIA

El desarrollo de los UAV se puede rastrear incluso hasta poco después de la I Guerra Mundial, siendo usados durante la II Segunda Guerra mundial con el fin de entrenar a los soldados que operaban los cañones antiaéreos. En aquellos tiempos de guerra, cuando los austriacos atacaron la ciudad de Venecia, globos no tripulados cargados con explosivos fueron controlados desde una estación remota con el fin de evitar pérdidas humanas durante el ataque. Los resultados, sin ser los esperados, dejaron como precedente el aumento de capacidades a través de estos ingenios recién creados. Paralelamente se empezó a experimentar con bombas aéreas no tripuladas [1].

Gracias a los avances en telecomunicaciones, particularmente en cuanto a la obtención, manejo y transmisión de información directamente de los campos de batalla, lo que permite conseguir comunicaciones mucho más seguras y difíciles de interferir, se empezó la construcción de aviones radio controlados para su uso en los campos de batalla.

Los UAV han demostrado en diferentes escenarios y especialmente en la guerra del Golfo y en la Guerra de Bosnia, el gran potencial que pueden tener, en cuanto a la obtención, manejo y transmisión de la información, gracias a la aplicación de nuevas técnicas de protección de la misma. El ejército estadounidense se dio cuenta de la importancia y necesidad de emplear este tipo de plataformas, con la operación Tormenta del desierto en 1991, la armada estadounidense utilizó el sistema UAV (Pioneer Israelí) para suministrar inteligencia a nivel táctico, en Afganistán durante la operación Paz Duradera el sistema UAV Predator realizó misiones de reconocimiento armado y en el 2003 en Irak atacó objetivos de gran valor para la coalición [2].

[4] R. MAYORGA. "Sistema de navegación para vehículos Aéreos Cuadricópteros", Proyecto de titulación en Ingeniería Técnica Aeronáutica, Universidad Politécnica Superior de Cataluña. España. 2009.

[5] "Unmanned Aerial Vehicles News", <http://defenseindustrydialy.com>



Figura 1.2 Breguet de Autogiro 1907^[6]

En los primeros años del siglo XX, la meta de cada inventor aeronáutico fue de levantar a una persona desde el suelo con un aparato más pesado que el aire, y permanecer el máximo tiempo posible en el aire.

El primer vehículo aéreo no tripulado fue construido por Lawrence y Elmer Sperry en 1916.

Según Leishman ^[6] el Autogiro (figura 1.2). Este quadrotor pionero tenía rotores de 8: 1 metros de diámetro, cada uno que consiste de cuatro palas de tela cubierta de tipo biplano de luz, dando un total de 32 superficies de elevación separadas. La Autogiro hizo un breve y bajo vuelo (según se informa 1: 5 metros sobre el suelo), que era sin duda logrado por el "efecto suelo". Un avión de 578kg con su diseño del rotor y la tecnología rudimentaria disponible necesitaría al menos 50hp entregado, que era el límite de motor de combustión del Autogiro ^[6].

Leishman también afirma que el resultado fue debido a la falta de capacidad de control disponibles, se aseguró "por la ayuda de varios hombres, uno en cada esquina de la estructura en forma de cruz, estabilizantes y tal vez incluso levantar la máquina". Curiosamente un equipo de la Universidad de Maryland, para competir por el Sykorsky premio numero 1, construyó y voló lo que llamaron "Gamera" ^[7], un quadrotor tracción humana, con un diseño inspirado y similar a la Autogiro. El primer vuelo se intentó el 12 de mayo de 2011 y fue un éxito.



Figura 1.3 Univ. 2011 Gamera^[7]

[6] <http://vtol.org/awards/HPHCBooklet.pdf>

[7] <http://www.youtube.com/watch?v=q70tM5sDQhc>

1.1.3 CARACTERÍSTICAS

Algunas características de los vehículos aéreos no tripulados son equivalentes a las de los tripulados (materiales, sistemas de propulsión, aerodinámica, etc.), pero algunas son específicas a estos. Las más importantes son: Sensores de navegación y microprocesadores: los sensores representan uno de los elementos más caros de este tipo de sistemas y son necesarios para la navegación y la ejecución de tareas.

Los procesadores permiten a los UAVs volar misiones enteras de manera autónoma con poca o nula intervención humana.

Sistemas de comunicación: los problemas principales de las tecnologías de comunicación son flexibilidad, adaptabilidad, seguridad y controlabilidad de eje de datos ^[8].

1.1.4 DESCRIPCION

Los vehículos aéreos no tripulados en su concepto tradicional están constituidos de un conjunto de sensores para la adquisición de la información del medio que lo rodea, y transmitirla hacia una estación base tierra. Para cumplir con sus objetivos un UAV debe estar conformado por las siguientes partes.

Plataforma aérea.

Está conformado por el vehículo aéreo propiamente dicho además de su conjunto de sensores para adquirir información del medio y transmitirlo en tiempo real, así como la instrumentación necesaria para mantenerlo estable, como son tarjetas IMU (Inertial Measurement Unit), es decir sensores de tipo inercial como son acelerómetros, giroscopios y magnetómetros que deben de trabajar en conjunto para obtener resultados óptimos. Un sistema de navegación es indispensable para evitar que el UAV pierda su trayectoria, para este fin estos suelen tener dispositivos GPS, cámaras de video inalámbricas entre otros.



Figura 1.4 UAV micro ^[8]

[8] "Blog acerca de UAVs micro" <http://www.flightglobal.com>

Plataforma terrestre.

Está compuesto por una estación de control terrestre, donde se recibe y procesa la información que envía el UAV en tiempo real, además donde se encuentra todo el material y equipamiento preciso para el lanzamiento, recuperación y puesta en operación del sistema. La estación de control, dependiendo de la complejidad del sistema, puede estar conformada desde un computador hasta una compleja red, que cumple las funciones de permitir la preparación y lanzamiento del UAV, guiado al área del banco, recepción y transmisión de imágenes, datos, etc.

1.1.5 CLASIFICACIÓN

Existen varias maneras de clasificar a los vehículos aéreos no tripulados. A continuación se presentan los principales aspectos que consideran para su clasificación:

- Ala fija: se refiere a aeroplanos que requieren un recorrido horizontal para despegar, aterrizar, o un lanzamiento en catapulta. Generalmente tienen largos tiempos de vuelo y pueden volar a altas velocidades.
- Ala rotativa: tienen la ventaja de poder quedarse suspendidos en el aire y de tener alta maniobrabilidad. Estas capacidades los hacen útiles en misiones de robótica, especialmente en aplicaciones civiles. Un UAV de ala rotativa puede tener varias configuraciones, con rotores principales y de cola (helicóptero convencional), rotores coaxiales, multirotores, etc.
- Dirigibles: como globos o zepelines, que son más ligeros que el aire. Tienen largos tiempos de vuelo, vuelan a bajas velocidades y generalmente son de tamaño grande.
- UAVs de aleteo: poseen alas flexibles inspiradas en pájaros y pequeños insectos.

Por el tipo de misión para el que fueron diseñados.

Básicamente se agrupan en los siguientes 6 tipos: De simulación de blancos (para confundir al enemigo), reconocimiento de terreno, combate, de logística (para transportar carga), investigación y desarrollo.

Por su alcance.

Se pueden encontrar los Handheld que no sobrepasan los 600 metros de altura ni los 2 kilómetros de distancia o los Orbitales que vuelan en órbitas bajas terrestres a altas velocidades (este tipo de vehículo se encuentra en etapa de estudio).

Christian Nadales Real de la Universidad Técnica de Cataluña realiza la siguiente tabla comparativa:

CATEGORIA	ACRÓNIMO	ALCANCE (km)	AUTONOMÍA (h)	ALTITUD DE VUELO (m)	CARGA MÁXIMA (kg)
Micro <250 gr	Micro	<10	1	250	<5
Mini<25 kg	Mini	<10	<2	150 y 300	<30
Alcance cercano	CR	10 a 30	2 a 4	3000	150
Alcance corto	SR	30 a 70	3 a 6	3000	200
Alcance medio	MR	70 a 200	6 a 10	5000	1250
Altitud baja	LADP	>250	0.5 a 1	50 a 9000	350
Altitud media	MRE	>500	10 a 18	8000	1250
Autonomía alta Altitud baja	LALE	>500	>24	3000	<30
Autonomía alta Altitud media	MALE	>500	24 a 48	14000	1500
Autonomía alta Altitud alta	HALE	>2000	24 a 48	20000	12000
Combate	UCAV	1500	2	10000	10000
Ofensivo	LETH	300	3 a 4	4000	250
Señuelo	DEC	500	4	5000	250
Estratosférico	STRATO	>2000	>48	2000 y 30000	ND
EXO-estratosférico	EXO	ND	ND	>30000	ND

Tabla 1.1 Clasificación de los UAV ^[9].

La mayoría de UAV's que son empleados para aplicaciones civiles, y los que están disponibles para el público en general se encuentran en las categorías de mini, micro y de alcance cercano.

Por el tipo de despegue.

En esta clasificación se puede identificar: los de despegue vertical, como los helicópteros y dirigibles, y los de despegue no vertical, como aeroplanos y parapente. La tabla 1.2 muestra un resumen de estas aeronaves.

[9] Nadales C., "Control de un Quadrotor mediante la plataforma Arduino", Proyecto de titulación en Ingeniería Técnica de Telecomunicaciones, Universidad Politécnica Superior de Cataluña. España. 2009.

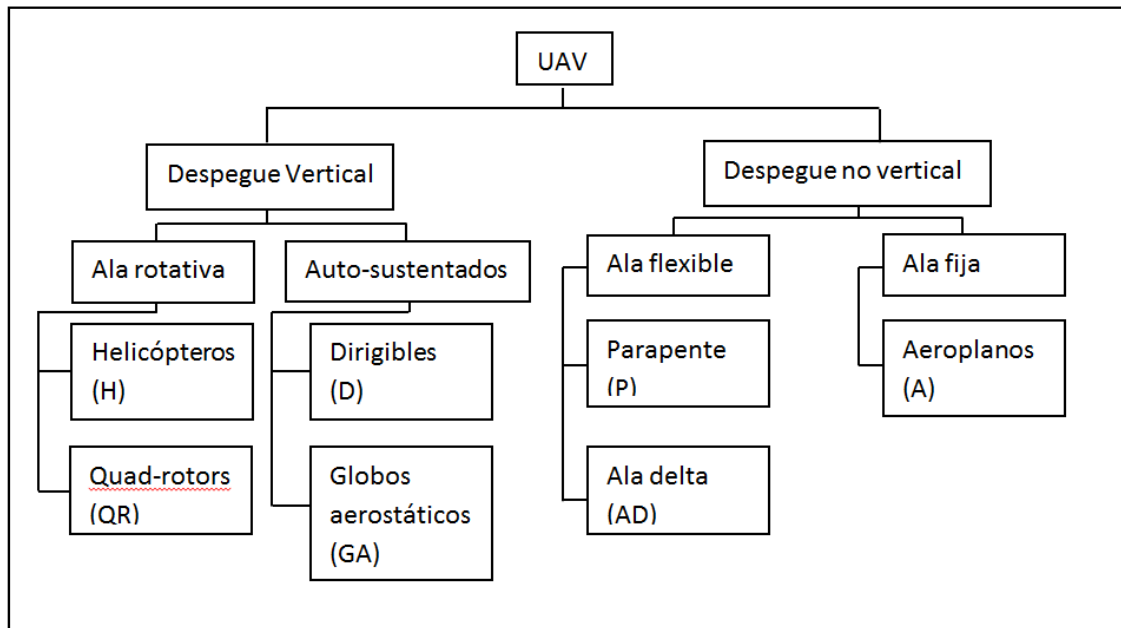


Tabla 1.2 Clasificación de los UAV's según el tipo de despegue ^[9].

Cada uno posee sus propias características, importantes de considerar a la hora de elegir un UAV, así por ejemplo los aeroplanos tienen una gran facilidad en el control y guiado, así como su alta fiabilidad ante condiciones extremas o difíciles, además, dependiendo del tipo de controlador que posea se puede obtener un buen alcance, sin embargo su problema más grande es su falta de maniobrabilidad en espacios reducidos. Los vehículos de hélice rotativa son muy adecuados para tareas de interiores, siempre que sean de dimensiones grandes como por ejemplo un estadio o salas de concierto, a diferencia de los aeroplanos su control es más complejo y no tienen los mismos tiempos de respuesta.

1.2 CUADRICÓPTERO O QUADROTOR

1.2.1 DEFINICIÓN

Un quadrotor se puede definir como una aeronave que se eleva y se desplaza por el movimiento de cuatro motores colocados en los extremos de una estructura en forma de cruz. Normalmente se utiliza el nombre inglés quadrotor aunque también existe la traducción cuadricóptero. Tal como se presenta (figura 1.6), el vehículo dispone de 4 motores con sus palas respectivas, se utiliza la velocidad de los motores para controlar la estabilidad y movimientos del vehículo aéreo.

[9] Nadales C., "Control de un Quadrotor mediante la plataforma Arduino", Proyecto de titulación en Ingeniería Técnica de Telecomunicaciones, Universidad Politécnica Superior de Cataluña. España. 2009.

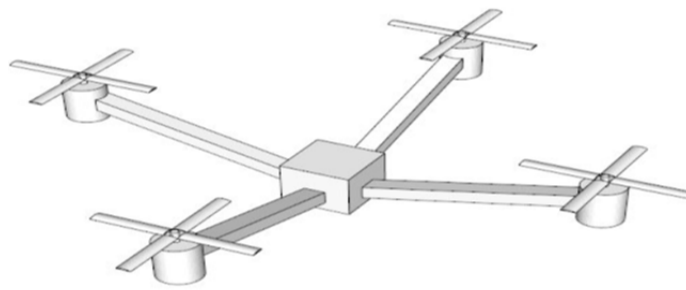


Figura 1.5 Sistema Quadrotor ^[8]

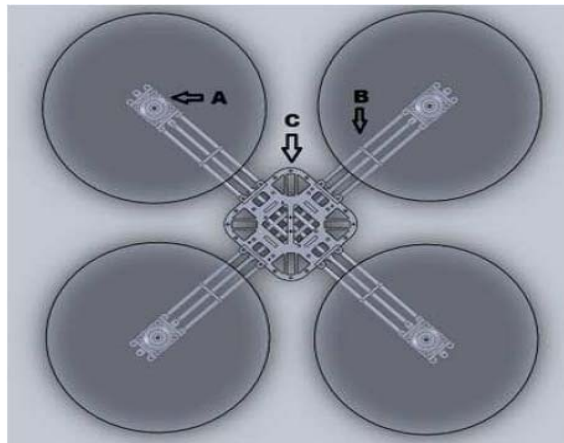


Figura 1.6 Vista superior del quadricóptero

1.2.2 HISTORIA DEL CUADRICÓPTERO

Louis y Jacques Breguet, dos hermanos que trabajaban para el profesor Charles Richet, fueron los primeros en construir un quadricóptero, al cual nombraron Breguet Richet Gyroplane No. 1" (figura 1.7). La primera demostración de vuelo fue conseguida el 29 de septiembre de 1907.



Figura 1.7 Aparato de los hermanos Richet ^[10]

[10] sinvestavtesisdequadrotor.PDF

Después, dos diseños adicionales fueron desarrollados y se realizaron vuelos experimentales.

El primero, construido por Georges de Bothezat e Ivan Jerome en 1922, tenía los rotores colocados en cada extremo de una estructura en forma de X, (figura 1.8).

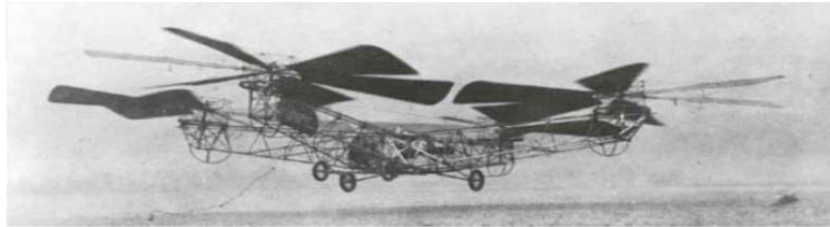


Figura 1.8 Cuadricóptero diseñado por George De Bothezat

Posteriormente en 1922 en Francia se crea un prototipo, llamado Convertawings Modelo A, fue diseñado para uso civil y militar. El diseño tenía dos motores que controlaban cuatro rotores. No tenía rotor de cola, ya que usaba la diferencia en las velocidades de giro para lograr el desplazamiento. La figura 1.9 muestra una imagen del vehículo.

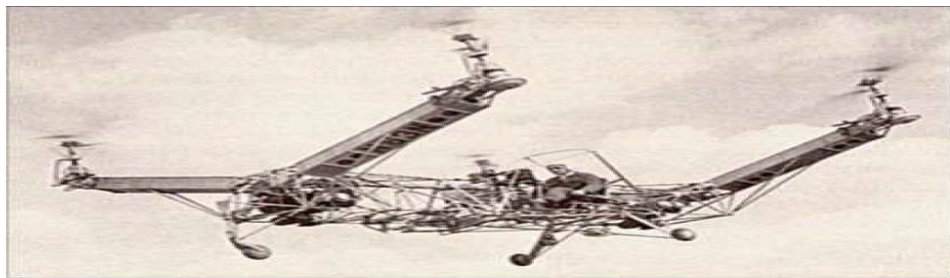


Figura 1.9 Convertawings Modelo A ^[11]

Con el advenimiento de una nueva tecnología se ha incrementado el interés en el diseño de cuadricópteros, la empresa Bell diseñó Quad-Tiltrotor, (Figura 1.10) capaz de llevar una gran carga útil, alcanzando altas velocidades, emplea un pequeño espacio tanto para el despegue como para el aterrizaje y puede modificar la dirección de sus cuatro rotores, hacia adelante para tener un vuelo vertical como un avión convencional ^[12].

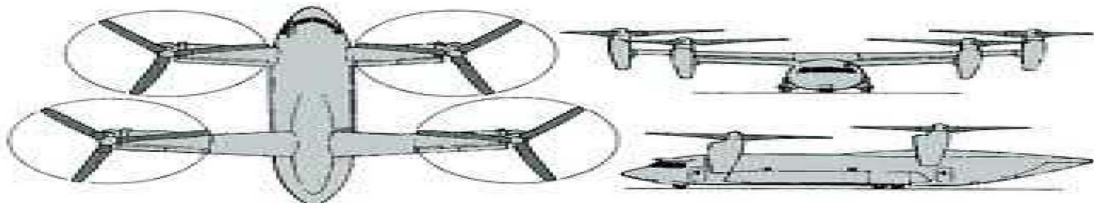


Figura 1.10 Quad Tiltrotor de la compañía BELL ^[13]

[11] <http://www.autogiroescuela.com>

[12] <http://www.escriint.cl/php/contenido/UAV%20CORREGIDO.pdf>

[13] <http://forum.nationstates.net>

El Moller Skycar el cual (figura 1.11), es un famoso cuadricóptero prototipo, a manera de auto volador. Consta de cuatro rotores en forma de ductos de ventilación que permiten una operación segura y eficiente a bajas velocidades. Su inventor Paul Moller lo intentó vender a través de una subasta, sin éxito alguno. Actualmente se ha enfocado en un modelo mejorado el M200G Volantor, que consta de 8 ductos de aire controlados por computador, capaz de levantarse a 3 metros sobre el suelo ^[14].



Figura 1.11 Moller Skycar

1.2.3 PRINCIPALES CARACTERÍSTICAS

Un cuadricóptero; como se mencionó en su definición; es un vehículo aéreo no tripulado que entra en la categoría de ala rotativa, a nivel “mini”. De acuerdo a la Tabla 1.1 esta categoría tiene las características de alcance menor a 10Km, una altitud menor de 300 m, una autonomía menor a 2 horas con la capacidad de llevar una carga máxima de 5 Kg. Aparte de las características técnicas propias de la categoría a la cual pertenece se pueden destacar las siguientes ^[15].

Maniobrabilidad.

Una de las características a destacar es la gran maniobrabilidad que posee este tipo de vehículo. Al disponer de cuatro motores el control es bastante exacto, lo que ayuda a utilizarlo en aplicaciones donde la exactitud de vuelo estacionario sea muy importante. Una aplicación donde se aprecia esta característica es en la navegación de interiores y sitios de espacio reducido.

Capacidad de vuelo vertical.

Como en el helicóptero, estos vehículos disponen de una capacidad de vuelo vertical que los hacen únicos, esta función es ventajosa cuando no queremos tener mucha velocidad horizontal y cuando queremos tener una buena capacidad de vuelo estacionario, lo que ayuda a elegir este tipo de sistemas cuando se quieren adquirir datos desde el vehículo ^[15].

[14] <http://moller.com>

[15] <http://iapss.scenebeta.com/noticia/ar-drone-el-cuadrocoptero-teledirigido-por-tuiphone-touch-ipad>

Control.

El problema fundamental de los quadrotores es su control. El sistema debe de incorporar mecanismos de estabilización para ayudar a la navegación. La **capacidad de carga** es bastante alta con relación al peso de todo el sistema (5 kg / 25 kg), podemos encontrar vehículos que soporten una carga superior al peso que tienen. Esta característica hace posible el incorporar un gran número de sensores.

Autonomía.

Una de las características más importantes a tener en cuenta en los sistemas de vuelo es la autonomía. La autonomía de vuelo no suele ser muy buena ($< 2h$), esta fue una de las limitaciones por la que los UAV tardaron un cierto tiempo en avanzar. Actualmente se están realizando avances importantes en las baterías, proporcionando más capacidad (mAh) y reduciendo los tamaños.

1.2.4 PRINCIPIO DE FUNCIONAMIENTO

En el quadricóptero en cada rotor es responsable de una cierta cantidad de torque y empuje en su centro de rotación. Sus hélices no son todas iguales, en efecto se dividen en dos pares: dos de empuje y dos de desplazamiento, que trabajan en contra rotación. Como consecuencia él, el torque neto resultante puede ser nulo si todas las hélices giran con la misma velocidad angular, permitiendo que el vehículo se mantenga estable en el aire alrededor de su centro de gravedad ^[16].

Similar a un helicóptero tradicional, un quadricóptero es un sistema de seis grados de libertad (x, y, z, pitch, roll y yaw), multivariable, fuertemente acoplado y subactuado.

Las fuerzas principales y momentos que actúan en un quadricóptero son producidas por sus rotores. Dos pares de motores rotan en direcciones opuestas para balancear el torque total del sistema. La figura muestra un diagrama de un quadrimotor típico.

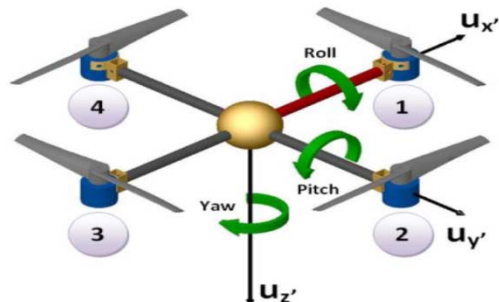


Figura 1.12 Ángulos de yaw, pitch y roll de un quadricóptero común ^[17].

[16] A. Barrientos, J Del Cerro, R. San Martín, C. Rossi, "Vehículos aéreos no tripulados para uso civil", Proyecto Titulación, Universidad Politécnica de Madrid. España.2012

[17] <http://techmez.com>

Como se puede observar, se utilizan dos sistemas de referencia: el sistema T fijo a la Tierra y el sistema B fijo al vehículo.

Un cuadrirotor es controlado, manipulando las fuerzas de empuje de los rotores, así como balanceando el torque entre ellos. Para suspenderlo en el aire, todos los rotores aplican una fuerza constante (figura 1.13). Para controlar el movimiento vertical, la velocidad de los motores se incrementa o decrementa simultáneamente. De este modo se obtiene una mayor o menor fuerza de empuje, pero sin afectar el balance.

Comportamiento de los rotores

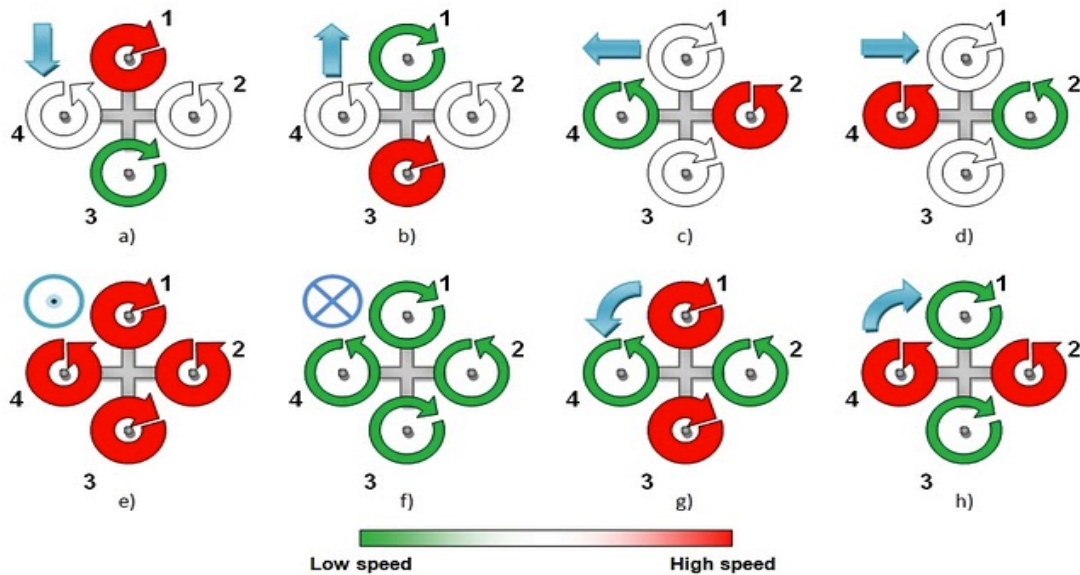


Figura 1.13 Comportamiento de rotores

Un cambio de ángulo de pitch se produce por la variación de velocidades de los motores 1 y 3, disminuyendo la velocidad angular del motor 3 y aumentando la del 1 o viceversa, se tendrá como resultado un movimiento de traslación hacia adelante o atrás. De la misma manera si se varía la velocidad angular de los motores 2 y 4 (figura 1.13) se producirá una variación en el ángulo roll, teniendo como resultado un movimiento de traslación hacia la derecha o hacia la izquierda.

El movimiento de yaw, o alrededor de su propio eje, es inducido por un desbalance aerodinámico de los torques, si se incrementa la velocidad de los motores 1 y 3 y se disminuyen la de los motores 2 y 4, se produce el desbalance mencionado y el cuadricóptero gira en torno a su propio eje en sentido anti horario. De esta manera cambiando estos tres ángulos se tendrá la capacidad de dirigir al vehículo en cualquier dirección y posicionarlo en un punto con gran precisión.

1.2.5 ARQUITECTURA DEL CUARICÓPTERO

Para analizar la arquitectura del cuadricóptero se lo dividirá en bloques.



Figura 1.14 Arquitectura genérica del cuadricóptero

1.2.5.1 SENSORES

La instrumentación que permitirá leer las magnitudes físicas involucradas para mantener estable y dar movimiento el cuadricóptero y además convertirlas a un valor digital para poder controlar los motores son los siguientes:

- Acelerómetro: mide la aceleración asociada a la variación de posición.
- Giroscopio: mide la aceleración angular.
- Magnetómetro: o brújula digital, empleando para no perder la orientación.
- Sensores ultrasónicos: permiten detectar y estimar la proximidad de un objeto, generalmente son usados tanto para el despegue como para el aterrizaje, y estimar la altura de vuelo.

La señal de los sensores es de tipo analógica (variaciones de voltaje), para poderlas procesar es necesario digitalizarlas a través de un convertor A/D. Un microprocesador será el encargado de procesar señales digitales y con los resultados calcular la respuesta que se debe enviar a los motores. Este conjunto forma un lazo cerrado pues la realimentación se produce a través del entorno físico por el que se mueve, variando sus magnitudes dependiendo de los requerimientos de altura o posición que sean demandados.

1.2.5.2 UNIDAD DE CONTROL

Los UAVs y en especial los cuadricópteros requieren de sensores inerciales para controlar tanto los movimientos como la estabilidad. Los datos provenientes de los sensores son procesados en varios microprocesadores de alta velocidad para evitar centralizar el control del sistema, ganando así velocidad de respuesta.

El uso de cámaras resulta atractivo hoy en día, los controles basados en visión artificial son temas de investigación en muchos laboratorios del mundo así por ejemplo Michael Moguenson del “Tufts University Center for Engineering Education and Outreach” realizó su tesis previo a la obtención del título de “Master of Science” de control de seguimiento e identificación de rostros con un cuadricóptero utilizando visión artificial en LabView.

Tanto la parte sensorial como el sistema de video integrado al cuadricóptero actúan en conjunto para brindar estabilidad y controlabilidad al sistema, incluso se han desarrollado colaboradores automáticos para hacerlo más fácil de manejar y hacerlo atractivo a usuarios inexpertos.

1.2.5.3 ETAPA DE POTENCIA

Los cuadricópteros generalmente empelan motores tipo brushless que se pueden traducir como “sin escobillas” para dar propulsión al sistema. Las escobillas son elementos que hacen contacto en el colector de un motor común. En los motores de DC más pequeños, son de una aleación de cobre y en motores más grandes son de un compuesto a base de carbón.

Los motores brushless en vez de funcionar con DC funcionan con AC, la mayoría de alimentan con una señal trifásica, esta señal idealmente debería ser sinusoidal, pero en la práctica son pulsos, sin embargo se los clasifica como motores de DC por que al igual que los motores comunes tienen imanes permanentes. Los imanes son atraídos por la polaridad de un campo magnético generado en las bobinas, las cuales como se explicó anteriormente reciben pulsos en un patrón específico. Si se desea que el motor gire más rápido, simplemente se hace girar al campo magnético secuencial a mayor velocidad, es decir sería necesario aumentar la frecuencia de pulsos. Un control de velocidad adecuado de los motores permitirá al cuadricóptero desplazarse en cualquier dirección.



Figura 1.15 Arquitectura genérica del cuadricóptero ^[18]

[18] <http://www.e-radiocontrol.com.ar>

1.3 TIPOS DE CUADRICÓPTEROS USADOS COMO PLATAFORMA DE INVESTIGACION

A lo largo de varios centros de investigación a nivel internacional se está usando cuadricópteros como plataformas aéreas para investigar nuevas y novedosas técnicas de control que van desde un control a distancia, hasta complejas aplicaciones militares cuales requieren de compleja instrumentación y la capacidad para responder rápidamente ante estímulos externos.

La mayoría de plataformas no están disponibles para el público en general, su complejidad implica un elevado costo que los hace poco asequibles y son destinados principalmente a laboratorios de investigación. Los cuadricópteros más usados en laboratorios de investigación se describen en los siguientes puntos.

1.3.1 CUADRICÓPTERO HUMMINGBIRD DE ASCENDING TECHNOLOGIES

El AscTec Hummingbird es el UAV que mejor ha sido desarrollado por Ascending Technologies, una empresa alemana, diseñado para realizar agresivas y rápidas maniobras de vuelo. Su marco robusto y hélices flexibles le permiten realizar aterrizajes bruscos y su estructura simple lo hace fácil de reparar. Esto lo convierte en la perfecta herramienta para investigar nuevas técnicas de control de vuelo y maniobras dinámicas ^[19].

Entre sus características se puede resaltar las siguientes:

- Ligero en peso y ágil, pesa 510gr y es capaz de llevar una carga de hasta 200gr.
- Programable vía AscTec SDK, además que consta de su propio toolkit para Simulink.
- Acelerómetro de 3 ejes, giroscopio de 3 ejes, magnetómetro de 3 ejes, sensor de presión barométrica y GPS.
- Costo en el mercado de aproximadamente \$5000.00.

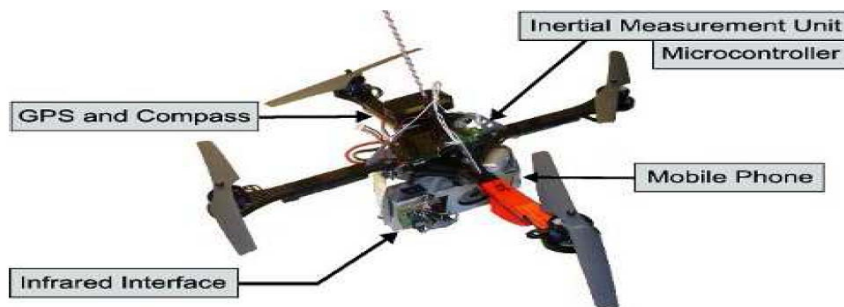


Figura 1.16 AscTec Hummingbird ^[19]

[19] <http://www.asctec.de/uav-applications/research/products/asctec-hummingbird>

1.3.2 DRAGANFLYER X4

El Draganflyer X4 es un UAV que sido diseñado para llevar cámaras de video inalámbricas cuyo propósito es realizar fotos y videos aéreos de alta calidad, esto es posible ya que la cámara HD que lleva bordo es giro estabilizada en los ejes de Pitch con lo que es aislada de la vibración de su estructura.

El UAV posee 7 sensores de vuelo que le permite auto estabilizarse en el aire, como acelerómetros, giroscopios, sensores de presión barométrica, etc. trabajando todo en conjunto con un avanzado control de vuelo que permite al usuario concentrarse al usuario en capturar la imagen deseada, todo se realiza mediante un control remoto de radiofrecuencia. Construida a partir de fibra de carbono de alta calidad, la relación de resistencia a peso de la aeronave es muy alta, junto con un bastidor plegable fuerte ayuda a reducir los daños en caso de aceleración brusca o golpear una obstrucción. Su costo en el mercado bordea en los \$12000.00.



Figura 1.17 Draganflyer X4 ^[20]

1.3.3 DIY DRONES ARDUOPTER.

ArduCopter es una herramienta fácil de configurar y de volar sin embargo sus funciones van mucho más allá que las de un simple helicóptero controlado remotamente, es completo UAV, que ofrece control remoto y vuelo autónomo además de incluir una completa telemetría que se muestra en una poderosa estación terrestre. ArduCopter está en la vanguardia de la robótica aérea y destinada a aquellas personas que quieran probar la tecnología avanzada, las principales técnicas y nuevos estilos de vuelo.

El proyecto ArduCopter se basa el ArduPilot Mega 2, un piloto automático creado por la comunidad DIY Drones. Entre sus características se pueden notar las siguientes:

[20] Brito J., Vissierre D., "Design of a Quadrotor Prototype", proyecto de titulación. Universidad Técnica de Lisboa, Portugal. 2009

1. Control de nivel y de altura de alta calidad que le permite volar nivelado y recto. Posee un modo de “vuelo sencillo”, lo que hace a ArduCopter más fácil de pilotar, el cuadricóptero se orienta automáticamente y se auto nivela permitiéndole al usuario únicamente deslizarlo en el aire en cualquier dirección.
2. No es necesario programar la tarjeta Arduino, mediante una aplicación la PC es posible controlarlo y observar a la vez la interfaz que provee información valiosa como datos de altitud, nivelación, etc.
3. Posee un GPS integrado que permite posicionar al ArduCopter en su interfaz, además de acelerómetros y giroscopios de 3 ejes empleados para su control.
4. Posee una estructura reforzada con aluminio y fibra de carbono la cual protege a los circuitos a bordo de posibles daños en caso de que se presente aterrizajes bruscos.
5. Su costo en el mercado bordea los \$800.

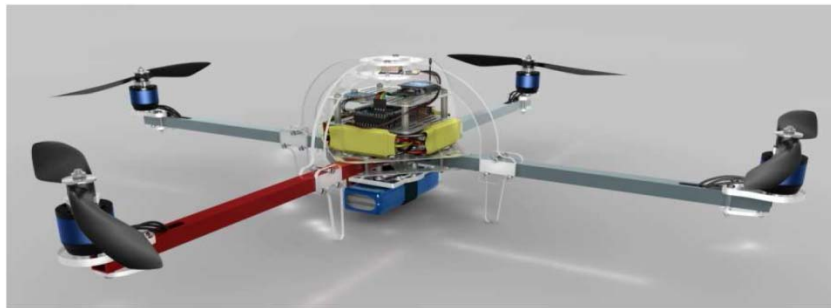


Figura 1.18 ArduCopter ^[21]

Las principales características de los cuadricóptero comerciales se observan en la siguiente tabla:

Cuadricóptero	Características	Precio (U.S.A)
AscTec Hummingbird	Acelerómetro, giroscopio y magnetómetro de 3 ejes, sensor barométrico de presión GPS.	\$5000.00
Dragan Fly	Acelerómetro y giroscopio de 3 ejes, sensor barométrico de presión y sistema autoestabilizador para cámaras.	\$12000.00
DIY Drones ArduCopter	Acelerómetro y giroscopio de 3 ejes y sistema de control Arduino.	\$800.00

Tabla 1.3 Cuadro comparativo entre cuadricópteros comerciales

[21] <http://code.google.com/p/arducopter/wiki/ArduCopter>

1.4 BÚSQUEDA DE PERSONAS EXTRAVIADAS

Uno de los casos que mayor problema causa a los equipos de socorros, son el extravío de personas en bosques y montañas.

Todos los años se producen un gran número de casos de personas que se pierden y que hay que intentar encontrarlas en el menor tiempo posible. Para poder acortar el tiempo es necesario planificar bien cómo vamos a realizar esa búsqueda, una buena planificación puede significar la diferencia entre la vida y la muerte de una persona desaparecida, ya que el tiempo que se tarde en encontrarla es decisivo, sobre todo con condiciones climatológicas adversas o en el caso de los niños.

La planificación se hace en el Puesto de Mando Avanzado, que lo compondrán los jefes de los grupos de búsqueda y rescate, dónde tiene que nombrarse un Jefe de PMA que será el que dará las órdenes oportunas, que es imprescindible seguir [22].

1.4.1 PRINCIPIO DE BÚSQUEDA

Toda búsqueda comienza por una adecuada toma de datos, hay que disponer de cuestionarios que nos permitan recoger la mayor información posible, con ella podremos hacer hipótesis que son de gran ayuda. Es importante recoger los aspectos psicológicos, porque la personalidad del sujeto nos puede indicar que posible decisión tomó al encontrarse perdido y la ruta que haya podido seguir. Por supuesto es imprescindible también contar con la información meteorológica, mapas de la zona, fotografías aéreas, peligros del terreno, etcétera [23].

1.4.2 ESTRATEGIAS DE BÚSQUEDA

La estrategia tiene como objetivo delimitar la zona de búsqueda dónde exista mayor probabilidad de encontrar al sujeto, marcando un área de la menor extensión posible. La extensión vendrá determinada por el tiempo que hace que el sujeto desapareció, distancia que ha podido recorrer y por los recursos disponibles.

[22] Planificacion_de_la_búsqueda_de_personas_desaparecidas.pdf

[23] <http://www.perrosdebusqueda.com>



Figura 1.19 Búsqueda de personas extraviadas

1.4.3 COMPOSICION DE UN EQUIPO DE BUSQUEDA^[24]

En todo equipo de búsqueda y rescate siempre encontraremos un jefe de búsqueda y un jefe de rescate ya que cada una de estas actividades requieren de diferentes habilidades y para ello siempre se debe de designar el mejor en cada una de las especialidades, pero si podemos decir que una característica define a estas dos personas, la serenidad y mente fría, la cual es de suma importancia en pos del resultado final.

Un equipo de búsqueda debe estar compuesto por no menos de tres personas lo ideal es cinco, cada una de ellas cumplirá con una función específica en el desarrollo del operativo.

Los integrantes deben cubrir los siguientes roles:

1. Jefe de grupo: coordina las operaciones entre varios grupos o entre un puesto de comando y un grupo operativo.
1. Guía: es la persona que brújula en mano marca el rumbo tanto de ida como de regreso.
1. Paramédico: o especialista en primeros auxilios.
1. Apoyo: es la persona que transporta provisiones y agua para los rescatados.
1. Asistente: cubre funciones generales generalmente trabaja en la apertura de camino y trochas si el rescate se realiza en monte y si el rescate es en serranías o montaña es el último del grupo y el que brinda seguridad a los demás.

Este es un equipo ideal de trabajo, el operativo también se puede realizar con más gente dividiéndolos en varios grupos de trabajo tomando en cuenta que un grupo mínimo de trabajo son tres personas.

[24] defensacivilmaimon.wordpress.com

1.4.4 TIPOS DE BÚSQUEDA ^[25]

Aquí describiremos los pasos a seguir en un proceso de búsqueda y rescate de personas que se pierden en bosques y montañas.

Generalmente se utilizan varias técnicas de búsqueda, las principales son: búsqueda por abanico, por espiral, por rumbo invertido y búsqueda por rastillaje. Estos procesos en términos generales deben llevarse a cabo en forma independiente.

1.4.5 MÉTODOS DE RESCATE ^[24]

En todo equipo de búsqueda y rescate siempre encontraremos un jefe de búsqueda y un jefe de rescate ya que cada una de estas actividades requieren de diferentes habilidades y para ello siempre se debe de designar el mejor en cada una de las especialidades, pero si podemos decir que una característica define a estas dos personas, la serenidad y mente fría, la cual es de suma importancia en pos del resultado final.

Cuando se comienza un operativo de búsqueda se debe de tener en cuenta los siguientes pasos a seguir:

1. 1. Buscar la mayor información posible:

Qué cantidad de personas que podrían haberse perdido y cuánto tiempo hace que deberían haber llegado.

Nombres, apellidos, edades, capacidades, historial médico, equipo con el que se cuenta.

Último lugar que se los vio, rumbo que tomaron, lugar al que querían llegar.

1. 2. Disponer de logística de apoyo para el grupo de trabajo y la atención posterior a los rescatados.

1. 3. Chequear el equipo que se utilizara en el operativo de búsqueda ya que debe tener el peso justo y por sobretodo cubrir con todas las necesidades de los rescatistas y de los rescatados hasta el momento de ponerlos a salvo.

1. 4. De ser posible buscar la asistencia de lugareños ya que ellos son los que conocen a cabalidad la zona.

[24] defensacivilmaimon.wordpress.com

[25] [Planificacion_de_la_búsqueda_de_personas_desaparecidas.pdf](#)

Con esta información, más una cartografía del lugar a mano, se establecerá una estrategia de búsqueda. Existen varios métodos de búsqueda como ser: por espiral (cuando no hay certeza del rumbo hacia donde se dirigió la o las personas), por rumbo invertido (cuando se tiene certeza del rumbo inicial y el punto de partida y llegada), abanico (cuando la superficie de búsqueda es relativamente reducida y se cuenta con poca gente de búsqueda), rastrillaje (cuando el equipo de rescate está conformado por mucha gente con conocimientos).

Búsqueda por espiral

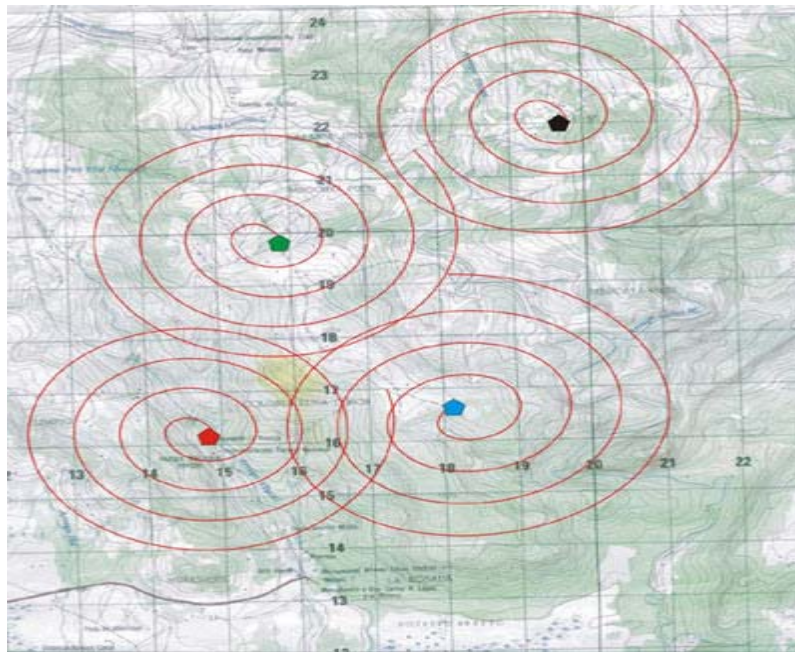


Figura 1.20 Búsqueda por espiral

En el gráfico se puede observar una serie de espirales, esto se da a llamar búsqueda compacta ya que por la superposición de los espirales nos permiten una cobertura de toda la superficie, obsérvese que el punto de partida de cada espiral es un polígono ubicado en un punto de fácil acceso para poder brindar logística al grupo de rescate, el punto de cierre del espiral también se da en otro punto de fácil acceso, generalmente este es el punto de extracción, de ubicarse a los extraviados se tomara la cartografía y se marcará un rumbo de extracción directo al punto de partida, al punto de cierre o a un punto pre fijada con antelación la elección se realizara por la distancia a la que se halla el grupo en el momento de contacto con los extraviados y estos tres puntos de referencia.

Búsqueda por Rumbo Invertido

Esta técnica nos permite desandar el camino tomado por los excursionistas si tenemos la información certera respecto a:

1. 1. Punto de partida
1. 2. Lugar propuesto de llegada
1. 3. Tiempo que se calculó para la llegada
1. 4. Conocimiento del grupo extraviado (supervivencia, orientación en monte, si son novatos en este tipo de prácticas o ya han tenido experiencias anteriores), nuestro rumbo se debe de planificar no en línea recta sino que un constante zigzag a traves del rumbo estimado de los excursionistas.



Figura 1.21 Búsqueda por rumbo invertido

En nuestra cartografía marcaremos puntos que consideramos críticos ya sea por la característica de la vegetación, por una alta densidad de serpientes venenosas, por accidentes geográficos abruptos, aguas rápidas etc. Se ha de controlar todos aquellos lugares que se encuentran cercanos al rumbo pre fijado con características especiales de riesgo.

Se debe tener en cuenta antes de comenzar la búsqueda cual será el punto de extracción de más accesibilidad a el equipo de logística y o equipo médico una vez determinado ese punto se denominará punto de extracción.

Búsqueda por Abanico

Para poder desarrollar esta técnica de rescate se debe de contar con un numeroso grupo de trabajo, cada persona debe contar con equipo de comunicaciones, cartografía y silbato, el grupo de trabajo partirá del mismo punto que los excursionistas y se asignará un punto de cierre en distancia generalmente se determina un accidente geográfico como ser ríos, cañones, macizos, etc, o también se puede tomar como referencia un camino.

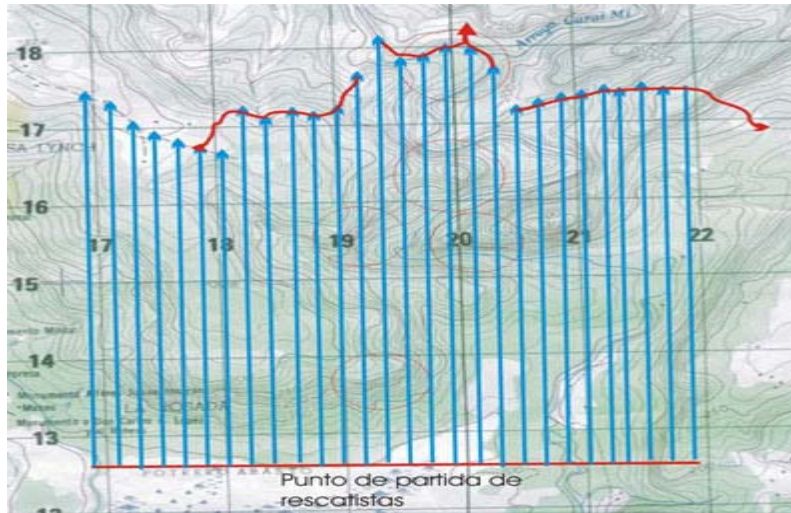


Figura 1.22 Búsqueda por abanico

Para poder desarrollar esta técnica de rescate se debe de contar con un numeroso grupo de trabajo, cada persona debe contar con equipo de comunicaciones, cartografía y silbato, el grupo de trabajo partirá del mismo punto que los excursionistas y se asignará un punto de cierre en distancia generalmente se determina un accidente geográfico como ser ríos, cañones, macizos, etc, o también se puede tomar como referencia un camino.

Búsqueda por Rastrillaje

Para realizar búsquedas por rastrillaje se debe de contar con un numeroso grupo de trabajo, cada persona debe contar con equipo de comunicaciones, cartografía y silbato, el grupo de trabajo partirá de una línea imaginaria que nos de la suficiente cobertura de terreno de la zona en la que se considera que se hallan los excursionistas se asignará un punto de cierre en distancia generalmente se determina un accidente geográfico como ser ríos, cañones, macizos, o también se puede tomar como referencia un camino siendo esta opción ideal para poder brindar asistencia rápida a los excursionistas como al equipo de búsqueda y rescate.

Generalmente con esta técnica cada célula del equipo se encuentra a poca distancia unos de otros, por ejemplo si se cuenta con 15 personas y se las ubica

cada 500 m. Así podremos cubrir con cada una, una superficie de 7.5 km, siempre debemos mantener reglas de trabajo entre el equipo:

1. 1. Deberán ponerse en contacto entre sí con una frecuencia de tiempo predeterminada reportando novedades

1. 2. Cada grupo estará pendiente de su compañero de derecha e izquierda

1. 3. La utilización del silbato será en casos muy definidos porque si no se generará confusión en la totalidad del grupo

1. 4. Una vez localizados los excursionistas, se convocará a los grupos más cercanos para organizar la extracción hacia un punto determinado, donde confluirán todos los grupos de trabajo.

Para poder realizar rescates en zonas montañosas debemos tener en cuenta que se necesita un gran dominio de las técnicas de escalada, rappel, transporte de heridos, lo común que cuando la gente que no está entrenada se lanza a realizar un rescate en zona muy difíciles terminen tanto o más accidentados que aquellos a quien desearon rescatar

CAPITULO 2

ANÁLISIS Y SELECCIÓN DE ELEMENTOS

En este capítulo se presenta la arquitectura de cuadricóptero, para ello se ha dividido todo el sistema en varias partes. Comenzaremos por una breve introducción a la dinámica del cuadricóptero para dar un enfoque práctico del funcionamiento de vuelo de dicho sistema. El segundo bloque se compone de las partes mecánicas, desde los motores hasta los sensores pasando por el sistema de procesado de información como podemos (figura 2.1).

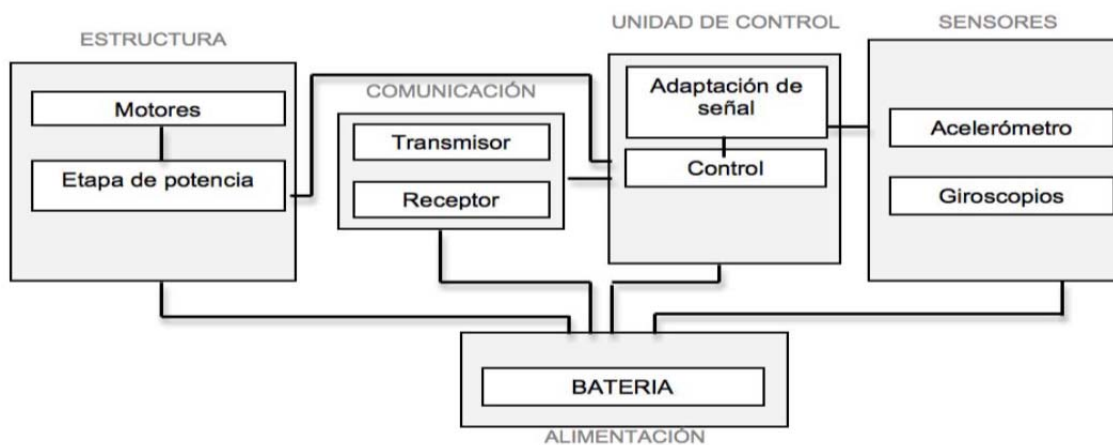


Figura 2.1 Arquitectura genérica del cuadricóptero

2.1 GENERALIDADES DE LOS PRINCIPALES DISPOSITIVOS ELECTRÓNICOS A EMPLEARSE

2.1.1 DINÁMICA

La dinámica de un sistema para un cuadricóptero generar cambios en la potencia entregada a cada motor.

Para el siguiente análisis se asume que la plataforma está volando estable con una potencia de motores PWM (en los cuatro motores). Los cuadricópteros como todo vehículo volador tiene 3 grados de libertad angulares, estos son roll, pitch y **yaw** ^[26].

[26] <http://electronica36.blogspot.mx>

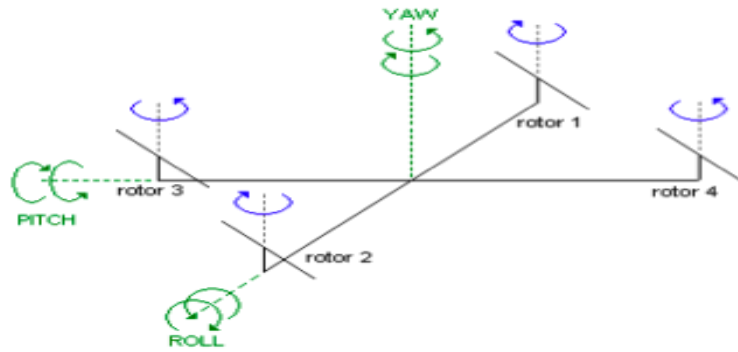


Figura 2.2 Grados de libertad de un cuadricóptero

2.1.1.1 MOVIMIENTO DE GUIÑADA (YAW)

Se refiere al movimiento cuando el vehículo gira sobre su eje vertical. El cuadricóptero logra este movimiento al aumentar por igual la potencia de giro de los rotores 1 y 3 y disminuir en igual magnitud los motores 2 y 4. Al disminuir esta potencia aumenta el par del motor creando un giro contrario a las hélices que están rotando con mayor potencia.

Las potencias de giro deben variar equilibradas para mantener el vehículo estacionario en el aire, es decir, la fuerza total constante.

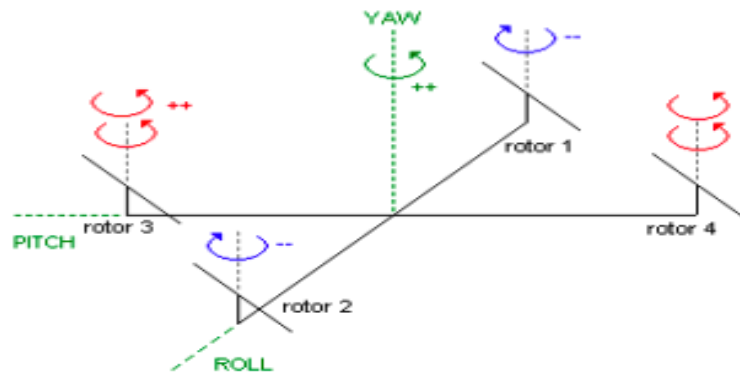


Figura 2.3 Movimiento de yaw

2.1.1.2 MOVIMIENTO DE INCLINACIÓN (PITCH)

Es el movimiento que permite el desplazamiento hacia adelante y atrás. El vehículo mantiene la potencia en el rotor 1 que es opuesto al sentido deseado, reduce al mínimo la del rotor 3 y deja los otros dos a potencia media, así la sustentación del rotor 1 hace que el vehículo se incline a favor del sentido deseado y se desplace.

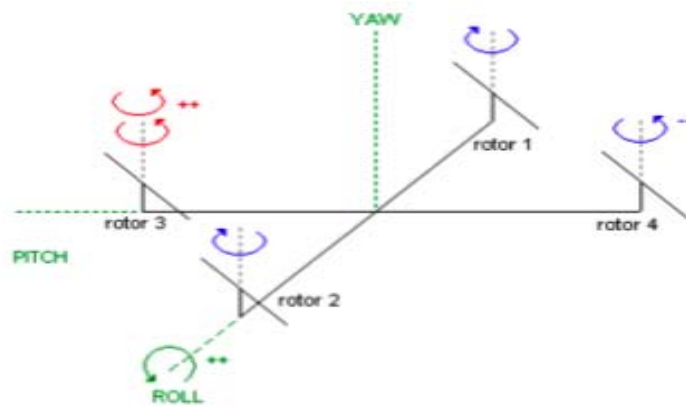


Figura 2.4 Movimiento de pitch

2.1.1.3 MOVIMIENTO DE BAMBOLEO (ROLL)

Permite realizar los movimientos a la izquierda o derecha. Usa el mismo principio que el de inclinación, pero lateralmente. La combinación de los tres movimientos mencionados son los que hacen maniobrar al cuadricóptero libremente.

Los movimientos de roll y pitch son giros en torno a los ejes horizontales del cuadricóptero. Una inclinación en cualquiera de estos ejes produce un movimiento lineal en el plano horizontal cuya velocidad depende del ángulo (esto se denomina ángulo de ataque) y la dirección depende de la orientación del cuadricóptero.

Yaw es un giro respecto a la vertical y no produce movimiento lineal ^[26].

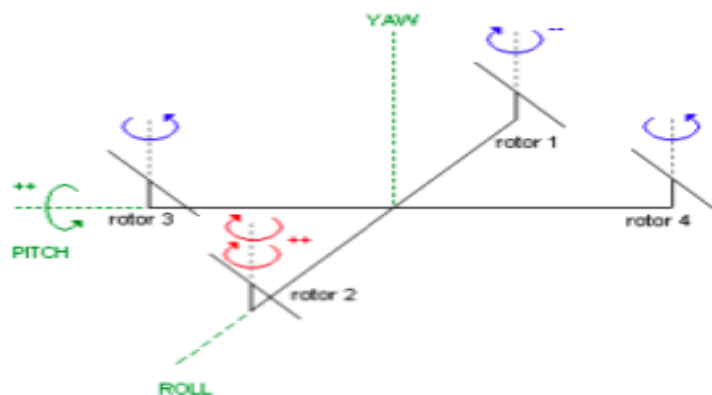


Figura 2.5 Movimiento de roll

[26] <http://electronica36.blogspot.mx>

2.1.1.4 MOVIMIENTO VERTICAL

Se puede hacer ascender, descender o mantener en vuelo estacionario al cuadricóptero, haciendo que la fuerza de sustentación generada por los 4 pares motores-hélices sea mayor que la fuerza peso generada por la atracción gravitatoria.

Este movimiento se logra al variar la potencia de los cuatro motores en igual medida para no modificar los demás grados de libertad. De esta forma la plataforma puede ascender o descender ^[26].

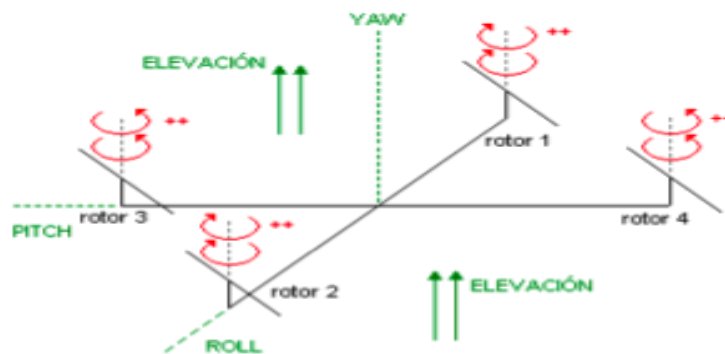


Figura 2.6 Movimiento vertical

Lo dicho anteriormente es para el caso ideal, donde los cuatro motores tienen exactamente las mismas constantes y giran a las mismas velocidades angulares. En el caso real, esto no es posible, porque siempre existen ciertas variaciones en las fuerzas generadas por los motores, ya sea porque las constantes de los motores son diferentes o las velocidades angulares giran a distintas revoluciones.

El diseño de un controlador para el cuadricóptero no es un problema trivial debido a varias razones como son el acoplamiento entre las fuerzas y momentos generados por los motores y hélices, el rozamiento con el aire, la fuerza de empuje del viento, etc.

[26] <http://electronica36.blogspot.mx>

2.2.1 MICROCONTROLADORES [27]

Un microcontrolador es un circuito integrado o chip que incluye en su interior las tres unidades funcionales de una computadora: CPU, Memoria y Unidades de E/S, es decir, se trata de una computadora completa en un solo circuito integrado programable y se destina a gobernar una sola tarea con el programa que reside en su memoria. Sus líneas de entrada/salida soportan el conexionado de los sensores y actuadores del dispositivo a controlar.

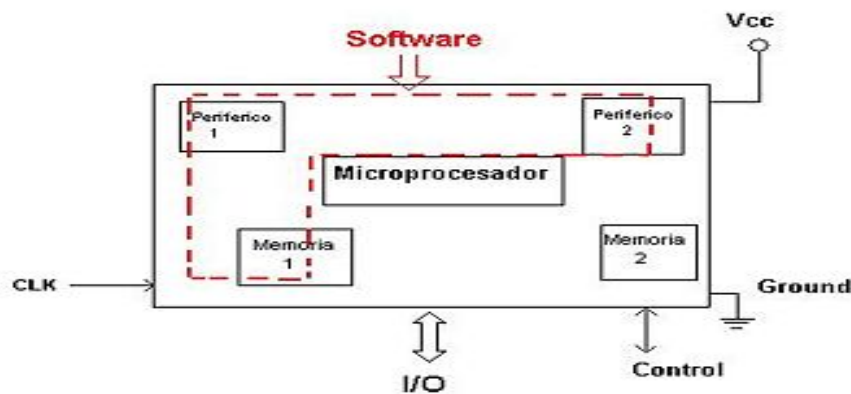


Figura 2.7 Esquema de un microcontrolador

No debemos de confundir los microcontroladores con los microprocesadores, ya que estos últimos sólo es un unidad central de procesos, que de manera externa, hay que conectarle la memoria, puertos y unidades de almacenamiento, tiene la ventaja de ser más escalable que un microcontrolador, lo que nos da la capacidad de poder agregarle más componentes según sea nuestra necesidad.

Para el control de un sistema de cuadricóptero es suficiente utilizar un microcontrolador, la utilización de los conversores A/D, los contadores y la pequeña memoria (RAM y ROM) que incorporan hacen que estos dispositivos sean los más utilizados en este ámbito. En el diseño del cuadricóptero se ha utilizado un microcontrolador del fabricante ATMEL utilizando los conversores A/D de 10 bits y las 6 salidas PWM.

[27] Control de un Quadrotor mediante la plataforma Arduino. Tesis de Ingeniería de Telecomunicaciones, especialidad Sistemas de Telecomunicaciones.

2.2.1.1 CARACTERÍSTICAS DE MICROCONTROLADOR ATMEGA 328

El ATmega328 es un solo chip de micro-controlador creado por Atmel y pertenece a la megaAVR serie.

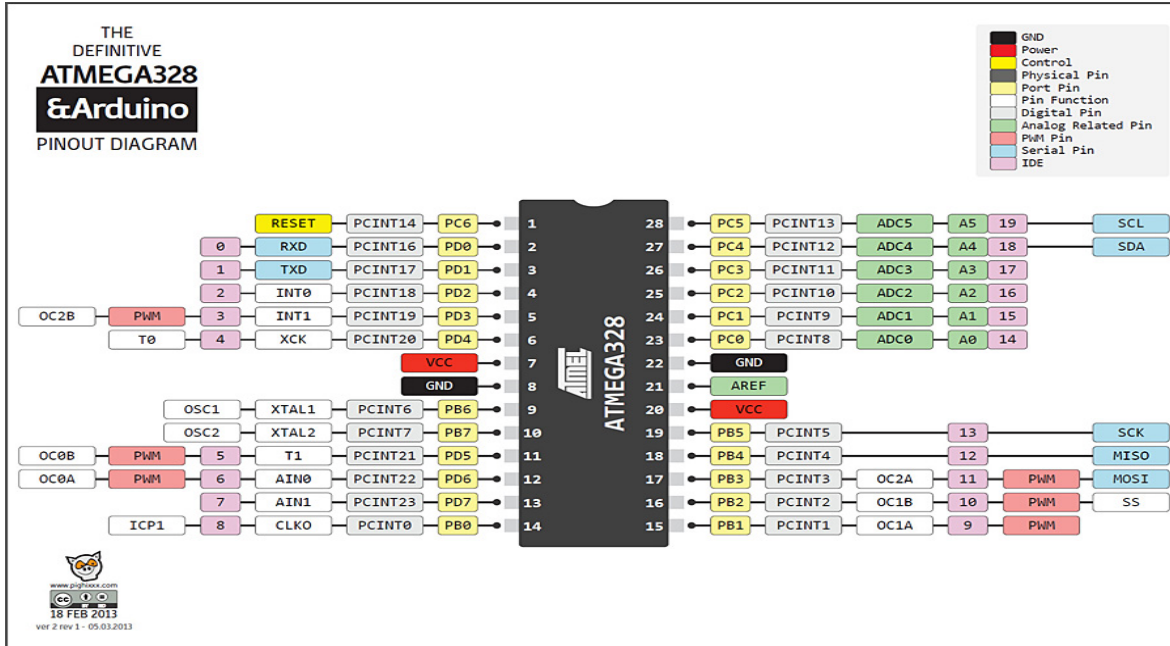


Figura 2.8 Esquema de ATMEGA 328

Especificaciones

El Atmel 8 bits AVR RISC -basado microcontrolador combina 32 KB **ISP** destello de memoria con capacidades de lectura y escritura, mientras, 1 KB **EEPROM**, 2 KB de **SRAM**, 23 de propósito general las líneas de E / S, 32 propósito de trabajo generales registros , tres temporizador flexible / contadores con comparan modos, internos y externos interrupciones , serial programable **USART**, una interfaz en serie de 2 hilos orientado a byte, **SPI** puerto serie, 6 canales de 10 bits convertidor A / D (8-canales en **TQFP** y **QFN** / **FML** paquetes) , programable temporizador de vigilancia interna con oscilador , y cinco de software modos de ahorro de energía seleccionable. El dispositivo funciona entre 1.8 hasta 5.5 voltios. El dispositivo logra rendimientos se acercan a 1 **MIPS** por MHz^[28].

[28] <http://en.wikipedia.org/wiki/ATmega328>

PARÁMETROS	VALOR
Flash	32 Kbytes
RAM	2 Kbytes
Número de pines	28
Max. Frecuencia de funcionamiento	20 MHz
UPC	8-bit AVR
# De Canales Touch	16
Hardware QTouch Adquisición	No
Max pines I / O	26
Ext Interrupciones	24
Interfaz USB	No
Speed USB	No

Tabla 2.1 Características ATMEGA 328^[28]

Hoy en día el ATmega328 se utiliza comúnmente en muchos proyectos y sistemas autónomos donde se necesita una, de bajo costo microcontrolador simple, de baja potencia. Tal vez la más común la aplicación de este chip es en la popular Arduino plataforma de desarrollo, a saber, los Arduino Uno y Arduino Nano modelos.

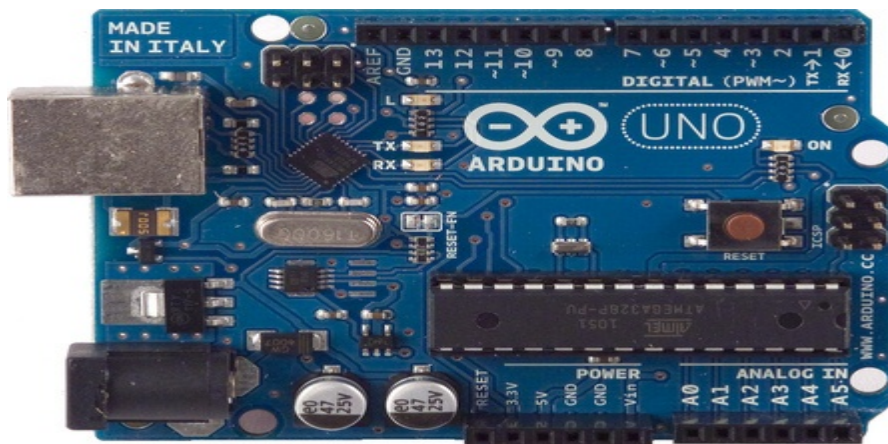


Figura 2.9 Placa de Arduino Uno con microcontrolador ATMEGA 328

2.2.1.2 MÓDULO BLUETOOTH HC-05

Los módulos de bluetooth HC-05 son módulos muy populares para aplicaciones con microcontroladores, PIC y Arduino. Se trata de dispositivos relativamente económicos y que habitualmente se venden en un formato que permite insertarlos en un protoboard y cablearlo directamente a cualquier microcontrolador, incluso sin realizar soldaduras^[29].

[29] <http://www.geekfactory.mx/radio/bluetooth-hc-05-y-hc-06-tutorial-de-configuracion/>

El módulo de bluetooth HC-05 es el que ofrece una mejor relación de precio y características, ya que es un módulo Maestro-Eslavo, quiere decir que además de recibir conexiones desde una PC o tablet, también es capaz de generar conexiones hacia otros dispositivos bluetooth. Esto nos permite por ejemplo, conectar dos módulos de bluetooth y formar una conexión punto a punto para transmitir datos entre dos microcontroladores o dispositivos.

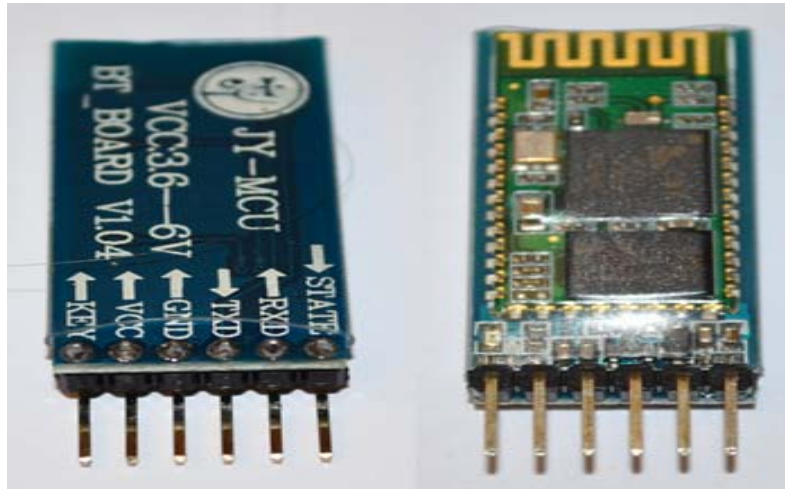


Figura 2.10 Bluetooth HC-05

2.2 PARTES CONSTITUTIVAS DEL CUADRICÓPTERO

2.2.1 MOTORES

La elección de los motores es una decisión importante. Existen dos tipos. Los motores con escobillas (brushed motors o DC motors) y los motores sin escobillas llamados, “brushless motors” o “brushless DC motors”.

Motores brushless ^[30]

Usualmente los llamados motores sin escobillas “brushless motors” son utilizados en vehículos aéreos no tripulados, ya que estos presentan muchas ventajas con respecto a los motores eléctricos convencionales (con escobillas), como:

- Son más baratos de fabricar.
- Pesan menos.
- Requieren menos mantenimiento.
- Son más eficientes.
- Generan menor interferencia electromagnética.

[30] <http://www.aeromodelismofacil.com/Motores%20brushless.htm>

Las desventajas;

- Mayor coste de construcción
- El control es caro y complejo
- Siempre hace falta un control electrónico para que funcione, que a veces duplica el costo.



Figura 2.11 Motor Brushless

Dentro de la familia de los brushless, existen dos tipos diferentes. De tipo *Inrunner* y *Outrunner*.

En los motores outrunner, los imanes están situados en el exterior de la estructura. Por tanto, se puede ver como la parte exterior del motor, gira. Sucede lo contrario en los motores inrunner. Los imanes están en el interior de la estructura, y por tanto se ve que lo único que gira es el eje.

Los motores outrunner giran mucho más despacio y el par es mucho mayor. La mayor ventaja es el hecho de que no es necesaria una caja de cambios, lo que los hace más silenciosos. Son ligeramente menos eficientes que los inrunner, pero es tan pequeña la diferencia que no debiera de ser un factor determinante a la hora de hacer la elección.

Los motores inrunner por su parte, son más eficientes cuanto más rápido gira el motor y en general son más eficientes que los outrunner. Necesitan de un elemento adicional entre el motor y la hélice. La parte negativa de los motores inrunner es que estas partes adicionales pueden y suelen dar problemas.

Motores DC

Los motores de corriente continua son muy utilizados en mini robots. Podemos encontrar motores de este tipo en prácticamente todos los coches teledirigidos y/o coches de juguete, los ventiladores de mano y accionamientos varios.

El rotor es el dispositivo que gira en el centro del motor DC y está compuesto de arrollados de cable conductores de corriente. Esta corriente es suministrada al rotor por medio de las "escobillas" generalmente fabricadas de carbón. La fuerza con la que el motor gira es proporcional a la corriente que hay por los conductores. A mayor tensión, mayor corriente y mayor par motor ^[31].



Figura 2.12 Motor DC

2.2.2 ETAPA POTENCIA / DRIVER

Esta etapa del sistema es muy importante, ya que va a ser la etapa que nos proporcione el control de los motores, es decir, nos va a "traducir" la información de nuestro centro de control / emisora, y va a proporcionar la señal adecuada para el control de los motores. Un aspecto muy importante a tener en cuenta en esta etapa es la corriente que vamos a tener circulando por nuestro sistema.

Según el motor que se utilice deberemos de elegir un diseño u otro. Podemos encontrar bastantes esquemas de control, pero todos se basan en la misma idea pero ofreciendo características diferentes como por ejemplo la potencia máxima que pueden soportar.

Explicaremos la etapa de potencia para motores DC, ya que es la que se ha utilizado en el diseño del cuadricóptero que veremos en el siguiente capítulo.

Podemos encontrar circuitos integrados que nos proporcionen todo lo necesario o podemos optar por circuitos con componentes discretos.

Con los motores brushless existen controladores que proporcionan todo lo necesario para utilizar estos motores.

Para conseguir variaciones de velocidad, ya sea desde nuestra emisora radiocontrol o desde una unidad de control se utiliza habitualmente una modulación por anchura de pulso (PWM).

Dicha señal la podemos generar desde un circuito integrado o lo más usual desde un microcontrolador.

[31] <http://www.mabuchi-motor.co.jp/>



Figura 2.13 Señal PWM

La señal PWM se utiliza como técnica para controlar circuitos analógicos. El periodo y la frecuencia del tren de pulsos pueden determinar la potencia entregada a dicho circuito. Si, por ejemplo, tenemos un voltaje de 9 V y lo modulamos con un ciclo de trabajo (**duty cycle**) del 10%, obtenemos 0.9 V de media. Para entenderlo mejor, realmente estamos alimentando el motor y apagándolo en períodos de tiempo del orden de milisegundos. Gracias a la inercia del propio motor estos cambios provocan que se mueva a una velocidad constante. Así pues, cuanto mayor sea el ancho del pulso, mayor velocidad obtendremos.

$$\text{DutyCycle} = \frac{\text{Ancho Pulso}}{\text{Periodo}} \quad (\text{Formula 2.1})$$

2.2.3 ALIMENTACIÓN

Las baterías de Li-po, que son de Polímero de Litio, estas baterías tienen la capacidad de ofrecer grandes corrientes, las puedes encontrar en 1s(3,7vdc), 2s(7,4vdc), 3s(11.1vdc), 4s(14.8vdc), 5s(18.5vdc), 6s(22.2vdc), estas baterías son livianas y tienen gran desempeño, otro factor a considerar es la tasa de descarga que es la "c", en estas baterías puedes encontrar un valor seguido de una "c" eso quiere decir el factor de multiplicación de corriente que le podemos exigir aunque los valores que vienen marcados son valores picos, nunca es bueno exigir al máximo ya que requieren cuidados especiales, así como carga especial, balanceo de celdas, etc ^[32].

Los parámetros a considerar de una batería son los siguientes:

- La tensión de salida, medida en voltios.
- La capacidad eléctrica, se mide por referencia a los tiempos de carga o de descarga en amperios hora (Ah) En las baterías se utiliza un múltiplo, el miliamperio hora (mAh). Este parámetro es muy importante y debemos fijarnos en él cuando compremos la batería pues cuanto mayor sea, más tiempo tardará en descargarse por el uso. El precio está en relación directa con este parámetro.

[32] <http://circuitronico.blogspot.mx/2012/03/baterias-li-po-y-esc.html>

A continuación mostramos los diferentes tipos de baterías que se utilizan en los equipos autónomos o de radio control.

Ni-Cd (baterías de níquel-cadmio) ^[33]

Durante mucho tiempo las baterías de níquel-cadmio fueron la única opción para propulsar a los vehículos hasta que en 1990 aparecieron las baterías de níquel-metal-hidruro y las de ion-litio. El voltaje de una célula de Ni-Cd es de 1.2 V. Poseen efecto memoria. El efecto memoria se produce cuando cargamos las baterías sin haberlas previamente descargado por completo, se crean unos cristales en el interior de dichas baterías que hace que no se carguen en toda su capacidad. Para prevenir el efecto memoria basta con realizar de vez en cuando una descarga-carga completa. Otro inconveniente es que el cadmio es muy tóxico.

Ni-MH (baterías de níquel-metal-hidruro) ^[33]

En estas baterías, debido a la alta toxicidad del cadmio, éste se sustituye por hidruros metálicos. Tienen mayor densidad de carga (capacidad), no contienen Cd (menos contaminantes), no poseen efectos de pérdida de capacidad por mal uso (efecto memoria). Soportan un menor número de cargas durante su vida útil que las de Ni-Cd. Tienen una resistencia interna superior lo que las limita para alimentar motores de alta potencia. Aceptan cargas rápidas.

Li-ion (baterías de iones de litio) ^[33]

La capacidad de una batería de Ion-Litio es aproximadamente el doble de la capacidad de una batería de Níquel-Cadmio. El litio es el metal más ligero que existe por lo que a igualdad de capacidad las baterías resultan mucho más ligeras. El voltaje de una célula de Ion-Litio es de 3.7 V. No tienen mantenimiento, no poseen efecto memoria. Tienen como desventaja que requieren de un circuito de control que se emplean para limitar el voltaje máximo de cada célula de la batería.

Li-Po (baterías de polímero de litio) ^[33]

Son una variación de las Baterías Litio-Ion (Li-ion). Tienen una densidad de energía entre 5 y 12 veces las de Ni-Cd o las de Ni-MH. Son baterías mucho más ligeras y que pueden adoptar cualquier forma. No poseen efecto memoria. El voltaje de cada elemento es de 3.7 V. La desventaja es que necesitan una carga mucho más lenta que las de Ni-MH. Este tipo de baterías son ideales para alimentar motores muy potentes.

[33] <http://rc.lapipadelindio.com/general/baterias-automodelismo-radiocontrol>



Figura 2.14 Batería LiPo

Se debe cuidar de que el consumo máximo del motor sea menor que la descarga máxima de la batería para evitar que la vida de ésta se acorte demasiado. En la tabla 2.2 podemos observar una comparativa de los 4 modelos estudiados.

Tabla 2.2 Comparativa baterías ^[34]

	Voltaje por célula	Efecto memoria	Resistencia interna	Cargas vida útil	Velocidad de carga
Ni-Cd	1.25	SI	*	**	Lenta
Ni-MH	1.25	NO	**	*	Rápida
Li-ion	3.7	NO	***	***	Rápida
Li-Po	3.3	NO	****	****	Lenta

2.2.4 CÁMARA

La principal característica del cuadricóptero, y lo que le hace un producto realmente innovador, es su **cámara frontal**, que nos permite ver desde nuestro entorno en todo momento lo que "ve" el cuadricóptero.

Con la cámara graba y percibe imágenes las cuales nos demuestran una panorámica muy amplia de lo que el usuario quiere observar.

[34] <http://flightpowerbatteries.com/>

2.2.5 HÉLICES ^[35]

Para el funcionamiento del cuadricóptero se necesitan que 2 hélices giren en un sentido y las otras 2 en otro sentido, esto hace que si la sumatoria de las 2 velocidades de las que giran en un sentido es igual a la sumatoria de las que lo hacen en sentido contrario, el torque resultante se cancela.

Las hélices más comunes son las que se usan en el morro del avión, están son conocidas como hélices normales. Hay otro tipo de hélices llamadas empujadoras o pusher en ingles que son las que se utilizan en aeronaves que tienen generadores de empuje en la cola. Una posible solución a este problema será girar 180° la hélice sobre su eje y colocarla en el motor. Pero esta solución tiene corto alcance debido a que el empuje generado por la misma hélice cuando gira en su sentido normal es distinto al generado cuando gira en el otro sentido.

Las hélices empujadoras están diseñadas para que el empuje generado por su rotación sea más efectivo cuando giran al revés que una hélice normal.

Las hélices en general tienen 2 parámetros que las caracterizan: el paso y el diámetro.

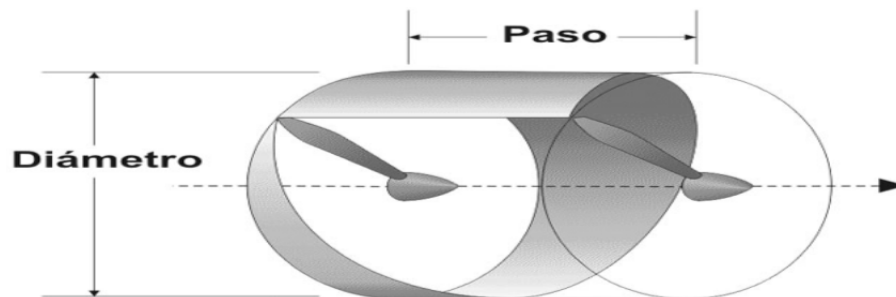


Figura 2.15 Paso y Diámetro de una hélice

El paso se calcula como la distancia que recorrerá la hélice en una vuelta si esta se moviese en un medio sólido (semejante a un tornillo) y el diámetro es el del círculo que generan cuando estas rotan. Este diámetro me indica el área del aire que entra en juego en el cálculo del empuje.

Las hélices definen el consumo de corriente de los motores, así como el empuje que se puede lograr.

[35] <http://electronica36.blogspot.mx/2012/05/informe-completo-sobre-como-hacer-y.html>



Figura 2.16 Hélice del tipo normal



Figura 2.17 Hélice del tipo pusher

2.3 UNIDAD DE CONTROL

La unidad de control es un elemento de los sistemas de cuadricóptero bastante peculiar. Como todo sistema aéreo debe de disponer de unos sensores para medir parámetros como la aceleración de los ejes o la velocidad angular, esto ayuda al sistema a proporcionar la estabilización del mismo. La función más compleja de la unidad de control es la estabilización del sistema, aunque debe de realizar otras funciones como el almacenamiento de datos entre otras.

Las funciones principales de una unidad de control son las siguientes:

- Estabilizar el sistema mediante los sensores.
- Proporcionar comunicación estación base.
- Interpretar comandos de control enviados desde la estación base.
- Guardar / enviar datos recogidos de los sensores.

2.3.1 SENSORES ^[27]

Una aeronave capaz de realizar vuelos autónomos con una mínima intervención del piloto necesita incluir algunos sensores que le ayuden a mantener un vuelo estable y seguro.

En un sistema del cuadricóptero podemos encontrar un gran número de sensores enfocados a la estabilización del sistema y a la navegación.

Algunos de los sensores más utilizados en estos sistemas son los acelerómetros y giroscopios, aunque en algunos sistemas más avanzados, también podemos encontrar sensores de presión. Por otra parte, tenemos los sensores o equipos utilizados para la navegación, que van desde sensores de proximidad hasta receptores GPS.

2.3.1.1 GIROSCOPIO ^[27]

Los giroscopios son dispositivos utilizados para medir o mantener una orientación específica. Antiguamente solo se contaba con los giroscopios mecánicos, que consistían en discos giratorios suspendidos sobre un eje que podía rotar en varias direcciones. La dinámica consistía en que el disco rotatorio permanecía en la misma posición indiferentemente de cómo rotara la plataforma sobre la cual estuviera montado.

Actualmente se cuenta con otro tipo de giroscopios, los cuales son basados en sistemas micro-electromecánicos, y han aumentado su popularidad en los dispositivos electrónicos debido a su reducido tamaño, bajo precio y buena precisión. Una de las mayores aplicaciones de los giroscopios es para el desarrollo de unidades de medición inercial (IMU) de vehículos o dispositivos que requieran de este tipo de mediciones, como es el caso de los vehículos aéreos no tripulados (UAV) para sus sistemas de estabilización.



Figura 2.18 Giroscopio

[27] Control de un Quadrotor mediante la plataforma Arduino. Tesis de Ingeniería de Telecomunicaciones, especialidad Sistemas de Telecomunicaciones.

Su detección se basa en que las piezas cerámicas en vibración están sujetas a una distorsión que se produce por el efecto **Coriolis** (cambios de velocidad angular). Como resultado de esta deformación el giroscopio genera un voltaje de salida que es proporcional a esta velocidad angular de rotación.

2.3.1.2 ACELERÓMETRO ^[36]

El acelerómetro es un instrumento que mide aceleraciones. Lo que nos interesa es la aceleración de coordenadas (cambio de la velocidad del dispositivo en el espacio) pero esto presenta varios problemas.

Para entender un acelerómetro, imaginemos una pelota de tenis “encerrada” en una especie de dado gigante. Este dado será nuestro acelerómetro, donde las paredes del mismo serán capaces de medir la fuerza que aplicamos sobre ellas. Si movemos el dado hacia la izquierda, la pelota chocará contra la pared izquierda. La fuerza que mediremos en esta pared será la manera de medir la aceleración. Si movemos el dado en diagonal, la pelota hará fuerza en dos paredes en vez de una, pero siguiendo la misma idea.

Pero qué pasa si dejamos quieto nuestro dado en el suelo? Debido a la gravedad, la pared inferior medirá una fuerza, ¡y sin embargo no hay cambio de velocidad en el dado (acelerómetro)!. Otro ejemplo claro es que un dado en caída gravitacional libre hacia el centro de la Tierra medirá un valor de cero, ya que, a pesar de que su velocidad es cada vez mayor, está en un marco de referencia en el que no tiene peso. El acelerómetro por tanto, mide todas las aceleraciones excepto las causadas por la gravedad.

2.3.2 ESC: Electronic Speed Controler ^[37]

Para controlar los motores Brushless utilizamos los variadores ESC (Electronic Speed Controler), que están diseñados para dirigir a motores sin escobillas brushless. Estos pequeños controladores varían la corriente continua procedente de las baterías LiPo a corriente alterna de diferentes frecuencias, controlando así la velocidad de salida del motor. El principal parámetro para dimensionarlos es el amperaje máximo admisible, que debe ser superior a la máxima corriente que van a consumir los motores en continua.

Normalmente, poseen, internamente, un microcontrolador programable para configurar algunos parámetros. Esto depende de cada modelo.

[36] <http://www.instructables.com/id/Accelerometer-Gyro-Tutorial/>

[37] http://diebotreise.blogspot.mx/2012/11/esc-electronic-speed-controler.html?utm_source=BP_recent

Los parámetros variables son:

- Activar el freno (yes/no)
- Tipo de batería (LiPo/NiMh)
- Tipo de Corte ante batería baja (soft-cut/ off-cut)
- Corte por voltaje (Low/Middle/High)
- Modo de arranque (Normal/soft/very soft)
- Temporizador (Low/Middle/High)
- Música en función de las células de la batería (D/C/B/A)
- Modo Governor (on/off)

Realmente los parámetros que vienen por defecto son los mejores y no se necesita cambiar nada, excepto si deseas un arranque súper suave, que pueden ser útiles si la batería o el propio ESC no están bien dimensionados para soportar el pico en el arranque.

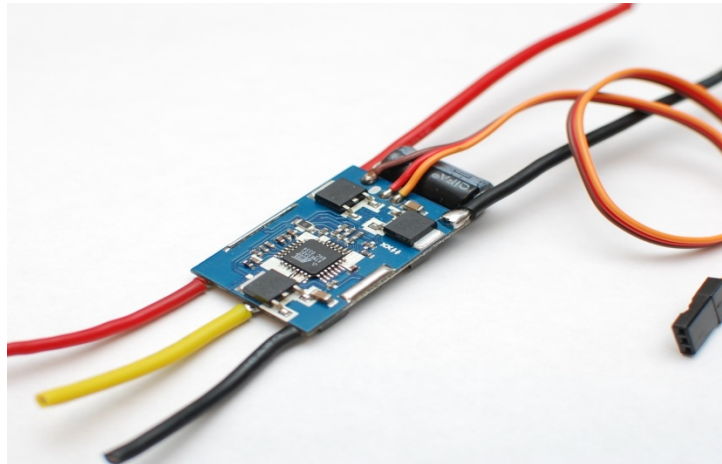


Figura 2.19 ejemplo ESC

CAPITULO 3

DISEÑO Y ENSAMBLAJE DE CUADRICÓPTERO

En este capítulo lo que se trata es el diseño propio de un sistema para cuadricóptero.

Un sistema para cuadricóptero que utiliza la estructura de fibra de vidrio. El sistema de propulsión se consigue a través de cuatro motores dc y está equipado con un acelerómetro de tres ejes y tres giroscopios que ayudan a la estabilización de todo el sistema. El procesado de todos los datos se realiza mediante el microcontrolador ATMEGA 328.

A continuación se va a describir de forma detallada, todas las partes que componen el cuadricóptero y los resultados obtenidos a partir de las pruebas realizadas sobre él. En el diagrama N°1 podemos observar el escenario inicial utilizado en el proyecto.

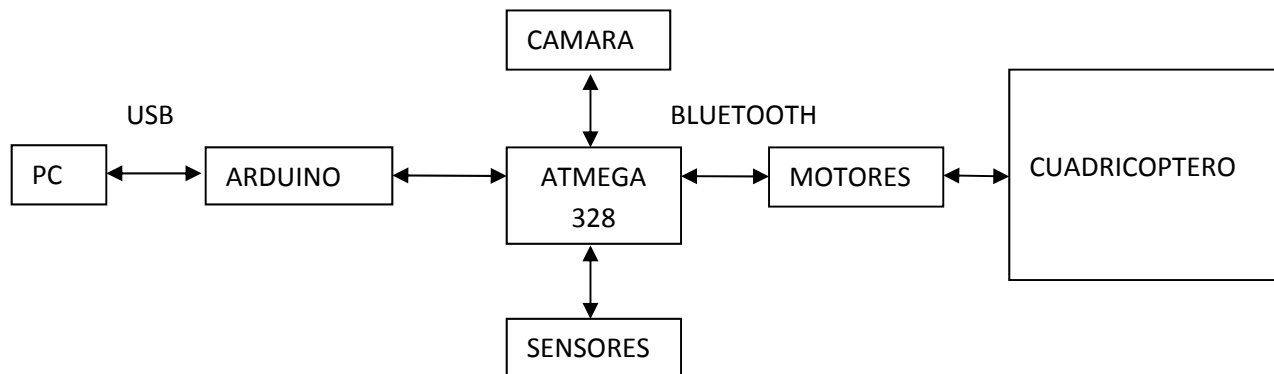


Diagrama N°1 Partes del cuadricóptero

Se comenzó utilizando el control del cuadricóptero mediante PC porque proporciona sencillez a la hora de programar por medio del Arduino Uno el micro ATMEGA 328 y ayuda a realizar las pruebas iniciales de comunicación.

3.1 ESTRUCTURA

La estructura es una de las primeras partes a elegir cuando realizamos el diseño del cuadricóptero. Esta estructura es de fibra de carbono con un peso de 123 gr y con unas dimensiones de 42 cm x 42 cm aproximadamente.

La estructura se compone de dos ejes perpendiculares formando una cruz simétrica. En sus extremos incorpora unos pequeños soportes para los motores.

Toda la electrónica y batería está colocado en el centro de la estructura para ayudar a su ensamblaje.

Un excelente marco fuerte Quadcopter fibra de vidrio ligero, con el tren de aterrizaje y un tablero de distribución de energía de bonificación (APP) para el cableado ordenado y fácil. Esta cuádruple tiene 480mm centros motores por lo que es ideal para los 9 y 10 apoyos pulgadas y también tiene un montaje de la batería ajustable para lograr la perfecta distribución del peso y el marco inferior está lista para tomar toda una serie de montajes y cardanes cámara por lo que es perfecto para FPV y proyectos de filmación.

El S500 se suministra con 2 x 2 x azules y blancos brazos que son súper camellones y tienen una varilla de fibra de carbono a través del centro de decisiones para los brazos más fuertes que hemos visto hasta la fecha en este estilo de marco. Los brazos azules y blancos hacen orientación mucho más fácil y no hay necesidad de diferentes accesorios de colores.

El tren de aterrizaje incluido tiene altura por lo que es adecuado para cámaras tipo GoPro o similar. Otra característica de la S500 es que cuando se montan los brazos tienen un ligero barrido de arriba, esto da el Quad un efecto diedro que ayuda a que sea muy estable.

Montaje de este marco es súper simple con pre-roscado lo que se requieren mangas para todos los pernos del marco no hay tuercas de seguridad. Utiliza uno pernos de tamaño para toda la construcción que significa que sólo quiere un hexágono tamaño para la asamblea. Otra característica es que las lengüetas de fijación, para los brazos que le permite alcanzar la perfecta alineación en la construcción. Los soportes del motor están listos para tomar una selección de motores con montaje central del taladro de 16, 19 y 25 mm, una vez más haciendo de este un marco muy universal.

Características:

- Construido a partir de fibra de vidrio de calidad y nylon poliamida
- Manguitos de latón roscadas con anterioridad para todos los pernos del marco
- Localización de las fichas de los brazos
- Brazos de colores para la orientación para ayudar a mantener volando en la dirección correcta
- AP para una fácil y ordenada cableado
- Fácil montaje

Especificaciones:

Centros de motor: **480 mm**

Altura: **170 mm**

Peso: **425g (marco solamente)**

Agujeros de montaje del motor de pernos: **16 ~ 19 ~ 25 mm**

Motores 800 ~ 1000KV

Controladores de velocidad 30A

LiPoly batería 1 x 2200mah 3S = 11.1V



Figura 3.1 esqueleto cuadricóptero

3.2 MOTORES

Los motores escogidos son motores de la casa Multi Star. Motor con las siguientes características:

- KV (RPM / V): **935**
- Células Lipo: **2-4s**
- Max Oleada Watts (W): **200**
- Max.current (10s): **15 amperios**
- Sin corriente de carga: **0,4 amperios**
- Interiores Resistencia: **0.180 ohmios**
- Número de polos: **14**
- Dimensiones (Dia.xL): **27,9 x 25,7 mm**

- (sin contar el eje de la hélice) del eje del motor: **3 mm**
- (no el eje de la hélice) eje de la hélice: **ranurada (DJI) estilo redondo 8mm (7mm piso en piso) renunció a 6 mm y una tuerca M6 prop**
- Prop : **10 x 4,7**
- (3 celdas) **8 x 4,5**
- (4cell) Peso: **55g**
- perno separación de agujeros: **16mm * 19mm**

El Turnigy Multistar outrunner están diseñados para estar maximizando el rendimiento multi-rotor. Cuentan con imanes de alta gama 45SH, cojinetes NMB de alta calidad, 14 polacos, montaje de encargo del motor y todos son de precisión equilibrada para un funcionamiento suave, estos motores se han diseñado específicamente para uso multi-rotor.

El 2213 se integra el adaptador de hélice en la caja del motor para problema apego prop libre y centrado. Tanto ranura (estilo DJI) y props estándar pueden ser utilizados, por lo que no es más el daño de su prop de pasar la noche tratando de asegurar sus dos adaptadores pieza prop al eje. El 2213 ha construido en aluminio de montaje para una instalación rápida y fácil en su marco multi-rotor.



Figura 3.2 motor brushless

3.3 ASPAS

Una parte muy importante, son las características de las aspas utilizadas.

Existen muchos diseños de aspas y dimensiones a utilizar. En este caso se utilizan unas aspas de plástico de dimensiones 18 cm de longitud aproximadamente, al estar hechas de dicho material ayudan a alargar la vida útil de ellas.

El movimiento de rotación sobre el mismo se obtiene con cada par de rotores, es decir, se acelera los dos rotores con sentido horario (M3,M4) mientras se desacelera los rotores con sentido anti-horario (M1,M2), y viceversa. Lo que conseguimos con cada par de motores es cancelar el movimiento sobre él, ya que cada par de motores como se ha comentado anteriormente rota en sentido contrario al otro par.



Figura 3.3 aspas

3.4 ESC

Los ESC generalmente aceptan una señal PWM a 50 Hz (ciclos por segundo), cuya amplitud de pulso, varía de 1 ms a 2 ms. Cuando el pulso es de 1 ms a 50 Hz, el ESC responde apagando el motor conectado a su salida. Una amplitud de pulso de 1.5 ms hará que el ciclo de trabajo sea del 50% lo que moverá el motor a media velocidad. Para que el motor trabaje a su máxima velocidad, el pulso tendrá que ser de 2 ms.

Los ESC de 10 amperes son recomendados para cuadricópteros por debajo de 1Kg. En nuestro caso nosotros utilizamos un ESC de 30 amperes por el peso que llevara nuestro drone.

Multistar ESC es un controlador de velocidad de calidad. Diseñado para una instalación simple y fácil configuración estos ESC s han demostrado ofrecer un rendimiento excelente cuando se utiliza en aplicaciones multi-rotor. Si su inician en multi-rotores y necesitan un fácil instalar la solución que se ha probado y demostrado que funciona con el arte del Multi-rotor entonces estos controladores de velocidad se encargarán de ello.

Estos ESC cuentan con una respuesta del acelerador lineal todavía muy suave, sin sacrificar una rápida y nítida, la respuesta a cualquier entrada del acelerador.

Todo Multistar serie ESC vienen con conectores de bala instalados que se adaptan a la ESC 'manejo de la potencia nominal s. Esto permite una comodidad plug and play, simplemente añadiendo el divisor adecuado suministro de la batería (3,4 o 6 ESC).

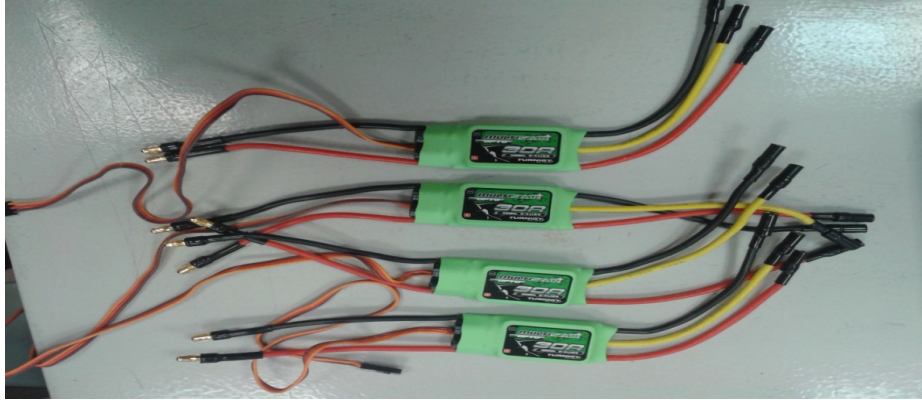


Figura 3.4 ESC para control

3.5 BATERÍA

Como se ha visto en el capítulo 2, una de las baterías más utilizadas en los sistemas aéreos o móviles, son las baterías Li-Po. Estas baterías permiten una mayor densidad de energía, así como una tasa de descarga bastante superior.

Estas baterías tienen un tamaño más reducido respecto a las de otros componentes.

Casi todos los aeromodelistas utilizan baterías Li-po, aunque hay que tener en cuenta que pueden ser peligrosas si se utilizan mal, pudiendo provocar alguna explosión.

En el cuadricóptero se ha utilizado una batería Li-po de 2800 ma.



Figura 3.5 batería Lipo

3.6 UNIDAD DE CONTROL

A causa de las no linealidades de los elementos, ya sean eléctricos o físicos tendremos que realizar una tarea de estabilización. La estabilización se llevará a cabo desde una unidad de control equipada con un microprocesador y diferentes sensores.

3.6.1 MICROCONTROLADOR

Se ha elegido utilizar un atmega 328 para el diseño del quadrotor porque proporciona una facilidad de uso muy grande y la programación que exige es bastante sencilla e intuitiva. Una de las grandes ventajas de la plataforma

A diferencia con otros microcontroladores esta plataforma facilita la configuración del microcontrolador gracias al arduino, el programa sigue la línea de código abierto pero en su versión hardware, esto quiere decir que toda la comunidad tiene acceso a toda la plataforma de manera libre.

El atmega 328 es un chip sencillo y de bajo coste que permite el desarrollo de múltiples diseños. Según ha ido avanzando el tiempo, el procesador se ha ido actualizando hasta llegar al modelo Atmega1280 con mejores prestaciones, como por ejemplo más memoria flash (128 kb) y 16 puertos de entrada y salida.

Al ser hardware libre, tanto su diseño como su distribución es libre, es decir, puede utilizarse libremente para el desarrollo de cualquier tipo de proyecto sin haber adquirido ninguna licencia.

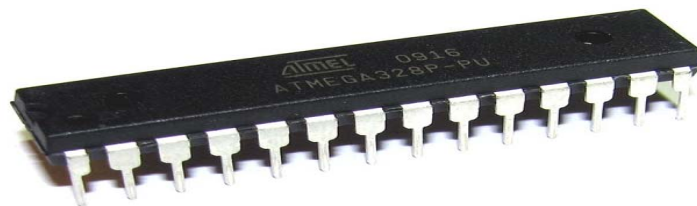


Figura 3.6 ATmega 328

3.6.2 SENSORES

Los sensores utilizados por el sistema cuadricóptero son 3 giroscopios y un acelerómetro de 3 ejes. Estos son los sensores básicos para conseguir una primera estabilización.

El InvenSense sensor MPU-6050 contiene un acelerómetro MEMS y un giroscopio MEMS en un solo chip. Es muy precisa, ya que contiene 16 bits de analógico a digital de hardware de conversión para cada canal. Para ello capta la x, y, z y el canal al mismo tiempo. El sensor utiliza el I2C -bus para interconectar con el Arduino en su caso el atmega 328.

El MPU-6050 no es caro, especialmente teniendo en cuenta el hecho de que combina un acelerómetro y un giroscopio.

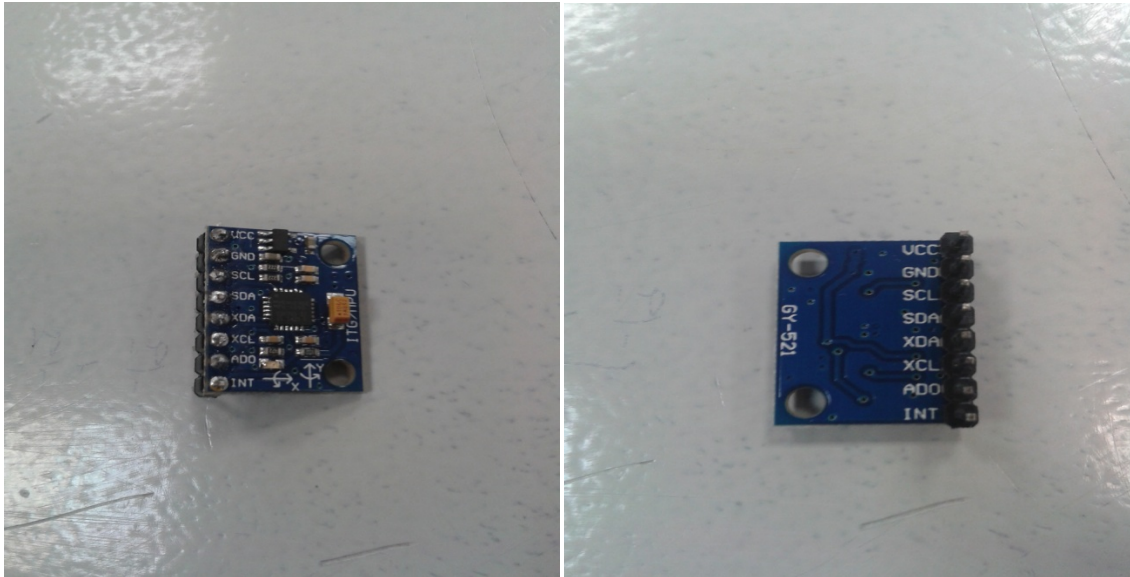


Figura 3.7 sensor MPU 6050

3.6.3 MODULO BLUETOOTH

Para poder controlar el vuelo no se utilizara un control remoto como ya se está acostumbrado en este tipo de drones o robots, en nuestro caso utilizamos un modulo bluetooth HC-05.

Para poder controlar el vuelo es necesario conectarlo a un equipo con sistema operativo android y así con un programa controlar tanto vuelo como velocidad en una tablet o celular Con dicho sistema operativo.

El funcionamiento, configuración y conexión de los módulos HC-05 vienen maestro y esclavo en una misma placa y vía comandos AT se configura cual de los dos modos queremos usar. El chip en si trabaja en 3.3v pero viene soldado sobre una placa que adapta todos sus pines para poder usarlo en 5V.

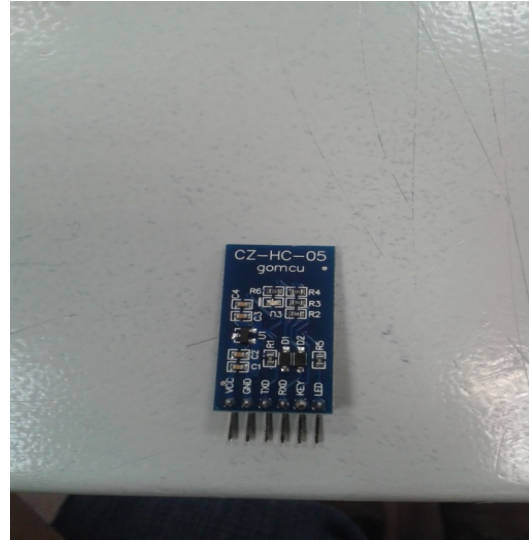


Figura 3.8 modulo bluetooth HC-05

3.7 PESO TOTAL y POR PARTES DEL CUADRICOPTERO

Se requiere saber del peso total de la estructura ya armada, para tener una idea de cuanta corriente tiene que demandar el ESC para que se pueda levantar la estructura.

Tabla 3.1 Pesos del cuadricóptero

Parte	Pieza(s)	Peso (gr)
Estructura	1	425
Motores	1 a 4	1=55 x 4=220 (gr)
ESC	1 a 4	1=32 x 4=128 (gr)
Propelas	1 a 4	1=20 x 4=80 (gr)
Pila	1	240 (gr)
Circuito (Electrónica)	1	40 (gr)
Cámara	1	40 (gr)

La estructura levanta **1600 (gr)** y la suma de los pesos nos da = **1173 (gr) totales**

3.8 CALCULOS

Para nuestros motores vemos que primeramente 'rpm' se refiere al número de rotaciones completadas cada minuto por un cuerpo que gira alrededor de un eje. Kv es el numero de rpm-s que un motor girara, por cada volt aplicado. Al haber elegido una batería de 3 celdas (11.1V) calculamos el número de giros completos que hará el eje de nuestro motor por minuto (revoluciones por minuto de nuestro sistema).

$$kv = \frac{rpm}{v} \text{ (Formula 3.1)}$$

$$rpm = kv * v = (935kv)(volts \text{ bateria } \approx 11.1v) = \mathbf{10378.5 rpm}$$

La capacidad de nuestra batería la medimos por el ratio de descarga. El ratio de descarga C de una batería, es la máxima cantidad de corriente que puede proporcionar.

Nosotros usamos una batería de 3 celdas (3S) de 2800mAh es decir que es capaz de dar 2800mA o (2.8 Amperios) en una hora. 30 C es el ratio de descarga o la máxima cantidad de corriente que puede proporcionar. Podemos calcular la corriente total (I)

$$I_{tot} = I_{pila}(\text{tasa de descarga}) \text{ (Formula 3.2)}$$

$$2800mAh = 2.8A * 30C = 84A$$

Al dividir el valor dado para los 4 motores nos da:

$$\frac{84A}{4} = 21A$$

Este valor nos sirve ya que la pila maneja valores pico y se distribuye a los 4 motores los cuales manejan 15A de I_{max} .

Se pueden calcular los watts que tiene cada motor y los caballos de fuerza:

$$watts = v * I \text{ (Formula 3.3)}$$

$$11.1v * 21A = 233.1$$

Los caballos de fuerza que manejan los motores son:

$$1 \text{ horse power} = 746 \text{ watts} \text{ (Formula 3.4)}$$

$$\frac{233.1}{746} = .3124 \text{ horses}$$

Para el cálculo del tiempo de vuelo primeramente checamos el consumo de los motores a su máxima potencia:

$$P_{max} = V_{bat} * I_{max} \text{ (Formula 3.5)}$$

P_{max} = potencia máxima de motores

V_{bat} = voltaje de la batería

I_{max} = corriente máxima del motor

$$P_{max} = 11.1V * 15A = 166.5 W$$

Ahora se calcula el consumo total de los motores:

$$P_{max4} = P_{max} * 4 \text{ (Formula 3.6)}$$

P_{max4} = potencia máxima de 4 motores

$$P_{max4} = 166.5W * 4 = 666W$$

Entonces ya con estos cálculos podemos sacar la fórmula para determinar el tiempo de vuelo de nuestro cuadricóptero a su máxima potencia:

$$t_{vuelo} = \frac{E}{P_{max4}} \text{ (Formula 3.7)}$$

t_{vuelo} = tiempo de vuelo

E= energía requerida

$$t_{vuelo} = \frac{2.8Ah * 11.1V}{666W} = .0466h$$

Haciendo la conversión

$$min = \frac{h * 60min}{1h} \text{ (Formula 3.8)}$$

Min= minutos

$$min = \frac{.0466 * 60min}{1h} = 2.8min$$

Este es el tiempo volando a máxima potencia, pero tras varias pruebas a media potencia la autonomía puede llegar hasta 4 o 5 min promedio.

3.8.1 CÁLCULOS CIRCUITO

Para nuestro microcontrolador necesitamos un manejo de 5V, pero nosotros alimentaremos con una pila de 9V, es por eso que utilizamos un diodo zener como regulador de voltaje de 5.1V para que los 9V los convierta en 5V.

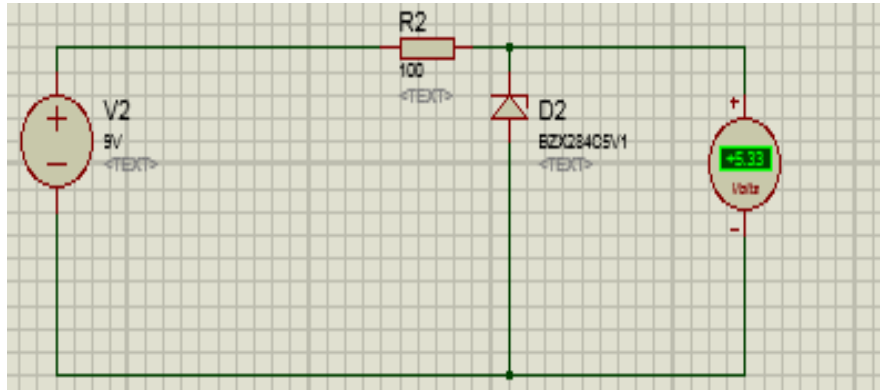


Figura 3.9 Diodo Zener como regulador de voltaje 5.1V

Para encontrar que resistencia usar se realizaron las siguientes formulas.

$$R_s = \frac{V_{in} - V_z}{1.1(I_{lmax})} \text{ (Formula 3.9)}$$

R_s = resistor limitador de corriente

V_{in} = voltaje mínimo de entrada

V_z = voltaje del diodo zener

I_{lmax} = corriente máxima que pedirá la carga

$$R_s = \frac{9v - 5.1v}{1.1(40mA)} = 88.63\Omega \approx 100\Omega$$

Ahora obtenemos la potencia máxima del diodo zener:

$$PD = \left[\frac{(V_{in} - V_z)}{R_s} \right] V_z \text{ (Formula 3.10)}$$

PD=potencia del diodo

$$PD = \left[\frac{(9v - 5.1v)}{100\Omega} \right] 5.1v = .198W \approx 198mW$$

La potencia de la resistencia es:

$$P_{Res} = I^2 * R_s \text{ (Formula 3.11)}$$

I = corriente máxima de la carga

$$P_{Res} = (40mA)^2 * 100\Omega = .16W \approx 160mW$$

Después de calcular la resistencia y los demás parámetros, necesitamos un voltaje de 3.3v para poder alimentar el modulo Bluetooth y el MPU 6050 y realizamos los mismos pasos pero ahora con un diodo zener de 3.3V

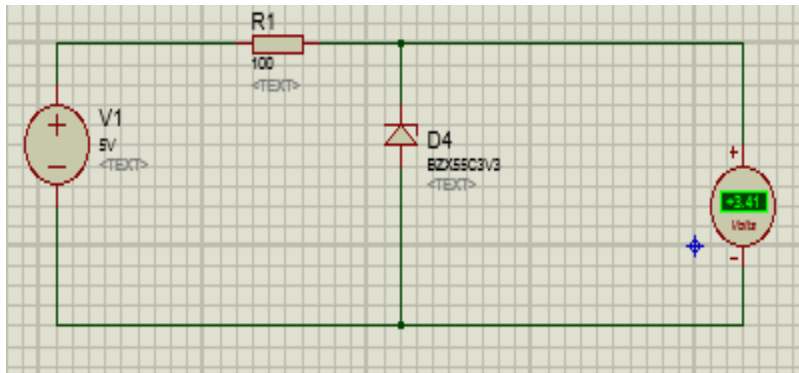


Figura 3.10 Diodo Zener como regulador de voltaje 3.3V

Para la resistencia:

$$R_s = \frac{9v - 3.3v}{1.1(50mA)} = 103.63\Omega \approx 100\Omega$$

Potencia máxima del diodo zener:

$$PD = \left[\frac{(9v - 3.3v)}{100\Omega} \right] 3.3v = .1881W \approx 188mW$$

La potencia de la resistencia es:

$$P_{Res} = (50mA)^2 * 100\Omega = .25W \approx 250mW$$

Seguidamente del cálculo de los diodos zener como reguladores, seguimos con el Pull-Up el cual calculamos sus resistencia con la ley de ohm.

$$R = \frac{V_{in}}{I_{in}} \text{ (Formula 3.12)}$$

$$R = \frac{5.1v}{500\mu A} = 10.2 \times 10^3 \approx 10K\Omega$$

Ahora calculamos el led para revisar que este energizado nuestro circuito.

$$R_{led} = \frac{V_{in} - V_{led}}{I_{led}} \text{ (Formula 3.13)}$$

R_{led} = resistencia del led

V_{in} = voltaje de entrada

V_{led} = voltaje led

I_{led} = corriente led

$$R_{led} = \frac{5.1v - 1.7v}{10mA} = 340\Omega \approx 330\Omega$$

Para la potencia de la resistencia es:

$$P_{Rled} = I_{Rled}^2 * R_{led} \text{ (Formula 3.14)}$$

P_{Rled} = potencia de resistencia de led

I_{Rled} = corriente de resistencia del led

$$P_{Rled} = 10mA^2 * 330\Omega = .033W \approx 33mW$$

Los capacitores de desacoplo son necesarios para contra restar el efecto de inductancia a la salida de la fuente de alimentación.

Una forma de calcularlos es la siguiente:

$$q = \frac{I_{fl} * t_c}{2} \text{ (Formula 3.15)}$$

$$C_d = \frac{q}{V_{fl}} \text{ (Formula 3.16)}$$

q = carga dada en coloumb

I_{fl} = corriente dada en amperes

t_c = Tiempo de comnmutacion dada en segundos

C_d = Capacitancia de desacoplo dad en Faradios

V_{fl} = voltaje de flutuacion dada en Volts

Calculando los valores con los datos que nos da el fabricante del ATmega328

$$t_c = 0.0000005 s$$

Se propone un una corriente de fluctuación y un voltaje de fluctuación.

$$I_{fl} = .01 A$$

$$V_{fl} = 0.05 V$$

Sustituyendo valores se obtiene

$$q = \frac{0.01A * 0.0000005}{2} = 2.5 \times 10^{-9} C$$

$$C_d = \frac{2.5 \times 10^{-9} C}{0.05 V} = 5 \times 10^{-8} F$$

$$C_d = 50 nF$$

En la práctica se utilizan capacitores de desacoplo de 10 a 100 nF. En este caso utilizamos 100nF

En base a lo ya visto con anterioridad la dinámica que manejamos para nuestro cuadricóptero es basada en:

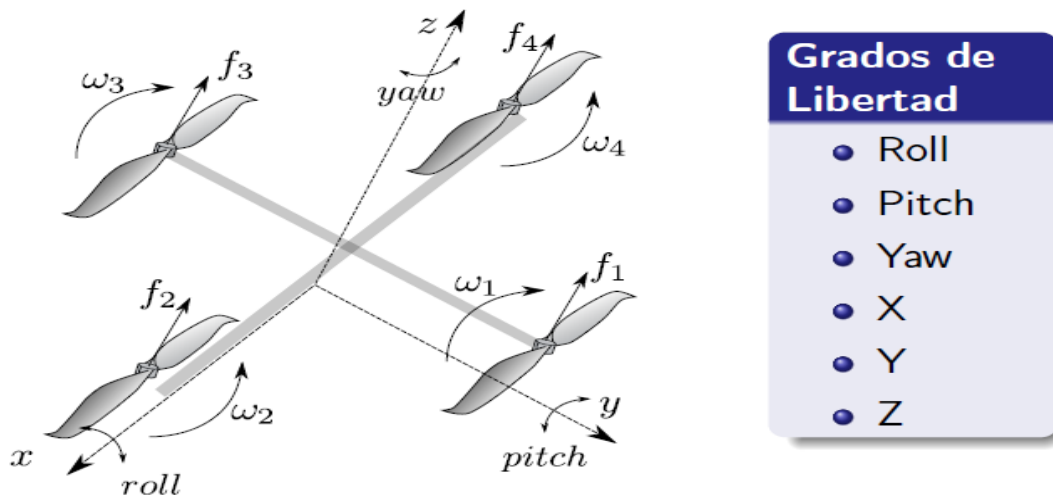


Figura 3.11 función cuadricóptero

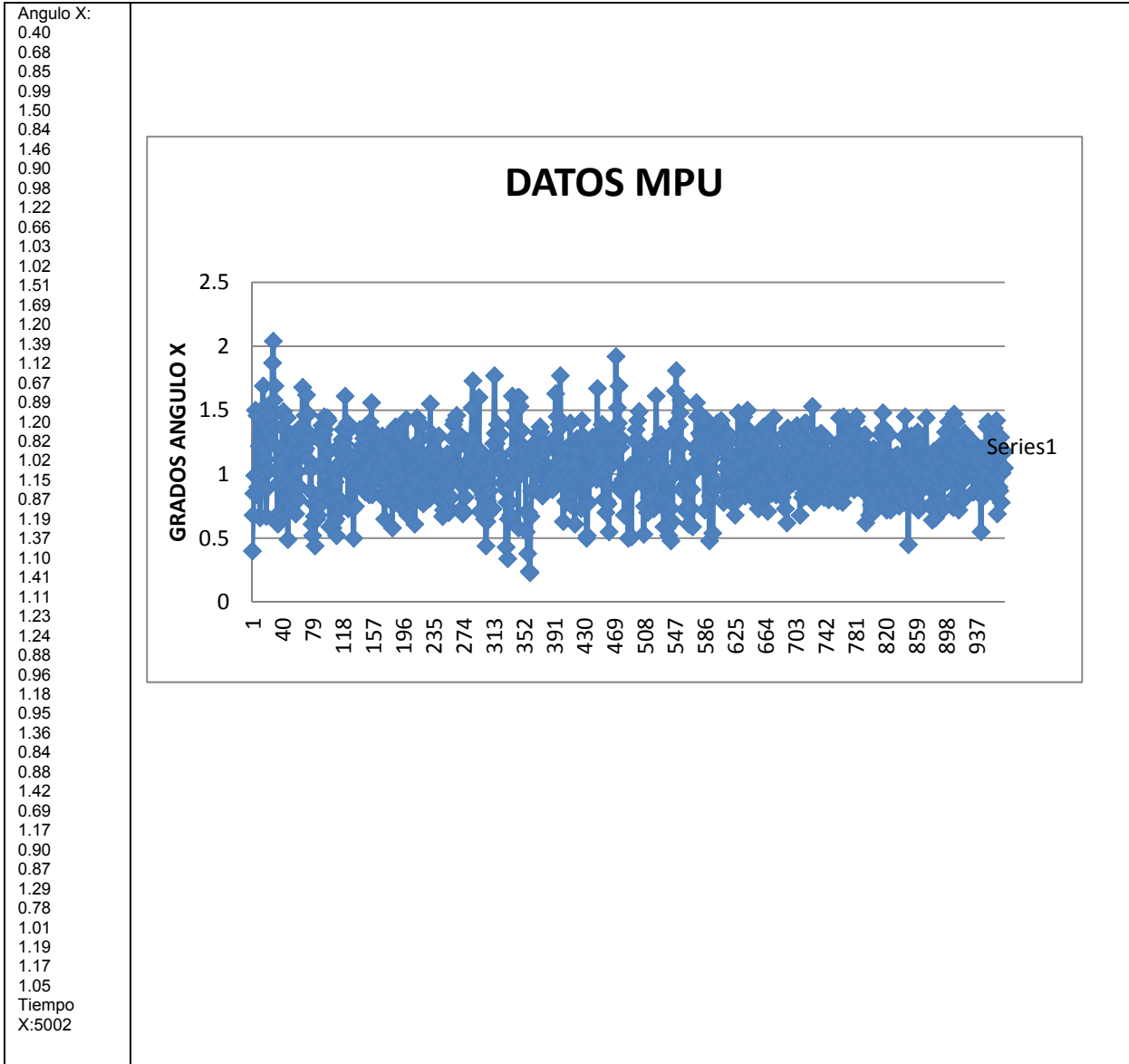
Y con esto podemos dar los cálculos de movimiento y de estabilidad para que nuestro cuadricóptero tenga una elevación y equilibrio óptimo; es por eso que se realizaron cálculos de la respuesta de nuestro MPU 6050 sin motor para probar que tan estable puede quedar en el momento de vuelo.

RESPUESTA DEL MPU
SIN MOTOORES NI MOVIMIENTO

Pulsar 'A' para iniciar...

Listo...

Iniciando Test de respuesta de angulo X & Y



GRAFICA SIN PERTURBACIONES

EN ESTA GRAFICA SE MUESTRA EL MPU SIN MOVIMIENTO ALGUNO DURANTE 5 SEGUNDOS

ES DECIR 973 DATOS ES A 5 SEGUNDOS

EN LA GRAFICA SE PUEDE OBSERVAR QUE EL VALOR DEL ANGULO EJE X SE ENCUENTRA ENTRE 0.5 Y 1.5

ES DECIR ESTOS VALORES ESTAN SOBRE LOS 1 GRADOS QUE TENEMOS DE REFERENCIA DEL PMU

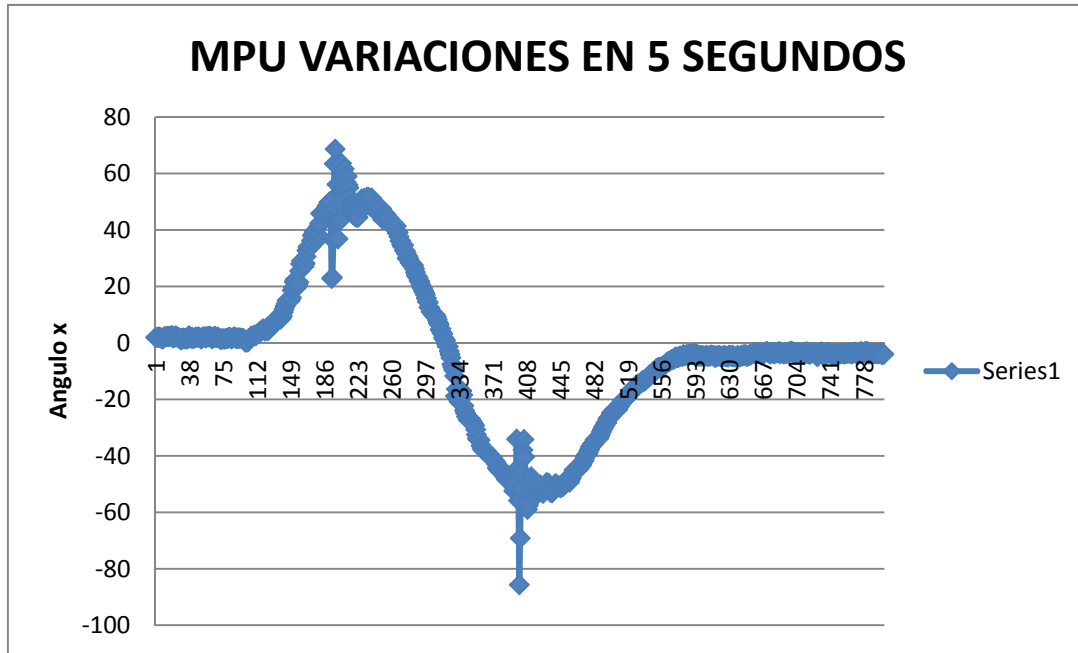
TENIENDO LA DIFERENCIA ANGULO SUPERIOR = $1.5 - 1 = 0.5$

TENIENDO LA DIFERENCIA ANGULO INFERIOR = $1 - 0.5 = 0.5$

LA GRAFICA MUESTRA QUE EL MPU LEE ± 0.5 GRADOS

Grafica 1: eje X MPU

Angulo
 X:
 1.93
 2.02
 1.96
 1.96
 2.20
 1.84
 1.70
 1.41
 1.43
 2.14
 2.00
 2.27
 2.32
 2.26
 2.23
 2.28
 2.40
 2.38
 2.63
 1.78
 2.09
 1.98
 2.55
 2.22
 1.99
 1.88
 1.70
 1.42
 20.50
 21.51
 25.47
 27.99
 29.06
 30.56
 32.70
 3.68
 3.15
 1.84
 1.22
 0.86
 -0.20
 -1.17
 -1.32
 -2.94
 -4.08
 -5.32
 -7.57
 -3.65
 -3.68
 -3.85
 -3.74
 -3.91
 -3.67
 -3.68
 -3.46
 -4.20
 -3.96
 -3.98
 Tiempo
 X:5003

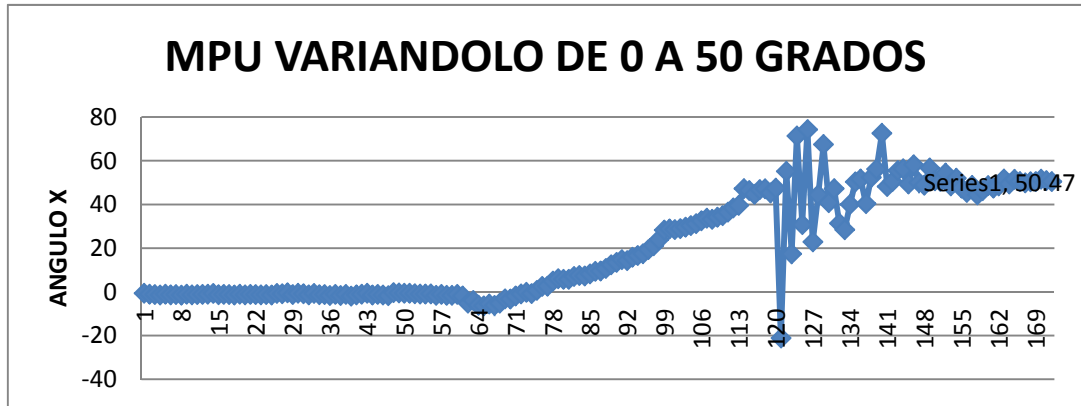


EN ESTA GRAFICA SON 800 LINEAS QUE REPRESENTAN 5 SEGUNDOS
 SE VEN LOS CAMBIOS DE ANGULO DE 50 A -50 GRADOS

A CONTINUACION PARA VER MEJOR LOS CAMBIOS DE ANGULO SE ARAN LAS GRAFICAS A 1 SEGUNDO

Grafica 2: eje X MPU

Angulo
X:
-0.65
-1.07
-1.24
-1.42
-1.06
-1.40
-1.26
-1.53
-1.07
-1.88
-0.96
-0.21
-0.80
0.70
2.50
2.57
4.96
6.04
10.71
12.42
13.48
14.77
14.33
15.81
16.64
17.43
19.36
21.13
44.41
46.27
48.62
47.48
48.51
51.61
49.41
51.49
50.45
49.87
50.39
50.50
51.60
50.90
50.47
Tiempo
X:1003



GRAFICA CON 172 MEDICIONES QUE ESTAN TOMADAS EN 1 SGUNDO, CADA MEDICION EQUIVALE 5.81 ms
EL BRAZO DONDE ESTA EL MPU SE MOVIO DE 0 A 50 GRADOS

SE TOMA EL PUNTO 73=424.13 ms CON VALOR -0.21
SE TOMA EL PUNTO 120=697.2 ms CON VALOR 45.25
VELOCIDAD=(45.25 - (-0.21))/697.2-424.13= 45.46 GRADOS/273.07 ms =166.47 GRADOS/SEGUNDO

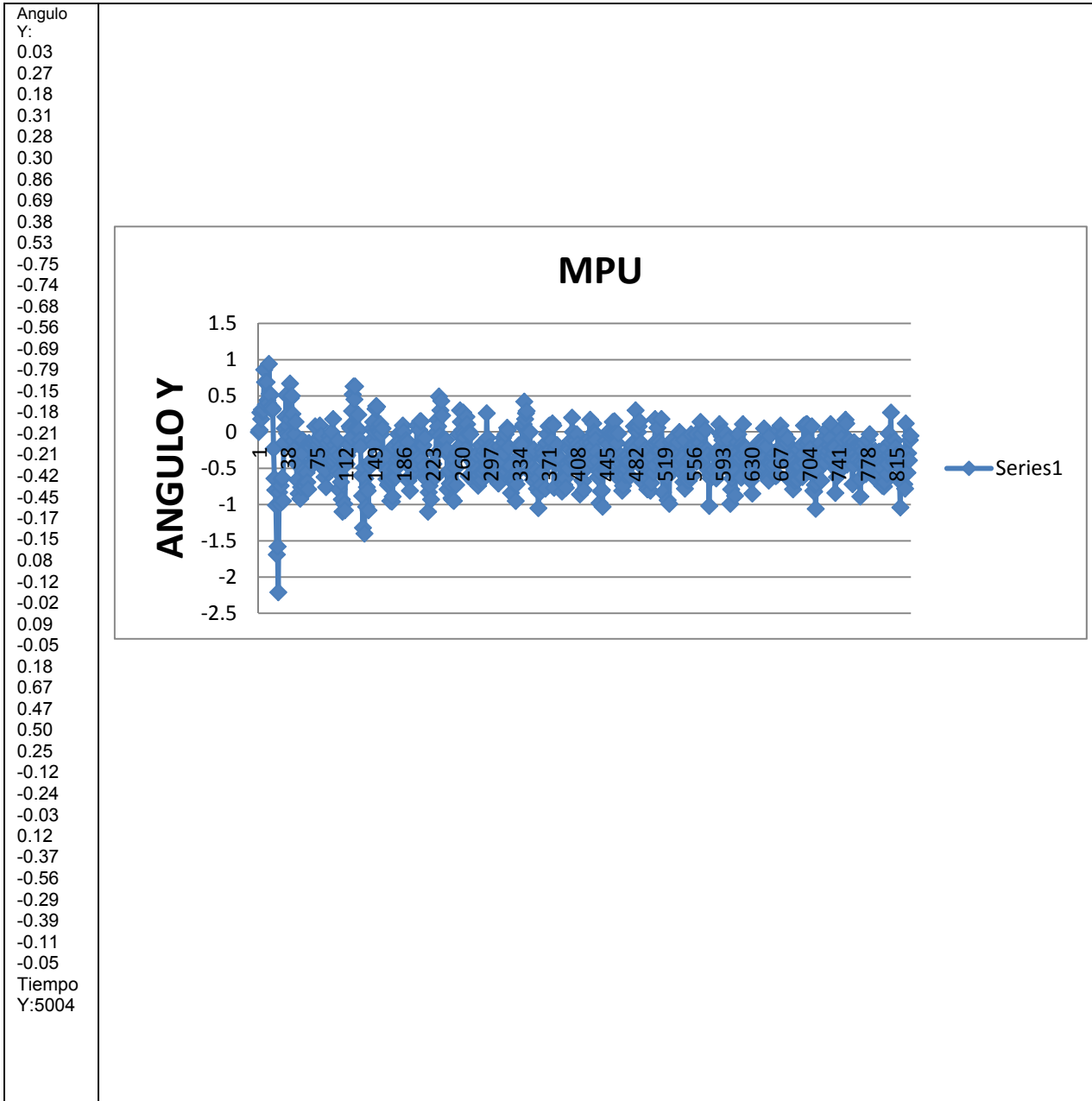
CON UNA VELOCIDAD DE 166.47 GRADOS/SEGUNDO LEYO 120 - 73 = 47 DATOS
ESTO TOMADO IDEALMENTE DE 0 A 50 GRADOS

Grafica 3: eje X MPU

Pulsar 'A' para iniciar...

Listo...

Iniciando Test de respuesta de angulo X & Y



GRAFICA SIN PERTURBACIONES

EN ESTA GRAFICA SE MUESTRA EL MPU SIN MOVIMIENTO ALGUNO DURANTE 5 SEGUNDOS

ES DECIR 833 DATOS ES A 5 SEGUNDOS

EN LA GRAFICA SE PUEDE OBSERVAR QUE EL VALOR DEL ANGULO EJE Y SE ENCUENTRA ENTRE -1 Y 0.25

ES DECIR ESTOS VALORES ESTAN SOBRE LOS -0.625 GRADOS QUE TENEMOS DE REFERENCIA DEL PMU

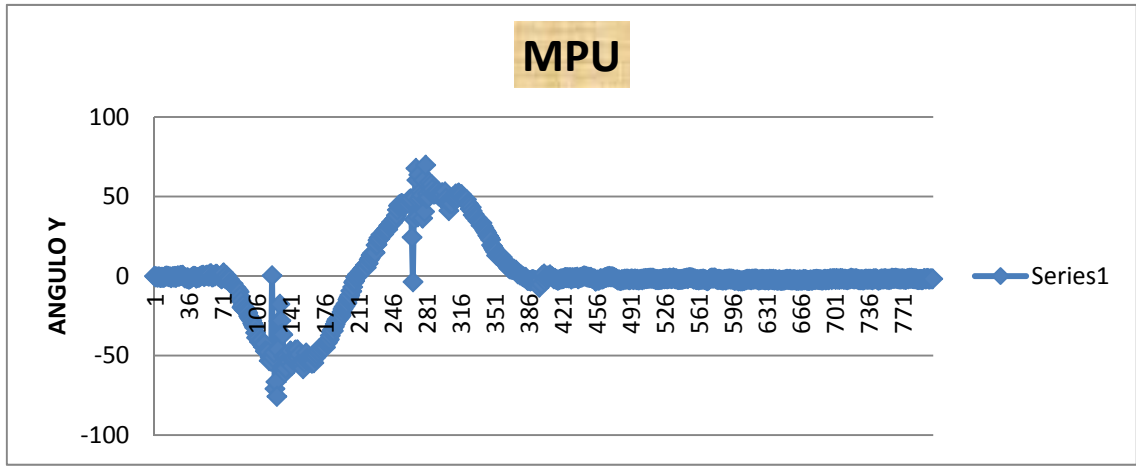
TENIENDO LA DIFERENCIA ANGULO SUPERIOR $= 0.25 - 0.625 = 0.375$

TENIENDO LA DIFERENCIA ANGULO INFERIOR $= -0.625 + 1 = 0.375$

LA GRAFICA MUESTRA QUE EL MPU LEE ± 0.375 GRADOS

Grafica 4: eje Y MPU

Angulo
 Y:0.00
 -0.56
 -0.45
 -0.71
 -0.47
 -0.51
 -0.95
 -0.74
 -0.69
 -0.88
 -0.70
 -0.17
 0.06
 -57.97
 -50.37
 -52.34
 -52.54
 -54.19
 -47.01
 -56.37
 -52.54
 -52.20
 -51.30
 -46.39
 -49.40
 -46.33
 -51.64
 -49.19
 -54.98
 18.69
 17.40
 16.41
 15.22
 12.90
 12.87
 13.05
 12.63
 -2.42
 -2.63
 -2.51
 -2.29
 -2.62
 -2.76
 -2.66
 -2.59
 -2.55
 -2.39
 -1.65
 -1.74
 -1.74
 Tiempo
 Y:5000

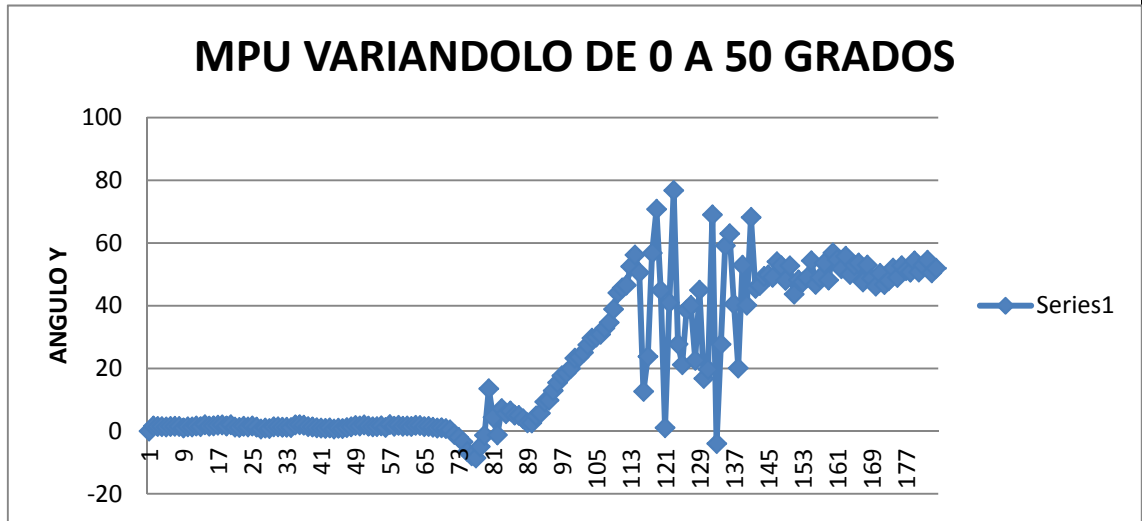


EN ESTA GRAFICA SON 800 LINEAS QUE REPRESENTAN 5 SEGUNDOS
 SE VEN LOS CAMBIOS DE ANGULO DE 50 A -50 GRADOS

A CONTINUACION PARA VER MEJOR LOS CAMBIOS DE ANGULO SE ARAN LAS GRAFICAS A 1 SEGUNDO

Grafica 5: eje Y MPU

Angulo
 Y:0.00
 1.66
 1.50
 1.43
 1.31
 1.50
 1.52
 1.48
 1.02
 1.38
 1.32
 1.62
 1.44
 2.06
 1.60
 1.67
 1.89
 1.98
 1.62
 2.04
 1.29
 1.16
 1.53
 1.32
 1.65
 40.53
 20.13
 52.91
 40.16
 49.86
 48.08
 52.72
 43.67
 48.23
 47.59
 48.94
 54.35
 46.79
 48.99
 53.62
 48.21
 56.84
 54.70
 51.80
 51.21
 50.70
 54.33
 50.74
 52.88
 54.49
 50.46
 51.98
 Tiempo
 Y:1002



GRAFICA CON 184 MEDICIONES QUE ESTAN TOMADAS EN 1 SGUNDO, CADA MEDICION EQUIVALE 5.44ms
 EL BRAZO DOND ESTA EM MPU SE MOVIO DE 0 A 50 GRADOS

SE TOMA EL PUNTO 89=484.16 ms CON VALOR 2.67
 SE TOMA EL PUNTO 113= 614.72ms CON VALOR 52.52
 $VELOCIDA = (52.52 - 2.67) / (614.72 - 484.16) = 49.85 \text{ GRADOS} / 130.56 \text{ ms} = 381 \text{ GRADOS} / \text{SEGUNDO}$

CON UNA VELOCIDAD DE 381 GRADOS/SEGUNDO LEYO 113-89 = 24 DATOS
 ESTO TOMADO IDEALMNTNE DE 0 A 50 GRADOS

Grafica 6: eje Y MPU

Para calcular el calibre de los cables se considera la siguiente tabla:

Tabla 3.2 Calibre cable

Calibre	Inch	A
10	.1019	14.8
11	.0907	11.08
12	.0808	9.3
13	.0720	7.4
14	.0641	5.8
15	.0571	4.6
16	.0508	3.6
17	.0453	2.9
18	.0403	2.3
19	.0359	1.8
20	.0320	1.4
21	.0285	1.1
22	.0253	.91
23	.0226	.72
24	.0201	.57
25	.0179	.45
26	.0159	.36
27	.0142	.28
28	.0133	.22
29	.0126	.18
30	.01	.14

De esta forma podemos ver que el calibre que manejamos es del N°16 para nuestros controladores o ESC.

Para nuestros anchos de pista nos basamos en una tabla que muestra esencialmente la corriente máxima que utilizaremos, el ancho de pista en pulgadas como en milímetros y la densidad del cobre que manejaremos.

Tabla 3.3 Anchos de pista

Corriente máxima		Ancho de la pista		Densidad de la capa de cobre (CU)
A	mA	Inch	Mm	Oz/H ²
.13	130	.005	.127	.5
.5	500	.01	.254	.5
.7	700	.02	.508	.5
1	1000	.03	.762	.5
.5	500	.005	.127	1
.8	800	.01	.254	1
1.4	1400	.02	.508	1
1.9	1900	.03	.762	1
.7	700	.005	.127	2
1.4	1400	.01	.254	2
2.2	2200	.02	.508	2
4	4000	.03	.762	2
3	3000	.025	.127	3

Para el cálculo del ancho de pista en mm se hace una regla de tres:

1 pulgada = 25.4 mm

$$\text{ejemplo: } \frac{.01 * 25.4}{.03} = .76$$

3.9 ESQUEMÁTICO Y PCB DEL PROYECTO

3.9.1 ESQUEMÁTICO

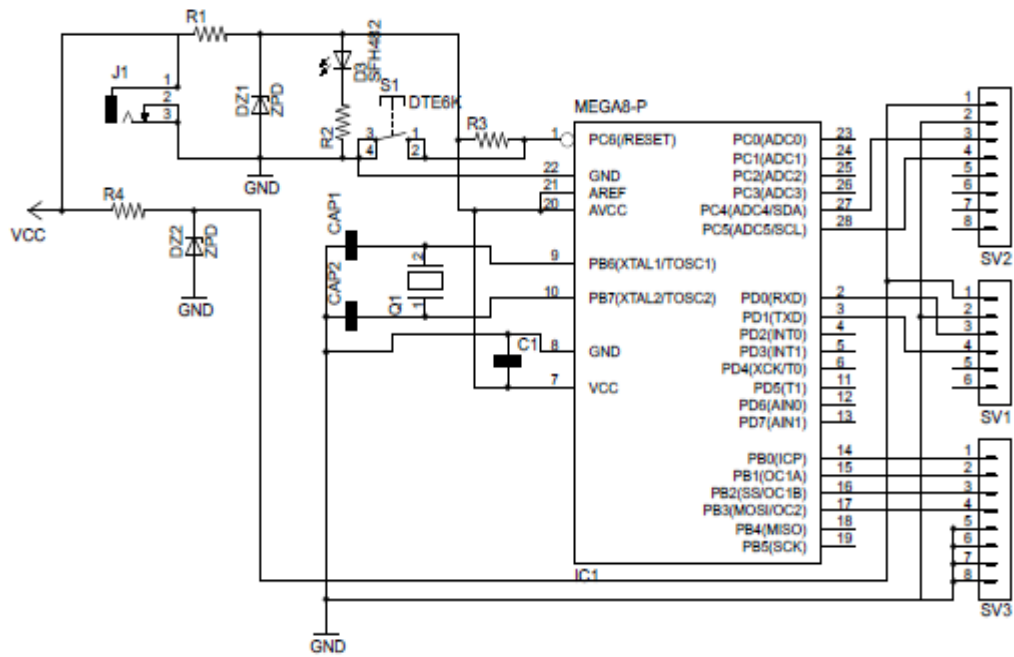


Diagrama N°2 esquemático de placa

3.9.2 PCB

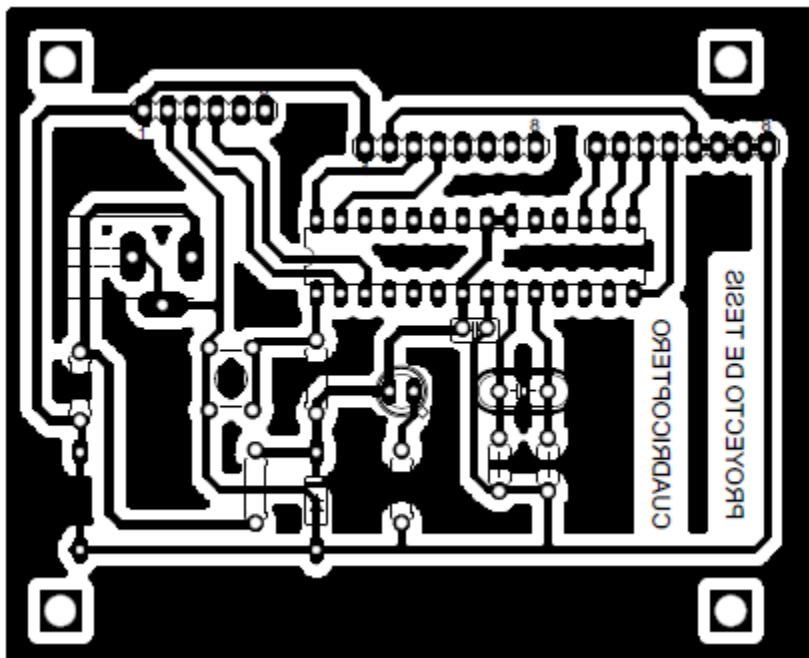


Figura 3.12 PCB

3.10 PROGRAMA DE CONTROL

```
1. //*****
   //*****/
2. /*Programa con el que se evalua y controa el desempeño de los motores 1 & 3, 2 & 4 que conforman
   el movimiento del ejeX como el ejeY */
3. //*****
   //*****/
4. #include <Servo.h>
5. #include <Wire.h>
6. #define PINMOTOR1 11
7. #define PINMOTOR3 9
8. #define PINMOTOR2 10
9. #define PINMOTOR4 6
10. #define MAXIMOPWM 2000
11. #define MINIMOPWM 1000
12. #define BAUD 9600
13. #define Paso 5
14. //*****
15. int NumInstruccion;
16. boolean iniciado = false;
17. //*****
18. int BufferMotor1 = 0;
19. int BufferMotor3 = 0;
20. int BufferMotor2 = 0;
21. int BufferMotor4 = 0;
22. Servo servo1;
23. Servo servo3;
24. Servo servo2;
25. Servo servo4;
26. //*****
27. byte recibiendoByte ;
28. //-----MPU-----
29. #define MPU 0x68 //Direccion I2C de la IMU
30. #define A_R 16384.0 //Ratios de conversion
31. #define G_R 131.0
32. #define RAD_A_DEG = 57.295779 //Conversion de radianes a grados 180/PI
33. //MPU-6050 da los valores en enteros de 16 bits
34. //Valores sin refinar
35. int16_t AcX, AcY, AcZ, GyX, GyY, GyZ; // Variables enteras de 16 bits
36. //Angulos AccX,AccY,GyX,GyY,AngleX & AngleY
37. float Acc[2];
38. float Gy[2];
39. float Angle[2];
40. //-----
41. //-----INICIALIZACION-----
42. void setup()
43. {
44.   Serial.begin(BAUD);
45.   //
46.   //-----
47.   Wire.begin();
48.   Wire.beginTransmission(MPU);
49.   Wire.write(0x6B);
50.   Wire.write(0);
51.   Wire.endTransmission(true);
52.   //-----
53.   //Inicianzo el ESC en el PIN determinado, se crea servo 'n' como parte de libreria servo.h
54.   //-----
55.   servo1.attach(PINMOTOR1);
56.   servo3.attach(PINMOTOR3);
```

```

57. servo2.attach(PINMOTOR2);
58. servo4.attach(PINMOTOR4);
59. //-----
60. Serial.println ("Pulsar 'A' para iniciar... \n");
61. while ( iniciado==false )
62. {
63.   //---inicializando motores-----
64.   servo1.writeMicroseconds(0);
65.   servo3.writeMicroseconds(0);
66.   servo2.writeMicroseconds(0);
67.   servo4.writeMicroseconds(0);
68.   BufferMotor1=constrain(BufferMotor1 , MINIMOPWM, MAXIMOPWM);
69.   BufferMotor3=constrain(BufferMotor3 , MINIMOPWM, MAXIMOPWM);
70.   BufferMotor2=constrain(BufferMotor2 , MINIMOPWM, MAXIMOPWM);
71.   BufferMotor4=constrain(BufferMotor4 , MINIMOPWM, MAXIMOPWM);
72.   //-----
73.   recibiendoByte = Serial.read();
74.   if (recibiendoByte == 65 || recibiendoByte ==97)
75.   {
76.     iniciado=true;
77.   }
78. }
79. //-----
80. //
81. servo1.writeMicroseconds(BufferMotor1);
82. servo3.writeMicroseconds(BufferMotor3);
83. servo2.writeMicroseconds(BufferMotor2);
84. servo4.writeMicroseconds(BufferMotor4);
85. //-----MENU-----
86. Serial.println ("Esperando el pitido de confirmacion... \n");
87. delay(1500);
88. Serial.println ("Listo... \n ");
89. Serial.println (" Pulsar 'A' para acelerar Motores 1 & 3 \n Pulsar 'Z' para desacelerar Motores 1 & 3
\n");
90. Serial.println (" Pulsar 'D' para acelerar Motor 1 \n Pulsar 'C' para desacelerar Motor 1 \n");
91. Serial.println (" Pulsar 'F' para acelerar Motor 3 \n Pulsar 'V' para desacelerar Motor 3 \n");
92. Serial.println (" Pulsar 'S' para parar motores 1 & 3 \n");
93. }
94. //----Cico infinito-----
95. void loop()
96. {
97.   LecturaSensor();
98.   //if(Acc[0]>15 || Acc[0]<-15 || Acc[1]>1 5 || Acc[0]<-15){StopMotores();}
99.   NumInstruccion =RecibeInstruccion();
100.  if(NumInstruccion>0)
101.  {
102.    switch (NumInstruccion)
103.    {
104. //----Orden numero 1 AUMENTA la velocidad de motores 1,2,3&4-----
105.    case 1:
106.      AumentaVelocidadMotor1();
107.      AumentaVelocidadMotor3();
108.      AumentaVelocidadMotor2();
109.      AumentaVelocidadMotor4();
110.      break;
111. //-----
112. //----Orden numero 2 DESMINUYE la velocidad de motores 1,2,3&4-----
113.    case 2:
114.      DisminuirVelocidadMotor1();
115.      DisminuirVelocidadMotor3();
116.      DisminuirVelocidadMotor2();
117.      DisminuirVelocidadMotor4();

```

```

118.   break;
119.//-----
120.//----Orden numero 3 AUMENTA la velocidad de motor 1-----
121.   case 3:
122.     AumentaVelocidadMotor1();
123.     break;
124.//-----
125.//----Orden numero 4 DESMINUYE la velocidad de motor 1-----
126.   case 4:
127.     DisminuirVelocidadMotor1();
128.     break;
129.//-----
130.//----Orden numero 5 AUMENTA la velocidad de motor 3-----
131.   case 5:
132.     AumentaVelocidadMotor3();
133.     break;
134.//-----
135.//----Orden numero 6 DESMINUYE la velocidad de motor 3-----
136.   case 6:
137.     DisminuirVelocidadMotor3();
138.     break;
139.//-----
140.//----Orden numero 7 AUMENTA la velocidad de motor 2-----
141.   case 7:
142.     AumentaVelocidadMotor2();
143.     break;
144.//-----
145.//----Orden numero 8 DESMINUYE la velocidad de motor 2-----
146.   case 8:
147.     DisminuirVelocidadMotor2();
148.     break;
149.//-----
150.//----Orden numero 9 AUMENTA la velocidad de motor 4-----
151.   case 9:
152.     AumentaVelocidadMotor4();
153.     break;
154.//-----
155.//----Orden numero 10 DESMINUYE la velocidad de motor 4-----
156.   case 10:
157.     DisminuirVelocidadMotor4();
158.     break;
159.//-----
160.//----Orden numero 7 paro de motores-----
161.   case 11:
162.     StopMotores();
163.     break;
164.//-----
165.   }// Imprime los pulsos de cada comotor y los angulos que otorga el MPU
166.   Serial.print("BufferMotor1-->");Serial.print(BufferMotor1);Serial.print("\n");
167.   Serial.print("BufferMotor2-->");Serial.print(BufferMotor2);Serial.print("\n");
168.   Serial.print("BufferMotor3-->");Serial.print(BufferMotor3);Serial.print("\n");
169.   Serial.print("BufferMotor4-->");Serial.print(BufferMotor4);Serial.print("\n");
170.   Serial.print("Angulo X-->");Serial.print(Angle[0]);Serial.print("\n");
171.   Serial.print("Angulo Y-->");Serial.print(Angle[1]);Serial.print("\n");
172.   }
173.   delay(5);
174.}
175.//Funcion que envia un entero
176.int RecibeInstruccion()
177. {
178.   int orden=0; // Variable que sera retornada de la funcion
179.   if (Serial.available() > 0)// Se detecta el teclado si es mayor a cero entra

```



```

180. {
181.   recibiendoByte = Serial.read(); // Leemos el Byte recibido
182.   /******
183.   /*A continuacion se hace una comparacion del byte recibido
184.   /*para que la variable orden sea retornada al void loop y se guarde en NumInstruccion
185.   *///*****
186.   switch (recibiendoByte)
187.   {
188.     case 'A':
189.       orden = 1;
190.       break;
191.     case 'Z':
192.       orden = 2;
193.       break;
194.     case 'D':
195.       orden = 3;
196.       break;
197.     case 'C':
198.       orden = 4;
199.       break;
200.     case 'F':
201.       orden = 5;
202.       break;
203.     case 'V':
204.       orden = 6;
205.       break;
206.     case 'G':
207.       orden = 7;
208.       break;
209.     case 'B':
210.       orden = 8;
211.       break;
212.     case 'H':
213.       orden = 9;
214.       break;
215.     case 'N':
216.       orden = 10;
217.       break;
218.     case 'S':
219.       orden = 11;
220.       break;
221.   }
222.   //Se termina el switch
223. }
224. return (orden);
225. }
226.//-----
227.//*****
228.///-----FUNCIONES PARA ECELERARA & DESACELERAR EL MOTOR NUMERO
    UNO -----//
229.// Funcion para aumentar la velocidad del Motor1-----
230.void AumentaVelocidadMotor1()
231.{
232.   BufferMotor1=BufferMotor1 + Paso;
233.   //
234.   //
235.   //
236.   BufferMotor1=constrain(BufferMotor1 , MINIMOPWM, MAXIMOPWM);
237.   //
238.   servo1.writeMicroseconds(BufferMotor1);
239. }
240.//-----

```

```

241.// Funcion para disminuir la velocidad del Motor1-----
242.void DisminuirVelocidadMotor1()
243.{
244.    BufferMotor1=BufferMotor1 - Paso;
245.    //
246.    //
247.    //
248.    BufferMotor1=constrain(BufferMotor1 , MINIMOPWM, MAXIMOPWM);
249.    //
250.    servo1.writeMicroseconds(BufferMotor1);
251.}
252.//-----
253.//-----FUNCIONES PARA ECELERARA & DESACELERAR EL MOTOR NUMERO
    TRES -----//
254.// Funcion para aumentar la velocidad del Motor3-----
255.void AumentaVelocidadMotor3()
256.{
257.    BufferMotor3=BufferMotor3 + Paso;
258.    //
259.    //
260.    //
261.    BufferMotor3=constrain(BufferMotor3 , MINIMOPWM, MAXIMOPWM);
262.    //
263.    servo3.writeMicroseconds(BufferMotor3);
264. }
265.//-----
266.// Funcion para disminuir la velocidad del Motor3-----
267.void DisminuirVelocidadMotor3()
268.{
269.    BufferMotor3=BufferMotor3 - Paso;
270.    //
271.    //
272.    //
273.    BufferMotor3=constrain(BufferMotor3 , MINIMOPWM, MAXIMOPWM);
274.    //
275.    servo3.writeMicroseconds(BufferMotor3);
276. }
277.//-----
278.//-----FUNCIONES PARA ECELERARA & DESACELERAR EL MOTOR NUMERO
    DOS -----//
279.// Funcion para aumentar la velocidad del Motor2-----
280.void AumentaVelocidadMotor2()
281.{
282.    BufferMotor2=BufferMotor2 + Paso;
283.    //
284.    //
285.    //
286.    BufferMotor2=constrain(BufferMotor2 , MINIMOPWM, MAXIMOPWM);
287.    //
288.    servo2.writeMicroseconds(BufferMotor2);
289. }
290.//-----
291.// Funcion para disminuir la velocidad del Motor1-----
292.void DisminuirVelocidadMotor2()
293.{
294.    BufferMotor2=BufferMotor2 - Paso;
295.    //
296.    //
297.    //
298.    BufferMotor2=constrain(BufferMotor2 , MINIMOPWM, MAXIMOPWM);
299.    //
300.    servo2.writeMicroseconds(BufferMotor2);

```

```

301.}
302.//-----
303.//-----FUNCIONES PARA ECELERARA & DESACELERAR EL MOTOR NUMERO
    CUATRO -----//
304.// Funcion para aumentar la velocidad del Motor4-----
305.void AumentaVelocidadMotor4()
306.{
307.    BufferMotor4=BufferMotor4 + Paso;
308.    //
309.    //
310.    //
311.    BufferMotor4=constrain(BufferMotor4 , MINIMOPWM, MAXIMOPWM);
312.    //
313.    servo4.writeMicroseconds(BufferMotor4);
314. }
315.//-----
316.// Funcion para disminuir la velocidad del Motor4-----
317.void DisminuirVelocidadMotor4()
318.{
319.    BufferMotor4=BufferMotor4 - Paso;
320.    //
321.    //
322.    //
323.    BufferMotor4=constrain(BufferMotor4 , MINIMOPWM, MAXIMOPWM);
324.    //
325.    servo4.writeMicroseconds(BufferMotor2);
326.}
327.//-----
328.//Funcion para frenar todos los motores
329.void StopMotores()
330.{
331.    BufferMotor1=0;
332.    BufferMotor3=0;
333.    BufferMotor2=0;
334.    BufferMotor4=0;
335.    BufferMotor1=constrain(BufferMotor1 , MINIMOPWM, MAXIMOPWM);
336.    BufferMotor3=constrain(BufferMotor3 , MINIMOPWM, MAXIMOPWM);
337.    BufferMotor2=constrain(BufferMotor2 , MINIMOPWM, MAXIMOPWM);
338.    BufferMotor4=constrain(BufferMotor4 , MINIMOPWM, MAXIMOPWM);
339.    servo1.writeMicroseconds(BufferMotor1);
340.    servo3.writeMicroseconds(BufferMotor3);
341.    servo2.writeMicroseconds(BufferMotor2);
342.    servo4.writeMicroseconds(BufferMotor4);
343. }
344.//-----
345. void LecturaSensor()
346.{
347. //Leer los valores del Acelerometro de la IMU
348. Wire.beginTransmission(MPU);
349. Wire.write(0x3B);
350. Wire.endTransmission(false);
351. Wire.requestFrom(MPU,6,true);
352. AcX=Wire.read()<<8|Wire.read();
353. AcY=Wire.read()<<8|Wire.read();
354. AcZ=Wire.read()<<8|Wire.read();
355. //Se calculan los angulos Y, X respectivamente.
356. Acc[1] = atan(-1*(AcX/A_R)/sqrt(pow((AcY/A_R),2) + pow((AcZ/A_R),2)))*RAD_TO_DEG;
357. Acc[0] = atan((AcY/A_R)/sqrt(pow((AcX/A_R),2) + pow((AcZ/A_R),2)))*RAD_TO_DEG;
358. //Leer los valores del Giroscopio
359. Wire.beginTransmission(MPU);
360. Wire.write(0x43);
361. Wire.endTransmission(false);

```

```

362. Wire.requestFrom(MPU,4,true);
363. GyX=Wire.read()<<8|Wire.read();
364. GyY=Wire.read()<<8|Wire.read();
365. //Calculo del angulo del Giroscopio
366. Gy[0] = GyX/G_R;
367. Gy[1] = GyY/G_R;
368. //Aplicar el Filtro Complementario
369. Angle[0] = 0.98 *(Angle[0]+Gy[0]*0.010) + 0.02*Acc[0]; //Angulo X
370. Angle[1] = 0.98 *(Angle[1]+Gy[1]*0.010) + 0.02*Acc[1]; //Angulo Y
371. }

```

3.11 APLICACIÓN CON ANDROID

Nosotros realizamos un control remoto basado en la tecnología Android, ya que por este medio hicimos pruebas de encendido, elevación, descenso y frenado (stop). Con el módulo de bluetooth HC-05 comunicamos nuestro cuadricóptero a cualquier dispositivo que cuente con sistema operativo Android.

La aplicación se realizó mediante un programa llamado MIT App Inventor 2 el cual es un entorno muy sencillo de manejar y crea aplicaciones exclusivamente para el sistema operativo Android.

En la siguiente figura se presenta como se verá la aplicación de control en el teléfono:

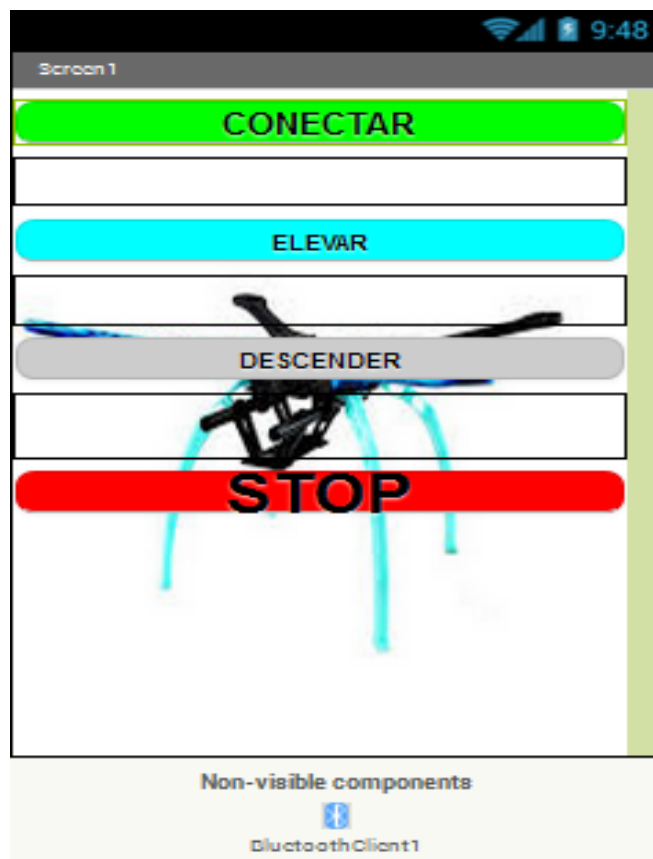


Figura 3.13 presentación de aplicación

A continuación mostramos el entorno de trabajo con el cual realizamos nuestra aplicación Android.

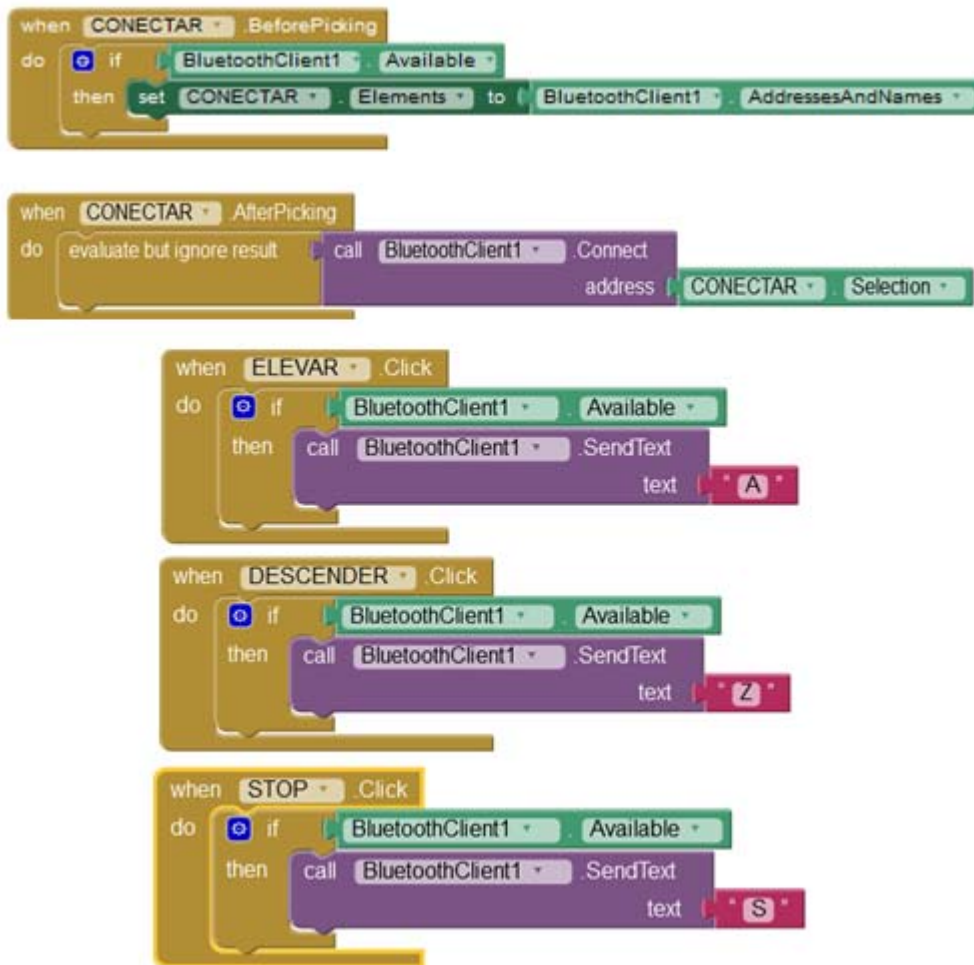


Figura 14 Programa control remoto del cuadricóptero.

3.12 CONTROLADOR PID

La familia de controladores PID, es una estructura de control mediante realimentación, que calcula la desviación entre un valor medido y un valor que se quiere obtener, haciendo una corrección en consecuencia. Este tipo de controladores han demostrado ser robustos y es por eso que son utilizados en más del 95% de los procesos industriales en lazo cerrado.

El algoritmo de control calcula tres parámetros diferentes: el proporcional, el integral y el derivativo. El Proporcional, es directamente proporcional al error actual, el Integral hace una corrección del error acumulado en el tiempo (integral del error) y el Derivativo determina la reacción del tiempo en el que el error se produce.

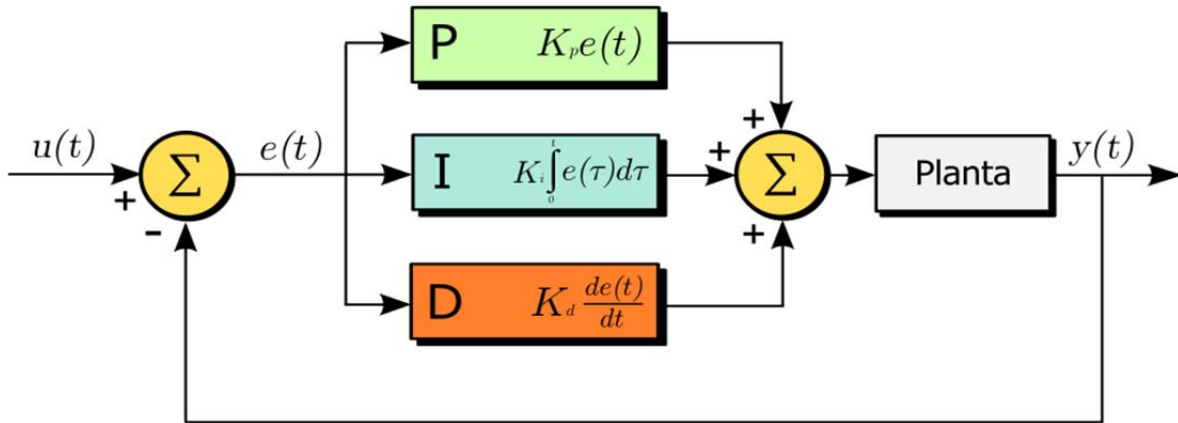


Figura 3.15 esquema PID ^[38]

Las variables que nos interesan son:

- PV (process variable o process value) Valor medido.
- SP (setpoint). Valor deseado u(t)
- MV (manipulated variable). Valor de entrada al sistema.
- E (Error). Diferencia entre valor medido (PV) y el deseado (SP)

3.13 DESARROLLO

Para el desarrollo del cuadricóptero se inició con una interfaz gráfica para simular el control de estabilidad utilizando el MPU 6050 y como referencia un pequeño avión simulado.

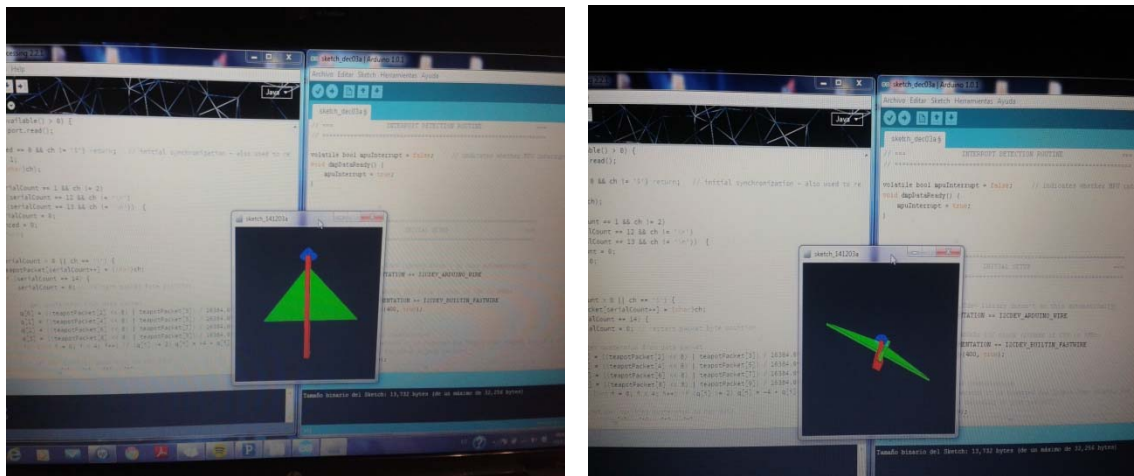


Figura 3.16 simulación estabilidad 1

[38]http://es.wikipedia.org/wiki/Proporcional_integral_derivativo#mediaviewer/File:PID.svg

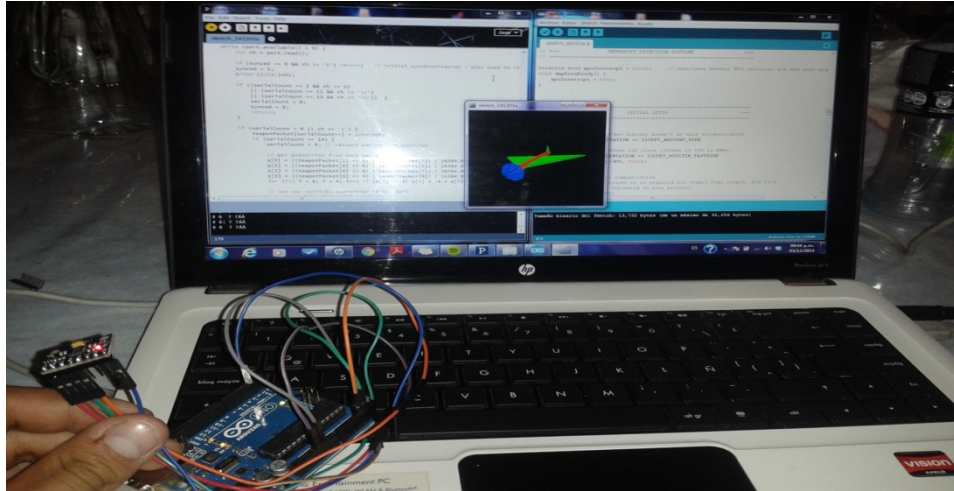


Figura 3.17 simulación estabilidad 2

Después de haber realizado lo que fue las pruebas con el avión, se iniciaron las pruebas con los motores.

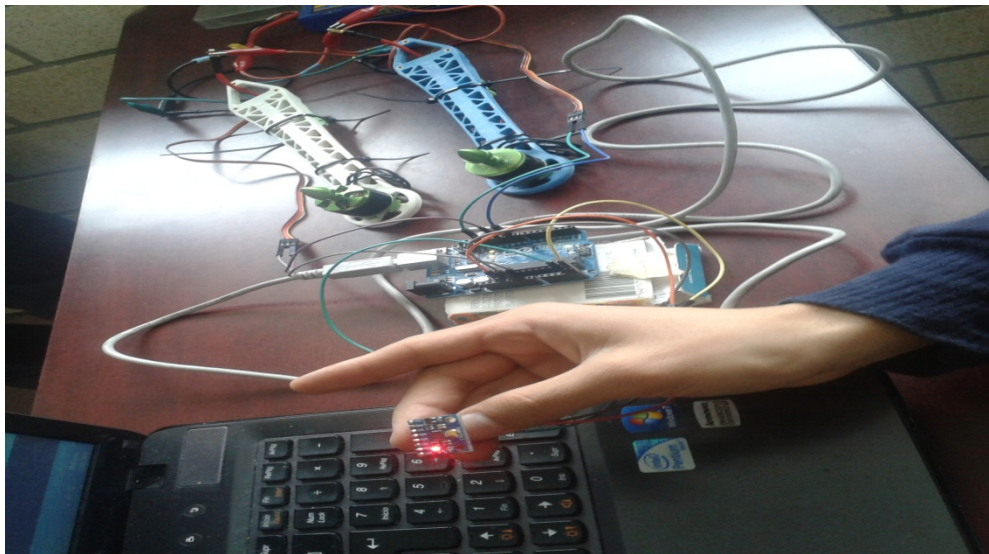


Figura 3.18 prueba estabilidad motores

Continuando con el desarrollo se realizaron pruebas con un balancín para poder lograr dar estabilidad a 2 motores.

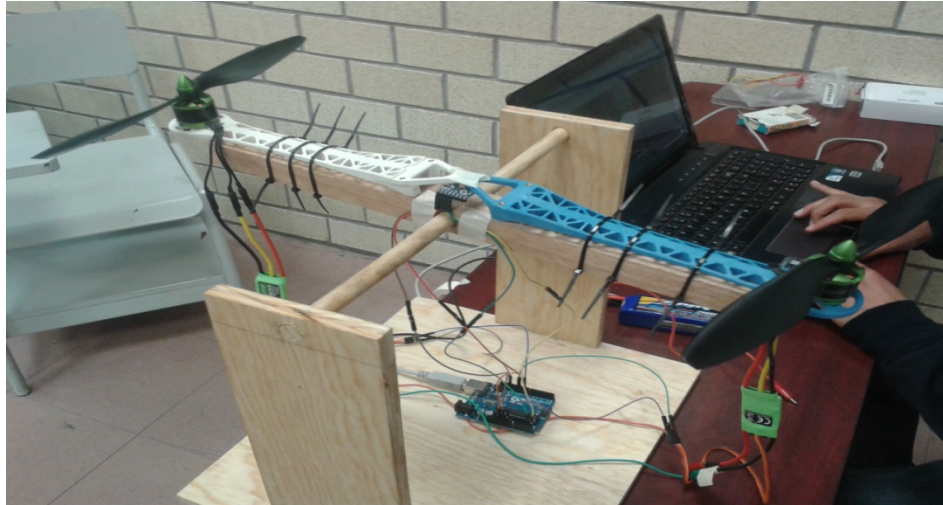


Figura 3.19 prueba estabilidad motores con balancín

Para continuar con las pruebas armamos el esqueleto de nuestro cuadricóptero.



Figura 3.20 armado 1

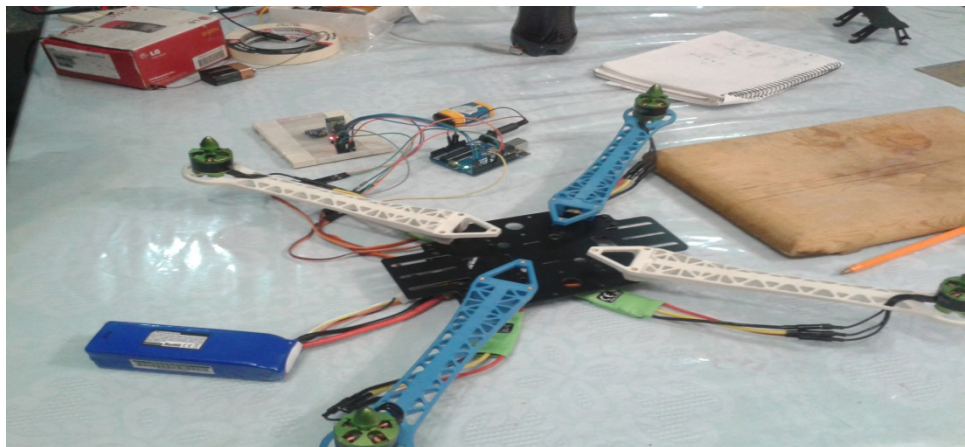


Figura 3.21 armado 2

Después de armar el cuadricóptero se procede a conectar los ESC y a soldarlos a la placa de alimentación.

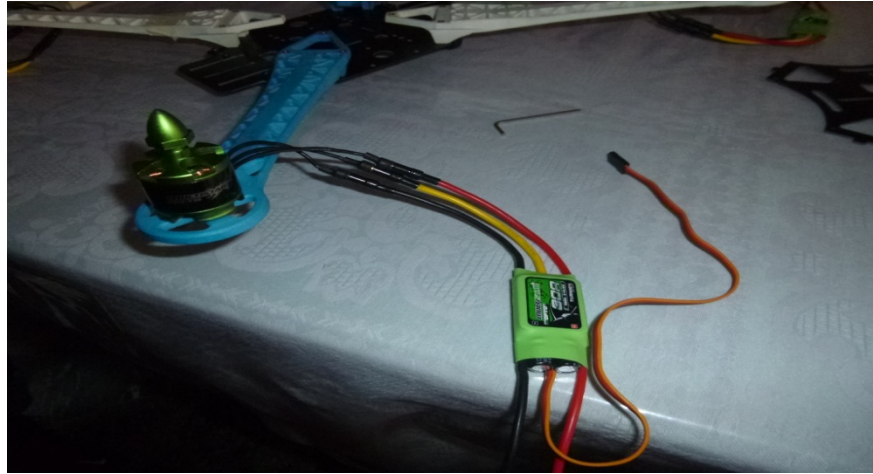


Figura 3.22 conexión ESC

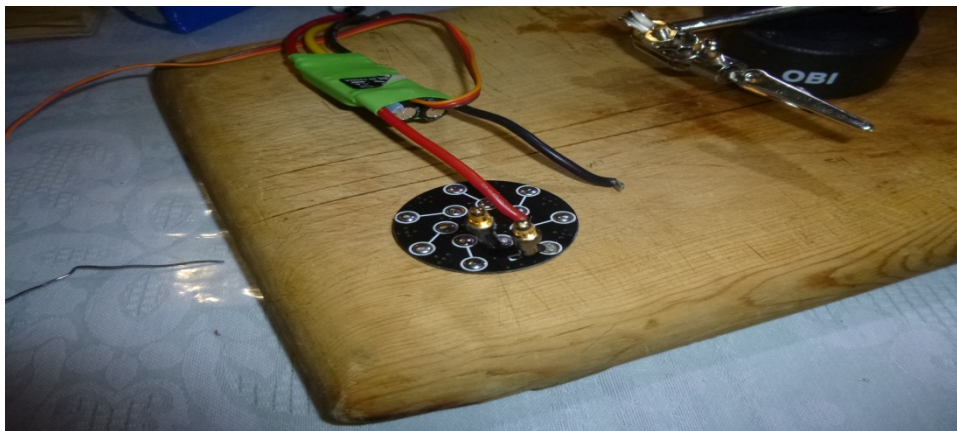


Figura 3.23 ESC con placa de alimentación

Seguido se termina de armar el cuadricóptero y se realizaron pruebas con los motores sin hélices y con hélices.



Figura 3.24 termino de armado



Figura 3.25 prueba motor sin hélices



Figura 3.26 prueba motor con hélices

CAPITULO 4

VIABILIDAD DEL PROYECTO

En este capítulo trataremos con lo que fueron los gastos al desarrollar el proyecto, así como la viabilidad que tiene al desarrollarse en nuestro país, mejoras, mantenimiento y adaptaciones que se pueden implementar en un futuro para el beneficio del usuario.

4.1 PRECIOS DE MERCADO

Actualmente los cuadricópteros o drones son una innovación tecnológica que ha revolucionado el mundo entero, pero como tal, estos equipos no son de un bajo costo ya que no en cualquier lugar podemos conseguir refacciones o piezas para la fabricación de estos.

Hoy en día los países que tienen mayor apogeo en la fabricación de cuadricópteros o aeromodelismo son China y España; como tal China es un gran importador de piezas o refacciones para todo el mundo vía internet y una de las tiendas más famosas es “Hobby King” quien varía sus precios según la pieza, la cantidad y la marca sin dejar a un lado que si se hace un pedido desde México, se cobrara un impuesto aparte del precio del producto.

Hablando de la venta del cuadricóptero ya fabricado y funcionando, estamos hablando de alrededor de la mínima cotización que es \$500 mexicanos hasta la más alta cotización que varia alrededor de los \$20,000 mexicanos aproximadamente. Cabe mencionar que no todos estos productos cuentan con un servicio de reparación en caso de fallas ya que como lo hemos mencionado las piezas son de importación y su costo es algo elevado.

4.2 COSTOS DEL PROYECTO

Para nuestro proyecto realizamos un gasto económico un poco elevado ya que la mayoría de nuestro material fue enviado pues las tiendas mexicanas que se dedican a la venta de piezas para aeromodelismo dan precios no tan accesibles pues las piezas son de importación.

En las siguientes tablas mostraremos los gastos que se hicieron unitariamente e indirectamente al igual que en envió que hubo para la fabricación de nuestro proyecto dando un total de la utilidad invertida y un precio de venta en caso de venta del proyecto.

Tabla 4.1 Gastos unitarios

Nombre	Precio
Motor con propela	\$ 450
Controlado (ESC)	\$ 200
Pila	\$ 480
Cargador	\$ 350
Base cuadricóptero	\$ 700
Modulo Bluetooth	\$ 150
MPU (acelerómetro – giroscopio)	\$ 95
AT MEGA328	\$ 69
Pin header hembra juego	\$ 30
Cristal 16 MHZ	\$ 8
Capacitor 22pf	\$ 2
Push button	\$ 2
Resistencia 100 ohm	\$.40
Resistencia 10 k ohm	\$.40
Jack de alimentación	\$ 5
Diodo Cenar 3.3v	\$ 2
Diodo Cenar 5.1v	\$ 2
Placa fibra de vidrio	\$ 100
Batería 9v	\$ 15
Cinchos de plástico	\$ 25
Carrete soldadura	\$ 200
Pasta para soldar	\$10
Diodo Led	\$ 1
Zocalo para microcontrolador	\$ 4
Cámara	\$ 250
Cloruro férrico	\$ 20
Suma total	\$3170.8

La tabla 4.1 muestra los gastos unitarios de las piezas utilizadas y ahora mostramos una segunda tabla que es la 4.2 con todas las piezas adquiridas y su total en costo.

Tabla 4.2 Gasto por piezas

Nombre	# Piezas	Total \$
Motor con propela	4	\$ 1800
Controlado (ESC)	4	\$ 800
Pila	1	\$ 480
Cargador	1	\$ 350
Base cuadricóptero	1	\$ 700
Modulo Bluetooth	2	\$ 300
MPU (acelerómetro – giroscopio)	1	\$ 95
AT MEGA328	2	\$ 138
Pin header hembra juego	1	\$ 30
Cristal 16 MHZ	2	\$ 16
Capacitor 22pf	4	\$ 8
Push button	2	\$ 4
Resistencia 100 ohm	5	\$ 5
Resistencia 10 k ohm	5	\$ 5
Jack de alimentación	1	\$ 5
Diodo Zener 3.3v	1	\$ 2
Diodo Zener 5.1v	1	\$ 2
Placa fibra de vidrio	1	\$100
Batería 9v	1	\$ 15
Cinchos de plástico	1	\$ 25
Carrete soldadura	1	\$ 200
Pasta para soldar	1	\$10
Diodo Led	4	\$ 4
Zocalo para microcontrolador	1	\$ 4
Cámara	1	\$ 250
Cloruro férrico	1	\$ 20
Suma total	de precios	\$ 5368

La siguiente tabla mostrara solamente los gastos de envío pagados para las piezas de importación que adquirimos bajo pedidos.

Tabla 4.3 envíos de material

Nombre	# Piezas	Gasto envío
Motor con propela	4	\$ 200
Controlado (ESC)	4	\$ 200
Pila	1	\$ 50
Cargador	1	\$ 50
Base cuadricóptero	1	\$ 69
Suma total	de envío	\$ 369

Con la suma total del material que nos dio \$5368 y los envíos que nos dieron \$369 nos da la cantidad de **\$5737** que es el total de lo que gastamos.

Ahora vemos los gastos indirectos que se llevaron a cabo en la tabla 4.4.

Tabla 4.4 Gastos indirectos

Gastos indirecto
Tiempo de trabajo variable de 2 a 5 horas diarias
Consumo de agua para la elaboración de la placa
Consumo de luz al trabajar de noche y utilizar computadoras
Comidas como recesos durante el trabajo
Utilización de teléfono para comunicación entre compañeros de equipo
Uso de internet para búsqueda de información

Como tal no aparecen costos ya que en los gastos indirectos no hay una evaluación de lo que se gasto.

4.2.1 PRECIO DE VENTA

Podemos decir que en base a los gastos realizados damos un precio de venta aproximado de **\$5500** para este proyecto y hay que recordar el uso que tendrá y que hay materiales o componentes que se compraron de más por cualquier inconveniente, así como los beneficios de usar estos componentes y piezas para la elaboración de este proyecto de un cuadricóptero.

4.2.2 SERVICIO DE MANTENIMIENTO Y REFACCIONES

Como ya se había mencionado, los cuadricópteros no tienen como tal un servicio de mantenimiento y las refacciones son algo caras.

Sin embargo, para nuestro proyecto no será un problema ya que el equipo como tal cuenta con materiales de bajo costo si es que se requiere de algún tipo de refacción, así como un servicio de mantenimiento óptimo pues está muy bien detallado el funcionamiento y la estructura dándonos oportunidad de repararlo y hasta mejorarlo sin ningún inconveniente.

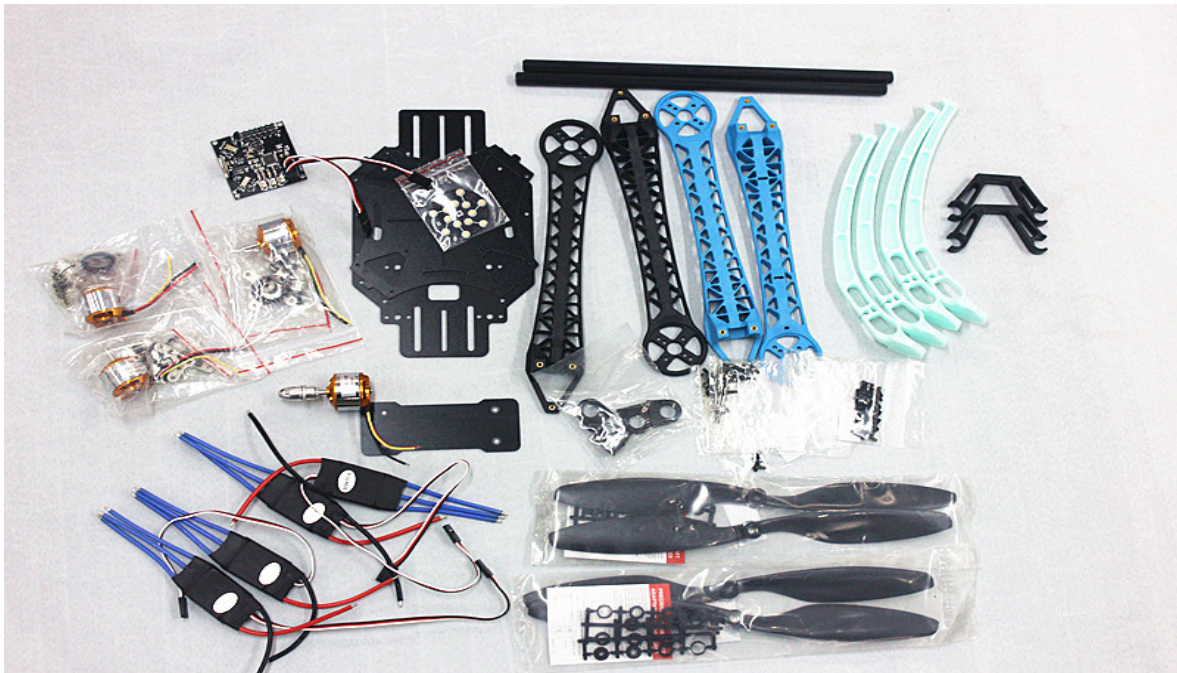


Figura 4.1 Equipo y refacciones de un cuadricóptero

4.3 ADAPTACIONES ESPECIALES A NECESIDADES DEL CLIENTE

Nuestro cuadricóptero es apto para poder tener algún tipo de cambio si es necesario ya que se le pueden adaptar equipos de video, sonido, transmisión e iluminación si el cliente así lo desea siempre y cuando cumpla con las normas de seguridad y no sobrepasen el peso que este tiene para cargar y así poder desplazarse libremente y cumpla su función.



Figura 4.2 cuadricóptero adaptado

4.4 MEJORAS Y PROYECTOS FUTUROS

4.4.1 MEJORAS

Existen muchas mejoras para este tipo de cuadricóptero pero habría que analizar una por una es por eso que solo daremos tres ejemplos.

Una de las mejoras que se le pueden hacer a este proyecto es el cambio de la cámara ya que nosotros utilizamos por cuestiones de presupuesto una que es pequeña pero que se ajusta a las necesidades del cliente, sin embargo este cuadricóptero es capaz de soportar una cámara como lo son las Go Pro pues estas cámaras son de alta definición y una mejor calidad, pero cabe mencionar que el costo de estas es elevado y rondan alrededor de los \$3000.



Figura 4.3 Cámaras Go Pro

Nota: como tal este proyecto es para búsqueda de personas extraviadas y lo idóneo es una transmisión en vivo, pero dada la situación que este equipo tiene una alta potencia la transmisión puede fallar, es por eso que se utilizan cámaras que grabaran y por medio de su micro sd podremos ver el video sin necesidad de perder la señal.

Otra mejora es la adaptación de micrófonos con los cuales nosotros podremos escuchar con más claridad los sonidos que se encuentren en el entorno de nuestro cuadricóptero, sin dejar a un lado que la cámara posee la habilidad de grabar sonido mientras graba video pero con una serie de micrófonos se escuchara mejor.

Y la ultima mejora seria el cambio del modulo de bluetooth por un modulo Xbee el cual posee un mejor alcance de señal para poder controlar nuestro cuadricóptero a una mayor distancia y sin ningún inconveniente, pero así como la cámara Go Pro es cara, el modulo Xbee también es un poco caro a comparación del modulo de bluetooth ya que este ronda los \$150 y el Xbee hasta los \$1000 aproximadamente dependiendo el alcance que se requiera.



Figura 4.4 Modulo Xbee

4.4.2 PROYECTOS FUTUROS

Como parte de un proyecto futuro nosotros vemos que nuestro cuadricóptero no solo pueda buscar personas si no que de algún modo pueda llevar algún tipo de ayuda ya sean medicamentos, lámparas, agua, alimentos, etc.

También que se busque otro medio por el cual nuestro cuadricóptero pueda ayudar en la exploración de lugares que no haya alguna forma de alcance por parte del humano como puede ser un deslave, un incendio, una inundación entre otros acontecimientos.



Figura 4.5 Cuadrícóptero a futuro

Una última mejora será el control del PID para poder tener una excelente estabilidad de nuestro cuadrícóptero ya que este es muy importante para su buen funcionamiento y manejo.

CONCLUSIONES DE TESIS

Este proyecto de alguna manera es muy ambicioso, pero de igual forma es algo complicado por el desarrollo que se tiene sin embargo, pudimos demostrar que al construir este proyecto podemos alcanzar cualquier objetivo propuesto.

Las pruebas realizadas nos dieron resultados que esperábamos pero también algunos que no para nuestro control; igual la comunicación que realizamos vía bluetooth nos da más ideas para mejorarla con otros dispositivos y así en un futuro poder tener un mayor alcance.

El uso del microcontrolador fue de gran ayuda y gracias a una fácil programación pudimos resolver los problemas que se fueron sucintados a lo largo del desarrollo de la tesis.

El uso de una aplicación por el sistema de ANDROID fue de gran apoyo ya que no se hizo un gran gasto y es innovador para la venta comercial.

Una cosa que debemos dejar bien en claro es que el uso de un cuadricóptero no es tan sencillo e incluso puede ser peligroso por la potencia que manejan los rotores, el nuestro al principio tuvo fallas de estabilidad y se realizaron pruebas con mucha precaución, de igual forma al volarlo se tienen que tomar consideraciones para no tener accidentes y poder dañar el cuadricóptero.

BIBLIOGRAFIA

[1] Monitoreo y control en tiempo real de un vehículo aéreo no tripulado, "Adriana Pérez Pedraza, Miguel Alejandro Ruiz de los Santos"

[2]<http://eleconomista.com.mx/seguridad-publica/2011/03/16/que-son-drones-o-aviones-no-tripulados>

[3] Desarrollo de un cuadricóptero operado por ROS, Iván Monzón, Proyecto de fin de Carrera

[4] R. MAYORGA. "Sistema de navegación para vehículos Aéreos Cuadricópteros", Proyecto de titulación en Ingeniería Técnica Aeronáutica, Universidad Politécnica Superior de Cataluña. España. 2009.

[5] "Unmanned Aerial Vehicles News", <http://defenseindustrydialy.com>

[6] <http://vtol.org/awards/HPHCBooklet.pdf>

[7] <http://www.youtube.com/watch?v=q70tM5sDQhc>

[8] "Blog acerca de UAVs micro" <http://www.flightglobal.com>

[9] Nadales C., "Control de un Quadrotor mediante la plataforma Arduino", Proyecto de titulación en Ingeniería Técnica de Telecomunicaciones, Universidad Politécnica Superior de Cataluña. España. 2009.

[10] [sinvestavtesisdequadrotor.PDF](#)

[11] <http://www.autogiroescuela.com>

[12] <http://www.escriint.cl/php/contenido/UAV%20CORREGIDO.pdf>

[13] <http://forum.nationstates.net>

[14] <http://moller.com>

[15] <http://iapss.scenebeta.com/noticia/ar-drone-el-cuadrocoptero-teledirigido-por-tuiphone-touch-ipad>

[16] A. Barrientos, J Del Cerro, R. San Martín, C. Rossi, "Vehículos aéreos no tripulados para uso civil", Proyecto Titulación, Universidad Politécnica de Madrid. España.2012

[17] <http://techmez.com>

[18] <http://www.e-radiocontrol.com.ar>

[19] <http://www.asctec.de/uav-applications/research/products/asctec-hummingbird>

[20] Brito J., Vissierre D., "Design of a Quadrotor Prototype", proyecto de titulación. Universidad Técnica de Lisboa, Portugal. 2009

[21] <http://code.google.com/p/arducopter/wiki/ArduCopter>

[22] [Planificacion_de_la_busqueda_de_personas_desaparecidas.pdf](#)

[23] <http://www.perrosdebusqueda.com>

- [24] defensacivilmaimon.wordpress.com
- [25] Planificacion_de_la_busqueda_de_personas_desaparecidas.pdf
- [26] <http://electronica36.blogspot.mx>
- [27] Control de un Quadrotor mediante la plataforma Arduino. Tesis de Ingeniería de Telecomunicaciones, especialidad Sistemas de Telecomunicaciones.
- [28] <http://en.wikipedia.org/wiki/ATmega328>
- [29] <http://www.geekfactory.mx/radio/bluetooth-hc-05-y-hc-06-tutorial-de-configuracion/>
- [30] <http://www.aerodelismofacil.com/Motores%20brushless.htm>
- [31] <http://www.mabuchi-motor.co.jp/>
- [32] <http://circuitronico.blogspot.mx/2012/03/baterias-li-po-y-esc.html>
- [33] <http://rc.lapipadelindio.com/general/baterias-automodelismo-radiocontrol>
- [34] <http://flightpowerbatteries.com/>
- [35] <http://electronica36.blogspot.mx/2012/05/informe-completo-sobre-como-hacer-y.html>
- [36] <http://www.instructables.com/id/Accelerometer-Gyro-Tutorial/>
- [37] http://diebotreise.blogspot.mx/2012/11/esc-electronic-speed-controller.html?utm_source=BP_recent
- [38] http://es.wikipedia.org/wiki/Proporcional_integral_derivativo#mediaviewer/File:PID.svg

GLOSARIO

ACELERÓMETRO: Instrumento para la medida de la aceleración o la deceleración.

AEROPLANO: Aeronave que, siendo más pesada que el aire, se sostiene en él debido a la presión del viento sobre los planos que la componen y la velocidad. Sus dos elementos principales son la fuerza motora que lo impulsa en la dirección del vuelo y el plano de sustentación.

AERODINÁMICA: Rama de la mecánica de fluidos que estudia las acciones que aparecen sobre los cuerpos sólidos cuando existe un movimiento relativo entre éstos y el fluido que los baña, siendo éste último un gas.

AERONAVE: Vehículo capaz de desplazarse por el aire o por el espacio.

ALERON: Cada una de las partes articuladas situadas en el margen posterior de las alas y del estabilizador horizontal de cola de un avión con que se regula el equilibrio transversal del mismo.

ALGORITMO: En aritmética y programación informática, conjunto ordenado de instrucciones sistemáticas que permite hallar la solución de un problema específico.

ALTITUD: Distancia vertical que separa un punto respecto de otro que le sirve de referencia, generalmente el nivel del mar

AMPERE: Unidad de intensidad de la corriente eléctrica del Sistema Internacional, de símbolo A.

ANDROID: Sistema operativo basado en el núcleo Linux. Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tablets.

AUTOGIRO: Aeronave de ala giratoria, es decir, vuela como los aviones pero su ala es un rotor que gira por la acción del viento relativo que lo atraviesa de abajo a arriba.

BARÓMETRO: Instrumento para determinar la presión atmosférica.

BAUDIO: Es una unidad de medida utilizada en telecomunicaciones, que representa el número de símbolos por segundo en un medio de transmisión digital

BRUSHLESS: Se refiere a motores sin escobillas.

BLUETOOTH: especificación industrial para Redes Inalámbricas de Área Personal que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2,4GHz.

BUFFER: Memoria de almacenamiento temporal de información que permite transferir los datos entre unidades funcionales con características de transferencia diferentes.

CELULAS LIPO: Celda de polímero de litio.

CPU: Sigla de la expresión inglesa central processing unit, 'unidad central de proceso', que es la parte de una computadora en la que se encuentran los elementos que sirven para procesar datos.

DRIVER: Controlador, rutina o programa que enlaza un dispositivo periférico al sistema operativo.

DRONES: Vehículos aéreos no tripulados o robots voladores.

ESC: Electronic speed control. Control electrónico de Velocidad.

FRECUANCIA: Número de veces que aparece, sucede o se realiza una cosa durante un período o un espacio determinados.

GIROSCOPIO: Dispositivo mecánico que sirve para medir, mantener o cambiar la orientación en el espacio de algún aparato o vehículo.

HANDHELD: Anglicismo que significa en castellano "palmar" y describe en este caso cuadricóptero portátil que se puede llevar en una mano.

IMU: Unidad de medición inercial o IMU (inertial measurement unit), dispositivo capaz de medir la fuerza (aceleración) y la velocidad. Genéricamente consta de un Acelerómetro y un Giroscopio.

ISM (Industrial, Scientific and Medical) son bandas reservadas internacionalmente para uso no comercial de radiofrecuencia electromagnética en áreas industrial, científica y médica.

ISP: Proveedor de servicios de Internet.

MIT App Inventor 2: Herramienta de programación basada en los bloques que permite a todos iniciar la programación y construir aplicaciones totalmente funcionales para los dispositivos Android.

JOULE: Medida que sirve para contar la cantidad de energía que se utiliza y es representado con la letra J. Esto significa que si se consume 1 Joule en un segundo, se estaría consumiendo 1 watt de potencia eléctrica.

MAGNETÓMETRO: Dispositivo que sirve para cuantificar en fuerza o dirección la señal magnética de una muestra.

MICROCONTROLADOR: Circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica.

MICROPROCESADOR: O procesador, es el circuito integrado central y más complejo de un sistema informático; a modo de ilustración, se le suele llamar por analogía el «cerebro» de un computador.

MIPS: Abreviación de las palabras "M"illones de "I"nstrucciones "P"or "S"egundo".

MPU: Pick up magnetic, Sensor de proximidad magnético.

OLEODUCTO: Tubería que sirve para llevar petróleo de un lugar a otro.

PERIODO: Espacio de tiempo durante el cual se realiza una acción o se desarrolla un acontecimiento.

PIC: Familia de microcontroladores tipo RISC fabricados por Microchip Technology Inc.

PID: Proporcional integral derivativo, mecanismo de control por realimentación que calcula la desviación o error entre un valor medido y el valor que se quiere obtener, para aplicar una acción correctora que ajuste el proceso

PITCH: Angulo de navegación (Cabeceo) es una inclinación del morro del avión, o rotación respecto al eje ala-ala.

PWM: Modulación por ancho de pulsos (PWM, siglas en inglés de pulse-width modulation) de una señal o fuente de energía, es una técnica en la que se modifica el ciclo de trabajo de una señal periódica.

RAM: La memoria de acceso aleatorio (Random-Access Memory, RAM) se utiliza como memoria de trabajo de computadoras para el sistema operativo, los programas y la mayor parte del software.

ROLL: Angulo de navegación (Guiñada) rotación intrínseca alrededor del eje vertical perpendicular al avión.

SENSOR: Dispositivo que capta variaciones de luz, temperatura o sonido a corta y larga distancia y sirve para activar un mecanismo.

SIMULINK: Herramienta de simulación de modelos o sistemas, con cierto grado de abstracción de los fenómenos físicos involucrados en los mismos. Se hace hincapié en el análisis de sucesos, a través de la concepción de sistemas (cajas negras que realizan alguna operación).

SRAM: Son las siglas de la voz inglesa Static Random Access Memory, que significa memoria estática de acceso aleatorio

TOOLKIT: Conjunto exhaustivo de herramientas y orientación, como partner de Microsoft, puede usar sin costo alguno.

TORQUE: Fuerza que produce rotación.

TQFP: Tipo encapsulado cuadrado plano o Quad Flat Package

UART: "Universal Asynchronous Receiver-Transmitter. Es un dispositivo que sirve para transmitir o recibir datos secuenciales de manera asíncrona.

UAV: Unmanned Aircraft, avión no tripulado.

USART: "Universal Synchronous/Asynchronous Receiver-Transmitter. Es un dispositivo que sirve para transmitir o recibir datos secuenciales de manera síncrona o asíncrona.

USB: Sigla del inglés universal serial bus, periférico que permite conectar diferentes periféricos a una computadora.

VELOCIDAD: Magnitud vectorial física que relaciona el desplazamiento que realiza un móvil entre dos posiciones con el tiempo que tarda en desplazarse.

VOLT: Unidad de tensión eléctrica, potencial eléctrico y fuerza electromotriz del Sistema Internacional, de símbolo V.

WATT: Unidad de potencia del Sistema Internacional, de símbolo W, que equivale a la potencia capaz de conseguir una producción de energía igual a 1 julio por segundo.

WLAN red de área local inalámbrica, también conocida como WLAN (del inglés wireless local area network).

WPAN: Wireless Personal Área Network (WPAN), Red Inalámbrica de Área Personal o red de área personal inalámbrica

YAW: Angulo de navegación (Alabeo) rotación respecto de un eje morro-cola del avión.

500 mW DO-35 Glass
Zener Voltage Regulator Diodes
GENERAL DATA APPLICABLE TO ALL SERIES IN
THIS GROUP

500 Milliwatt
Hermetically Sealed
Glass Silicon Zener Diodes

GENERAL
DATA
500 mW
DO-35 GLASS

GLASS ZENER DIODES
500 MILLIWATTS
1.8–200 VOLTS

Specification Features:

- Complete Voltage Range — 1.8 to 200 Volts
- DO-204AH Package — Smaller than Conventional DO-204AA Package
- Double Slug Type Construction
- Metallurgically Bonded Construction

Mechanical Characteristics:

CASE: Double slug type, hermetically sealed glass

MAXIMUM LEAD TEMPERATURE FOR SOLDERING PURPOSES: 230°C, 1/16" from case for 10 seconds

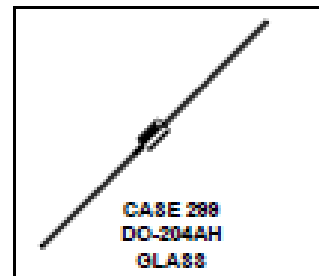
FINISH: All external surfaces are corrosion resistant with readily solderable leads

POLARITY: Cathode indicated by color band. When operated in zener mode, cathode will be positive with respect to anode

MOUNTING POSITION: Any

WAFER FAB LOCATION: Phoenix, Arizona

ASSEMBLY/TEST LOCATION: Seoul, Korea



MAXIMUM RATINGS (Motorola Devices)*

Rating	Symbol	Value	Unit
DC Power Dissipation and $T_L < 75^\circ\text{C}$ Lead Length = 3/8" Derate above $T_L = 75^\circ\text{C}$	P_D	500 4	mW mW/°C
Operating and Storage Temperature Range	$T_{L, T_{stg}}$	-65 to +200	°C

* Some part number series have lower JEDEC registered ratings.

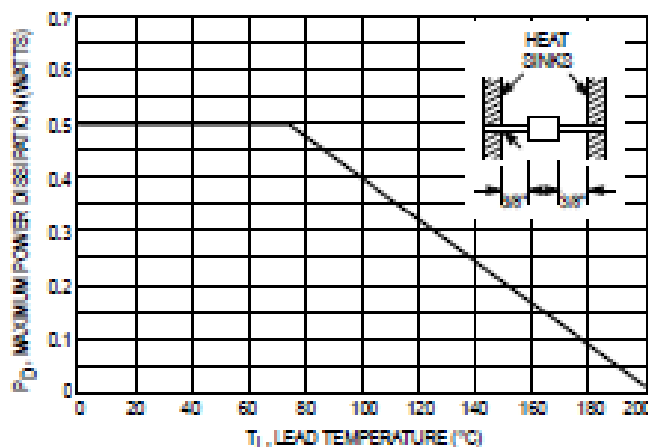


Figure 1. Steady State Power Derating

GENERAL DATA — 500 mW DO-35 GLASS

APPLICATION NOTE — ZENER VOLTAGE

Since the actual voltage available from a given zener diode is temperature dependent, it is necessary to determine junction temperature under any set of operating conditions in order to calculate its value. The following procedure is recommended:

Lead Temperature, T_L , should be determined from:

$$T_L = \theta_{LA} P_D + T_A$$

θ_{LA} is the lead-to-ambient thermal resistance ($^{\circ}\text{C}/\text{W}$) and P_D is the power dissipation. The value for θ_{LA} will vary and depends on the device mounting method. θ_{LA} is generally 30 to $40^{\circ}\text{C}/\text{W}$ for the various clips and tie points in common use and for printed circuit board wiring.

The temperature of the lead can also be measured using a thermocouple placed on the lead as close as possible to the tie point. The thermal mass connected to the tie point is normally large enough so that it will not significantly respond to heat surges generated in the diode as a result of pulsed operation once steady-state conditions are achieved. Using the measured value of T_L , the junction temperature may be determined by:

$$T_J = T_L + \Delta T_{JL}$$

ΔT_{JL} is the increase in junction temperature above the lead temperature and may be found from Figure 2 for dc power:

$$\Delta T_{JL} = \theta_{JL} P_D$$

For worst-case design, using expected limits of I_Z , limits of P_D and the extremes of T_J (ΔT_{JL}) may be estimated. Changes in voltage, V_Z , can then be found from:

$$\Delta V = \theta_{VZ} T_J$$

θ_{VZ} , the zener voltage temperature coefficient, is found from Figures 4 and 5.

Under high power-pulse operation, the zener voltage will vary with time and may also be affected significantly by the zener resistance. For best regulation, keep current excursions as low as possible.

Surge limitations are given in Figure 7. They are lower than would be expected by considering only junction temperature, as current crowding effects cause temperatures to be extremely high in small spots, resulting in device degradation should the limits of Figure 7 be exceeded.

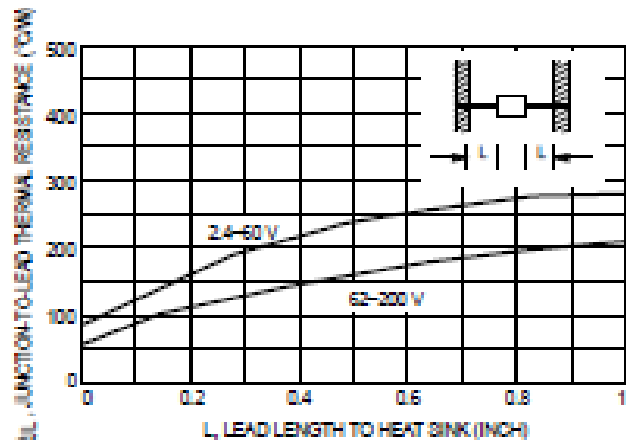


Figure 2. Typical Thermal Resistance

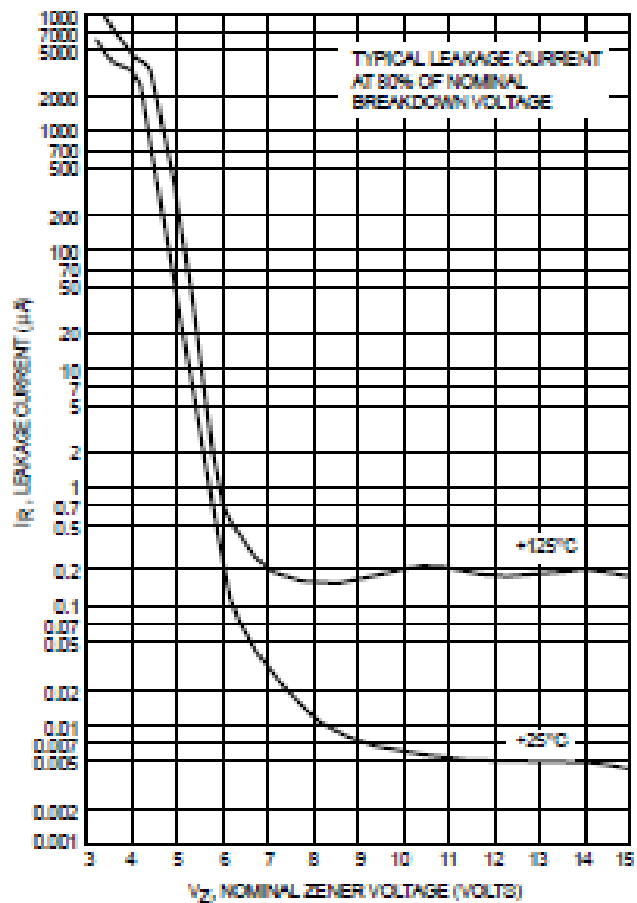
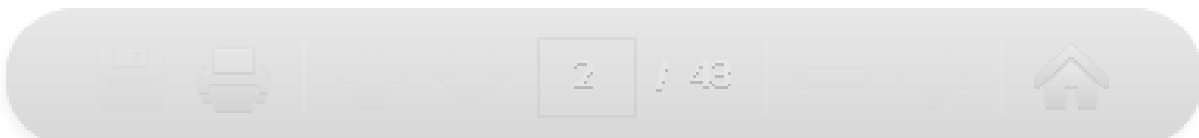


Figure 3. Typical Leakage Current



GENERAL DATA — 500 mW DO-35 GLASS

TEMPERATURE COEFFICIENTS

(-55°C to +150°C temperature range; 90% of the units are in the ranges indicated.)

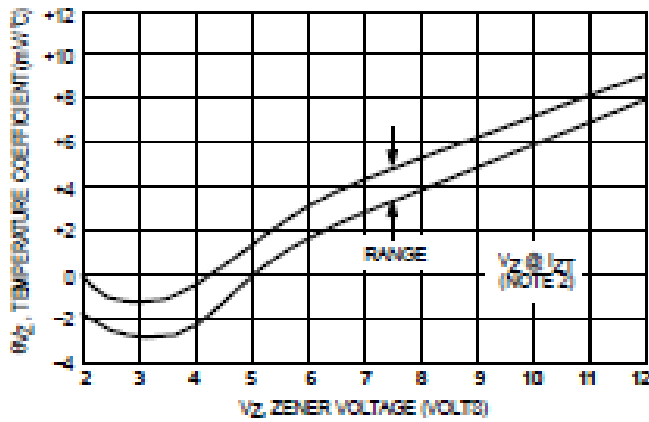


Figure 4a. Range for Units to 12 Volts

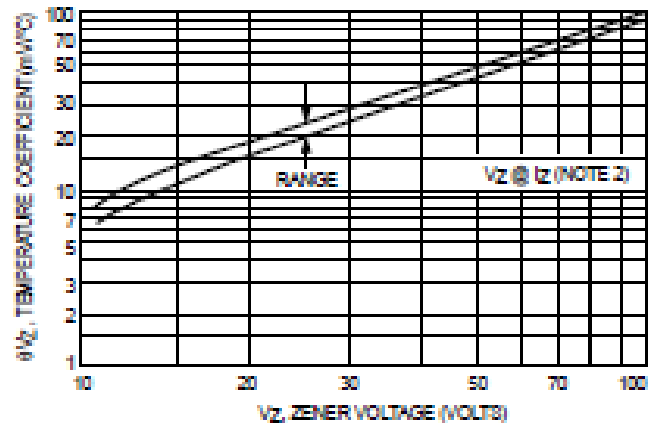


Figure 4b. Range for Units 12 to 100 Volts

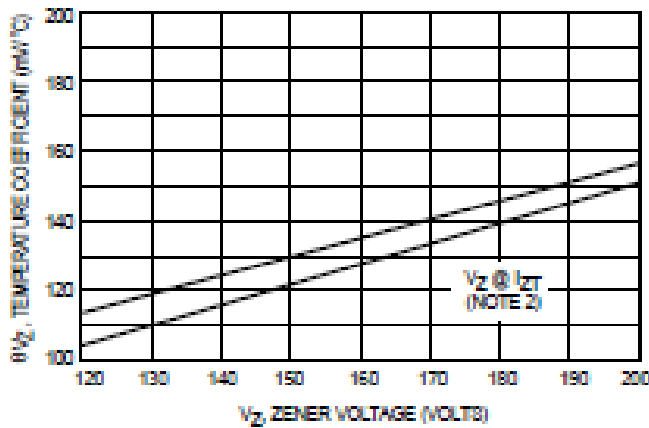


Figure 4c. Range for Units 120 to 200 Volts

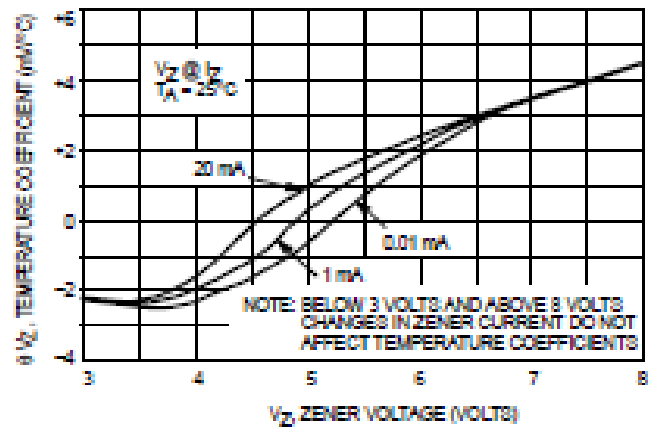


Figure 5. Effect of Zener Current

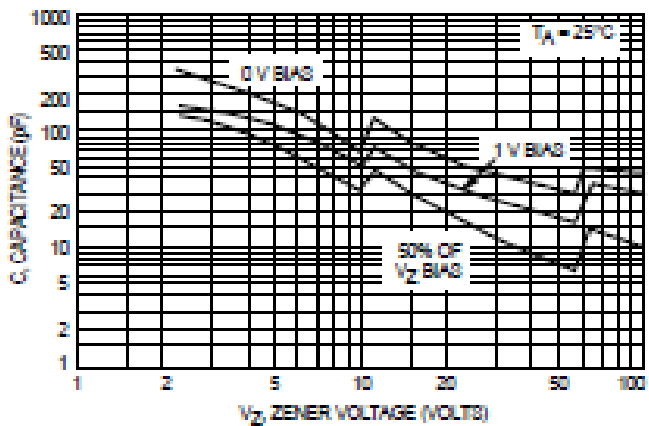


Figure 6a. Typical Capacitance 2.4-100 Volts

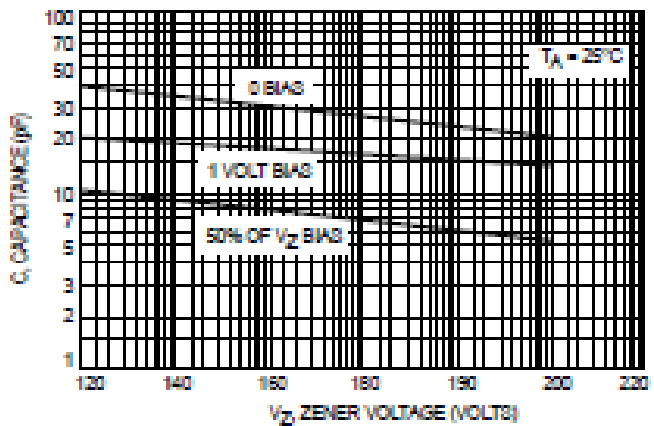


Figure 6b. Typical Capacitance 120-200 Volts

GENERAL DATA — 500 mW DO-35 GLASS

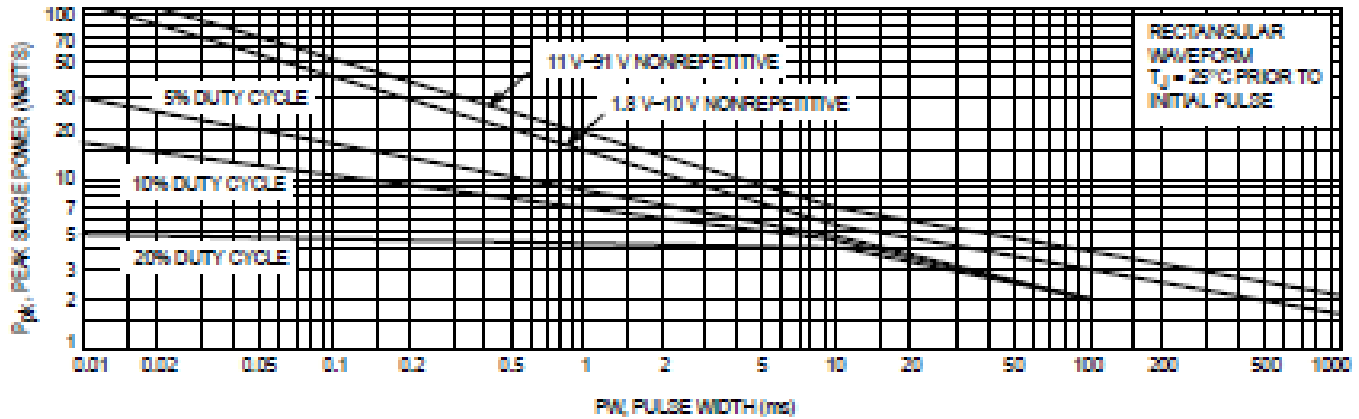


Figure 7a. Maximum Surge Power 1.8–31 Volts

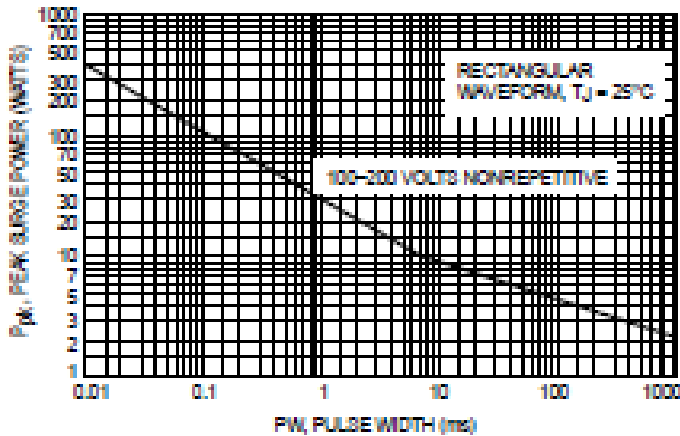


Figure 7b. Maximum Surge Power DO-204AH
100–200 Volts

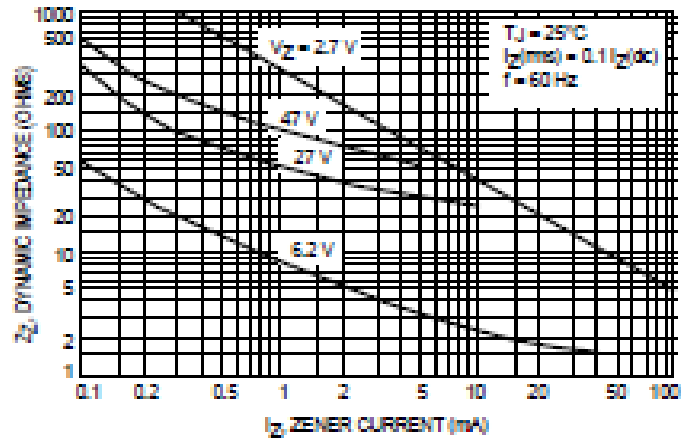


Figure 8. Effect of Zener Current on
Zener Impedance

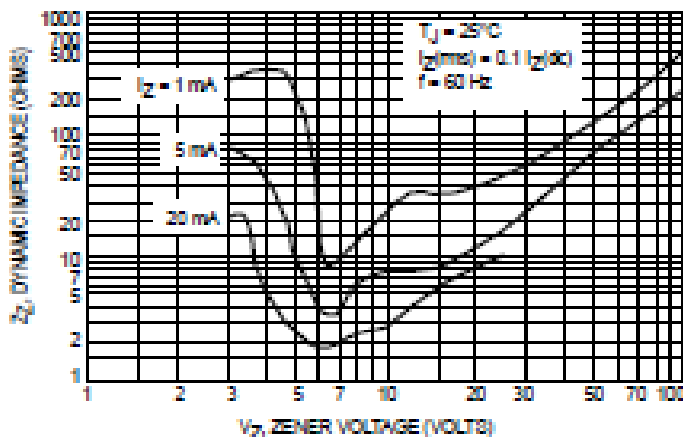


Figure 9. Effect of Zener Voltage on Zener Impedance

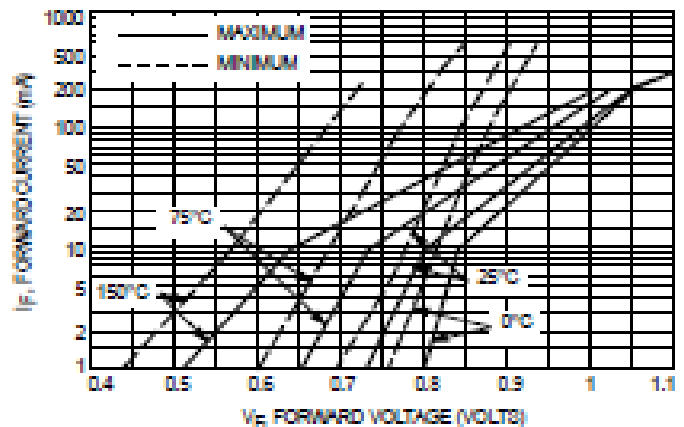


Figure 10. Typical Forward Characteristics

GENERAL DATA — 500 mW DO-35 GLASS

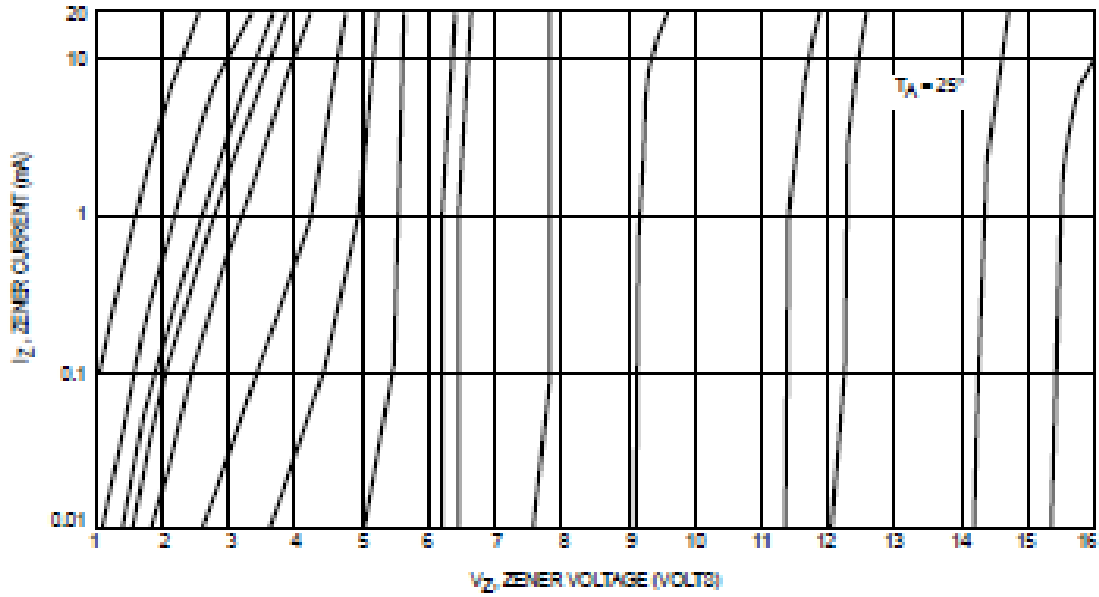


Figure 11. Zener Voltage versus Zener Current — $V_Z = 1$ thru 16 Volts

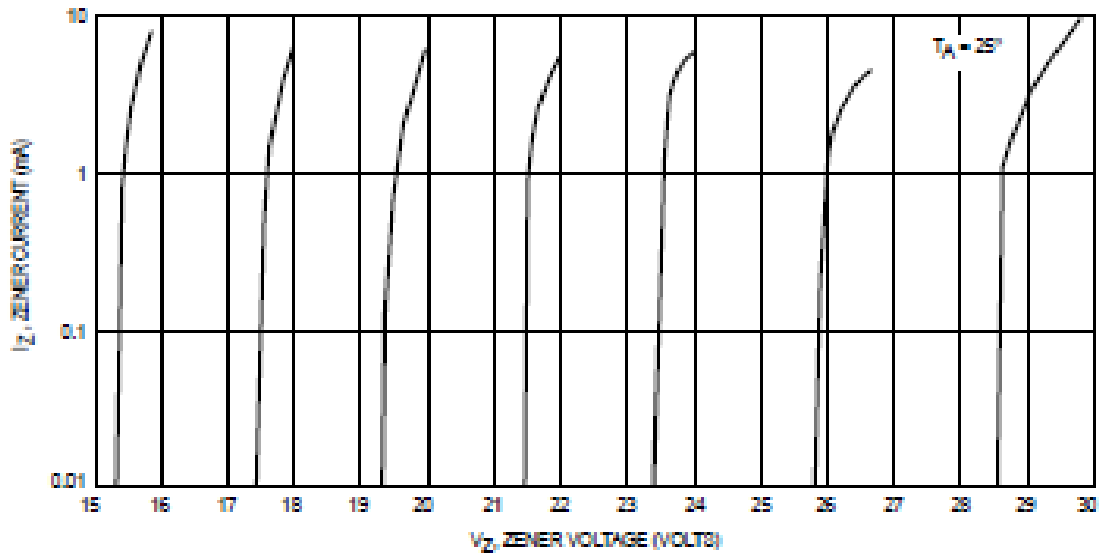


Figure 12. Zener Voltage versus Zener Current — $V_Z = 15$ thru 30 Volts

GENERAL DATA — 500 mW DO-35 GLASS

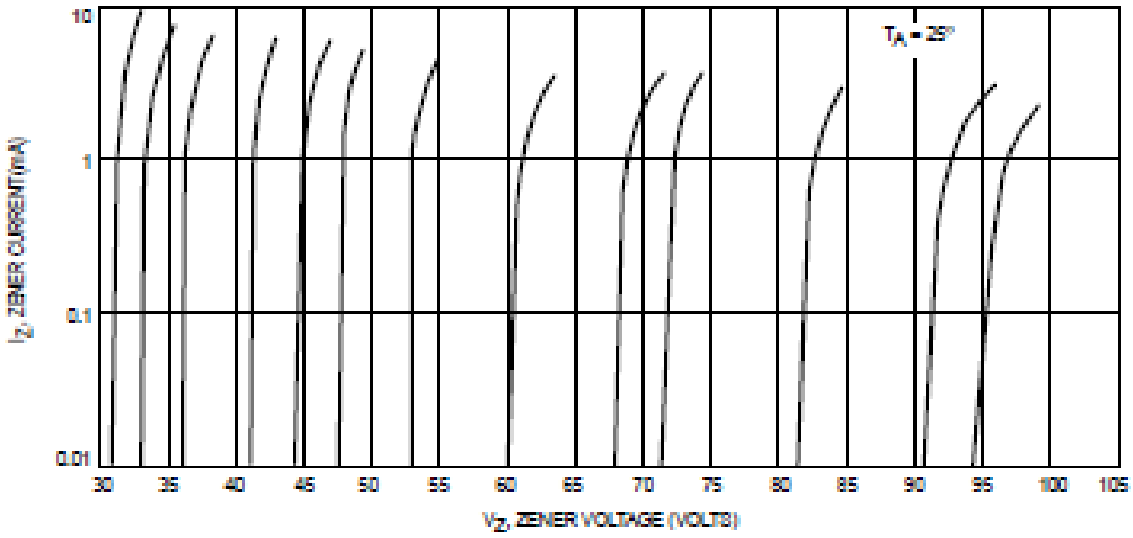


Figure 13. Zener Voltage versus Zener Current — $V_Z = 30$ thru 105 Volts

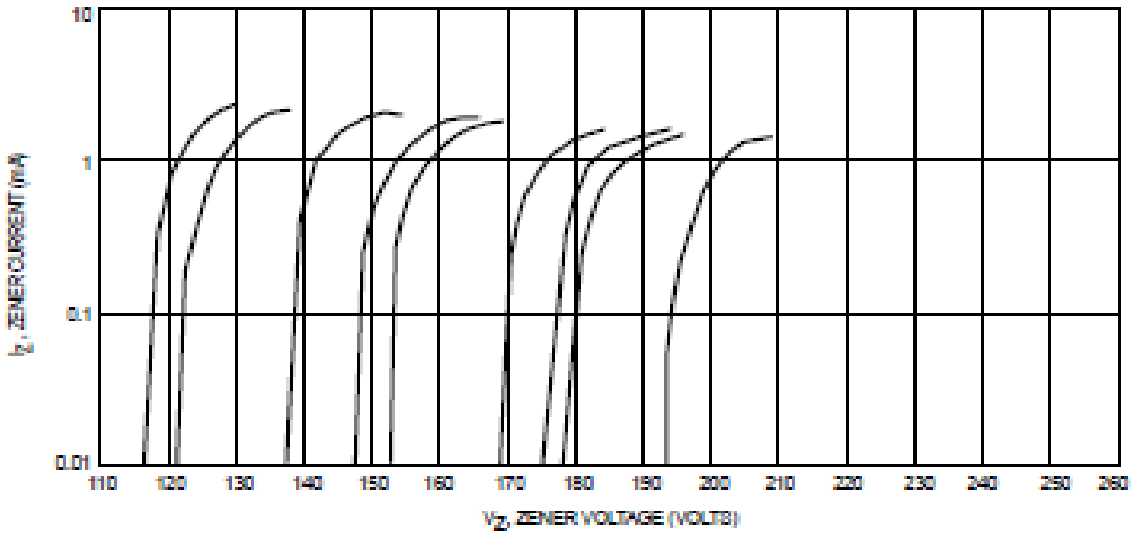


Figure 14. Zener Voltage versus Zener Current — $V_Z = 110$ thru 220 Volts

GENERAL DATA — 500 mW DO-35 GLASS

ELECTRICAL CHARACTERISTICS (T_A = 25°C, V_F = 1.5 V Max at 200 mA for all types)

Type Number (Note 1)	Nominal Zener Voltage V _Z @ I _{ZT} (Note 2) Volts	Test Current I _{ZT} mA	Maximum Zener Impedance Z _{ZT} @ I _{ZT} (Note 3) Ohms	Maximum DC Zener Current I _{ZM} (Note 4) mA	Maximum Reverse Leakage Current	
					T _A = 25°C I _R @ V _R = 1 V μA	T _A = 160°C I _R @ V _R = 1 V μA
1N4370A	2.4	20	30	150	100	200
1N4371A	2.7	20	30	135	75	150
1N4372A	3	20	29	120	50	100
1N748A	3.3	20	28	110	10	30
1N747A	3.6	20	24	100	10	30
1N748A	3.9	20	23	95	10	30
1N749A	4.3	20	22	85	2	30
1N750A	4.7	20	19	75	2	30
1N751A	5.1	20	17	70	1	20
1N752A	5.6	20	11	66	1	20
1N753A	6.2	20	7	60	0.1	20
1N754A	6.8	20	5	55	0.1	20
1N755A	7.5	20	6	50	0.1	20
1N756A	8.2	20	8	45	0.1	20
1N757A	9.1	20	10	40	0.1	20
1N758A	10	20	17	35	0.1	20
1N759A	12	20	30	30	0.1	20

Type Number (Note 1)	Nominal Zener Voltage V _Z (Note 2) Volts	Test Current I _{ZT} mA	Maximum Zener Impedance (Note 3)			Maximum DC Zener Current I _{ZM} (Note 4) mA	Maximum Reverse Current	
			Z _{ZT} @ I _{ZT} Ohms	Z _{ZK} @ I _{ZK} Ohms	I _{ZK} mA		I _R Maximum μA	Test Voltage V _{do} V _R
1N957B	6.8	18.5	4.5	700	1	47	150	5.2
1N958B	7.5	16.5	5.5	700	0.5	42	75	5.7
1N959B	8.2	15	6.5	700	0.5	38	50	6.2
1N960B	9.1	14	7.5	700	0.5	35	25	6.9
1N961B	10	12.5	8.5	700	0.25	32	10	7.6
1N962B	11	11.5	9.5	700	0.25	28	5	8.4
1N963B	12	10.5	11.5	700	0.25	26	5	9.1
1N964B	13	9.5	13	700	0.25	24	5	9.9
1N965B	15	8.5	16	700	0.25	21	5	11.4
1N966B	16	7.8	17	700	0.25	19	5	12.2
1N967B	18	7	21	750	0.25	17	5	13.7
1N968B	20	6.2	25	750	0.25	15	5	15.2
1N969B	22	5.6	29	750	0.25	14	5	16.7
1N970B	24	5.2	33	750	0.25	13	5	18.2
1N971B	27	4.6	41	750	0.25	11	5	20.6
1N972B	30	4.2	49	1000	0.25	10	5	22.8
1N973B	33	3.8	58	1000	0.25	9.2	5	25.1
1N974B	36	3.4	70	1000	0.25	8.5	5	27.4
1N975B	39	3.2	80	1000	0.25	7.8	5	29.7
1N976B	43	3	93	1500	0.25	7	5	32.7
1N977B	47	2.7	105	1500	0.25	6.4	5	35.8
1N978B	51	2.5	125	1500	0.25	5.9	5	38.8
1N979B	56	2.2	150	2000	0.25	5.4	5	42.6
1N980B	62	2	185	2000	0.25	4.9	5	47.1

GENERAL DATA — 500 mW DO-35 GLASS

Type Number (Note 1)	Nominal Zener Voltage V_Z (Note 2) Volts	Test Current I_{ZT} mA	Maximum Zener Impedance (Note 3)			Maximum DC Zener Current I_{ZM} (Note 4) mA	Maximum Reverse Leakage Current	
			Z_{ZT} @ I_{ZT} Ohms	Z_{ZK} @ I_{ZK} Ohms	I_{ZK} mA		I_R Maximum μA	Test Voltage Vdc V_R
1N981B	68	1.8	230	2000	0.25	4.5	5	51.7
1N982B	75	1.7	270	2000	0.25	4.1	5	56
1N983B	82	1.5	330	3000	0.25	3.7	5	62.2
1N984B	91	1.4	400	3000	0.25	3.3	5	69.2
1N985B	100	1.3	500	3000	0.25	3	5	76
1N986B	110	1.1	750	4000	0.25	2.7	5	83.6
1N987B	120	1	900	4500	0.25	2.5	5	91.2
1N988B	130	0.95	1100	5000	0.25	2.3	5	98.8
1N989B	150	0.85	1500	6000	0.25	2	5	114
1N990B	160	0.8	1700	6500	0.25	1.9	5	121.6
1N991B	180	0.68	2200	7100	0.25	1.7	5	136.8
1N992B	200	0.65	2500	8000	0.25	1.5	5	152

NOTE 1. TOLERANCE AND VOLTAGE DESIGNATION

Tolerance Designation

The type numbers shown have tolerance designations as follows:

1N4370A series: $\pm 5\%$ units, C for $\pm 2\%$, D for $\pm 1\%$.

1N746A series: $\pm 5\%$ units, C for $\pm 2\%$, D for $\pm 1\%$.

1N987B series: $\pm 5\%$ units, C for $\pm 2\%$, D for $\pm 1\%$.

NOTE 2. ZENER VOLTAGE (V_Z) MEASUREMENT

Nominal zener voltage is measured with the device junction in thermal equilibrium at the lead temperature of $30^\circ C \pm 1^\circ C$ and $3/8"$ lead length.

NOTE 3. ZENER IMPEDANCE (Z_Z) DERIVATION

Z_{ZT} and Z_{ZK} are measured by dividing the ac voltage drop across the device by the ac current applied. The specified limits are for $I_Z(ac) = 0.1 I_Z(dc)$ with the ac frequency = 60 Hz.

NOTE 4. MAXIMUM ZENER CURRENT RATINGS (I_{ZM})

Values shown are based on the JEDEC rating of 400 mW. Where the actual zener voltage (V_Z) is known at the operating point, the maximum zener current may be increased and is limited by the derating curve.

GENERAL DATA — 500 mW DO-35 GLASS

Low level oxide passivated zener diodes for applications requiring extremely low operating currents, low leakage, and sharp breakdown voltage.

- Zener Voltage Specified @ $I_{ZT} = 50 \mu\text{A}$
- Maximum Delta V_Z Given from 10 to 100 μA

ELECTRICAL CHARACTERISTICS ($T_A = 25^\circ\text{C}$, $V_F = 1.5 \text{ V Max}$ at $I_F = 100 \text{ mA}$ for all types)

Type Number (Note 1)	Zener Voltage V_Z @ $I_{ZT} = 50 \mu\text{A}$ Volts			Maximum Reverse Current $I_R \mu\text{A}$ (Note 3)	Test Voltage V_R Volts	Maximum Zener Current $I_{ZM} \text{ mA}$ (Note 2)	Maximum Voltage Change ΔV_Z Volts (Note 4)
	Nom (Note 1)	Min	Max				
1N4678	1.8	1.71	1.89	7.5	1	120	0.7
1N4679	2	1.9	2.1	5	1	110	0.7
1N4680	2.2	2.09	2.31	4	1	100	0.75
1N4681	2.4	2.28	2.52	2	1	95	0.8
1N4682	2.7	2.565	2.835	1	1	90	0.85
1N4683	3	2.85	3.15	0.8	1	85	0.9
1N4684	3.3	3.135	3.465	7.5	1.5	80	0.95
1N4685	3.6	3.42	3.78	7.5	2	75	0.95
1N4686	3.9	3.705	4.095	5	2	70	0.97
1N4687	4.3	4.085	4.515	4	2	65	0.99
1N4688	4.7	4.465	4.935	10	3	60	0.99
1N4689	5.1	4.845	5.355	10	3	55	0.97
1N4690	5.6	5.32	5.88	10	4	50	0.96
1N4691	6.2	5.89	6.51	10	5	45	0.95
1N4692	6.8	6.46	7.14	10	5.1	35	0.9
1N4693	7.5	7.125	7.875	10	5.7	31.8	0.75
1N4694	8.2	7.79	8.61	1	6.2	29	0.5
1N4695	8.7	8.265	9.135	1	6.6	27.4	0.1
1N4696	9.1	8.645	9.555	1	6.9	26.2	0.08
1N4697	10	9.5	10.5	1	7.6	24.8	0.1
1N4698	11	10.45	11.55	0.05	8.4	21.6	0.11
1N4699	12	11.4	12.6	0.05	9.1	20.4	0.12
1N4700	13	12.35	13.65	0.05	9.8	19	0.13
1N4701	14	13.3	14.7	0.05	10.6	17.5	0.14
1N4702	15	14.25	15.75	0.05	11.4	16.3	0.15
1N4703	16	15.2	16.8	0.05	12.1	15.4	0.16
1N4704	17	16.15	17.85	0.05	12.9	14.5	0.17
1N4705	18	17.1	18.9	0.05	13.6	13.2	0.18
1N4706	19	18.05	19.95	0.05	14.4	12.5	0.19
1N4707	20	19	21	0.01	15.2	11.9	0.2
1N4708	22	20.9	23.1	0.01	16.7	10.8	0.22
1N4709	24	22.8	25.2	0.01	18.2	9.9	0.24
1N4710	25	23.75	26.25	0.01	19	9.5	0.25
1N4711	27	25.65	28.35	0.01	20.4	8.8	0.27
1N4712	28	26.6	29.4	0.01	21.2	8.5	0.28
1N4713	30	28.5	31.5	0.01	22.8	7.9	0.3
1N4714	33	31.35	34.65	0.01	25	7.2	0.33
1N4715	36	34.2	37.8	0.01	27.3	6.6	0.36
1N4716	39	37.05	40.95	0.01	29.6	6.1	0.39
1N4717	43	40.85	45.15	0.01	32.6	5.5	0.43

NOTE 1. TOLERANCE AND VOLTAGE DESIGNATION (V_Z)

The type numbers shown have a standard tolerance of $\pm 5\%$ on the nominal Zener voltage, C for $\pm 2\%$, D for $\pm 1\%$.

NOTE 2. MAXIMUM ZENER CURRENT RATINGS (I_{ZM})

Maximum Zener current ratings are based on maximum Zener voltage of the individual units and JEDEC 250 mW rating.

NOTE 3. REVERSE LEAKAGE CURRENT (I_R)

Reverse leakage currents are guaranteed and measured at V_R as shown on the table.

NOTE 4. MAXIMUM VOLTAGE CHANGE (ΔV_Z)

Voltage change is equal to the difference between V_Z at 100 μA and V_Z at 10 μA .

NOTE 5. ZENER VOLTAGE (V_Z) MEASUREMENT

Nominal Zener voltage is measured with the device junction in thermal equilibrium at the lead temperature at $30^\circ\text{C} \pm 1^\circ\text{C}$ and 3/16" lead length.

1N4728A - 1N4758A

Zener Diodes

Tolerance = 5%



DO-41 Glass case

COLOR BAND DENOTES CATHODE

Absolute Maximum Ratings * $T_a = 25^\circ\text{C}$ unless otherwise noted

Symbol	Parameter	Value	Units
P_D	Power Dissipation @ $T_L \leq 50^\circ\text{C}$, Lead Length = 3/8"	1.0	W
	Derate above 50°C	6.67	mW/°C
T_J, T_{STG}	Operating and Storage Temperature Range	-65 to +200	°C

* These ratings are limiting values above which the serviceability of the diode may be impaired.

Electrical Characteristics $T_a = 25^\circ\text{C}$ unless otherwise noted

Device	V_Z (V) @ I_Z (Note 1)			Test Current I_Z (mA)	Max. Zener Impedance			Leakage Current		Non-Repitive Peak Reverse Current I_{ZRM} (mA) (Note 2)
	Min.	Typ.	Max.		Z_Z @ I_Z (Ω)	Z_{ZK} @ I_{ZK} (Ω)	I_{ZK} (mA)	I_R (μA)	V_R (V)	
1N4728A	3.135	3.3	3.465	75	10	400	1	100	1	1380
1N4729A	3.42	3.6	3.78	69	10	400	1	100	1	1260
1N4730A	3.705	3.9	4.095	64	9	400	1	50	1	1190
1N4731A	4.085	4.3	4.515	58	9	400	1	10	1	1070
1N4732A	4.465	4.7	4.935	53	8	500	1	10	1	970
1N4733A	4.845	5.1	5.355	49	7	550	1	10	1	890
1N4734A	5.32	5.6	5.88	45	5	600	1	10	2	810
1N4735A	5.89	6.2	6.51	41	2	700	1	10	3	730
1N4736A	6.46	6.8	7.14	37	3.5	700	1	10	4	660
1N4737A	7.125	7.5	7.875	34	4	700	0.5	10	5	605
1N4738A	7.79	8.2	8.61	31	4.5	700	0.5	10	6	550
1N4739A	8.645	9.1	9.555	28	5	700	0.5	10	7	500
1N4740A	9.5	10	10.5	25	7	700	0.25	10	7.6	454
1N4741A	10.45	11	11.55	23	8	700	0.25	5	8.4	414
1N4742A	11.4	12	12.6	21	9	700	0.25	5	9.1	380

Device	V_Z (V) @ I_Z (Note 1)			Test Current I_Z (mA)	Max. Zener Impedance			Leakage Current		Non-Repetitive Peak Reverse Current I_{ZSM} (mA) (Note 2)
	Min.	Typ.	Max.		Z_Z @ I_Z (Ω)	Z_{ZK} @ I_{ZK} (Ω)	I_{ZK} (mA)	I_R (μ A)	V_R (V)	
1N4743A	12.35	13	13.65	19	10	700	0.25	5	9.9	344
1N4744A	14.25	15	15.75	17	14	700	0.25	5	11.4	304
1N4745A	15.2	16	16.8	15.5	16	700	0.25	5	12.2	285
1N4746A	17.1	18	18.9	14	20	750	0.25	5	13.7	250
1N4747A	19	20	21	12.5	22	750	0.25	5	15.2	225
1N4748A	20.9	22	23.1	11.5	23	750	0.25	5	16.7	205
1N4749A	22.8	24	25.2	10.5	25	750	0.25	5	18.2	190
1N4750A	25.65	27	28.35	9.5	35	750	0.25	5	20.6	170
1N4751A	28.5	30	31.5	8.5	40	1000	0.25	5	22.8	150
1N4752A	31.35	33	34.65	7.5	45	1000	0.25	5	25.1	135
1N4753A	34.2	36	37.8	7	50	1000	0.25	5	27.4	125
1N4754A	37.05	39	40.95	6.5	60	1000	0.25	5	29.7	115
1N4755A	40.85	43	45.15	6	70	1500	0.25	5	32.7	110
1N4756A	44.65	47	49.35	5.5	80	1500	0.25	5	35.8	95
1N4757A	48.45	51	53.55	5	95	1500	0.25	5	38.8	90
1N4758A	53.2	56	58.8	4.5	110	2000	0.25	5	42.6	80

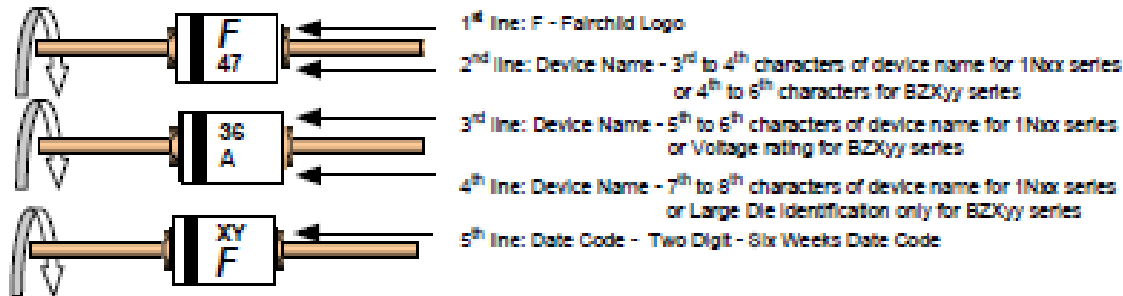
Notes:1. Zener Voltage (V_Z)The zener voltage is measured with the device junction in the thermal equilibrium at the lead temperature (T_L) at $30^\circ\text{C} \pm 1^\circ\text{C}$ and 30" lead length.

2. 2 Square wave Reverse Surge at 0.3 msec peak time.

Top Mark Information

Device	Line 1	Line 2	Line 3	Line 4	Line 5
1N4728A	LOGO	47	28	A	XY
1N4729A	LOGO	47	29	A	XY
1N4730A	LOGO	47	30	A	XY
1N4731A	LOGO	47	31	A	XY
1N4732A	LOGO	47	32	A	XY
1N4733A	LOGO	47	33	A	XY
1N4734A	LOGO	47	34	A	XY
1N4735A	LOGO	47	35	A	XY
1N4736A	LOGO	47	36	A	XY
1N4737A	LOGO	47	37	A	XY
1N4738A	LOGO	47	38	A	XY
1N4739A	LOGO	47	39	A	XY
1N4740A	LOGO	47	40	A	XY
1N4741A	LOGO	47	41	A	XY
1N4742A	LOGO	47	42	A	XY
1N4743A	LOGO	47	43	A	XY
1N4744A	LOGO	47	44	A	XY
1N4745A	LOGO	47	45	A	XY
1N4746A	LOGO	47	46	A	XY
1N4747A	LOGO	47	47	A	XY
1N4748A	LOGO	47	48	A	XY
1N4749A	LOGO	47	49	A	XY
1N4750A	LOGO	47	50	A	XY
1N4751A	LOGO	47	51	A	XY
1N4752A	LOGO	47	52	A	XY
1N4753A	LOGO	47	53	A	XY
1N4754A	LOGO	47	54	A	XY
1N4755A	LOGO	47	55	A	XY
1N4756A	LOGO	47	56	A	XY
1N4757A	LOGO	47	57	A	XY
1N4758A	LOGO	47	58	A	XY

Top Mark Information (Continued)



General Requirements:

- 1.0 Cathode Band
- 2.0 First Line: F - Fairchild Logo
- 3.0 Second Line: Device name - For 1Nxx series: 3rd to 4th characters of the device name.
For BZXxx series: 4th to 6th characters of the device name.
- 4.0 Third Line: Device name - For 1Nxx series: 5th to 6th characters of the device name.
For BZXyy series: Voltage rating
- 5.0 Third Line: Device name - For 1Nxx series: 7th to 8th characters of the device name.
(the 8th character is the large die identification)
For BZXyy series: Large Die Identification character
- 6.0 Fourth Line: Date Code - Two Digit - Six Weeks Date Code
Where: X represents the last digit of the calendar year
Y represents the Six weeks numeric code
- 7.0 Devices shall be marked as required in the device specification (PID or FSC Test Spec).
- 8.0 Maximum no. of marking lines: 5
- 9.0 Maximum no. of digits per line: 3
- 10.0 FSC logo must be 20 % taller than the alphanumeric marking and should occupy the 2 characters of the specified line.
- 11.0 Marking Font: Arial (Except FSC Logo)
- 12.0 First character of each marking line must be aligned vertically.
- 13.0 All device markings must be based on Fairchild device specification.



TRADEMARKS

The following includes registered and unregistered trademarks and service marks, owned by Fairchild Semiconductor and/or its global subsidiaries, and is not intended to be an exhaustive list of all such trademarks.

Auto-SPM™
Build it Now™
CorePLUS™
CorePOWER™
CROSSVOLT™
CTL™
Current Transfer Logic™
EcoSPARK®
EfficientMax™
EZSWITCH™
E3™
F™
Fairchild®
Fairchild Semiconductor®
FACT Quiet Series™
FACT®
FAST®
FlashCore™
FETBench™
FlashWriter®
FPS™

F-PPS™
FRFET®
Global Power Resource™
Green FPS™
Green FPS™ e-Series™
Gmax™
GTC™
IntelIMAX™
ISOPLANAR™
Megabuck™
MICROCOUPLER™
MicroFET™
MicroPak™
MillerDrive™
MotionMax™
Motion-SPM™
OPTOLOGIC®
OPTOPLANAR®
PDP SPM™
Power-SPM™

PowerTrench®
PowerXS™
Programmable Active Diode™
GFET®
QS™
Quiet Series™
RapidConfigure™

Saving our world, 1mW/W at a time™
SmartMax™
SMART START™
SPM®
STEALTH™
SuperFET™
SuperSOT™-3
SuperSOT™-6
SuperSOT™-8
SuperMOS™
SyncFET™
Sync-Lock™


The Power franchise®

TinyBoost™
TinyBuck™
TinyLogic®
TINYOPTO™
TinyPower™
TinyPWM™
TinyWire™
TIPault Detect™
TRUECURRENT™
µSerDes™

UHC
Ultra FRFET™
UniFET™
VCM™
VisualMax™
XS™

* Trademarks of System General Corporation, used under license by Fairchild Semiconductor.

DISCLAIMER

FAIRCHILD SEMICONDUCTOR RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN TO IMPROVE RELIABILITY, FUNCTION, OR DESIGN. FAIRCHILD DOES NOT ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT DESCRIBED HEREIN; NEITHER DOES IT CONVEY ANY LICENSE UNDER ITS PATENT RIGHTS, NOR THE RIGHTS OF OTHERS. THESE SPECIFICATIONS DO NOT EXPAND THE TERMS OF FAIRCHILD'S WORLDWIDE TERMS AND CONDITIONS, SPECIFICALLY THE WARRANTY THRESHOLD, WHICH COVERS THESE PRODUCTS.

LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF FAIRCHILD SEMICONDUCTOR CORPORATION.

As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body or (b) support or sustain life, and (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury of the user.
2. A critical component in any component of a life support, device, or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

ANTI-COUNTERFEITING POLICY

Fairchild Semiconductor Corporation's Anti-Counterfeiting Policy. Fairchild's Anti-Counterfeiting Policy is also stated on our external website, www.fairchildsemi.com, under Sales Support.

Counterfeiting of semiconductor parts is a growing problem in the industry. All manufacturers of semiconductor products are experiencing counterfeiting of their parts. Customers who inadvertently purchase counterfeit parts experience many problems such as loss of brand reputation, substandard performance, failed applications, and increased cost of production and manufacturing delays. Fairchild is taking strong measures to protect ourselves and our customers from the proliferation of counterfeit parts. Fairchild strongly encourages customers to purchase Fairchild parts either directly from Fairchild or from Authorized Fairchild Distributors who are listed by country on our web page cited above. Products customers buy either from Fairchild directly or from Authorized Fairchild Distributors are genuine parts, have full traceability, meet Fairchild's quality standards for handling and storage and provide access to Fairchild's full range of up-to-date technical and product information. Fairchild and our Authorized Distributors will stand behind all warranties and will appropriately address any warranty issues that may arise. Fairchild will not provide any warranty coverage or other assistance for parts bought from Unauthorized Sources. Fairchild is committed to combat this global problem and encourage our customers to do their part in stopping this practice by buying direct or from authorized distributors.

PRODUCT STATUS DEFINITIONS

Definition of Terms

Datasheet Identification	Product Status	Definition
Advance Information	Formative / In Design	Datasheet contains the design specifications for product development. Specifications may change in any manner without notice.
Preliminary	FIRST PRODUCTION	Datasheet contains preliminary data; supplementary data will be published at a later date. Fairchild Semiconductor reserves the right to make changes at any time without notice to improve design.
No Identification Needed	Full Production	Datasheet contains final specifications. Fairchild Semiconductor reserves the right to make changes at any time without notice to improve the design.
Obsolete	Not in Production	Datasheet contains specifications on a product that is discontinued by Fairchild Semiconductor. The datasheet is for reference information only.

Rev. 40

ELECTRONIC SPEED CONTROLLER USERS MANUAL

Thank you for purchasing our products! For the high power of this brushless system, failure to use may result in injury yourself and damage of the whole device. So we highly recommend you to read carefully and abide by the operating procedures of this manual before the first flight. Turnigy is not responsible for your misuse of this product or any damage including incidental losses or indirect losses you may cause. Moreover, we have not any responsibility for the modification of our products without authorization. We have the right to change the design, features, functions and operating requirements of our products without any advanced notice!

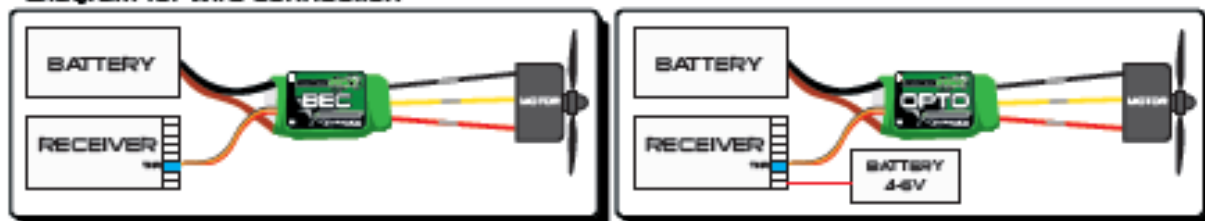
IMPORTANT PRECAUTIONS

- Read the manual correctly before your operating.
- Do not connect oppositely the polarity between the battery pack and the Electronic Speed Controller(for short ESC).
- Wrong connecting of polarity will damage the ESC.
- The working range of ESC do not exceed the corresponding voltage and current.
- Do not disassemble any electronic components of ESC, or else it will cause permanent damage or information losses.
- Do not allow any unqualified battery pack.
- Do not connect a degaussed motors.
- Do not use any substandard cable connector.
- Do not allow chemical agent and water onto the ESC.
- Do not take the battery away when the motor is rotating, or else it may cause high burst current to damage the ESC.
- The ESC should be in a position which allows good airflow and heat dissipation.
- Always disconnect the battery from the ESC when not in use.

FEATURES

- Super fine and smooth touch of speed controlling, first-rate accurate linearity and quick-respond speed of throttle.
- Separate voltage regulator IC for MCU(Micro Controller Unit), high capability of anti-interference to decrease the possibility out of control.
- Low-voltage cut-off protection of battery, blocked rotation protection and throttle signal loss protection, etc. All these functions can prolong the service life of ESC effectively.
- It can be compatible with a programming card(optional component), which has a simple and visual surface so as to change the parameters conveniently at any time anywhere [reference to the manual for programming card].
- Good and safe performance of power-on. The motor won't be started no matter which position the throttle stick is on when the battery is connected.

Diagram for wire connection



- Plug the JR connector into the throttle control channel of the receiver.
- Switch "on" the transmitter and move the stick of throttle to the lowest position.
- Connect the main power pack to ESC (pay attention to the polarity).
- The motor transmits a single or double beeps to confirm the correct connection. The ESC is ready to be operated and the motor can be switched on.
- A single beep announces that the brake is on; the double beep says that the brake is off.
- If you didn't hear the above "beep", please disconnect the ESC to battery pack and check whether the JR connector is connected correctly to the throttle control channel of receiver, and whether the throttle stick is at the lowest position or you choose the right direction of "NOR/REV" of throttle channel in the transmitter.
- The rotating direction of motor can be changed by exchanging two of three wires from ESC or setting the Prog-Card.

Option Parameter Assistant by Transmitter

- The transmitter can set only one parameter of ESC at a time. If you want to set several parameters, please repeat the following procedures:
- Plug the JR connector into throttle control channel of receiver. Switch "on" the transmitter and move the stick of throttle to the highest position.
- Connect the main power pack to ESC (pay attention to the polarity).
- Wait for 5 seconds, you will hear 4 beeps, that means it entered the programming mode.
- And then you will hear 5 "single beep", then 5 "double beep", then 5 "triple beep", then 5 "quartet beeps" and then 5 "penta groups". And these sounds will circulate continuously.
- Each group of 5 sounds stands for a different parameter of ESC respectively.
- You can put the stick of throttle to the lowest position during one group of 5 sounds, then the corresponding mode is saved.
- Hear 1 "1" (Brake-Medium) or "1111" (Brake-Off), then you can exit the setting mode after saving the mode.
- When the mode is saved, you can disconnect the ESC to the battery pack.

■ Brake

Hear the first "1111" in the above circulation, put the throttle stick to the lowest position, the braking mode is changed from Brake-off into Brake-Medium. If you want to change back, please repeat the above procedures, and vice versa.

■ Battery Type

When you hear 5 groups "1" or "11" or "111", move the stick of throttle to the lowest position.

1 "beep": LiXX battery

2 "beep": NiCd, NiMH battery

3 "beep": LiFe battery

■ Timing

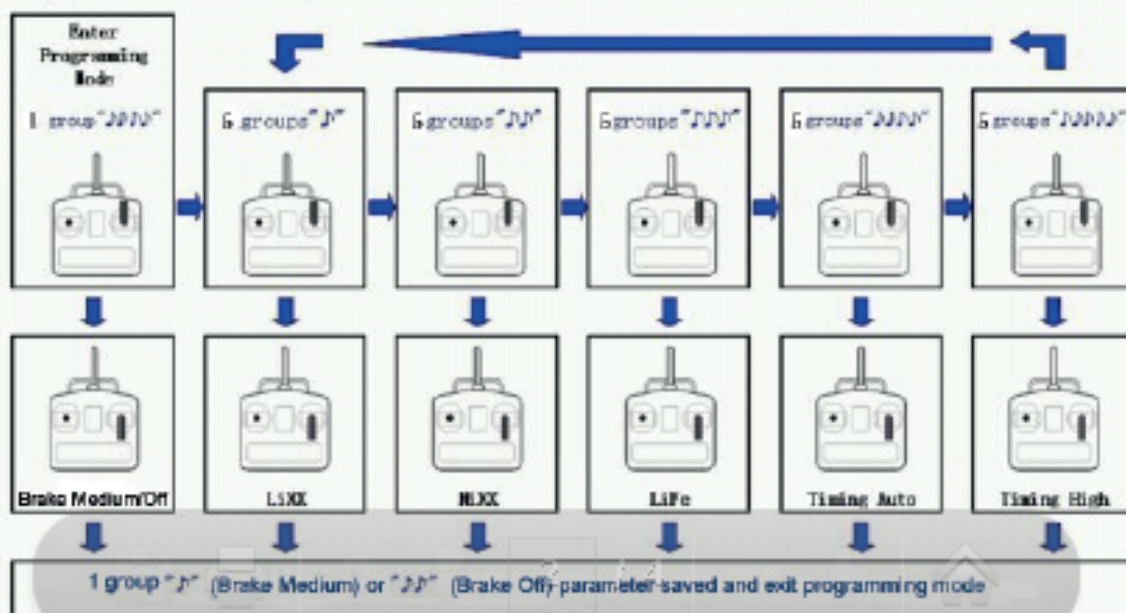
Hear 5 groups "1111" or "11111", move the stick of throttle to the lowest position.

4 "beep": Automatically change the timing of motor (recommended for all types of motors)

5 "beep": Timing-High (recommended for 10 [or more] poles motors and outrunner motors)

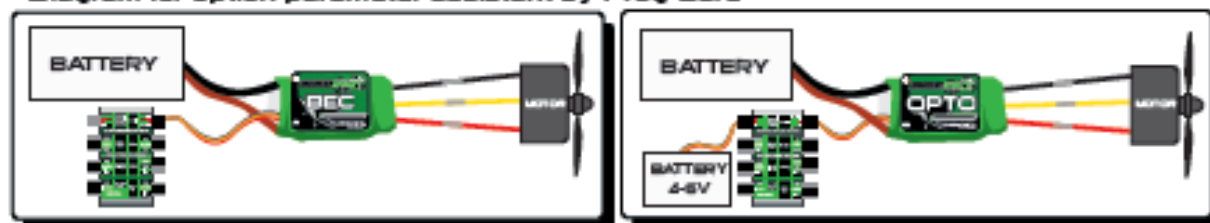
Note: When the timing mode of motor is saved, please adjust the motor on the ground before the flight.

Diagram for option parameter assistant by transmitter



Operation - For ESC with Prog Card

Diagram for option parameter assistant by Prog Card



- Put the six jumper connector to the required positions.
- Plug JR connector (part of ESC) to the specified socket on Prog-Card [orange wire-signal, brown wire-cathode(-), red wire-anode(+)].
- Connect the motor to the ESC and connect the power to the ESC (Take care safety).
- For OPTO ESC without BEC, connect the 4.8V of battery pack to the Prog-card.
- 1 "beep" will be heard in a second, which means your setting has been saved ("beep" will not be heard if there is no change of parameter).
- Disconnect the power pack (For OPTO ESC, disconnect the Prog-Card to it's battery pack).

Option parameter assistant by Prog-Card

■ Brake Type

Brake - Off: Brake is switched off.

Brake - Medium: The brake is on and its middle effect is fit for gear drivers

Brake - Hard: The intensity of brake becomes much higher

■ Timing Mode

Timing - Automatic: automatic timing (for all types of motors)

Timing - High: hard timing (recommended for more than 10 poles motors and outrunners motors)

Timing - Low: soft timing (recommended for motors of 2-8 poles motors and inrunner motors)

Note: When the timing mode is saved, please adjust the motor on the ground before the flight.

■ Acceleration

Acceleration - High: fast acceleration or deceleration of motor

Acceleration - Medium: acceleration or deceleration in middle speed

Acceleration - Soft: slow acceleration or deceleration

■ Battery type

Battery Type - NiXX: Ni-Cd & Ni-MH

Battery Type - LiXX: Li-Ion & Li-polymer

Battery Type - LiFe

■ Cut-off Voltage - Cut-off Voltage of ESC

Cut-off Voltage - High: high cut-off voltage of battery type you selected

Cut-off Voltage - Medium: middle cut-off voltage of battery type you selected

Cut-off Voltage - Low: low cut-off voltage of battery type you selected

Cut off Voltage mode	Li-Ion & Li-polymer	Ni-Cd & Ni-MH	LiFe
High	3.2V	0.9V	2.6V
Medium	3.0V	0.8V	2.5V
Low	2.8V	0.6V	2.2V

■ Direction of motor rotation

The direction of rotation of motor can be achieved by exchanging any two of three output wires from ESC. If you don't want to change the wire between the ESC and motor, you can set the ESC to change the direction of rotation. If the direction of rotation is needed, when the jumper connector is put to the position, the ESC only receives the change of direction of motor rotation but the other changes from the Prog-Card. Continuously "beep" will be heard until the power pack will be cut off. When the direction of rotation is changed, please put the jumper connector to the former position.

 Factory Default Setting

Brake	Brake is On, Medium effect
Timing of motor	Automatic
Acceleration	Medium
Battery Type	Li-ion & Li-polymer
Cut off Voltage	Medium
Cut off mode	Slow down
Direction of rotation	Rightward



4

/ 4



DATASHEET

BLUETOOTH TO SERIAL PORT

MODULE

HC05



Overview

HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup.

Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Bluecore 04-External single chip Bluetooth system with CMOS technology and with AFH (Adaptive Frequency Hopping Feature). It has the

footprint as small as 12.7mmx27mm. Hope it will simplify your overall design/development cycle.

www.electronica60norte.com
electronica60norte@hotmail.com

Specifications

Hardware features

- Typical -80dBm sensitivity.
- Up to +4dBm RF transmit power.
- Low Power 1.8V Operation, 3.3 to 5 V I/O.
- PIO control.
- UART interface with programmable baud rate.
- With integrated antenna.
- With edge connector.

Software features

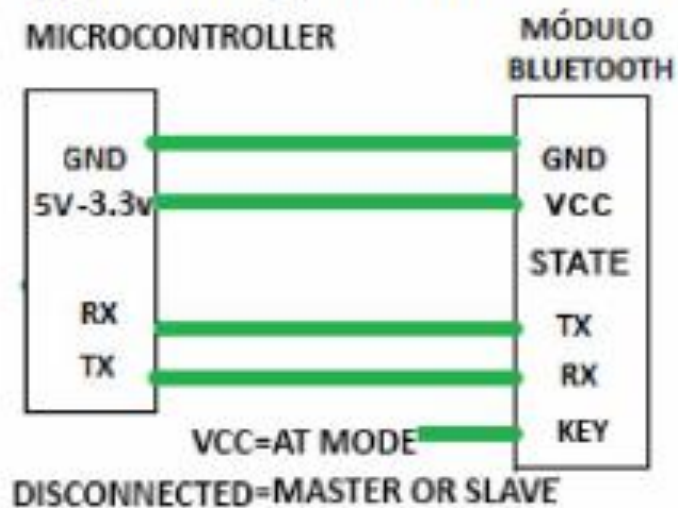
- Slave default Baud rate: 9600, Data bits:8, Stop bit:1,Parity:No parity.
- PIO9 and PIO8 can be connected to red and blue led separately. When master and slave are paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2times/s.
- Auto-connect to the last device on power as default.
- Permit pairing device to connect as default.
- Auto-pairing PINCODE:"1234" as default.
- Auto-reconnect in 30 min when disconnected as a result of beyond the range of connection.

www.electronica60norte.com
electronica60norte@hotmail.com

Pin out configuration



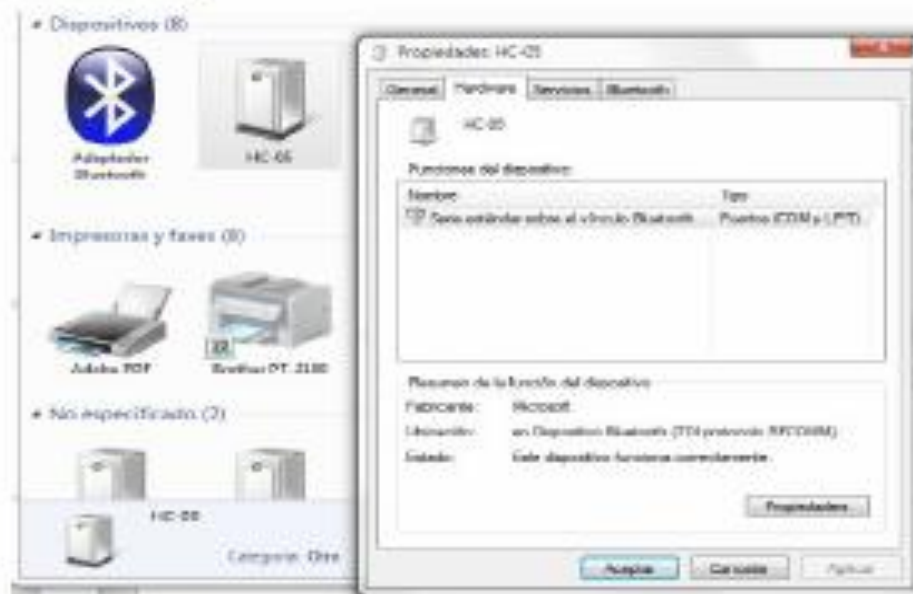
Typical Application Circuit



www.electronica80norte.com
electronica80norte@hotmail.com

After connect the Bluetooth module, scan for new devices from the PC and you will find the module with the device name "HC-05", after that, click to connect, if some message appears asking about "Pairing code" just put "1234" as default code.

BLUE LED = ACTIVE (Blinking 500ms period inactive connection, change 1seg with active connection)



Open a serial terminal and select the serial COM x port number that assigned Windows to Bluetooth Module.

Configure the serial terminal with these parameters:

- Baud rate: 9600.
- Data bits:8.
- Stop bit:1.
- Parity: No parity.

www.electronica80norte.com
electronica80norte@hotmail.com

Open connection and you will be ready to send and receive data from module Bluetooth like Serial Port COM



AT COMMANDS

How to get to AT COMMAND mode

- 1: Connect KEY pin to VCC.
- 2: Supply power to module. Then the module will enter into AT MODE. In this mode you have to use baud rate at 38400. In this way, user should change the baud rate for SLAVE AND MASTER mode.

How to set this module as “Master - Host” role

- 1: Input high level to KEY.
 - 2: Supply power to the module. And the module will enter to AT COMMAND.
 - 3: Set the parameters of the hyper terminal or the other serial tools (baud rate: 38400, data bit:8, stop bit:1, no parity bit, no Flow Control).
 - 4: Sent the characters “AT+ROLE=1\r\n” through serial, then receive the characters “OK\r\n”. Here, “\r\n” is the CRLF.
 - 5: Sent the characters “AT+CMODE=1\r\n” through serial, then receive the characters “OK\r\n”. Here, “\r\n” is the CRLF.
 - 6: Default factory password passkey is: 1243, this must be the same in the Bluetooth slave module if you want to pair it.
- To read passkey use this command: “AT+PSWD?”.
- To Reset the password command sent the characters “AT+PSWD=XXXX”.
- The password must be 4-bits.

7: Leave free KEY, and supply power to the module again. Then this module will become master role and search the other module (slave role) automatically to build the connection (baud rate:9600, data bit:8, stop bit:1, no parity bit, no Flow Control).

How to set this module be the “Slave - Device” role

1: Input high level to KEY.

2: Supply power to the module. And the module will enter to AT COMMAND.

3: Set the parameters of the super terminal or the other serial tools (baud rate:

38400, data bit:8, stop bit:1, no parity bit, no Flow Control).

4: Sent the characters “AT+ROLE=0\r\n” through serial, then receive the characters “OK\r\n”. Here, “\r\n” is the CRLF.

5: Sent the characters “AT+CMODE=0\r\n” through serial, then receive the characters “OK\r\n”. Here, “\r\n” is the CRLF.

6: Default factory password passkey is: 1243, this must be the same in the Bluetooth master module if you want to pair it.

To read passkey sent the characters “AT+PSWD?”.

To Reset the password command sent the characters “AT+PSWD=XXXX”.

The password must be 4-bits.

7: Leave free KEY, and supply power to the module again. Then this module will become slave role and wait to be discover it by the other module (master role) automatically to build the connection (baud rate:38400, data bit:8, stop bit:1, no parity bit, no Flow Control).

How to get to the standard communication mode

1: Leave free KEY, don't connect it to VDD neither GND.

2: Supply power to the module. Then the module will enter to communication mode. It can be used for pairing.

Notes

(1) HC-05's command should end up with “\r\n”. It means when you finish programming, you should add terminator (“ENTER” or “0x0d 0x0a”) to the program.

(2) The most common commands for HC-05 are: AT+ROLE (set master-slave), AT+CMODE(set address pairing) , AT+PSWD (set password).

If you want the master module has the function of remembering slave module,the most simply way is: First, set AT+CMODE=1. Make the master module pair with the slave module. Second, set AT+CMODE=0. Then the master module just can make pair with that specified slave module.

www.electronicab0norte.com

electronicab0norte@hotmail.com



Atmel 8-bit Microcontroller with 4/8/16/32KBytes In-System Programmable Flash

**ATmega48A; ATmega48PA; ATmega88A; ATmega88PA;
ATmega168A; ATmega168PA; ATmega328; ATmega328P**

Features

- High Performance, Low Power Atmel[®] AVR[®] 8-Bit Microcontroller Family
- Advanced RISC Architecture
 - 131 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 20 MIPS Throughput at 20MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
 - 4/8/16/32KBytes of In-System Self-Programmable Flash program memory
 - 256/512/512/1KBytes EEPROM
 - 512/1K/1K/2KBytes Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- Atmel[®] QTouch[®] library support
 - Capacitive touch buttons, sliders and wheels
 - QTouch and QMatrix[®] acquisition
 - Up to 64 sense channels
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Six PWM Channels
 - 8-channel 10-bit ADC in TQFP and QFN/MLF package
 - Temperature Measurement
 - 6-channel 10-bit ADC in PDIP Package
 - Temperature Measurement
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Byte-oriented 2-wire Serial Interface (Philips I²C compatible)
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
 - 23 Programmable I/O Lines
 - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
 - 1.8 - 5.5V
- Temperature Range:
 - -40°C to 85°C
- Speed Grade:
 - 0 - 4MHz @ 1.8 - 5.5V, 0 - 10MHz @ 2.7 - 5.5V, 0 - 20MHz @ 4.5 - 5.5V
- Power Consumption at 1MHz, 1.8V, 25°C
 - Active Mode: 0.2mA
 - Power-down Mode: 0.1µA
 - Power-save Mode: 0.75µA (Including 32kHz RTC)

1. Pin Configurations

Figure 1-1. Pinout ATmega48A/PA/88A/PA/168A/PA/328/P

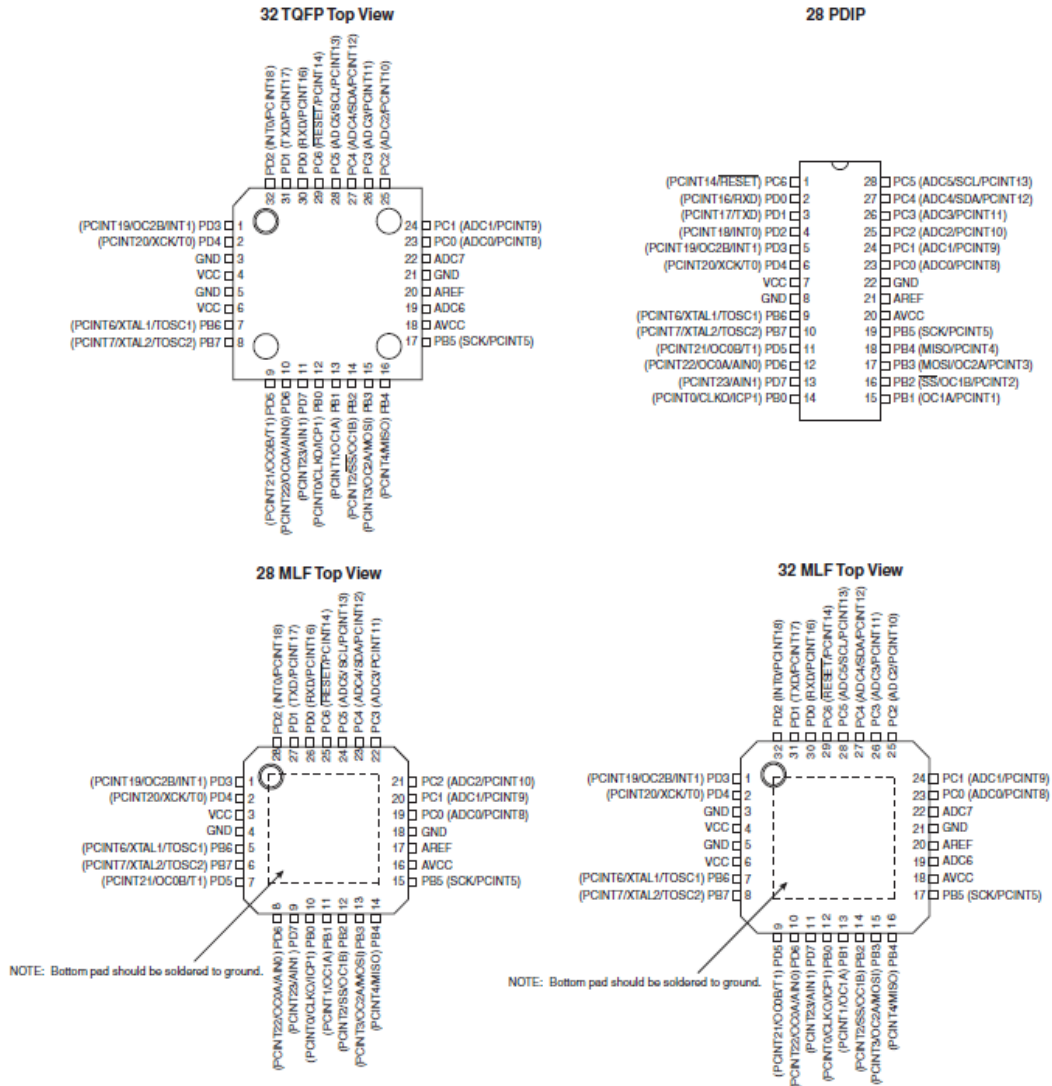


Table 1-1. 32UFBGA - Pinout ATmega48A/48PA/88A/88PA/168A/168PA

	1	2	3	4	5	6
A	PD2	PD1	PC6	PC4	PC2	PC1
B	PD3	PD4	PD0	PC5	PC3	PC0
C	GND	GND			ADC7	GND
D	VDD	VDD			AREF	ADC6
E	PB6	PD6	PB0	PB2	AVDD	PB5
F	PB7	PD5	PD7	PB1	PB3	PB4

1.1 Pin Descriptions

1.1.1 VCC

Digital supply voltage.

1.1.2 GND

Ground.

1.1.3 Port B (PB7:0) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.

If the Internal Calibrated RC Oscillator is used as chip clock source, PB7...6 is used as TOSC2...1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

The various special features of Port B are elaborated in ["Alternate Functions of Port B" on page 83](#) and ["System Clock and Clock Options" on page 26](#).

1.1.4 Port C (PC5:0)

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5...0 output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

1.1.5 PC6/ $\overline{\text{RESET}}$

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. The minimum pulse length is given in [Table 29-16 on page 312](#). Shorter pulses are not guaranteed to generate a Reset.

The various special features of Port C are elaborated in ["Alternate Functions of Port C" on page 86](#).|

1.1.6 Port D (PD7:0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

The various special features of Port D are elaborated in ["Alternate Functions of Port D" on page 89](#).

1.1.7 AV_{CC}

AV_{CC} is the supply voltage pin for the A/D Converter, PC3:0, and ADC7:6. It should be externally connected to V_{CC}, even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter. Note that PC6...4 use digital supply voltage, V_{CC}.

1.1.8 AREF

AREF is the analog reference pin for the A/D Converter.

1.1.9 ADC7:6 (TQFP and QFN/MLF Package Only)

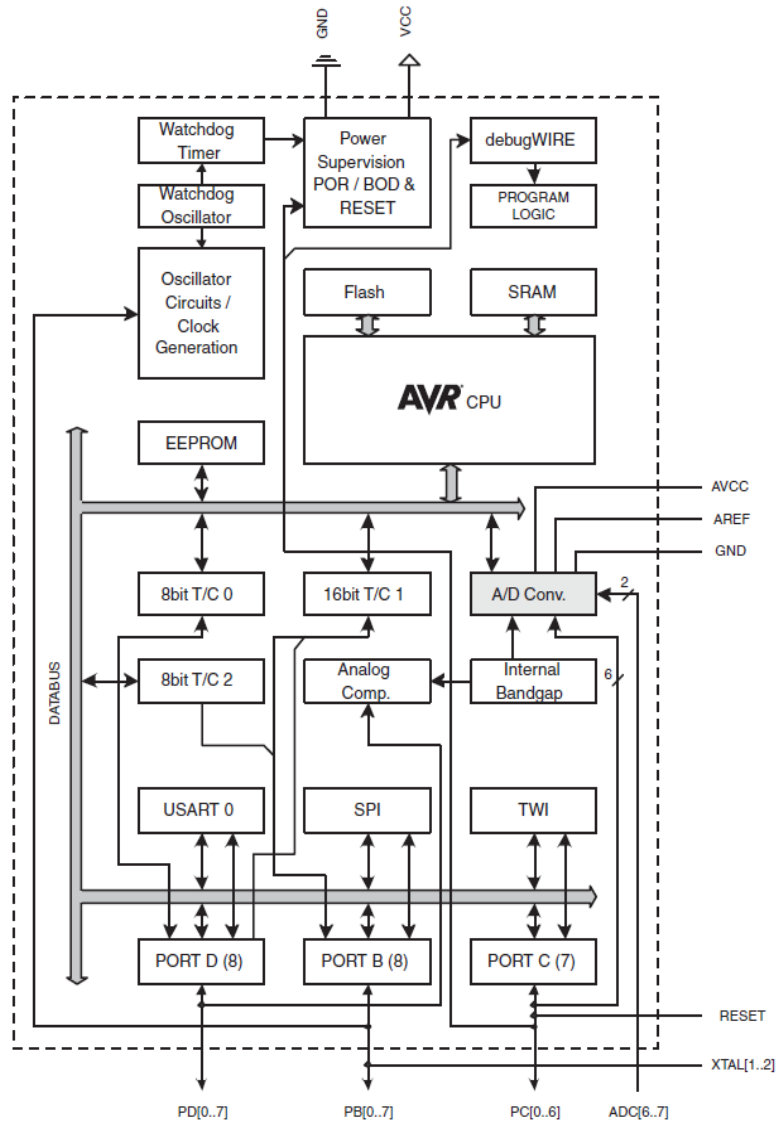
In the TQFP and QFN/MLF package, ADC7:6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.

2. Overview

The ATmega48A/PA/88A/PA/168A/PA/328/P is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega48A/PA/88A/PA/168A/PA/328/P achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

2.1 Block Diagram

Figure 2-1. Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega48A/PA/88A/PA/168A/PA/328/P provides the following features: 4K/8Kbytes of In-System Programmable Flash with Read-While-Write capabilities, 256/512/512/1Kbytes EEPROM, 512/1K/1K/2Kbytes SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, a byte-oriented 2-wire Serial Interface, an SPI serial port, a 6-channel 10-bit ADC (8 channels in TQFP and QFN/MLF packages), a programmable Watchdog Timer with internal Oscillator, and five software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, USART, 2-wire Serial Interface, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or hardware reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption.

Atmel® offers the QTouch® library for embedding capacitive touch buttons, sliders and wheels functionality into AVR® microcontrollers. The patented charge-transfer signal acquisition offers robust sensing and includes fully debounced reporting of touch keys and includes Adjacent Key Suppression® (AKS™) technology for unambiguous detection of key events. The easy-to-use QTouch Suite toolchain allows you to explore, develop and debug your own touch applications.

The device is manufactured using Atmel's high density non-volatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed In-System through an SPI serial interface, by a conventional non-volatile memory programmer, or by an On-chip Boot program running on the AVR core. The Boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega48A/PA/88A/PA/168A/PA/328/P is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega48A/PA/88A/PA/168A/PA/328/P AVR is supported with a full suite of program and system development tools including: C Compilers, Macro Assemblers, Program Debugger/Simulators, In-Circuit Emulators, and Evaluation kits.

2.2 Comparison Between Processors

The ATmega48A/PA/88A/PA/168A/PA/328/P differ only in memory sizes, boot loader support, and interrupt vector sizes. Table 2-1 summarizes the different memory and interrupt vector sizes for the devices.

Table 2-1. Memory Size Summary

Device	Flash	EEPROM	RAM	Interrupt Vector Size
ATmega48A	4KBytes	256Bytes	512Bytes	1 instruction word/vector
ATmega48PA	4KBytes	256Bytes	512Bytes	1 instruction word/vector
ATmega88A	8KBytes	512Bytes	1KBytes	1 instruction word/vector
ATmega88PA	8KBytes	512Bytes	1KBytes	1 instruction word/vector
ATmega168A	16KBytes	512Bytes	1KBytes	2 instruction words/vector
ATmega168PA	16KBytes	512Bytes	1KBytes	2 instruction words/vector
ATmega328	32KBytes	1KBytes	2KBytes	2 instruction words/vector
ATmega328P	32KBytes	1KBytes	2KBytes	2 instruction words/vector

ATmega48A/PA/88A/PA/168A/PA/328/P support a real Read-While-Write Self-Programming mechanism. There is a separate Boot Loader Section, and the SPM instruction can only execute from there. In ATmega 48A/48PA there

is no Read-While-Write support and no separate Boot Loader Section. The SPM instruction can execute from the entire Flash

3. Resources

A comprehensive set of development tools, application notes and datasheets are available for download on <http://www.atmel.com/avr>.

4. Data Retention

Reliability Qualification results show that the projected data retention failure rate is much less than 1 PPM over 20 years at 85°C or 100 years at 25°C.

5. About Code Examples

This documentation contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. Be aware that not all C compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C compiler documentation for more details.

For I/O Registers located in extended I/O map, “IN”, “OUT”, “SBIS”, “SBIC”, “CBI”, and “SBI” instructions must be replaced with instructions that allow access to extended I/O. Typically “LDS” and “STS” combined with “SBRS”, “SBRC”, “SBR”, and “CBR”.

6. Capacitive Touch Sensing

The Atmel® QTouch® Library provides a simple to use solution to realize touch sensitive interfaces on most Atmel AVR® microcontrollers. The QTouch Library includes support for the Atmel QTouch and Atmel QMatrix® acquisition methods.

Touch sensing can be added to any application by linking the appropriate Atmel QTouch Library for the AVR Microcontroller. This is done by using a simple set of APIs to define the touch channels and sensors, and then calling the touch sensing API's to retrieve the channel information and determine the touch sensor states.

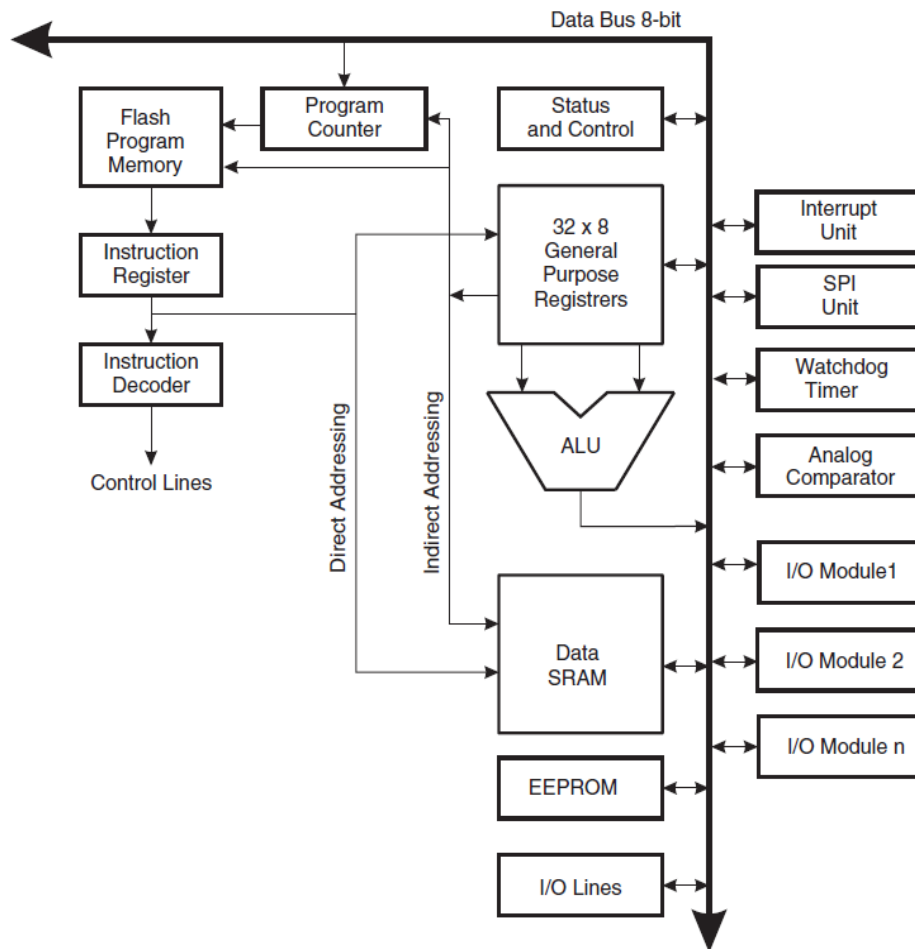
The QTouch Library is FREE and downloadable from the Atmel website at the following location: www.atmel.com/qtouchlibrary. For implementation details and other information, refer to the [Atmel QTouch Library User Guide](#) - also available for download from Atmel website.

7. AVR CPU Core

7.1 Overview

This section discusses the AVR core architecture in general. The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories, perform calculations, control peripherals, and handle interrupts.

Figure 7-1. Block Diagram of the AVR Architecture



In order to maximize performance and parallelism, the AVR uses a Harvard architecture – with separate memories and buses for program and data. Instructions in the program memory are executed with a single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is In-System Reprogrammable Flash memory.

The fast-access Register File contains 32 x 8-bit general purpose working registers with a single clock cycle access time. This allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typical ALU operation, two oper-

ands are output from the Register File, the operation is executed, and the result is stored back in the Register File – in one clock cycle.

Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing – enabling efficient address calculations. One of these address pointers can also be used as an address pointer for look up tables in Flash program memory. These added function registers are the 16-bit X-, Y-, and Z-register, described later in this section.

The ALU supports arithmetic and logic operations between registers or between a constant and a register. Single register operations can also be executed in the ALU. After an arithmetic operation, the Status Register is updated to reflect information about the result of the operation.

Program flow is provided by conditional and unconditional jump and call instructions, able to directly address the whole address space. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

Program Flash memory space is divided in two sections, the Boot Program section and the Application Program section. Both sections have dedicated Lock bits for write and read/write protection. The SPM instruction that writes into the Application Flash memory section must reside in the Boot Program section.

During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the Stack. The Stack is effectively allocated in the general data SRAM, and consequently the Stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the SP in the Reset routine (before sub-routines or interrupts are executed). The Stack Pointer (SP) is read/write accessible in the I/O space. The data SRAM can easily be accessed through the five different addressing modes supported in the AVR architecture.

The memory spaces in the AVR architecture are all linear and regular memory maps.

A flexible interrupt module has its control registers in the I/O space with an additional Global Interrupt Enable bit in the Status Register. All interrupts have a separate Interrupt Vector in the Interrupt Vector table. The interrupts have priority in accordance with their Interrupt Vector position. The lower the Interrupt Vector address, the higher the priority.

The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, SPI, and other I/O functions. The I/O Memory can be accessed directly, or as the Data Space locations following those of the Register File, 0x20 - 0x5F. In addition, the ATmega48A/PA/88A/PA/168A/PA/328/P has Extended I/O space from 0x60 - 0xFF in SRAM where only the ST/STS/STD and LD/LDS/LDD instructions can be used.

7.2 ALU – Arithmetic Logic Unit

The high-performance AVR ALU operates in direct connection with all the 32 general purpose working registers. Within a single clock cycle, arithmetic operations between general purpose registers or between a register and an immediate are executed. The ALU operations are divided into three main categories – arithmetic, logical, and bit-functions. Some implementations of the architecture also provide a powerful multiplier supporting both signed/unsigned multiplication and fractional format. See the “Instruction Set” section for a detailed description.

7.3 Status Register

The Status Register contains information about the result of the most recently executed arithmetic instruction. This information can be used for altering program flow in order to perform conditional operations. Note that the Status Register is updated after all ALU operations, as specified in the Instruction Set Reference. This will in many cases remove the need for using the dedicated compare instructions, resulting in faster and more compact code.

The Status Register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt. This must be handled by software.

7.3.1 SREG – AVR Status Register

The AVR Status Register – SREG – is defined as:

Bit	7	6	5	4	3	2	1	0	
0x3F (0x5F)	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – I: Global Interrupt Enable**

The Global Interrupt Enable bit must be set for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the Global Interrupt Enable Register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts. The I-bit can also be set and cleared by the application with the SEI and CLI instructions, as described in the instruction set reference.

- **Bit 6 – T: Bit Copy Storage**

The Bit Copy instructions BLD (Bit Load) and BST (Bit Store) use the T-bit as source or destination for the operated bit. A bit from a register in the Register File can be copied into T by the BST instruction, and a bit in T can be copied into a bit in a register in the Register File by the BLD instruction.

- **Bit 5 – H: Half Carry Flag**

The Half Carry Flag H indicates a Half Carry in some arithmetic operations. Half Carry Is useful in BCD arithmetic. See the “Instruction Set Description” for detailed information.

- **Bit 4 – S: Sign Bit, $S = N \oplus V$**

The S-bit is always an exclusive or between the Negative Flag N and the Two’s Complement Overflow Flag V. See the “Instruction Set Description” for detailed information.

- **Bit 3 – V: Two’s Complement Overflow Flag**

The Two’s Complement Overflow Flag V supports two’s complement arithmetic. See the “Instruction Set Description” for detailed information.

- **Bit 2 – N: Negative Flag**

The Negative Flag N indicates a negative result in an arithmetic or logic operation. See the “Instruction Set Description” for detailed information.

- **Bit 1 – Z: Zero Flag**

The Zero Flag Z indicates a zero result in an arithmetic or logic operation. See the “Instruction Set Description” for detailed information.

- **Bit 0 – C: Carry Flag**

The Carry Flag C indicates a carry in an arithmetic or logic operation. See the “Instruction Set Description” for detailed information.

7.4 General Purpose Register File

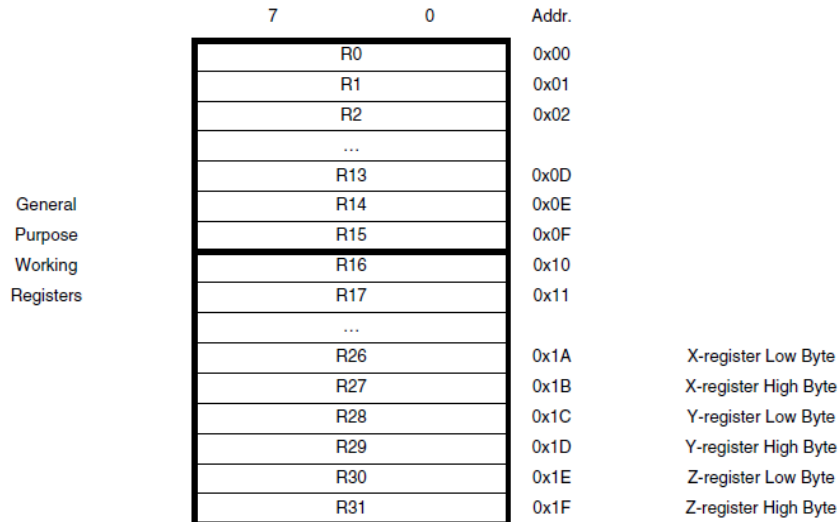
The Register File is optimized for the AVR Enhanced RISC instruction set. In order to achieve the required performance and flexibility, the following input/output schemes are supported by the Register File:

- One 8-bit output operand and one 8-bit result input
- Two 8-bit output operands and one 8-bit result input
- Two 8-bit output operands and one 16-bit result input

- One 16-bit output operand and one 16-bit result input

Figure 7-2 shows the structure of the 32 general purpose working registers in the CPU.

Figure 7-2. AVR CPU General Purpose Working Registers



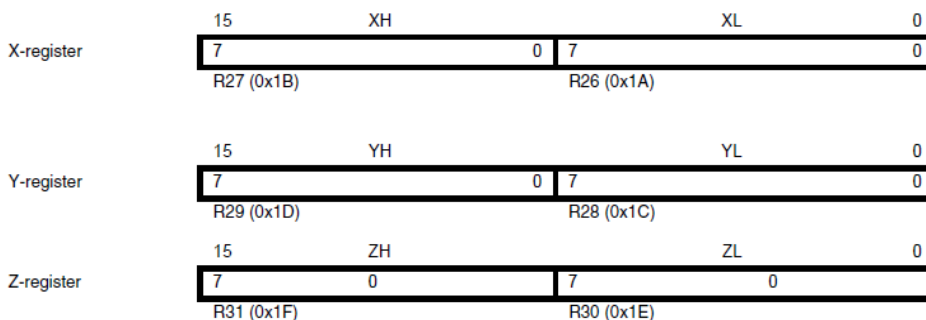
Most of the instructions operating on the Register File have direct access to all registers, and most of them are single cycle instructions.

As shown in Figure 7-2, each register is also assigned a data memory address, mapping them directly into the first 32 locations of the user Data Space. Although not being physically implemented as SRAM locations, this memory organization provides great flexibility in access of the registers, as the X-, Y- and Z-pointer registers can be set to index any register in the file.

7.4.1 The X-register, Y-register, and Z-register

The registers R26...R31 have some added functions to their general purpose usage. These registers are 16-bit address pointers for indirect addressing of the data space. The three indirect address registers X, Y, and Z are defined as described in Figure 7-3.

Figure 7-3. The X-, Y-, and Z-registers



In the different addressing modes these address registers have functions as fixed displacement, automatic increment, and automatic decrement (see the instruction set reference for details).

7.5 Stack Pointer

The Stack is mainly used for storing temporary data, for storing local variables and for storing return addresses after interrupts and subroutine calls. Note that the Stack is implemented as growing from higher to lower memory locations. The Stack Pointer Register always points to the top of the Stack. The Stack Pointer points to the data SRAM Stack area where the Subroutine and Interrupt Stacks are located. A Stack PUSH command will decrease the Stack Pointer.

The Stack in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. Initial Stack Pointer value equals the last address of the internal SRAM and the Stack Pointer must be set to point above start of the SRAM, see [Table 8-3 on page 18](#).

See [Table 7-1](#) for Stack Pointer details.

Table 7-1. Stack Pointer instructions

Instruction	Stack pointer	Description
PUSH	Decremented by 1	Data is pushed onto the stack
CALL ICALL RCALL	Decremented by 2	Return address is pushed onto the stack with a subroutine call or interrupt
POP	Incremented by 1	Data is popped from the stack
RET RETI	Incremented by 2	Return address is popped from the stack with return from subroutine or return from interrupt

The AVR Stack Pointer is implemented as two 8-bit registers in the I/O space. The number of bits actually used is implementation dependent. Note that the data space in some implementations of the AVR architecture is so small that only SPL is needed. In this case, the SPH Register will not be present.

7.5.1 SPH and SPL – Stack Pointer High and Stack Pointer Low Register

Bit	15	14	13	12	11	10	9	8	
0x3E (0x5E)	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SPH
0x3D (0x5D)	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	

7.6 Instruction Execution Timing

This section describes the general access timing concepts for instruction execution. The AVR CPU is driven by the CPU clock clk_{CPU} , directly generated from the selected clock source for the chip. No internal clock division is used.

Figure 7-4 shows the parallel instruction fetches and instruction executions enabled by the Harvard architecture and the fast-access Register File concept. This is the basic pipelining concept to obtain up to 1 MIPS per MHz with the corresponding unique results for functions per cost, functions per clocks, and functions per power-unit.

Figure 7-4. The Parallel Instruction Fetches and Instruction Executions

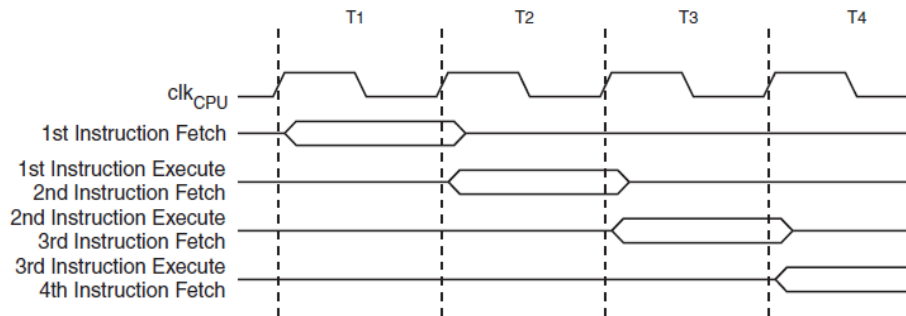
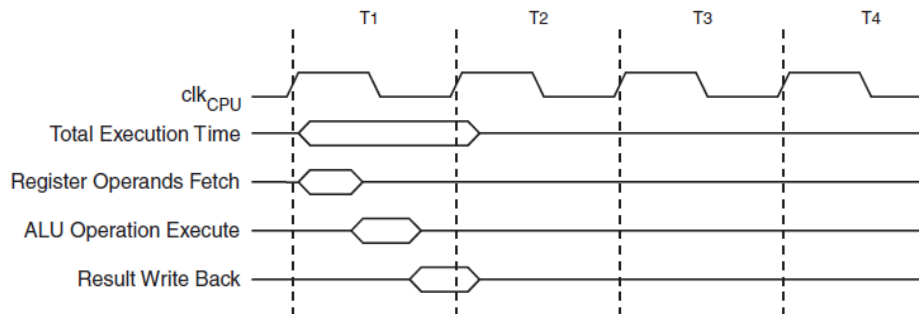


Figure 7-5 shows the internal timing concept for the Register File. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.

Figure 7-5. Single Cycle ALU Operation



7.7 Reset and Interrupt Handling

The AVR provides several different interrupt sources. These interrupts and the separate Reset Vector each have a separate program vector in the program memory space. All interrupts are assigned individual enable bits which must be written logic one together with the Global Interrupt Enable bit in the Status Register in order to enable the interrupt. Depending on the Program Counter value, interrupts may be automatically disabled when Boot Lock bits BLB02 or BLB12 are programmed. This feature improves software security. See the section "Memory Programming" on page 285 for details.

The lowest addresses in the program memory space are by default defined as the Reset and Interrupt Vectors. The complete list of vectors is shown in "Interrupts" on page 57. The list also determines the priority levels of the different interrupts. The lower the address the higher is the priority level. RESET has the highest priority, and next is INT0 – the External Interrupt Request 0. The Interrupt Vectors can be moved to the start of the Boot Flash section by setting the IVSEL bit in the MCU Control Register (MCUCR). Refer to "Interrupts" on page 57 for more information. The Reset Vector can also be moved to the start of the Boot Flash section by programming the BOOTRST Fuse, see "Boot Loader Support – Read-While-Write Self-Programming" on page 269.

When an interrupt occurs, the Global Interrupt Enable I-bit is cleared and all interrupts are disabled. The user software can write logic one to the I-bit to enable nested interrupts. All enabled interrupts can then interrupt the current interrupt routine. The I-bit is automatically set when a Return from Interrupt instruction – RETI – is executed.

There are basically two types of interrupts. The first type is triggered by an event that sets the Interrupt Flag. For these interrupts, the Program Counter is vectored to the actual Interrupt Vector in order to execute the interrupt handling routine, and hardware clears the corresponding Interrupt Flag. Interrupt Flags can also be cleared by writing a logic one to the flag bit position(s) to be cleared. If an interrupt condition occurs while the corresponding interrupt enable bit is cleared, the Interrupt Flag will be set and remembered until the interrupt is enabled, or the flag is cleared by software. Similarly, if one or more interrupt conditions occur while the Global Interrupt Enable bit is cleared, the corresponding Interrupt Flag(s) will be set and remembered until the Global Interrupt Enable bit is set, and will then be executed by order of priority.

The second type of interrupts will trigger as long as the interrupt condition is present. These interrupts do not necessarily have Interrupt Flags. If the interrupt condition disappears before the interrupt is enabled, the interrupt will not be triggered.

When the AVR exits from an interrupt, it will always return to the main program and execute one more instruction before any pending interrupt is served.

Note that the Status Register is not automatically stored when entering an interrupt routine, nor restored when returning from an interrupt routine. This must be handled by software.

When using the CLI instruction to disable interrupts, the interrupts will be immediately disabled. No interrupt will be executed after the CLI instruction, even if it occurs simultaneously with the CLI instruction. The following example shows how this can be used to avoid interrupts during the timed EEPROM write sequence.

Assembly Code Example
<pre>in r16, SREG ; store SREG value cli ; disable interrupts during timed sequence sbi EECR, EEMPE ; start EEPROM write sbi EECR, EEPE out SREG, r16 ; restore SREG value (I-bit)</pre>
C Code Example
<pre>char cSREG; cSREG = SREG; /* store SREG value */ /* disable interrupts during timed sequence */ _cli(); EECR = (1<<EEMPE); /* start EEPROM write */ EECR = (1<<EEPE); SREG = cSREG; /* restore SREG value (I-bit) */</pre>

When using the SEI instruction to enable interrupts, the instruction following SEI will be executed before any pending interrupts, as shown in this example.

Assembly Code Example

```
sei ; set Global Interrupt Enable  
sleep; enter sleep, waiting for interrupt  
; note: will enter sleep before any pending interrupt(s)
```

C Code Example

```
__enable_interrupt(); /* set Global Interrupt Enable */  
__sleep(); /* enter sleep, waiting for interrupt */  
/* note: will enter sleep before any pending interrupt(s) */
```

7.7.1 Interrupt Response Time

The interrupt execution response for all the enabled AVR interrupts is four clock cycles minimum. After four clock cycles the program vector address for the actual interrupt handling routine is executed. During this four clock cycle period, the Program Counter is pushed onto the Stack. The vector is normally a jump to the interrupt routine, and this jump takes three clock cycles. If an interrupt occurs during execution of a multi-cycle instruction, this instruction is completed before the interrupt is served. If an interrupt occurs when the MCU is in sleep mode, the interrupt execution response time is increased by four clock cycles. This increase comes in addition to the start-up time from the selected sleep mode.

A return from an interrupt handling routine takes four clock cycles. During these four clock cycles, the Program Counter (two bytes) is popped back from the Stack, the Stack Pointer is incremented by two, and the I-bit in SREG is set.



InvenSense Inc.
1197 Borregas Ave, Sunnyvale, CA 94089 U.S.A.
Tel: +1 (408) 988-7339 Fax: +1 (408) 988-8104
Website: www.invensense.com

Document Number: PS-MPU-6000A-00
Revision: 3.4
Release Date: 08/19/2013

MPU-6000 and MPU-6050 Product Specification Revision 3.4

5 Features

5.1 Gyroscope Features

The triple-axis MEMS gyroscope in the MPU-60X0 includes a wide range of features:

- Digital-output X-, Y-, and Z-Axis angular rate sensors (gyroscopes) with a user-programmable full-scale range of ± 250 , ± 500 , ± 1000 , and $\pm 2000^\circ/\text{sec}$
- External sync signal connected to the FSYNC pin supports image, video and GPS synchronization
- Integrated 16-bit ADCs enable simultaneous sampling of gyros
- Enhanced bias and sensitivity temperature stability reduces the need for user calibration
- Improved low-frequency noise performance
- Digitally-programmable low-pass filter
- Gyroscope operating current: 3.6mA
- Standby current: 5 μ A
- Factory calibrated sensitivity scale factor
- User self-test

5.2 Accelerometer Features

The triple-axis MEMS accelerometer in MPU-60X0 includes a wide range of features:

- Digital-output triple-axis accelerometer with a programmable full scale range of $\pm 2g$, $\pm 4g$, $\pm 8g$ and $\pm 16g$
- Integrated 16-bit ADCs enable simultaneous sampling of accelerometers while requiring no external multiplexer
- Accelerometer normal operating current: 500 μ A
- Low power accelerometer mode current: 10 μ A at 1.25Hz, 20 μ A at 5Hz, 60 μ A at 20Hz, 110 μ A at 40Hz
- Orientation detection and signaling
- Tap detection
- User-programmable interrupts
- High-G interrupt
- User self-test

5.3 Additional Features

The MPU-60X0 includes the following additional features:

- 9-Axis MotionFusion by the on-chip Digital Motion Processor (DMP)
- Auxiliary master I²C bus for reading data from external sensors (e.g., magnetometer)
- 3.9mA operating current when all 6 motion sensing axes and the DMP are enabled
- VDD supply voltage range of 2.375V-3.46V
- Flexible VLOGIC reference voltage supports multiple I²C interface voltages (MPU-6050 only)
- Smallest and thinnest QFN package for portable devices: 4x4x0.9mm
- Minimal cross-axis sensitivity between the accelerometer and gyroscope axes
- 1024 byte FIFO buffer reduces power consumption by allowing host processor to read the data in bursts and then go into a low-power mode as the MPU collects more data
- Digital-output temperature sensor
- User-programmable digital filters for gyroscope, accelerometer, and temp sensor
- 10,000 g shock tolerant
- 400kHz Fast Mode I²C for communicating with all registers
- 1MHz SPI serial interface for communicating with all registers (MPU-6000 only)
- 20MHz SPI serial interface for reading sensor and interrupt registers (MPU-6000 only)

- MEMS structure hermetically sealed and bonded at wafer level
- RoHS and Green compliant

5.4 MotionProcessing

- Internal Digital Motion Processing™ (DMP™) engine supports 3D MotionProcessing and gesture recognition algorithms
- The MPU-60X0 collects gyroscope and accelerometer data while synchronizing data sampling at a user defined rate. The total dataset obtained by the MPU-60X0 includes 3-Axis gyroscope data, 3-Axis accelerometer data, and temperature data. The MPU's calculated output to the system processor can also include heading data from a digital 3-axis third party magnetometer.
- The FIFO buffers the complete data set, reducing timing requirements on the system processor by allowing the processor burst read the FIFO data. After burst reading the FIFO data, the system processor can save power by entering a low-power sleep mode while the MPU collects more data.
- Programmable interrupt supports features such as gesture recognition, panning, zooming, scrolling, tap detection, and shake detection
- Digitally-programmable low-pass filters
- Low-power pedometer functionality allows the host processor to sleep while the DMP maintains the step count.

5.5 Clocking

- On-chip timing generator $\pm 1\%$ frequency variation over full temperature range
- Optional external clock inputs of 32.768kHz or 19.2MHz

6.9 Absolute Maximum Ratings

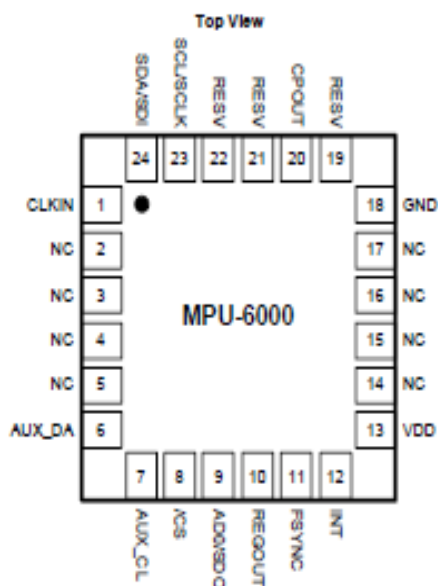
Stress above those listed as "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only and functional operation of the device at these conditions is not implied. Exposure to the absolute maximum ratings conditions for extended periods may affect device reliability.

Parameter	Rating
Supply Voltage, VDD	-0.5V to +6V
VLOGIC Input Voltage Level (MPU-6050)	-0.5V to VDD + 0.5V
REGOUT	-0.5V to 2V
Input Voltage Level (CLKIN, AUX_DA, AD0, FSYNC, INT, SCL, SDA)	-0.5V to VDD + 0.5V
CPOUT (2.5V ≤ VDD ≤ 3.6V)	-0.5V to 30V
Acceleration (Any Axis, unpowered)	10,000g for 0.2ms
Operating Temperature Range	-40°C to +105°C
Storage Temperature Range	-40°C to +125°C
Electrostatic Discharge (ESD) Protection	2kV (HBM); 250V (MM)
Latch-up	JEDEC Class II (2), 125°C ±100mA

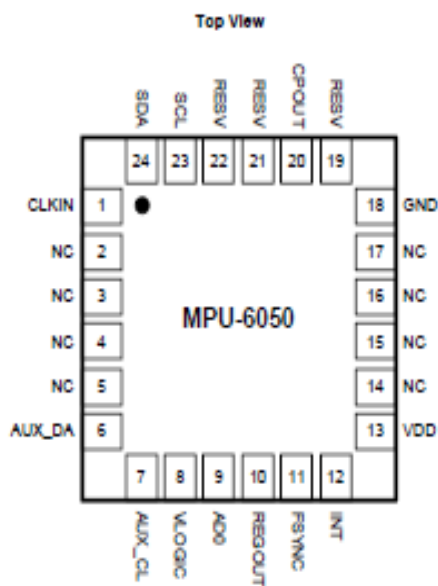
7 Applications Information

7.1 Pin Out and Signal Description

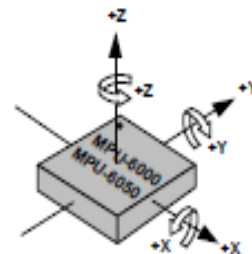
Pin Number	MPU-6000	MPU-6050	Pin Name	Pin Description
1	Y	Y	CLKIN	Optional external reference clock input. Connect to GND if unused.
6	Y	Y	AUX_DA	I ² C master serial data, for connecting to external sensors
7	Y	Y	AUX_CL	I ² C Master serial clock, for connecting to external sensors
8	Y		/CS	SPI chip select (0=SPI mode)
8		Y	VLOGIC	Digital I/O supply voltage
9	Y		AD0 / SDO	I ² C Slave Address LSB (AD0); SPI serial data output (SDO)
9		Y	AD0	I ² C Slave Address LSB (AD0)
10	Y	Y	REGOUT	Regulator filter capacitor connection
11	Y	Y	FSYNC	Frame synchronization digital input. Connect to GND if unused.
12	Y	Y	INT	Interrupt digital output (totem pole or open-drain)
13	Y	Y	VDD	Power supply voltage and Digital I/O supply voltage
18	Y	Y	GND	Power supply ground
19, 21	Y	Y	RESV	Reserved. Do not connect.
20	Y	Y	CPOUT	Charge pump capacitor connection
22	Y	Y	RESV	Reserved. Do not connect.
23	Y		SCL / SCLK	I ² C serial clock (SCL); SPI serial clock (SCLK)
23		Y	SCL	I ² C serial clock (SCL)
24	Y		SDA / SDI	I ² C serial data (SDA); SPI serial data input (SDI)
24		Y	SDA	I ² C serial data (SDA)
2, 3, 4, 5, 14, 15, 16, 17	Y	Y	NC	Not internally connected. May be used for PCB trace routing.



QFN Package
24-pin, 4mm x 4mm x 0.8mm

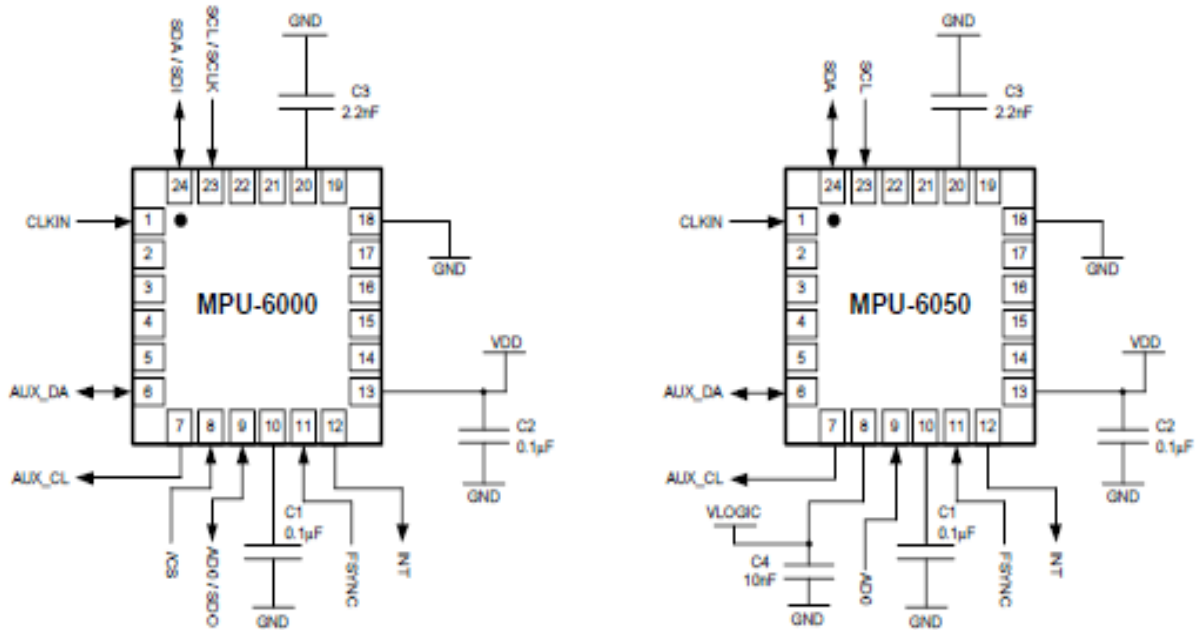


QFN Package
24-pin, 4mm x 4mm x 0.8mm



Orientation of Axes of Sensitivity and
Polarity of Rotation

7.2 Typical Operating Circuit



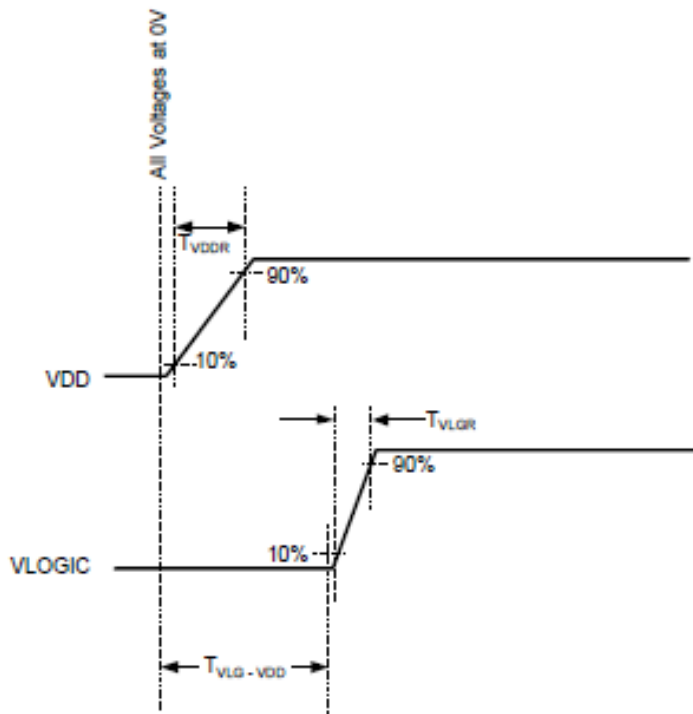
Typical Operating Circuits

7.3 Bill of Materials for External Components

Component	Label	Specification	Quantity
Regulator Filter Capacitor (Pin 10)	C1	Ceramic, X7R, 0.1µF ±10%, 2V	1
VDD Bypass Capacitor (Pin 13)	C2	Ceramic, X7R, 0.1µF ±10%, 4V	1
Charge Pump Capacitor (Pin 20)	C3	Ceramic, X7R, 2.2nF ±10%, 50V	1
VLOGIC Bypass Capacitor (Pin 8)	C4*	Ceramic, X7R, 10nF ±10%, 4V	1

* MPU-6050 Only.

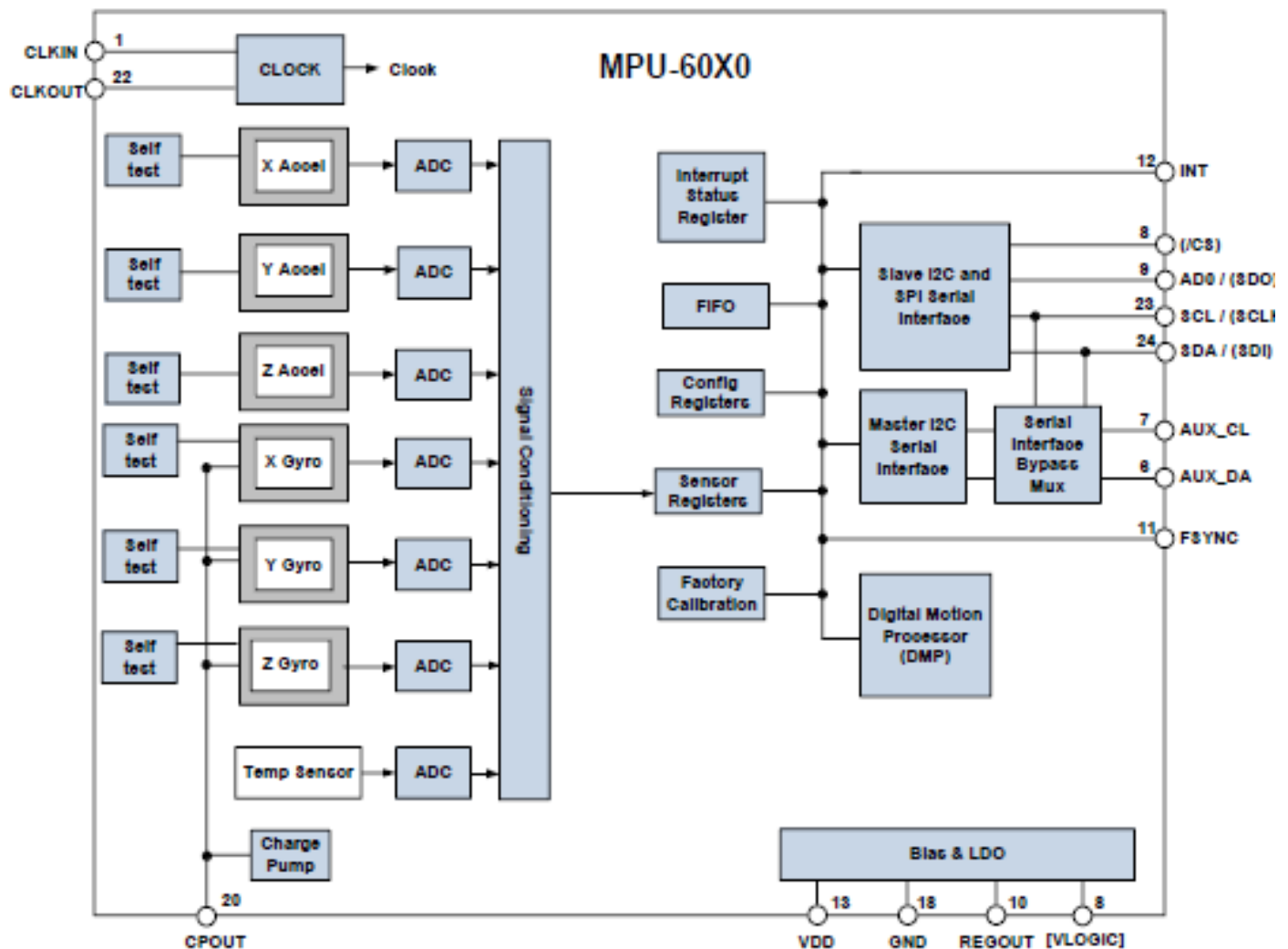
7.4 Recommended Power-on Procedure



Power-Up Sequencing

1. VLOGIC amplitude must always be $\leq VDD$ amplitude
2. T_{VDDR} is VDD rise time: Time for VDD to rise from 10% to 90% of its final value
3. T_{VDDR} is $\leq 100\text{ms}$
4. T_{VLGR} is VLOGIC rise time: Time for VLOGIC to rise from 10% to 90% of its final value
5. T_{VLGR} is $\leq 3\text{ms}$
6. $T_{VLG-VDD}$ is the delay from the start of VDD ramp to the start of VLOGIC rise
7. $T_{VLG-VDD}$ is ≥ 0
8. VDD and VLOGIC must be monotonic ramps

7.5 Block Diagram



Note: Pin names in round brackets () apply only to MPU-6000
Pin names in square brackets [] apply only to MPU-6050

7.6 Overview

The MPU-60X0 is comprised of the following key blocks and functions:

- Three-axis MEMS rate gyroscope sensor with 16-bit ADCs and signal conditioning
- Three-axis MEMS accelerometer sensor with 16-bit ADCs and signal conditioning
- Digital Motion Processor (DMP) engine
- Primary I²C and SPI (MPU-6000 only) serial communications interfaces
- Auxiliary I²C serial interface for 3rd party magnetometer & other sensors
- Clocking
- Sensor Data Registers
- FIFO
- Interrupts
- Digital-Output Temperature Sensor
- Gyroscope & Accelerometer Self-test
- Bias and LDO
- Charge Pump

7.7 Three-Axis MEMS Gyroscope with 16-bit ADCs and Signal Conditioning

The MPU-60X0 consists of three independent vibratory MEMS rate gyroscopes, which detect rotation about the X-, Y-, and Z- Axes. When the gyros are rotated about any of the sense axes, the Coriolis Effect causes a vibration that is detected by a capacitive pickoff. The resulting signal is amplified, demodulated, and filtered to produce a voltage that is proportional to the angular rate. This voltage is digitized using individual on-chip 16-bit Analog-to-Digital Converters (ADCs) to sample each axis. The full-scale range of the gyro sensors may be digitally programmed to ± 250 , ± 500 , ± 1000 , or ± 2000 degrees per second (dps). The ADC sample rate is programmable from 8,000 samples per second, down to 3.9 samples per second, and user-selectable low-pass filters enable a wide range of cut-off frequencies.

7.8 Three-Axis MEMS Accelerometer with 16-bit ADCs and Signal Conditioning

The MPU-60X0's 3-Axis accelerometer uses separate proof masses for each axis. Acceleration along a particular axis induces displacement on the corresponding proof mass, and capacitive sensors detect the displacement differentially. The MPU-60X0's architecture reduces the accelerometers' susceptibility to fabrication variations as well as to thermal drift. When the device is placed on a flat surface, it will measure 0g on the X- and Y-axes and +1g on the Z-axis. The accelerometers' scale factor is calibrated at the factory and is nominally independent of supply voltage. Each sensor has a dedicated sigma-delta ADC for providing digital outputs. The full scale range of the digital output can be adjusted to $\pm 2g$, $\pm 4g$, $\pm 8g$, or $\pm 16g$.

7.9 Digital Motion Processor

The embedded Digital Motion Processor (DMP) is located within the MPU-60X0 and offloads computation of motion processing algorithms from the host processor. The DMP acquires data from accelerometers, gyroscopes, and additional 3rd party sensors such as magnetometers, and processes the data. The resulting data can be read from the DMP's registers, or can be buffered in a FIFO. The DMP has access to one of the MPU's external pins, which can be used for generating interrupts.

The purpose of the DMP is to offload both timing requirements and processing power from the host processor. Typically, motion processing algorithms should be run at a high rate, often around 200Hz, in order to provide accurate results with low latency. This is required even if the application updates at a much lower rate; for example, a low power user interface may update as slowly as 5Hz, but the motion processing should still run at 200Hz. The DMP can be used as a tool in order to minimize power, simplify timing, simplify the software architecture, and save valuable MIPS on the host processor for use in the application.

7.10 Primary I²C and SPI Serial Communications Interfaces

The MPU-60X0 communicates to a system processor using either a SPI (MPU-6000 only) or an I²C serial interface. The MPU-60X0 always acts as a slave when communicating to the system processor. The LSB of the of the I²C slave address is set by pin 9 (AD0).

The logic levels for communications between the MPU-60X0 and its master are as follows:

- MPU-6000: The logic level for communications with the master is set by the voltage on VDD
- MPU-6050: The logic level for communications with the master is set by the voltage on VLOGIC

For further information regarding the logic levels of the MPU-6050, please refer to Section 10.

7.12 Self-Test

Please refer to the MPU-6000/MPU-6050 Register Map and Register Descriptions document for more details on self test.

Self-test allows for the testing of the mechanical and electrical portions of the sensors. The self-test for each measurement axis can be activated by means of the gyroscope and accelerometer self-test registers (registers 13 to 16).

When self-test is activated, the electronics cause the sensors to be actuated and produce an output signal. The output signal is used to observe the self-test response.

The self-test response is defined as follows:

$$\text{Self-test response} = \text{Sensor output with self-test enabled} - \text{Sensor output without self-test enabled}$$

The self-test response for each accelerometer axis is defined in the accelerometer specification table (Section 6.2), while that for each gyroscope axis is defined in the gyroscope specification table (Section 6.1).

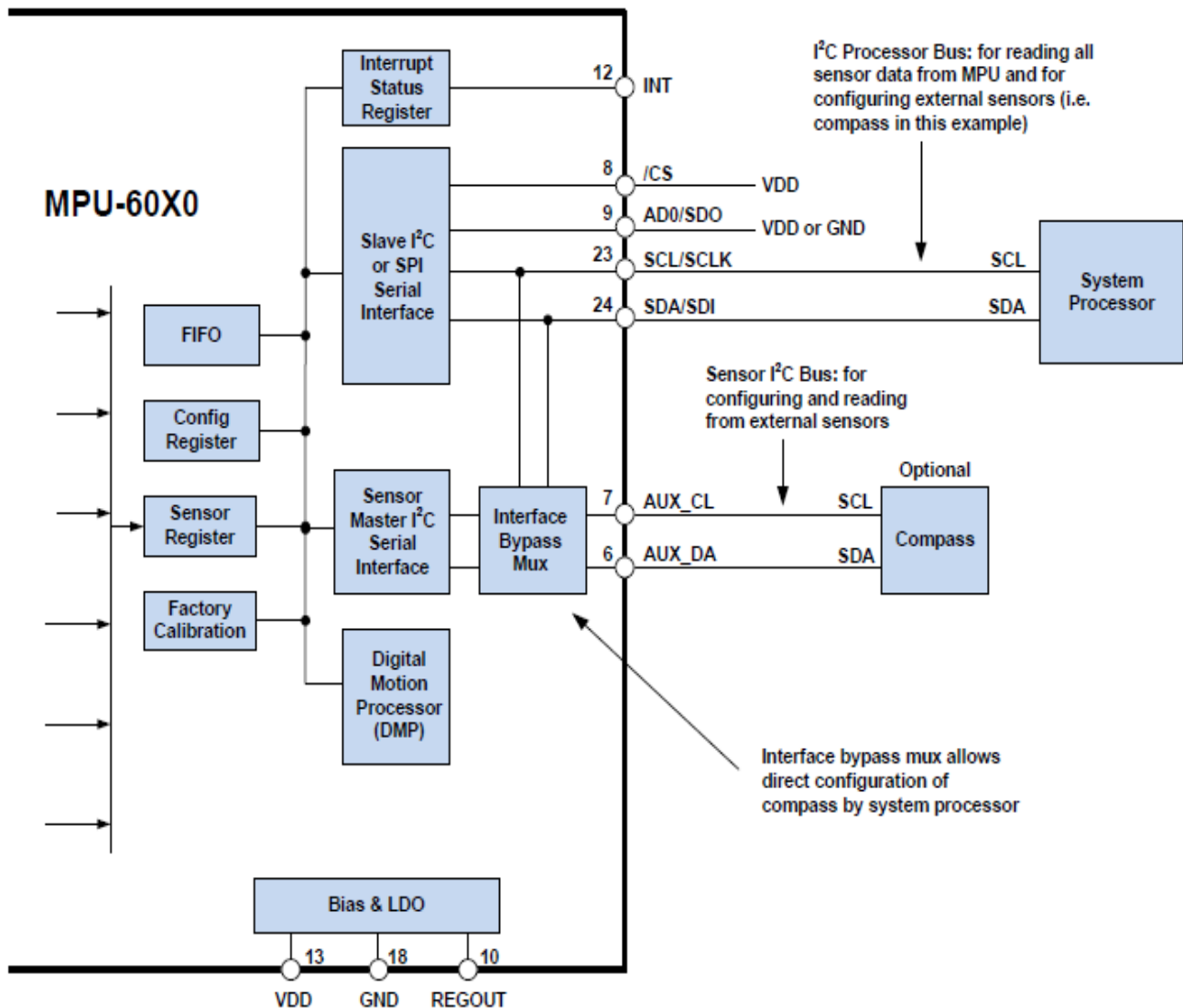
When the value of the self-test response is within the min/max limits of the product specification, the part has passed self test. When the self-test response exceeds the min/max values, the part is deemed to have failed self-test. Code for operating self test code is included within the MotionApps software provided by InvenSense.

7.13 MPU-60X0 Solution for 9-axis Sensor Fusion Using I²C Interface

In the figure below, the system processor is an I²C master to the MPU-60X0. In addition, the MPU-60X0 is an I²C master to the optional external compass sensor. The MPU-60X0 has limited capabilities as an I²C Master, and depends on the system processor to manage the initial configuration of any auxiliary sensors. The MPU-60X0 has an interface bypass multiplexer, which connects the system processor I²C bus pins 23 and 24 (SDA and SCL) directly to the auxiliary sensor I²C bus pins 6 and 7 (AUX_DA and AUX_CL).

Once the auxiliary sensors have been configured by the system processor, the interface bypass multiplexer should be disabled so that the MPU-60X0 auxiliary I²C master can take control of the sensor I²C bus and gather data from the auxiliary sensors.

For further information regarding I²C master control, please refer to Section 10.



7.14 MPU-6000 Using SPI Interface

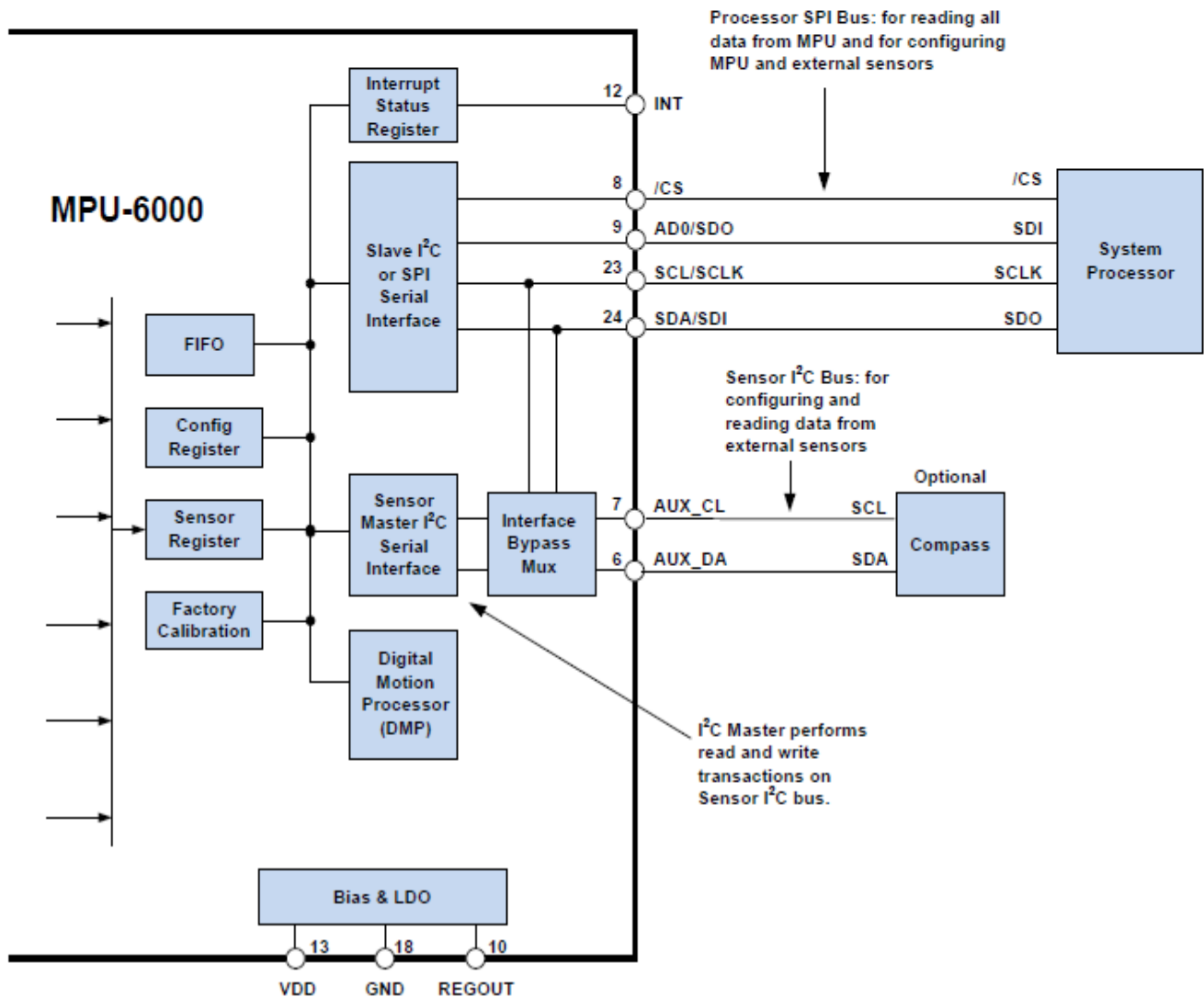
In the figure below, the system processor is an SPI master to the MPU-6000. Pins 8, 9, 23, and 24 are used to support the /CS, SDO, SCLK, and SDI signals for SPI communications. Because these SPI pins are shared with the I²C slave pins (9, 23 and 24), the system processor cannot access the auxiliary I²C bus through the interface bypass multiplexer, which connects the processor I²C interface pins to the sensor I²C interface pins.

Since the MPU-6000 has limited capabilities as an I²C Master, and depends on the system processor to manage the initial configuration of any auxiliary sensors, another method must be used for programming the sensors on the auxiliary sensor I²C bus pins 6 and 7 (AUX_DA and AUX_CL).

When using SPI communications between the MPU-6000 and the system processor, configuration of devices on the auxiliary I²C sensor bus can be achieved by using I²C Slaves 0-4 to perform read and write transactions on any device and register on the auxiliary I²C bus. The I²C Slave 4 interface can be used to perform only single byte read and write transactions.

Once the external sensors have been configured, the MPU-6000 can perform single or multi-byte reads using the sensor I²C bus. The read results from the Slave 0-3 controllers can be written to the FIFO buffer as well as to the external sensor registers.

For further information regarding the control of the MPU-60X0's auxiliary I²C interface, please refer to the MPU-6000/MPU-6050 Register Map and Register Descriptions document.



7.15 Internal Clock Generation

The MPU-60X0 has a flexible clocking scheme, allowing a variety of internal or external clock sources to be used for the internal synchronous circuitry. This synchronous circuitry includes the signal conditioning and ADCs, the DMP, and various control circuits and registers. An on-chip PLL provides flexibility in the allowable inputs for generating this clock.

Allowable internal sources for generating the internal clock are:

- An internal relaxation oscillator
- Any of the X, Y, or Z gyros (MEMS oscillators with a variation of $\pm 1\%$ over temperature)

Allowable external clocking sources are:

- 32.768kHz square wave
- 19.2MHz square wave

Selection of the source for generating the internal synchronous clock depends on the availability of external sources and the requirements for power consumption and clock accuracy. These requirements will most likely vary by mode of operation. For example, in one mode, where the biggest concern is power consumption, the user may wish to operate the Digital Motion Processor of the MPU-60X0 to process accelerometer data, while keeping the gyros off. In this case, the internal relaxation oscillator is a good clock choice. However, in another mode, where the gyros are active, selecting the gyros as the clock source provides for a more accurate clock source.

Clock accuracy is important, since timing errors directly affect the distance and angle calculations performed by the Digital Motion Processor (and by extension, by any processor).

There are also start-up conditions to consider. When the MPU-60X0 first starts up, the device uses its internal clock until programmed to operate from another source. This allows the user, for example, to wait for the MEMS oscillators to stabilize before they are selected as the clock source.

7.16 Sensor Data Registers

The sensor data registers contain the latest gyro, accelerometer, auxiliary sensor, and temperature measurement data. They are read-only registers, and are accessed via the serial interface. Data from these registers may be read anytime. However, the interrupt function may be used to determine when new data is available.

For a table of interrupt sources please refer to Section 8.

7.17 FIFO

The MPU-60X0 contains a 1024-byte FIFO register that is accessible via the Serial Interface. The FIFO configuration register determines which data is written into the FIFO. Possible choices include gyro data, accelerometer data, temperature readings, auxiliary sensor readings, and FSYNC input. A FIFO counter keeps track of how many bytes of valid data are contained in the FIFO. The FIFO register supports burst reads. The interrupt function may be used to determine when new data is available.

For further information regarding the FIFO, please refer to the MPU-6000/MPU-6050 Register Map and Register Descriptions document.

7.18 Interrupts

Interrupt functionality is configured via the Interrupt Configuration register. Items that are configurable include the INT pin configuration, the interrupt latching and clearing method, and triggers for the interrupt. Items that can trigger an interrupt are (1) Clock generator locked to new reference oscillator (used when switching clock

sources); (2) new data is available to be read (from the FIFO and Data registers); (3) accelerometer event interrupts; and (4) the MPU-60X0 did not receive an acknowledge from an auxiliary sensor on the secondary I²C bus. The interrupt status can be read from the Interrupt Status register.

For further information regarding interrupts, please refer to the MPU-60X0 Register Map and Register Descriptions document.

For information regarding the MPU-60X0's accelerometer event interrupts, please refer to Section 8.

7.19 Digital-Output Temperature Sensor

An on-chip temperature sensor and ADC are used to measure the MPU-60X0 die temperature. The readings from the ADC can be read from the FIFO or the Sensor Data registers.

7.20 Bias and LDO

The bias and LDO section generates the internal supply and the reference voltages and currents required by the MPU-60X0. Its two inputs are an unregulated VDD of 2.375 to 3.46V and a VLOGIC logic reference supply voltage of 1.71V to VDD (MPU-6050 only). The LDO output is bypassed by a capacitor at REGOUT. For further details on the capacitor, please refer to the Bill of Materials for External Components (Section 7.3).

7.21 Charge Pump

An on-board charge pump generates the high voltage required for the MEMS oscillators. Its output is bypassed by a capacitor at CPOUT. For further details on the capacitor, please refer to the Bill of Materials for External Components (Section 7.3).