



industriales
etsii

Escuela Técnica
Superior
de Ingeniería
Industrial

UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Escuela Técnica Superior de Ingeniería Industrial

Diseño de un Vehículo Aéreo no Tripulado multipropósito

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA ELECTRÓNICA
INDUSTRIAL Y AUTOMÁTICA

Autor: Alicia Heras Almagro
Director: Manuel M. Sánchez Nieto



Universidad
Politécnica
de Cartagena

Cartagena, 9 de octubre de 2017

Índice

Agradecimientos.	<u>4</u>
Introducción.	<u>5</u>
0.1 Justificación académica.	<u>5</u>
0.2 Elección del proyecto.	<u>6</u>
0.3 Objetivos.	<u>7</u>
0.3.1 Documentación.	<u>7</u>
0.3.2 Selección de componentes.	<u>7</u>
0.3.3 Control	<u>7</u>
0.3.4 Sensores.	<u>7</u>
0.3.5 Módulo de comunicación.	<u>7</u>
0.3.6 App.	<u>7</u>
Documentación.	<u>8</u>
1.1 Historia.	<u>8</u>
1.1.1 Historia militar.	<u>8</u>
1.1.2 Historia civil.	<u>10</u>
1.2 Aplicaciones actuales.	<u>11</u>
1.2.1 Aplicaciones al control de la calidad del aire.	<u>11</u>
1.2.2 Aplicaciones cartográficas.	<u>11</u>
1.2.3 Aplicaciones a la prospección y explotación de recursos minerales.	<u>12</u>
1.2.4 Aplicaciones hidrológicas.	<u>12</u>
1.2.5 Aplicaciones agrícolas.	<u>12</u>
1.2.6 Aplicaciones a la extinción nocturna de incendios forestales: «Proyecto Nitrofirex».	<u>12</u>
1.2.7 Aplicaciones al control de obras y evaluación de impactos.	<u>14</u>
1.2.8 Aplicaciones al mantenimiento de líneas eléctricas.	<u>14</u>
1.2.9 Búsqueda de personas desaparecidas.	<u>14</u>
1.2.10 Cooperación de VANTs.	<u>15</u>
1.3 Normativa.	<u>15</u>
1.3.1 Ley de Navegación Aérea.	<u>15</u>
1.3.2 Ley 18/2014.	<u>15</u>
1.3.3 Real Decreto 552/2014.	<u>15</u>
1.3.4 Vuelo en recintos cerrados.	<u>16</u>
Selección de componentes.	<u>17</u>
2.1 Control.	<u>18</u>
2.2 Sensores.	<u>19</u>
2.2.1 IMU.	<u>19</u>
2.2.2 Fotorresistencia.	<u>19</u>
2.3 Actuadores.	<u>20</u>
2.3.1 Motores.	<u>20</u>
2.3.2 Rotores.	<u>20</u>

2.3.3 LEDs.....	<u>21</u>
2.4 Módulo de comunicación.....	<u>22</u>
2.5 Placa soporte y de conexionado.....	<u>22</u>
2.6 Batería.....	<u>23</u>
2.7 Estructura.....	<u>24</u>
Control.....	<u>26</u>
3.1 Arduino.....	<u>26</u>
3.2 App.....	<u>27</u>
3.2.1 App Inventor.....	<u>27</u>
3.2.1.1 App Inventor Designer.....	<u>28</u>
3.2.1.2 App Inventor Blocks Editor.....	<u>28</u>
3.2.2 ArdDrone.....	<u>29</u>
3.2.2.1 Conexión con dispositivo.....	<u>29</u>
3.2.2.2 Interfaz.....	<u>31</u>
3.2.2.3 Programación de bloques.....	<u>34</u>
Sensores.....	<u>36</u>
4.1 MPU-6050. Funcionamiento.....	<u>36</u>
4.1.1 Sistemas de coordenadas.....	<u>37</u>
4.1.2 Acelerómetros.....	<u>38</u>
4.1.3 Giróscopos.....	<u>39</u>
4.1.4 Error en las medidas.....	<u>40</u>
4.1.5 Protocolo I ² C.....	<u>41</u>
4.1.6 Conexión.....	<u>43</u>
4.1.7 Implementación en Arduino.....	<u>44</u>
4.2 Fotorresistencia (LDR).....	<u>47</u>
4.2.1 Conexión.....	<u>47</u>
4.2.2 Cálculos.....	<u>48</u>
4.2.3 Código.....	<u>49</u>
Actuadores.....	<u>50</u>
5.1 Motores.....	<u>50</u>
5.1.1 Motores DC brushless.....	<u>50</u>
5.1.2 Motores DC con escobillas. L293D.....	<u>51</u>
5.1.3 Conexión.....	<u>53</u>
5.2 Diodos LED.....	<u>54</u>
5.2.1 Conexión.....	<u>56</u>
5.2.2 Cálculos.....	<u>56</u>
5.2.3 Código.....	<u>56</u>
Comunicación.....	<u>57</u>
6.1 Radio-control.....	<u>57</u>
6.2 Hyperion.....	<u>57</u>
6.3 Satélite.....	<u>58</u>
6.4 Sin comunicación.....	<u>58</u>
6.5 Bluetooth.....	<u>59</u>
6.5.1 Conexión.....	<u>60</u>
6.5.2 Implementación en Arduino.....	<u>61</u>

Consumo.	<u>63</u>
7.1 Arduino.	<u>63</u>
7.2 Módulo Bluetooth.	<u>64</u>
7.3 IMU.	<u>64</u>
7.4 Fotorresistencia.	<u>64</u>
7.5 Diodos LED.	<u>64</u>
7.6 Conclusión.	<u>65</u>
Ensamblaje y pruebas.	<u>66</u>
8.1 Peso.	<u>66</u>
8.2 Ensamblaje.	<u>66</u>
8.3 Esquema.	<u>70</u>
8.4 Conexionado.	<u>71</u>
8.5 Pruebas.	<u>72</u>
Conclusiones.	<u>73</u>
Trabajos futuros.	<u>74</u>
Bibliografía.	<u>75</u>

Agradecimientos

A Manuel M. Sánchez, por su disponibilidad y ayuda

A mi madre, por su apoyo incondicional, por enseñarme a tomarme las cosas de otra manera y por recordarme que es necesario “afilarse el hacha”

A mi padre, porque se puede ser experto sin un título universitario

A mis amigos, porque las distracciones eventuales son imprescindibles

Capítulo 0

Introducción

Nos referimos a ellos mediante distintos nombres, entre los que se encuentran *Unmanned Aerial Vehicles* (uAv), minidrones o microdrones (en función del tamaño y peso). También son denominados *Unmanned Aircraft Systems* (UAS), *Unmanned Aerial Systems* (UAS), *Remotely Piloted Aircraft Systems* (RPAS) y *Unmanned Air Vehicles* (UAV). Pero sin duda, la denominación mundialmente conocida a día de hoy es la de *drone*.

El documento <<Joint Publication 1-02, Department of Defense Dictionary>>, editado por el Ministerio de Defensa de los Estados Unidos define UAV (o VANT) como: «*Un vehículo aéreo motorizado que no lleva a bordo a un operador humano, utiliza las fuerzas aerodinámicas para generar la sustentación, puede volar autónomamente o ser tripulado de forma remota, que puede ser fungible o recuperable, y que puede transportar una carga de pago letal o no. No se consideran UAV a los misiles balísticos o semibalísticos, misiles crucero y proyectiles de artillería*».

Hoy en día, los drones se pueden considerar como robots no antropomorfos con gran diversidad de características, pudiendo variar en tamaño, forma, autonomía y propósito en función de los sensores que incorpore. Su control puede llevarse a cabo de forma manual o automática, o mediante una combinación de ambas.

Casi a diario encontramos nuevos dispositivos con aplicaciones de lo más dispares, lo que nos da una idea del potencial de estos artilugios en el futuro.

0.1 Justificación académica

Según lo dispuesto en el Real Decreto 1393/2007, de 29 de octubre (BOE número 206 del 30 de octubre de 2007), por el que se establece la ordenación de las enseñanzas universitarias oficiales, capítulo III: *Enseñanzas universitarias oficiales de Grado*, y por el Artículo 12: *Directrices para el diseño de títulos de Graduado*.

1. Los planes de estudios conducentes a la obtención del título de Graduado serán elaborados por las universidades y verificados de acuerdo con lo establecido en el presente real decreto.
2. Los planes de estudios tendrán 240 créditos, que contendrán toda la formación teórica y práctica que el estudiante deba adquirir: aspectos básicos de la rama de conocimiento, materias obligatorias u optativas, seminarios, practicas externas, trabajos dirigidos, trabajo de fin de Grado u otras actividades formativas. En los supuestos en que ello venga determinado por normas de derecho comunitario, el Gobierno, previo informe del Consejo de Universidades, podrá asignar un numero mayor de créditos.
3. Estas enseñanzas concluirán con la elaboración y defensa de un trabajo de fin de Grado.

0.2 Elección del proyecto

En los últimos años estamos presenciando una revolución en el sector aeroespacial con los comúnmente conocidos como drones a varios niveles. Con mayor o menor intensidad, todos mostramos interés por las nuevas tecnologías y aplicaciones que aparecen casi a diario en los medios de comunicación sobre este fenómeno.

En mi vida universitaria, dediqué dos años a estudiar primer y segundo curso del Grado en Ingeniería Aeroespacial en la Universidad Politécnica de Madrid, estudios que no pude continuar. Esta elección se fundamentó en la inclinación hacia este campo de mis gustos e intereses durante mi época de instituto.

Tras estos dos años, comencé a estudiar el Grado en Ingeniería Electrónica Industrial y Automática en la Universidad Politécnica de Cartagena.

Con todo lo anterior, cuando visualicé la oferta de Trabajo de Fin de Grado específico titulado “Diseño de un Vehículo Aéreo No Tripulado”, mi mente hizo las asociaciones oportunas: el proyecto me permitiría congrega mis estudios universitarios en una investigación que daría lugar al diseño de una aeronave mediante la aplicación de los conocimientos electrónicos adquiridos durante los años de estudio del Grado en Ingeniería Electrónica y Automática en la Universidad Politécnica de Cartagena.

0.3 Objetivos

El proyecto está centrado en el desarrollo de un prototipo de VANT controlado a través de una aplicación de Smartphone para el sistema operativo Android. Se empleará una placa de desarrollo de hardware y software de la compañía Arduino, y la comunicación se establecerá mediante Bluetooth. En cuanto a los elementos actuadores y a la estructura, se aprovecharán algunos componentes de los que se disponía con anterioridad.

Además, se recopilará información acerca de la historia y las aplicaciones actuales.

A continuación se explican las diferentes partes que incluirá el proyecto.

0.3.1 Documentación

Se llevará a cabo una búsqueda de información sobre la historia de los VANT, tanto militar como civil.

Además, se investigará sobre las últimas aplicaciones, tecnologías y normativa.

0.3.2 Selección de componentes

Se seleccionarán los componentes necesarios para completar el diseño del VANT teniendo en cuenta los elementos de los que ya se disponía inicialmente.

0.3.3 Control

Se diseñará la programación de la placa de control que se encargará de actuar sobre los motores en función de las entradas de los sensores y de las órdenes comandadas.

0.3.4 Sensores

Se configurarán los sensores necesarios para el control del VANT, y se diseñará el código de control en lazo cerrado mediante el cual las sucesivas lecturas de la desviación del plano horizontal modificarán el comportamiento global del dispositivo para conseguir un control de estabilidad.

0.3.5 Módulo de comunicación

Se realizará la configuración del módulo encargado de la comunicación que se debe establecer para transferir a la placa de control las acciones comandadas de forma remota.

0.3.6 App

Se diseñará e instalará la aplicación de Smartphone para sistema operativo Android a través de la cual se controlará el comportamiento del VANT.

Capítulo 1

Documentación

1.1 Historia

En la acepción más amplia del término, la aviación no tripulada abarca un amplio espectro de aeronaves. La genealogía de las aeronaves no tripuladas esconde sus raíces en el desarrollo de los llamados «torpedos aéreos», antecesores de los actuales misiles crucero, que posteriormente se desarrollaron a través de las ramas de las bombas guiadas (no propulsadas), los blancos aéreos (llamados «drones» en la terminología anglosajona), los señuelos, los modelos recreacionales y/o deportivos de radio-control, las aeronaves de investigación, las aeronaves de reconocimiento, las de combate, e incluso algunos modelos más extraños de vuelo extra-atmosférico.

1.1.1 Historia militar

Los datos más antiguos que se tiene sobre el uso de plataformas aéreas no tripuladas datan de 1849, cuando el ejército austriaco usó en una batalla contra la ciudad de Venecia globos cargados con explosivos lanzados desde un barco, que explotaban al caer verticalmente cuando el globo se desinflaba mediante una batería galvánica con un hilo de cobre. Aunque estos globos no concuerdan con la definición actual de los drones, el concepto sí se ajusta al de “plataforma no tripulada que porta una carga útil”, en este caso explosivos.

Primera Guerra Mundial

Durante la Primera Guerra Mundial la aviación no tripulada se vio frenada por falta de recursos tecnológicos. Este conflicto bélico fue la cuna de las primeras aeronaves no tripuladas tal como las conocemos, y su desarrollo fue paralelo al de los misiles como forma de llevar a cabo el seguimiento y guiado de los explosivos a un objetivo.

Periodo de entreguerras

Tras la Primera Guerra Mundial, los británicos desarrollaron el blanco radiocontrolado Fairey Queen, que fundamentalmente se utilizó para el entrenamiento de las fuerzas de artillería durante la Segunda Guerra Mundial. Poco después se fabricó el “DH.82B Queen Bee” (abeja reina), del que se cree que derivó el término “drone” (zángano).



Figura 1 DH. 82B Queen Bee

http://www.ptaeromuseum.com/aircraft_projects/dehavilland_dh82b/dh82b_s.jpg

Segunda Guerra Mundial

En EE.UU se desarrolló el RP4, que se produjo a millares como sistema de entrenamiento de las fuerzas armadas. A través de estos aviones se fue desarrollado la tecnología y el uso de control remoto por radio.

El concepto de los misiles crucero fue rescatado por la Alemania nazi con el «V1 Vengeance Weapon», que fue el primer misil crucero equipado con motor de reacción. Se basaba en un sistema barométrico para regular velocidad y altura, y un anemómetro usado para estimar la distancia recorrida.



Figura 2 “V1 Vengeance Weapon” siendo transportado

https://upload.wikimedia.org/wikipedia/commons/b/bf/Bundesarchiv_Bild_146-1975-117-26,_Marschflugk%C3%B6rper_V1_vor_Start.jpg

La década de los 50

En la posguerra, la compañía Radioplane desarrolló con éxito una serie de blancos aéreos no tripulados llamados «Falconer» o «Shelduck», que evolucionaron hasta que se llegaron a producir en cantidades mucho mayores que cualquier otro blanco aéreo propulsado con motor alternativo.

En esta época también se desarrollaron señuelos antiradar.

La década de los 60. La Guerra Fría

Muchos UAV de esta época fueron equipados con cámaras para misiones de reconocimiento sobre territorio enemigo, controlados mediante radio desde una estación de control en tierra o desde una aeronave tripulada tipo «stand-off». Una vez completada su misión, se recuperaban con un paracaídas.

La década de los 70 y el auge de las misiones de reconocimiento

Con la presión de la intensificación de la Guerra Fría, los sistemas dedicados a las misiones de reconocimiento y vigilancia se fueron haciendo más sofisticados tanto en los requisitos de misión como en la seguridad de sus comunicaciones.

El fabricante Westland y otros empezaban a considerar el uso de UAV de despegue vertical. El sistema Westland Wisp, especialmente diseñado para el Ministerio de Defensa del Reino Unido, utilizaba un sencillo piloto automático basado en giróscopos para la estabilidad y se operaba manualmente por radio-control en línea de vista. Estaba equipado con una cámara de TV diurna que enviaba imágenes en tiempo real al operador y adoptó, por primera vez, una configuración de rotores coaxiales.

Los años 80 y la expansión internacional de los RPAs

El desarrollo en paralelo de cargas de pago con sensores cada vez más sofisticados abrió el campo para misiones de guerra electrónica, designación láser y relé de comunicaciones, entre otras.

Los años 90. El desarrollo de las grandes plataformas

La mayor disponibilidad del sistema de posicionamiento global (Global Positioning System; GPS) y de las comunicaciones satélite liberó a los UAVs de operar dentro del alcance de la señal de radio. De esta forma, junto con los sistemas digitales de control de vuelo (Digital Flight Control System; DFCS), el alcance y la precisión de la navegación mejoraron apreciablemente, desarrollándose sistemas de medio y largo alcance.

La década de los 2000

Para poder transportar radares de apertura sintética (SAR) a cada vez mayores distancias hacía falta aumentar el alcance y la autonomía de las plataformas existentes, lo que obligó a desarrollar UAS cada vez más grandes y pesados. De esta forma apareció el modelo Predator B, equipado ya con un motor turbohélice, y el mayor aún y de mayor altitud Global Hawk, con un motor turbofan.

Otras líneas de avance son las que tratan de aumentar la automatización, reduciendo la carga de trabajo y los errores de las tripulaciones en tierra.

Más allá del 2010

La tendencia comercial en el mercado de la robótica está en constante crecimiento. La tecnología está apoyando estas tendencias, con la disponibilidad de microprocesadores cada vez más baratos y capaces que fomentan estos desarrollos. Los principales obstáculos para la continuación de esta tendencia son la cantidad (y el coste) del software involucrado y la barrera de fabricación de dichos chips, en cantidad y calidad suficientes.

Los retos a abordar en el futuro próximo van a venir de la mano del adecuado desarrollo e implementación de los siguientes aspectos:

- Nuevas tecnologías: nuevos materiales, configuraciones y actuación bio-mimética, operación completamente autónoma, capacidad de «ver y evitar», etc.
- Nuevos «roles»: evolución de las misiones semejantes a las de las aeronaves tripuladas hacia misiones completamente nuevas.
- Nuevos retos: para desarrollar aún más la aviación no tripulada se necesita una mayor fiabilidad, un marco regulador apropiado y una cartera de pedidos estable, en este orden.

1.1.2 Historia civil

Tras años de presencia en el sector militar, los VANT se han posicionado como una tecnología presente en multitud de aplicaciones que nada tienen que ver con conflictos bélicos.

En la actualidad, las aplicaciones de estos UAV civiles son cada vez mayores. Además de su versatilidad, el coste económico por operación que suponen es mucho menor que el de los sistemas más tradicionales (aviones y helicópteros convencionales).

Sin embargo, el desarrollo de los drones en el sector civil debe superar aún las trabas burocráticas que encuentra a su paso.

En Europa la compañía española [Flightech Systems](#) ha sido la pionera obteniendo los primeros Certificados de Aeronavegabilidad Experimental otorgados en Europa por AESA.

Otras compañías civiles están siguiendo el mismo camino en busca de la certificación definitiva que permita operar los UAV civiles, pero todo parece indicar que estos dispositivos tienen un gran potencial de actividad a corto plazo.

1.2 Aplicaciones actuales

La aplicación de los drones se encuentra en continua expansión y es común encontrar a diario en los medios de comunicación noticias acerca de nuevas aplicaciones y modelos de negocio generados entorno al uso de un vehículo no tripulado.

En la mayoría de los casos, la ventaja principal de emplear drones para sustituir o complementar multitud de actividades del ser humano es la flexibilidad de uso del recurso que representa el vehículo y la reducción del riesgo para la vida humana, además de la simplicidad de uso y la reducción de costes.

1.2.1 Aplicaciones al control de la calidad del aire

El control de la calidad de aire es una preocupación constante tanto para la ciudadanía como para la administración. Tradicionalmente, para llevar a cabo esta tarea, se han empleado instrumentos acoplados en diferentes vectores de observación, como aviones y globos. Esto se suma al control en tierra a través de la instalación de redes de medida que combinan la vigilancia meteorológica con la medida de la contaminación del aire. El empleo de drones para llevar a cabo el control de la calidad del aire consiste en la instalación en los mismos de los instrumentos de medida de la contaminación necesarios para el análisis del aire. Sin embargo, la medida realizada puede ser ligeramente alterada por otras especies químicas distintas a las que se desea medir, por lo que la precisión de estos sistemas no alcanza en la actualidad la de los sistemas clásicos de observación.

1.2.2 Aplicaciones cartográficas

Antiguamente se realizaba cartografía en papel a pie de campo, o se empleaban satélites o aviones tripulados para representar el terreno; hoy, gracias al desarrollo de la tecnología, se hace de manera digital.

En los últimos años, los drones se han posicionado como una herramienta de obtención de este tipo de información con un ahorro de tiempo y de costes mediante la técnica de la estereoscopia: el drone realiza la captación de imágenes de forma subortogonal (casi nunca son totalmente ortogonales, pues no es imprescindible que lo sean).

La altura del dispositivo no deberá ser muy elevada a la hora de realizar el vuelo, condicionada por la normativa en vigor en España, que también establece que sólo está permitido llevar a cabo operaciones con RPAs en espacio aéreo no controlado y fuera de núcleos de población, lo que limita su uso. Además, es obligatorio operar con condiciones de vuelo VFR (Visual Flight Rules), es decir, con meteorología de visibilidad determinada. Con todo esto, la utilización de drones en esta aplicación supone más ventajas que inconvenientes, lo que, con la reducción de costes que proporcionará a medio plazo, resultará en una tecnología a tener en cuenta para aplicaciones cartográficas.

1.2.3 Aplicaciones a la prospección y explotación de recursos minerales

Las actividades de extracción de los minerales y de los hidrocarburos del subsuelo requieren un conjunto de actividades: para comenzar, de prospección para localizar los yacimientos y, posteriormente, de investigación, para conocer su morfología, características y propiedades. En una fase posterior se realiza la explotación o extracción de los minerales.

La instrumentación utilizada en minería, y concretamente en prospección geológico-minera tiene unas dimensiones reducidas, por lo que es idónea para instalarse en los UAV.

El uso de la prospección geofísica aérea tiene, entre otras, una serie de ventajas:

- Son métodos no intrusivos que no provocan impactos o daños medioambientales.
- No es necesario solicitar permisos de ocupación, ni licencias relacionadas.
- Se pueden obtener datos de zonas remotas, accidentadas y con gran densidad de cobertura vegetal.
- Los UAV son más sostenibles, pues requieren menos materiales para su fabricación, necesitan menos combustible, generan menos ruido y son más fáciles de reutilizar una vez terminada su vida útil.

1.2.4 Aplicaciones hidrológicas

La complejidad de operación y caracterización de las grandes masas de agua (ya sean continentales u oceánicas) siempre ha supuesto un gran reto para el hombre. Además, su influencia en fenómenos como el cambio climático o el calentamiento global ha hecho evidente la necesidad de generar procesos de evaluación continua del impacto de las actividades del hombre en estas masas de agua. Los vehículos no tripulados se están haciendo hueco como potentes y flexibles herramientas de su monitorización y control.

1.2.5 Aplicaciones agrícolas

La agricultura de precisión consiste en el manejo diferenciado de los cultivos empleando el conocimiento de la variabilidad existente en una explotación agrícola, para cuya caracterización se emplean herramientas tecnológicas como los Sistemas de Posicionamiento Global (GPS), sensores plantaclima-suelo e imágenes multiespectrales obtenidas a partir de satélites, aviones o UAV.

Los datos obtenidos a partir de los sensores se almacenan en formato digital en forma de tablas y mapas, pudiéndose generar a partir de ellos la información aprovechable por el agricultor para llevar a cabo la toma de decisiones en campo (fertilización, podas o clareos). Además, pueden llevar a cabo operaciones de dispersión de plagas aviares o generación de inventarios de cultivos, entre otras, facilitando además la detección de incidencias, desde problemas de fertilización hasta infestaciones de plagas y hongos que no son visibles a nivel de suelo.

1.2.6 Aplicaciones a la extinción nocturna de incendios forestales: «Proyecto Nitrofirex»

Actualmente se emplean aeronaves de ala fija en la extinción de incendios desde el aire, que son turbohélices lentos que realizan sus descargas de manera aislada y por medios visuales en una operación que implica riesgo para las tripulaciones. Permiten descargar gran cantidad de agente extintor, pero presentan una escasa maniobrabilidad a baja altura, y no se pueden emplear en «primer ataque» al fuego por sus elevados tiempos de reacción y costos de operación.

La madurez de las tecnologías para el guiado y control de los Sistemas Aéreos no tripulados (UAV) permite plantear innovadoras opciones operativas como es la capacidad de «rocíar»

(un líquido) o esparcir (un sólido en polvo) una cantidad importante de agente en un punto programado de la atmósfera. Es el caso de NITROFIREX, un innovador proyecto español que integra tecnologías de la industria de defensa que pueden ser de aplicación en la lucha contra incendios forestales, en el combate contra una emergencia atómica, química o biológicas, para actuar en fenómenos meteorológicos (provocar lluvia, evitar pedrisco, diseminar nieblas) para luchar contra plagas o sembrar en lugares remotos y/o inaccesibles e incluso para la fumigación nocturna de plantaciones de droga.

El concepto NITROFIREX consiste en unos depósitos planeadores Autónomos (DPA) no tripulados que contienen una cantidad de 2.500 ± 250 litros de carga útil. Tras ser lanzados desde la rampa trasera de aviones de transporte pesado, que actúan como Avión Lanzador (AL), se dirigen autónomamente al foco del incendio para descargar su contenido con gran precisión. Una vez efectuada la descarga del agente extintor, los depósitos se recuperan mediante una maniobra de «escape» de la zona de peligro aprovechando tanto su gran y repentina pérdida de peso como su velocidad remanente. Antes de alcanzar el punto más alto de la maniobra de escape el vehículo, ya vacío, comienza a propulsarse con su reactor, lo que le permite retornar autónomamente a la base de operaciones del avión lanzador para estar disponible lo antes posible.



Figura 3 Representación de un DPA de NITROFIREX en funcionamiento
<http://www.unvex12.com/galeria/nitrofirex04.jpg>

El empleo de depósitos planeadores autónomamente guiados permite la operación nocturna desde el aire en la extinción de incendios forestales, aumentando la precisión y concentración de las descargas, además de incrementar la cantidad de agua lanzada tanto por hora de operación como por hora de vuelo y abaratar el coste del litro lanzado con respecto a los medios aéreos empleados en la actualidad. Además, al ser los aviones lanzadores aviones de transporte pesados no exclusivos, permite disminuir en gran medida el coste final por hora de vuelo realizada.

1.2.7 Aplicaciones al control de obras y evaluación de impactos

Hasta el momento actual, las técnicas empleadas para la supervisión geométrica, de impacto visual y medioambiental y para el control económico de la ejecución de cualquier obra han sido la topografía clásica y la fotogrametría.

La evolución de la tecnología ha permitido el desarrollo de soluciones basadas en datos capturados por drones, que permiten hacer de forma sistemática, precisa, rápida y económica seguimiento de las obras ya sea con carácter métrico para realización de certificaciones. La unificación de los datos en cuanto a calidad, precisión, cantidad, homogeneidad temporal, capacidad de procesamiento y análisis en series temporales, hace de los UAV una herramienta muy apropiada para el control de las obras y el análisis del impacto de las mismas.

1.2.8 Aplicaciones al mantenimiento de líneas eléctricas

Las líneas y subestaciones eléctricas tienen, en general, elementos que se ubican en altura, alejados por seguridad del alcance de las personas. Las líneas eléctricas, además, son infraestructuras lineales que pueden discurrir por entornos de difícil acceso. En este escenario, los vehículos aéreos permiten un acceso directo y cercano a los elementos de las instalaciones.

Actualmente, las empresas realizan operaciones rutinarias de inspección en las líneas eléctricas a modo de mantenimiento preventivo, que se llevan a cabo de diferentes maneras, una de ellas mediante el uso de helicópteros tripulados que obtienen imágenes visuales (para detectar defectos o anomalías en los elementos de las líneas) y termográficas (para buscar puntos denominados «calientes» cuya temperatura es mayor, que pueden estar asociados a un contacto inadecuado o a otro tipo de defecto que conviene analizar).

1.2.9 Búsqueda de personas desaparecidas

Uno o más drones de tipo helicóptero pueden operar en la búsqueda de personas desaparecidas en lugares abiertos o de difícil acceso como zonas montañosas o nevadas. Su permanente disponibilidad debido a su facilidad de almacenamiento por su reducido tamaño disminuye considerablemente el tiempo de búsqueda. Además, su bajo coste, en comparación con el coste de un helicóptero tradicional (al menos 30 veces más caro), los hace idóneos para esta tarea. Pero la ventaja económica que suponen no es la única: la eliminación del riesgo de vidas humanas es otra de las grandes virtudes del empleo de VANTs para este cometido.



Figura 4 Aplicación de VANTs para búsqueda de personas desaparecidas.

http://www.iuavs.com/pages/_images/imagenesiuavs01/

1.2.10 Cooperación de VANTs

Un sistema con múltiples robots UAV presenta mayor robustez que un único dispositivo operando en solitario, debido a la redundancia que proporciona.

La cooperación puede llevarse a cabo en paralelo entre los drones, ayudándose unos a otros para, por ejemplo, cubrir grandes áreas en exteriores o crear redes de sensores móviles. Estos enjambres de vehículos aéreos no tripulados pueden desplegarse para realizar tareas de búsqueda ante cualquier tipo de desastre natural, como terremotos o ataques terroristas, ayudando a localizar a personas que puedan necesitar ayuda.

1.3 Normativa

Dada la reciente intensificación de la presencia de los drones en cada vez más entornos de diversos tipos, se hace necesaria la regularización de su uso.

La Agencia Estatal de Seguridad Aérea (AESA) es la entidad responsable de controlar el uso de aeronaves tripuladas por control remoto en España. Su objetivo es clarificar en qué circunstancias está permitido el uso de drones y en cuáles está prohibida su operación, y fijar las consecuencias de no cumplir con lo establecido.

1.3.1 Ley de Navegación Aérea

El artículo 150.2 de la Ley de Navegación Aérea establece que:

2. Las aeronaves civiles pilotadas por control remoto, cualesquiera que sean las finalidades a las que se destinen excepto las que sean utilizadas exclusivamente con fines recreativos o deportivos, quedarán sujetas asimismo a lo establecido en esta Ley y en sus normas de desarrollo, en cuanto les sean aplicables. Estas aeronaves no estarán obligadas a utilizar infraestructuras aeroportuarias autorizadas, salvo en los supuestos en los que así se determine expresamente en su normativa específica.

Esto significa que, igual que un avión, los drones de uso comercial y civil no pueden ser operados en espacio aéreo controlado sin autorización, ya que se consideran aeronaves y deben cumplir las leyes aeronáuticas vigentes.

1.3.2 Ley 18/2014

La Ley 18/2014 es aplicable sólo al uso de aeronaves pilotadas por control remoto, de peso inferior a 150 kg y con fines comerciales o profesionales (como la investigación, vigilancia, etc.). En los artículos que van del 50 al 53 se regula la operación de aeronaves civiles pilotadas por control remoto. Entre otras condiciones, se establece la obligación de no superar los 120 metros de altura y siempre tener el drone a la vista.

1.3.3 Real Decreto 552/2014

La existencia de la Ley de Navegación Aérea, sin embargo, no implica la libertad de vuelo de VANTs en cualquier espacio aéreo no controlado.

El Real Decreto 552/2014, de 27 de junio, por el que se desarrolla el Reglamento del aire y disposiciones operativas comunes para los servicios y procedimientos de navegación aérea y se modifica el Real Decreto 57/2002, de 18 de enero, por el que se aprueba el Reglamento de Circulación Aérea en su Capítulo II Artículo 4.1 establece que:

1. No se realizará ningún vuelo sobre aglomeraciones por debajo de las alturas mínimas previstas en SERA.3105, salvo aquellas operaciones que, excepcionalmente y por razones de interés general debidamente justificadas, se autoricen por el Director de Seguridad de

Aeronaves de la Agencia Estatal de Seguridad Aérea a instancias del operador, sin perjuicio de las alturas mínimas que resulten de aplicación conforme a la normativa específica que regule las distintas actividades aeronáuticas y de las exenciones para operaciones especiales.

SERA 3105: “Excepto cuando sea necesario para despegar o aterrizar, o cuando se tenga permiso de la autoridad competente, las aeronaves no volarán sobre aglomeraciones de edificios en ciudades, pueblos o lugares habitados, o sobre una reunión de personas al aire libre, a menos que se vuele a una altura que permita, en caso de emergencia, efectuar un aterrizaje sin peligro excesivo para las personas o la propiedad que se encuentren en la superficie. [...]”

Ante las consultas ciudadanas, AESA decidió responder explicando por las normas a seguir para el uso recreativo de su dron. Comunicó que era necesario cumplir el punto 2.3.1.1 del Reglamento de la Circulación Aérea, por lo que *"deberá evitar volar en zonas en que se realicen vuelos de aeronaves tripuladas a baja altura así como en zonas urbanas y donde haya presencia de personas"*:

2.3.1.1. Operación negligente o temeraria de aeronaves. Ninguna aeronave podrá conducirse negligente o temerariamente de modo que ponga en peligro la vida o propiedad ajenas. Dicho Reglamento ya no está en vigor porque se ha tenido que adaptar la norma europea SERA con el Real Decreto 552/2014 mencionado anteriormente, pero en dicho reglamento impuesto por la UE recogen algo similar en su artículo SERA.3101 y las conclusiones pueden ser las mismas:

SERA.3101: Ninguna aeronave podrá conducirse negligente o temerariamente de modo que ponga en peligro la vida o propiedad ajenas.

1.3.4 Vuelo en recintos cerrados

Los recintos completamente cerrados (un pabellón industrial o deportivo, un centro de convenciones, un domicilio particular, etc.) no están sujetos a la jurisdicción de AESA, pues no forman parte del espacio aéreo. La autorización del vuelo de drones en su interior, así como de dictaminar los límites a respetar, corresponde al titular del recinto en cuestión.

La AESA ha dispuesto una página web con preguntas frecuentes y sus correspondientes respuestas sobre normativa, requisitos y límites sobre el vuelo de drones en el espacio aéreo del territorio español, a la que se puede acceder pinchando [aquí](#).

Como conclusión, podemos extraer que el uso de los VANT está intensamente regulado y limitado. A la velocidad a la que este sector se está desarrollando, en ocasiones las trabas burocráticas con las que se encuentran pueden significar un obstáculo para la evolución y el comercio de estos dispositivos

Capítulo 2

Selección de componentes

En nuestro punto de partida disponemos de un VANT averiado del que se estaba en posesión con anterioridad al inicio del proyecto. Se trata del modelo X5C, de la marca [Syma](#).

El sistema de comunicación utilizado para controlar el Syma X5C es el radiocontrol, pero no emplearemos la placa de control en nuestro proyecto.

Los elementos que se aprovecharán para nuestro cometido serán la estructura o chasis, los rotores, los motores y la batería.



Figura 5 Syma X5C

<http://www.quadcopterpatrol.com/syma-x5c-1-in-depth-review/>

Será útil tener siempre en mente el siguiente diagrama de bloques, en el que se interconectan los diferentes elementos del dispositivo:

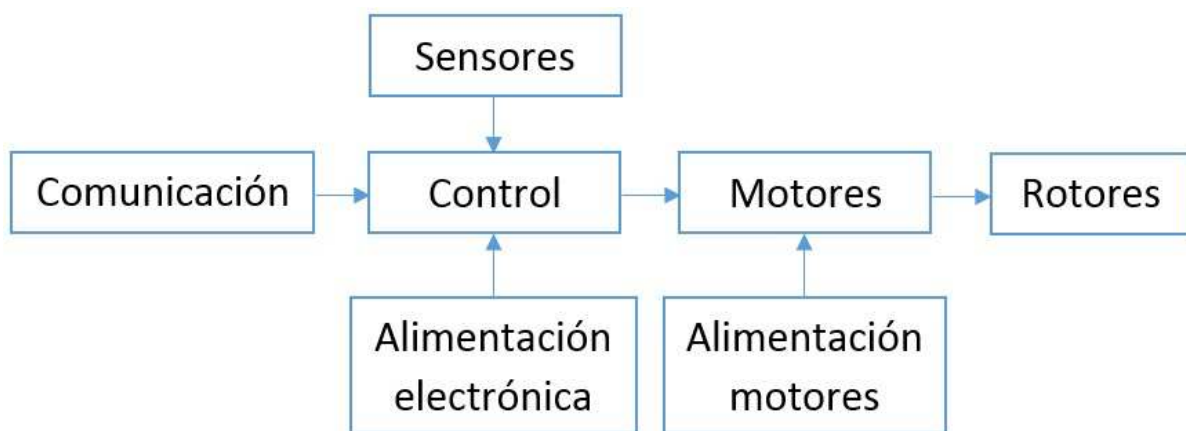


Figura 6 Diagrama de bloques

2.1 Control

Para que el VANT sea capaz de desempeñar una correcta operación, es necesario definir la actuación sobre los motores en función de los datos obtenidos mediante los sensores y de las órdenes comandadas por el usuario empleando una placa de control.

En el mercado encontramos infinidad de posibilidades, basadas principalmente en el sistema de comunicación empleado y la aplicación para la que se desarrolle el diseño del dispositivo.

En el proyecto que nos ocupa, se emplea una placa de la marca Arduino para implementar el algoritmo de control del dispositivo. La decisión se basa en la formación sobre la plataforma recibida durante los estudios universitarios, además de la activa participación de internautas en los foros de Arduino, que colaboran proponiendo soluciones a los problemas planteados por los participantes, constituyendo así una amplia red de apoyo.



Figura 7 Arduino UNO R-3

<https://store.arduino.cc/usa/arduino-uno-rev3>

2.2 Sensores

Se pueden utilizar infinidad de sensores en función de la aplicación para la que se diseñe el dron.

2.2.1 IMU

Lo imprescindible para controlar la orientación y la estabilidad es una IMU (Inertial Measurement Unit), un dispositivo capaz de medir la fuerza (aceleración) y la velocidad a la que se somete, para a través de una serie de cálculos obtener la inclinación del dispositivo respecto al suelo y utilizar esta información en el control.

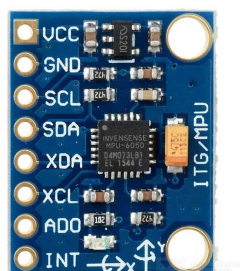


Figura 8
IMU MPU-6050,
de InvenSense

http://img.dxcdn.com/productimages/sku_154602_2.jpg

La IMU seleccionada para nuestro proyecto es la MPU-6050, de *InvenSense*. Tiene 6DF (Degrees Of Freedom o Grados de Libertad), repartidos entre un acelerómetro (que mide la aceleración que experimenta) y un giróscopo o giroscopio (que mide la velocidad angular), de tres ejes cada uno. Además, ofrece la posibilidad de añadir un magnetómetro de tres ejes a modo de brújula.

2.2.2 Fotorresistencia

Además, en nuestro proyecto emplearemos una fotorresistencia a modo de sensor de intensidad lumínica que proporcionará la información necesaria para encender/apagar unos diodos LED que ya incluía la estructura aprovechada.

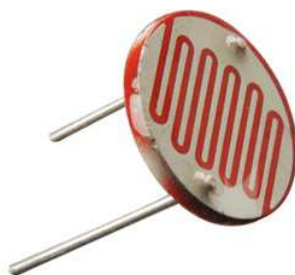


Figura 9
Fotorresistencia

<https://www.prometec.net/wp-content/uploads/2016/01/ldr-1.jpg>

2.3 Actuadores

Los elementos que transforman el comportamiento programado en la placa de control en un cambio externo apreciable a simple vista se denominan actuadores. En nuestro proyecto encontramos los siguientes:

2.3.1 Motores

Los motores se encargan de proporcionar la tracción necesaria para que los rotores se pongan en movimiento a través de un sistema de engranajes que provoca una reducción en la velocidad de los rotores respecto a la que experimentan los motores.

Se utilizan 4 motores DC con escobillas, dos de tipo A y dos de tipo B, cuya disposición en la estructura es antisimétrica.

SYMA



Figura 11 Motor de tipo A

<http://www.symatoystore.com/Syma-Spare-Parts/Syma-X5C-Spare-Parts/X5C-07-Motor-A>

SYMA



Figura 10 Motor de tipo B

<http://www.symatoystore.com/Syma-Spare-Parts/Syma-X5C-Spare-Parts/X5C-08-Motor-B>



Figura 12

Disposición de los motores en el Syma X5C

<http://www.symatoys.com/goodshow/x5c-syma-x5c-explorers.html>

2.3.2 Rotores

Los rotores son los elementos que generan la sustentación aerodinámica, proporcionando el empuje necesario para que el dispositivo se eleve en el aire. Están compuestos por dos palas, que giran a la velocidad comandada en función de la potencia que se proporciona a los motores.



Figura 13 Rotores

En nuestro montaje contamos con dos tipos de hélices, debiendo situarlas de manera antisimétrica.

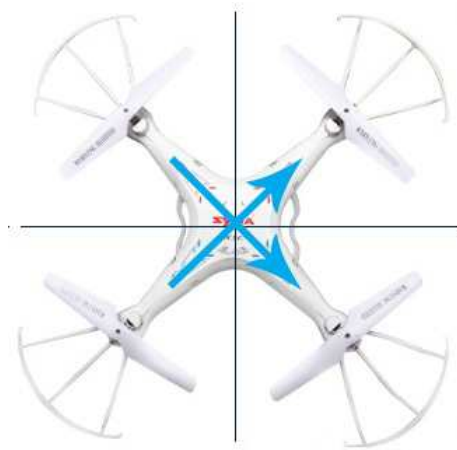


Figura 14 Rotores situados de forma antisimétrica

<http://www.symatoys.com/goodshow/x5c-syma-x5c-explorers.html>

2.3.3 LEDs

Se actuará sobre unos diodos LED presentes en la estructura del dron en función del nivel de intensidad lumínica ambiental medido mediante una Fotorresistencia. Estos LEDs permitirán mantener el dispositivo localizado en todo momento, incluso cuando la luminosidad ambiental no es suficiente para visualizar correctamente todas las partes del mismo.

Además, se programará el funcionamiento de los LEDs para que actúen como indicadores externos de que se ha recibido un comando desde el Smartphone.



Figura 16 Diodos LED



Figura 15 Diodos LED

2.4 Módulo de comunicación

En el presente proyecto, la comunicación entre la aplicación y Arduino se lleva a cabo mediante Bluetooth.

Se ha utilizado el módulo HC-05, de fácil uso y SPP (Serial Port Protocol). Es configurable como Maestro (Master) o como esclavo (Slave) mediante comandos AT. Trabaja a 3.3V, pero está adaptado para poder usarlo a 5V, lo que lo hace totalmente compatible con Arduino. Presenta una excelente relación calidad-precio, por lo que es ampliamente utilizado.

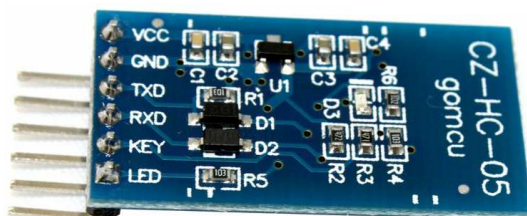


Figura 17 Módulo de comunicación Bluetooth HC-05

<https://img.pccomponentes.com/articles/7/79483/modulo-hc-05-bluetooth-compatible-con-arduino-2.jpg>

2.5 Placa soporte y de conexionado

Para el montaje y conexionado de los distintos elementos se empleará una placa de pruebas o protoboard.

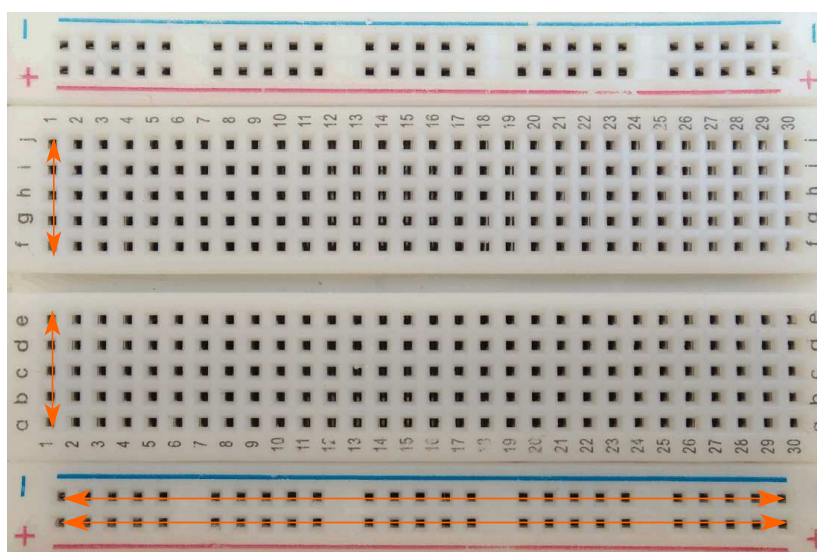


Figura 18 Protoboard

Cuando se utilizan este tipo de placas para hacer el montaje de un circuito electrónico es importante conocer las conexiones internas de la placa para evitar provocar cortocircuitos que pueden dañar algunos de los elementos del conjunto.

Las filas que en la imagen adjunta se señalan con dos flechas horizontales están conectadas en el sentido de las mismas y son independientes entre sí.

Las filas numeradas del 1 al 30 están conectadas verticalmente como indican las flechas verticales, y son independientes entre sí.

2.6 Batería

Las salidas digitales de Arduino sólo son capaces de suministrar 40 mA, lo que no es suficiente para alimentar a los motores. Por ello, se utilizará una batería de Li-Po (compuestas de Litio y polímero), que ofrece 3,7 V y 500mAh para energizar los motores de manera exclusiva.



Figura 19 Batería Syma X5C

http://cdn.shopify.com/s/files/1/0384/5221/products/syma_x5c_3.7v_lipo_grande.jpg?v=1413854087

2.7 Estructura

La estructura es la base que incorpora todos los componentes. Nos encontramos básicamente con dos tipos de estructuras: de ala fija y de ala rotatoria. El ala fija tiene el aspecto y forma de un avión de aerodelismo tradicional y de ala rotatoria hallamos dos formas de dispositivos: los helicópteros y los multirrotores.

Los multirrotores son el tipo de estructura más extendida actualmente. Proporciona una gran versatilidad y eficacia en las operaciones por su simple manejo y por la velocidad de montaje. Es una plataforma estable por naturaleza, debido a que los motores se sitúan alrededor y a la misma distancia del centro de gravedad de la aeronave.

Según la cantidad de motores encontramos tricópteros (3 motores), cuadricópteros (4 motores), hexacópteros (6 motores) y octocópteros (8 motores). Estos motores pueden situarse en brazos con diferentes configuraciones:

- “Y” (i griega)
- “Y invertida” (i griega invertida)
- “X” (equis)
- “+” (cruz)

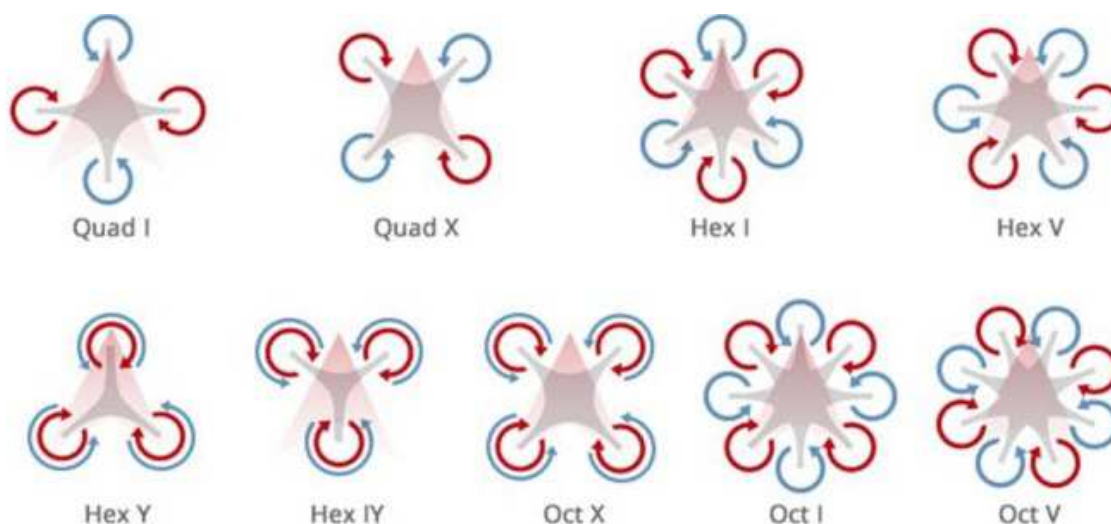


Figura 20 Tipos de estructuras y configuraciones de los multirrotores

<http://dronespain.pro/tipos-de-drones-aereos/>

A la hora de seleccionar un multirrotor, debemos tener en cuenta que un incremento de brazos implicará un aumento de estabilidad, seguridad y propulsión, pero también de peso, motores y consumo.

En nuestro proyecto utilizaremos la estructura del Syma X5C, que presenta una configuración de cuadricóptero y una disposición en “X”, con un motor y su correspondiente rotor en cada uno de los cuatro brazos.



Figura 21 Estructura

Capítulo 3

Control

3.1 Arduino

Arduino es una plataforma de diseño de objetos interactivos (Physical Computing o Physical Interaction Design) que integra una tarjeta microcontroladora open-source y un entorno de desarrollo de software basado en el lenguaje de programación Processing. Se utiliza para desarrollar objetos y entornos interactivos relacionados con el diseño de mecatrónica (para consumo e industrial) y proyectos DIY (Do It Yourself) para aficionados.

Los objetos pueden ser autónomos (stand-alone) o pueden comunicarse con cualquier software a través de puerto serie, bien por cable o de manera inalámbrica.

El entorno de desarrollo (IDE, Integrated Development Environment) puede ser descargado de forma gratuita desde la [página web](#) de Arduino:

Veamos algunas de sus ventajas:

- Es un entorno multiplataforma, pudiéndose instalar en varios sistemas operativos.
- El IDE de Arduino está basado en Processing, un entorno de desarrollo de uso sencillo.
- No es necesario un puerto serie, pudiéndose programar vía USB.
- Tanto el hardware como el software es de tipo open-source.
- Es un recurso económico.



Figura 22 Entorno de programación de Arduino

El código de programación desarrollado se explica en detalle en el **Anexo 1: Código en Arduino**

El modelo de Arduino utilizado en el presente proyecto es el UNO-R3, cuyas características más importantes se especifican a continuación:

Arduino UNO R-3	
Microprocesador	ATmega328P
Memoria FLASH	32 Kb
Memoria RAM	2 Kb
Velocidad de reloj	16 MHz
Entradas/salidas digitales	12, de las cuales 6 pueden ser usadas como salidas PWM
Entradas analógicas	6
Conexiones	USB
	Jack

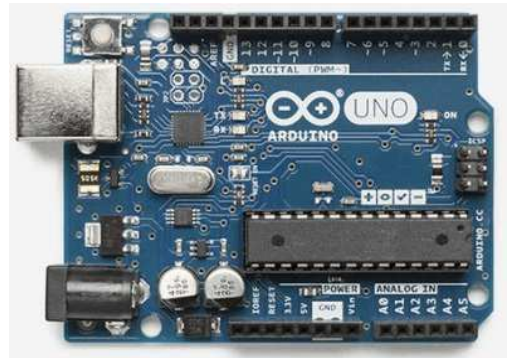


Figura 23 Arduino UNO R-3

<https://store.arduino.cc/usa/arduino-uno-rev3>

3.2 App

La interfaz a través de la que se realiza la entrada de comandos para actuar sobre el comportamiento del VANT es una App de Smartphone para Android.

La programación de la aplicación se ha llevado a cabo mediante Google App Inventor, de Google Labs.

3.2.1 App Inventor

App Inventor es un entorno de desarrollo de aplicaciones para dispositivos Android. Para el desarrollo es necesario un navegador web y un teléfono o table Android, aunque también se pueden probar las aplicaciones en un emulador.

Se trata de una herramienta de desarrollo visual muy fácil de usar, con la que el usuario debe enlazar una serie de bloques para crear la aplicación. El sistema es gratuito y se puede descargar fácilmente de la web.

Al construir la aplicación se trabaja con dos herramientas: App Inventor Designer y App Inventor Blocks Editor.

3.2.1.1 App Inventor Designer

En Designer se construye el Interfaz de Usuario, eligiendo y situando los elementos con los que interactuará el usuario y los componentes que utilizará la aplicación.

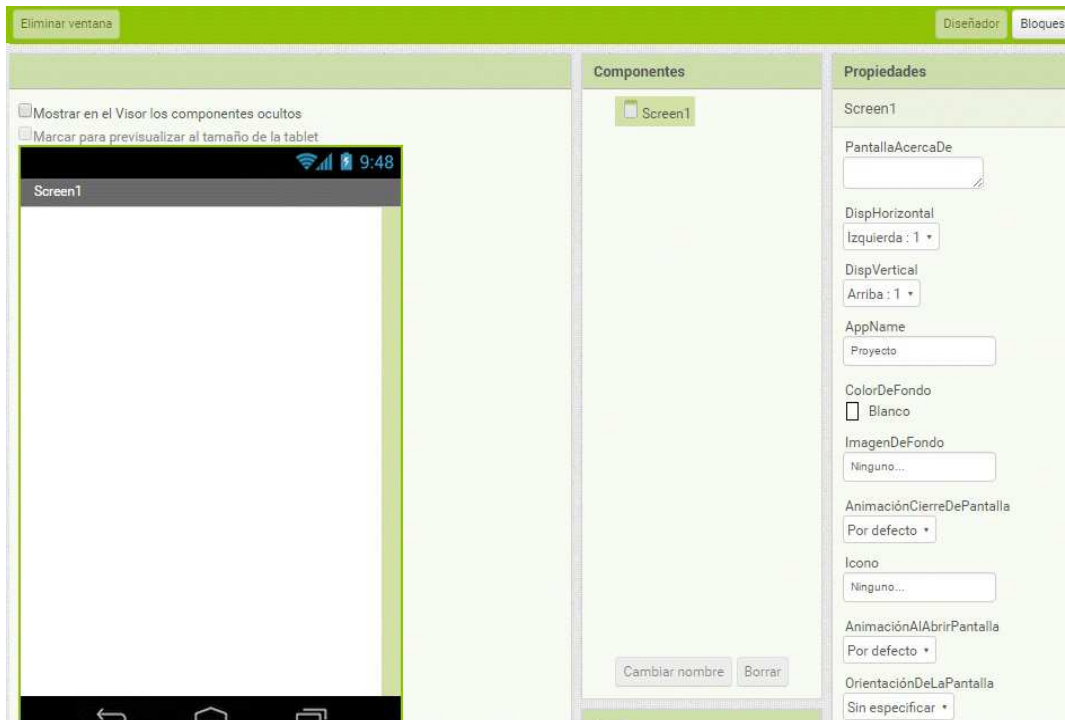


Figura 24 Vista Diseñador en App Inventor

3.2.1.2 App Inventor Blocks Editor

En Blocks Editor se desarrolla la programación que definirá el comportamiento de los distintos elementos de la aplicación.

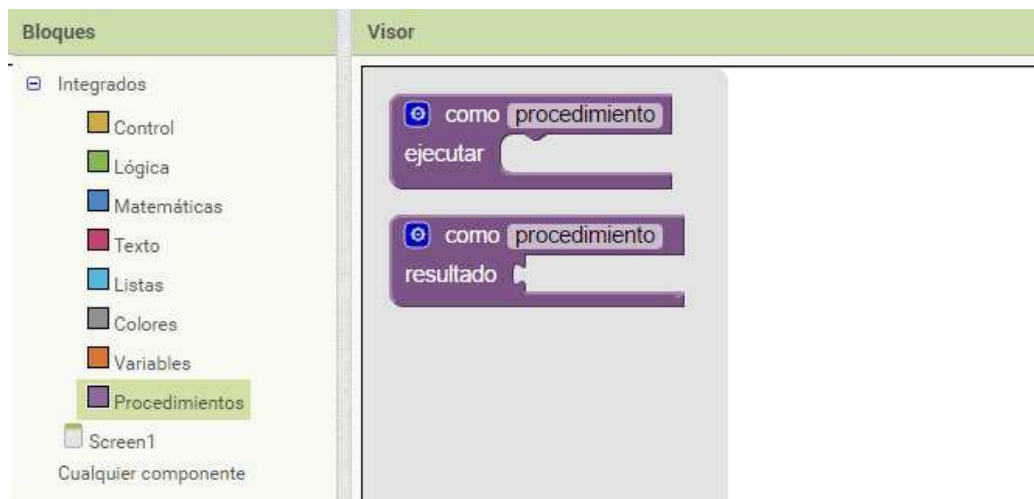


Figura 25 Entorno de programación Blocks Editor en App Inventor

3.2.2 ArdDrone

La aplicación propia desarrollada para el presente proyecto ha recibido el nombre de *ArdDrone*. Su diseño, funcionamiento y utilización de detallan a continuación.

3.2.2.1 Conexión con dispositivo

Antes de iniciar la aplicación, es necesario vincular el dispositivo con el teléfono del usuario. Para ello, accederemos a los ajustes Bluetooth del Smartphone.

Activando la funcionalidad y haciendo una búsqueda de dispositivos disponibles, nos aparecerá “ArdDrone”, que deberemos seleccionar.

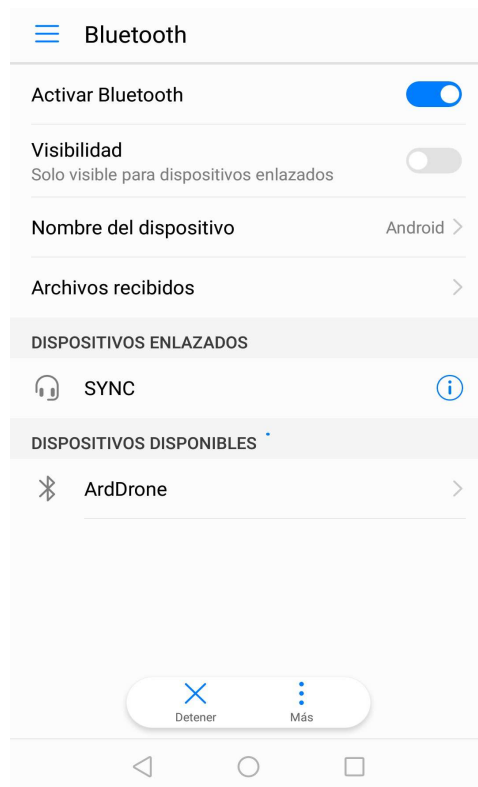


Figura 26 Enlazar dispositivo Bluetooth



Figura 27 Introducir PIN

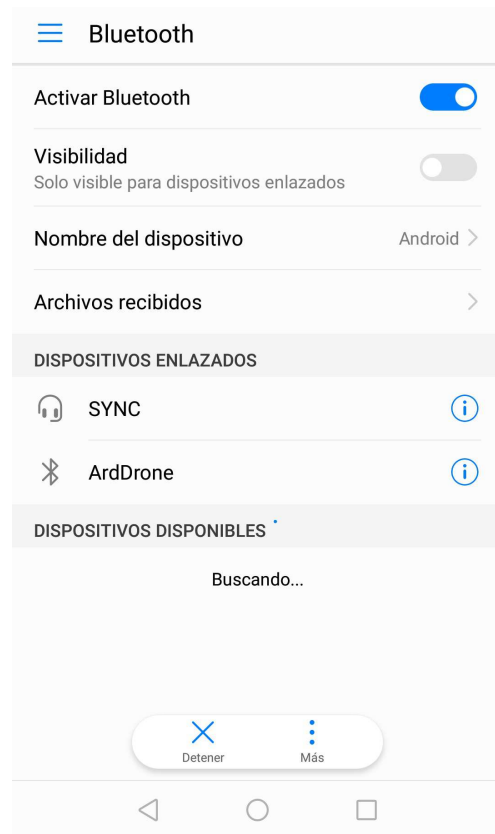


Figura 28 Dispositivo enlazado

Se abrirá una nueva ventana solicitando un PIN. Introducimos “1234” y pulsamos “Aceptar”. Una vez establecido el enlace, el dispositivo aparecerá en la lista de Dispositivos enlazados.

PIN: 1234

3.2.2.2 Interfaz

El interfaz que visualizará el usuario en su Smartphone tras iniciar la aplicación pulsando sobre la miniatura es la siguiente:



Figura 30
Miniatura
ArdDrone

Figura 29 Interfaz ArdDrone antes de conectar

Encontramos 8 botones:

Conectar	Al accionarlo se abrirá una lista con los dispositivos Bluetooth disponibles
Ascender	Al pulsarlo aumentará la velocidad de todos los rotores, por lo que el VANT ascenderá
Descender	Al pulsarlo disminuirá la velocidad de todos los rotores, por lo que el VANT ascenderá
Parar	Detendrá todos los motores
Arriba	Los motores delanteros disminuirán su velocidad, con lo que el VANT avanzará hacia delante
Derecha	Los motores laterales derechos disminuirán su velocidad, con lo que el VANT se desplazará en esa dirección
Abajo	Los motores traseros disminuirán su velocidad, con lo que el VANT retrocederá

Izquierda

Los motores laterales izquierdos disminuirán su velocidad, con lo que el VANT se desplazará en esa dirección

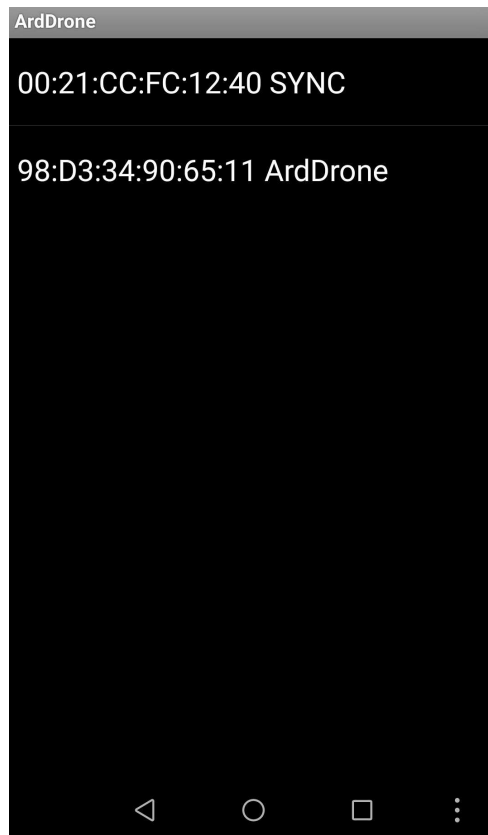


Figura 32
Botón
“Conectar”

Figura 31 Lista de dispositivos
accesibles

Al pulsar el botón “Conectar” se mostrará la lista de dispositivos accesibles. Tras seleccionar el dispositivo ArdDrone, se establecerá la conexión.

Una vez seleccionado el dispositivo HC-05 en la lista de dispositivos Bluetooth disponibles y establecida la conexión aparecerá el mensaje “Conectado” en la parte superior de la pantalla; si en algún momento se pierde la conexión, el mensaje mostrado será “No conectado” en tono rojo.



Figura 33 Dispositivo conectado y listo para recibir comandos



Figura 34 Se ha perdido la conexión con el dispositivo

3.2.2.3 Programación de bloques

A continuación se detalla la programación de bloques desarrollada para la aplicación ArdDrone.

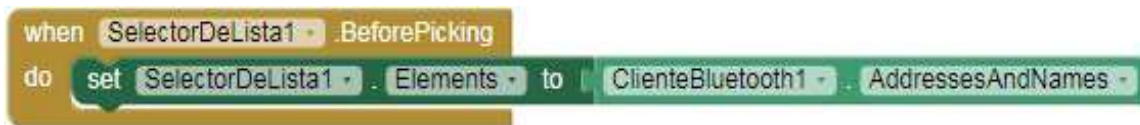


Figura 35 Programación de bloques es App Inventor Blocks Editor

Al

iniciarse la aplicación, se genera la lista de dispositivos Bluetooth accesibles.

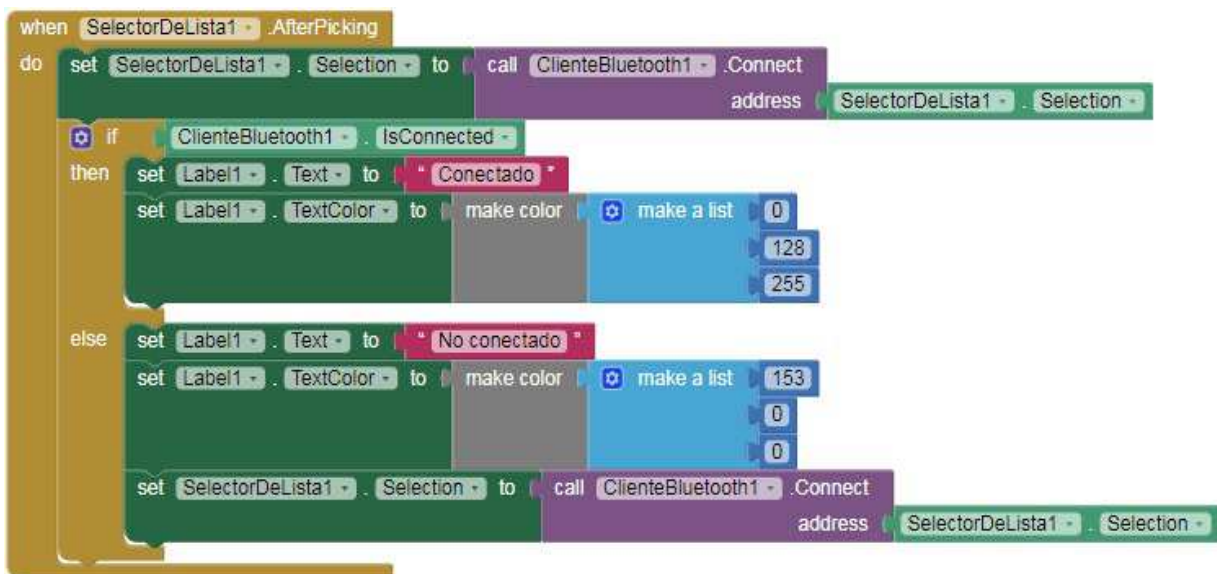


Figura 36 Programación de bloques es App Inventor Blocks Editor

Cuando se acciona el botón “Conectar”, se muestra la lista de dispositivos disponibles. Al seleccionar el dispositivo con el que se desea establecer la conexión, se retorna al interfaz original, y aparece la palabra “Conectado” sobre el botón mencionado; si en cualquier momento se pierde la conexión, el mensaje mostrado cambia a “No conectado”, en tono rojo, y se intenta conectar con el dispositivo seleccionado al arranque.



Figura 37 Programación de bloques en App Inventor Block Editor



Figura 38 Programación de bloques en App Inventor Block Editor



Figura 39 Programación de bloques en App Inventor Blocks Editor

Al accionar cada uno de los botones disponibles, se enviará mediante el módulo Bluetooth un carácter determinado para cada acción solicitada, que se transformará a través de la programación en Arduino en una modificación de la velocidad de los motores.

Capítulo 4

Sensores

4.1 MPU-6050. Funcionamiento

Como se introdujo en el Capítulo 2: Selección de componentes, en nuestro proyecto emplearemos la IMU MPU-6050, de *InvenSense*.

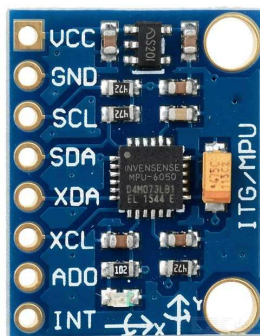


Figura 40 IMU MPU6050,
de *InvenSense*

http://imgdxcn.com/productimages/sku_154602_2.jpg

Se trata de un sensor muy económico y ampliamente utilizado en la actualidad.

La MPU-60X0 es el primer dispositivo de monitorización de movimiento de 6 ejes que combina un giroscopio y un acelerómetro, de 3 ejes cada uno, y un Digital Motion Processor™ (DMP) en un PCB de 4x4x0.9mm. Con su sensor bus de I²C, acepta directamente entradas de una brújula externa de 3 ejes para proporcionar una salida de MotionFusion™ completa. A través de su puerto auxiliar I²C también se pueden integrar otros sensores digitales no inerciales, como sensores de presión.

Presenta flexibilidad en la fuente de alimentación, operando en un rango de voltaje VDD de entre 2.375 V y 3.46 V. Además, la MPU-6050 presenta un pin VLOGIC, que establece los niveles lógicos del interfaz I²C, necesitando un voltaje de 1.8V±5% o VDD.

Para más información, consultar el **Anexo 2: Datasheet MPU-60X0**.

Para comprender el funcionamiento de estos sensores, es necesario entender los principios físicos que miden, que se explican a continuación.

4.1.1 Sistemas de coordenadas

El sistema de coordenadas más habitual en nuestra cultura es el cartesiano, en el que para definir un punto en un espacio de tres dimensiones se emplea la proyección del mismo en tres ejes planos perpendiculares entre sí, quedando así definido por tres coordenadas (x,y,z) con respecto al sistema de referencia. De esta forma podríamos definir la posición en el espacio de, por ejemplo, del centro de gravedad de un avión. Sin embargo, debemos definir también su orientación, para lo que necesitamos conocer los ángulos de desviación respecto a los ejes de referencia, que se denominan Yaw, Pitch y Roll (Guiñada, Cabeceo y Balanceo en castellano).

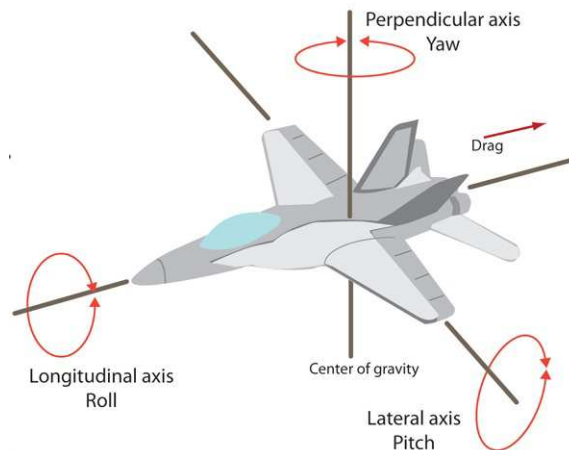


Figura 41
Ángulos de desviación respecto a los ejes de referencia

<http://machinedesign.com/site-files/machinedesign.com/files/uploads/2014/06/PRY.gif>

Para mantener la estabilidad de un sistema móvil, además de su posición en el espacio necesitamos conocer las fuerzas por las que se ve afectado para actuar en consecuencia. Para ello se emplean dos tipos de sensores que colectan distinta información: acelerómetros y giróscopos.

El principio de funcionamiento de giróscopos y acelerómetros se fundamenta en la primera ley de Newton o ley de la inercia, según la cual todo objeto en movimiento tiende a mantener su estado (Inercia) hasta que una fuerza externa lo perturbe.

Un sistema de referencia que no está sometido a ninguna fuerza externa se denomina sistema de referencia inercial. Si aplicamos una fuerza externa, nuestro sistema de referencia deja de ser inercial por definición y percibimos unas fuerzas opuestas a las aplicadas que tienden a mantener nuestro movimiento previo. Para detectar este tipo de fuerzas usamos los acelerómetros.

4.1.2 Acelerómetros

Un acelerómetro detecta las fuerzas de inercia que afectan al sensor y las descomponen en las tres direcciones principales de referencia.

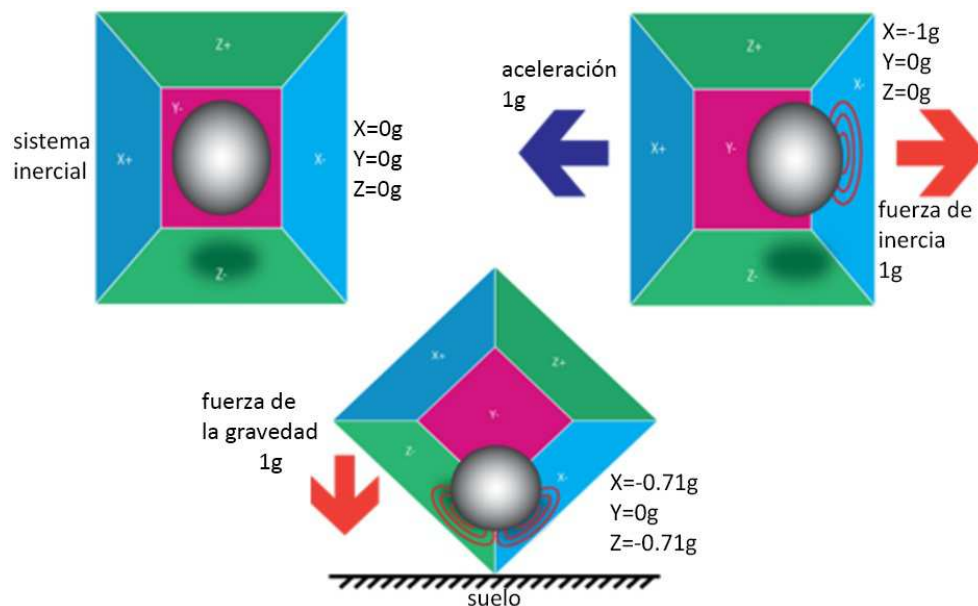


Figura 42 Estructura interna de un acelerómetro mecánico
<http://www.iescamp.es/miarduino/wp-content/uploads/2016/05/acc.png>

Cuando un acelerómetro es sometido a una fuerza lateral en una dirección, el sensor interno (la bolita en la imagen) tiende a retrasarse en la dirección contraria a la de la fuerza que actúa y al detectar en qué paredes choca y con qué intensidad, podemos calcular la dirección e intensidad de la fuerza aplicada.

Hoy los acelerómetros han pasado de ser mecánicos, con bolitas en su interior, a fabricarse como dispositivos electrónicos integrados, y se basan en el efecto piezoeléctrico, que provoca pequeñas corrientes inducidas cuando la inercia deforma ligeramente cristales diseñados para ello.

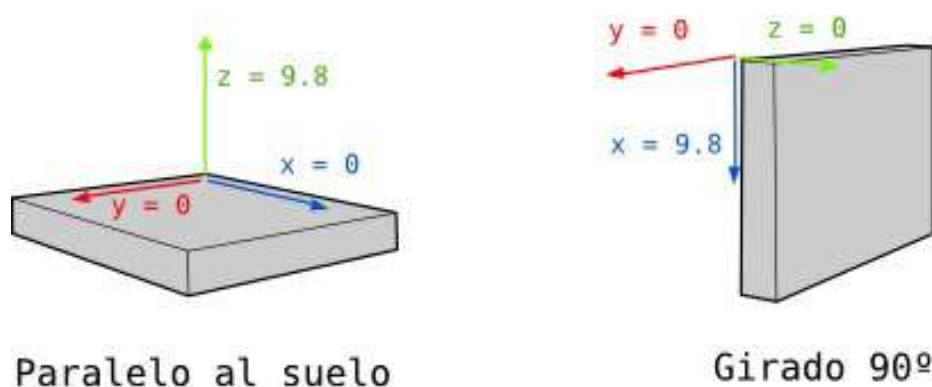


Figura 43 Fuerza de la gravedad sobre un acelerómetro
http://robologs.net/wp-content/uploads/2014/10/imu_ac_ejemplos.png

Sabiendo que la gravedad es 9.8 m/s^2 , y conociendo la medida en los tres ejes del acelerómetro, por trigonometría es posible calcular el ángulo de inclinación de la IMU. Una buena fórmula para calcular el ángulo es:

$$\phi_x = \tan^{-1}\left(\frac{y}{\sqrt{x^2 + z^2}}\right) \quad \phi_y = \tan^{-1}\left(\frac{x}{\sqrt{y^2 + z^2}}\right)$$

4.1.3 Giróscopos

Los giróscopos detectan la rotación del sensor, que no es fácilmente calculable empleando las medidas obtenidas por acelerómetros. Los más modernos detectan las fuerzas centrífugas que el giro provoca alrededor de los tres ejes principales, devolviendo un cero cuando no exista giro. Un giro continuo, a velocidad angular constante, provocará una medida continua en el giróscopo.

En un giróscopo, la velocidad de giro se puede medir en revoluciones por segundo o en grados por segundo ($^\circ/\text{seg}$).

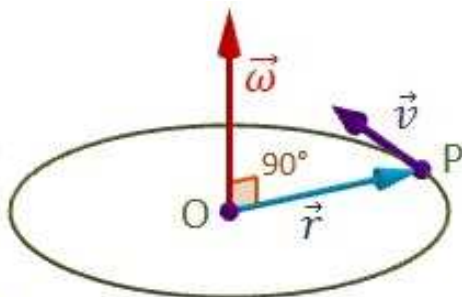


Figura 45

Velocidad angular

<http://www.universoformulas.com/imagenes/fisica/cinematica/velocidad-tangencial-velocidad-angular.jpg>

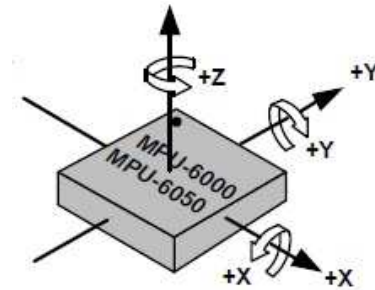


Figura 46 Orientación de los ejes en la MPU-60X0

MPU/6050 Datasheet

Sabiendo el ángulo inicial de la IMU, podemos sumarle el valor que marca el giróscopo para saber el nuevo ángulo en cada momento. Supongamos que iniciamos la IMU a 0° . Si el giróscopo realiza una medida cada segundo, y marca 3 en el eje X, tendremos el ángulo con esta sencilla fórmula:

$$\theta_{y+1} = \theta_y + \text{lectura}_y \cdot \Delta t$$

Donde

θ_{y+1} es el último ángulo que queremos calcular

θ_y es el último ángulo calculado

lectura_y es la lectura del ángulo Y del giróscopo

Δt es el tiempo transcurrido entre cálculo de ángulos

El procedimiento es el mismo para los ejes X y Z. Sin embargo, no es posible calcular el ángulo Z con el acelerómetro, siendo necesario para ello utilizar un magnetómetro.

4.1.4 Error en las medidas

Al emplear sensores para tomar medidas de cualquier tipo, encontramos dos problemas que separan el valor obtenido del real: el ruido y los errores.

El ruido incluye todas aquellas interferencias que afectan a los dispositivos electrónicos.

El acelerómetro es capaz de medir cualquier ángulo, pero sus lecturas son ruidosas y tienen un cierto margen de error. Además, detecta cualquier aceleración que no sea la de la gravedad, por lo que si se mueve la IMU sin girarla, al aplicar una aceleración en otro eje, la IMU lo detectará como un cambio de rotación.

A diferencia del acelerómetro, el giróscopo obtiene las medidas con mucha precisión. Pero al realizar los cálculos del ángulo es inevitable que se produzca un pequeño error, que se acumula con cada cálculo, lo que en inglés se denomina *drift*.

Una manera de minimizar el ruido y los errores es empleando un filtro. Debemos conseguir eliminar el ruido y el drift y conseguir que el acelerómetro no cambie de ángulo al detectar otra fuerza que no sea la gravedad.

Utilizaremos el Filtro Complementario o Complementary Filter, ideal para implementar con Arduino: es fácil de utilizar, con bajo coste de procesamiento y posee una muy buena precisión.

En realidad se trata de la unión de dos filtros diferentes: un High-Pass Filter (que deja pasar los valores por encima de un determinado límite) para el giróscopo y un Low-Pass Filter (que permite el paso a los valores por debajo de un límite establecido) para el acelerómetro.

La fórmula resultante de combinar los dos filtros es:

$$\alpha_{i+1} = 0.98 \cdot (\alpha_i + \varphi_{\text{giro}} \cdot \Delta t) + 0.02 \cdot \varphi_{\text{acel}}$$

Donde

α_{i+1} es el ángulo que queremos calcular

α_i es el último ángulo calculado

φ_{giro} es el ángulo del giróscopo previamente calculado

φ_{acel} es el ángulo del acelerómetro previamente calculado

Δt es el tiempo transcurrido entre cálculo de ángulos

El procedimiento es el mismo para los ejes X e Y.

Los valores 0.98 y 0.02 pueden variar, siempre y cuando el resultado de sumar ambos sea 1.

4.1.5 Protocolo I²C

El estándar Inter Integrated Circuits bus, o IIC (I²C), presenta las siguientes características:

- Protocolo de dos hilos de control:
 - SDA, para transmitir los datos
 - SCL, el reloj asíncrono que indica cuándo leer los datos
 - GND y 5V, cuando se requiera
- Cada dispositivo conectado al bus I²C tiene su dirección exclusiva, de 7 bits, por lo que en teoría se pueden conectar $2^7 = 128$ dispositivos
- Uno de estos componentes debe actuar como master, es decir, controla el reloj, y establece su velocidad
- Es multi-master, lo que significa que el master puede cambiar, pero solo uno puede estar activo a la vez. Proporciona un protocolo de arbitraje y detección de colisiones.

Arduino soporta el protocolo I²C de fábrica con una librería estándar (Wire), que utiliza dos de los pines analógicos para las funciones SDA (Datos) y SCL (Clock).

En el Arduino UNO, los pines I²C están en los pines analógicos A4 (SDA) y A5 (SCL).

La dirección de la IMU utilizada, especificada en la documentación oficial de la MPU-6050, es 0x68.

Comunicación

La comunicación en el bus I²C comienza cuando el maestro establece la condición START (S) en el bus, que se define como una transición HIGH-LOW de la línea SDA mientras la línea SCL está en HIGH. El bus está ocupado hasta que el maestro establece la condición STOP (P) en el bus, pasando SDA de LOW a HIGH mientras SCL está HIGH.

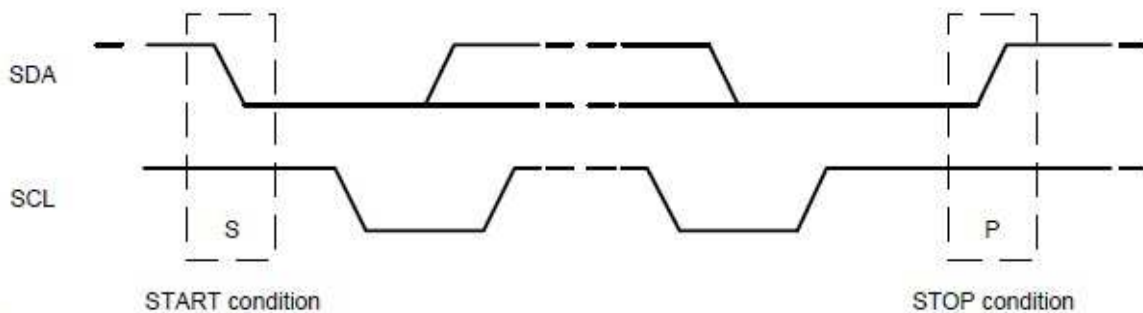


Figura 57 MPU-6050: Comunicación I²C. Condición START/STOP

MPU-6000 Datasheet

Los datos se transfieren en un byte en I²C (8 bits). No hay límite de bytes transmitidos en cada transferencia de datos. Cada byte transmitido debe ir seguido de una señal acknowledge (ACK), cuyo reloj es generado por el maestro, mientras que el receptor genera la señal de acknowledge real manteniendo la línea SDA en estado bajo durante del pulso de reloj correspondiente al acknowledge.

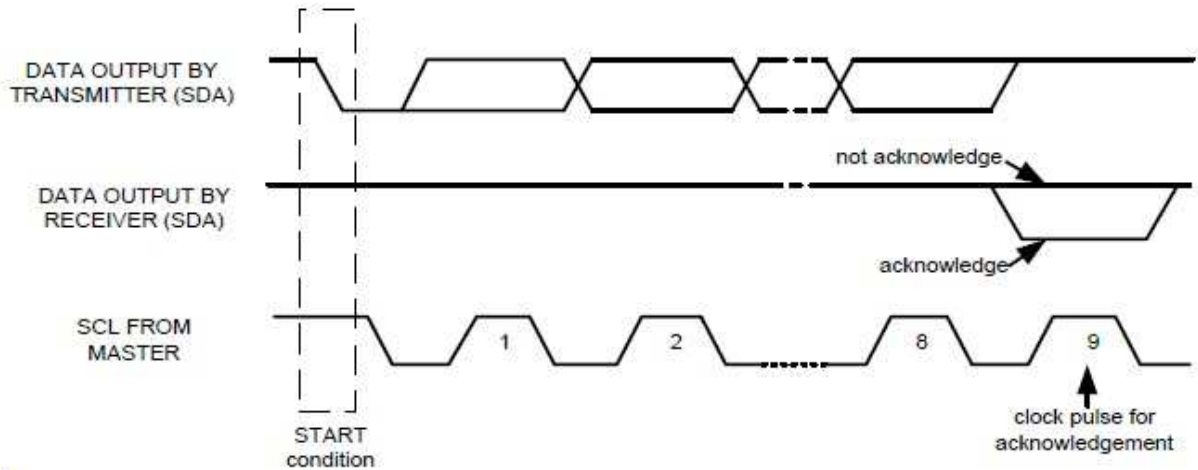


Figura 58 MPU-6050: Señal de acknowledge
MPU-6000 Datasheet

Tras comenzar la comunicación con la condición de START, el maestro envía una dirección de esclavo en 7 bits seguida de un octavo bit de lectura/escritura. Este último bit indica si el maestro está escribiendo o recibiendo datos del esclavo. Entonces el maestro libera la línea SDA y espera a la señal de acknowledge del esclavo. La transmisión finaliza con la condición de STOP (P), liberando la línea de comunicación. Sin embargo, el maestro puede establecer una nueva transmisión sin pasar por la condición de STOP.

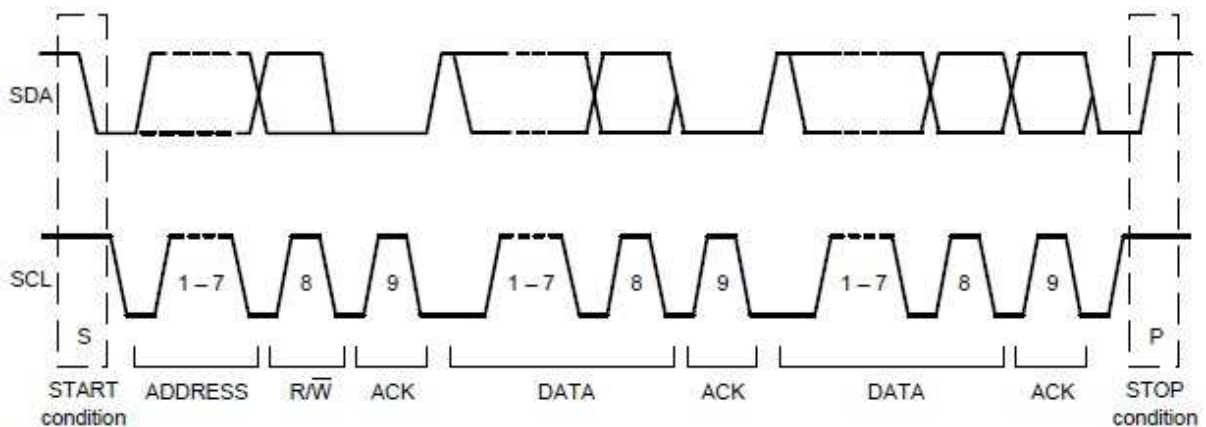
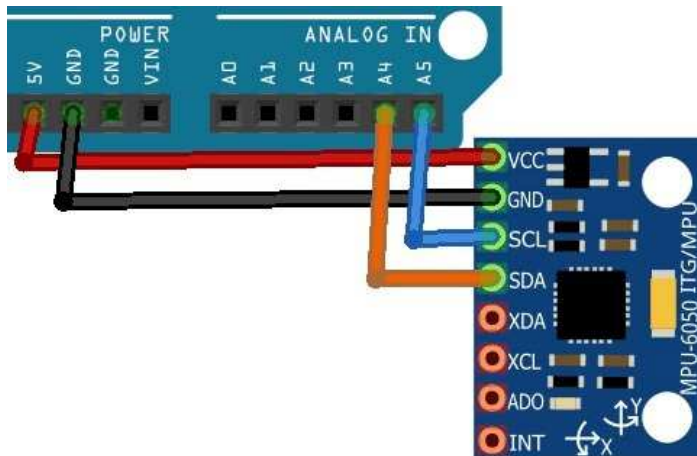


Figura 59 MPU-6050: Transferencia de datos I²C
MPU-6000 Datasheet

4.1.6 Conexión

Las conexiones son muy sencillas.



MPU-6050	Arduino
VCC	Salida 5V
GND	GND
SCL	Pin A5
SDA	Pin A4

Figura 60 Conexiones de Arduino con MPU-6050

VCC se conecta a la salida de 5V de Arduino, mientras que las entradas analógicas A4 y A5 servirán para recibir los datos de la IMU.

4.1.7 Implementación en Arduino

El MPU-6050 dará unos raw values (“valores en bruto”) que después habrá que refinar (dividir por una constante) a fin de conseguir valores utilizables.

La web oficial de Arduino dispone de un ejemplo para leer datos de la MPU-6050, del que se parte para llevar a cabo la programación.

Para la parte de obtención de valores de la IMU del proyecto se parte del código escrito por el autor del artículo “Tutorial de Arduino y MPU-6050” de la página robologs.net, y se explica a continuación.

```
#include <Wire.h>

//Direccion I2C de la IMU
#define MPU 0x68

//Ratios de conversion
#define A_R 16384.0
#define G_R 131.0

//Conversión de radianes a grados (180/PI)
#define RAD_A_DEG = 57.295779

//Valores medidos por la MPU-6050
int16_t AcX, AcY, AcZ, GyX, GyY, GyZ;

//Angulos
float Acc[2];
float Gy[2];
float Angle[2];
```

La librería *Wire* gestiona el protocolo de comunicaciones en I²C

Figura 61 MPU-6050. Programación en Arduino

```

void setup()
{
Wire.begin();
Wire.beginTransmission(MPU);
Wire.write(0x6B);
Wire.write(0);
Wire.endTransmission(true);
Serial.begin(9600);
}

void loop()
{
//Lectura de los valores del Acelerometro
Wire.beginTransmission(MPU);
Wire.write(0x3B); //Pedir el registro 0x3B - corresponde al AcX
Wire.endTransmission(false);
Wire.requestFrom(MPU, 6, true); //A partir del 0x3B, se piden 6 registros
AcX=Wire.read()<<8|Wire.read(); //Cada valor ocupa 2 registros
AcY=Wire.read()<<8|Wire.read();
AcZ=Wire.read()<<8|Wire.read();

//A partir de los valores del acelerometro, se calculan los angulos Y, X
//respectivamente, con la formula de la tangente.
Acc[1] = atan(-1*(AcX/A_R)/sqrt(pow((AcY/A_R),2) + pow((AcZ/A_R),2)))*RAD_TO_DEG;
Acc[0] = atan((AcY/A_R)/sqrt(pow((AcX/A_R),2) + pow((AcZ/A_R),2)))*RAD_TO_DEG;

//Lectura de los valores del Giroscopio
Wire.beginTransmission(MPU);
Wire.write(0x43);
Wire.endTransmission(false);
Wire.requestFrom(MPU, 4, true); //Solo se piden 4 registros
GyX=Wire.read()<<8|Wire.read();
GyY=Wire.read()<<8|Wire.read();

//Calculo del angulo del Giroscopio
Gy[0] = GyX/G_R;
Gy[1] = GyY/G_R;

//Aplicar el Filtro Complementario
Angle[0] = 0.98 *(Angle[0]+Gy[0]*0.010) + 0.02*Acc[0];
Angle[1] = 0.98 *(Angle[1]+Gy[1]*0.010) + 0.02*Acc[1];

//Mostrar los valores por consola
Serial.print("Angle X: "); Serial.print(Angle[0]); Serial.print("\n");
Serial.print("Angle Y: "); Serial.print(Angle[1]); Serial.print("\n-----\n");

delay(10); //Para las fórmulas, el intervalo de tiempo entre cálculos será 0.01
}

```

Figura 62 MPU-6050. Programación en Arduino

En control de estabilidad se lleva a cabo a través de las sucesivas lecturas de la IMU. Se emplearán los sucesivos ángulos calculados para actuar sobre el motor correspondiente una vez de superan los 5° de desviación.



Figura 63 Ángulo de desviación

La actuación sobre la velocidad de los motores será proporcional al ángulo calculado, con lo que cuanto mayor sea la desviación, mayor será la disminución de velocidad en los motores del lado contrario.

$$\text{Vel_motor} = \text{Velocidad} - (0.01 * \text{Angulo_calculado}) * \text{Velocidad}$$

4.2 Fotorresistencia (LDR)

La fotorresistencia es un componente electrónico que posee una resistencia interna variable en función de la luz que incide sobre ella. La resistencia disminuye si la intensidad de la luz aumenta, y se incrementa con el descenso de la misma.

También se conoce como fotoconductor, célula fotoeléctrica o resistor dependiente de la luz. Las fotorresistencias convencionales se conocen como LDR (Light-Dependent-Resistor).

Para más información, consultar el **Anexo 3: Datasheet GL5528**.

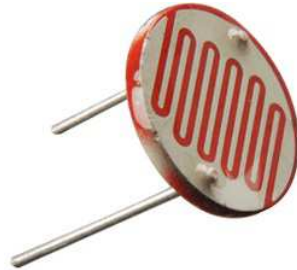


Figura 64
Fotorresistencia

<https://www.prometec.net/wp-content/uploads/2016/01/ldr-1.jpg>

4.2.1 Conexión

Emplearemos un divisor de tensión para monitorizar el voltaje en una resistencia de valor conocido. El valor de la LDR es inversamente proporcional a la intensidad lumínica que incide sobre ella, con lo que al disminuir la luz la tensión leída a la salida aumentará.

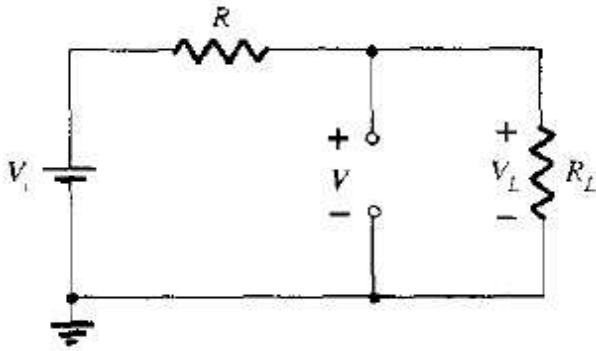


Figura 65 Divisor de tensión

Robert L. Boylestad; Louis Nashelsky.
"ELECTRÓNICA: TEORÍA DE CIRCUITOS"

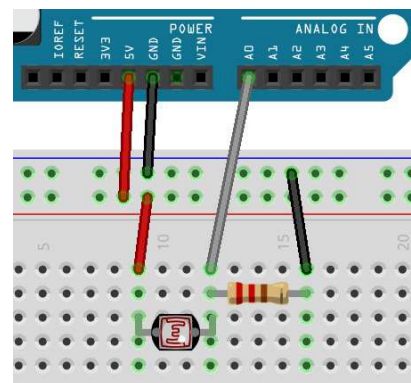


Figura 66 Conexión LDR

Para construir el divisor de tensión, una patilla de la LDR debe conectarse a la salida de 5 V de Arduino, mientras que la otra va a una resistencia antes de acabar en GND. La medida se hace a la salida de la LDR.

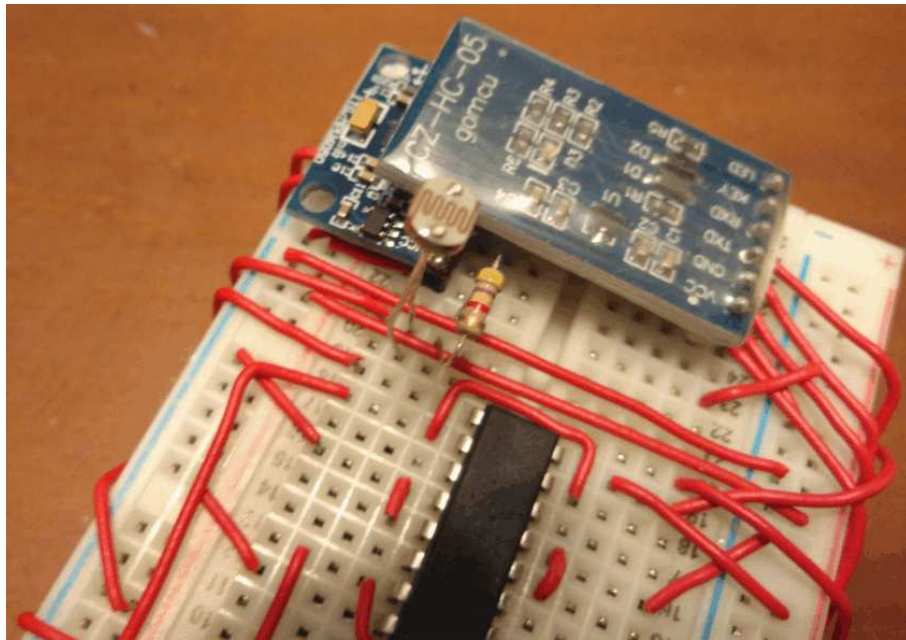


Figura 67 Montaje LDR

4.2.2 Cálculos

Vamos a calcular los valores de voltaje máximo y mínimo que obtendremos al utilizar la GL5528 en un divisor de tensión en función de la variación de la resistencia.

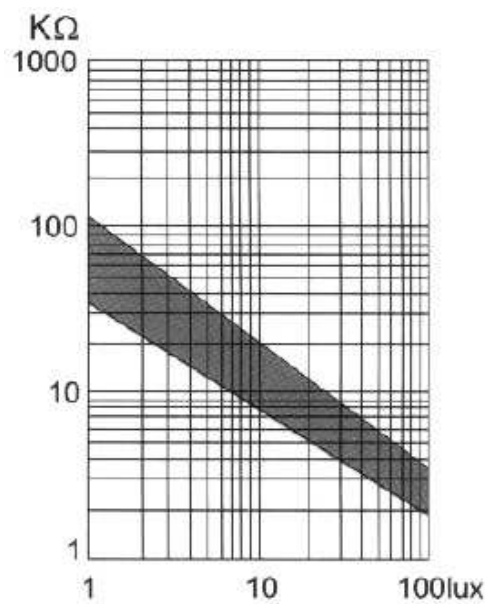


Figura 68 Nivel de luminosidad frente a resistencia

GL5528 Datasheet

Como se observa en la gráfica, el valor máximo de la resistencia a nivel mínimo de luminosidad es de unos 100 kΩ; el valor mínimo, a máximo nivel de luminosidad, estará alrededor de 1 kΩ .

Si empleamos una resistencia de 4,7 kΩ (R en nuestra fórmula) para el divisor de tensión, obtenemos que para plena luminosidad se leerá un valor de unos 2,35 V, mientras que con luminosidad nula el valor de entrada será de unos 4,12 V.

$$V_{in} = R \cdot \frac{V_{Arduíno}}{R + R_{LDR}} = 4,7 \cdot 10^3 \cdot \frac{5}{4,7 \cdot 10^3 + R_{LDR}}$$

Las entradas analógicas del modelo UNO de Arduino leen valores de tensión de 0 a 5 V con una resolución de 1024 (10 bits). Esto quiere decir que el pin nos devolverá un valor entre 0 y 1023, correspondiéndose éstos con 0 y 5 V. De esta forma podremos establecer un valor límite a partir del cual encenderemos o apagaremos los LEDs.

4.2.3 Código

Se conecta la salida de la LDR a la entrada analógica A0 de Arduino, y mostramos el valor leído.

```
#include <SoftwareSerial.h>

#define LDR 0
int valorLDR = 0;

void setup() {
  Serial.begin(9600);
}

void loop() {
  valorLDR = analogRead(LDR);
  Serial.println(valorLDR);
}
```

Figura 69 Código LDR

Capítulo 5

Actuadores

5.1 Motores

Los motores eléctricos se basan en interacciones electromagnéticas para transformar energía eléctrica en mecánica. Son los elementos mediante los que el VANT consigue elevarse en el aire y realizar los desplazamientos correspondientes en cada momento, así como mantener y recuperar la estabilidad tras la actuación de una fuerza externa o la ejecución de una orden del usuario.

En el mercado encontramos distintos tipos de motores que pueden ser utilizados en la construcción de un drone, y cada uno de ellos debe ser controlado mediante un procedimiento distinto. La selección dependerá principalmente del peso y aplicación de la aeronave.

5.1.1 Motores DC brushless

Los motores más utilizados en los drones son los de tipo brushless (sin escobillas). Son más eficientes que los motores con escobillas, ya que presentan un menor rozamiento y consumo, lo que se traduce en un calentamiento y mantenimiento necesario inferiores. Se alimentan con corriente alterna (de ahí los 3 cables de cada motor), para lo que es necesario un ESC (Electronic Speed Controller) o variador de velocidad que se encarga de transformar la corriente continua en corriente alterna a una determinada frecuencia para configurar la velocidad adecuada en cada momento.



Figura 70 Motor DC brushless

<http://www.robotshop.com/media/catalog/product/cache/1/thumbnail/9df78eab33525d08d6e5fb8d27136e95/u/a/uav-brushless-motor-a2208-17-1100kv.jpg>



Figura 71 ESC (Electronic Speed Controller)

<http://site.hobbypartz.com/80A-ESC.jpg>

5.1.2 Motores DC con escobillas. L293D

Este tipo de motores requieren de conmutadores rotativos para generar el campo magnético adecuado, y la velocidad de giro depende generalmente de la tensión aplicada.

En el presente proyecto se utilizan cuatro motores DC con escobillas de los que se disponía con anterioridad. Para controlarlos a través de Arduino, necesitaremos emplear un driver que permita proporcionar más corriente al motor, ya que la corriente máxima que suministran las salidas digitales es de 40 mA. De esta manera podremos alimentar los motores con una batería externa.

El control de los motores se lleva a cabo mediante el circuito integrado L293D, de *Texas Instruments*. Este control es necesario porque la potencia entregada por Arduino no es suficiente para alimentar los 4 motores además del resto de módulos del proyecto, con lo que la alimentación de los motores se realiza a través de este componente.

Con un solo circuito integrado es posible realizar el control de dos motores, por lo que para controlar los motores del cuadricóptero desarrollado en el presente proyecto serán necesarios dos circuitos L293D. Este circuito integrado presenta 16 pines:

Pin/es	Propósito
1, 9	Habilitación de los motores. Se pueden conectar a salidas PWM de Arduino para controlar la velocidad.
8	Alimentación de los motores.
16	Alimentación de la electrónica del circuito. Debe conectarse a la salida de 5V de Arduino.
4, 5, 12, 13	Masa. Deben conectarse a GND.
O1, O2	Puntos de conexión del motor 1.
O3, O4	Puntos de conexión del motor 2.
I1, I2	A través de pines de salida de Arduino, permiten configurar el sentido de giro del motor 1 de la siguiente manera: I1 I2 Acción 1 0 Giro a derechas 0 1 Giro a izquierdas 0 0 Motor parado 1 1 Motor parado
I3, I4	Permiten configurar el sentido de giro del motor 2 de igual manera que I1, I2.

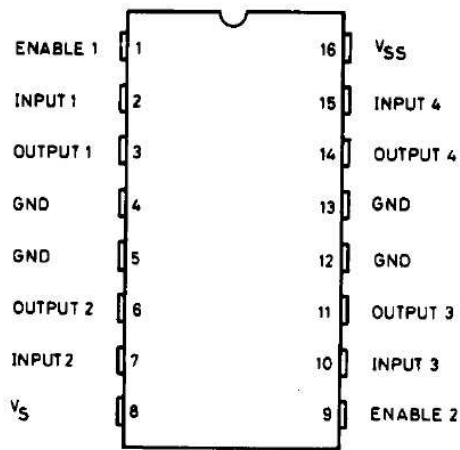


Figura 72 Esquemático del circuito integrado L293D
Datasheet L293D

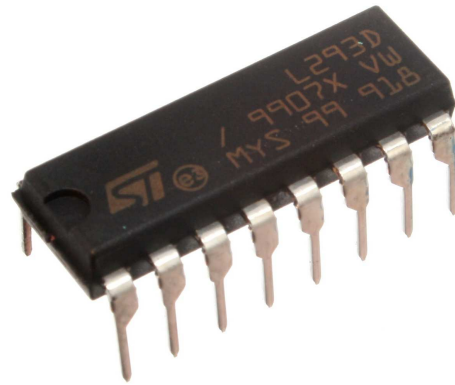


Figura 73 Circuito integrado L293D
<http://www.prometec.net/wp-content/uploads/2015/03/circuito-integrado-L293D.jpg>

Se puede encontrar más información sobre la estructura interna y el funcionamiento en el **Anexo 4: Datasheet L293D**.

5.1.3 Conexión

Los pines PWM de Arduino se pueden utilizar como salidas analógicas. El valor de tensión de salida se codifica en un byte (8 bits), con lo que en la programación se escribirá un número entre 0 y 255 ($2^8 = 256$) para modelar la tensión de salida entre 0 y 5 V.

La patilla 16 se conecta a los 5 V de Arduino para alimentar el circuito, y los motores se alimentan a través de la patilla 8 con una batería Li-Po de 500 mAh y 3.7 V.

Un motor se conecta a las salidas O1, O2, que operará en función de la velocidad comandada mediante el pin 5 de Arduino (PWM), mientras que el otro se conecta a O3 y O4, operado por el pin 3.

Las masas se conectan a GND.

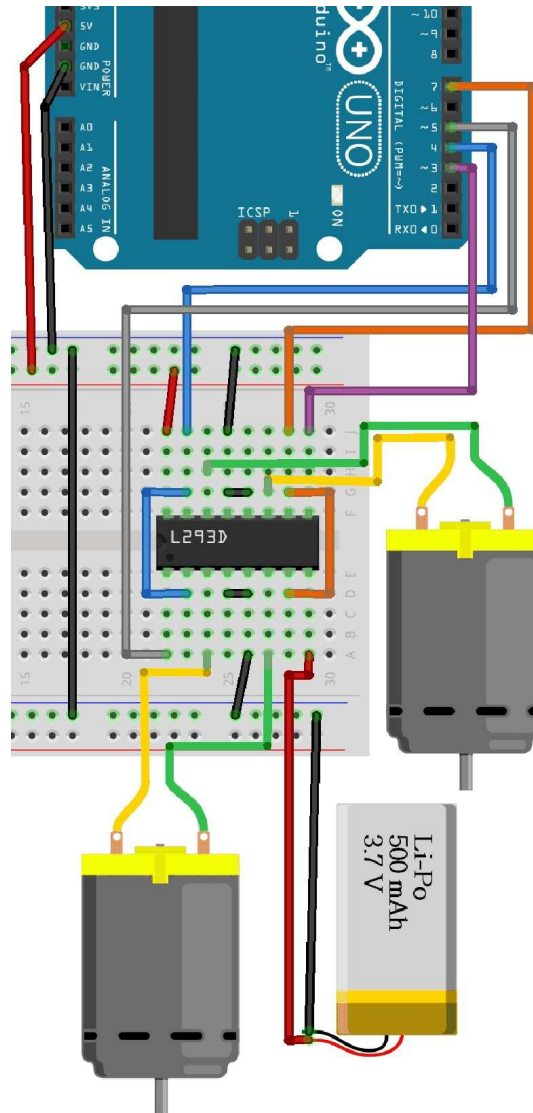


Figura 74 Conexiones L293D con dos motores

5.2 Diodos LED

El diodo emisor de luz o LED (Light Emitting Diode) es un dispositivo electrónico de unión p-n con dos terminales que sólo permite el paso de corriente en una dirección. El paso de la corriente se produce de ánodo a cátodo, como se denominan estos dos terminales.

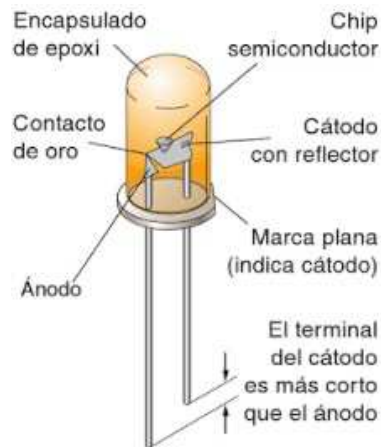


Figura 75 Esquema de un diodo LED

A. Carretero, F. J. Ferrero, J. A. Sánchez-Infantes, P. Sánchez-Infantes, F. J. Valero: "ELECTRÓNICA"

En todas las uniones p-n de semiconductor, parte de la energía que se transfiere en cada recombinación electrón-hueco se emite como calor y otra parte en forma de fotones, que en el caso de los LEDs es suficiente para crear una fuente de luz muy visible, fenómeno que se denomina electroluminiscencia.

La intensidad mínima para que un diodo LED emita luz visible es de 4 mA y, por precaución, como máximo debe aplicarse 50 mA. El valor típico de intensidad que se suele utilizar es de 10 mA.

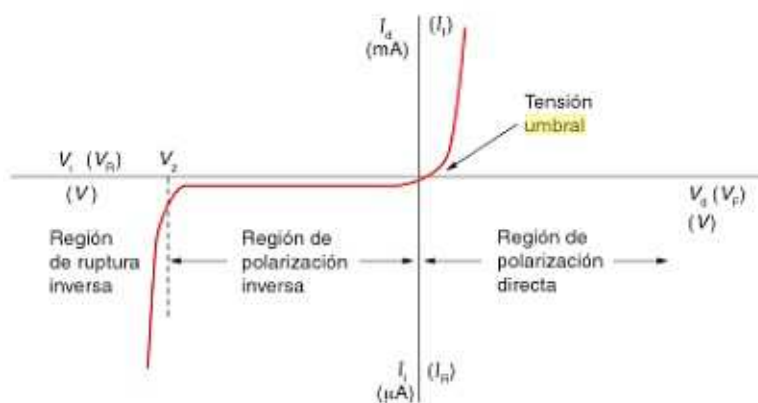


Figura 76 Curva característica de un diodo

A. Carretero, F. J. Ferrero, J. A. Sánchez-Infantes, P. Sánchez-Infantes, F. J. Valero: "ELECTRÓNICA"



Figura 77 Diodo LED

Robert L. Boylestad; Louis Nashelsky. "ELECTRÓNICA: TEORÍA DE CIRCUITOS"

Radiación luminosa	Infrarrojo	Rojo	Naranja	Ámbar	Verde	Azul
Tensión umbral típica	1,3 V	1,7 V	2,0 V	2,5 V	2,5 V	4,0 V

El modelo X5C de Syma aprovechado del que se disponía con anterioridad incluía unos diodos LED que se han decidido utilizar para ser activados o desactivados en función de la intensidad lumínica ambiental medida a través de una LDR incorporada en el dispositivo. Además, su estado se invertirá momentáneamente cuando se reciba un comando a través del módulo de Bluetooth para indicar que la orden se ha transferido con éxito.



Figura 78 LEDs

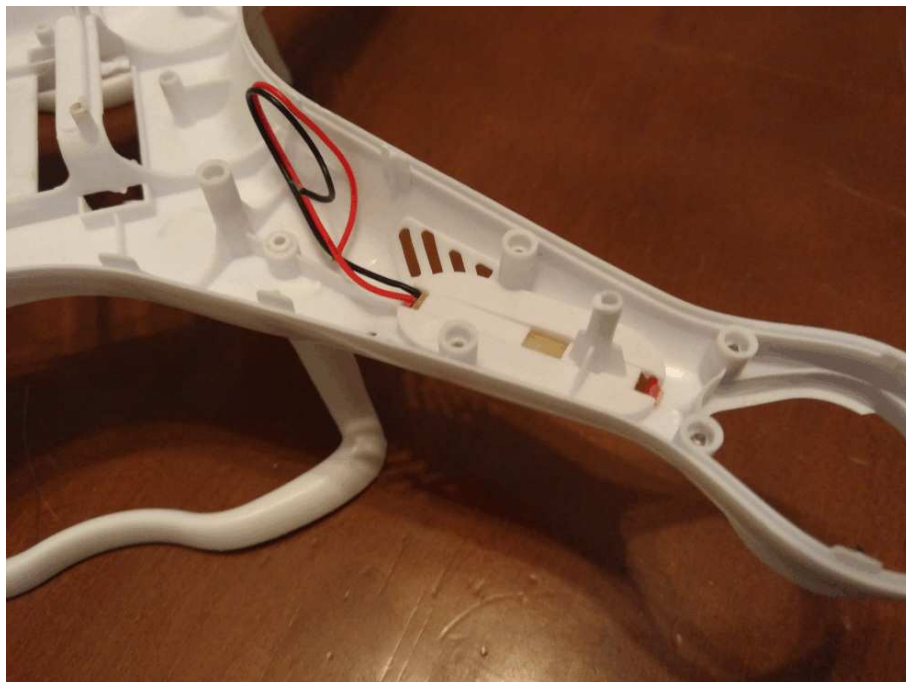


Figura 79 Diodos LED

5.2.1 Conexión

Las conexiones son muy sencillas:

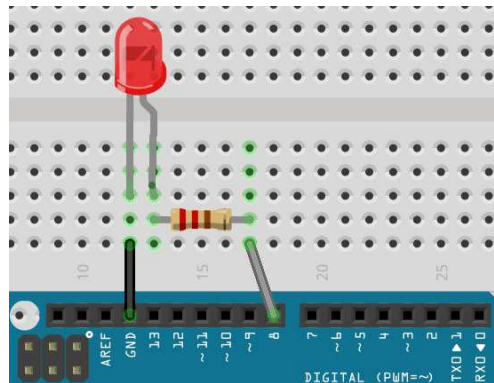


Figura 80 Conexión diodo LED

La salida digital de Arduino se conecta a una resistencia que provocará una caída de tensión antes de conectarse a los diodos LED en paralelo, cerrando el circuito en GND.

5.2.2 Cálculos

Como ya hemos visto, necesitaremos una tensión de unos 2,5 V en cada uno de los diodos para que se iluminen, y disponemos de 40 mA como máximo; por ello, el montaje se hará con dos líneas en paralelo de dos LEDs en serie por línea. De esta manera, los 5 V del pin se repartirán entre ambas líneas, con lo que obtendremos 2,5 V por LED sin necesidad de emplear resistencia alguna.

5.2.3 Código

```
#include <SoftwareSerial.h>

#define LDR 0
#define LEDES 8
int valorLDR = 0;

void setup() {
  pinMode(LEDES, OUTPUT);
  digitalWrite(LEDES, LOW);
  Serial.begin(9600);
}

void loop() {
  valorLDR = analogRead(LDR);
  Serial.println(valorLDR);

  if(valorLDR < 1000) {
    digitalWrite(LEDES, LOW);
  }
  if(valorLDR > 1000) {
    digitalWrite(LEDES, HIGH);
  }
}
```

El control se ha programado para que los LEDs se enciendan cuando la LDR mide un valor por encima de un límite, lo que significa que la iluminación ambiental es baja.

Figura 81 Código diodo LED

Capítulo 6

Comunicación

Existen diferentes sistemas de comunicación inalámbrica disponibles para establecer una transmisión de datos entre el usuario y un VANT. A continuación se presentan algunos de ellos.

6.1 Radio-control

El método de comunicación más extendido en el mundo de los VANT es el radio-control. Al accionar las entradas deseadas, el transmisor envía una señal de radio que recibe el receptor, colocado dentro del modelo. Los datos recibidos son descodificados y transformados en impulsos eléctricos que actúan sobre los servos o actuadores que provocan el movimiento del drone.

6.2 Hyperion

La transmisión de datos vía radio es limitada, y es una comunicación vulnerable a interferencias. El sistema Hyperion utiliza un haz de luz láser, lo que proporciona una alta velocidad, no requiere licencia y es inmune a interferencias; además, no puede ser interceptado ni hackeado por parte de terceros.

Este sistema pretende solventar el problema de mantener el haz láser dirigido a la aeronave en vuelo, utilizando tecnología retro reflexiva de baja potencia. Para ello, se ha diseñado el SLM o modulador espacial de luz, que permite la dirección horizontal y vertical precisa del haz láser, así como el control de la divergencia del propio haz y el enfoque del mismo, calculando 35 veces por segundo los ángulos de dirección.

El proyecto tiene el potencial de cambiar la manera en que creamos nuestros enlaces de datos, permitiendo el transporte aéreo más seguro y operaciones de drones más eficientes.



Figura 82 Proyecto Hyperion

<http://www.dronepedia.es/wp-content/uploads/2016/03/hyperion-400x260.jpg>

6.3 Satélite

La Universidad de Vigo ha desarrollado una tecnología centrada en un nuevo sistema de comunicaciones embarcado en un drone y su conexión en tiempo real a través de un satélite. El reto de este proyecto, llamado TÁCTICA, reside en mejorar los actuales sistemas de comunicación vía satélite para escenarios de movilidad en el ámbito civil (drones, coches, aviones, barcos), puesto que actualmente ya existen drones en el ámbito militar que incorporan sistemas de comunicación por satélite muy sofisticados.

Para ello, AtlantTIC ha desarrollado algoritmos de adaptación de enlace que permiten adecuar las transmisiones a las condiciones en las que se encuentra el canal. De esta manera se mejora la fiabilidad de la comunicación en las diferentes condiciones de vuelo, y se aumenta la velocidad de transmisión de datos.



Figura 83 Proyecto TÁCTICA

<http://www.dronepedia.es/wp-content/uploads/2015/12/dronepedia-atlantia-c-2-539x303.jpg>

6.4 Sin comunicación

En algunas aplicaciones no es necesario disponer de una comunicación continua con el VANT, siendo suficiente la programación de las especificaciones de despegue, desplazamiento, actuación, regreso y aterrizaje.

Aquí entra en juego la autonomía del dispositivo, que limita enormemente el tiempo de vuelo y, por tanto, de operación.

6.5 Bluetooth

Como ya se comentó con anterioridad, la comunicación entre la aplicación y Arduino se lleva a cabo a través de Bluetooth mediante el HC-05, de fácil uso y SPP (Serial Port Protocol). Es configurable como Maestro (Master) o como esclavo (Slave) mediante comandos AT. Trabaja a 3.3V, pero está adaptado para poder usarlo a 5V, lo que lo hace totalmente compatible con Arduino.

El módulo se define como UART (Universal Asynchronous Receiver-Transmitter), lo que significa que existe transferencia de datos sólo en el momento en el que son registrados, sin que se establezca una transmisión continua de información.

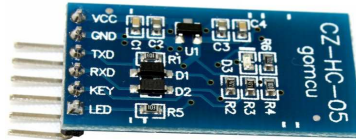


Figura 84 Módulo de comunicación Bluetooth HC-05

<https://img.pccomponentes.com/articles/7/79483/modulo-hc-05-bluetooth-compatible-con-arduino-2.jpg>

Lo primero que se debe hacer al conectar el módulo es configurar el funcionamiento deseado mediante comandos AT, modo que hay que activar mediante un estado alto en el pin KEY. Una vez activado el modo AT, es posible introducir una serie de comandos a través del monitor serie que definirán el comportamiento de nuestro módulo, entre otros:

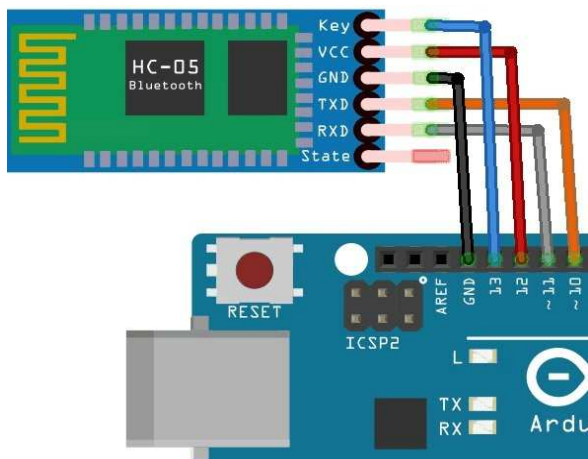
AT	Confirma que el modo AT está activado. Si es así, devolverá "OK"
AT+NAME	Devuelve el nombre del módulo
AT+ROLE	Devuelve un 0 o un 1 en función de si el módulo está configurado en modo Esclavo o Maestro, respectivamente
AT+ROLE=0	Configura el módulo en modo Esclavo. Si se iguala a 1 se configura en modo Maestro
AT+VERSION	Devuelve la versión del firmware
AT+BAUD	Devuelve la velocidad de comunicación, que se establece de la siguiente forma: 1 configura 1200bps 2 configura 2400bps 3 configura 4800bps 4 configura 9600bps (Default) 5 configura 19200bps 6 configura 38400bps 7 configura 57600bps 8 configura 115200bps

AT+BAUD=X	Configura la velocidad de comunicación, siendo X uno de los números correspondientes a las velocidades de la celda anterior
AT+PIN	Devuelve el PIN del módulo
AT+PIN=XXXX	Configura un nuevo PIN para el módulo, siendo el nuevo “XXXX”

Existen otros comandos AT, que se pueden consultar en el **Anexo 5: Datasheet HC-05**.

6.5.1 Conexión

Las conexiones son muy sencillas.



HC-05	Arduino
VCC	Pin 12
GND	GND
TXD	Pin 10
RXD	Pin 11
KEY	Pin 13
LED	-

Figura 85 Conexiones Arduino con módulo de Bluetooth HC-05

El pin 12 de Arduino se utilizará para encender el módulo por programación.

El pin 13, conectado a KEY, establece el modo de operación del HC-05: en nivel alto el módulo entra en modo AT, permitiendo modificar la configuración, mientras que en nivel bajo opera como módulo de comunicación.

Los pines 10 y 11 se conectan a TXD y RXD para que se lleve a cabo la transferencia de información.

6.5.2 Implementación en Arduino

La programación en Arduino para el funcionamiento del HC-05 es similar a la que se necesita desarrollar para la recepción de datos a través del puerto serie conectado al ordenador mediante un cable USB. TXD y RXD se deben conectar cruzados con los pines de comunicación de Arduino, que se utilizarán mediante la librería “Software Serial”.

A continuación se explica el código que enciende y apaga un diodo LED a través del módulo Bluetooth para entender su funcionamiento.

```
#include <SoftwareSerial.h>

char valor; //variable para guardar el dato recibido
int RXD = 10;
int TXD = 11;
int RST = 13;
int KEY = 9;

SoftwareSerial BTSerial(RXD,TXD);

void setup() {
  pinMode(LED, OUTPUT);
  pinMode(RST, OUTPUT);
  pinMode(KEY, OUTPUT);

  //HC-05 apagado
  digitalWrite(RST, LOW);

  //Modo de comunicación (si quisiéramos entrar en
  //en el modo AT, lo pondríamos este pin a HIGH)
  digitalWrite(KEY, LOW);

  //Módulo encendido
  digitalWrite(RST, HIGH);
}
```

Figura 86 HC-05 en Arduino

```
//Configuración del puerto serie de HC-05 y vaciado el buffer
BTSerial.begin(9600);
BTSerial.flush();
delay(500);

//Configuración puerto serie de Arduino
Serial.begin(9600);
}
```

Figura 87 HC-05 en Arduino

```

void loop() {
  //Si el HC-05 envía un dato, lo guardamos en una variable
  if(BTSerial.available()){
    valor = BTSerial.read();

  }
  // Si valor es 'a', el diodo se apaga
  if (valor == 'a') {
    digitalWrite(LED, LOW);
    Serial.println("LED apagado");
  }

  // Si valor es 'e', el diodo se enciende
  else
  if (state == 'e') {
    digitalWrite(LED, HIGH);
    Serial.println("LED encendido");
  }
}
}

```

Figura 88 HC-05 en Arduino

Capítulo 7

Consumo

En cualquier montaje electrónico es muy importante analizar el consumo de los componentes por separado para asegurar el correcto funcionamiento de todos los elementos y evitar sobretensiones que pueden causar daños al conjunto o a los elementos.

En función de la fuente de alimentación que empleemos para el Arduino, dispondremos de un máximo de corriente; en nuestro caso lo alimentaremos a través del puerto USB (*Universal Serial Bus*), que dependiendo de la versión soportará una corriente máxima. Los más comunes en la actualidad son los siguientes:

USB 2.0	500 mA
USB 3.0	900 mA

En cualquiera de los dos casos el límite de corriente disponible será de 500 mA, pues el modelo de Arduino que utilizamos está limitado a esta corriente mediante un fusible.

Recapitulemos la información del consumo de cada elemento por separado.

7.1 Arduino

El Arduino UNO R-3 dispone de un fusible que protege el puerto USB del ordenador de cortes y sobrecargas. Si se demandan más de 500 mA el fusible corta la alimentación automáticamente hasta que el cortocircuito o la sobrecarga se elimina. Esto significa que dispondremos de 500 mA, menos lo que consume el Arduino (unos 46 mA medidos con polímetro), para alimentar el resto de componentes.

Intensidad máxima disponible: 450 mA

Las entradas/salidas digitales de Arduino UNO soportan una corriente máxima de 40 mA, mientras que el pin de 3.3 V resiste hasta 50 mA.

7.2 Módulo Bluetooth

Durante el establecimiento de la conexión, la corriente consumida fluctúa entre 30-40 mA. La corriente media son 25 mA. Después de conectar, sin importar si se está realizando una transferencia de información o no, el consumo es de 8 mA.

7.3 IMU

La IMU que empleamos para medir la desviación del dispositivo tiene un consumo de 3,8 mA.

6.3 Electrical and Other Common Specifications

VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T_A = 25°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	Units
TEMPERATURE SENSOR					
Range			-40 to +85		°C
Sensitivity	Untrimmed		340		LSB/°C
Temperature Offset	35°C		-521		LSB
Linearity	Best fit straight line (-40°C to +85°C)		±1		°C
VDD POWER SUPPLY					
Operating Voltages		2.375		3.46	V
Normal Operating Current	Gyroscope + Accelerometer + DMP		3.9		mA
	Gyroscope + Accelerometer (DMP disabled)		3.8		mA
	Gyroscope + DMP (Accelerometer disabled)		3.7		mA
	Gyroscope only (DMP & Accelerometer disabled)		3.6		mA
	Accelerometer only (DMP & Gyroscope disabled)		500		µA

Figura 89 Consumo MPU6050

MPU6050 Datasheet

7.4 Fotorresistencia

El divisor de tensión empleado para obtener el valor de la fotorresistencia presenta una impedancia muy alta, con lo que el consumo de este conjunto es despreciable (de unos 0,8 µA).

7.5 Diodos LED

Para alimentar los LEDs se aprovecharán los 40 mA que es capaz de suministrar la salida digital de Arduino que los activa.

7.6 Conclusión

Elemento	Consumo máximo
Arduino Uno R3	50 mA
HC-05	40 mA
MPU 6050	3,8 mA
Fotorresistencia	Despreciable
Diodos LED	40 mA
Máximo consumo instantáneo	133,8 mA
Total disponible	450 mA

Tras el análisis comprobamos que disponemos de intensidad de corriente suficiente para alimentar el conjunto electrónico.

Como ya comentamos anteriormente, los motores deben ser alimentados mediante una batería externa a través de los circuitos integrados L293D.

Capítulo 8

Ensamblaje y pruebas

8.1 Peso

Hagamos un análisis del peso del drone:

- El drone original, de Syma, estaba diseñado para sustentar un peso total de 96 gramos.
- El peso total de nuestro conjunto es de 198 gramos.

Concluimos que el peso del dispositivo es demasiado elevado para que los motores suministren la potencia suficiente para que los rotores generen el empuje requerido para levantarlo en el aire.

Por ello, y para probar el control, se empleará un recipiente con agua sobre la que se posicionará el dispositivo anclado a una pieza de poliestireno extruido. De esta forma comprobaremos que los comandos introducidos a través de la App se transforman en un movimiento real del dispositivo.

8.2 Ensamblaje

Se construye una base de poliestireno extruido que proporcionará suspensión al conjunto. Además, se fabrica un soporte de cartón pluma sobre el que se instalará el conjunto Arduino-protoboard con todos los elementos conectados, a falta de los motores y los LEDs, que están situados en el chasis.

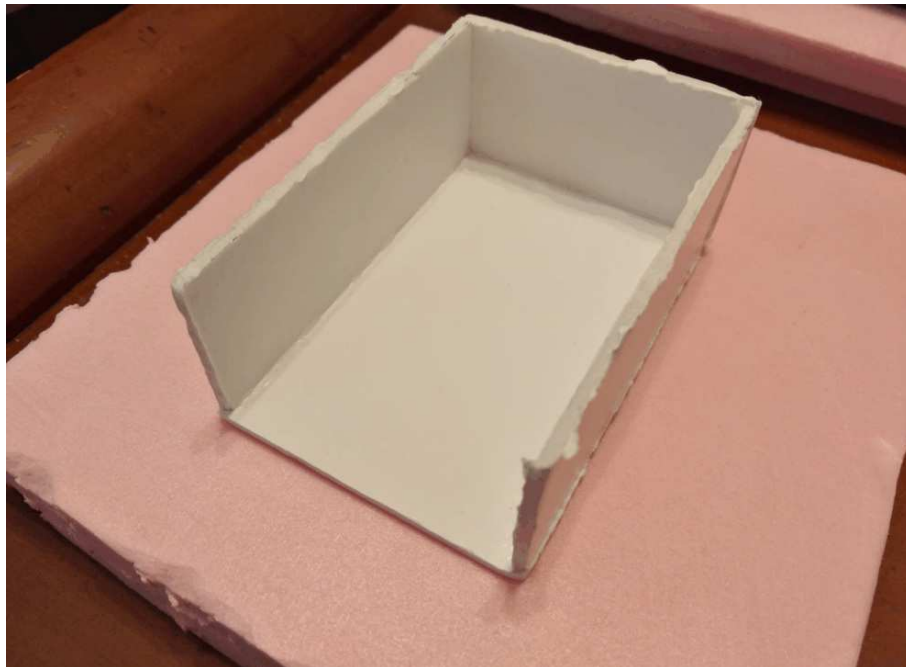


Figura 90 Base de poliestireno extruido con soporte de cartón pluma

Se añade una pantalla de cartón pluma para aislar el módulo de Bluetooth de la fotorresistencia, ya que el primero incorpora un LED que falsea el valor de intensidad lumínica medido por la LDR.

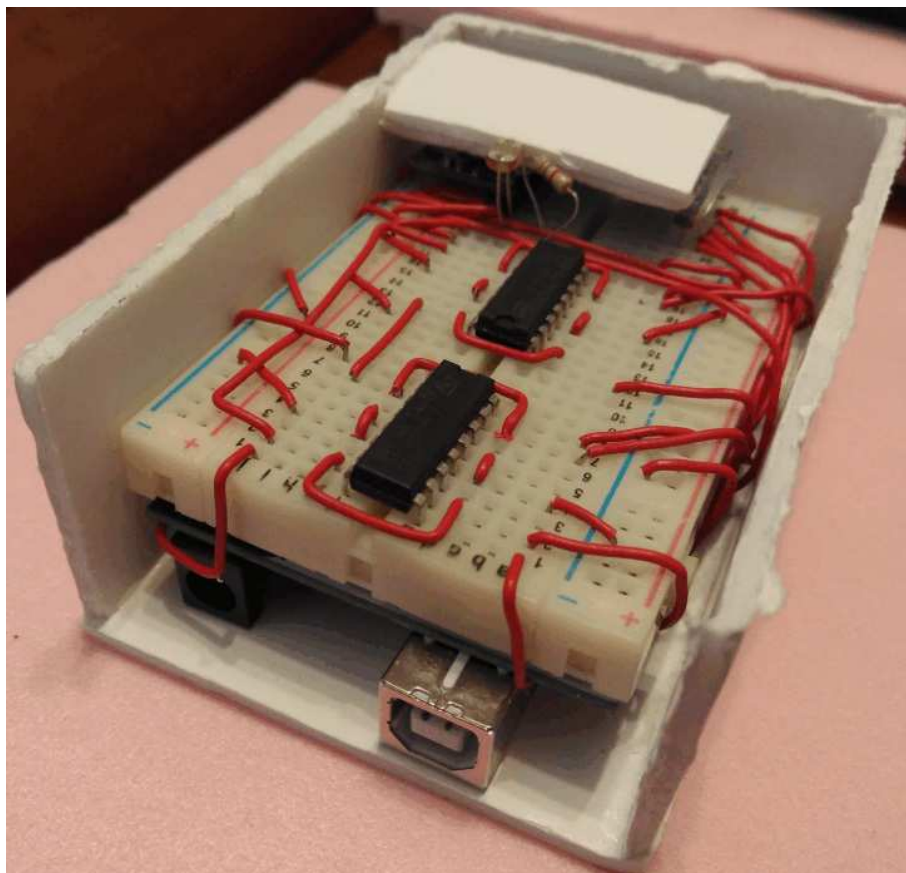


Figura 91 Soporte con Arduino y pantalla aislante

Se terminan de conectar los elementos restantes.

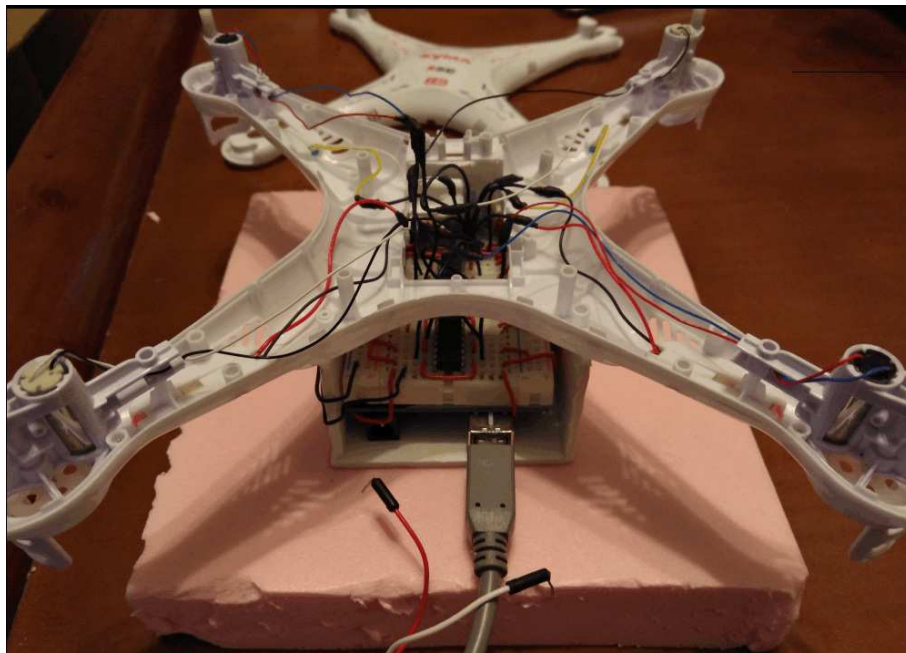


Figura 92 Conexionado de motores y LEDs

Se coloca la cubierta que termina de fijar los rotores y protege todos los elementos del conjunto.



Figura 93 Dispositivo con cubierta

Se posiciona el dispositivo sobre la superficie con agua para comprobar que las órdenes enviadas desde el Smartphone se traducen en los movimientos reales deseados.



Figura 94 Dispositivo sobre recipiente con agua

8.3 Esquema

El montaje final esquematizado es el siguiente:

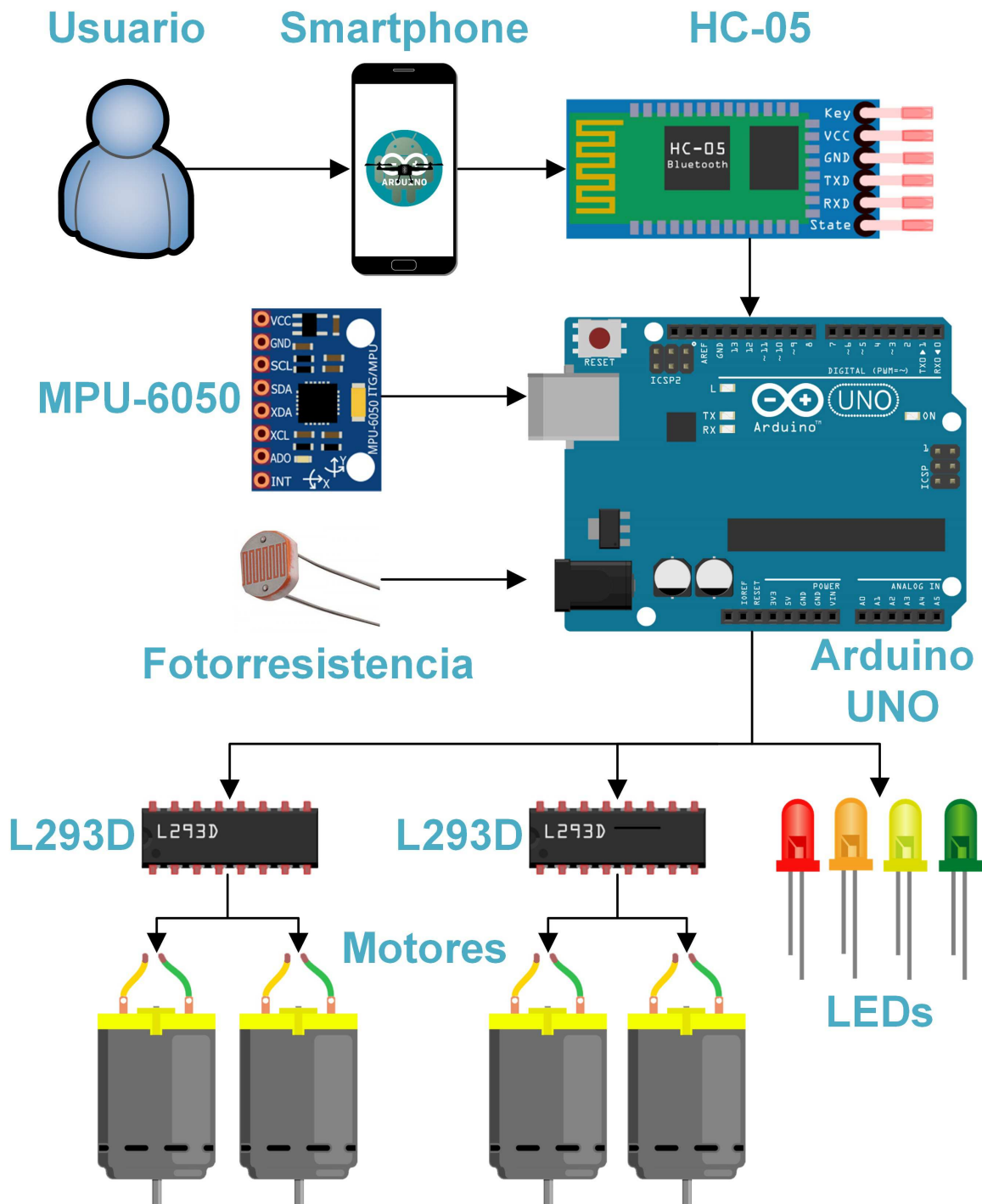


Figura 95 Representación esquemática de conexiones

El usuario da la orden deseada a través de la app en su Smartphone, que se envía al dispositivo a través de módulo de Bluetooth HC-05 que, conectado a Arduino, actúa sobre los motores mediante los L293D. Además, el comportamiento de los motores se modifica en función de las sucesivas lecturas del MPU6050 para mantener la estabilidad, y los LEDs se encienden en función del valor de la fotorresistencia y de los comandos recibidos.

8.4 Conexionado

El conexionado final de todos los elementos es el siguiente:

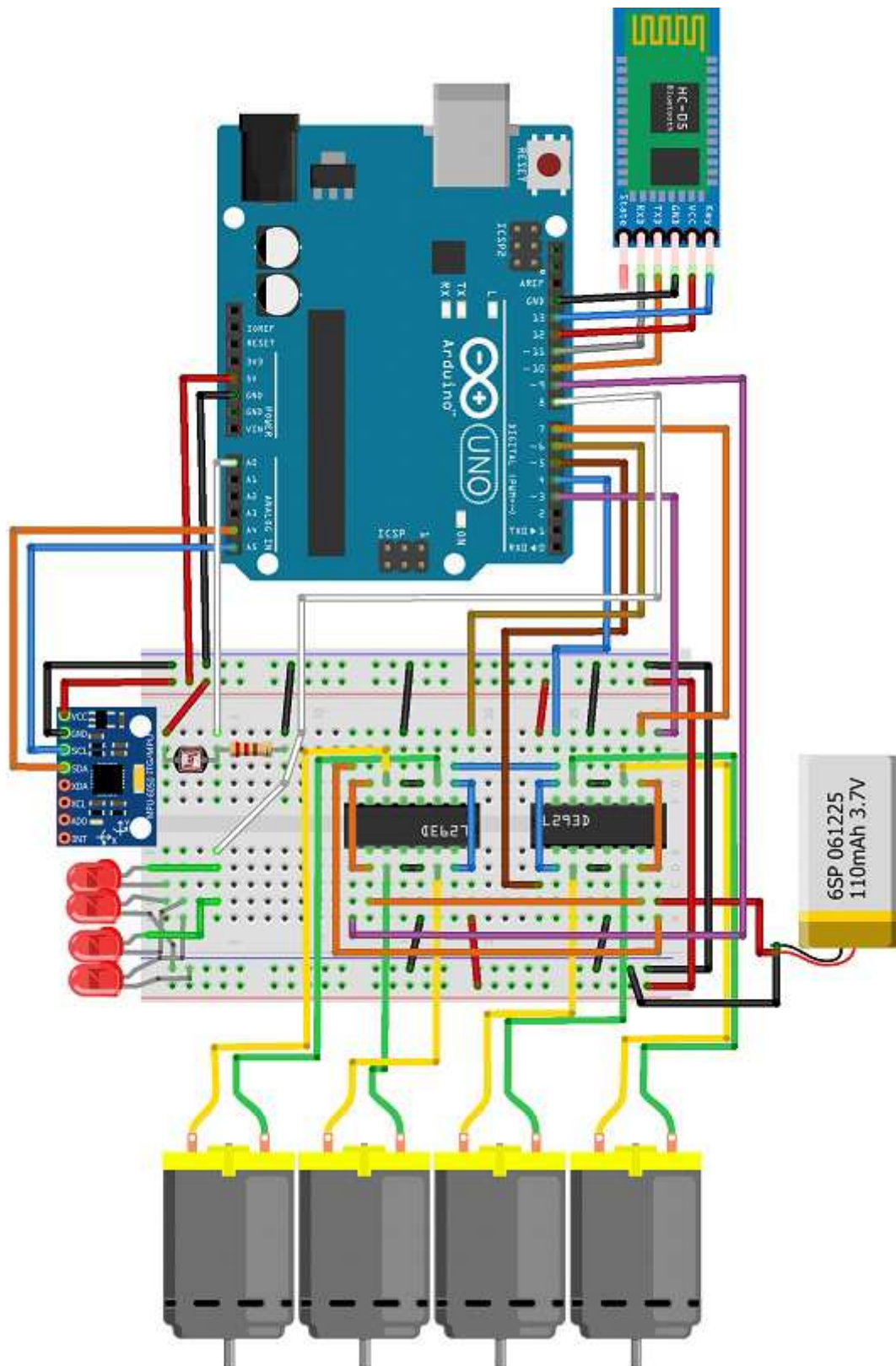


Figura 96 Conexionado

8.5 Pruebas

Para probar ajustar el control remoto y comprobar su correcto funcionamiento se llevan a cabo las pruebas siguientes:

- Se conecta la batería a los motores.
- Se conecta el cable USB para alimentar la electrónica del conjunto.
- Se inicia la aplicación en el dispositivo móvil, y se activa el Bluetooth. Se lleva a cabo una búsqueda de los dispositivos disponibles, y se selecciona el denominado *ArdDrone*
- Se incrementa la velocidad de los motores accionando el botón *Ascender* hasta que se alcanza el máximo de velocidad.
- Se envían comandos de movimiento utilizando las flechas en la parte inferior de la pantalla, observando que el dispositivo se desplaza en la dirección deseada.
- Se comprueba que los diodos LED experimentan el comportamiento esperado en función de la intensidad lumínica ambiental y de los comandos enviados desde el Smartphone.
- Se observa cierta dificultad al intentar desplazar el dispositivo en la dirección contraria al lateral donde está conectado el cable que alimenta la placa de control, que ofrece una cierta resistencia al movimiento en este sentido. Por ello, se incrementa la disminución de velocidad de los motores de la parte delantera del dron cuando se ordena un movimiento en esta trayectoria para que el empuje generado por la parte trasera sea suficiente para provocar un desplazamiento del conjunto.
- Se comprueba que esta modificación mejora el comportamiento del dispositivo.
- Se valida la implementación del control remoto.

Capítulo 9

Conclusiones

El proyecto se ha culminado con el desarrollo de un prototipo de VANT controlado a través de una aplicación de Smartphone para el sistema operativo Android.

Además, se ha llevado a cabo una recopilación de información acerca de la historia y las diversas aplicaciones, así como de la normativa por la que se rige la operación de estos dispositivos en la actualidad.

La actuación sobre los diferentes componentes a través de la programación del código en Arduino permite la coordinación de las acciones deseadas para mantener la estabilidad del dispositivo.

Se ha analizado el consumo de los elementos por separado para evitar sobretensiones y determinar la capacidad de alimentación de las fuentes.

Durante las pruebas realizadas se ha comprobado que la potencia de los motores no es suficiente para proporcionar suspensión al dispositivo debido al elevado peso que añaden varios de los componentes utilizados.

Se ha implementado un control en lazo cerrado en función de la lectura en tiempo real de la IMU MPU6050 aplicando un filtro para que los valores sean estables, y se han programado las acciones requeridas en función de su lectura.

Además, se ha desarrollado la aplicación de control para Android mediante AppInventor, y se ha configurado el módulo de Bluetooth HC-05 para establecer la comunicación con la placa de control.

Por último, se ha incluido la activación y desactivación de los LEDs del chasis del VANT en función del nivel de luminosidad ambiental medido a través de una LDR GL5528.

Se ha realizado el montaje y conexionado de todos los elementos a la placa de control.

Desde otra perspectiva, se extrae como conclusión que la creciente accesibilidad a Internet favorece el desarrollo de proyectos de electrónica caseros cada vez más complejos, gracias a la gran cantidad de información disponible y la comunidad de usuarios experimentados en todas las partes del mundo con los que se forman equipos de trabajo a grandes distancias.

Trabajos futuros

El desarrollo del prototipo de UAV ha sido satisfactorio, pero se encuentran varias líneas de investigación futuras:

- Utilización de otro modelo de Arduino para reducir el peso. Los modelos NANO y MINI son suficientes para realizar el control del dispositivo, y presentan un tamaño y peso mucho más reducidos.
- Construcción de una placa de circuito impreso (PCB) para reducir el peso que en el prototipo supone la protoboard sobre la que se conectan los distintos componentes.
- Realización de pruebas y mejoras en la programación orientadas a conseguir una mejor estabilidad.
- Adición de infinidad de sensores de diversos tipos en función de la aplicación a la que se quiera destinar el dispositivo.

Capítulo 10

Bibliografía

1. Comunidad de Madrid, Consejería de Economía y Hacienda
“Los drones y sus aplicaciones a la ingeniería civil”
[Enlace](#)
2. Martin J. Dougherty
El gran mundo de los drones. Una historia ilustrada
3. Web Mundo Drone
“Historia de los Drones”
[Enlace](#)
4. ADEIT. Fundación Universidad-Empresa de la Universitat de València
“Origen y desarrollo de los drones”
[Enlace](#)
5. AESA (Agencia Estatal de Seguridad Aérea)
“El uso de los drones en España”
[Enlace](#)
6. Ministerio de la presidencia y para las administraciones territoriales
Agencia Estatal. Boletín Oficial de Estado
“Ley 48/1960, de 21 de julio, de Navegación Aérea”
[Enlace](#)
7. Ministerio de la presidencia y para las administraciones territoriales
Agencia Estatal. Boletín Oficial de Estado
“Ley 18/2014, de 15 de octubre, de aprobación de medidas urgentes para el crecimiento, la competitividad y la eficiencia”
[Enlace](#)
8. Ministerio de la presidencia y para las administraciones territoriales
Agencia Estatal. Boletín Oficial de Estado
“Real Decreto 552/2014, de 27 de junio, por el que se desarrolla el Reglamento del aire y disposiciones operativas comunes para los servicios y procedimientos de navegación aérea y se modifica el Real Decreto 57/2002, de 18 de enero, por el que se aprueba el Reglamento de Circulación Aérea.”
[Enlace](#)
9. Zima Robotics, empresa operadora de drones autorizada por la AESA
“Tipos de drones aéreos”

- [Enlace](#)
10. Web oficial Arduino
[Enlace](#)
 11. Andrés Iborra García
Apuntes de la asignatura “*Microrrobótica*” del Grado en Ingeniería Electrónica Industrial y Automática de la Universidad Politécnica de Cartagena
 12. Juan Antonio López Riquelme y Juan Suardíaz Muro
Apuntes de la asignatura “*Microrrobótica*” del Grado en Ingeniería Electrónica Industrial y Automática de la Universidad Politécnica de Cartagena
 13. ST
“Push-Pull four channel driver with diodes” (L293D Datasheet)
[Enlace](#)
 14. Antonio Ricoy Riego
Web “*App Inventor en Español. Primeros pasos*”
[Enlace](#)
 15. Web MIT App Inventor
[Enlace](#)
 16. Web Prometec
“*El módulo Bluetooth HC-05*”
[Enlace](#)
 17. InvenSense
“*MPU-6000 and MPU-6050 Product Specification. Revision 3.4*” (MPU-6050 Datasheet)
[Enlace](#)
 18. Web Prometec
“*Usando el MPU6050*”
[Enlace](#)
 19. M. Margolis
“*Arduino Cookbook*”
[Enlace](#)
 20. Robert L. Boylestad y Louis Nashelsky
“*Electrónica: teoría de circuitos*”
[Enlace](#)
 21. A. Carretero, F. J. Ferrero, J. A. Sánchez-Infantes, P. Sánchez-Infantes, F. J. Valero
“*ELECTRÓNICA*”
[Enlace](#)

Anexo 1

Código en

Arduino

```

//*****LIBRERIAS*****
//Librería para conectar el módulo Bluetooth
#include <SoftwareSerial.h>

//Librería para gestionar el protocolo I²C
#include <Wire.h>

//*****DEFINICIONES*****

//Enable L293D
#define I1 4
#define I2 7

//Potencia motores L293D
#define E1 3
#define E2 5
#define E3 6
#define E4 9

//Pines LDR - LEDS
#define LDR 0
#define LEDS 8

//Ratios de conversion datos IMU
#define A_R 16384.0
#define G_R 131.0
//Conversión de radianes a grados (180/PI)
#define RAD_A_DEG = 57.295779
//Dirección I²C de la IMU
#define MPU 0x68

//*****VARIABLES*****
//Flag de parada
bool Paro;

//Valores medidos por la MPU-6050
int16_t AcX, AcY, AcZ, GyX, GyY, GyZ;

//Ángulos
float Acc[2];
float Gy[2];
float Angle[2];

//Control velocidad motores
int velocidad = 0;
int velmovimiento;
char lect = 0;

//Valor fotorresistencia
int valorLDR = 0;

```

```

//Variables para HC-05
int TXD = 10;
int RXD = 11;
int RST = 12;
int KEY = 13;

SoftwareSerial BTSerial(TXD, RXD);

//*****SETUP*****

void setup() {

  //Escribir dirección de la IMU
  Wire.begin();
  Wire.beginTransmission(MPU);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);

  //Encender el módulo de Bluetooth HC-05
  pinMode(RST, OUTPUT);
  pinMode(KEY, OUTPUT);

  //Estado inicial, Apagado
  digitalWrite(RST, LOW);
  //Modo de comunicación
  digitalWrite(KEY, LOW);

  //Encendemos HC-05
  digitalWrite(RST, HIGH);
  //Configuración puerto serie HC-05
  BTSerial.begin(9600);
  delay(500);

  //Definición de pines como I/O
  pinMode(I1, OUTPUT);
  pinMode(I2, OUTPUT);
  pinMode(E1, OUTPUT);
  pinMode(E2, OUTPUT);
  pinMode(E3, OUTPUT);
  pinMode(E4, OUTPUT);
  pinMode(LEDs, OUTPUT);

  //Habilitar L293D
  digitalWrite(I1, HIGH);
  digitalWrite(I2, LOW);

  //LEDs apagados
  digitalWrite(LEDs, LOW);

  //Configuración puerto serie Arduino
  Serial.begin(9600);

```



```

Serial.println("Listo para conectar por Bluetooth");

Paro = 1;
//Flag para evitar que el control en lazo cerrado actúe si el dron no se está moviendo
}

//*****PROGRAMA*****
void loop() {

//Notificar si el HC-05 se desconecta
if (RST == 0) Serial.println("DESCONECTADO");

//Encender LEDs en función de la luminosidad ambiental
valorLDR = analogRead(LDR);

//Encender LEDs en función de la luminosidad ambiental
if(valorLDR < 850) {
digitalWrite(LEDS, LOW);
}
if(valorLDR > 850) {
digitalWrite(LEDS, HIGH);
}

//*****Lectura puerto serie*****
if (Serial.available()) {
lect = Serial.read();
//Serial.println("Lectura puerto serie");
digitalWrite(LEDS, !digitalRead(LEDS));
Serial.println(lect);
}
//Si hay una lectura los LEDs parpadean para tener un
//indicativo externo de que el dato se ha recibido

//*****Lectura HC-05*****
if (BTSerial.available()) {
lect = BTSerial.read();
Serial.println("Lectura HC-05");
digitalWrite(LEDS, !digitalRead(LEDS));
Serial.println(lect);
}

//*****Actuación sobre motores según comando*****
//Ascender
if (lect == '+' && velocidad < 255) {
//Incrementamos la velocidad
velocidad += 15;
Paro = 0;
Serial.println("Ascenso");
Serial.println(velocidad);
delay(100);
}
//Descender

```

```

if (lect == '-' && velocidad >= 15) {
//Decrementamos la velocidad y la limitamos a 0
    velocidad -= 15;
    Paro = 0;
    Serial.println("Descenso");
    Serial.println(velocidad);
    delay(100);
}
//Avanzar
if (lect == 'a') {
    velmovimiento = velocidad - 150;
//Reducimos la velocidad de los motores del lado hacia el que se quiere desplazar,
    limitándola a 0 cuando la resta hace el valor negativo
    Paro = 0;
    if (velmovimiento>=0) {
        analogWrite(E3, velmovimiento);
        analogWrite(E4, velmovimiento);
    }
    else {
        analogWrite(E3,0);
        analogWrite(E4,0);
    }
    Serial.println("Avanzar");
    Serial.println(velmovimiento);
    Serial.println(velocidad);
    delay(1500);
//Delay para que la orden se traduzca en un movimiento real
}
//Retroceder
if (lect == 'r') {
    velmovimiento = velocidad-100;
    Paro = 0;
    if (velmovimiento>=0) {
        analogWrite(E1,velmovimiento);
        analogWrite(E2,velmovimiento);
    }
    else {
        analogWrite(E1,0);
        analogWrite(E2,0);
    }
    Serial.println("Retroceder");
    Serial.println(velmovimiento);
    Serial.println(velocidad);
    delay(1500);
}

//Derecha
if (lect == 'd') {
    velmovimiento = velocidad-100;
    Paro = 0;
    if (velmovimiento>=0) {
        analogWrite(E1,velmovimiento);
        analogWrite(E3,velmovimiento);
    }
}

```

```

    }
    else {
        analogWrite(E1,0);
        analogWrite(E3,0);
    }
    Serial.println("Derecha");
    Serial.println(velmovimiento);
    Serial.println(velocidad);
    delay(1500);
}
//Izquierda
if (lect == 'i') {
    velmovimiento = velocidad-100;
    Paro = 0;
    if (velmovimiento>=0) {
        analogWrite(E2,velmovimiento);
        analogWrite(E4,velmovimiento);
    }
    else {
        analogWrite(E2,0);
        analogWrite(E4,0);
    }
    Serial.println("Izquierda");
    Serial.println(velmovimiento);
    Serial.println(velocidad);
    delay(1500);
}
//Paro
if (lect == 'p') { //Paro
    velocidad = 0;
    Paro = 1;
    Serial.println("Parar");
}

//*****Lectura valores acelerómetro*****
Wire.beginTransmission(MPU);
Wire.write(0x3B);
//Pedir el registro 0x3B corresponde al AcX
Wire.endTransmission(false);

Wire.requestFrom(MPU,6,true);
//A partir de 0x3B, se piden 6 registros
AcX=Wire.read()<<8|Wire.read();
AcY=Wire.read()<<8|Wire.read();
AcZ=Wire.read()<<8|Wire.read();
//Cada valor ocupa 2 registros

//Cálculo de los ángulos X, Y
//A partir de los valores del acelerómetro, se calculan los ángulos Y, X respectivamente,
//con la formula de la tangente.
Acc[1] = atan(-1*(AcX/A_R)/sqrt(pow((AcY/A_R),2) +

```

```

pow((AcZ/A_R),2)) *RAD_TO_DEG;
  Acc[0] = atan((AcY/A_R)/sqrt(pow((AcX/A_R),2) +
pow((AcZ/A_R),2)) *RAD_TO_DEG;

//*****Lectura valores giroscopio*****
  Wire.beginTransaction(MPU);
  Wire.write(0x43); //Direccion acelerometro
  Wire.endTransmission(false);
  Wire.requestFrom(MPU,4,true); //Se piden 4 registros
  GyX=Wire.read()<<8|Wire.read();
  GyY=Wire.read()<<8|Wire.read();

//*****Cálculo ángulo giroscopio*****
  Gy[0] = GyX/G_R;
  Gy[1] = GyY/G_R;

//*****Filtro complementario*****
  Angle[0] = 0.99 * (Angle[0]+Gy[0]*0.01) + 0.01*Acc[0];
  //El cálculo se hace cada 10 ms
  Angle[1] = 0.99 * (Angle[1]+Gy[1]*0.01) + 0.01*Acc[1];

//*****Control de estabilidad*****

  //El control sólo actúa cuando el dispositivo está en movimiento
  if (Paro==0) { //Si no está parado

  //Disminuir velocidad de los motores de la zona contraria al desvío de forma proporcional
  al ángulo leído para que el control sea más rápido
  if ((Angle[0]>0) && (abs(Angle[0])>5)) {
    velmovimiento = velocidad-0.01*(abs(Angle[0]))*velocidad;
  //Limitar velocidad a 0
    if (velmovimiento<0) {
      velmovimiento = 0;
    }
    analogWrite(E1,velmovimiento);
    analogWrite(E2,velmovimiento);
    Serial.println("Desvio corregido Angle[0]>0");
    Serial.println(velmovimiento);
    Serial.println(velocidad);
    delay(100);
    //Delay para que el desvío se corrija
  }

  else if ((Angle[0]<0) && (abs(Angle[0])>5)) {
    velmovimiento = velocidad-0.01*(abs(Angle[0]))*velocidad;
    if (velmovimiento<0) {
      velmovimiento = 0;
    }
    analogWrite(E3,velmovimiento);
    analogWrite(E4,velmovimiento);
    Serial.println("Desvio corregido Angle[0]<0");
  }

```

```

Serial.println(velmovimiento);
Serial.println(velocidad);
delay(100);
}

if ((Angle[1]>0) && (abs(Angle[1])>5)) {
velmovimiento = velocidad-0.01*(abs(Angle[1]))*velocidad;
  if (velmovimiento<0) {
    velmovimiento = 0;
  }
analogWrite(E1,velmovimiento);
analogWrite(E3,velmovimiento);
Serial.println("Desvio corregido Angle[1]>0");
Serial.println(velmovimiento);
Serial.println(velocidad);
delay(100);
}
else if ((Angle[1]<0) && (abs(Angle[1])>5)) {
velmovimiento = velocidad-0.01*(abs(Angle[1]))*velocidad;
if (velmovimiento<0) {
  velmovimiento = 0;
}
analogWrite(E2,velmovimiento);
analogWrite(E4,velmovimiento);
Serial.println("Desvio corregido Angle[1]<0");
Serial.println(velmovimiento);
Serial.println(velocidad);
delay(100);
}

delay(10); //El cálculo se hace cada 10 ms

} //Cierre condición Paro=0

//Volver a la velocidad anterior al comando
analogWrite(E1, velocidad);
analogWrite(E2, velocidad);
analogWrite(E3, velocidad);
analogWrite(E4, velocidad);
if (lect!=0) digitalWrite(LEDs, !digitalRead(LEDs));
//Invertir estado de los LEDs tras lectura

//Resetear lectura
lect = 0;

} //Cierre loop

```

Anexo 2

Datasheet

MPU 60X0



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



CONTENTS

1	REVISION HISTORY	5
2	PURPOSE AND SCOPE	6
3	PRODUCT OVERVIEW	7
3.1	MPU-60X0 OVERVIEW	7
4	APPLICATIONS.....	9
5	FEATURES	10
5.1	GYROSCOPE FEATURES.....	10
5.2	ACCELEROMETER FEATURES	10
5.3	ADDITIONAL FEATURES	10
5.4	MOTIONPROCESSING.....	11
5.5	CLOCKING.....	11
6	ELECTRICAL CHARACTERISTICS	12
6.1	GYROSCOPE SPECIFICATIONS	12
6.2	ACCELEROMETER SPECIFICATIONS.....	13
6.3	ELECTRICAL AND OTHER COMMON SPECIFICATIONS.....	14
6.4	ELECTRICAL SPECIFICATIONS, CONTINUED	15
6.5	ELECTRICAL SPECIFICATIONS, CONTINUED	16
6.6	ELECTRICAL SPECIFICATIONS, CONTINUED	17
6.7	I ² C TIMING CHARACTERIZATION.....	18
6.8	SPI TIMING CHARACTERIZATION (MPU-6000 ONLY)	19
6.9	ABSOLUTE MAXIMUM RATINGS	20
7	APPLICATIONS INFORMATION	21
7.1	PIN OUT AND SIGNAL DESCRIPTION.....	21
7.2	TYPICAL OPERATING CIRCUIT.....	22
7.3	BILL OF MATERIALS FOR EXTERNAL COMPONENTS	22
7.4	RECOMMENDED POWER-ON PROCEDURE	23
7.5	BLOCK DIAGRAM	24
7.6	OVERVIEW	24
7.7	THREE-AXIS MEMS GYROSCOPE WITH 16-BIT ADCS AND SIGNAL CONDITIONING.....	25
7.8	THREE-AXIS MEMS ACCELEROMETER WITH 16-BIT ADCS AND SIGNAL CONDITIONING	25
7.9	DIGITAL MOTION PROCESSOR	25
7.10	PRIMARY I ² C AND SPI SERIAL COMMUNICATIONS INTERFACES	25
7.11	AUXILIARY I ² C SERIAL INTERFACE	26



- 7.12 SELF-TEST27
- 7.13 MPU-60X0 SOLUTION FOR 9-AXIS SENSOR FUSION USING I²C INTERFACE28
- 7.14 MPU-6000 USING SPI INTERFACE29
- 7.15 INTERNAL CLOCK GENERATION30
- 7.16 SENSOR DATA REGISTERS30
- 7.17 FIFO30
- 7.18 INTERRUPTS30
- 7.19 DIGITAL-OUTPUT TEMPERATURE SENSOR31
- 7.20 BIAS AND LDO31
- 7.21 CHARGE PUMP31
- 8 PROGRAMMABLE INTERRUPTS.....32**
- 9 DIGITAL INTERFACE33**
 - 9.1 I²C AND SPI (MPU-6000 ONLY) SERIAL INTERFACES33
 - 9.2 I²C INTERFACE33
 - 9.3 I²C COMMUNICATIONS PROTOCOL.....33
 - 9.4 I²C TERMS36
 - 9.5 SPI INTERFACE (MPU-6000 ONLY)37
- 10 SERIAL INTERFACE CONSIDERATIONS (MPU-6050)38**
 - 10.1 MPU-6050 SUPPORTED INTERFACES.....38
 - 10.2 LOGIC LEVELS38
 - 10.3 LOGIC LEVELS DIAGRAM FOR AUX_VDDIO = 0.....39
- 11 ASSEMBLY40**
 - 11.1 ORIENTATION OF AXES40
 - 11.2 PACKAGE DIMENSIONS41
 - 11.3 PCB DESIGN GUIDELINES42
 - 11.4 ASSEMBLY PRECAUTIONS43
 - 11.5 STORAGE SPECIFICATIONS.....46
 - 11.6 PACKAGE MARKING SPECIFICATION.....46
 - 11.7 TAPE & REEL SPECIFICATION.....47
 - 11.8 LABEL48
 - 11.9 PACKAGING.....49
 - 11.10 REPRESENTATIVE SHIPPING CARTON LABEL.....50
- 12 RELIABILITY51**
 - 12.1 QUALIFICATION TEST POLICY51



12.2 QUALIFICATION TEST PLAN51

13 ENVIRONMENTAL COMPLIANCE.....52



1 Revision History

Revision Date	Revision	Description
11/24/2010	1.0	Initial Release
05/19/2011	2.0	For Rev C parts. Clarified wording in sections (3.2, 5.1, 5.2, 6.1-6.4, 6.6, 6.9, 7, 7.1-7.6, 7.11, 7.12, 7.14, 8, 8.2-8.4, 10.3, 10.4, 11, 12.2)
07/28/2011	2.1	Edited supply current numbers for different modes (section 6.4)
08/05/2011	2.2	Unit of measure for accelerometer sensitivity changed from LSB/mg to LSB/g
10/12/2011	2.3	Updated accelerometer self test specifications in Table 6.2. Updated package dimensions (section 11.2). Updated PCB design guidelines (section 11.3)
10/18/2011	3.0	For Rev D parts. Updated accelerometer specifications in Table 6.2. Updated accelerometer specification note (sections 8.2, 8.3, & 8.4). Updated qualification test plan (section 12.2).
10/24/2011	3.1	Edits for clarity Changed operating voltage range to 2.375V-3.46V Added accelerometer Intelligence Function increment value of 1mg/LSB (Section 6.2) Updated absolute maximum rating for acceleration (any axis, unpowered) from 0.3ms to 0.2ms (Section 6.9) Modified absolute maximum rating for Latch-up to Level A and ± 100 mA (Section 6.9, 12.2)
11/16/2011	3.2	Updated self-test response specifications for Revision D parts dated with date code 1147 (YYWW) or later. Edits for clarity Added Gyro self-test (sections 5.1, 6.1, 7.6, 7.12) Added Min/Max limits to Accel self-test response (section 6.2) Updated Accelerometer low power mode operating currents (Section 6.3) Added gyro self test to block diagram (section 7.5) Updated packaging labels and descriptions (sections 11.8 & 11.9)
5/16/2012	3.3	Updated Gyro and Accelerometer self test information (sections 6.1, 6.2, 7.12) Updated latch-up information (Section 6.9) Updated programmable interrupts information (Section 8) Changed shipment information from maximum of 3 reels (15K units) per shipper box to 5 reels (25K units) per shipper box (Section 11.7) Updated packing shipping and label information (Sections 11.8, 11.9) Updated reliability references (Section 12.2)
8/19/2013	3.4	Updates section 4



2 Purpose and Scope

This product specification provides advanced information regarding the electrical specification and design related information for the MPU-6000™ and MPU-6050™ MotionTracking™ devices, collectively called the MPU-60X0™ or MPU™.

Electrical characteristics are based upon design analysis and simulation results only. Specifications are subject to change without notice. Final specifications will be updated based upon characterization of production silicon. For references to register map and descriptions of individual registers, please refer to the MPU-6000/MPU-6050 Register Map and Register Descriptions document.

The self-test response specifications provided in this document pertain to Revision D parts with date codes of 1147 (YYWW) or later. Please see Section 11.6 for package marking description details.



3 Product Overview

3.1 MPU-60X0 Overview

MotionInterface™ is becoming a “must-have” function being adopted by smartphone and tablet manufacturers due to the enormous value it adds to the end user experience. In smartphones, it finds use in applications such as gesture commands for applications and phone control, enhanced gaming, augmented reality, panoramic photo capture and viewing, and pedestrian and vehicle navigation. With its ability to precisely and accurately track user motions, MotionTracking technology can convert handsets and tablets into powerful 3D intelligent devices that can be used in applications ranging from health and fitness monitoring to location-based services. Key requirements for MotionInterface enabled devices are small package size, low power consumption, high accuracy and repeatability, high shock tolerance, and application specific performance programmability – all at a low consumer price point.

The MPU-60X0 is the world’s first integrated 6-axis MotionTracking device that combines a 3-axis gyroscope, 3-axis accelerometer, and a Digital Motion Processor™ (DMP) all in a small 4x4x0.9mm package. With its dedicated I²C sensor bus, it directly accepts inputs from an external 3-axis compass to provide a complete 9-axis MotionFusion™ output. The MPU-60X0 MotionTracking device, with its 6-axis integration, on-board MotionFusion™, and run-time calibration firmware, enables manufacturers to eliminate the costly and complex selection, qualification, and system level integration of discrete devices, guaranteeing optimal motion performance for consumers. The MPU-60X0 is also designed to interface with multiple non-inertial digital sensors, such as pressure sensors, on its auxiliary I²C port. The MPU-60X0 is footprint compatible with the MPU-30X0 family.

The MPU-60X0 features three 16-bit analog-to-digital converters (ADCs) for digitizing the gyroscope outputs and three 16-bit ADCs for digitizing the accelerometer outputs. For precision tracking of both fast and slow motions, the parts feature a user-programmable gyroscope full-scale range of ± 250 , ± 500 , ± 1000 , and $\pm 2000^\circ/\text{sec}$ (dps) and a user-programmable accelerometer full-scale range of $\pm 2g$, $\pm 4g$, $\pm 8g$, and $\pm 16g$.

An on-chip 1024 Byte FIFO buffer helps lower system power consumption by allowing the system processor to read the sensor data in bursts and then enter a low-power mode as the MPU collects more data. With all the necessary on-chip processing and sensor components required to support many motion-based use cases, the MPU-60X0 uniquely enables low-power MotionInterface applications in portable applications with reduced processing requirements for the system processor. By providing an integrated MotionFusion output, the DMP in the MPU-60X0 offloads the intensive MotionProcessing computation requirements from the system processor, minimizing the need for frequent polling of the motion sensor output.

Communication with all registers of the device is performed using either I²C at 400kHz or SPI at 1MHz (MPU-6000 only). For applications requiring faster communications, the sensor and interrupt registers may be read using SPI at 20MHz (MPU-6000 only). Additional features include an embedded temperature sensor and an on-chip oscillator with $\pm 1\%$ variation over the operating temperature range.

By leveraging its patented and volume-proven Nasiri-Fabrication platform, which integrates MEMS wafers with companion CMOS electronics through wafer-level bonding, InvenSense has driven the MPU-60X0 package size down to a revolutionary footprint of 4x4x0.9mm (QFN), while providing the highest performance, lowest noise, and the lowest cost semiconductor packaging required for handheld consumer electronic devices. The part features a robust 10,000g shock tolerance, and has programmable low-pass filters for the gyroscopes, accelerometers, and the on-chip temperature sensor.

For power supply flexibility, the MPU-60X0 operates from VDD power supply voltage range of 2.375V-3.46V. Additionally, the MPU-6050 provides a VLOGIC reference pin (in addition to its analog supply pin: VDD), which sets the logic levels of its I²C interface. The VLOGIC voltage may be $1.8V \pm 5\%$ or VDD.

The MPU-6000 and MPU-6050 are identical, except that the MPU-6050 supports the I²C serial interface only, and has a separate VLOGIC reference pin. The MPU-6000 supports both I²C and SPI interfaces and has a single supply pin, VDD, which is both the device’s logic reference supply and the analog supply for the part. The table below outlines these differences:



Primary Differences between MPU-6000 and MPU-6050

Part / Item	MPU-6000	MPU-6050
VDD	2.375V-3.46V	2.375V-3.46V
VLOGIC	n/a	1.71V to VDD
Serial Interfaces Supported	I ² C, SPI	I ² C
Pin 8	/CS	VLOGIC
Pin 9	AD0/SDO	AD0
Pin 23	SCL/SCLK	SCL
Pin 24	SDA/SDI	SDA



4 Applications

- *BlurFree*[™] technology (for Video/Still Image Stabilization)
- *AirSign*[™] technology (for Security/Authentication)
- *TouchAnywhere*[™] technology (for “no touch” UI Application Control/Navigation)
- *MotionCommand*[™] technology (for Gesture Short-cuts)
- Motion-enabled game and application framework
- InstantGesture[™] iG[™] gesture recognition
- Location based services, points of interest, and dead reckoning
- Handset and portable gaming
- Motion-based game controllers
- 3D remote controls for Internet connected DTVs and set top boxes, 3D mice
- Wearable sensors for health, fitness and sports
- Toys

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 Electrical Characteristics

6.1 Gyroscope Specifications

VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T_A = 25°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
GYROSCOPE SENSITIVITY						
Full-Scale Range	FS_SEL=0		±250		°/s	
	FS_SEL=1		±500		°/s	
	FS_SEL=2		±1000		°/s	
	FS_SEL=3		±2000		°/s	
Gyroscope ADC Word Length			16		bits	
Sensitivity Scale Factor	FS_SEL=0		131		LSB/(°/s)	
	FS_SEL=1		65.5		LSB/(°/s)	
	FS_SEL=2		32.8		LSB/(°/s)	
	FS_SEL=3		16.4		LSB/(°/s)	
Sensitivity Scale Factor Tolerance	25°C	-3		+3	%	
Sensitivity Scale Factor Variation Over Temperature			±2		%	
Nonlinearity	Best fit straight line; 25°C		0.2		%	
Cross-Axis Sensitivity			±2		%	
GYROSCOPE ZERO-RATE OUTPUT (ZRO)						
Initial ZRO Tolerance	25°C		±20		°/s	
ZRO Variation Over Temperature	-40°C to +85°C		±20		°/s	
Power-Supply Sensitivity (1-10Hz)	Sine wave, 100mVpp; VDD=2.5V		0.2		°/s	
Power-Supply Sensitivity (10 - 250Hz)	Sine wave, 100mVpp; VDD=2.5V		0.2		°/s	
Power-Supply Sensitivity (250Hz - 100kHz)	Sine wave, 100mVpp; VDD=2.5V		4		°/s	
Linear Acceleration Sensitivity	Static		0.1		°/s/g	
SELF-TEST RESPONSE						
Relative	Change from factory trim	-14		14	%	1
GYROSCOPE NOISE PERFORMANCE	FS_SEL=0					
Total RMS Noise	DLPCFCFG=2 (100Hz)		0.05		°/s-rms	
Low-frequency RMS noise	Bandwidth 1Hz to10Hz		0.033		°/s-rms	
Rate Noise Spectral Density	At 10Hz		0.005		°/s/√Hz	
GYROSCOPE MECHANICAL FREQUENCIES						
X-Axis		30	33	36	kHz	
Y-Axis		27	30	33	kHz	
Z-Axis		24	27	30	kHz	
LOW PASS FILTER RESPONSE						
	Programmable Range	5		256	Hz	
OUTPUT DATA RATE						
	Programmable	4		8,000	Hz	
GYROSCOPE START-UP TIME						
ZRO Settling (from power-on)	DLPCFCFG=0 to ±1°/s of Final		30		ms	

- Please refer to the following document for further information on Self-Test: *MPU-6000/MPU-6050 Register Map and Descriptions*



6.2 Accelerometer Specifications

VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T_A = 25°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
ACCELEROMETER SENSITIVITY						
Full-Scale Range	AFS_SEL=0		±2		g	
	AFS_SEL=1		±4		g	
	AFS_SEL=2		±8		g	
	AFS_SEL=3		±16		g	
ADC Word Length	Output in two's complement format		16		bits	
Sensitivity Scale Factor	AFS_SEL=0		16,384		LSB/g	
	AFS_SEL=1		8,192		LSB/g	
	AFS_SEL=2		4,096		LSB/g	
	AFS_SEL=3		2,048		LSB/g	
Initial Calibration Tolerance			±3		%	
Sensitivity Change vs. Temperature	AFS_SEL=0, -40°C to +85°C		±0.02		%/°C	
Nonlinearity	Best Fit Straight Line		0.5		%	
Cross-Axis Sensitivity			±2		%	
ZERO-G OUTPUT						
Initial Calibration Tolerance	X and Y axes		±50		mg	1
	Z axis		±80		mg	
Zero-G Level Change vs. Temperature	X and Y axes, 0°C to +70°C		±35			
	Z axis, 0°C to +70°C		±60		mg	
SELF TEST RESPONSE						
Relative	Change from factory trim	-14		14	%	2
NOISE PERFORMANCE						
Power Spectral Density	@10Hz, AFS_SEL=0 & ODR=1kHz		400		μg/√Hz	
LOW PASS FILTER RESPONSE						
	Programmable Range	5		260	Hz	
OUTPUT DATA RATE						
	Programmable Range	4		1,000	Hz	
INTELLIGENCE FUNCTION INCREMENT			32		mg/LSB	

1. Typical zero-g initial calibration tolerance value after MSL3 preconditioning
2. Please refer to the following document for further information on Self-Test: *MPU-6000/MPU-6050 Register Map and Descriptions*



6.3 Electrical and Other Common Specifications

VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T_A = 25°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	Units	Notes
TEMPERATURE SENSOR						
Range			-40 to +85		°C	
Sensitivity	Untrimmed		340		LSB/°C	
Temperature Offset	35°C		-521		LSB	
Linearity	Best fit straight line (-40°C to +85°C)		±1		°C	
VDD POWER SUPPLY						
Operating Voltages		2.375		3.46	V	
Normal Operating Current	Gyroscope + Accelerometer + DMP		3.9		mA	
	Gyroscope + Accelerometer (DMP disabled)		3.8		mA	
	Gyroscope + DMP (Accelerometer disabled)		3.7		mA	
	Gyroscope only (DMP & Accelerometer disabled)		3.6		mA	
	Accelerometer only (DMP & Gyroscope disabled)		500		µA	
Accelerometer Low Power Mode Current	1.25 Hz update rate		10		µA	
	5 Hz update rate		20		µA	
	20 Hz update rate		70		µA	
	40 Hz update rate		140		µA	
Full-Chip Idle Mode Supply Current			5		µA	
Power Supply Ramp Rate	Monotonic ramp. Ramp rate is 10% to 90% of the final value			100	ms	
VLOGIC REFERENCE VOLTAGE						
Voltage Range	MPU-6050 only	1.71		VDD	V	
Power Supply Ramp Rate	VLOGIC must be ≤VDD at all times			3	ms	
Normal Operating Current	Monotonic ramp. Ramp rate is 10% to 90% of the final value		100		µA	
TEMPERATURE RANGE						
Specified Temperature Range	Performance parameters are not applicable beyond Specified Temperature Range	-40		+85	°C	



6.4 Electrical Specifications, Continued

VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, TA = 25°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	Units	Notes
SERIAL INTERFACE						
SPI Operating Frequency, All Registers Read/Write	MPU-6000 only, Low Speed Characterization		100 ±10%		kHz	
	MPU-6000 only, High Speed Characterization		1 ±10%		MHz	
SPI Operating Frequency, Sensor and Interrupt Registers Read Only	MPU-6000 only		20 ±10%		MHz	
	All registers, Fast-mode			400	kHz	
I ² C Operating Frequency	All registers, Standard-mode			100	kHz	
I²C ADDRESS						
	AD0 = 0		1101000			
	AD0 = 1		1101001			
DIGITAL INPUTS (SDI/SDA, AD0, SCLK/SCL, FSYNC, /CS, CLKIN)						
V _{IH} , High Level Input Voltage	MPU-6000	0.7*VDD			V	
	MPU-6050	0.7*VLOGIC			V	
V _{IL} , Low Level Input Voltage	MPU-6000			0.3*VDD	V	
	MPU-6050			0.3*VLOGIC	V	
C _i , Input Capacitance			< 5		pF	
DIGITAL OUTPUT (SDO, INT)						
V _{OH} , High Level Output Voltage	R _{LOAD} =1MΩ; MPU-6000	0.9*VDD			V	
	R _{LOAD} =1MΩ; MPU-6050	0.9*VLOGIC			V	
V _{OL1} , LOW-Level Output Voltage	R _{LOAD} =1MΩ; MPU-6000			0.1*VDD	V	
	R _{LOAD} =1MΩ; MPU-6050			0.1*VLOGIC	V	
V _{OL,INT1} , INT Low-Level Output Voltage	OPEN=1, 0.3mA sink Current			0.1	V	
Output Leakage Current	OPEN=1		100		nA	
t _{INT} , INT Pulse Width	LATCH_INT_EN=0		50		μs	



6.5 Electrical Specifications, Continued

Typical Operating Circuit of Section 7.2, VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, TA = 25°C

Parameters	Conditions	Typical	Units	Notes
Primary I²C I/O (SCL, SDA)				
V _{IL} , LOW-Level Input Voltage	MPU-6000	-0.5 to 0.3*VDD	V	
V _{IH} , HIGH-Level Input Voltage	MPU-6000	0.7*VDD to VDD + 0.5V	V	
V _{hys} , Hysteresis	MPU-6000	0.1*VDD	V	
V _{IL} , LOW Level Input Voltage	MPU-6050	-0.5V to 0.3*VLOGIC	V	
V _{IH} , HIGH-Level Input Voltage	MPU-6050	0.7*VLOGIC to VLOGIC + 0.5V	V	
V _{hys} , Hysteresis	MPU-6050	0.1*VLOGIC	V	
V _{OL1} , LOW-Level Output Voltage	3mA sink current	0 to 0.4	V	
I _{OL} , LOW-Level Output Current	V _{OL} = 0.4V	3	mA	
	V _{OL} = 0.6V	5	mA	
Output Leakage Current		100	nA	
t _{of} , Output Fall Time from V _{IHmax} to V _{ILmax}	C _b bus capacitance in pF	20+0.1C _b to 250	ns	
C _I , Capacitance for Each I/O pin		< 10	pF	
Auxiliary I²C I/O (AUX_CL, AUX_DA)				
MPU-6050: AUX_VDDIO=0				
V _{IL} , LOW-Level Input Voltage		-0.5V to 0.3*VLOGIC	V	
V _{IH} , HIGH-Level Input Voltage		0.7*VLOGIC to VLOGIC + 0.5V	V	
V _{hys} , Hysteresis		0.1*VLOGIC	V	
V _{OL1} , LOW-Level Output Voltage	VLOGIC > 2V; 1mA sink current	0 to 0.4	V	
V _{OL3} , LOW-Level Output Voltage	VLOGIC < 2V; 1mA sink current	0 to 0.2*VLOGIC	V	
I _{OL} , LOW-Level Output Current	V _{OL} = 0.4V	1	mA	
	V _{OL} = 0.6V	1	mA	
Output Leakage Current		100	nA	
t _{of} , Output Fall Time from V _{IHmax} to V _{ILmax}	C _b bus capacitance in pF	20+0.1C _b to 250	ns	
C _I , Capacitance for Each I/O pin		< 10	pF	



6.6 Electrical Specifications, Continued

Typical Operating Circuit of Section 7.2, VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T_A = 25°C

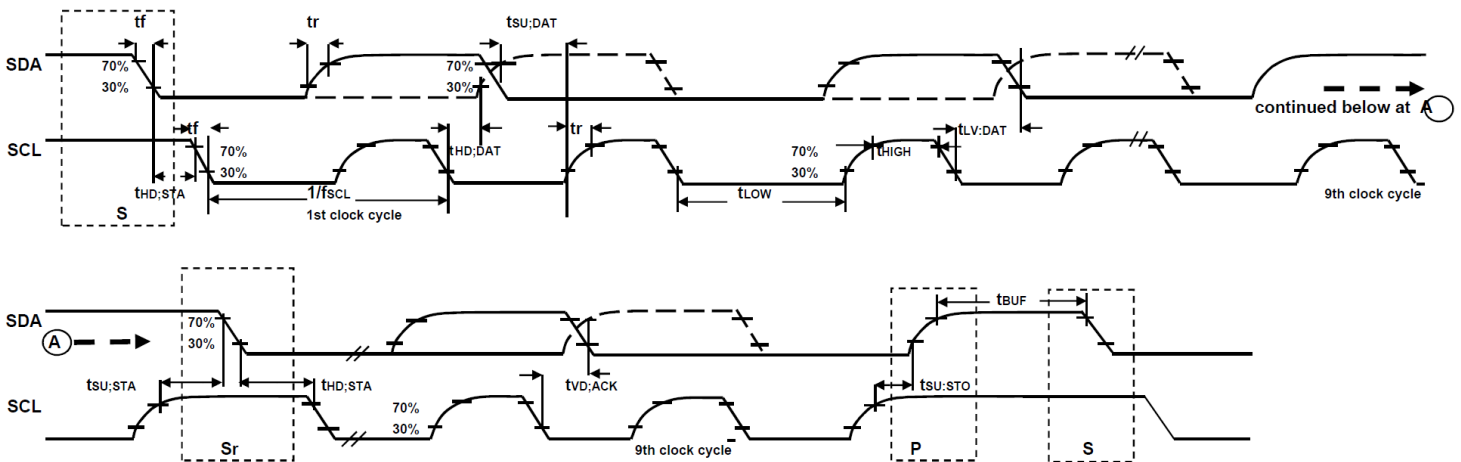
Parameters	Conditions	Min	Typical	Max	Units	Notes
INTERNAL CLOCK SOURCE	CLK_SEL=0,1,2,3					
Gyroscope Sample Rate, Fast	DLPFCFG=0 SAMPLERATEDIV = 0		8		kHz	
Gyroscope Sample Rate, Slow	DLPFCFG=1,2,3,4,5, or 6 SAMPLERATEDIV = 0		1		kHz	
Accelerometer Sample Rate			1		kHz	
Clock Frequency Initial Tolerance	CLK_SEL=0, 25°C	-5		+5	%	
Frequency Variation over Temperature	CLK_SEL=1,2,3; 25°C	-1		+1	%	
PLL Settling Time	CLK_SEL=0 CLK_SEL=1,2,3		-15 to +10 ±1 1		% %	
				10	ms	
EXTERNAL 32.768kHz CLOCK	CLK_SEL=4					
External Clock Frequency			32.768		kHz	
External Clock Allowable Jitter	Cycle-to-cycle rms		1 to 2		µs	
Gyroscope Sample Rate, Fast	DLPFCFG=0 SAMPLERATEDIV = 0		8.192		kHz	
Gyroscope Sample Rate, Slow	DLPFCFG=1,2,3,4,5, or 6 SAMPLERATEDIV = 0		1.024		kHz	
Accelerometer Sample Rate			1.024		kHz	
PLL Settling Time			1	10	ms	
EXTERNAL 19.2MHz CLOCK	CLK_SEL=5					
External Clock Frequency			19.2		MHz	
Gyroscope Sample Rate	Full programmable range	3.9		8000	Hz	
Gyroscope Sample Rate, Fast Mode	DLPFCFG=0 SAMPLERATEDIV = 0		8		kHz	
Gyroscope Sample Rate, Slow Mode	DLPFCFG=1,2,3,4,5, or 6 SAMPLERATEDIV = 0		1		kHz	
Accelerometer Sample Rate			1		kHz	
PLL Settling Time			1	10	ms	

6.7 I²C Timing Characterization

Typical Operating Circuit of Section 7.2, VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T_A = 25°C

Parameters	Conditions	Min	Typical	Max	Units	Notes
I²C TIMING						
I²C FAST-MODE						
f _{SCL} , SCL Clock Frequency				400	kHz	
t _{HD,STA} , (Repeated) START Condition Hold Time		0.6			µs	
t _{LOW} , SCL Low Period		1.3			µs	
t _{HIGH} , SCL High Period		0.6			µs	
t _{SU,STA} , Repeated START Condition Setup Time		0.6			µs	
t _{HD,DAT} , SDA Data Hold Time		0			µs	
t _{SU,DAT} , SDA Data Setup Time		100			ns	
t _r , SDA and SCL Rise Time	C _b bus cap. from 10 to 400pF	20+0.1C _b		300	ns	
t _f , SDA and SCL Fall Time	C _b bus cap. from 10 to 400pF	20+0.1C _b		300	ns	
t _{SU,STO} , STOP Condition Setup Time		0.6			µs	
t _{BUF} , Bus Free Time Between STOP and START Condition		1.3			µs	
C _b , Capacitive Load for each Bus Line			< 400		pF	
t _{VD,DAT} , Data Valid Time				0.9	µs	
t _{VD,ACK} , Data Valid Acknowledge Time				0.9	µs	

Note: Timing Characteristics apply to both Primary and Auxiliary I²C Bus

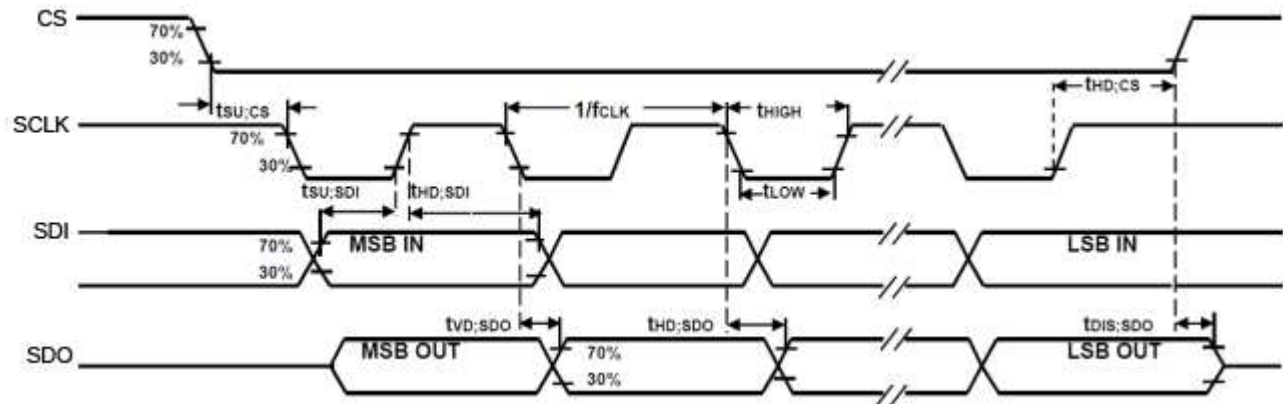


I²C Bus Timing Diagram

6.8 SPI Timing Characterization (MPU-6000 only)

Typical Operating Circuit of Section 7.2, VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T_A = 25°C, unless otherwise noted.

Parameters	Conditions	Min	Typical	Max	Units	Notes
SPI TIMING						
f _{SCLK} , SCLK Clock Frequency				1	MHz	
t _{LOW} , SCLK Low Period		400			ns	
t _{HIGH} , SCLK High Period		400			ns	
t _{SU,CS} , CS Setup Time		8			ns	
t _{HD,CS} , CS Hold Time		500			ns	
t _{SU,SDI} , SDI Setup Time		11			ns	
t _{HD,SDI} , SDI Hold Time		7			ns	
t _{VD,SDO} , SDO Valid Time	C _{load} = 20pF			100	ns	
t _{HD,SDO} , SDO Hold Time	C _{load} = 20pF	4			ns	
t _{DIS,SDO} , SDO Output Disable Time				10	ns	



SPI Bus Timing Diagram

**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.

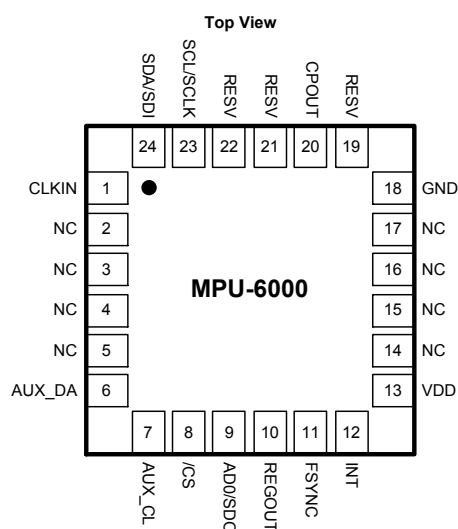
Parameter	Rating
Supply Voltage, VDD	-0.5V to +6V
VLOGIC Input Voltage Level (MPU-6050)	-0.5V to VDD + 0.5V
REGOUT	-0.5V to 2V
Input Voltage Level (CLKIN, AUX_DA, AD0, FSYNC, INT, SCL, SDA)	-0.5V to VDD + 0.5V
CPOUT (2.5V ≤ VDD ≤ 3.6V)	-0.5V to 30V
Acceleration (Any Axis, unpowered)	10,000g for 0.2ms
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



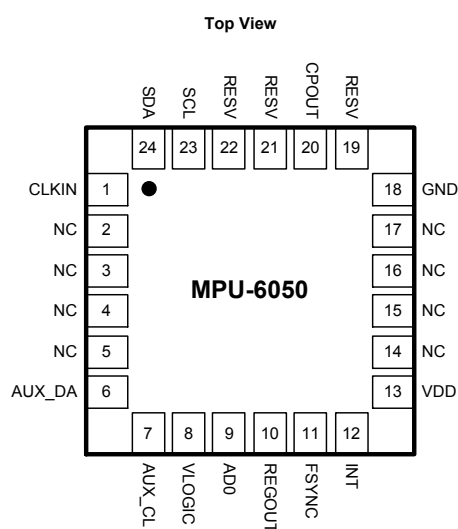
7 Applications Information

7.1 Pin Out and Signal Description

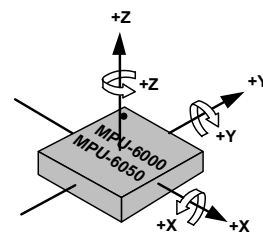
Pin Number	MPU-6000	MPU-6050	Pin Name	Pin Description
1	Y	Y	CLKIN	Optional external reference clock input. Connect to GND if unused.
6	Y	Y	AUX_DA	I ² C master serial data, for connecting to external sensors
7	Y	Y	AUX_CL	I ² C Master serial clock, for connecting to external sensors
8	Y		/CS	SPI chip select (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 0.9mm

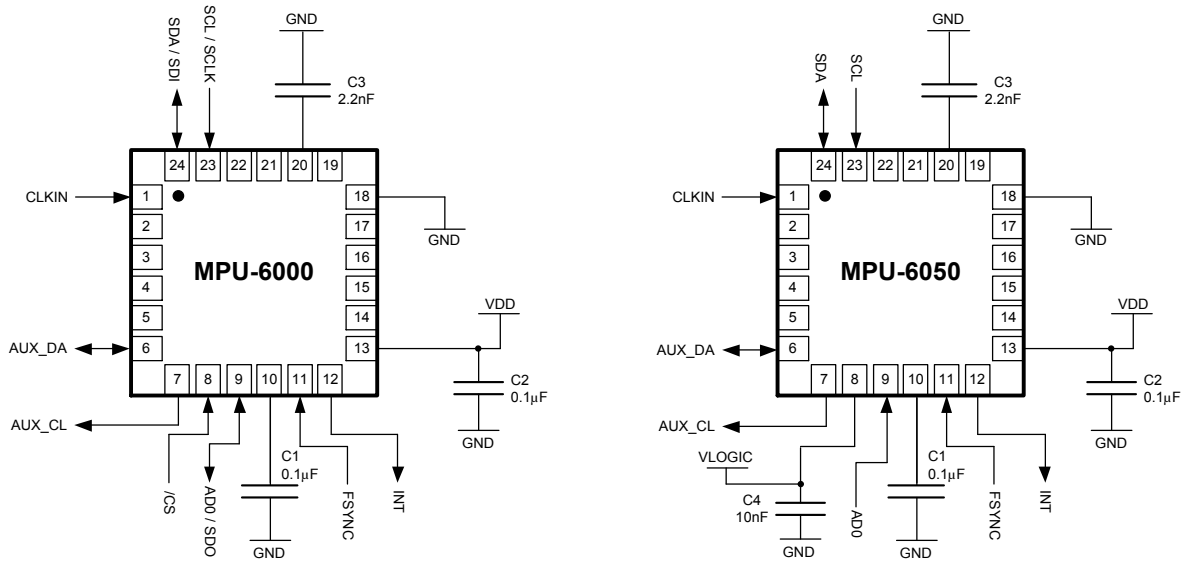


QFN Package
24-pin, 4mm x 4mm x 0.9mm



Orientation of Axes of Sensitivity and
Polarity of Rotation

7.2 Typical Operating Circuit

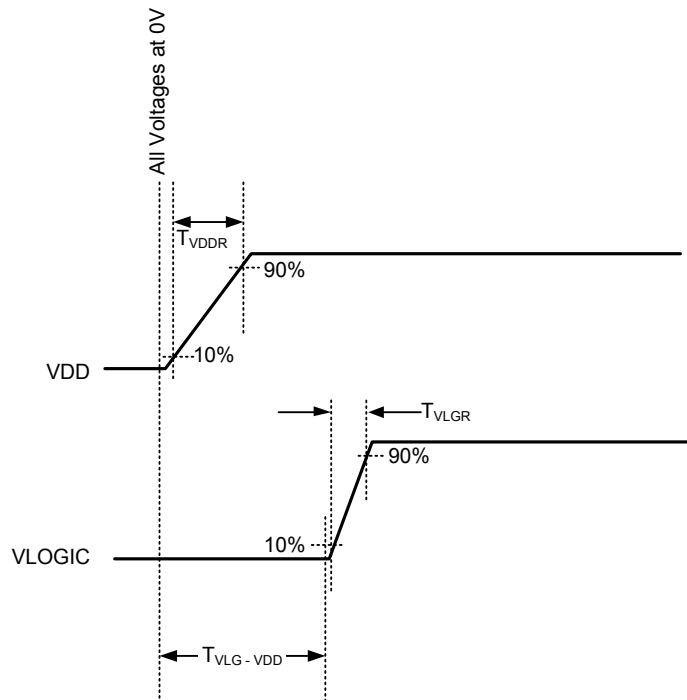


Typical Operating Circuits

7.3 Bill of Materials for External Components

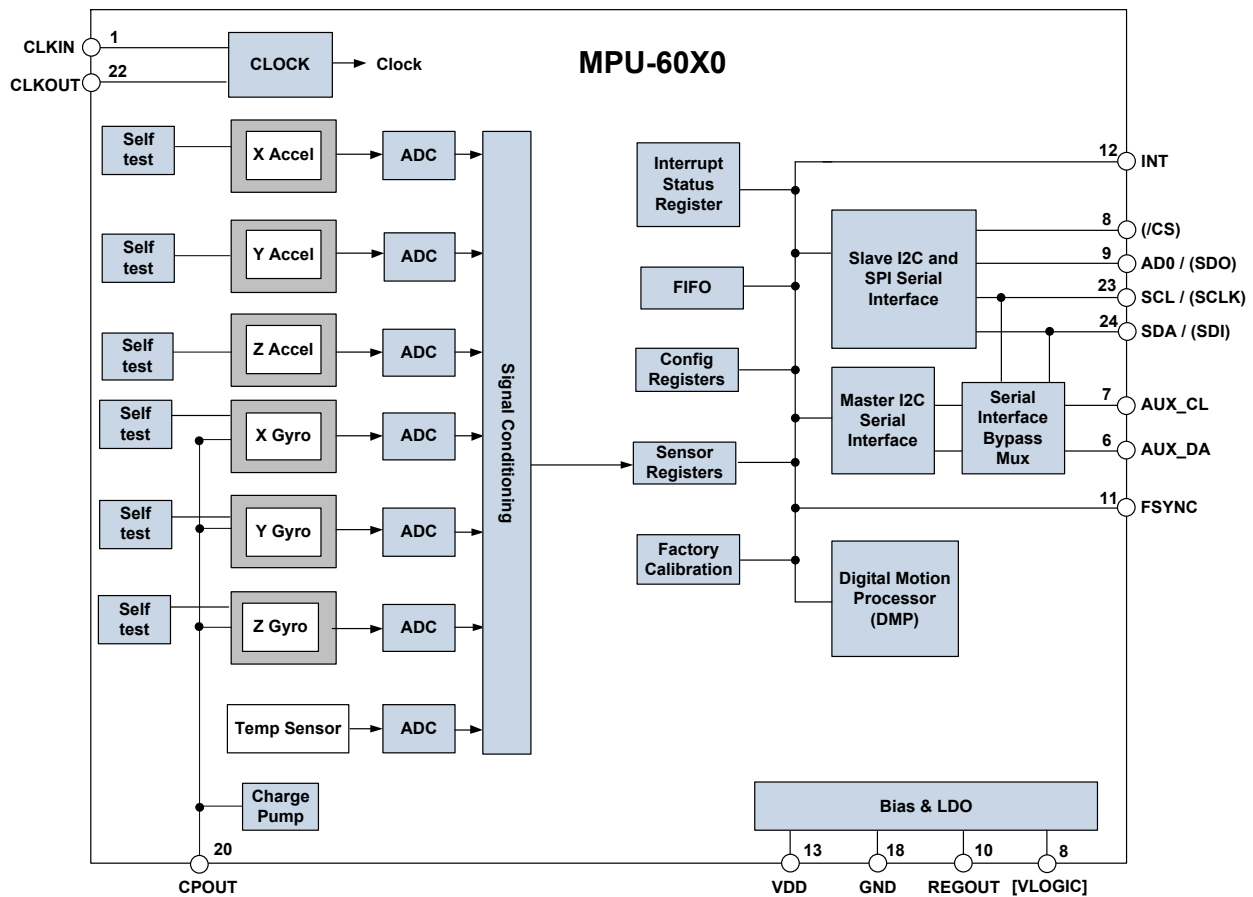
Component	Label	Specification	Quantity
Regulator Filter Capacitor (Pin 10)	C1	Ceramic, X7R, 0.1µF ±10%, 2V	1
VDD Bypass Capacitor (Pin 13)	C2	Ceramic, X7R, 0.1µF ±10%, 4V	1
Charge Pump Capacitor (Pin 20)	C3	Ceramic, X7R, 2.2nF ±10%, 50V	1
VLOGIC Bypass Capacitor (Pin 8)	C4*	Ceramic, X7R, 10nF ±10%, 4V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

7.11 Auxiliary I²C Serial Interface

The MPU-60X0 has an auxiliary I²C bus for communicating to an off-chip 3-Axis digital output magnetometer or other sensors. This bus has two operating modes:

- I²C Master Mode: The MPU-60X0 acts as a master to any external sensors connected to the auxiliary I²C bus
- Pass-Through Mode: The MPU-60X0 directly connects the primary and auxiliary I²C buses together, allowing the system processor to directly communicate with any external sensors.

Auxiliary I²C Bus Modes of Operation:

- I²C Master Mode: Allows the MPU-60X0 to directly access the data registers of external digital sensors, such as a magnetometer. In this mode, the MPU-60X0 directly obtains data from auxiliary sensors, allowing the on-chip DMP to generate sensor fusion data without intervention from the system applications processor.

For example, In I²C Master mode, the MPU-60X0 can be configured to perform burst reads, returning the following data from a magnetometer:

- X magnetometer data (2 bytes)
- Y magnetometer data (2 bytes)
- Z magnetometer data (2 bytes)

The I²C Master can be configured to read up to 24 bytes from up to 4 auxiliary sensors. A fifth sensor can be configured to work single byte read/write mode.

- Pass-Through Mode: Allows an external system processor to act as master and directly communicate to the external sensors connected to the auxiliary I²C bus pins (AUX_DA and AUX_CL). In this mode, the auxiliary I²C bus control logic (3rd party sensor interface block) of the MPU-60X0 is disabled, and the auxiliary I²C pins AUX_DA and AUX_CL (Pins 6 and 7) are connected to the main I²C bus (Pins 23 and 24) through analog switches.

Pass-Through Mode is useful for configuring the external sensors, or for keeping the MPU-60X0 in a low-power mode when only the external sensors are used.

In Pass-Through Mode the system processor can still access MPU-60X0 data through the I²C interface.

Auxiliary I²C Bus IO Logic Levels

- MPU-6000: The logic level of the auxiliary I²C bus is VDD
- MPU-6050: The logic level of the auxiliary I²C bus can be programmed to be either VDD or VLOGIC

For further information regarding the MPU-6050's logic levels, please refer to Section 10.2.



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:

Self-test response = Sensor output with self-test enabled – 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).

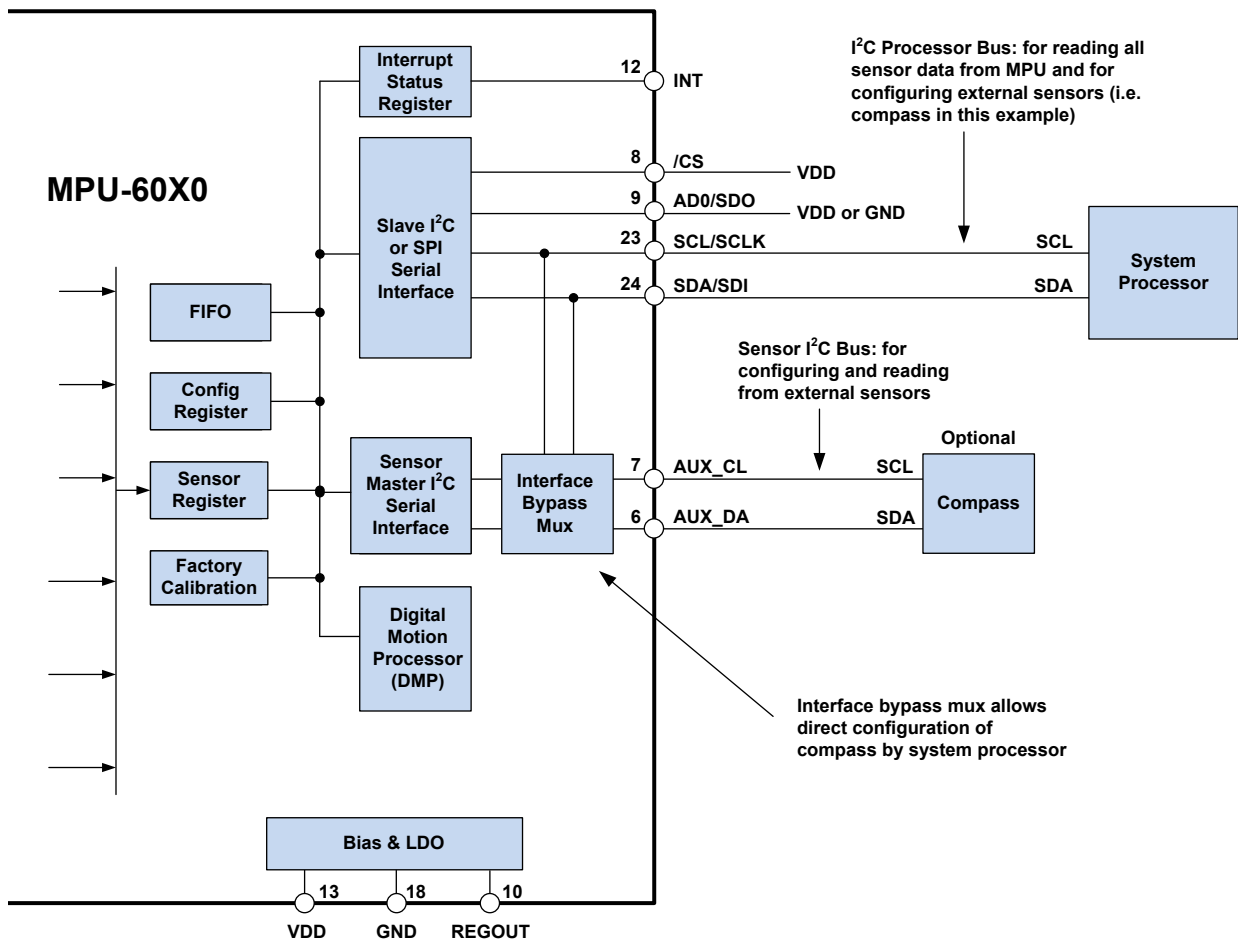
When the value of the self-test response is within the min/max limits of the product specification, the part has passed self test. When the self-test response exceeds the min/max values, the part is deemed to have failed self-test. Code for operating self test code is included within the MotionApps software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

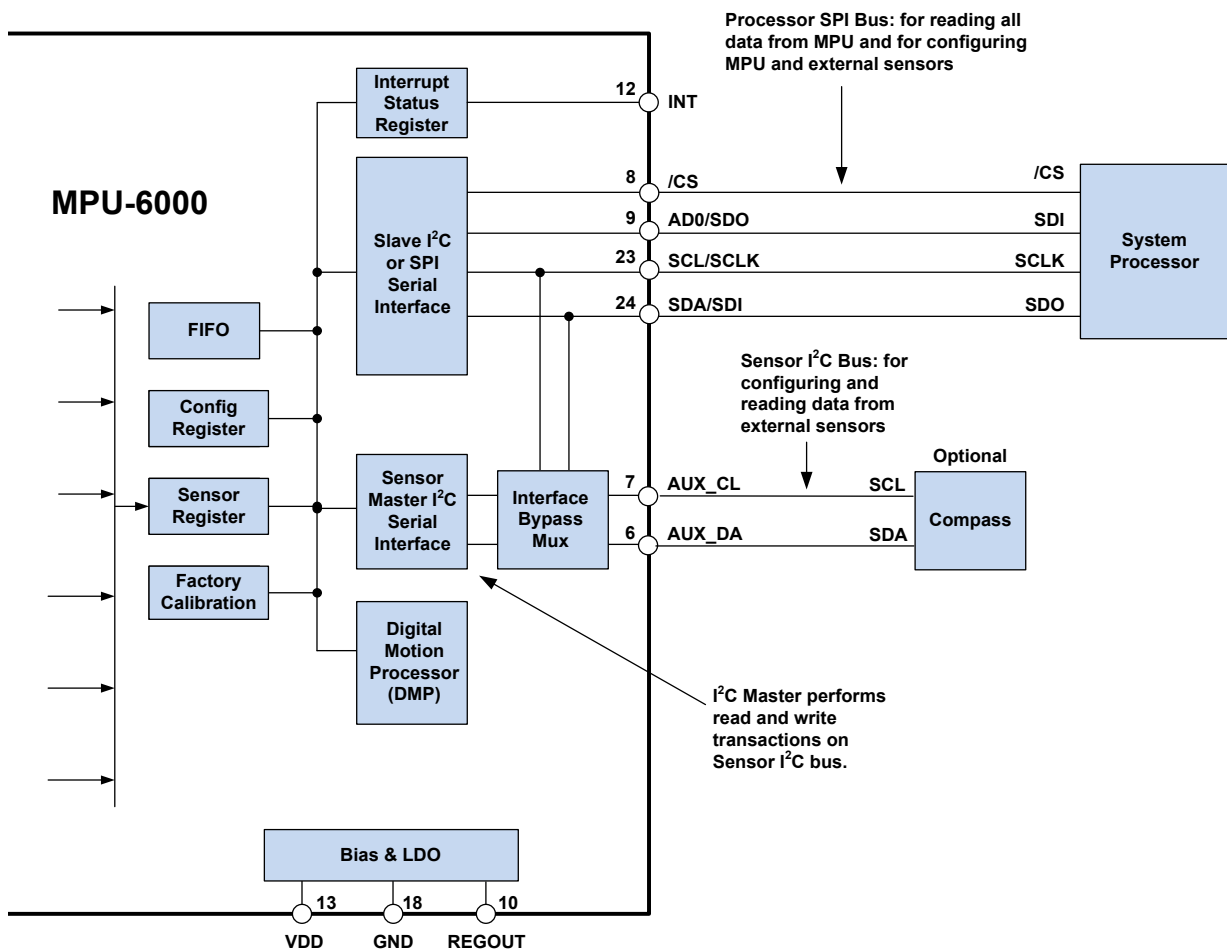
In the figure below, the system processor is an SPI master to the MPU-6000. Pins 8, 9, 23, and 24 are used to support the /CS, SDO, SCLK, and SDI signals for SPI communications. Because these SPI pins are shared with the I²C slave pins (9, 23 and 24), the system processor cannot access the auxiliary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



8 Programmable Interrupts

The MPU-60X0 has a programmable interrupt system which can generate an interrupt signal on the INT pin. Status flags indicate the source of an interrupt. Interrupt sources may be enabled and disabled individually.

Table of Interrupt Sources

Interrupt Name	Module
FIFO Overflow	FIFO
Data Ready	Sensor Registers
I ² C Master errors: Lost Arbitration, NACKs	I ² C Master
I ² C Slave 4	I ² C Master

For information regarding the interrupt enable/disable registers and flag registers, please refer to the MPU-6000/MPU-6050 Register Map and Register Descriptions document. Some interrupt sources are explained below.



9 Digital Interface

9.1 I²C and SPI (MPU-6000 only) Serial Interfaces

The internal registers and memory of the MPU-6000/MPU-6050 can be accessed using either I²C at 400 kHz or SPI at 1MHz (MPU-6000 only). SPI operates in four-wire mode.

Serial Interface

Pin Number	MPU-6000	MPU-6050	Pin Name	Pin Description
8	Y		/CS	SPI chip select (0=SPI enable)
8		Y	VLOGIC	Digital I/O supply voltage. VLOGIC must be \leq VDD at all times.
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
23	Y		SCL / SCLK	I ² C serial clock (SCL); SPI serial clock (SCLK)
23		Y	SCL	I ² C serial clock
24	Y		SDA / SDI	I ² C serial data (SDA); SPI serial data input (SDI)
24		Y	SDA	I ² C serial data

Note:

To prevent switching into I²C mode when using SPI (MPU-6000), the I²C interface should be disabled by setting the *I2C_IF_DIS* configuration bit. Setting this bit should be performed immediately after waiting for the time specified by the “Start-Up Time for Register Read/Write” in Section 6.3.

For further information regarding the *I2C_IF_DIS* bit, please refer to the MPU-6000/MPU-6050 Register Map and Register Descriptions document.

9.2 I²C Interface

I²C is a two-wire interface comprised of the signals serial data (SDA) and serial clock (SCL). In general, the lines are open-drain and bi-directional. In a generalized I²C interface implementation, attached devices can be a master or a slave. The master device puts the slave address on the bus, and the slave device with the matching address acknowledges the master.

The MPU-60X0 always operates as a slave device when communicating to the system processor, which thus acts as the master. SDA and SCL lines typically need pull-up resistors to VDD. The maximum bus speed is 400 kHz.

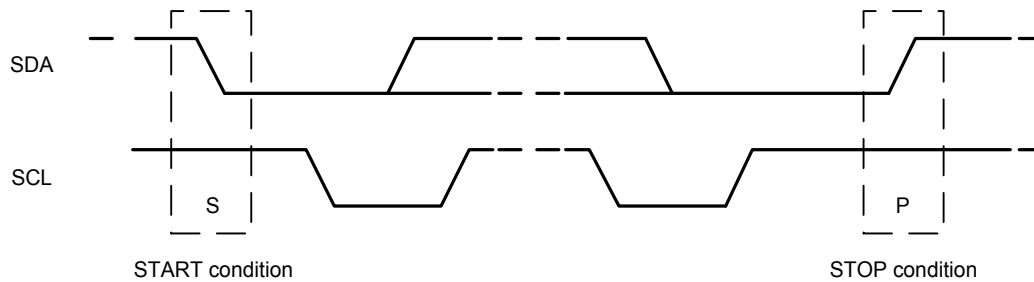
The slave address of the MPU-60X0 is b110100X which is 7 bits long. The LSB bit of the 7 bit address is determined by the logic level on pin AD0. This allows two MPU-60X0s to be connected to the same I²C bus. When used in this configuration, the address of the one of the devices should be b1101000 (pin AD0 is logic low) and the address of the other should be b1101001 (pin AD0 is logic high).

9.3 I²C Communications Protocol

START (S) and STOP (P) Conditions

Communication on the I²C bus starts when the master puts the START condition (S) on the bus, which is defined as a HIGH-to-LOW transition of the SDA line while SCL line is HIGH (see figure below). The bus is considered to be busy until the master puts a STOP condition (P) on the bus, which is defined as a LOW to HIGH transition on the SDA line while SCL is HIGH (see figure below).

Additionally, the bus remains busy if a repeated START (Sr) is generated instead of a STOP condition.

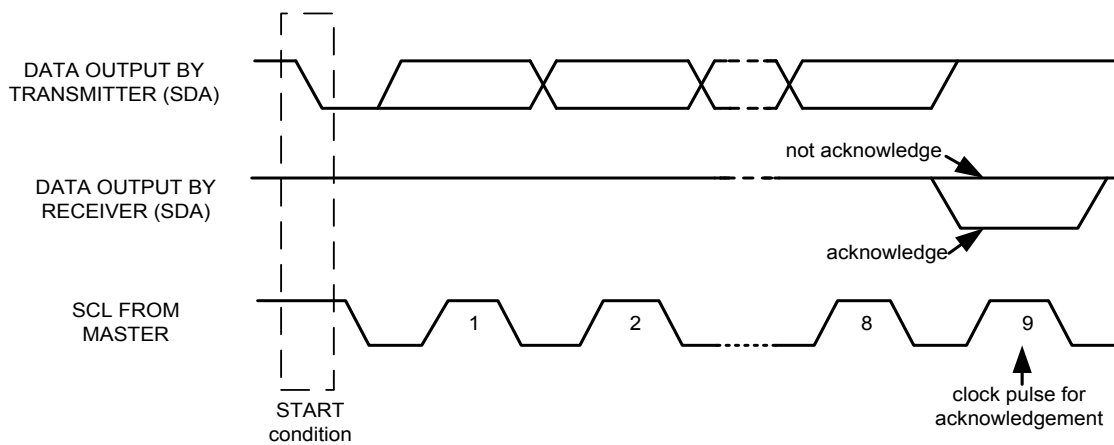


START and STOP Conditions

Data Format / Acknowledge

I²C data bytes are defined to be 8-bits long. There is no restriction to the number of bytes transmitted per data transfer. Each byte transferred must be followed by an acknowledge (ACK) signal. The clock for the acknowledge signal is generated by the master, while the receiver generates the actual acknowledge signal by pulling down SDA and holding it low during the HIGH portion of the acknowledge clock pulse.

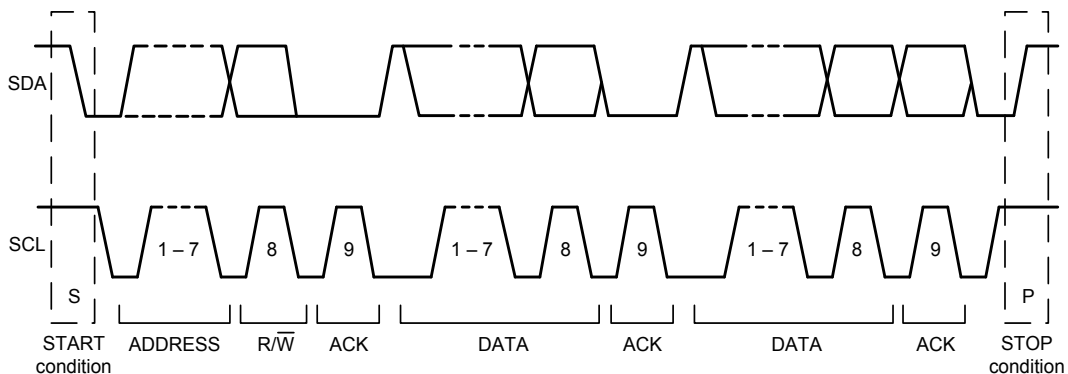
If a slave is busy and cannot transmit or receive another byte of data until some other task has been performed, it can hold SCL LOW, thus forcing the master into a wait state. Normal data transfer resumes when the slave is ready, and releases the clock line (refer to the following figure).



Acknowledge on the I²C Bus

Communications

After beginning communications with the START condition (S), the master sends a 7-bit slave address followed by an 8th bit, the read/write bit. The read/write bit indicates whether the master is receiving data from or is writing to the slave device. Then, the master releases the SDA line and waits for the acknowledge signal (ACK) from the slave device. Each byte transferred must be followed by an acknowledge bit. To acknowledge, the slave device pulls the SDA line LOW and keeps it LOW for the high period of the SCL line. Data transmission is always terminated by the master with a STOP condition (P), thus freeing the communications line. However, the master can generate a repeated START condition (Sr), and address another slave without first generating a STOP condition (P). A LOW to HIGH transition on the SDA line while SCL is HIGH defines the stop condition. All SDA changes should take place when SCL is low, with the exception of start and stop conditions.



Complete I²C Data Transfer

To write the internal MPU-60X0 registers, the master transmits the start condition (S), followed by the I²C address and the write bit (0). At the 9th clock cycle (when the clock is high), the MPU-60X0 acknowledges the transfer. Then the master puts the register address (RA) on the bus. After the MPU-60X0 acknowledges the reception of the register address, the master puts the register data onto the bus. This is followed by the ACK signal, and data transfer may be concluded by the stop condition (P). To write multiple bytes after the last ACK signal, the master can continue outputting data rather than transmitting a stop signal. In this case, the MPU-60X0 automatically increments the register address and loads the data to the appropriate register. The following figures show single and two-byte write sequences.

Single-Byte Write Sequence

Master	S	AD+W		RA		DATA		P
Slave			ACK		ACK		ACK	

Burst Write Sequence

Master	S	AD+W		RA		DATA		DATA		P
Slave			ACK		ACK		ACK		ACK	



To read the internal MPU-60X0 registers, the master sends a start condition, followed by the I²C address and a write bit, and then the register address that is going to be read. Upon receiving the ACK signal from the MPU-60X0, the master transmits a start signal followed by the slave address and read bit. As a result, the MPU-60X0 sends an ACK signal and the data. The communication ends with a not acknowledge (NACK) signal and a stop bit from master. The NACK condition is defined such that the SDA line remains high at the 9th clock cycle. The following figures show single and two-byte read sequences.

Single-Byte Read Sequence

Master	S	AD+W		RA		S	AD+R			NACK	P
Slave			ACK		ACK			ACK	DATA		

Burst Read Sequence

Master	S	AD+W		RA		S	AD+R			ACK		NACK	P
Slave			ACK		ACK			ACK	DATA		DATA		

9.4 I²C Terms

Signal	Description
S	Start Condition: SDA goes from high to low while SCL is high
AD	Slave I ² C address
W	Write bit (0)
R	Read bit (1)
ACK	Acknowledge: SDA line is low while the SCL line is high at the 9 th clock cycle
NACK	Not-Acknowledge: SDA line stays high at the 9 th clock cycle
RA	MPU-60X0 internal register address
DATA	Transmit or received data
P	Stop condition: SDA going from low to high while SCL is high

9.5 SPI Interface (MPU-6000 only)

SPI is a 4-wire synchronous serial interface that uses two control lines and two data lines. The MPU-6000 always operates as a Slave device during standard Master-Slave SPI operation.

With respect to the Master, the Serial Clock output (SCLK), the Serial Data Output (SDO) and the Serial Data Input (SDI) are shared among the Slave devices. Each SPI slave device requires its own Chip Select (/CS) line from the master.

/CS goes low (active) at the start of transmission and goes back high (inactive) at the end. Only one /CS line is active at a time, ensuring that only one slave is selected at any given time. The /CS lines of the non-selected slave devices are held high, causing their SDO lines to remain in a high-impedance (high-z) state so that they do not interfere with any active devices.

SPI Operational Features

1. Data is delivered MSB first and LSB last
2. Data is latched on the rising edge of SCLK
3. Data should be transitioned on the falling edge of SCLK
4. The maximum frequency of SCLK is 1MHz
5. SPI read and write operations are completed in 16 or more clock cycles (two or more bytes). The first byte contains the SPI Address, and the following byte(s) contain(s) the SPI data. The first bit of the first byte contains the Read/Write bit and indicates the Read (1) or Write (0) operation. The following 7 bits contain the Register Address. In cases of multiple-byte Read/Writes, data is two or more bytes:

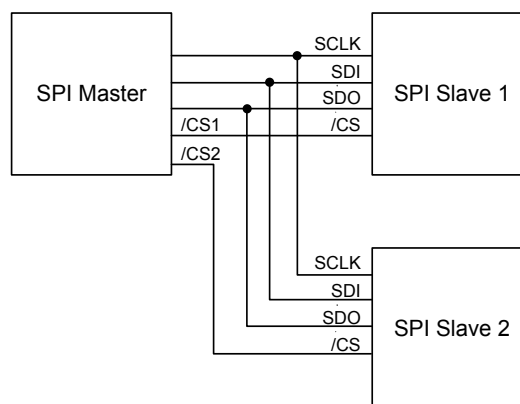
SPI Address format

MSB							LSB
R/W	A6	A5	A4	A3	A2	A1	A0

SPI Data format

MSB							LSB
D7	D6	D5	D4	D3	D2	D1	D0

6. Supports Single or Burst Read/Writes.



Typical SPI Master / Slave Configuration



10 Serial Interface Considerations (MPU-6050)

10.1 MPU-6050 Supported Interfaces

The MPU-6050 supports I²C communications on both its primary (microprocessor) serial interface and its auxiliary interface.

10.2 Logic Levels

The MPU-6050's I/O logic levels are set to be VLOGIC, as shown in the table below. AUX_VDDIO must be set to 0.

I/O Logic Levels vs. AUX_VDDIO

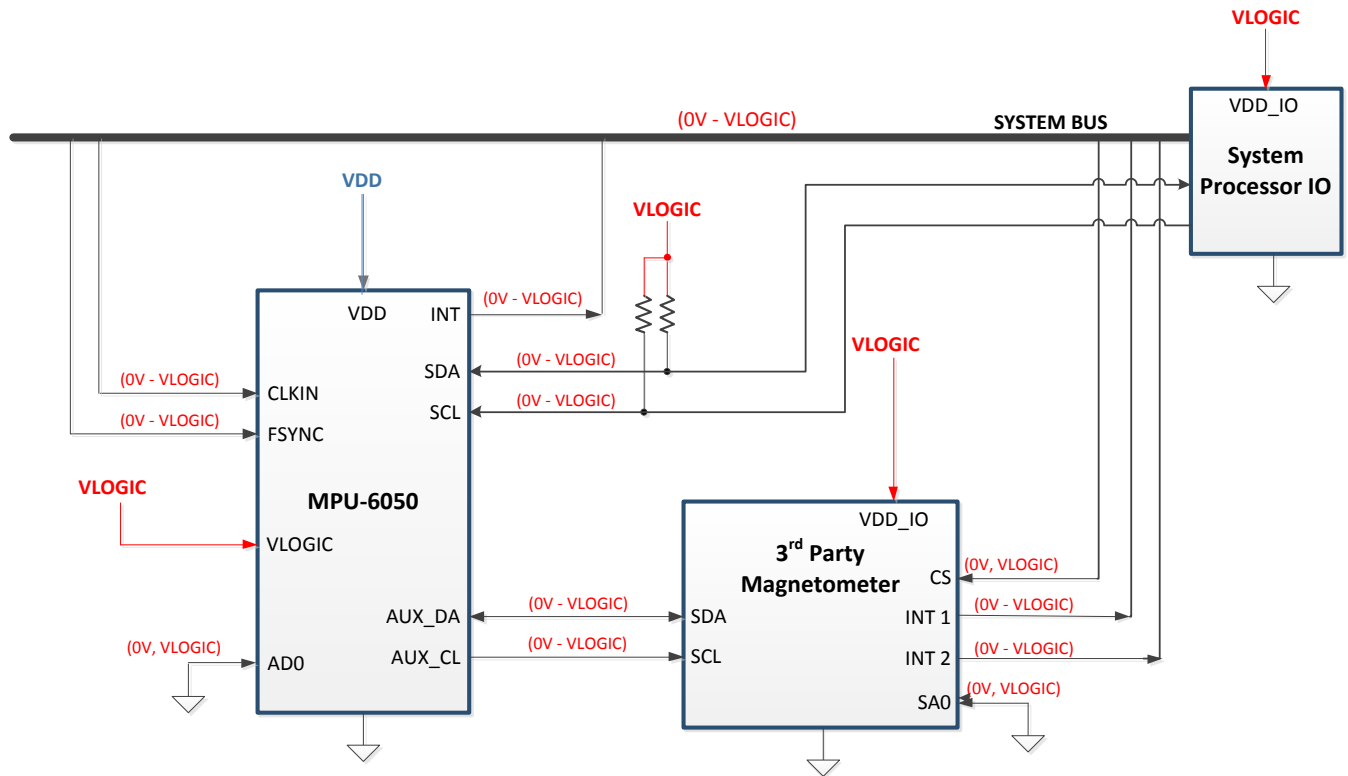
AUX_VDDIO	MICROPROCESSOR LOGIC LEVELS (Pins: SDA, SCL, AD0, CLKIN, INT)	AUXILIARY LOGIC LEVELS (Pins: AUX_DA, AUX_CL)
0	VLOGIC	VLOGIC

Note: The power-on-reset value for AUX_VDDIO is 0.

When AUX_VDDIO is set to 0 (its power-on-reset value), VLOGIC is the power supply voltage for both the microprocessor system bus and the auxiliary I²C bus, as shown in the figure of Section 10.3.

10.3 Logic Levels Diagram for AUX_VDDIO = 0

The figure below depicts a sample circuit with a third party magnetometer attached to the auxiliary I²C bus. It shows logic levels and voltage connections for AUX_VDDIO = 0. Note: Actual configuration will depend on the auxiliary sensors used.



I/O Levels and Connections for AUX_VDDIO = 0

Notes:

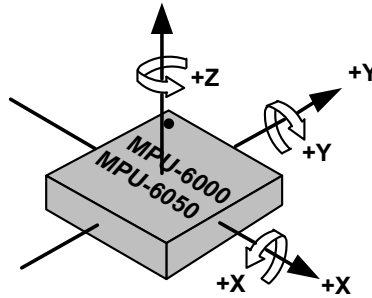
1. AUX_VDDIO determines the IO voltage levels of AUX_DA and AUX_CL (0 = set output levels relative to VLOGIC)
2. All other MPU-6050 logic IOs are referenced to VLOGIC.

11 Assembly

This section provides general guidelines for assembling InvenSense Micro Electro-Mechanical Systems (MEMS) gyros packaged in Quad Flat No leads package (QFN) surface mount integrated circuits.

11.1 Orientation of Axes

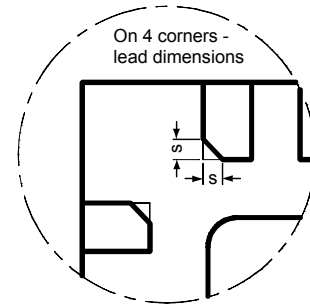
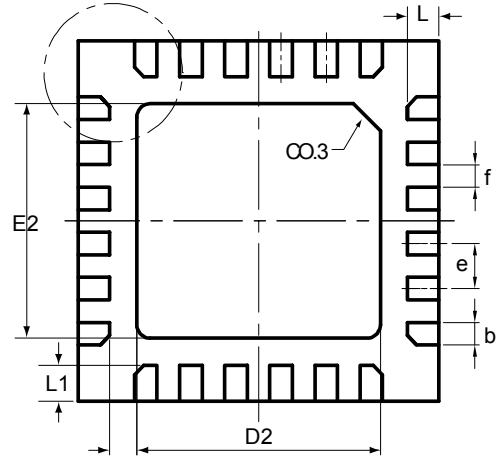
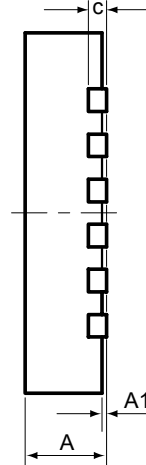
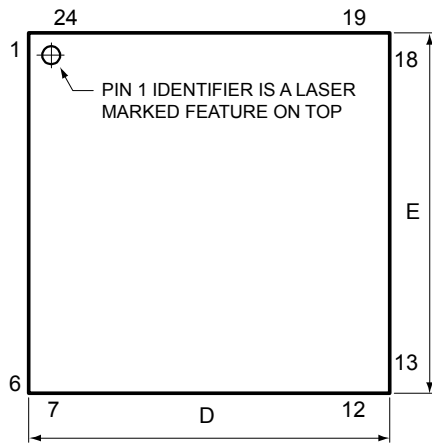
The diagram below shows the orientation of the axes of sensitivity and the polarity of rotation. Note the pin 1 identifier (•) in the figure.



Orientation of Axes of Sensitivity and
Polarity of Rotation

11.2 Package Dimensions

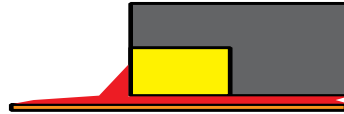
24 Lead QFN (4x4x0.9) mm NiPdAu Lead-frame finish



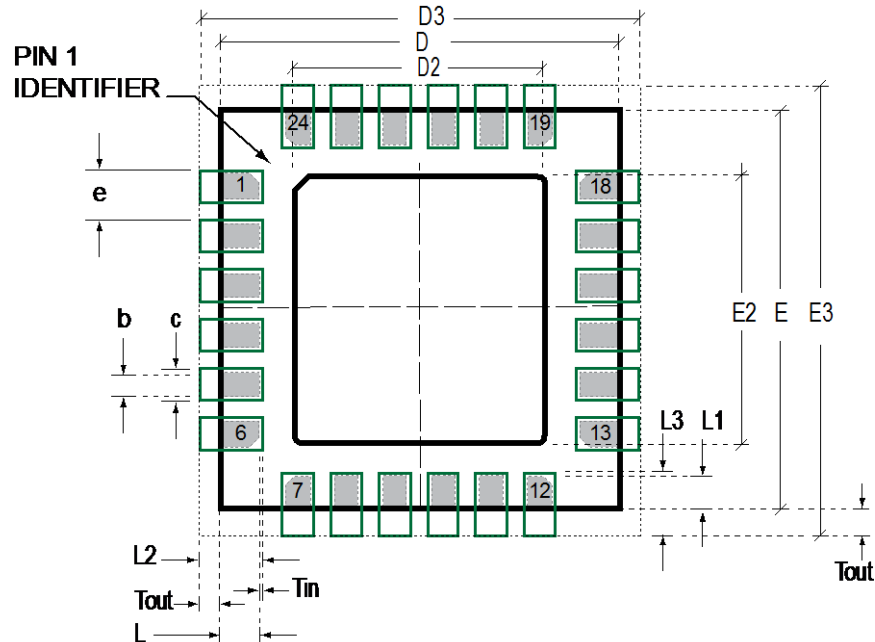
SYMBOLS	DIMENSIONS IN MILLIMETERS		
	MIN	NOM	MAX
A	0.85	0.90	0.95
A1	0.00	0.02	0.05
b	0.18	0.25	0.30
c	---	0.20 REF	---
D	3.90	4.00	4.10
D2	2.65	2.70	2.75
E	3.90	4.00	4.10
E2	2.55	2.60	2.65
e	---	0.50	---
f (e-b)	---	0.25	---
K	0.25	0.30	0.35
L	0.30	0.35	0.40
L1	0.35	0.40	0.45
s	0.05	---	0.15

11.3 PCB Design Guidelines

The Pad Diagram using a JEDEC type extension with solder rising on the outer edge is shown below. The Pad Dimensions Table shows pad sizing (mean dimensions) recommended for the MPU-60X0 product.



JEDEC type extension with solder rising on outer edge


PCB Layout Diagram

SYMBOLS	DIMENSIONS IN MILLIMETERS	NOM
Nominal Package I/O Pad Dimensions		
e	Pad Pitch	0.50
b	Pad Width	0.25
L	Pad Length	0.35
L1	Pad Length	0.40
D	Package Width	4.00
E	Package Length	4.00
D2	Exposed Pad Width	2.70
E2	Exposed Pad Length	2.60
I/O Land Design Dimensions (Guidelines)		
D3	I/O Pad Extent Width	4.80
E3	I/O Pad Extent Length	4.80
c	Land Width	0.35
Tout	Outward Extension	0.40
Tin	Inward Extension	0.05
L2	Land Length	0.80
L3	Land Length	0.85

PCB Dimensions Table (for PCB Lay-out Diagram)

11.4 Assembly Precautions

11.4.1 Gyroscope Surface Mount Guidelines

InvenSense MEMS Gyros sense rate of rotation. In addition, gyroscopes sense mechanical stress coming from the printed circuit board (PCB). This PCB stress can be minimized by adhering to certain design rules:

When using MEMS gyroscope components in plastic packages, PCB mounting and assembly can cause package stress. This package stress in turn can affect the output offset and its value over a wide range of temperatures. This stress is caused by the mismatch between the Coefficient of Linear Thermal Expansion (CTE) of the package material and the PCB. Care must be taken to avoid package stress due to mounting.

Traces connected to pads should be as symmetric as possible. Maximizing symmetry and balance for pad connection will help component self alignment and will lead to better control of solder paste reduction after reflow.

Any material used in the surface mount assembly process of the MEMS gyroscope should be free of restricted RoHS elements or compounds. Pb-free solders should be used for assembly.

11.4.2 Exposed Die Pad Precautions

The MPU-60X0 has very low active and standby current consumption. The exposed die pad is not required for heat sinking, and should not be soldered to the PCB. Failure to adhere to this rule can induce performance changes due to package thermo-mechanical stress. There is no electrical connection between the pad and the CMOS.

11.4.3 Trace Routing

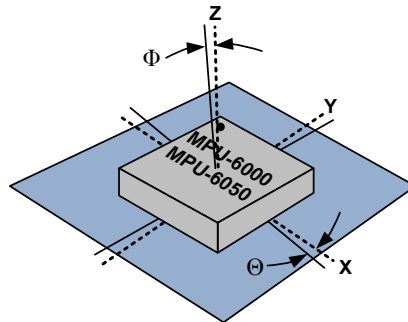
Routing traces or vias under the gyro package such that they run under the exposed die pad is prohibited. Routed active signals may harmonically couple with the gyro MEMS devices, compromising gyro response. These devices are designed with the drive frequencies as follows: X = $33\pm 3\text{KHz}$, Y = $30\pm 3\text{KHz}$, and Z = $27\pm 3\text{KHz}$. To avoid harmonic coupling don't route active signals in non-shielded signal planes directly below, or above the gyro package. Note: For best performance, design a ground plane under the e-pad to reduce PCB signal noise from the board on which the gyro device is mounted. If the gyro device is stacked under an adjacent PCB board, design a ground plane directly above the gyro device to shield active signals from the adjacent PCB board.

11.4.4 Component Placement

Do not place large insertion components such as keyboard or similar buttons, connectors, or shielding boxes at a distance of less than 6 mm from the MEMS gyro. Maintain generally accepted industry design practices for component placement near the MPU-60X0 to prevent noise coupling and thermo-mechanical stress.

11.4.5 PCB Mounting and Cross-Axis Sensitivity

Orientation errors of the gyroscope and accelerometer mounted to the printed circuit board can cause cross-axis sensitivity in which one gyro or accel responds to rotation or acceleration about another axis, respectively. For example, the X-axis gyroscope may respond to rotation about the Y or Z axes. The orientation mounting errors are illustrated in the figure below.



Package Gyro & Accel Axes (- - -) Relative to PCB Axes (———) with Orientation Errors (Θ and Φ)

The table below shows the cross-axis sensitivity as a percentage of the gyroscope or accelerometer's sensitivity for a given orientation error, respectively.

Cross-Axis Sensitivity vs. Orientation Error

Orientation Error (θ or Φ)	Cross-Axis Sensitivity ($\sin\theta$ or $\sin\Phi$)
0°	0%
0.5°	0.87%
1°	1.75%

The specifications for cross-axis sensitivity in Section 6.1 and Section 6.2 include the effect of the die orientation error with respect to the package.

11.4.6 MEMS Handling Instructions

MEMS (Micro Electro-Mechanical Systems) are a time-proven, robust technology used in hundreds of millions of consumer, automotive and industrial products. MEMS devices consist of microscopic moving mechanical structures. They differ from conventional IC products, even though they can be found in similar packages. Therefore, MEMS devices require different handling precautions than conventional ICs prior to mounting onto printed circuit boards (PCBs).

The MPU-60X0 has been qualified to a shock tolerance of 10,000g. InvenSense packages its gyroscopes as it deems proper for protection against normal handling and shipping. It recommends the following handling precautions to prevent potential damage.

- Do not drop individually packaged gyroscopes, or trays of gyroscopes onto hard surfaces. Components placed in trays could be subject to g-forces in excess of 10,000g if dropped.
- Printed circuit boards that incorporate mounted gyroscopes should not be separated by manually snapping apart. This could also create g-forces in excess of 10,000g.
- Do not clean MEMS gyroscopes in ultrasonic baths. Ultrasonic baths can induce MEMS damage if the bath energy causes excessive drive motion through resonant frequency coupling.

11.4.7 ESD Considerations

Establish and use ESD-safe handling precautions when unpacking and handling ESD-sensitive devices.

- Store ESD sensitive devices in ESD safe containers until ready for use. The Tape-and-Reel moisture-sealed bag is an ESD approved barrier. The best practice is to keep the units in the original moisture sealed bags until ready for assembly.

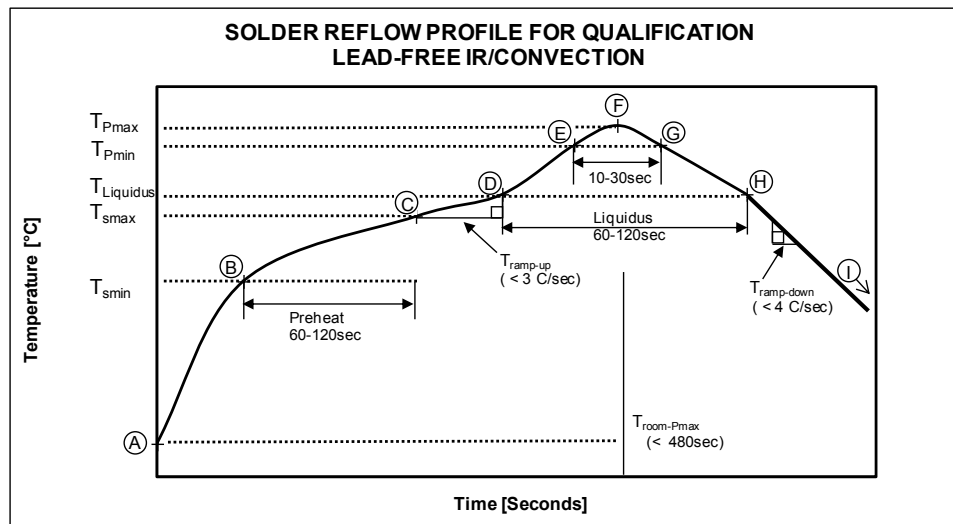
Restrict all device handling to ESD protected work areas that measure less than 200V static charge. Ensure that all workstations and personnel are properly grounded to prevent ESD.

11.4.8 Reflow Specification

Qualification Reflow: The MPU-60X0 was qualified in accordance with IPC/JEDEC J-STD-020D.1. This standard classifies proper packaging, storage and handling in order to avoid subsequent thermal and mechanical damage during the solder reflow attachment phase of PCB assembly.

The qualification preconditioning process specifies a sequence consisting of a bake cycle, a moisture soak cycle (in a temperature humidity oven), and three consecutive solder reflow cycles, followed by functional device testing.

The peak solder reflow classification temperature requirement for package qualification is (260 +5/-0°C) for lead-free soldering of components measuring less than 1.6 mm in thickness. The qualification profile and a table explaining the set-points are shown below:



Temperature Set Points Corresponding to Reflow Profile Above

Step	Setting	CONSTRAINTS		
		Temp (°C)	Time (sec)	Max. Rate (°C/sec)
A	T _{room}	25		
B	T _{Smin}	150		
C	T _{Smax}	200	60 < t _{BC} < 120	
D	T _{Liquidus}	217		r _(TLiquidus-TPmax) < 3
E	T _{Pmin} [255°C, 260°C]	255		r _(TLiquidus-TPmax) < 3
F	T _{Pmax} [260°C, 265°C]	260	t _{AF} < 480	r _(TLiquidus-TPmax) < 3
G	T _{Pmin} [255°C, 260°C]	255	10 < t _{EG} < 30	r _(TPmax-TLiquidus) < 4
H	T _{Liquidus}	217	60 < t _{DH} < 120	
I	T _{room}	25		

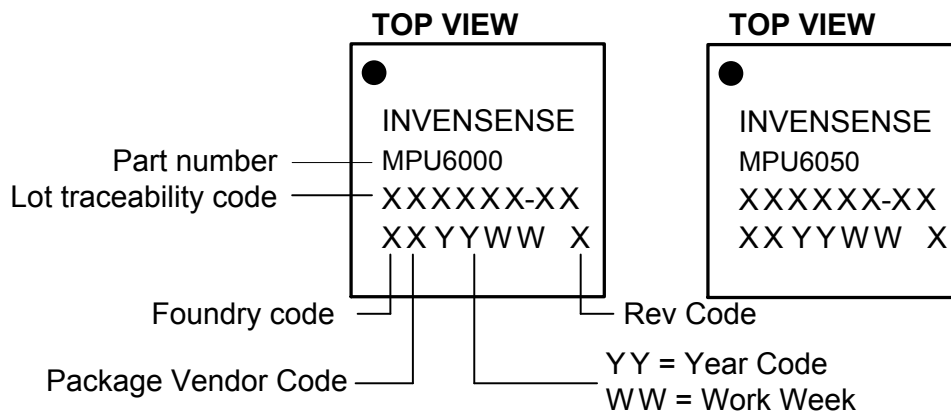
Notes: Customers must never exceed the Classification temperature (T_{Pmax} = 260°C).
 All temperatures refer to the topside of the QFN package, as measured on the package body surface.

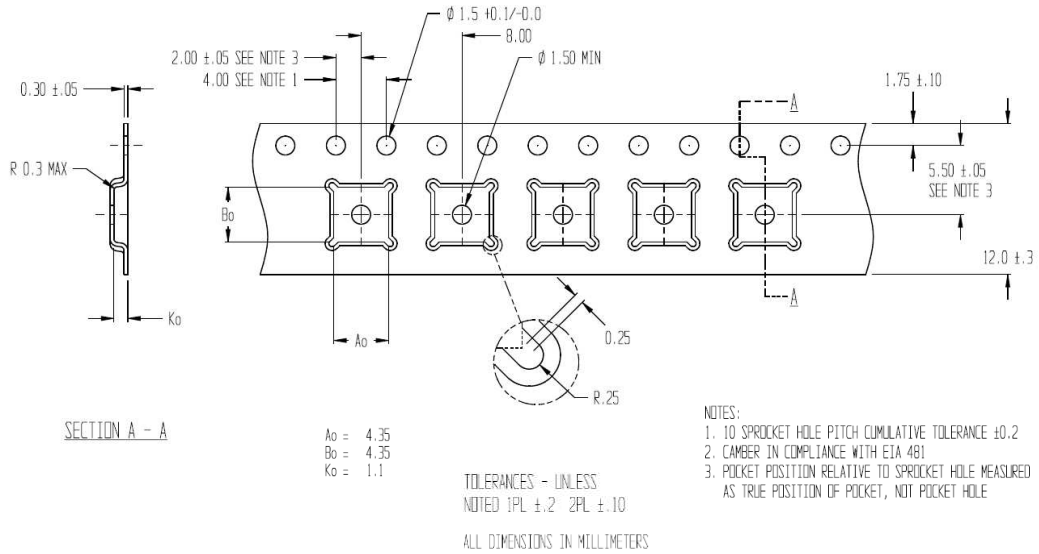
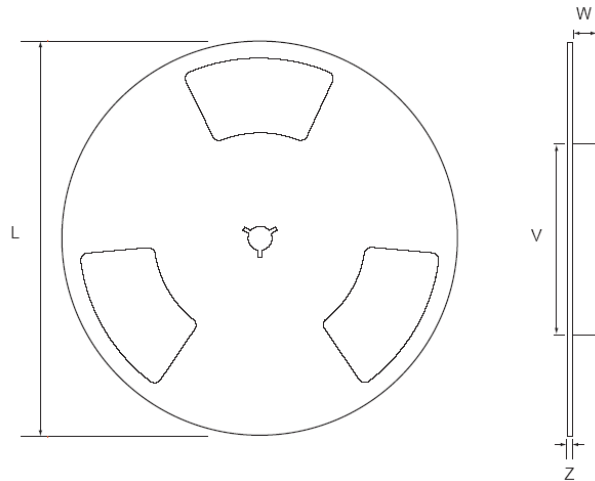
Production Reflow: Check the recommendations of your solder manufacturer. For optimum results, use lead-free solders that have lower specified temperature profiles (T_{pmax} ~ 235°C). Also use lower ramp-up and ramp-down rates than those used in the qualification profile. Never exceed the maximum conditions that we used for qualification, as these represent the maximum tolerable ratings for the device.

11.5 Storage Specifications

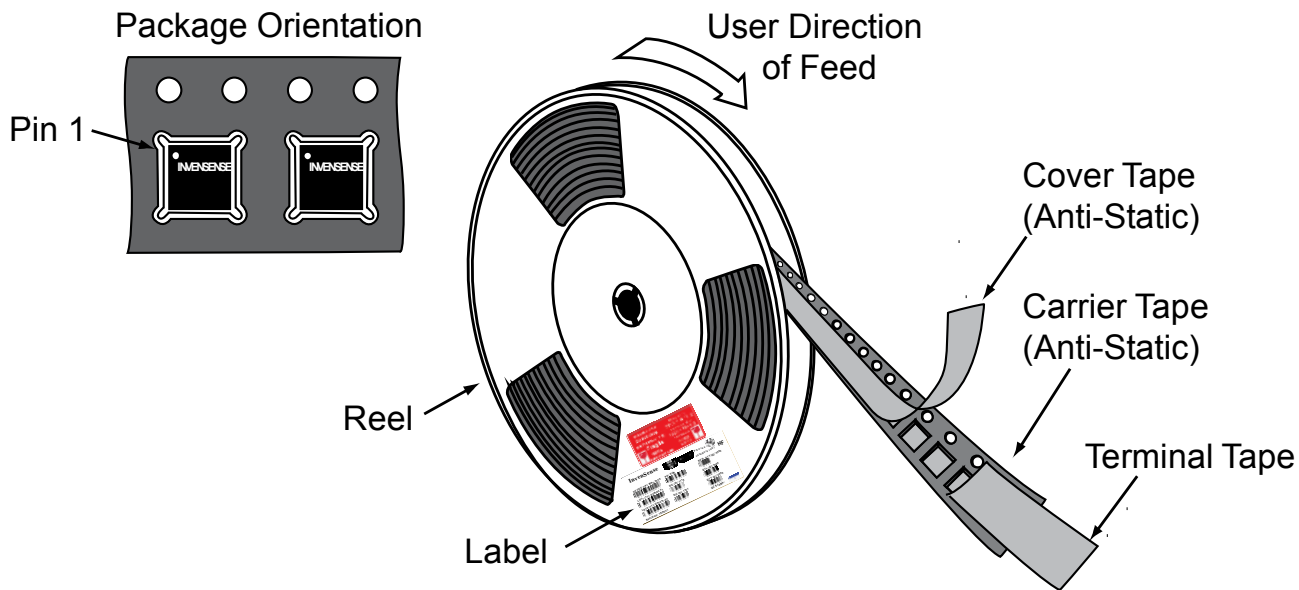
The storage specification of the MPU-60X0 conforms to IPC/JEDEC J-STD-020D.1 Moisture Sensitivity Level (MSL) 3.

Calculated shelf-life in moisture-sealed bag	12 months -- Storage conditions: <40°C and <90% RH
After opening moisture-sealed bag	168 hours -- Storage conditions: ambient ≤30°C at 60%RH

11.6 Package Marking Specification

Package Marking Specification

11.7 Tape & Reel Specification

Tape Dimensions

Reel Outline Drawing
Reel Dimensions and Package Size

PACKAGE SIZE	REEL (mm)			
	L	V	W	Z
4x4	330	102	12.8	2.3



Tape and Reel Specification

Reel Specifications

Quantity Per Reel	5,000
Reels per Box	1
Boxes Per Carton (max)	5
Pcs/Carton (max)	25,000

11.8 Label



Barcode Label



Location of Label on Reel

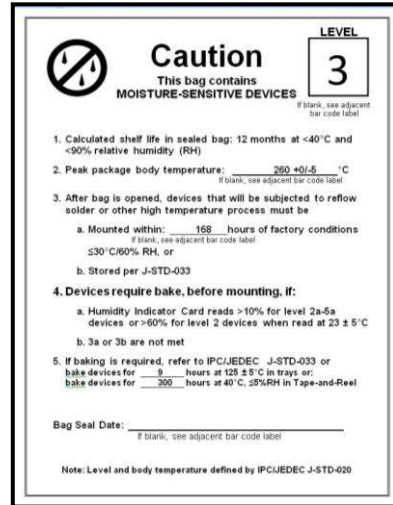
11.9 Packaging



REEL – with Barcode & Caution labels



Vacuum-Sealed Moisture Barrier Bag with ESD, MSL3, Caution, and Barcode Labels



MSL3 Label



Caution Label



ESD Label



Inner Bubble Wrap



Pizza Box



Pizza Boxes Placed in Foam-Lined Shipper Box



Outer Shipper Label



MPU-6000/MPU-6050 Product Specification

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

11.10 Representative Shipping Carton Label

		INV. NO: 111013-99	
From: InvenSense Taiwan, Ltd. 1F, 9 Prosperity 1st Road, Hsinchu Science Park, HsinChu City, 30078, Taiwan TEL: +886 3 6686999 FAX: +886 3 6686777		Ship To: Customer Name Street Address City, State, Country ZIP Attn: Buyer Name Phone: Buyer Phone Number	
SUPP PROD ID: MPU-6050			
LOT#: Q2R994-F1		LOT#:	
QTY: 5615		QTY: 0	
LOT#: Q3X785-G1		LOT#:	
QTY: 4385		QTY: 0	
LOT#: Q3Y196-02		LOT#:	
QTY: 5000		QTY: 0	
LOT#:		LOT#:	
QTY: 0		QTY: 0	
Total Quantity/Carton 15000		Weight: (KG) 4.05	
Pb-free	Shipping Carton:	Category (e4) HF	
	1	3	



12 Reliability

12.1 Qualification Test Policy

InvenSense's products complete a Qualification Test Plan before being released to production. The Qualification Test Plan for the MPU-60X0 followed the JESD471 Standards, "Stress-Test-Driven Qualification of Integrated Circuits," with the individual tests described below.

12.2 Qualification Test Plan

Accelerated Life Tests

TEST	Method/Condition	Lot Quantity	Sample / Lot	Acc / Reject Criteria
(HTOL/LFR) High Temperature Operating Life	JEDEC JESD22-A108D, Dynamic, 3.63V biased, $T_j > 125^{\circ}\text{C}$ [read-points 168, 500, 1000 hours]	3	77	(0/1)
(HAST) Highly Accelerated Stress Test ⁽¹⁾	JEDEC JESD22-A118A Condition A, 130°C , 85%RH, 33.3 psia. unbiased, [read-point 96 hours]	3	77	(0/1)
(HTS) High Temperature Storage Life	JEDEC JESD22-A103D, Cond. A, 125°C Non-Bias Bake [read-points 168, 500, 1000 hours]	3	77	(0/1)

Device Component Level Tests

TEST	Method/Condition	Lot Quantity	Sample / Lot	Acc / Reject Criteria
(ESD-HBM) ESD-Human Body Model	JEDEC JS-001-2012, (2KV)	1	3	(0/1)
(ESD-MM) ESD-Machine Model	JEDEC JESD22-A115C, (250V)	1	3	(0/1)
(LU) Latch Up	JEDEC JESD-78D Class II (2), 125°C ; $\pm 100\text{mA}$	1	6	(0/1)
(MS) Mechanical Shock	JEDEC JESD22-B104C, Mil-Std-883, Method 2002.5, Cond. E, $10,000g's$, 0.2ms, $\pm X, Y, Z - 6$ directions, 5 times/direction	3	5	(0/1)
(VIB) Vibration	JEDEC JESD22-B103B, Variable Frequency (random), Cond. B, 5-500Hz, X, Y, Z - 4 times/direction	3	5	(0/1)
(TC) Temperature Cycling ⁽¹⁾	JEDEC JESD22-A104D Condition G [-40°C to $+125^{\circ}\text{C}$], Soak Mode 2 [5'], 1000 cycles	3	77	(0/1)

Board Level Tests

TEST	Method/Condition	Lot Quantity	Sample / Lot	Acc / Reject Criteria
(BMS) Board Mechanical Shock	JEDEC JESD22-B104C, Mil-Std-883, Method 2002.5, Cond. E, $10000g's$, 0.2ms, $+X, Y, Z - 6$ directions, 5 times/direction	1	5	(0/1)
(BTC) Board Temperature Cycling ⁽¹⁾	JEDEC JESD22-A104D Condition G [-40°C to $+125^{\circ}\text{C}$], Soak mode 2 [5'], 1000 cycles	1	40	(0/1)

(1) Tests are preceded by MSL3 Preconditioning in accordance with JEDEC JESD22-A113F



13 Environmental Compliance

The MPU-6000/MPU-6050 is RoHS and Green compliant.

The MPU-6000/MPU-6050 is in full environmental compliance as evidenced in report HS-MPU-6000, Materials Declaration Data Sheet.

Environmental Declaration Disclaimer:

InvenSense believes this environmental information to be correct but cannot guarantee accuracy or completeness. Conformity documents for the above component constitutes are on file. InvenSense subcontracts manufacturing and the information contained herein is based on data received from vendors and suppliers, which has not been validated by InvenSense.

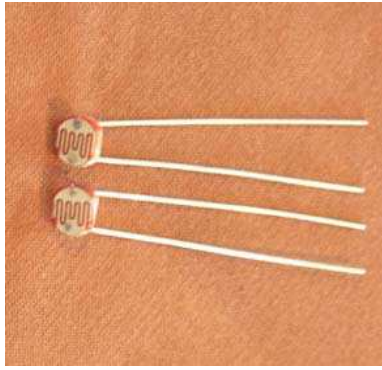
This information furnished by InvenSense is believed to be accurate and reliable. However, no responsibility is assumed by InvenSense for its use, or for any infringements of patents or other rights of third parties that may result from its use. Specifications are subject to change without notice. InvenSense reserves the right to make changes to this product, including its circuits and software, in order to improve its design and/or performance, without prior notice. InvenSense makes no warranties, neither expressed nor implied, regarding the information and specifications contained in this document. InvenSense assumes no responsibility for any claims or damages arising from information contained in this document, or from the use of products and services detailed therein. This includes, but is not limited to, claims or damages based on the infringement of patents, copyrights, mask work and/or other intellectual property rights.

Certain intellectual property owned by InvenSense and described in this document is patent protected. No license is granted by implication or otherwise under any patent or patent rights of InvenSense. This publication supersedes and replaces all information previously supplied. Trademarks that are registered trademarks are the property of their respective companies. InvenSense sensors should not be used or sold in the development, storage, production or utilization of any conventional or mass-destructive weapons or for any other weapons or life threatening applications, as well as in any other life critical applications such as medical equipment, transportation, aerospace and nuclear instruments, undersea equipment, power plant equipment, disaster prevention and crime prevention equipment.

InvenSense® is a registered trademark of InvenSense, Inc. MPU™, MPU-6000™, MPU-6050™, MPU-60X0™, Digital Motion Processor™, DMP™, Motion Processing Unit™, MotionFusion™, MotionInterface™, MotionTracking™, and MotionApps™ are trademarks of InvenSense, Inc.



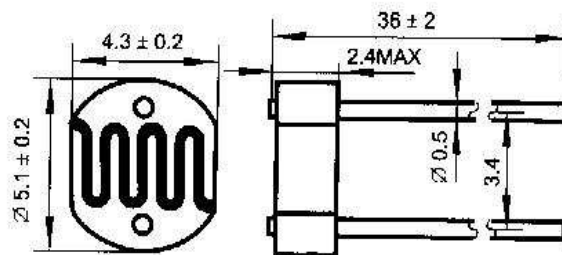
Anexo 3
Datasheet
GL5528



- ▲ Epoxy encapsulated
- ▲ Quick response
- ▲ Small size
- ▲ High sensitivity
- ▲ Reliable performance
- ▲ Good characteristic of spectrum

Light Resistance at 10Lux (at 25°C)	8~20KΩ
Dark Resistance at 0 Lux	1.0MΩ(min)
Gamma value at 100-10Lux	0.7
Power Dissipation(at 25°C)	100mW
Max Voltage (at 25°C)	150V
Spectral Response peak (at 25°C)	540nm
Ambient Temperature Range:	- 30~+70°C

Outline

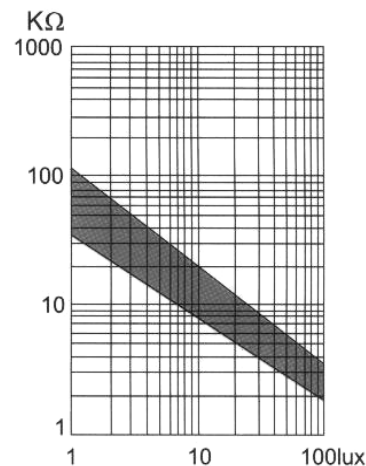


Measuring Conditions

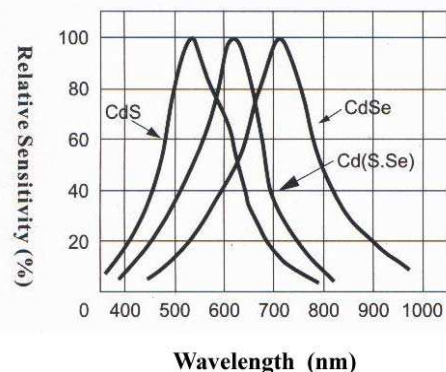
1. Light Resistance:
measured at 10 lux with standard light A (2854k color temperature) and 2h pre-illumination at 400-600 lux prior to testing.
2. Dark Resistance:
measured 10 seconds after pulsed 10 lux.
3. Gamma Characteristic:
between 10 lux and 100 lux and given by

$$T = \frac{\log(R_{10}/R_{100})}{\log(100/10)} = \log(R_{10}/R_{100})$$
 R10, R100 cell resistance at 10 lux and 100 lux.
The error of T is +0.1.
4. Pmax:
Max. power dissipation at ambient temperature of 25°C.
5. Vmax:
Max. voltage in darkness that may be applied to the cell continuously.

Illuminance Vs. Photo Resistance



Spectral Response



LIDA OPTICAL&ELECTRONIC CO., LTD.

254, Zhong zhou road, Nanyang, Henan, P.R.C

TEL: +86-377-6313 0034

FAX: +86-377-6315 2372

E-mail: sale@nylida.com

[Http://www.nylida.com](http://www.nylida.com)

Anexo 4
Datasheet
L293D



PUSH-PULL FOUR CHANNEL DRIVER WITH DIODES

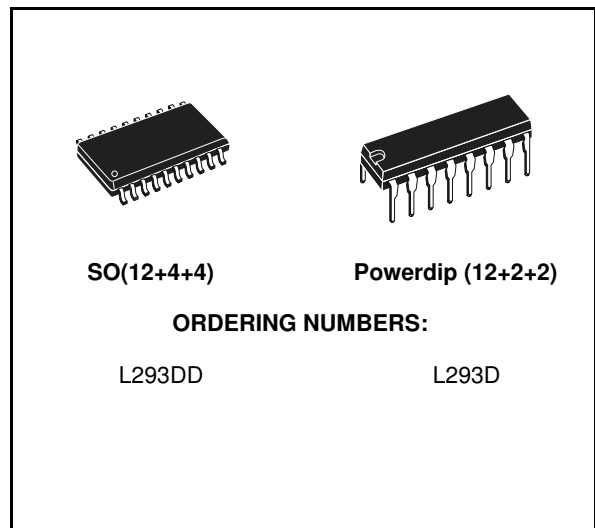
- 600mA OUTPUT CURRENT CAPABILITY PER CHANNEL
- 1.2A PEAK OUTPUT CURRENT (non repetitive) PER CHANNEL
- ENABLE FACILITY
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)
- INTERNAL CLAMP DIODES

DESCRIPTION

The Device is a monolithic integrated high voltage, high current four channel driver designed to accept standard DTL or TTL logic levels and drive inductive loads (such as relays solenoids, DC and stepping motors) and switching power transistors.

To simplify use as two bridges each pair of channels is equipped with an enable input. A separate supply input is provided for the logic, allowing operation at a lower voltage and internal clamp diodes are included.

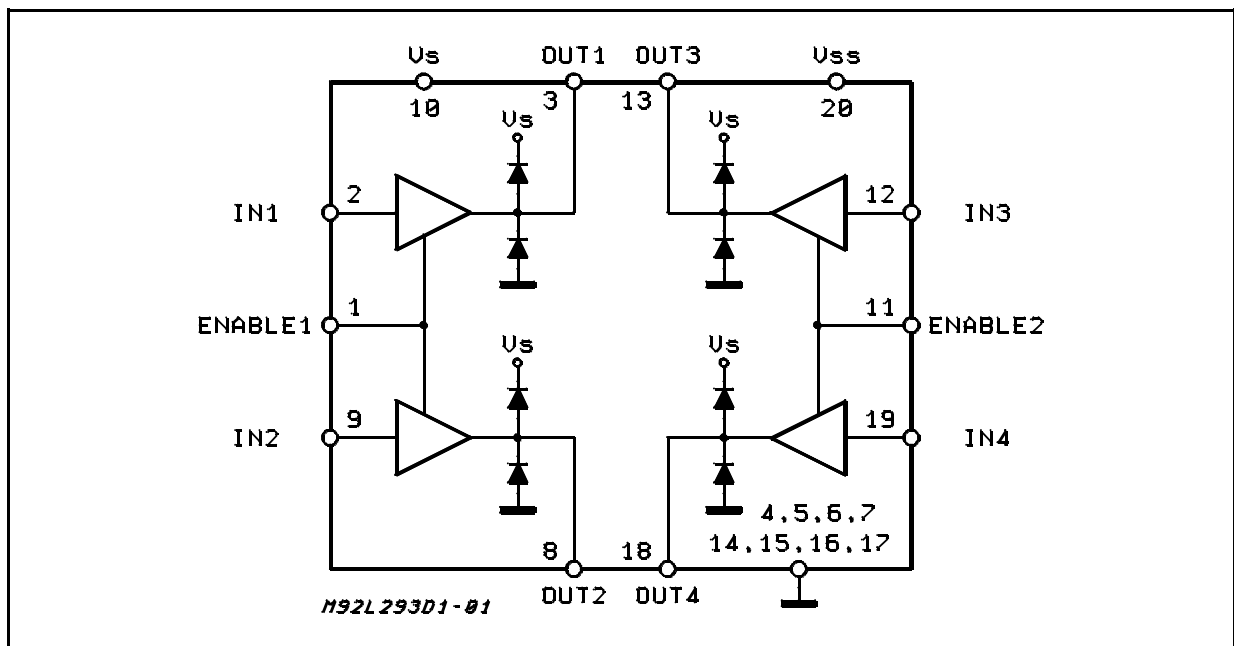
This device is suitable for use in switching applications at frequencies up to 5 kHz.



The L293D is assembled in a 16 lead plastic package which has 4 center pins connected together and used for heatsinking

The L293DD is assembled in a 20 lead surface mount which has 8 center pins connected together and used for heatsinking.

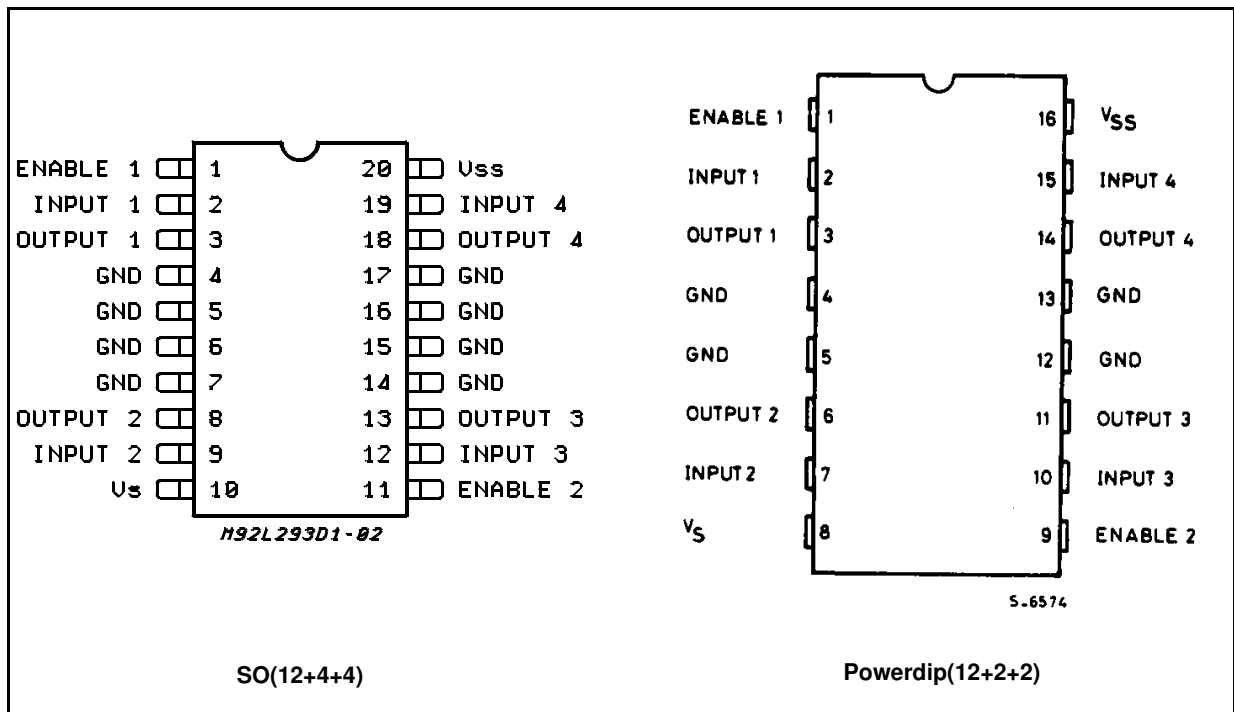
BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V _S	Supply Voltage	36	V
V _{SS}	Logic Supply Voltage	36	V
V _i	Input Voltage	7	V
V _{en}	Enable Voltage	7	V
I _o	Peak Output Current (100 μs non repetitive)	1.2	A
P _{tot}	Total Power Dissipation at T _{pins} = 90 °C	4	W
T _{stg} , T _j	Storage and Junction Temperature	- 40 to 150	°C

PIN CONNECTIONS (Top view)



THERMAL DATA

Symbol	Description	DIP	SO	Unit
R _{th j-pins}	Thermal Resistance Junction-pins	max.	14	°C/W
R _{th j-amb}	Thermal Resistance junction-ambient	max.	50 (*)	°C/W
R _{th j-case}	Thermal Resistance Junction-case	max.	-	

(*) With 6sq. cm on board heatsink.

ELECTRICAL CHARACTERISTICS (for each channel, $V_S = 24\text{ V}$, $V_{SS} = 5\text{ V}$, $T_{amb} = 25\text{ }^\circ\text{C}$, unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_S	Supply Voltage (pin 10)		V_{SS}		36	V
V_{SS}	Logic Supply Voltage (pin 20)		4.5		36	V
I_S	Total Quiescent Supply Current (pin 10)	$V_i = L$; $I_O = 0$; $V_{en} = H$		2	6	mA
		$V_i = H$; $I_O = 0$; $V_{en} = H$		16	24	mA
		$V_{en} = L$			4	mA
I_{SS}	Total Quiescent Logic Supply Current (pin 20)	$V_i = L$; $I_O = 0$; $V_{en} = H$		44	60	mA
		$V_i = H$; $I_O = 0$; $V_{en} = H$		16	22	mA
		$V_{en} = L$		16	24	mA
V_{IL}	Input Low Voltage (pin 2, 9, 12, 19)		-0.3		1.5	V
V_{IH}	Input High Voltage (pin 2, 9, 12, 19)	$V_{SS} \leq 7\text{ V}$	2.3		V_{SS}	V
		$V_{SS} > 7\text{ V}$	2.3		7	V
I_{IL}	Low Voltage Input Current (pin 2, 9, 12, 19)	$V_{IL} = 1.5\text{ V}$			-10	μA
I_{IH}	High Voltage Input Current (pin 2, 9, 12, 19)	$2.3\text{ V} \leq V_{IH} \leq V_{SS} - 0.6\text{ V}$		30	100	μA
V_{enL}	Enable Low Voltage (pin 1, 11)		-0.3		1.5	V
V_{enH}	Enable High Voltage (pin 1, 11)	$V_{SS} \leq 7\text{ V}$	2.3		V_{SS}	V
		$V_{SS} > 7\text{ V}$	2.3		7	V
I_{enL}	Low Voltage Enable Current (pin 1, 11)	$V_{enL} = 1.5\text{ V}$		-30	-100	μA
I_{enH}	High Voltage Enable Current (pin 1, 11)	$2.3\text{ V} \leq V_{enH} \leq V_{SS} - 0.6\text{ V}$			± 10	μA
$V_{CE(sat)H}$	Source Output Saturation Voltage (pins 3, 8, 13, 18)	$I_O = -0.6\text{ A}$		1.4	1.8	V
$V_{CE(sat)L}$	Sink Output Saturation Voltage (pins 3, 8, 13, 18)	$I_O = +0.6\text{ A}$		1.2	1.8	V
V_F	Clamp Diode Forward Voltage	$I_O = 600\text{ nA}$		1.3		V
t_r	Rise Time (*)	0.1 to 0.9 V_O		250		ns
t_f	Fall Time (*)	0.9 to 0.1 V_O		250		ns
t_{on}	Turn-on Delay (*)	0.5 V_i to 0.5 V_O		750		ns
t_{off}	Turn-off Delay (*)	0.5 V_i to 0.5 V_O		200		ns

(*) See fig. 1.

L293D - L293DD

TRUTH TABLE (one channel)

Input	Enable (*)	Output
H	H	H
L	H	L
H	L	Z
L	L	Z

Z = High output impedance

(*) Relative to the considered channel

Figure 1: Switching Times

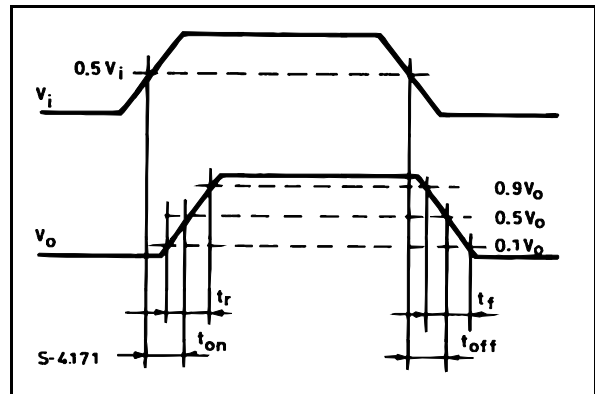
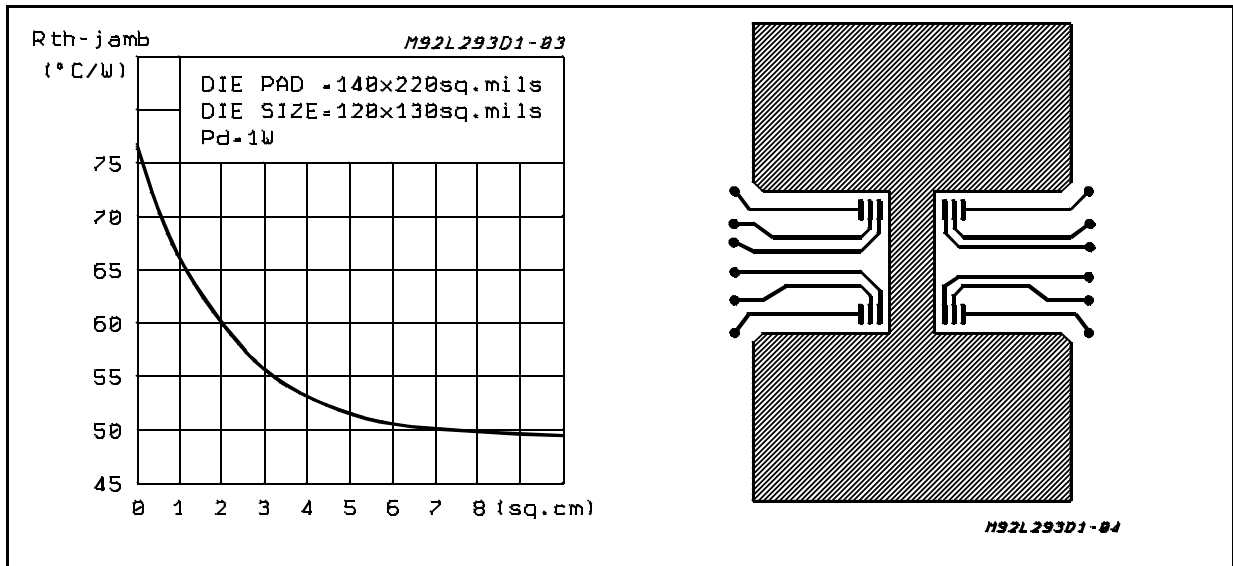
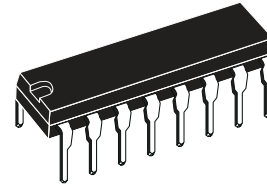


Figure 2: Junction to ambient thermal resistance vs. area on board heatsink (SO12+4+4 package)

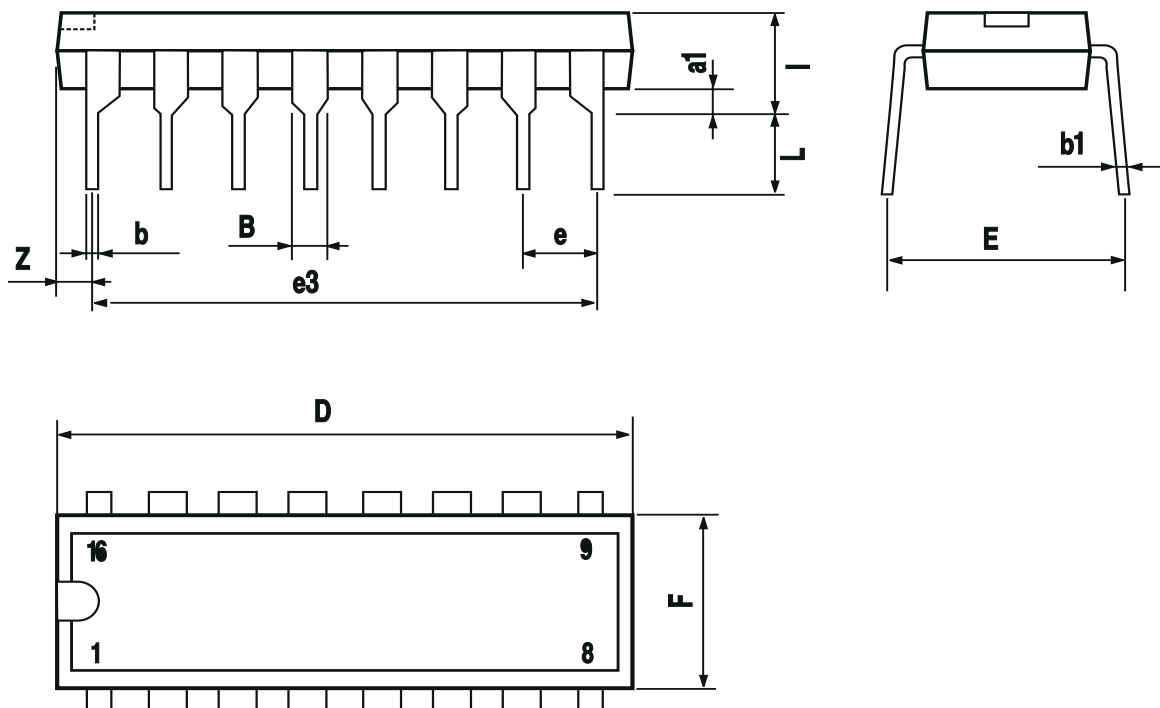


DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
a1	0.51			0.020		
B	0.85		1.40	0.033		0.055
b		0.50			0.020	
b1	0.38		0.50	0.015		0.020
D			20.0			0.787
E		8.80			0.346	
e		2.54			0.100	
e3		17.78			0.700	
F			7.10			0.280
I			5.10			0.201
L		3.30			0.130	
Z			1.27			0.050

OUTLINE AND MECHANICAL DATA



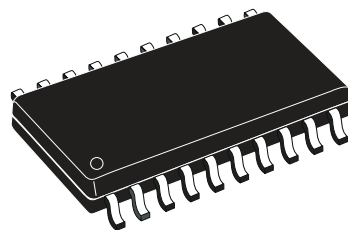
Powerdip 16



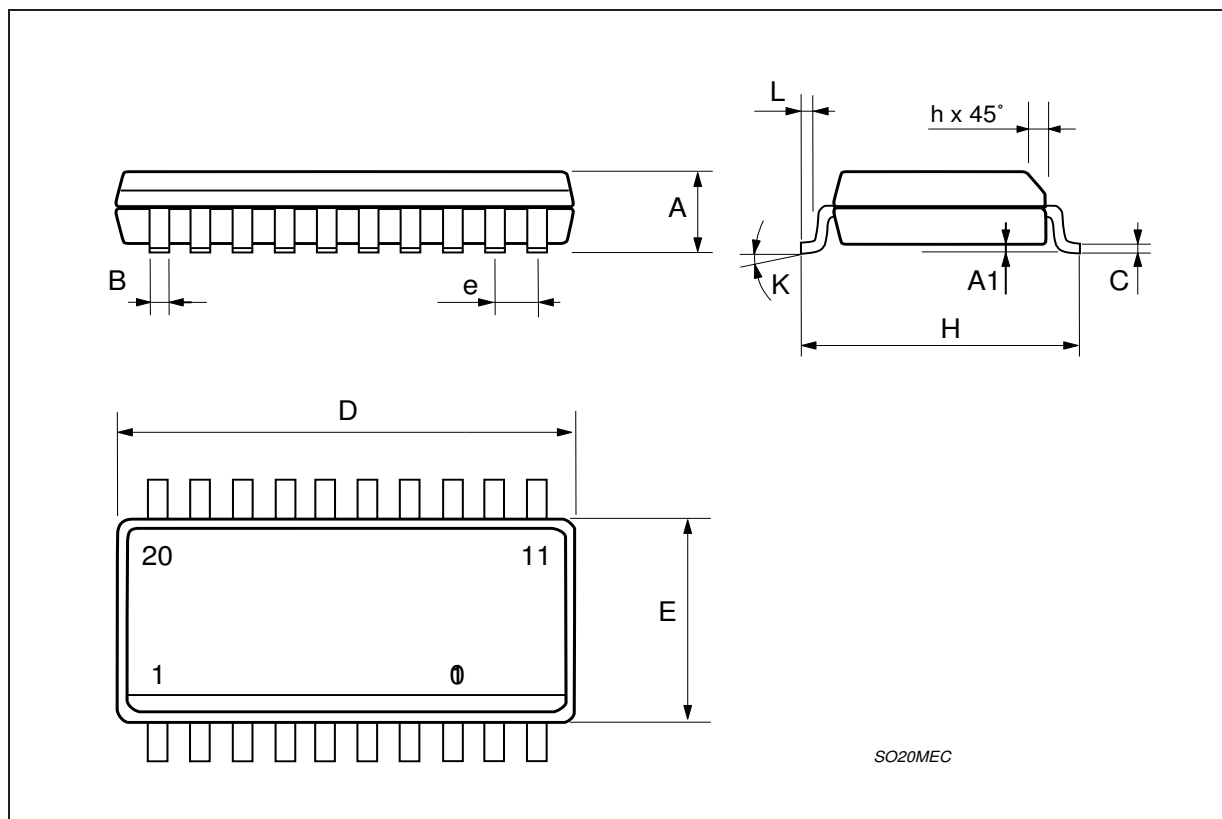
L293D - L293DD

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A	2.35		2.65	0.093		0.104
A1	0.1		0.3	0.004		0.012
B	0.33		0.51	0.013		0.020
C	0.23		0.32	0.009		0.013
D	12.6		13	0.496		0.512
E	7.4		7.6	0.291		0.299
e		1.27			0.050	
H	10		10.65	0.394		0.419
h	0.25		0.75	0.010		0.030
L	0.4		1.27	0.016		0.050
K	0° (min.)8° (max.)					

OUTLINE AND MECHANICAL DATA



SO20



SO20MEC

Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specification mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

The ST logo is a registered trademark of STMicroelectronics

© 2003 STMicroelectronics – Printed in Italy – All Rights Reserved

STMicroelectronics GROUP OF COMPANIES

Australia - Brazil - Canada - China - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States.

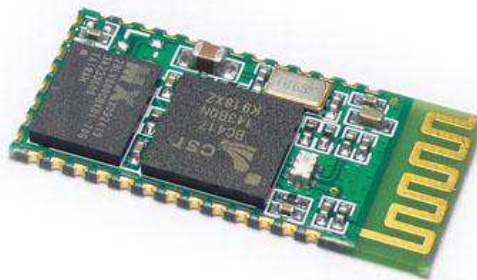
<http://www.st.com>

Anexo 5
Datasheet
HC-05

HC-05

-Bluetooth to Serial Port Module

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.

Specifications

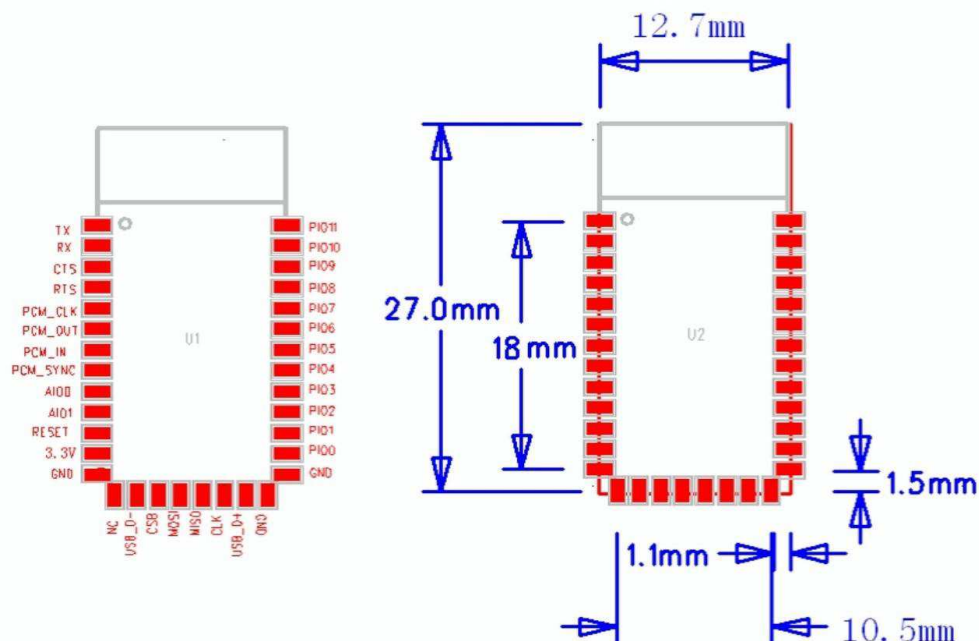
Hardware features

- Typical -80dBm sensitivity
- Up to +4dBm RF transmit power
- Low Power 1.8V Operation ,1.8 to 3.6V I/O
- PIO control
- UART interface with programmable baud rate
- With integrated antenna
- With edge connector

Software features

- Default Baud rate: 38400, Data bits:8, Stop bit:1,Parity:No parity, Data control: has. Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.
- Given a rising pulse in PIO0, device will be disconnected.
- Status instruction port PIO1: low-disconnected, high-connected;
- PIO10 and PIO11 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:"0000" as default
- Auto-reconnect in 30 min when disconnected as a result of beyond the range of connection.

Hardware



PIN Name	PIN #	Pad type	Description	Note
GND	13 21 22	VSS	Ground pot	
3.3 VCC	12	3.3V	Integrated 3.3V (+) supply with On-chip linear regulator output within 3.15-3.3V	
AIO0	9	Bi-Directional	Programmable input/output line	
AIO1	10	Bi-Directional	Programmable input/output line	
PIO0	23	Bi-Directional RX EN	Programmable input/output line, control output for LNA(if fitted)	
PIO1	24	Bi-Directional TX EN	Programmable input/output line, control output for PA(if fitted)	

PIO2	25	Bi-Directional	Programmable input/output line	
PIO3	26	Bi-Directional	Programmable input/output line	
PIO4	27	Bi-Directional	Programmable input/output line	
PIO5	28	Bi-Directional	Programmable input/output line	
PIO6	29	Bi-Directional	Programmable input/output line	
PIO7	30	Bi-Directional	Programmable input/output line	
PIO8	31	Bi-Directional	Programmable input/output line	
PIO9	32	Bi-Directional	Programmable input/output line	
PIO10	33	Bi-Directional	Programmable input/output line	
PIO11	34	Bi-Directional	Programmable input/output line	

RESETB	11	CMOS input with weak internal pull-up	Reset if low.input debounced so must be low for >5MS to cause a reset	
UART_RTS	4	CMOS output, tri-stable with weak internal pull-up	UART request to send, active low	
UART_CTS	3	CMOS input with weak internal pull-down	UART clear to send, active low	
UART_RX	2	CMOS input with weak internal pull-down	UART Data input	
UART_TX	1	CMOS output, Tri-stable with weak internal pull-up	UART Data output	
SPI_MOSI	17	CMOS input with weak internal pull-down	Serial peripheral interface data input	

SPI_CSB	16	CMOS input with weak internal pull-up	Chip select for serial peripheral interface, active low	
SPI_CLK	19	CMOS input with weak internal pull-down	Serial peripheral interface clock	
SPI_MISO	18	CMOS input with weak internal pull-down	Serial peripheral interface data Output	
USB_-	15	Bi-Directional		

USB_+	20	Bi-Directional		
NC	14			
PCM_CLK	5	Bi-Directional	Synchronous PCM data clock	
PCM_OUT	6	CMOS output	Synchronous PCM data output	
PCM_IN	7	CMOS Input	Synchronous PCM data input	
PCM_SYNC	8	Bi-Directional	Synchronous PCM data strobe	

AT command Default:

How to set the mode to server (master):

1. Connect PIO11 to high level.
2. Power on, module into command state.
3. Using baud rate 38400, sent the "AT+ROLE=1\r\n" to module, with "OK\r\n" means setting successes.
4. Connect the PIO11 to low level, repower the module, the module work as server (master).

AT commands: (all end with \r\n)

1. Test command:

Command	Respond	Parameter
AT	OK	-

2. Reset

Command	Respond	Parameter
AT+RESET	OK	-

3. Get firmware version

Command	Respond	Parameter
AT+VERSION?	+VERSION:<Param> OK	Param : firmware version

Example:

```
AT+VERSION?\r\n
+VERSION:2.0-20100601
OK
```

4. Restore default

Command	Respond	Parameter
AT+ORGL	OK	-

Default state:

Slave mode, pin code :1234, device name: H-C-2010-06-01 ,Baud 38400bits/s.

5. Get module address

Command	Respond	Parameter
AT+ADDR?	+ADDR:<Param> OK	Param: address of Bluetooth module

Bluetooth address: NAP: UAP : LAP

Example:

AT+ADDR?\r\n

+ADDR:1234:56:abcdef

OK

6. Set/Check module name:

Command	Respond	Parameter
AT+NAME=<Param>	OK	Param: Bluetooth module name (Default :HC-05)
AT+NAME?	+NAME:<Param> OK (/FAIL)	

Example:

AT+NAME=HC-05\r\n set the module name to "HC-05"

OK

AT+NAME=ITeadStudio\r\n

OK

AT+NAME?\r\n

+NAME: ITeadStudio

OK

7. Get the Bluetooth device name:

Command	Respond	Parameter
AT+RNAME?<Param1>	1. +NAME:<Param2> OK 2. FAIL	Param1,Param 2 : the address of Bluetooth device

Example: (Device address 00:02:72:od:22:24, name: ITead)

AT+RNAME? 0002, 72, od2224\r\n

+RNAME:ITead

OK

8. Set/Check module mode:

Command	Respond	Parameter
AT+ROLE=<Param>	OK	Param:
AT+ROLE?	+ROLE:<Param>	0- Slave

	OK	1-Master 2-Slave-Loop
--	----	--------------------------

9. Set/Check device class

Command	Respond	Parameter
AT+CLASS=<Param>	OK	Param: Device Class
AT+ CLASS?	1. +CLASS:<Param> OK 2. FAIL	

10. Set/Check GIAC (General Inquire Access Code)

Command	Respond	Parameter
AT+IAC=<Param>	1.OK 2. FAIL	Param: GIAC (Default : 9e8b33)
AT+IAC	+IAC:<Param> OK	

Example:

```
AT+IAC=9e8b3f\r\n
OK
AT+IAC?\r\n
+IAC: 9e8b3f
OK
```

11. Set/Check -- Query access patterns

Command	Respond	Parameter
AT+INQM=<Param>,<Param2>,<Param3>	1.OK 2. FAIL	Param: 0— inquiry_mode_standard 1— inquiry_mode_rssi Param2: Maximum number of Bluetooth devices to respond to Param3: Timeout (1-48 : 1.28s to 61.44s)
AT+ INQM?	+INQM : <Param>,<Param2>,<Param3> OK	

Example:

```
AT+INQM=1,9,48\r\n
OK
AT+INQM\r\n
+INQM:1, 9, 48
OK
```

12. Set/Check PIN code:

Command	Respond	Parameter
AT+PSWD=<Param>	OK	Param: PIN code (Default 1234)
AT+ PSWD?	+ PSWD : <Param> OK	

13. Set/Check serial parameter:

Command	Respond	Parameter
AT+UART=<Param>,<Param2>,<Param3>	OK	Param1: Baud Param2: Stop bit Param3: Parity
AT+ UART?	+UART=<Param>,<Param2>,<Param3> OK	

Example:

```
AT+UART=115200, 1,2,\r\n
OK
AT+UART?
+UART:115200,1,2
OK
```

14. Set/Check connect mode:

Command	Respond	Parameter
AT+CMODE=<Param>	OK	Param: 0 - connect fixed address 1 - connect any address 2 - slave-Loop
AT+ CMODE?	+ CMODE:<Param> OK	

15. Set/Check fixed address:

Command	Respond	Parameter
AT+BIND=<Param>	OK	Param: Fixed address (Default 00:00:00:00:00:00)
AT+ BIND?	+ BIND:<Param> OK	

Example:

```
AT+BIND=1234, 56, abcdef\r\n
OK
AT+BIND?\r\n
+BIND:1234:56:abcdef
OK
```

16. Set/Check LED I/O

Command	Respond	Parameter
AT+POLAR=<Param1,<Param2>	OK	Param1:
AT+ POLAR?	+ POLAR=<Param1>,<Param2> OK	0- PIO8 low drive LED 1- PIO8 high drive LED

		Param2: 0- PIO9 low drive LED 1- PIO9 high drive LED
--	--	--

17. Set PIO output

Command	Respond	Parameter
AT+PIO=<Param1>,<Param2>	OK	Param1: PIO number Param2: PIO level 0- low 1- high

Example:

1. PIO10 output high level

```
AT+PIO=10, 1\r\n
```

```
OK
```

18. Set/Check – scan parameter

Command	Respond	Parameter
AT+IPSCAN=<Param1>,<Param2>,<Param3>,<Param4>	OK	Param1: Query time interval
AT+IPSCAN?	+IPSCAN:<Param1>,<Param2>,<Param3>,<Param4> OK	Param2: Query duration Param3: Paging interval Param4: Call duration

Example:

```
AT+IPSCAN =1234,500,1200,250\r\n
```

```
OK
```

```
AT+IPSCAN?
```

```
+IPSCAN:1234,500,1200,250
```

19. Set/Check – SHIFF parameter

Command	Respond	Parameter
AT+SNIFF=<Param1>,<Param2>,<Param3>,<Param4>	OK	Param1: Max time Param2: Min time
AT+ SNIFF?	+SNIFF:<Param1>,<Param2>,<Param3>,<Param4> OK	Param3: Retry time Param4: Time out

20. Set/Check security mode

Command	Respond	Parameter
AT+SENM=<Param1>,<Param2>	1. OK 2. FAIL	Param1: 0—sec_mode0+off
AT+ SENM?	+ SENM:<Param1>,<Param2>	1—sec_mode1+non_se

	OK	cure 2—sec_mode2_service 3—sec_mode3_link 4—sec_mode_unknow n Param2: 0—hci_enc_mode_off 1—hci_enc_mode_pt_t o_pt 2—hci_enc_mode_pt_t o_pt_and_bcast
--	----	--

21. Delete Authenticated Device

Command	Respond	Parameter
AT+PMSAD=<Param>	OK	Param: Authenticated Device Address

Example:

AT+PMSAD =1234,56,abcdef\r\n

OK

22. Delete All Authenticated Device

Command	Respond	Parameter
AT+ RMAAD	OK	-

23. Search Authenticated Device

Command	Respond	Parameter
AT+FSAD=<Param>	1. OK 2. FAIL	Param: Device address

24. Get Authenticated Device Count

Command	Respond	Parameter
AT+ADCN?	+ADCN: <Param> OK	Param: Device Count

25. Most Recently Used Authenticated Device

Command	Respond	Parameter
AT+MRAD?	+ MRAD: <Param> OK	Param: Recently Authenticated Device Address

26. Get the module working state

Command	Respond	Parameter
---------	---------	-----------

AT+ STATE?	+ STATE: <Param> OK	Param: "INITIALIZED" "READY" "PAIRABLE" "PAIRED" "INQUIRING" "CONNECTING" "CONNECTED" "DISCONNECTED" "NUKNOW"
------------	------------------------	--

27. Initialize the SPP profile lib

Command	Respond	Parameter
AT+INIT	1. OK 2. FAIL	-

28. Inquiry Bluetooth Device

Command	Respond	Parameter
AT+INQ	+INQ: <Param1> , <Param2> , <Param3> OK	Param1: Address Param2: Device Class Param3 : RSSI Signal strength

Example:

```

AT+INIT\r\n
OK
AT+IAC=9e8b33\r\n
OK
AT+CLASS=0\r\n
AT+INQM=1,9,48\r\n
At+INQ\r\n
+INQ:2:72:D2224,3E0104,FFBC
+INQ:1234:56:0,1F1F,FFC1
+INQ:1234:56:0,1F1F,FFC0
+INQ:1234:56:0,1F1F,FFC1
+INQ:2:72:D2224,3F0104,FFAD
+INQ:1234:56:0,1F1F,FFBE
+INQ:1234:56:0,1F1F,FFC2
+INQ:1234:56:0,1F1F,FFBE
+INQ:2:72:D2224,3F0104,FFBC
OK
  
```

28. Cancel Inquiring Bluetooth Device

Command	Respond	Parameter
AT+ INQC	OK	-

29. Equipment Matching

Command	Respond	Parameter
AT+PAIR=<Param1>,<Param2>	1. OK 2. FAIL	Param1: Device Address Param2: Time out

30. Connect Device

Command	Respond	Parameter
AT+LINK=<Param>	1. OK 2. FAIL	Param: Device Address

Example:

AT+FSAD=1234,56,abcdef\r\n

OK

AT+LINK=1234,56,abcdef\r\n

OK

31. Disconnect

Command	Respond	Parameter
AT+DISC	1. +DISC:SUCCESS OK 2. +DISC:LINK_LOSS OK 3. +DISC:NO_SLC OK 4. +DISC:TIMEOUT OK 5. +DISC:ERROR OK	Param: Device Address

32. Energy-saving mode

Command	Respond	Parameter
AT+ENSNIFF=<Param>	OK	Param: Device Address

33. Exerts Energy-saving mode

Command	Respond	Parameter
AT+ EXSNIFF =<Param>	OK	Param: Device Address

Revision History

Rev.	Description	Release date
v1.0	Initial version	7/18/2010