“*SALIENT OBJECT FOLLOWING SELF BALANCING ROBOT*”

A Dissertation Submitted for the Partial Fulfilment for the Degree of

Master of Technology
in
Signal Processing and Digital Design

Submitted By

Somya Gupta
Roll No. 2k12/SPD/21

Under the guidance of

Dr. S. Indu
Department of Electronics and Communication Engineering

Delhi Technological University
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Bawana Road
New Delhi -110042

CERTIFICATE

It is certified that **Ms. Somya Gupta** Roll No. **2k12/SPD/21**, student of **M.Tech. Signal Processing and Digital Design**, Department of Electronic and Communication Engineering, Delhi Technological University, has submitted the dissertation entitled **"Salient Object Following Self Balancing Robot"** under my guidance towards partial fulfilment of the requirements for the award of the degree of Master of Technology (Signal Processing and Digital Design).

The dissertation is a bonafide work record of project work carried out by her under my guidance and supervision. Her work is found to be outstanding and her discipline impeccable during the course of the project.

I wish her success in all her endeavours.

Dr. S. Indu
Associate Professor
Department of Electronics and Communication Engineering
Delhi Technological University

ACKNOWLEDGMENT

The completion of any project brings with it a sense of satisfaction, but it is never complete without thanking those people who made it possible and whom constant support has crowned my efforts with success.

One cannot even imagine the power of the force that guides us all and neither can we succeed without acknowledging it. My deepest gratitude to my grandfather, **Late Sh. Lok Nath Gupta,** who has been my god and my love**,** for holding my hands and guiding me throughout my life.

I would like to thank my beloved parents, **Sh. Sanjeev Gupta** and **Smt. Rekha Gupta**, who always give me strong inspirations, moral supports, and helpful suggestions. Without them, my study career would never have begun. It is only because of them, my life has always been full of abundant blessing.

I would like to devote my gratitude and thanks to my guide **Dr. S. Indu**, **Associate Professor, Department of Electronics and Communication Engineering, Delhi Technological University, Delhi** for her valuable guidance, constant encouragement and helpful discussions throughout the course of this work. Obviously, the progress I had now will be uncertain without her guidance.

I would also like to thank **Prof. Rajiv Kapoor**, **H.O.D. Electronics and Communication Engineering Department, Delhi Technological University, Delhi** for providing me better facilities and constant encouragement.

At last but not least I would like to express my vote of thanks to my brother **Mr. Rishi Gupta** for his criticism, support and encouragement.

SOMYA GUPTA
2k12/SPD/21

ABSTRACT

Robotic mobility technology have evolved exponentially over the past few years and not only do they find use in military and security sector, they are also fast gaining popularity in industry and consumer products. A number of techniques have been proposed to increase robotic mobility in dynamic environments. One such widely used technique based on the inverted pendulum model is used to provide greater mobility to a robotic platform.

The instability of inverted pendulum systems has always been an ideal test bed for control theory experimentation. The presented document will display the procedures involved in balancing an unstable robotic platform with the aim to design a complete discrete digital control system that will provide the stability required by the system for optimum performance.

The platform will be an ideal test bed for the implementations of both PID digital control and Kalman filter algorithms. Both these algorithms will provide the imperative control for the system. Therefore the presented project will examine the performance of both PID digital control and Kalman filter algorithms. PID control algorithm is used to offer system stability whereas Kalman filter, an estimator, provides fused data of the sensors (Accelerometer and gyroscope). The digital filter provides a more reliable sensor data used to calculate the tilt angle of the robot. To collect performance results for both the PID controller and Kalman filter, Test software was written.

The control system performance directly depends on Kalman filter and PID controller input parameters and the outcome clearly shows how the adjustable parameters on the control system directly impacted the overall system performance. The results also indicate the performance and the need of the Kalman filter to remove sensor noise.The nearly reliable sensor data increases PID controller performance to drive the robotic platform to vertical equilibrium.

Further to this, the raw noisy sensor data was compared against the accumulated results for the Kalman filter. The plots for this comparison are shown in the Kalman filter results section. PID controller output response data was also collected and plotted. The PID output response results were used in the controller tuning process.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

Robotic mobility technologies have gained momentum in the last few years. Numerous techniques have suggested the increase of robotic mobility in dynamic environments. Robotic platform based on the inverted pendulum model is one such technique to provide greater maneuverability. The characteristics of these robots is that it can balance on its two wheels and spin on a spot. This added maneuverability allows effortless navigation on various terrains, sharp corners and go over small steps. A motorised wheelchair employing this technology will provide greater motility and will provide access to places which healthy people take for granted. Also Segway PT, battery-powered electric vehicle, invented by Dean Kamen, allows people to travel short distances in factories or malls without polluting the area.

Another popular research area in today’s time is the development of service robots to aid human in their day to day work. For example, the robots can be used for security and surveillance, hospital assistance, assistance for elderly, etc.

The first aim is to fabricate a complete discrete digital system that will provide help to balance an unstable robotic platform. A digital control algorithm and Kalman filter for sensor fusion need to be examined. Proportional-Integral-Derivative controller is used as the digital controller algorithm. As the technology is advancing microcontrollers have become cheaper, faster and more reliable. Arduino Uno 328 functions as the digital controller for the robot and is used to program the digital control system and the filter using a high level programming language C. PID control algorithm is used to offer system stability whereas Kalman filter, an estimator, provides fused data of the sensors (Accelerometer and gyroscope). The digital filter provides a more reliable sensor data used to calculate the tilt angle of the robot. Arduino Uno 328 functions as the digital controller for the robot. The maneuverability for the autonomous self balancing robot will be attained though by two dc geared motors.

The second objective of the project is to make the robot follow a salient object. For this purpose a high definition IR sensor is used. The object in the scene is detected and tracked using the sensor with the aid of the microcontroller.

1.1 Purpose of the Project

The purpose of the project is to develop the understanding of digital control algorithms, techniques for detecting and tracking an object and how these can be fused together in a real time environment. The goal of the project is to implement a Kalman filter, PID controller and Ultrasonic distance sensor (serial based) on Arduino based microcontroller.

1.2 Limitations of the project

Few of the major constraints of the project are listed.

1. Sampling time that will be used to execute the control algorithms.
2. The noises present in the feedback sensors effect the control algorithm. Noisy sensor data give inaccurate results that will lead disrupt the performance of the PID controller.
3. The computation time needed by the microcontroller to run the control algorithms. If there is a delay, the tilt angle cannot be judged in time and will lead to falling of the robot.

1.3 Organization of the Dissertation

Rest of work is organized as follows:

Chapter 2: Literature Survey

This section provides literature review for the models designed based on inverted pendulum model, Kalman filter and PID controller. It also contains information regarding various other techniques used for the implementation of two-wheeled self balancing robot.

Chapter 3: The Balancing Robot System

This chapter talks about the hardware components used in designing the self balancing robot.

Chapter 4: Kalman Filter

It explains the process and mathematics involved in Kalman filter. It also briefly elucidates the basic algorithm of the Discrete Kalman filter.

Chapter 5: The PID Algorithm

It describes proportional algorithm, proportional integral algorithm and proportional derivative algorithm in brief. Also, it explains the various topologies of PID controller and tuning methods used in PID controller.

Chapter 6: The Proposed Method

This chapter tells how the project was executed.

Chapter 7: Results and Discussions

In this, the results coming from inertial sensors, Kalman filter and PID and depicted in a tabular form. In addition to a comparison is drawn between the angles obtained by the inertial sensors and Kalman filter.

References: This section gives the reference details of the thesis.

Appendix

CHAPTER 2
LITERATURE SURVEY

2. LITERATURE SURVEY

Performing initial review research is very crucial in grasping self balancing robot control techniques. The study of research related literature carried out for this project will encapsulate some of the topics associated to the methods used for the balancing of robot. This chapter will provide the information needed on the technology that is available in the 'filed of self balancing robot. The methodologies and the procedures used by other researchers around the globe on the two wheel balancing robots topic will also be discussed.

2.1 Ballbot

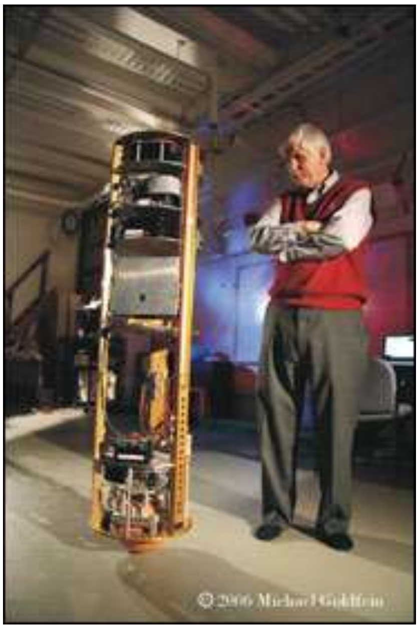


Fig. 2.1 Ralph Hollis with Ballbot

Ralph Hollis, research professor at Carnegie Mellon University, developed an exceptional balancing robot that maintains equilibrium on top of a ball. It's called "Ballbot" [1]. For robots to be fruitful in our daily lives, he believes that some problems should be addressed first. One of the crucial problems is the overall assembly of the robot. Ralph Hollis states, "Robots tall enough to interact effectively in human environments have a high centre of gravity and must accelerate and decelerate slowly, as well as avoid steep ramps, to keep from falling over. To counter this problem, statically stable robots tend to have broad bodies on wide wheel bases, which greatly restricts their mobility through doorways and around furniture or people" [1]. The mobility of the robot is affected by its size. To counter this problem Hollis came up with an enhanced design that improved the structure and mobility of the robot. He built a five foot high, swift, and skinny robot.

Another challenged he faced was to keep the robot stable. To overcome this problem he implemented advance sensors and control algorithm. Gyroscope and accelerometer were used as sensors, which were placed orthogonal to each other. Linear Quadratic Regulator (LQR), a control algorithm, was used to keep the robot in a stable upright state. The LQR is based on optimal control theory. Optimal control techniques minimize the effort to stabilize the robot in vertical position.

The Ballbot's incorporates optimal control algorithms. These algorithms facilitated in increasing system's stability and heftiness. Robot's major forte is the inertial measurement units which gives tilt angle information. The Ballbot's weaknesses were its size and inability to climb stairway.

2.2 nBot and Legway

"nBot is a two-wheeled balancing robot built by David P. Anderson. This robot uses commercially available off the shelf inertial sensors and motor encoders to balance the system" [2]. nBot uses an accelerometer and a gyroscope as its inertial sensors.

"Steven Hassenplug has successfully constructed a balancing robot called Legway using the LEGO Mindstorms robotics kit. Two Electro-Optical Proximity Detector sensors are used to provide the tilt angle information" [3]. A specially created high level programming language for LEGO Mindstrom was used to program the controller. Two optical proximity detectors were used to stabilize the two wheel LEGO robot.

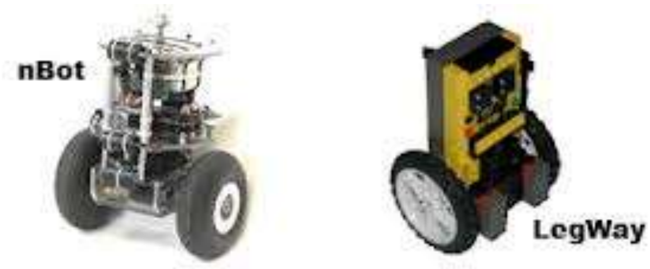


Fig. 2.2 nBot and Legway

Since both these robots use off the shelf parts that are easily accessible and available, they have lower building cost which is their major strength. A disadvantage of both the Legway and nBot is that they cannot travel in all environments and terrains.

2.3 Segway PT

In recent years, the utilization of human transport vehicles has gained popularity. The Segway PT is a common personal vehicle that is accessible to the public. It was invented by Dean Kamen and has the same dynamics as that of an inverted pendulum. The two wheel platform model added mobility to Segway. The Segway integrates a control algorithm, two tilt sensors, five gyroscopes, and two electric motors. For balancing the Segway only three out of five gyroscopes are used, the other two are kept as backup. The Segway's current model can achieve a top speed of 12.5 mph. The Segway is able to steer through rough terrain. The Segway is usually found in urban areas.

The Segway is a personal transporter and is used for outdoor recreation. This is its biggest strength. It is useful for people who cannot walk long distances. It runs on rechargeable batteries which makes it eco-friendly. The major disadvantage of Segway is its cost.

2.4 Kalman Filter

In early 1960's a paper tilled, "A New Approach to Linear Filtering and Prediction Problems." was published by R.E. Kalman. It provided solution to linear filtering problem. It is massively used in the areas of embedded control systems and navigation systems. Greg Welch and Gary Bishop stated, "The Kalman filter is a set of mathematical equations that provides an efficient computational (recursive) means to estimate the state of a process, in a way that minimizes the mean of the squared error. The filter is very powerful in several aspects: it supports estimations of past, present, and even future states, and it can do so even when the precise nature of the modelled system is unknown." [4]. The filter was used in NASA Apollo program to estimate the exact position of the spacecrafts.

Kalman filter is popular in other fields of engineering too. It is frequently used in digital control engineering. It helps to remove measurement noise that can affect the system's performance and gives estimate of the current state of the system. Dan Simon wrote an article, "Kalman Filtering" in which he stated, "The Kalman filter is a tool that can estimate the variables of a' wide range of processes. In mathematical terms we would say that a Kalman filter estimates the states of a linear system. The Kalman filter not only works well in practice, but it is theoretically attractive because it can be shown that of all possible filters, it is the one that minimizes the variance of the estimation error. Kalman filters are often implemented in embedded control systems because in order to control a process, you first need an accurate estimate of the process variables" [3].

Kalman filter present a good estimate of the title angle which is the basic requirement for balancing the robot. It also removes the noise present in the gyroscope and accelerometer data. But the filter has disadvantages too. There is no standard methodology to implement the kalman filter. The filter equations have been expressed in different ways by various authors so it makes it difficult for the robotic enthusiasts to learn and implement the filter. Also, the complex matrix manipulations required to run the filter burdens the microcontroller.

2.5 PID Control System

In two wheeled robot, control system development is a crucial process to guarantee stability. There are numerous control techniques that can be applied to balance the robot but we need to employ the technique that is inexpensive and which doesn't limit the strength and functioning of the controller. System's model and sensors used to obtain title angle decides the elements used in the balance control algorithm.

Control techniques can be divided into two separate groups: Linear control model and Nonlinear controller model. In linear control method the process is developed around a required operating point. Linear method is very sufficient in stabilizing the robot by bringing it back to its vertical position. On the contrary, unrealistic dynamics model is used by the non-linear controller to design a system. Nonlinear controllers offer firmer system implementation. But, since non-liner methods are more difficult to implement and are complex, so researcher usually opt for linear controller to model a system.

For self balancing robot, linear controllers are used. It will be applied through PID (Proportional, Integral and Derivative) controller. It is one of the popular techniques used by control engineering community. The author of article Vance J. VanDoren stated, “For more than 60 years after the introduction of Proportional-Integral-Derivative controllers, remain the workhorse of industrial process control” [5].

2.6 Other Techniques

- Junfeng Wu, Yuxin Liang and Zhe Wang published a paper in 2011 titled, “A Robust Control Method of Two-Wheeled Self Balancing Robot” [6]. They modelled two-wheeled self-balancing robot and designed Sliding Mode Control (SMC) for the system. Highly non-linear mathematical model is derived for the robot. After ignoring the effects of rotation, final model is expressed in state space. The GBOT 1001 robot, manufactured by Googol Technology Ltd., was studied and its mathematical model was established by the use of sliding mode control theory. SMC was used to manage the speed and position of the robot. The advantage of the SMC is that it makes the system robust to the external disturbance and parameter perturbation.

  The simulation results depict that at high speed rate both angle and position of the robot can be stable if SMC is employed. Also, the quality of the system can be improved if the system chattering is reduced. This is done by using a saturated function instead of an ideal sliding mode of symbol function.

- “Design of Fuzzy Logic Controller for Two-wheeled Self-balancing Robot” was published in 2011 by Junfeng Wu and Wanying Zhang [7]. Two-wheeled self-balancing robot is a high order, multiple-variables, non-linear, strong coupling, and instability system. For designing the structure model, kinetic equations were built using Newton dynamics mechanics theory. After that, two controllers with fine simulations curves at identical disturbance force are designed. These controllers are pole placement state feedback controller and fuzzy logic controller. Now the system is exposed to undisturbed and disturbed environment and simulation experiments are done.

The simulation results reveal that fuzzy logic control algorithm is efficient for realizing self balance control and doesn't let the robot to fall. It satisfies the control goal and gives better performance.

- A paper titled, "H∞ Robust Control of Self-Balancing Two-Wheeled Robot" was presented by Xiaogang Ruan and Jing Chen in 2010 at Proceedings of the 8th World Congress on Intelligent Control and Automation [8]. The paper proposed a method based on H∞ Robust Control for movement balancing and position control problems faced by self balancing robot.

  In earlier work, linear state-space model was used to design trajectory-tracking algorithm by studying WIPMR (wheeled inverse pendulum type mobile robot) [9]. Then, F. Grasser derived the system's model [10] and to design a controller the equations were linearized about an operating point. The dynamics of three degree of freedom (3-DOF) system were completely described by six state-space variables. Now, these six state-space model was used to control the robot. H∞ robust control method was implemented to control movement balancing and position.

  The simulation results showed that the robot was balanced at a fixed position in a better way and the performance was not disrupted by external interference.

- In 2011, Fourth International Conference on Intelligent Computation Technology and Automation was conducted in which Wu Junfeng and Zhang Wanying presented a paper titled, "Research on Control Method of Two-wheeled Self-balancing Robot" [11]. A thorough mathematical model is provided by studying two-wheeled robot based on Newton dynamics mechanics theory. A linear state-space equation is developed using a realistic method. . After that, the LQR controller and state feedback controller based on pole placement theory are both designed. After performing a number of experiments, best closed-loop poles and Q, R matrix, having good simulation curves at identical disturbance force, were calculated. It was found that LQR controller is better than pole placement state-feedback controller by studying their respective curves.

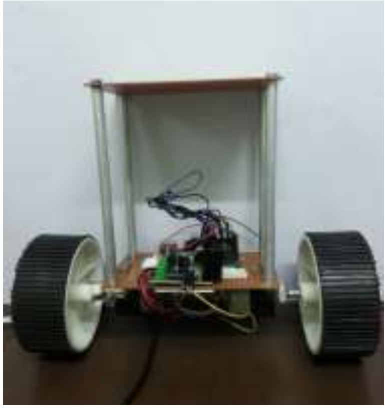
CHAPTER 3

THE BALANCING ROBOT SYSTEM

3. THE BALANCING ROBOT SYSTEM

The self balancing two wheeled robot system is built as part of the master's project. The design of the robot is kept as simple as possible so that it does not affect the purpose of the project which is to balance the robot at its vertical position. To meet the goal of balancing the robot a microcontroller, DC motors and drivers, and inertial sensors are used.

3.1 Robot Physical Structure



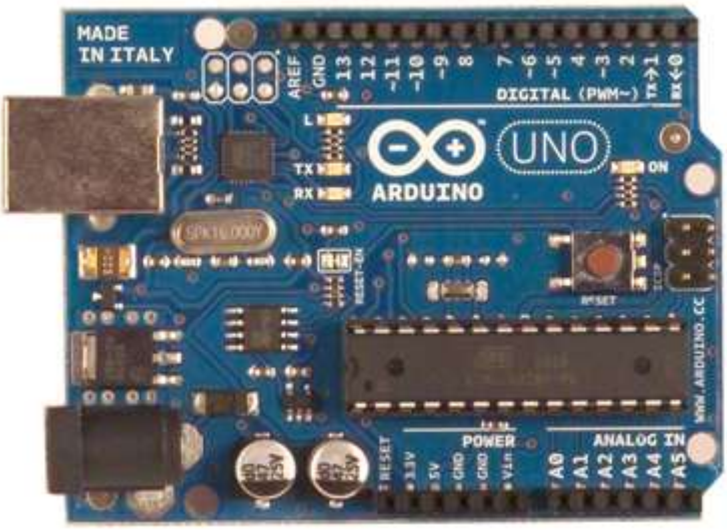
Robot Chassis

The two wheeled robot structure used in this project is a very basic design. The robot chassis design is supported on two 6-3/8inch x 3-7/8inch x 1/16inch glass epoxy board. The sheets are placed on top of one another with the help of 6mm aluminium rods. The space between the sheets helps in assembling electrical components and hardware with ease. Approximate height of the robot, including the wheels, is about 13inch. The two DC motors are bolted to two aluminium L brackets. The brackets are then attached to the lower glass epoxy board and help to attach the motors to the robot chassis.

3.2 Arduino Uno

The mastermind behind the two wheeled robot system is Arduino Uno, a microcontroller board based on Atmega328. It contains 14 digital input/output pins, 6 analog inputs, a USB connection, 16MHz crystal oscillator, a power jack, ICSP header and a reset button.

It integrates everything that microcontroller needs, all we need to do is connecting it to the computer with a USB or power it with AC-to-DC adapter or battery.



Arduino Uno

The Arduino Uno can be powered either by a USB connection or an external power supply. The power source selection is automatic. The I/O pins operate at 5V and each pin can give or take a maximum of 40mA and consists of 20-50kOhms internal pull-up resistor. The 6 analog inputs provide 10bits of resolution.

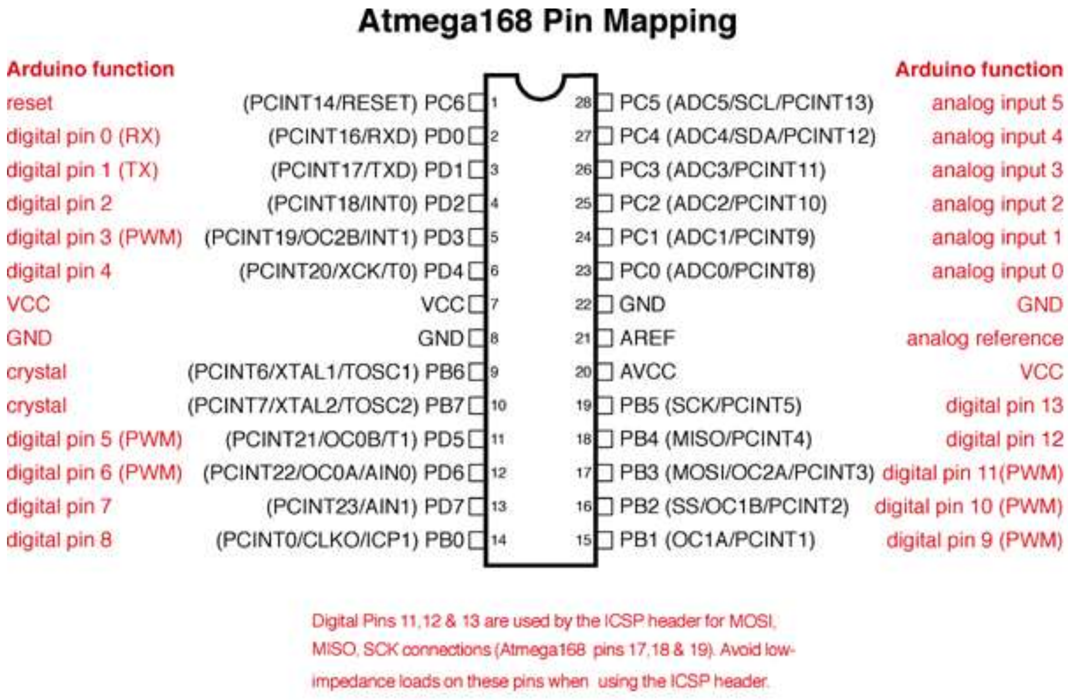
The ATmega328 offers UART TTL (5V) serial communication, available on digital pins 0 (RX) and 1 (TX). I2C and SPI communication are supported by ATmega328. In fact, it includes Wire library to aid I2C communication.

3.2.1 Technical Specifications of Arduino Uno:

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (6 pins provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40mA

DC Current for 3.3V Pin	50mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

3.2.2 Pin Configuration

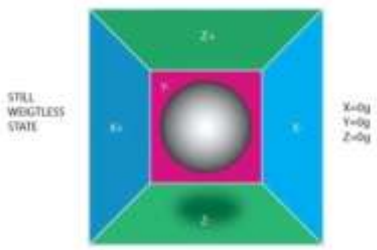


3.3 Inertial Sensors

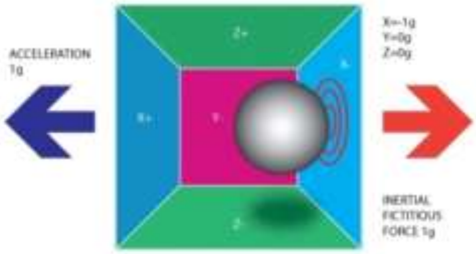
The basic aim of the project is to balance the robot on two wheels. For this purpose the inclination angle of the robot and rate of change of angle is required. The two important sensors used in the project are gyroscope and accelerometer. Accelerometer provides us with the angular data and gyroscope gives us the angular rate data. This data depicts the angle displacement of the robot from true vertical zero angle. The output of both the sensors is then given to Kalman filter, which removes the noise and gives a better estimate of the angle. The estimated angle is then fed to the control system.

3.3.1 Accelerometer

To understand the basic functioning of the accelerometer it is useful to picture a ball inside a box of shape of a cube.

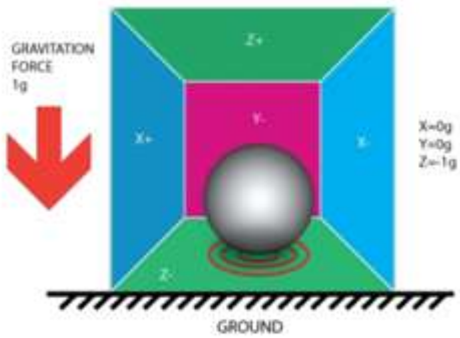


We assume that the box is placed where there is no gravitational field or any other field for that matter which can upset the position of the ball. This means the ball is simply floating in the middle of the box. As depicted in the figure, each axis is assigned to a pair of wall. If the box is moved in the left direction with an acceleration of 1g (9.8m/s^2) the ball exerts the pressure of -1g when it hits the wall (X-).

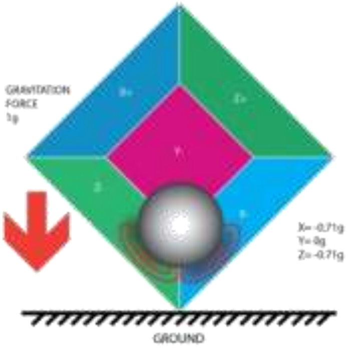


It can be inferred from this that the accelerometer measures the force in the direction that is opposite from the acceleration vector. The force is referred as Inertial force or Fictitious force. Also, accelerometer measures acceleration indirectly through the force applied to one of the walls (according to our model). This force can be due to acceleration or some other underlying factors as depicted in the next example.

If we place the box on Earth the ball will fall on the Z- wall. It will apply a force of 1g on the bottom wall. The pressure applied by the ball is due to the gravitation force. It can be any type of force. For example, if we take metallic ball and place a magnet next to the box, the ball would move and will ultimately hit the wall. This proves that the accelerometer measures force and not acceleration. It's just that acceleration instigate an inertial force that is calculated by the force detection mechanism of the accelerometer.



So far we have seen how single axis accelerometer work but in our case we need to employ a triaxial accelerometer. It detects inertial force on all the axis. In our box model if we rotate the box 45 degree to right than the ball will touch both Z- and X- walls.



Although this model was useful in understanding interaction of the accelerometer with the outside world but its more practical to fix the coordinate system to the axes of accelerometer and study the force vectors.

Assuming that each axis in the new model is perpendicular to respective walls of the box and R is the force vector that needs to be measured then,

$$\mathbf{R^2 = Rx^2 + Ry^2 + Rz^2}$$

Accelerometers are of two types: Digital and Analog. Whereas, Digital accelerometer uses I2C, SPI or USART serial protocols to give the information, Analog accelerometer gives a voltage value in a predefined range which later needs to be converted to digital value with the help of ADC (analog to digital convertor) module. In this project, digital accelerometer is used with I2C serial protocol.

After we get the data from the ADC (AdcRx, AdcRy, AdcRz) we need to calculate the angles. It doesn't matter which ACD we are using, we will get values in a certain range. Let us suppose we employ 16-bit ADC, so we will get values in the range of 0...65,535. Each ADC has a reference voltage (Vref).

To convert ADC values to voltage,

$$\mathbf{VoltsRx = AdcRx * Vref / 65535}$$

$$\mathbf{VoltsRy = AdcRy * Vref / 65535}$$

$$\mathbf{VoltsRz = AdcRz * Vref / 65535}$$

Each accelerometer has a specified zero-g voltage level which corresponds to 0g. To calculate the voltage shift from zero-voltage, we subtract the specified VzeroG from the voltage values.

$$\mathbf{DeltaVoltsRx = VoltsRx - VzeroG}$$

$$\mathbf{DeltaVoltsRy = VoltsRy - VzeroG}$$

$$\mathbf{DeltaVoltsRz = VoltsRz – VzeroG}$$

Accelerometer readings in Volts have now been calculated but we need the final force values in g (9.8 m/s^2). To do so we apply acceleration sensitivity (given in specifications).

$$\mathbf{Rx = DeltaVoltsRx / Sensitivity}$$

$$\mathbf{Ry = DeltaVoltsRy / Sensitivity}$$

$$\mathbf{Rz = DeltaVoltsRz / Sensitivity}$$

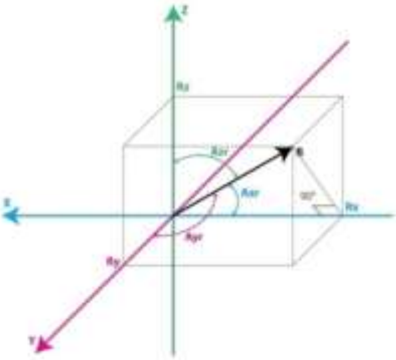
Combining all these steps together we have,

Rx = (AdcRx * Vref / 1023 - VzeroG) / Sensitivity

Ry = (AdcRy * Vref / 1023 - VzeroG) / Sensitivity

Rz = (AdcRz * Vref / 1023 - VzeroG) / Sensitivity

If you wish to calculate inclination of the device relative to the ground, we have to calculate the angle between force vector and Z axis.



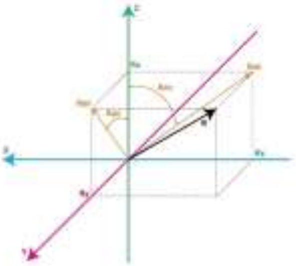
From the figure above we can say,

Axr = arccos(Rx/R)

Ayr = arccos(Ry/R)

Azr = arccos(Rz/R)

3.3.2 Gyroscope



Gyroscope measures rotation around some axis. For example, 2 axes gyroscope measures rotation about the X and Y axes.

Let,

Rxz - is the projection of the vector R on the XZ plane

Ryz - is the projection of the vector R on the YZ plane

Then,

$$\mathbf{Rxz^2 = Rx^2 + Rz^2}$$

$$\mathbf{Ryz^2 = Ry^2 + Rz^2}$$

Let Axz and Ayz be the angles between Rxz and Z axis and Ryz and Z axis respectively.

The rate of change of angles Axz and Ayz are measures by gyroscope, i.e., the output of gyroscope is linearly proportional to the rate of change of these angles. The output is expressed in deg/s.

Similar to accelerometer, ACD values need to be converted to deg/s using a formula given below. If 16bit ADC is used then,

RateAxz = (AdcGyroXZ * Vref / 65535 - VzeroRate) / Sensitivity

RateAyz = (AdcGyroYZ * Vref / 65535 - VzeroRate) / Sensitivity

Where,

AdcGyroXZ, AdcGyroYZ are the values given by ADC module representing the rotation of projection of R vector in XZ respectively in YZ planes.

Vref is the ADC reference voltage we'll use 3.3V in the example below

VzeroRate is the zero-rate voltage

Sensitivity - is the sensitivity of your gyroscope it is expressed in mV / (deg / s)

If the output has a negative sign then it just depicts that the device rotated in the direction opposite to the conventional positive direction.

3.3.3 MPU 6050

MPU-60X0 is the first device 6-axis motion tracking device to have integrated 3 axis gyroscope, 3-axis accelerometer and a digital motion processor. It comes in a 4x4x0.9mm package. It has a dedicated I2C sensor bus which directly accepts the data from 3-axis compass to generate 9-axis motion fusion output.

MPU-60X0 contains three 16-bit ADC each for digitizing outputs of gyroscope and accelerometer. It communicates with all the registers using I2C at 400 kHz. It operates on VDD (voltage range 2.375V-3.46V). It has an additional VLOGIC reference pin.

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

The MPU-60X0 contains three autonomous vibratory MEMS rate gyroscopes. When gyros are rotated, the Coriolis Effect instigates a vibration that is sensed by a capacitive pickoff. The resulting signal is then amplified, demodulated, and filtered to generate a voltage relative to the angular rate. The output is then digitized by the individual on-chip ADC to sample each axis. The entire of the gyro sensors is digitally programmed to ±250, ±500, ±1000, or ±2000 degrees per second (dps). The ADC can be programmed for sample rate ranging from 3.9 - 8,000 samples per second and wide range of cut-off frequencies can be enabled using user-selectable low-pass filters.

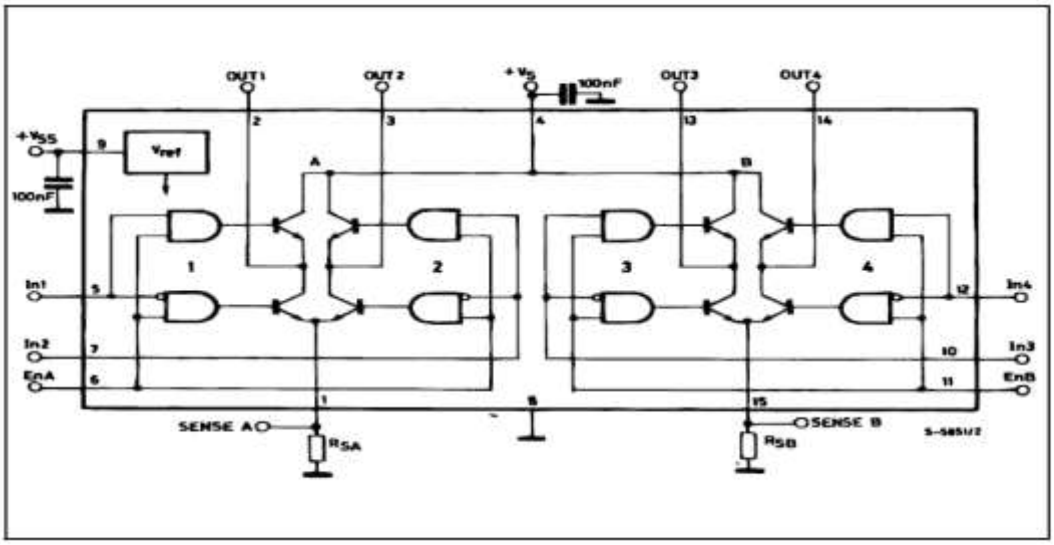
Three-Axis MEMS Accelerometer with 16-bit ADCs and Signal Conditioning

Separate proof masses are used for each axis of the MPU-60X0’s 3-Axis. When acceleration is applied along a particular axis, displacement on the corresponding proof mass is caused, and the displacement is detected differentially by capacitive. The accelerometers’ susceptibility to fabrication variations as well as to thermal drift is reduced because of MPU-60X0’s architecture. The device measures 0*g* on the X- and Y-axes and +1*g* on the Z-axis when placed on a flat surface. The scale factor of the accelerometer is independent of the supply voltage and is calibrated at the factory. It has a devoted sigma-delta ADC for supplying digital outputs. The digital output's full scale range can be attuned to ±2*g*, ±4*g*, ±8*g*, or ±16*g*.

3.4 DC Motor Driver

The motor drivers are very important for the functioning of any motor. They provide high voltage and current to drive the motors .In this project L298 is used as a motor driver. It is a fused monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It provides high voltage and current. It is a dual full bridge driver which acknowledges standard TTL logic levels. It is used to drive inductive loads like DC and stepper motors. It has an enable input which is used to enable or disable the device irrespective of the input signals.

3.4.1 Block Diagram

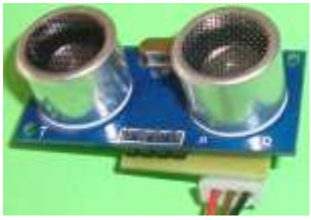


Block Diagram of L298

3.5 Ultrasonic Distance Sensor – (Serial Based)

These sensors are compact in size (SMD design) and have a high range. They are easy to implement so are commonly used for distance measurement and mapping. It has both transmitter and receiver to send and receive electric signals.

The output of the sensor can be read in every 100ms in serial ASCII format and hence it is easy to read and process it.



Ultrasonic Sensor

3.5.1 Features of the sensor

Measurable distances	10cm to 400cm(4 Meters)
Supply voltage	5V DC
Accuracy	+-1cm
Modulated	40 kHz
Serial Data Rate	9600 bps TTL level output

CHAPTER 4
KALMAN FILTER

4. KALMAN FILTER

R.E. Kalman, in 1960, published a paper in which the discrete data linear filtering problem was solved using an iterative process. Since then it has been used massively in control engineering community.

The Kalman filter is a recursive filter which estimates the state of any linear process in which noise values are uncorrelated with time. One of the features of the filter is its ability to minimise the mean square error. It is because of this feature is it used in control system exposed to noisy environments

While using Kalman filter one does not need to store all the previous state values. The present state (time step k) depends only on previous state (time step k-1). So it can be easily implemented on small microcontroller with limited storage capacity.

4.1 The Process to be Estimated

The Kalman filter guesstimate the state $x \in \Re^n$ of a discrete-time controlled process that is governed by the linear stochastic difference equation

$$X_{k+1} = AX_k + w_k \,,$$

with a measurement $z \in \Re^m$ that is

$$Y_k = HX_k + v_k \,.$$

Where,

- w_k is process noise with $p(w) \sim N(0 , Q)$
- v_k is measurement noise with $p(v) \sim N(0 , R)$
- A is nxn matrix which related current state (time step k) to previous state (time step k-1) when process noise is not present
- H is mxn matrix which related the state to measurement z_k

The value of Q (process noise covariance), R (measurement noise covariance), A and H might change during the process but here be consider them to be constant.

4.2 Mathematics involved

In Kalman filter we do prediction and correction to get good results.

Estimation = Prediction + Correction

- State vectors (equations of a moving object)

$$X_{k+1} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X\ coordinate \\ Y\ coordinate \\ V_x \\ V_y \end{bmatrix} + \begin{bmatrix} W_x \\ W_y \\ Wv_x \\ Wv_y \end{bmatrix}$$

$$\Rightarrow X_{k+1} = AX_k + w_k$$

Also,

$$Y_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} X\ coordinate \\ Y\ coordinate \\ V_x \\ V_y \end{bmatrix}_k + \begin{bmatrix} e_{sx} \\ e_{sy} \end{bmatrix}$$

$$Y_k = HX_k + v_k$$

- Estimate of X_k based on observations

$$E\left(X_{k+1} / Y^k\right) = E\left(AX_k / Y^k\right) + E\left(W_k / Y^k\right)$$

Since W_k is zero mean Gaussian noise i.e. $E(W_k) = 0$ and $E(W_k W_k^T) = Q$,

$$\Rightarrow \hat{X}_{k+1/k} = A\hat{X}_{k/k}$$

Error = Actual – Estimate

$$\tilde{X}_{k+1/k} = A\tilde{X}_{k/k} + w_k$$

Similarly,

$$\hat{Y}_k = E\left(HX_{k/k}\right) + E\left(v_k\right)$$

$$\hat{Y}_k = H\hat{X}_k$$

- Covariance of prediction error

$$P_{k+1/k} = E\left(\tilde{X}_{k+1/k}\, \tilde{X}^{T}_{k+1/k} \right)$$

$$P_{k+1/k} = E\left(\left(A\tilde{X}_{k/k} + w_k \right)\left(A\tilde{X}_{k/k} + w_k \right)^{T} \right)$$

$$= A\,E\left(\tilde{X}_{k/k}\, \tilde{X}^{T}_{k/k} \right)A^{T} + A\,E\left(\tilde{X}_{k/k}\, w_k^{T} \right) + E\left(w_k\, \tilde{X}^{T}_{k/k} \right)A^{T} + E\left(w_k w_k^{T} \right)$$

$$P_{k+1/k} = AP_{k/k}A^{T} + Q$$

- Estimate = Prediction + Correction

Correction = Observation – Estimation

$$\hat{X}_{k+1/k+1} = \hat{X}_{k+1/k} + K^{k+1}\left(Y_{k+1/k} - H\hat{X}_{k+1/k} \right)$$

Error in filtered estimate

$$\tilde{X}_{k+1/k+1} = \tilde{X}_{k+1/k} - K^{k+1}\left(HX_{k+1/k} + v_{k+1} - H\hat{X}_{k+1/k} \right)$$

$$= \tilde{X}_{k+1/k} - K^{k+1}\left(H\tilde{X}_{k+1/k} + v_{k+1} \right)$$

Error covariance matrix,

$$P_{k+1/k+1} = E\left(\tilde{X}_{k+1/k+1}\, \tilde{X}^{T}_{k+1/k+1} \right)$$

$$P_{k+1/k} = E\left(\left(\tilde{X}_{k+1/k} + K^{k+1}\left(H\tilde{X}_{k+1/k} + v_{k+1} \right) \right)\left(\tilde{X}_{k+1/k} + K^{k+1}\left(H\tilde{X}_{k+1/k} + V_{k+1} \right) \right)^{T} \right)$$

$$= E\left(\tilde{X}_{k+1/k}\, \tilde{X}^{T}_{k+1/k} \right) - E\left(\tilde{X}_{k+1/k}\, \tilde{X}^{T}_{k+1/k} \right)H^{T}K^{k+1^{T}} - K^{k+1}H\,E\left(\tilde{X}_{k+1/k}\, \tilde{X}^{T}_{k+1/k} \right)$$

$$+ K^{k+1}H\,E\left(\tilde{X}_{k+1/k}\, \tilde{X}^{T}_{k+1/k} \right)H^{T}K^{k+1^{T}} + E\left(v_k v_k^{T} \right)$$

$$P_{k+1/k+1} = P_{k+1/k} - P_{k+1/k}H^{T}K^{k+1^{T}} - K^{k+1}HP_{k+1/k} + K^{k+1}HP_{k+1/k}H^{T}K^{k+1^{T}} + K^{k+1}RK^{k+1^{T}}$$

Gain K^{k} should be chosen to minimize $P_{k+1/k+1}$

$$\frac{\partial\left(P_{k+1/k+1}\right)}{\partial\left(K^{k+1}\right)} = 0$$

$$-2P_{k+1/k}H^{T} + 2K^{k+1}\left[HP_{k+1/k}H^{T} + R\right] = 0$$

$$K^{k+1} = P_{k+1/k}H^{T}\left[HP_{k+1/k}H^{T} + R\right]^{-1}$$

from equation (5) *and* (6)

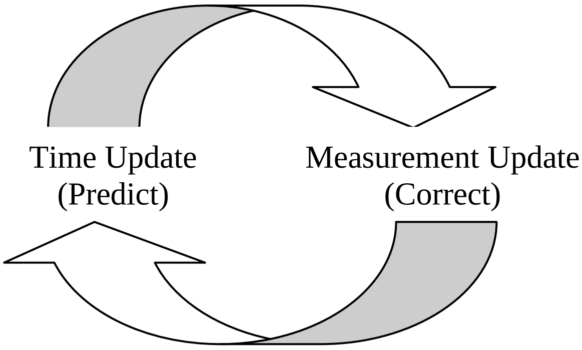
$$P_{k+1/k+1} = P_{k+1/k}\left[I - H^{T}K^{k+1^{T}}\right] - K^{k+1}HP_{k+1/k} + K^{k+1}HP_{k+1/k}H^{T}K^{k+1^{T}} + K^{k+1}RK^{k+1^{T}}$$

$$P_{k+1/k+1} = \left[I - HK^{k+1}\right]P_{k+1/k}$$

4.3 The Discrete Kalman Filter Algorithm

The Kalman filter uses feedback control to estimate a process. A process is approximated at some given time and the filter attains feedback in the form of noisy measurement. The equations of the Kalman filter can be divided into two factions: Time update equations and measurement update equations.

The time update equations are accountable for calculating a-prior estimates at a time step k when the current state and error covariance estimates are known for the time step k-1. The measurement update equations are responsible for integrating a measurement with a-prior estimate to calculate a-posterior estimate. So, time update equations can be referred as predictor equations and measurement update equations as corrector equations. Hence, the combination of the two resembles a predictor and corrector algorithm.



Kalman Filter Cycle

Discrete Kalman Filter Time Update Equations

$$\hat{X}_{k+1/k} = A\hat{X}_{k/k} \boxed{?}$$

$$P_{k+1/k} = AP_{k/k}A^{T} + Q$$

As shown in the table, the time update equations predicts the state and covariance estimates at time step k+1 from the values known at time step k.

Discrete Kalman Filter Measurement Update Equations

$$K^{k+1} = P_{k+1/k}H^{T}\left[HP_{k+1/k}H^{T} + R\right]^{-1}$$

$$\hat{X}_{k+1/k+1} = \hat{X}_{k+1/k} + K^{k+1}\left(Y_{k+1/k} - H\hat{X}_{k+1/k}\right)$$

$$P_{k+1/k+1} = \left[I - HK^{k+1}\right]P_{k+1/k}$$

The Kalman gain K^{k+1} is computed. In the next step the process is measured to get the value of $Y_{k+1/k}$ and a-posterior state is estimated. Then finally a-posterior error covariance estimate is obtained.

After each update pair the process is repeated and the previous a-posterior estimates are copied to the new a-prior estimate. It is because of this feature the practical implementation of Kalman filter is more feasible than of Wiener filter. [12]

4.4 Sensor fusion using Kalman Filter

Rich Ooi gave the reason for using sensor fusion which is as follows, “the accuracy and reliability of information regarding its operating environment for these mobile robots is critical, as these systems are usually autonomous. These requirements call for highly accurate sensors which are very expensive. Sensor fusion technology, where signal from several sensors are combined to provide an accurate estimate, is the most widely used solution. The Kalman filter is used in a number of multi-sensor systems when it is necessary to combine dynamic low-level redundant data in real time" [2].

The Kalman, "filter uses the statistical characteristics of a measurement model to recursively determine estimates for fused data that are optimal in a statistical sense. The recursive nature of the filter makes it appropriate for use in systems without large data storage capabilities" [2], such as Ardunio Uno.

A question always pop up in my mind that if the accelerometer is giving the inclination angle then why do we employ gyroscope? The answer to this question is very simple. The data obtained from the accelerometer cannot be trusted. Accelerometer measures inertial force caused generally by gravitation but it can also be caused by the movement of the device. So even if the accelerometer is in stable state, it is susceptible to mechanical noise and vibration.

Even though gyroscope measures rotation, gyroscope is not noise free. It is sensitive to drift.

Gyroscope has a rest average value, also called bias. The output given by the gyroscope in rest position is the average value. Gyroscope calculates angular rate change so when the object is tilted the output changes but it returns to the resting average value. Since gyroscope "drifts" from the actual value when it is motionless, it cannot be used as a tilt sensor.

Kalman filter is used for sensor fusion. This technique provides output signal in the same form as the original signal of the sensor but with better quality.

CHAPTER 5

THE PID ALGORITHM

5. THE PID ALGORITHM

The most common algorithm used in industrial process control is PID algorithm (Proportional, Integral and Derivative). In PID algorithm, the output is moved in the direction that makes the process reach a set point. The algorithm should incorporate feedback because if the path between the output to the input id broken, the output can never be reached.

There are a number of different controllers used in the industries. The controllers can be divided into two main groups: Conventional controllers and Unconventional controllers. For designing conventional controllers, mathematical model of the process should be known. P, PI, PID, Otto-Smith are some of the common conventional controllers known. On the other hand, for designing unconventional controllers, mathematical model of the process is generally not required. Fuzzy controller and neuro or neuro-fuzzy controllers are the examples of unconventional controllers.

Few processes are nonlinear and are hard to illustrate mathematically. But PID controllers can control nonlinear processes effectively provided the parameters are well tuned. The controllers are based on three basic behaviour types: Proportional (P), Integral (I), and Derivative (D).

5.1 A Proportional algorithm

The mathematical representation is,

$$\frac{mv(s)}{e(s)}=k_c \text{ (laplace domain) or } mv(t) = mv_{ss} + k_c e(t) \text{ (time domain)}$$

The output is adjusted in direct proportion to the controller input (error signal) in proportional mode. K_c, the controller gain is the adjustable parameter that needs to be specified. The controller output will change more for a given error is the k_c is increased. K_c is also referred as Proportional Band (PB). Proportional band is defined as the amount of change in the controlled variable required to drive the loop output from 0 to 100%.

Calibration around the steady-state operating point is required by the controller. It is expressed in the time domain expression and is denoted by the constant term mv_{ss}. This term disappears in the Laplace domain because of the 'deviation variable' representation.

The proportional controller does not eliminate the error but reduces it, i.e an offset will normally exist between the actual and desired value.

5.2 A Proportional Integral algorithm

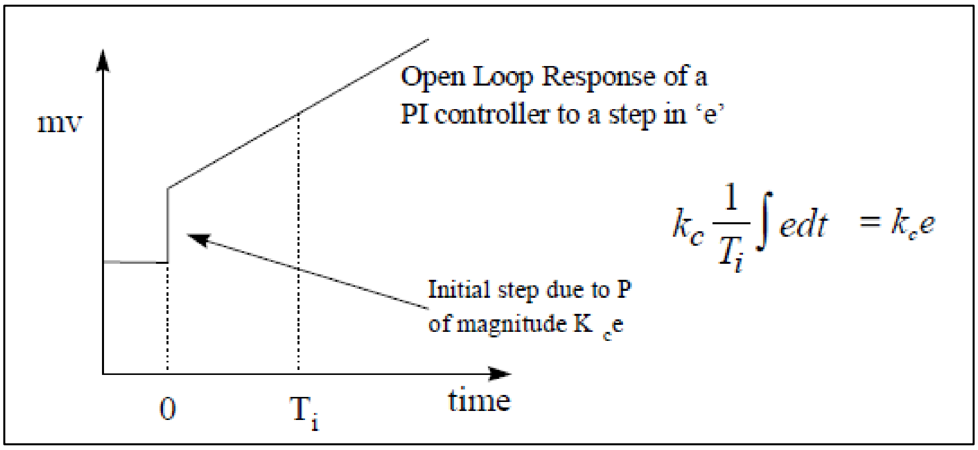
The mathematical representation is,

$$\frac{mv(s)}{e(s)} = k_c\left[1 + \frac{1}{T_i s}\right] \quad \text{or} \quad mv(t) = mv_{ss} + k_c\left[e(t) + \frac{1}{T_i}\int e(t)dt\right]$$

The integral mode added to the existing proportional mode is often referred as reset. The offset (error) that occurs between the desired value and the process output is corrected by the additional integral mode automatically over time. The integral time (T_i) must be specified.

- **What's Reset?**

It is used to express the integral mode. The time required by the integral action to produce a change in mv that is equivalent to the initial change produced by P mode is referred to as Reset.



The response of a PI algorithm to a step in error

The output of K_ce steps immediately due to P mode. The integral mode then triggers mv to ‘ramp’. The mv is again increased by K_ce over the period 'time 0 to time T_i'.

- **Integral wind-up**

The integral term of the controller (the one that possesses integral action) increases, at a rate administered by the integral time of the controller, if the controller receives an error signal for a considerable amount of time. This causes the manipulated variable to reach its maximum or minimum limit (i.e. 100% or 0% of its scale). This is referred as integral wind-up. Override is one of the major scenarios in which a sustained error can occur. It happens when a different controller takes control of a particular loop, e.g. because of safety reasons. During override the original controller doesn’t shut, i.e. it still receives the error signal. Because of this, integral wind-up occurs.

External reset feedback is one of the techniques that can stop the integral wind-up. Here, the controller also receives the signal from the control valve. The controller is designed in a way that it can integrate the error signal before it enters the control valve and breaks the loop if the valve is manipulated by the override controller.

5.3 A Proportional Integral Derivative algorithm

The mathematical representation is,

$$\frac{mv(s)}{e(s)} = k_c\left[1 + \frac{1}{T_i s} + T_D s\right] \quad \text{or} \quad mv(t) = mv_{ss} + k_c\left[e(t) + \frac{1}{T_i}\int e(t)dt + T_D\frac{de(t)}{dt}\right]$$

If rate change of the controlled variables is known, the derivative action predicts where the process is heading. The derivative action is characterised by ‘rate time’ ($T_{D,}$ in minutes). The derivative action ideally enhances the dynamic response and it is achieved by an iterative process. However, the derivative action is undesirable if noise signals are present because differentiating noise signal can cause unnecessary mv movement.

Unlike in P and I, the derivative action depends on the rate of change of the error. So, derivative action will have no effect if the error is constant.

5.4 Topology of PID Controllers

Question of topology (structure) of controller occurs when:

- control system needs to be designed (defining structure and controller parameters)
- controller needs to be tuned

There are a number of different PID controller structures.

5.4.1 Ideal PID

The mathematical representation of this algorithm is:

$$\frac{mv(s)}{e(s)} = k_c\left[1 + \frac{1}{T_i s} + T_D s\right]$$

One of the major disadvantages of the ideal configuration is that the derivative term aggravates when the setpoint changes suddenly. So, the final control elements receive a "derivative kick", which is not desirable.

To overcome this issue, derivative mode is made to act operate on measurement instead of error. The output will then gradually move after the change in setpoint, avoiding "derivative kick". This is the standard feature that most commercial controller presents.

$$mv(s) = k_c\left[1 + \frac{1}{T_i s}\right]e(s) + T_D s v(s)$$

5.4.2 Series (interacting) PID

The mathematical representation of this algorithm is:

$$\frac{mv(s)}{e(s)} = k_c\left[1 + \frac{1}{T_i s}\right]T_D s$$

A minor variation can be done in the ideal implementation of series PID. Here the derivative mode can work either on measurement or error depending upon the requirement. In this case, the mathematical representation is,

$$\frac{mv(s)}{e(s)} = k_c \left[1 + \frac{1}{T_i s}\right]$$ where e(s) = SP - $T_D$scv(s)

5.4.3 Parallel PID

The mathematical description is,

$$mv(s) = k_c e(s) + \frac{1}{T_i s} e(s) + T_D s e(s)$$

Here the proportional gain acts only on the error. However, in the ideal algorithm it acts on integral and derivative mode also.

5.5 PID Tuning Methods

The basic requirement is stability. The control parameters should be adjusted to finest values to attain desired control response. This is called Tuning. But different systems behave differently and different applications have dissimilar requirements and these requirements might conflict with each other.

There are just three simple parameters in PID algorithm but still they create a problem in PID tuning because a complex criteria needs to be satisfied keeping in mind the limitations of the PID controller. There are various techniques for loop tuning from which three are described:

- Manual tuning method
- Ziegler–Nichols tuning method
- PID tuning software methods

Parameter	Speed of Response	Stability	Accuracy
Increasing K	Increases	Deteriorate	Improves
Increasing K_i	Decreases	Deteriorate	Improves
Increasing K_D	Increases	Improves	No impact

Effects of Coefficients

5.5.1 Manual Tuning Method

In manual tuning method, parameters are manipulated according to the system responses. The desired system response is achieved by changing the values of K_p, K_i and K_D.

- **Example**

Initially, null values are given to K_i and K_D. Now, K_p is made to increase gradually till loop's output oscillates. Once the optimum value of K_p is obtained, it is replaced by the value that is approximately half of the optimum value so as to achieve "quarter amplitude decay" response. After this, the value of K_i is amplified till the time any offset is amended in adequate time for the process. Nevertheless, higher value of K_i causes instability. Finally, K_D is increased till the loop reaches its reference after load disturbance. However, excessive response and overshoot can be caused by too much K_D.

To reach the setpoint quickly, PID tuning usually overshoots. In few cases where overshoot is not acceptable, over-damped closed loop system is used. For this, K_p should be given a value that is less than half of value that causes oscillation.

Parameter	Rise Time	Overshoot	Settling Time	S.S. Error	Stability
K_p	Decrease	Increase	Small Change	Decrease	Worse
K_i	Decrease	Increase	Increase	Significant Decrease	Worse
K_D	Minor Decrease	Minor Decrease	Minor Decrease	No Change	If K_D small, better

Effects of changing control parameters

5.5.2 Ziegler-Nichols Tuning Method

In 1940s, John G. Ziegler and Nathaniel B. Nichols introduced this closed loop method which is based on experiments performed on a conventional control system.

The procedure is as follows:

1. The process is brought close to the operating point to guarantee that the controller is sensitive to the process dynamic and also to minimize the chances of variables

reaching their limit during the tuning process. The controller is put in manual mode, so that the parameters can be so adjusted that the process is brought to the operating point.

2. The PID controller is then turned into P controller by assigning '∞' and '0' to T_i and T_D respectively. Now, K_p is set to '0' and then put the controller in automatic mode.

3. The value of K_p is increased till the time sustained oscillations are reached in the control system. This value is denoted by K_{pu}, ultimate gain. In the setpoint, the excitation can be a step. It should be small in magnitude. For instance, 5% of the maximum setpoint range is taken as a step so that the process doesn't move away from the desired operating point where the process may exhibit different dynamic properties. Also, it the magnitude of the step should not be too small cause then the proper oscillation cannot be observed due to noise (measurement). The value of K_{pu} must be set up without control signals entering saturation limit during oscillations. So it can be said that smallest value of K_p, which moves the control loop into continuous oscillations, should be assigned to K_{pu}.

4. The critical period $_{Pu}$ of sustained oscillations is measured.

5. The values of the parameters are calculated according to the table given below. The stability of the controller can be improved by decreasing the value of K_p.

Control Type	K_p	K_i	K_D
P	0.5*K_u	-	-
PI	0.45*K_u	1.2K_p/T_u	-
PID	0.6*K_u	2*K_p/T_u	K_p*T_u/8

Ziegler–Nichols tuning method, gain parameter's calculation

5.5.3 PID Tuning Software

There are few softwares through which the values of gain parameters can be calculated. Any of the abstract techniques can be used in these softwares.

Some of the softwares used are:

- MATLAB Simulink PID Controller Tuning,
- BESTune, Exper Tune etc.

5.5.4 Overview of tuning Methods

Method	Advantage	Disadvantage
Manual Training	No math required. Online method	Requires experienced personnel
Ziegler-Nichols	Proven Method. Online method	Process upset, some trial-and-error, very aggressive tuning
Software tools	Consistent tuning. Online or offline method. May include valve and sensor analysis. Allow simulation before downloading. Can support Non-Steady State (NSS) tuning.	Some cost and training involved
Cohen-Conn	Good process models	Some math. Offline method. Only good for first-order processes.

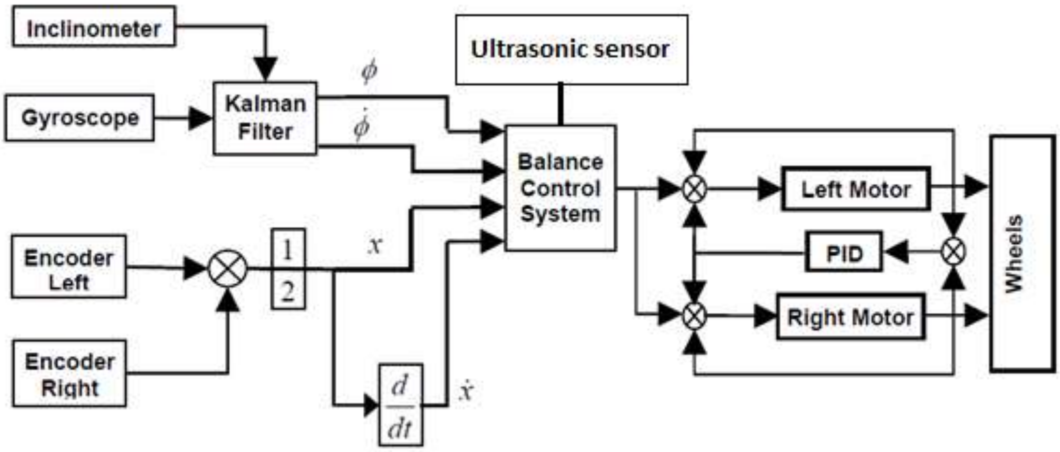
Overview of tuning methods

CHAPTER 6

THE PROPOSED METHOD

6. THE PROPOSED METHOD

6.1 Block Diagram



Functional block diagram of the two wheeled self balancing robot

6.2 Algorithm for Balancing the Robot

Initially, the inbuilt libraries are included. The libraries are

- **Wire.h** : to communicate with I2C / TWI devices.
- **I2Cdev.h** : to communicate with the sensors (MPU6050)
- **Mpu6050.h** : to interface the sensors with the Arduino board.

A. The raw data is first read form the sensor (MPU6050). The sensor gives us 3-axis accelerometer reading (ax, ay, az) and 3-axis gyroscope reading (gx, gy, gz).

B. For this project y axis and z axis values of the accelerometer and x axis value of the gyroscope are required.

By using the y and z axis values of accelerometer the angle is calculated which is then converted from radian to degrees. Finally the angles are mapped from -180º to 180º.

The gyroscope rate is calculated using x axis value and gyro SCL value (defined in the datasheet).

gyro_rate = gx / GYRO_SCL; (GYRO_SCL = 131.0 according to the datasheet)

C. The raw values obtained from the sensors are not proper so we need to correct them using Kalman filter. The Kalman filter gives us the filtered angle.

The data from form accelerometer is used by the Kalman filter to remove the drift problem of the gyroscope output. The Kalman filter is modelled as a single dimensional IMU. For the single dimensional IMU, the angle of the robot and the gyroscope bias values are tracked using two state Kalman filter.

Here , the initial values of Q and R are taken as

$$Q = \begin{bmatrix} Q_{acc} & 0 \\ 0 & Q_{gyro} \end{bmatrix}$$

$$= \begin{bmatrix} 0.001 & 0 \\ 0 & 0.003 \end{bmatrix}$$

And,

$$R = [R\_angle]$$

$$= [0.03]$$

The 2x2 error covariance matrix P is taken as 0.

Now, using the iterative Kalman filter, the filtered angle is obtained which is approximately noise free.

D. Now, PID is applied. The setpoint for this project is where the robot is balanced so the value of setpoint is taken as 0. Firstly, error, which is the distance away from the setpoint, is calculated by using the formula

error = Setpoint – filtered angle;

Where, filtered angle is the output obtained from the Kalman filter.

Now, the direction in which the robot should be moved in order to balance it is judged by the value of error. If error value is greater than one then the robot is moved in the forward direction else it is moved in the reverse direction.

Now, error is given as input to the PID. In this we first calculate the value for integral controller (ITerm) because to reduce the time the integral controller needs to settle down after the error was recovered. So if ITerm is greater than 255, it is given 255 and if it is less tahn -255 it is given -255. This is done to limit its value.

$$\text{ITerm} = (\text{Ki} * \text{error})$$

Finally the output of the PID is calculated using,

$$\text{output} = \text{Kp} * \text{error} + \text{ITerm} - \text{Kd} * \text{dInput}$$

Where,

$$\text{dInput} = (\text{filtered angle at time step } k - \text{filtered angle at time step } k\text{-}1)$$

The output is nothing but the speed variable.

This method can balance the robot till the maximum tilt of 30 ^{o}.

6.3 Algorithm for Object Following

The ultrasonic sensor is attached to the Adruino board. The emitter continuously emits light. When the light hits the target it is reflected back and the detector detects it. The distance is calculated between the robot and the object by using the time lapse.

In this project if the robot detects any object the robot is moved in the forward direction else it just performs the balancing function.

CHAPTER 7

RESULTS AND DISCUSSIONS

7. RESULTS AND DISCUSSIONS

The main objective of the project was to use nominal components to design a closed loop system. But doing so led to few drawbacks, like in case of accelerometer and gyroscope. Both these sensor created few problems initially which made them difficult to implement. The accelerometer data was corrupted with noise and gyroscope bias value used to drift away with time. To correct these problems Kalman filter was applied. It helped in minimizing the unwanted output. The PID controller depends on noise free input signal.

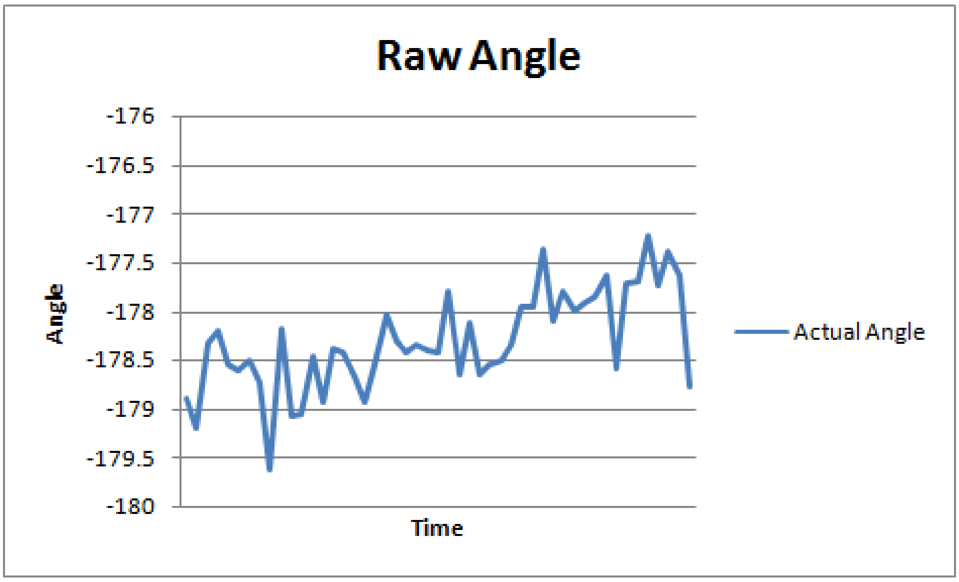
The data obtained from the sensors and Kalman filter and the PID controller are depicts in the tabular and graphical form.

7.1 Robot in base position

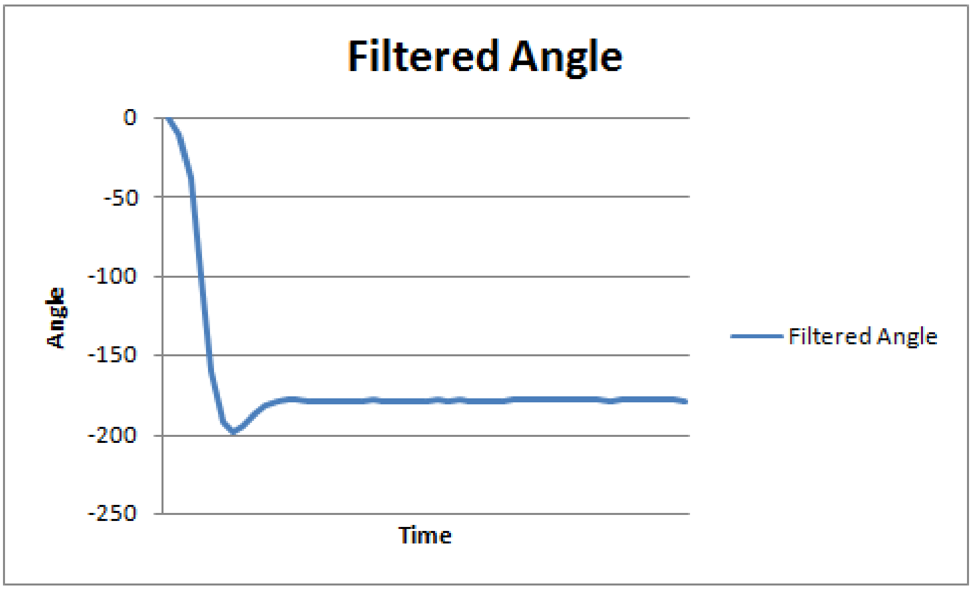
Actual Angle	Gyro Rate	Filtered Angle	Error	Proportional	Integral	Derivative	output	Direction	Speed
-178.89	-3.77	-0.02	0.02	0.58	0	-1.75	2.34	f	0
-179.2	-3.74	-9.76	9.76	292.68	0	-876.28	255	f	255
-178.33	-4.05	-38.01	38.01	1140.31	0	-2542.91	255	f	0
-178.21	-4.73	-100.84	100.84	3025.11	0	-5654.39	255	f	0
-178.55	-0.76	-159.37	159.37	4781.03	0	-5267.76	255	f	0
-178.6	-2.89	-190.96	190.96	5728.84	0	-2843.44	255	f	0
-178.51	-2.2	-197.93	197.93	5937.89	0	-627.14	255	f	0
-178.73	-3.15	-193.51	193.51	5805.16	0	398.2	255	f	0
-179.63	-3.76	-186.95	186.95	5608.5	0	589.97	255	f	0
-178.17	-4.54	-181.05	181.05	5431.45	0	531.13	255	f	0
-179.08	-3.13	-178.3	178.3	5349.02	0	247.3	255	f	0
-179.05	-3.63	-178.04	178.04	5341.3	0	23.17	255	f	0
-178.46	-3.14	-177.91	177.91	5337.32	0	11.94	255	f	0
-178.93	-3.59	-178.62	178.62	5358.67	0	-64.07	255	f	0
-178.38	-3.44	-178.51	178.51	5355.25	0	10.25	255	f	0
-178.43	-3.27	-178.41	178.41	5352.3	0	8.87	255	f	0
-178.67	-3.93	-178.73	178.73	5361.99	0	-29.08	255	f	0
-178.93	-2.99	-178.77	178.77	5363.02	0	-3.07	255	f	0
-178.54	-2.69	-178.5	178.5	5354.97	0	24.15	255	f	0
-178.04	-1.9	-177.91	177.91	5337.27	0	53.07	255	f	0
-178.31	-3.38	-178.38	178.38	5351.42	0	-42.45	255	f	0
-178.42	-2.9	-178.42	178.42	5352.63	0	-3.63	255	f	0

-178.35	-3.19	-178.43	178.43	5352.77	0	-0.4	255	f	0
-178.41	-3.05	-178.41	178.41	5352.18	0	1.76	255	f	0
-178.43	-2.76	-178.36	178.36	5350.92	0	3.77	255	f	0
-177.8	-4.15	-178.13	178.13	5343.92	0	21.01	255	f	0
-178.64	-2.44	-178.27	178.27	5348.12	0	-12.61	255	f	0
-178.11	-2.63	-178.13	178.13	5343.97	0	12.45	255	f	0
-178.64	-4.79	-178.9	178.9	5366.91	0	-68.8	255	f	0

- Graph depicting the raw data obtained from the sensors (base position)



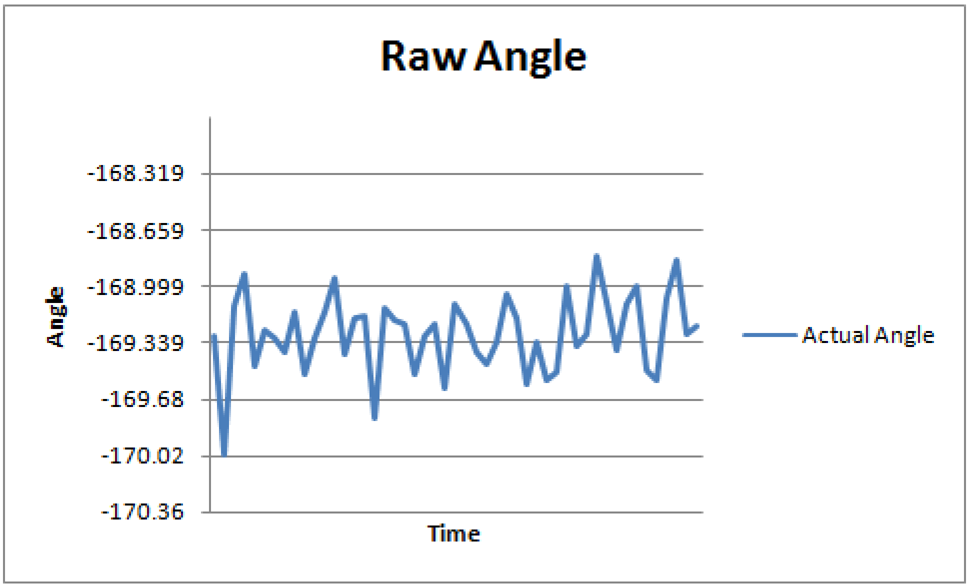
- Graph depicting the data obtained from the Kalman filter (base position)



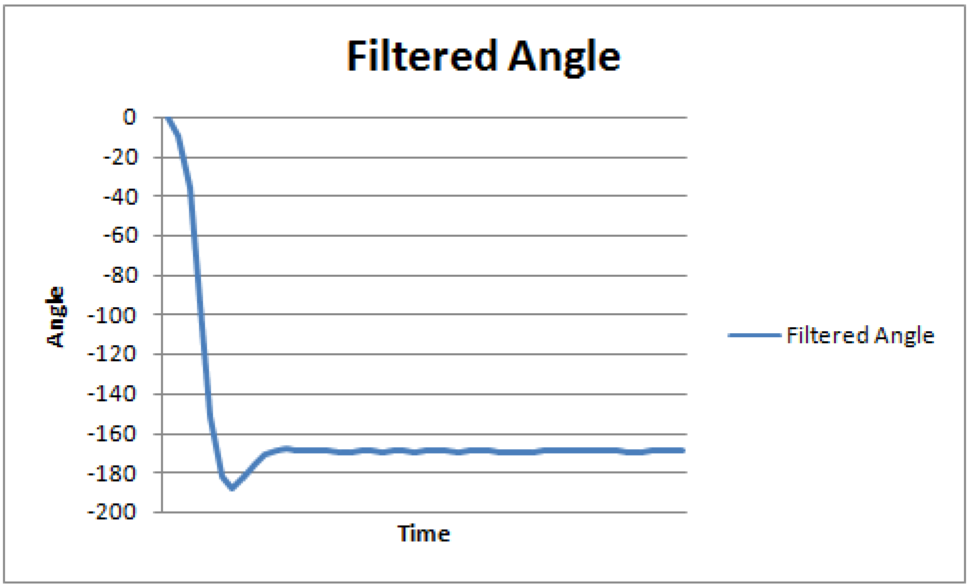
7.2 Robot tilted towards the front

Actual Angle	Gyro Rate	Filtered Angle	Error	Proportional	Integral	Derivative	output	Direction	Speed
-169.3	-3.55	-0.02	0.02	0.55	0	-1.65	2.21	f	0
-170.02	-3.38	-9.09	9.09	272.67	0	-816.37	255	f	255
-169.12	-3.27	-35.42	35.42	1062.57	0	-2369.69	255	f	0
-168.93	-3.37	-94.49	94.49	2834.62	0	-5316.14	255	f	0
-169.49	-3.15	-151.99	151.99	4559.73	0	-5175.32	255	f	0
-169.27	-3.42	-181.67	181.67	5450.04	0	-2670.96	255	f	0
-169.32	-3.25	-188.21	188.21	5646.22	0	-588.52	255	f	0
-169.4	-3.54	-183.52	183.52	5505.55	0	422.02	255	f	0
-169.16	-3.29	-176.09	176.09	5282.74	0	668.41	255	f	0
-169.54	-3.09	-171.08	171.08	5132.53	0	450.63	255	f	0
-169.32	-3.11	-168.81	168.81	5064.28	0	204.74	255	f	0
-169.16	-3.21	-168.34	168.34	5050.19	0	42.27	255	f	0
-168.95	-3.32	-168.5	168.5	5055.04	0	-14.53	255	f	0
-169.41	-3.4	-169.14	169.14	5074.09	0	-57.16	255	f	0
-169.2	-3.22	-169.23	169.23	5076.97	0	-8.65	255	f	0
-169.18	-3.03	-169.18	169.18	5075.38	0	4.78	255	f	0
-169.8	-3.29	-169.71	169.71	5091.2	0	-47.47	255	f	0
-169.14	-3.51	-169.41	169.41	5082.36	0	26.53	255	f	0
-169.21	-3.37	-169.22	169.22	5076.48	0	17.63	255	f	0
-169.23	-3.19	-169.15	169.15	5074.42	0	6.19	255	f	0
-169.54	-3.02	-169.4	169.4	5081.91	0	-22.47	255	f	0
-169.3	-3.18	-169.37	169.37	5081.15	0	2.26	255	f	0
-169.23	-3.08	-169.24	169.24	5077.29	0	11.6	255	f	0
-169.62	-3.24	-169.56	169.56	5086.72	0	-28.31	255	f	0
-169.11	-3.19	-169.23	169.23	5076.87	0	29.55	255	f	0
-169.23	-3.23	-169.2	169.2	5076.01	0	2.58	255	f	0
-169.4	-3.32	-169.37	169.37	5081.23	0	-15.67	255	f	0
-169.47	-3.57	-169.52	169.52	5085.66	0	-13.29	255	f	0
-169.34	-3.18	-169.34	169.34	5080.08	0	16.76	255	f	0

- Graph depicting the raw data obtained from the sensors (forward tilt)



- Graph depicting the data obtained from the Kalman filter (forward tilt)

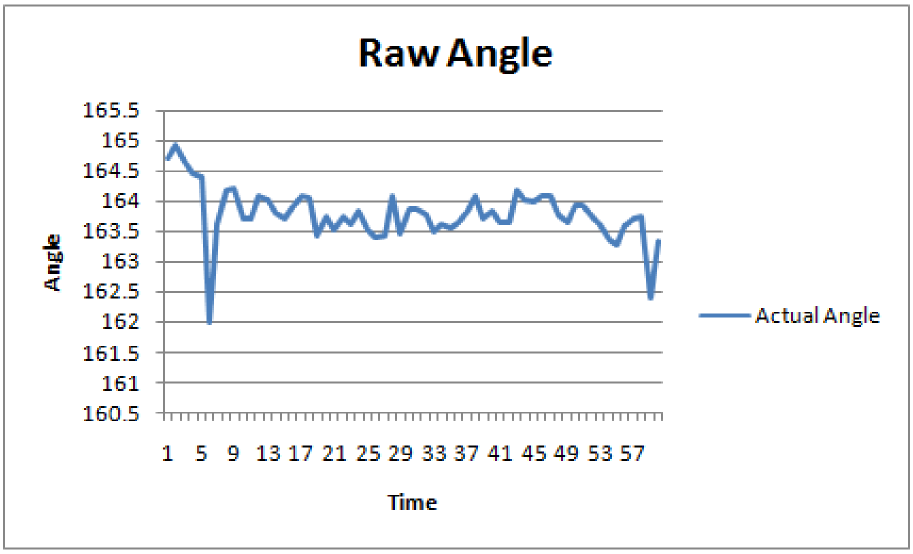


7.3 Robot tilted towards the back

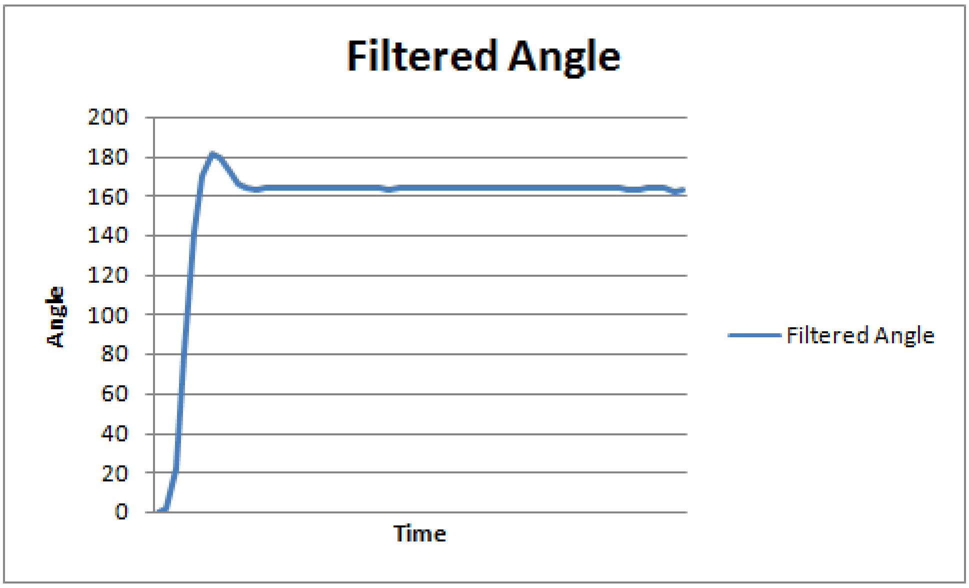
Actual Angle	Gyro Rate	Filtered Angle	Error	Proportio-nal	Integ-ral	Deriva-tive	output	Direc-tion	Speed
164.71	-2.09	0.01	-0.01	-0.2	0	0.61	0.82	r	0
164.94	-3.37	2.18	-2.18	-65.31	0	195.32	255	r	255
164.66	-3.95	22.7	-22.7	-680.85	0	1846.62	255	r	255

164.48	-3.13	81.17	-81.17	-2435.03	0	5262.54	255	r	0
164.4	-3.87	141.2	-141.2	-4236	0	5402.91	255	r	0
161.98	-9.41	170.24	-170.24	-5107.35	0	2614.03	255	r	0
163.62	-3.8	181.14	-181.14	-5434.17	0	980.47	255	r	0
164.2	-3.65	178.73	-178.73	-5361.76	0	-217.23	255	r	0
164.23	-3.48	172.18	-172.18	-5165.44	0	-588.96	255	r	0
163.7	-3.56	166.4	-166.4	-4992	0	-520.33	255	r	0
163.72	-3.18	163.65	-163.65	-4909.46	0	-247.61	255	r	0
164.1	-3.58	163.11	-163.11	-4893.2	0	-48.79	255	r	0
164.02	-3.02	163.55	-163.55	-4906.36	0	39.48	255	r	0
163.81	-3.21	163.69	-163.69	-4910.74	0	13.12	255	r	0
163.71	-3.4	163.66	-163.66	-4909.95	0	-2.37	255	r	0
163.91	-2.33	164.09	-164.09	-4922.69	0	38.21	255	r	0
164.09	-4.81	163.66	-163.66	-4909.71	0	-38.92	255	r	0
164.05	-3.63	164.03	-164.03	-4921.01	0	33.91	255	r	0
163.42	-3.72	163.61	-163.61	-4908.27	0	-38.22	255	r	0
163.74	-2.73	163.87	-163.87	-4915.98	0	23.13	255	r	0
163.53	-2.87	163.65	-163.65	-4909.49	0	-19.47	255	r	0
163.75	-3.63	163.56	-163.56	-4906.83	0	-8	255	r	0
163.61	-3.42	163.59	-163.59	-4907.66	0	2.49	255	r	0
163.84	-3.4	163.8	-163.8	-4914.06	0	19.2	255	r	0
163.51	-3.16	163.63	-163.63	-4909.05	0	-15.03	255	r	0
163.4	-3.13	163.44	-163.44	-4903.16	0	-17.67	255	r	0
163.44	-3.46	163.36	-163.36	-4900.77	0	-7.16	255	r	0
164.09	-3.17	163.99	-163.99	-4919.61	0	56.53	255	r	0
163.45	-3.43	163.58	-163.58	-4907.46	0	-36.44	255	r	0

- Graph depicting the raw data obtained from the sensors (backward tilt)



- Graph depicting the raw data obtained from the sensors(backward tilt)



REFERENCES

REFERENCES

[1] R. Hollis, “BallBots,” *Scientific American*, October 2006. Retrieved February 4, 2009 from the World Wide Web: http://www.sciam.com/article.cfm?id=ballbots

[2] R.C Ooi, “Balancing a Two-Wheeled Autonomous Robot,” 2003. Retrieved January 18, 2009 from the World Wide Web: http://robotics.ee.uwa.edu.au/theses/2003-Balance-Ooi.pdf>.

[3] D. Simon, “Kalman Filtering.” *Embedded Systems Programming*, 2001. Retrieved January 18, 2009 from the World Wide Web: http://www.embedded.com/9900168?_requestid=8285

[4] G. Welch and G. Bishop, “Kalman Filter.” *An Introduction to the Kalman Filter*, 2007. Retrieved February 16, 2009 from the World Wide Web: http://www.cs.unc.edu/~welch/kalman/kalmanIntro.html

[5] V.J. VanDoren, “PID: Still the One,” *Control Engineering*, October 2003. Retrieved January 19, 2009 from the World Wide Web: http://www.controleng.com/article/CA325983.html

[6] Junfeng Wu, Yuxin Liang and Zhe Wang, “A Robust Control Method of Two-Wheeled Self Balancing Robot”, 2011. Published in “The 6th International Forum on Strategic Technology”.

[7] Junfeng Wu and Wanying Zhang, “Design of Fuzzy Logic Controller for Two-wheeled Self-balancing Robot”, 2011. Published in “The 6th International Forum on Strategic Technology”.

[8] Xiaogang Ruan and Jing Chen, “H∞ Robust Control of Self-Balancing Two-Wheeled Robot” published in in 2010 at Proceedings of the 8th World Congress on Intelligent Control and Automation.

[9] Y.-S. Ha and S. Yuta, “Trajectory tracking control for navigation of the inverse pendulum type self-contained mobile robot,” Robot. Autonom. Syst., vol. 17, pp. 65–80, 1996

[10] A Blankespoor, R Roemer. Experimental verification of the dynamic model for a quarter size self-balancing wheelchair. [C]// Proc. ACC. Boston: ACC, 2004: 488-492.

[11] Wu Junfeng and Zhang Wanying, “Research on Control Method of Two-wheeled Self balancing Robot” in 2011 at “Fourth International Conference on Intelligent Computation Technology and Automation”.

[12] Brown, R. G. and P. Y. C. Hwang. 1992. Introduction to Random Signals and Applied Kalman Filtering, Second Edition, John Wiley & Sons, Inc.

APPENDIX

L298

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

Multiwatt15

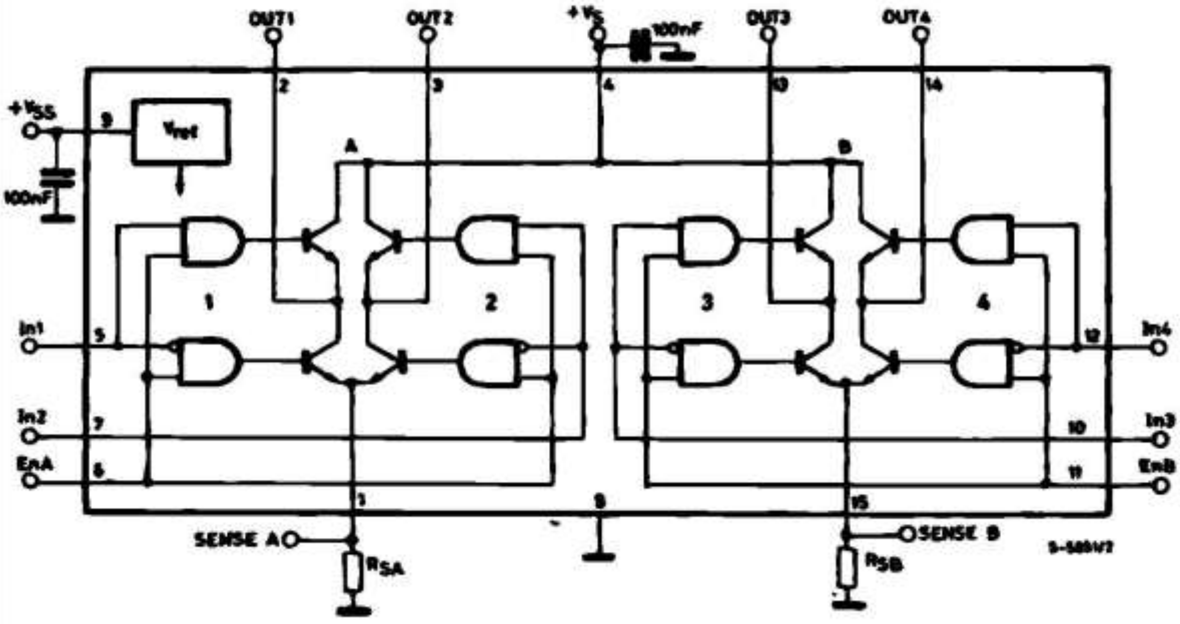
PowerSO20

ORDERING NUMBERS : L298N (Multiwatt Vert.)
L298HN (Multiwatt Horiz.)
L298P (PowerSO20)

DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver de-signed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the in-put signals. The emitters of the lower transistors of each bridge are connected together and the corre-sponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

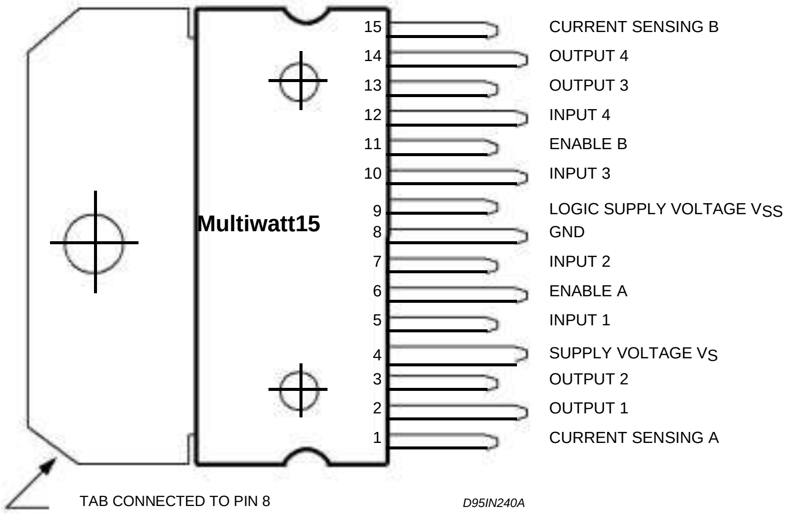
BLOCK DIAGRAM

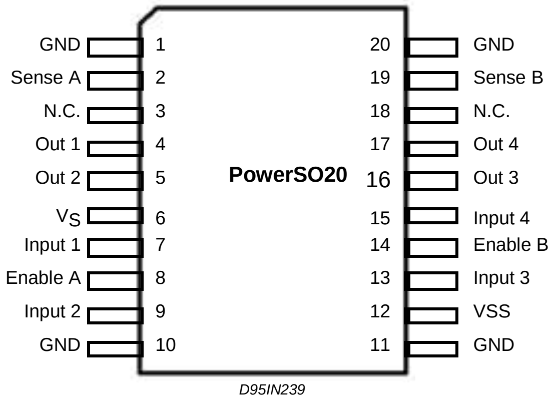


ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_I,V_{en}	Input and Enable Voltage	–0.3 to 7	V
I_O	Peak Output Current (each Channel) – Non Repetitive (t = 100μs) –Repetitive (80% on –20% off; t_{on} = 10ms) –DC Operation	 3 2.5 2	 A A A
V_{sens}	Sensing Voltage	–1 to 2.3	V
P_{tot}	Total Power Dissipation (T_{case} = 75°C)	25	W
T_{op}	Junction Operating Temperature	–25 to 130	°C
T_{stg}, T_j	Storage and Junction Temperature	–40 to 150	°C

PIN CONNECTIONS (top view)





THERMAL DATA

Symbol	Parameter		PowerSO20	Multiwatt15	Unit
$R_{th\ j\text{-}case}$	Thermal Resistance Junction-case	Max.	–	3	°C/W
$R_{th\ j\text{-}amb}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35	°C/W

(*) Mounted on aluminum substrate

PIN FUNCTIONS (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V_S	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	VSS	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
–	3;18	N.C.	Not Connected

ELECTRICAL CHARACTERISTICS (V_S = 42V; V_{SS} = 5V, T_j = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_S	Supply Voltage (pin 4)	Operative Condition	V_{IH} +2.5		46	V
V_{SS}	Logic Supply Voltage (pin 9)		4.5	5	7	V
I_S	Quiescent Supply Current (pin 4)	V_{en} = H; I_L = 0 V_i = L V_i = H		13 50	22 70	mA mA
		V_{en} = L V_i = X			4	mA
I_{SS}	Quiescent Current from V_{SS} (pin 9)	V_{en} = H; I_L = 0 V_i = L V_i = H		24 7	36 12	mA mA
		V_{en} = L V_i = X			6	mA
V_{iL}	Input Low Voltage (pins 5, 7, 10, 12)		–0.3		1.5	V
V_{iH}	Input High Voltage (pins 5, 7, 10, 12)		2.3		VSS	V
I_{iL}	Low Voltage Input Current (pins 5, 7, 10, 12)	V_i = L			–10	µA
I_{iH}	High Voltage Input Current (pins 5, 7, 10, 12)	Vi = H ≤ V_{SS} –0.6V		30	100	µA
V_{en} = L	Enable Low Voltage (pins 6, 11)		–0.3		1.5	V
V_{en} = H	Enable High Voltage (pins 6, 11)		2.3		V_{SS}	V
I_{en} = L	Low Voltage Enable Current (pins 6, 11)	V_{en} = L			–10	µA
I_{en} = H	High Voltage Enable Current (pins 6, 11)	V_{en} = H ≤ V_{SS} –0.6V		30	100	µA
$V_{CEsat\ (H)}$	Source Saturation Voltage	I_L = 1A I_L = 2A	0.95	1.35 2	1.7 2.7	V V
$V_{CEsat\ (L)}$	Sink Saturation Voltage	I_L = 1A (5) I_L = 2A (5)	0.85	1.2 1.7	1.6 2.3	V V
V_{CEsat}	Total Drop	I_L = 1A (5) I_L = 2A (5)	1.80		3.2 4.9	V V
V_{sens}	Sensing Voltage (pins 1, 15)		–1 (1)		2	V

ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
T_1 (V_i)	Source Current Turn-off Delay	0.5 V_i to 0.9 I_L (2); (4)		1.5		µs
T_2 (V_i)	Source Current Fall Time	0.9 I_L to 0.1 I_L (2); (4)		0.2		µs
T_3 (V_i)	Source Current Turn-on Delay	0.5 V_i to 0.1 I_L (2); (4)		2		µs
T_4 (V_i)	Source Current Rise Time	0.1 I_L to 0.9 I_L (2); (4)		0.7		µs
T_5 (V_i)	Sink Current Turn-off Delay	0.5 V_i to 0.9 I_L (3); (4)		0.7		µs
T_6 (V_i)	Sink Current Fall Time	0.9 I_L to 0.1 I_L (3); (4)		0.25		µs
T_7 (V_i)	Sink Current Turn-on Delay	0.5 V_i to 0.9 I_L (3); (4)		1.6		µs
T_8 (V_i)	Sink Current Rise Time	0.1 I_L to 0.9 I_L (3); (4)		0.2		µs
fc (V_i)	Commutation Frequency	I_L = 2A		25	40	KHz
T_1 (V_{en})	Source Current Turn-off Delay	0.5 V_{en} to 0.9 I_L (2); (4)		3		µs
T_2 (V_{en})	Source Current Fall Time	0.9 I_L to 0.1 I_L (2); (4)		1		µs
T_3 (V_{en})	Source Current Turn-on Delay	0.5 V_{en} to 0.1 I_L (2); (4)		0.3		µs
T_4 (V_{en})	Source Current Rise Time	0.1 I_L to 0.9 I_L (2); (4)		0.4		µs
T_5 (V_{en})	Sink Current Turn-off Delay	0.5 V_{en} to 0.9 I_L (3); (4)		2.2		µs
T_6 (V_{en})	Sink Current Fall Time	0.9 I_L to 0.1 I_L (3); (4)		0.35		µs
T_7 (V_{en})	Sink Current Turn-on Delay	0.5 V_{en} to 0.9 I_L (3); (4)		0.25		µs
T_8 (V_{en})	Sink Current Rise Time	0.1 I_L to 0.9 I_L (3); (4)		0.1		µs

1) 1)Sensing voltage can be –1 V for t ≤ 50 µsec; in steady state V_{sens} min ≥ – 0.5 V.
2) See fig. 2.
3) See fig. 4.
4) The load must be a pure resistor.

Figure 1 : Typical Saturation Voltage vs. Output Current.

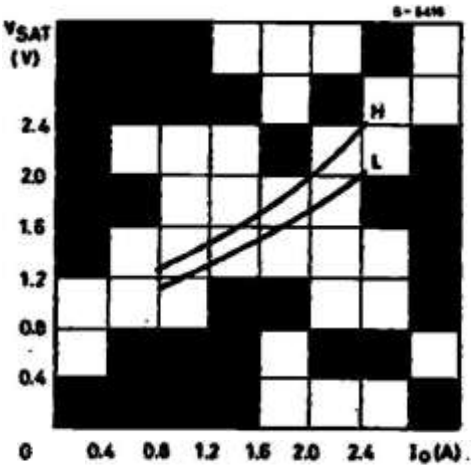
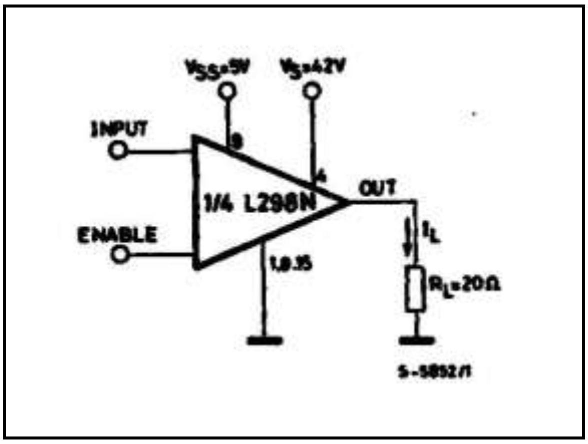


Figure 2 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
For ENABLE Switching, set IN = H

Figure 3 : Source Current Delay Times vs. Input or Enable Switching.

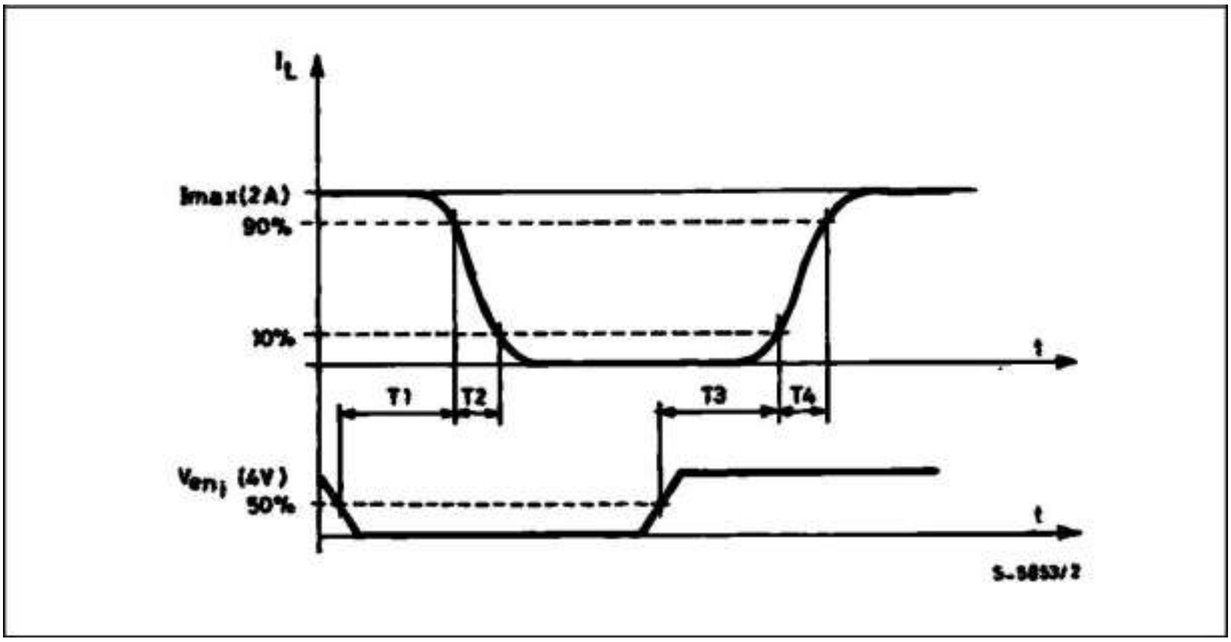
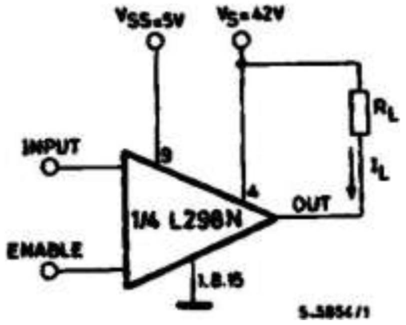


Figure 4 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
For ENABLE Switching, set IN = L

Figure 5 : Sink Current Delay Times vs. Input 0 V Enable Switching.

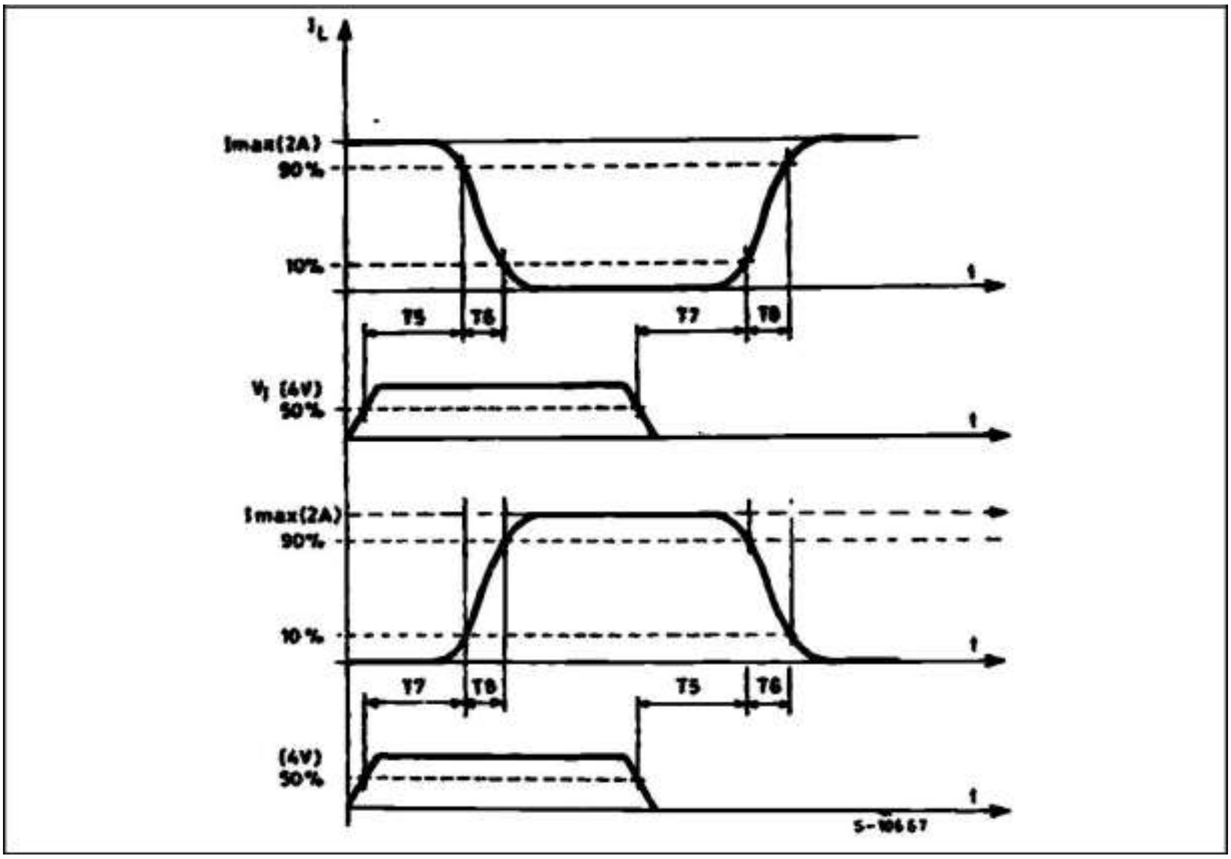
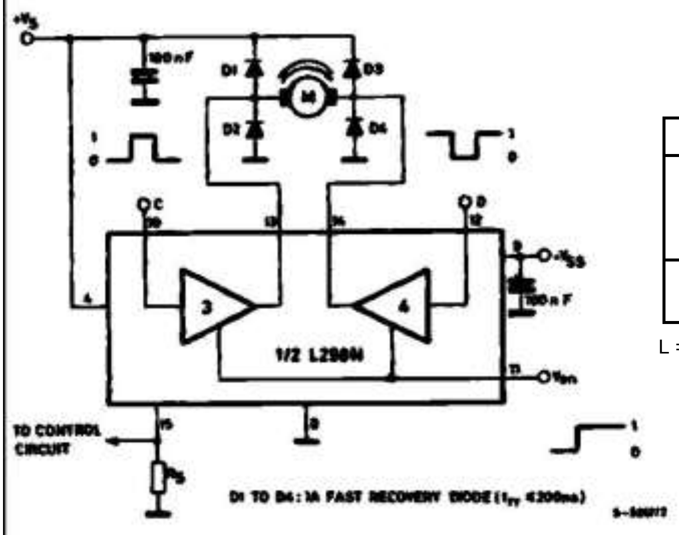


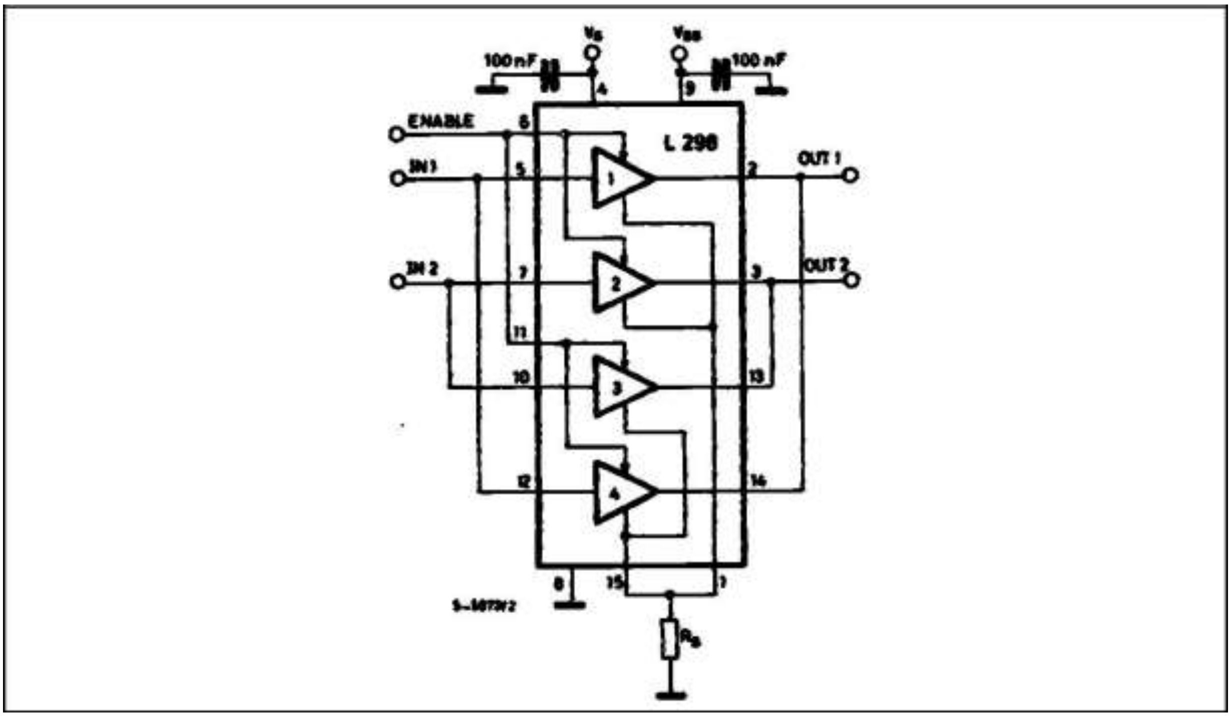
Figure 6 : Bidirectional DC Motor Control.



Inputs		Function
V_{en} = H	C = H ; D = L	Forward
	C = L ; D = H	Reverse
	C = D	Fast Motor Stop
V_{en} = L	C = X ; D = X	Free Running Motor Stop

L = Low H = High X = Don't care

Figure 7 : For higher currents, outputs can be paralleled. Take care to parallel channel 1 with channel 4 and channel 2 with channel 3.



APPLICATION INFORMATION (Refer to the block diagram)

1.1. POWER OUTPUT STAGE

The L298 integrates two power output stages (A ; B). The power output stage is a bridge configuration and its outputs can drive an inductive load in common or differenzial mode, depending on the state of the inputs. The current that flows through the load comes out from the bridge at the sense output : an external resistor (R_{SA} ; R_{SB}.) allows to detect the intensity of this current.

1.2. INPUT STAGE

Each bridge is driven by means of four gates the input of which are In1 ; In2 ; EnA and In3 ; In4 ; EnB. The In inputs set the bridge state when The En input is high ; a low state of the En input inhibits the bridge. All the inputs are TTL compatible.

2. SUGGESTIONS

A non inductive capacitor, usually of 100 nF, must be foreseen between both Vs and Vss, to ground, as near as possible to GND pin. When the large ca-pacitor of the power supply is too far from the IC, a second smaller one must be foreseen near the L298.

The sense resistor, not of a wire wound type, must be grounded near the negative pole of Vs that must be near the GND pin of the I.C.

Each input must be connected to the source of the driving signals by means of a very short path.

Turn-On and Turn-Off : Before to Turn-ON the Sup-ply Voltage and before to Turn it OFF, the Enable in-put must be driven to the Low state.

3. APPLICATIONS

Fig 6 shows a bidirectional DC motor control Sche-matic Diagram for which only one bridge is needed. The external bridge of diodes D1 to D4 is made by four fast recovery elements (trr ≤ 200 nsec) that must be chosen of a VF as low as possible at the worst case of the load current.

The sense output voltage can be used to control the current amplitude by chopping the inputs, or to pro-vide overcurrent protection by switching low the en-able input.

The brake function (Fast motor stop) requires that the Absolute Maximum Rating of 2 Amps must never be overcome.

When the repetitive peak current needed from the load is higher than 2 Amps, a paralleled configura-tion can be chosen (See Fig.7).

An external bridge of diodes are required when inductive loads are driven and when the inputs of the IC are chopped ; Shottky diodes would be preferred.

This solution can drive until 3 Amps In DC operation and until 3.5 Amps of a repetitive peak current.

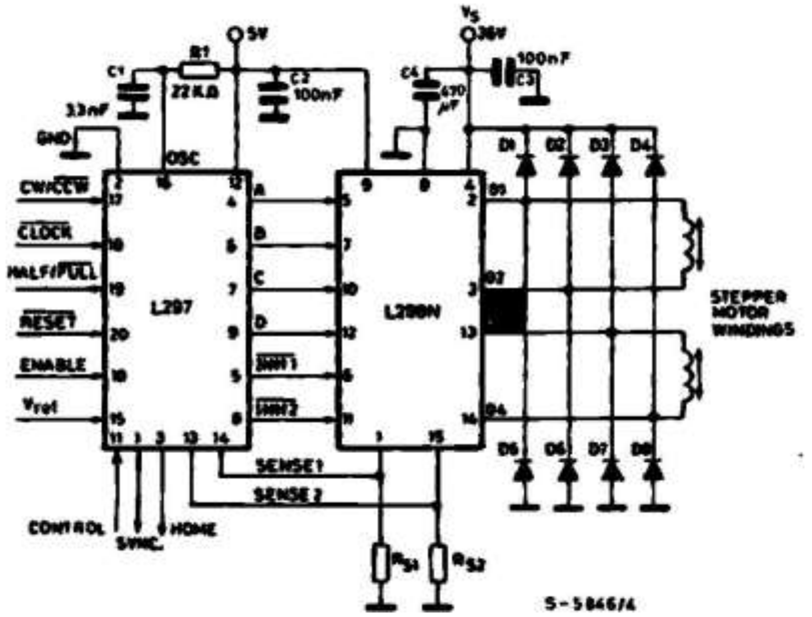
On Fig 8 it is shown the driving of a two phase bipolar stepper motor ; the needed signals to drive the in-puts of the L298 are generated, in this example, from the IC L297.

Fig 9 shows an example of P.C.B. designed for the application of Fig 8.

Fig 10 shows a second two phase bipolar stepper motor control circuit where the current is controlled by the I.C. L6506.

Figure 8 : Two Phase Bipolar Stepper Motor Circuit.

This circuit drives bipolar stepper motors with winding currents up to 2 A. The diodes are fast 2 A types.



$R_{S1} = R_{S2} = 0.5\ \Omega$

D1 to D8 = 2 A Fast diodes $\begin{cases} V_F \leq 1.2\text{ V @ I} = 2\text{ A} \\ trr \leq 200\text{ ns} \end{cases}$

Figure 9 : Suggested Printed Circuit Board Layout for the Circuit of fig. 8 (1:1 scale).

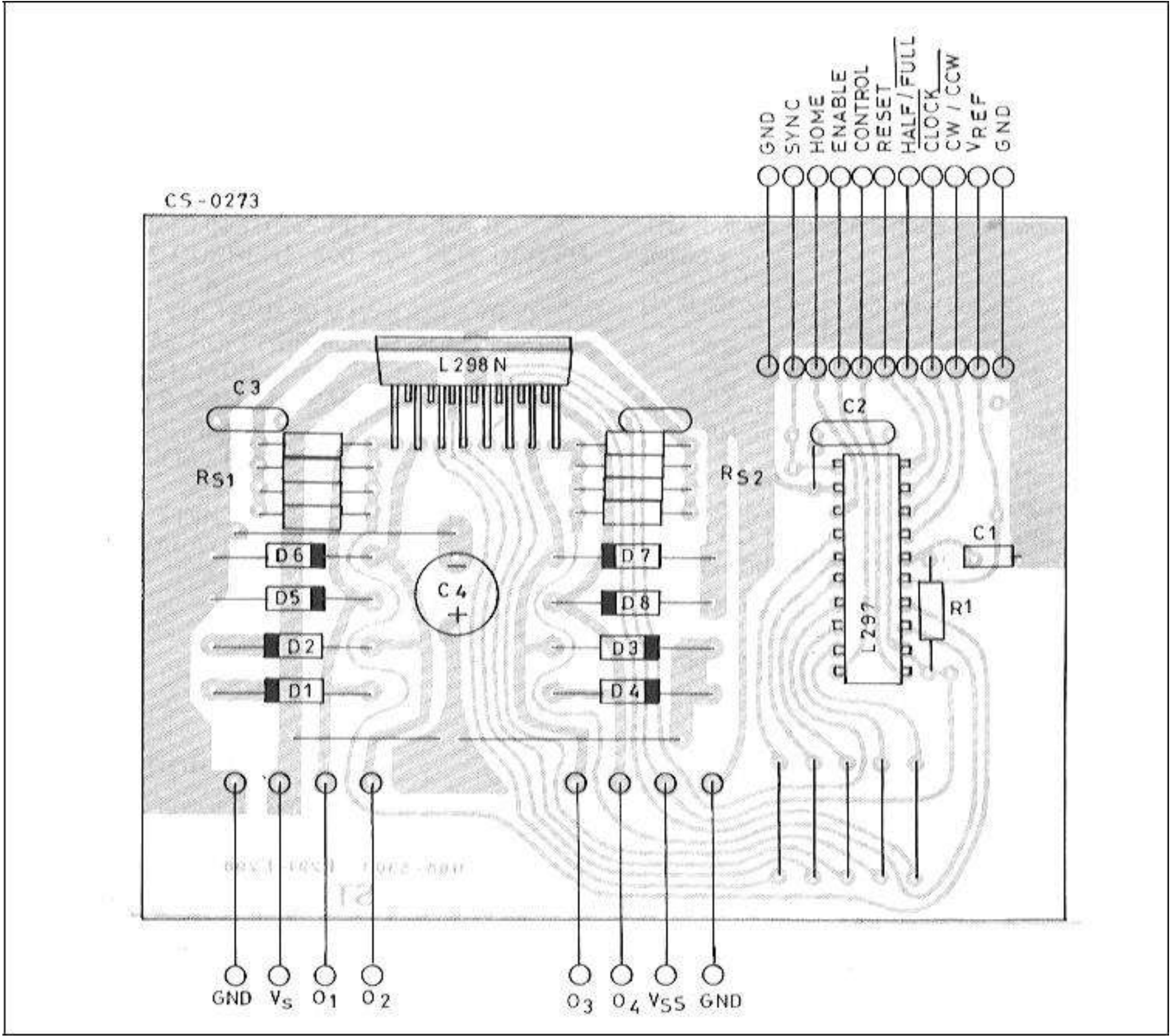
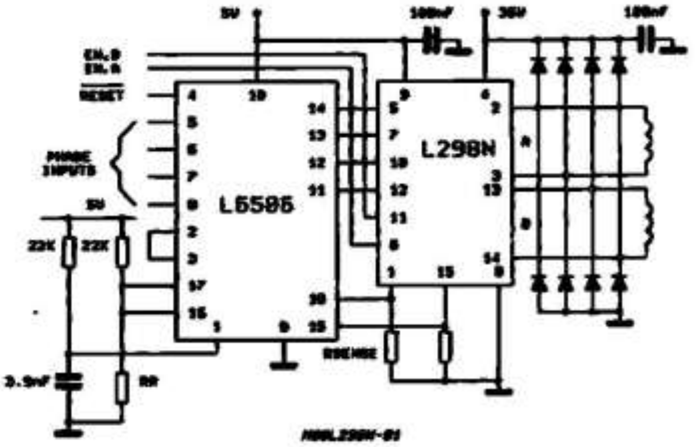


Figure 10 : Two Phase Bipolar Stepper Motor Control Circuit by Using the Current Controller L6506.

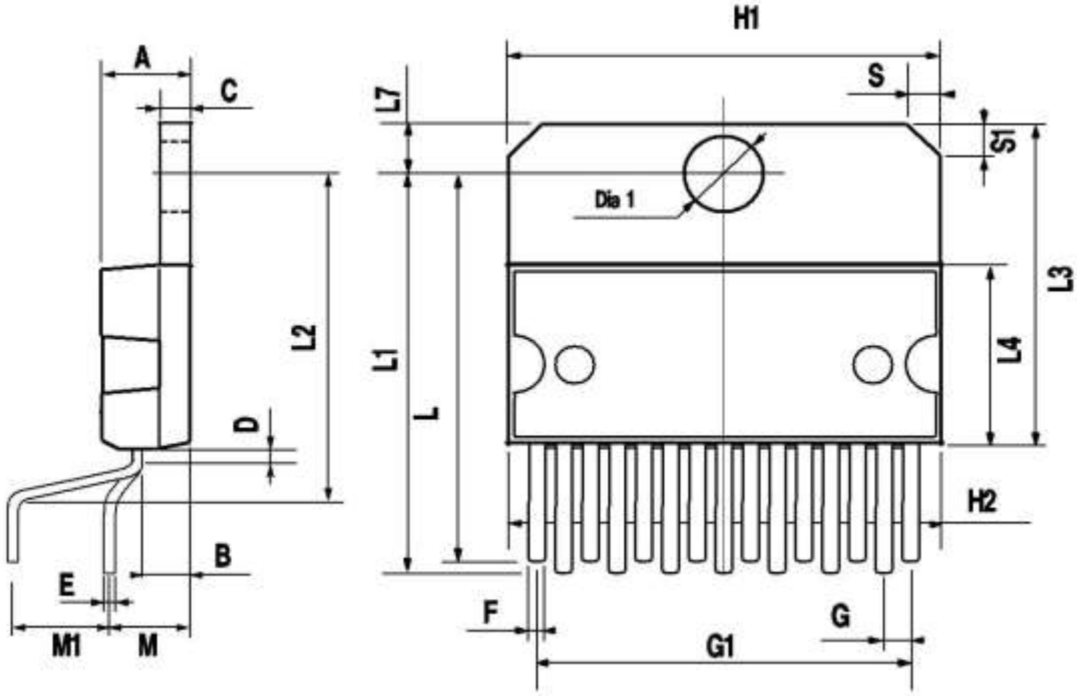


R_R and R_{sense} depend from the load current

OUTLINE AND MECHANICAL DATA

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
D		1			0.039	
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.02	1.27	1.52	0.040	0.050	0.060
G1	17.53	17.78	18.03	0.690	0.700	0.710
H1	19.6			0.772		
H2			20.2			0.795
L	21.9	22.2	22.5	0.862	0.874	0.886
L1	21.7	22.1	22.5	0.854	0.870	0.886
L2	17.65		18.1	0.695		0.713
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L7	2.65		2.9	0.104		0.114
M	4.25	4.55	4.85	0.167	0.179	0.191
M1	4.63	5.08	5.53	0.182	0.200	0.218
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

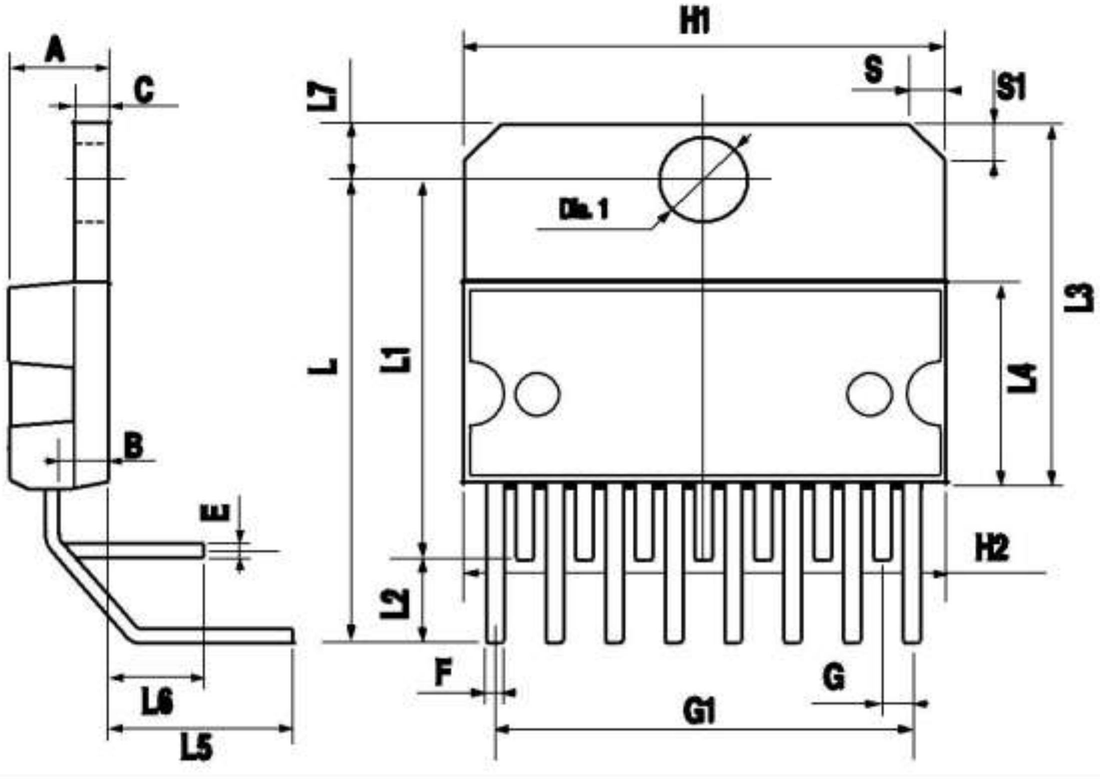
Multiwatt15 V



DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.14	1.27	1.4	0.045	0.050	0.055
G1	17.57	17.78	17.91	0.692	0.700	0.705
H1	19.6			0.772		
H2			20.2			0.795
L		20.57			0.810	
L1		18.03			0.710	
L2		2.54			0.100	
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L5		5.28			0.208	
L6		2.38			0.094	
L7	2.65		2.9	0.104		0.114
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

OUTLINE AND MECHANICAL DATA

Multiwatt15 H



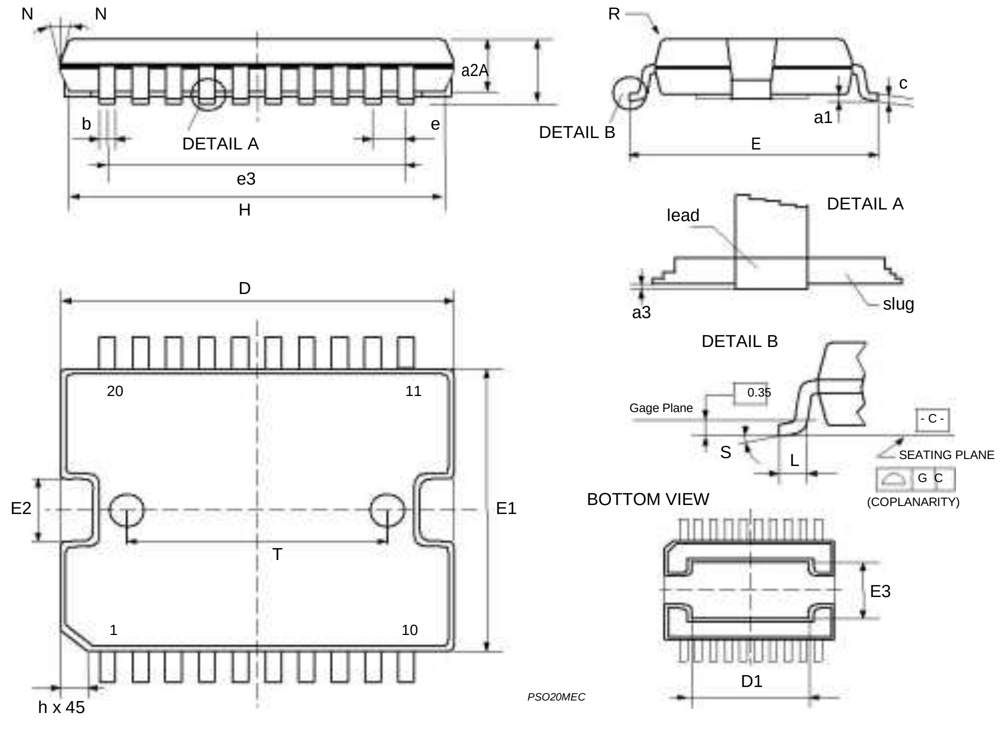
DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			3.6			0.142
a1	0.1		0.3	0.004		0.012
a2			3.3			0.130
a3	0		0.1	0.000		0.004
b	0.4		0.53	0.016		0.021
c	0.23		0.32	0.009		0.013
D (1)	15.8		16	0.622		0.630
D1	9.4		9.8	0.370		0.386
E	13.9		14.5	0.547		0.570
e		1.27			0.050	
e3		11.43			0.450	
E1 (1)	10.9		11.1	0.429		0.437
E2			2.9			0.114
E3	5.8		6.2	0.228		0.244
G	0		0.1	0.000		0.004
H	15.5		15.9	0.610		0.626
h			1.1			0.043
L	0.8		1.1	0.031		0.043
N	10˚ (max.)					
S	8˚ (max.)					
T		10			0.394	

(1) "D and F" do not include mold flash or protrusions.
- Mold flash or protrusions shall not exceed 0.15 mm (0.006").
- Critical dimensions: "E", "G" and "a3"

OUTLINE AND MECHANICAL DATA

JEDEC MO-166

PowerSO20



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 I^2C 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 I^2C 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 ±2000°/sec (dps) and a user-programmable accelerometer full-scale range of ±2*g*, ±4*g*, ±8*g*, and ±16*g*.

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 I^2C 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 ±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 10,000*g* 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 I^2C interface. The VLOGIC voltage may be 1.8V±5% or VDD.

The MPU-6000 and MPU-6050 are identical, except that the MPU-6050 supports the I^2C serial interface only, and has a separate VLOGIC reference pin. The MPU-6000 supports both I^2C 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^2C, SPI	I^2C
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 ±2000°/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: 5µA
- 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 ±2*g*, ±4*g*, ±8*g* and ±16*g*
- Integrated 16-bit ADCs enable simultaneous sampling of accelerometers while requiring no external multiplexer
- Accelerometer normal operating current: 500µA
- Low power accelerometer mode current: 10µA at 1.25Hz, 20µA at 5Hz, 60µA at 20Hz, 110µA 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 I^2C 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 I^2C 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 I^2C 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 ±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		º/s	
	FS_SEL=1		±500		º/s	
	FS_SEL=2		±1000		º/s	
	FS_SEL=3		±2000		º/s	
Gyroscope ADC Word Length			16		bits	
Sensitivity Scale Factor	FS_SEL=0		131		LSB/(º/s)	
	FS_SEL=1		65.5		LSB/(º/s)	
	FS_SEL=2		32.8		LSB/(º/s)	
	FS_SEL=3		16.4		LSB/(º/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		º/s	
ZRO Variation Over Temperature	-40°C to +85°C		±20		º/s	
Power-Supply Sensitivity (1-10Hz)	Sine wave, 100mVpp; VDD=2.5V		0.2		º/s	
Power-Supply Sensitivity (10 - 250Hz)	Sine wave, 100mVpp; VDD=2.5V		0.2		º/s	
Power-Supply Sensitivity (250Hz - 100kHz)	Sine wave, 100mVpp; VDD=2.5V		4		º/s	
Linear Acceleration Sensitivity	Static		0.1		º/s/*g*	
SELF-TEST RESPONSE						
Relative	Change from factory trim	-14		14	%	1
GYROSCOPE NOISE PERFORMANCE	**FS_SEL=0**					
Total RMS Noise	DLPFCFG=2 (100Hz)		0.05		º/s-rms	
Low-frequency RMS noise	Bandwidth 1Hz to10Hz		0.033		º/s-rms	
Rate Noise Spectral Density	At 10Hz		0.005		º/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	**DLPFCFG=0**					
ZRO Settling (from power-on)	to ±1º/s of Final		30		ms	

1. 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		m*g*	1
	Z axis		±80		m*g*	
Zero-G Level Change vs. Temperature	X and Y axes, 0°C to +70°C		±35			
	Z axis, 0°C to +70°C		±60		m*g*	
SELF TEST RESPONSE						
Relative	Change from factory trim	-14		14	%	2
NOISE PERFORMANCE						
Power Spectral Density	@10Hz, AFS_SEL=0 & ODR=1kHz		400		µ*g*/ √ Hz	
LOW PASS FILTER RESPONSE						
	Programmable Range	5		260	Hz	
OUTPUT DATA RATE						
	Programmable Range	4		1,000	Hz	
INTELLIGENCE FUNCTION INCREMENT			32		m*g*/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/ºC	
Temperature Offset	35^{o}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	MPU-6050 only					
Voltage Range	VLOGIC must be ≤VDD at all times	1.71		VDD	V	
Power Supply Ramp Rate	Monotonic ramp. Ramp rate is 10% to 90% of the final value			3	ms	
Normal Operating Current			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, T_A = 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	
I^2C Operating Frequency	All registers, Fast-mode			400	kHz	
	All registers, Standard-mode			100	kHz	
I^2C 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, T_A = 25°C

Parameters	Conditions	Typical	Units	Notes
Primary I^2C 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	
VIL, LOW Level Input Voltage	MPU-6050	-0.5V to 0.3*VLOGIC	V	
VIH, HIGH-Level Input Voltage	MPU-6050	0.7*VLOGIC to VLOGIC + 0.5V	V	
Vhys, 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 I^2C 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

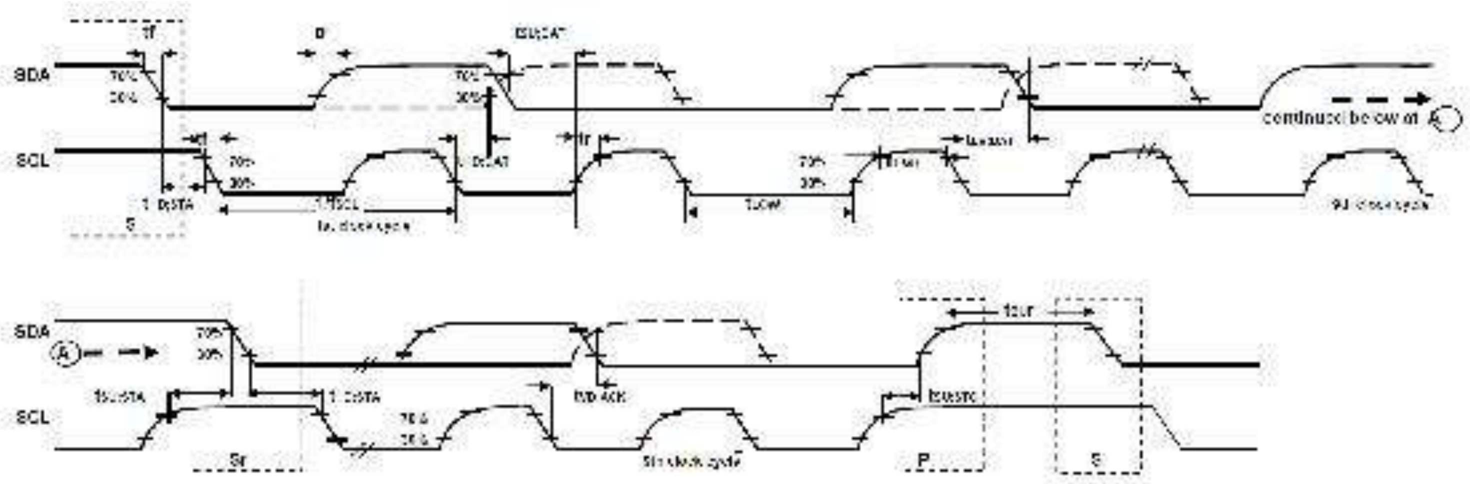
Parameters	Conditions	Min	Typical	Max	Units	Notes
INTERNAL CLOCK SOURCE	**CLK_SEL=0,1,2,3**					
Gyroscope Sample Rate, Fast	DLPFCFG=0 SAMPLERATEDIV = 0		8		kHz	
Gyroscope Sample Rate, Slow	DLPFCFG=1,2,3,4,5, or 6 SAMPLERATEDIV = 0		1		kHz	
Accelerometer Sample Rate			1		kHz	
Clock Frequency Initial Tolerance	CLK_SEL=0, 25°C	-5		+5	%	
	CLK_SEL=1,2,3; 25°C	-1		+1	%	
Frequency Variation over Temperature	CLK_SEL=0		-15 to +10		%	
	CLK_SEL=1,2,3		±1		%	
PLL Settling Time	CLK_SEL=1,2,3		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

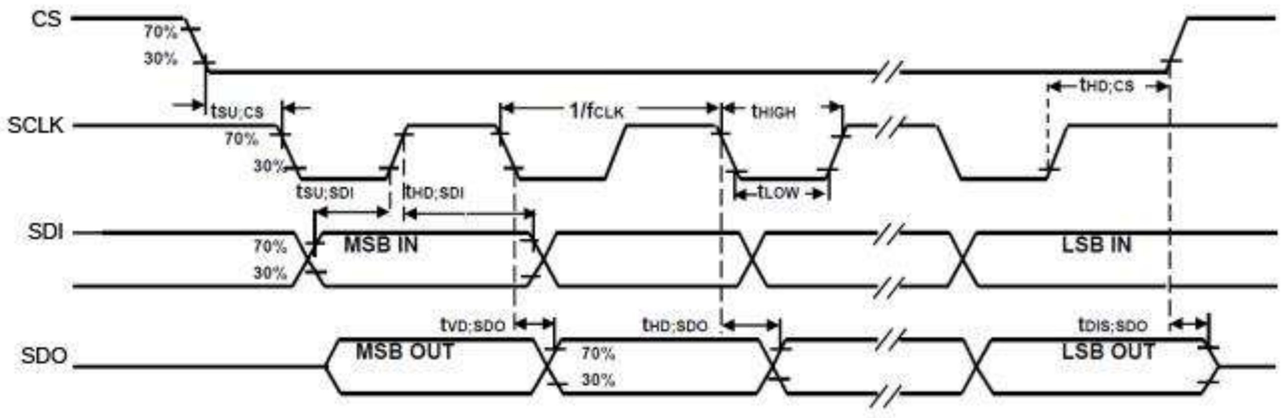


I²C Bus Timing Diagram

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

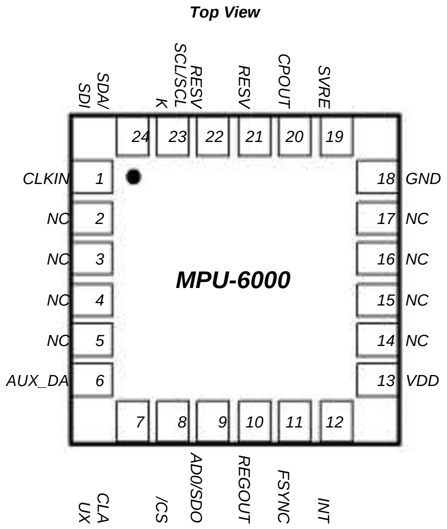
Stress above those listed as “Absolute Maximum Ratings” may cause permanent damage to the device. These are stress ratings only and functional operation of the device at these conditions is not implied. Exposure to the absolute maximum ratings conditions for extended periods may affect device reliability.

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

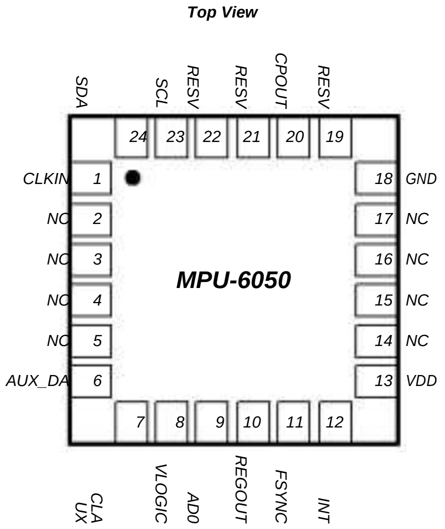
7 Applications Information

7.1 Pin Out and Signal Description

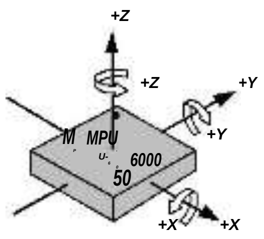
Pin Number	MPU-6000	MPU-6050	Pin Name	Pin Description
1	Y	Y	CLKIN	Optional external reference clock input. Connect to GND if unused.
6	Y	Y	AUX_DA	I^2C master serial data, for connecting to external sensors
7	Y	Y	AUX_CL	I^2C Master serial clock, for connecting to external sensors
8	Y		/CS	SPI chip select (0=SPI mode)
8		Y	VLOGIC	Digital I/O supply voltage
9	Y		AD0 / SDO	I^2C Slave Address LSB (AD0); SPI serial data output (SDO)
9		Y	AD0	I^2C 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^2C serial clock (SCL); SPI serial clock (SCLK)
23		Y	SCL	I^2C serial clock (SCL)
24	Y		SDA / SDI	I^2C serial data (SDA); SPI serial data input (SDI)
24		Y	SDA	I^2C 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

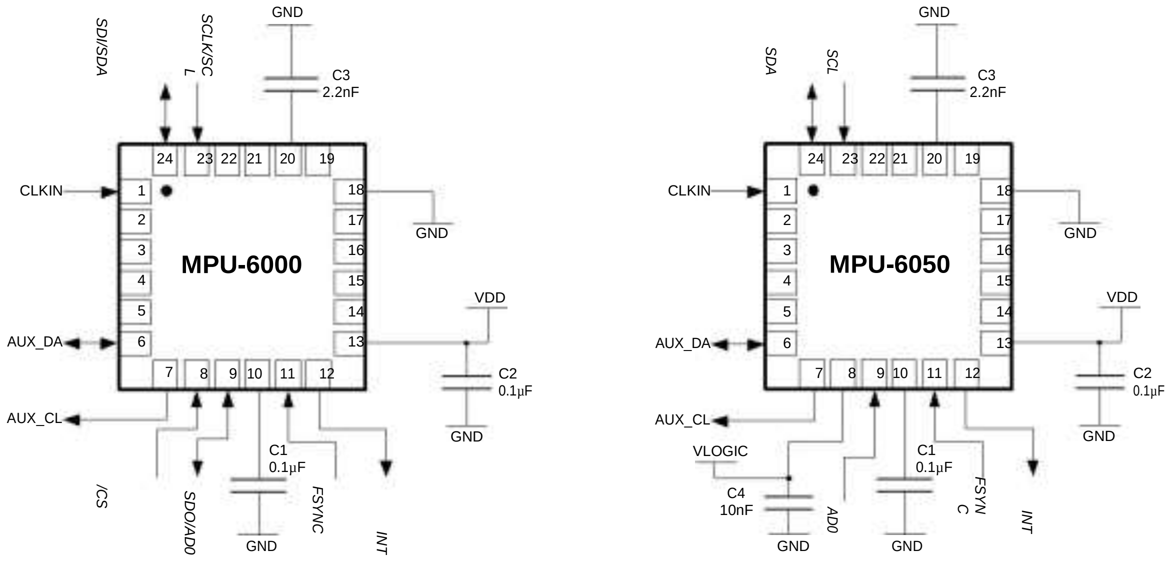


QFN Package
24-pin, 4mm x 4mm x 0.9mm



Orientation of Axes of Sensitivity and Polarity of Rotation

7.2 Typical Operating Circuit



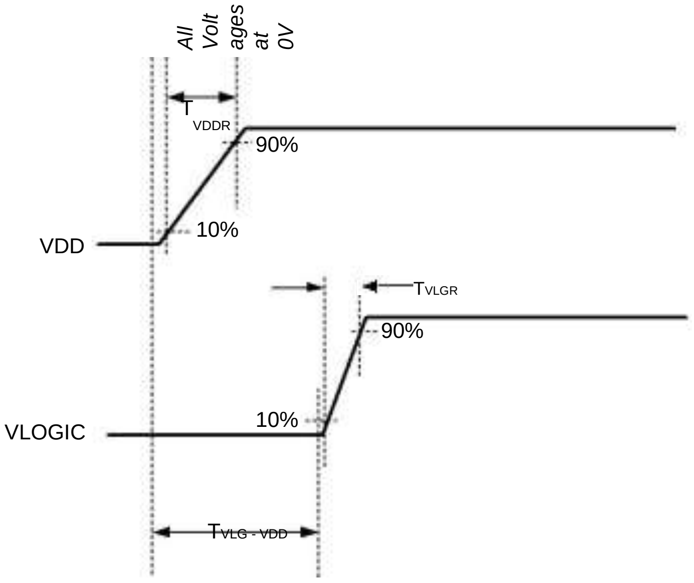
Typical Operating Circuits

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.

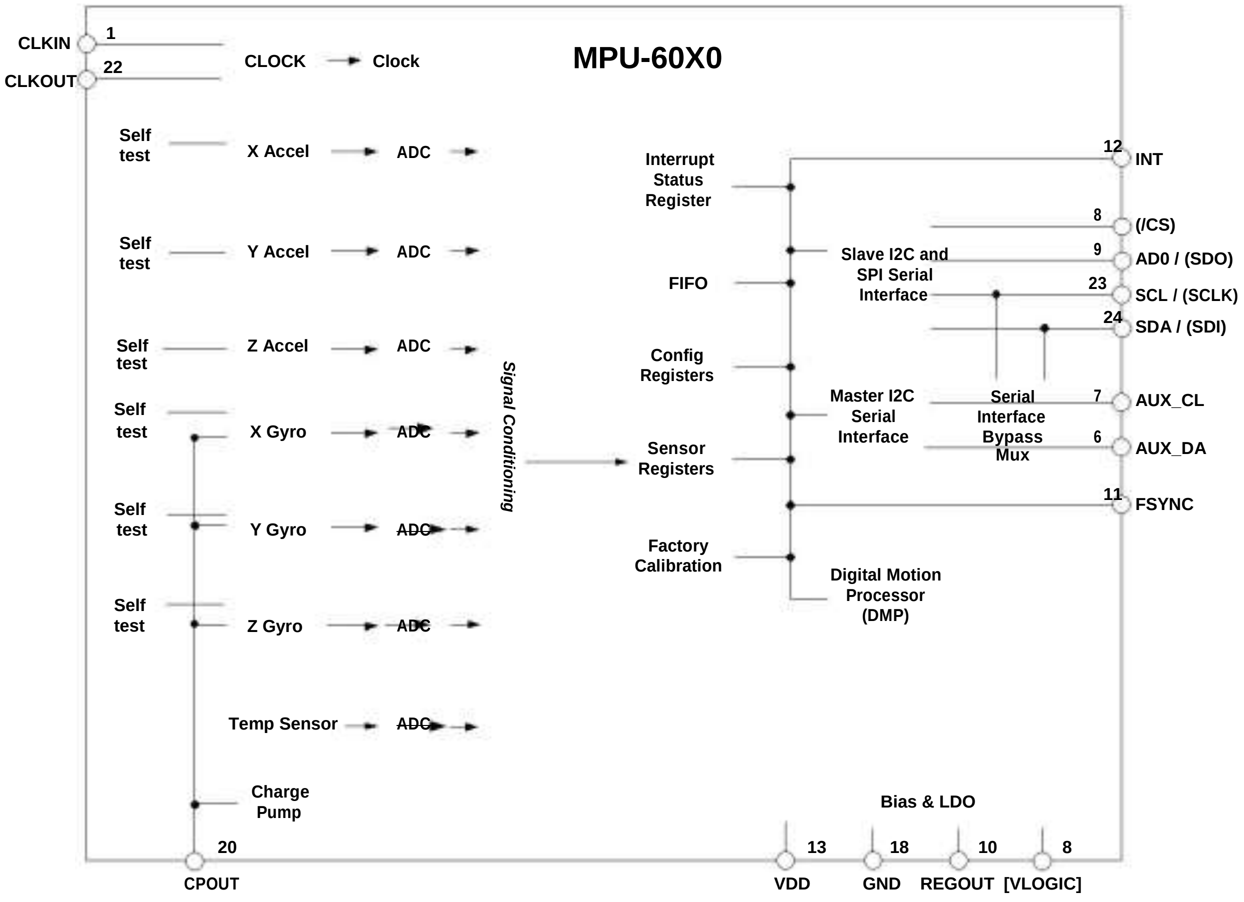
7.4 Recommended Power-on Procedure



Power-Up Sequencing

1. VLOGIC amplitude must always be ≤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 ≤100ms
4. T_{VLGR} is VLOGIC rise time: Time for VLOGIC to rise from 10% to 90% of its final value
5. T_{VLGR} is ≤3ms
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 I^{2}C and SPI (MPU-6000 only) serial communications interfaces
- Auxiliary I^{2}C 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 0*g* on the X- and Y-axes and +1*g* 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 ±2*g*, ±4*g*, ±8*g*, or ±16*g*.

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 I^{2}C and SPI Serial Communications Interfaces

The MPU-60X0 communicates to a system processor using either a SPI (MPU-6000 only) or an I^{2}C serial interface. The MPU-60X0 always acts as a slave when communicating to the system processor. The LSB of the of the I^{2}C 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 I^{2}C Serial Interface

The MPU-60X0 has an auxiliary I^{2}C bus for communicating to an off-chip 3-Axis digital output magnetometer or other sensors. This bus has two operating modes:

- I^{2}C Master Mode: The MPU-60X0 acts as a master to any external sensors connected to the auxiliary I^{2}C bus
- Pass-Through Mode: The MPU-60X0 directly connects the primary and auxiliary I^{2}C buses together, allowing the system processor to directly communicate with any external sensors.

Auxiliary I^{2}C Bus Modes of Operation:

- I^{2}C 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 I^{2}C 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 I^{2}C 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 I^{2}C bus pins (AUX_DA and AUX_CL). In this mode, the auxiliary I^{2}C bus control logic (3rd party sensor interface block) of the MPU-60X0 is disabled, and the auxiliary I^{2}C pins AUX_DA and AUX_CL (Pins 6 and 7) are connected to the main I^{2}C 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 I^{2}C interface.

Auxiliary I^{2}C Bus IO Logic Levels

- MPU-6000: The logic level of the auxiliary I^{2}C bus is VDD
- MPU-6050: The logic level of the auxiliary I^{2}C 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).

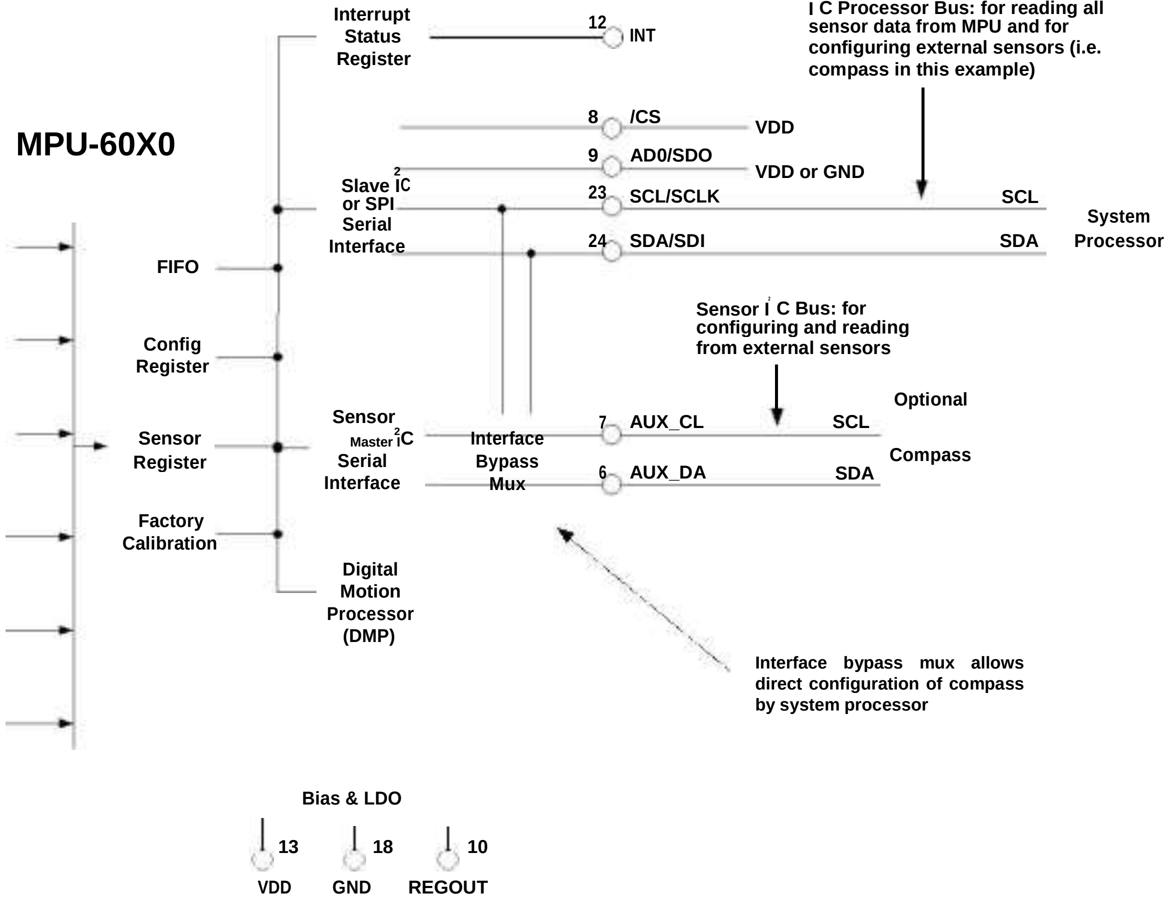
When the value of the self-test response is within the min/max limits of the product specification, the part has passed self test. When the self-test response exceeds the min/max values, the part is deemed to have failed self-test. Code for operating self test code is included within the MotionApps software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

7.13 MPU-60X0 Solution for 9-axis Sensor Fusion Using I^{2}C Interface

In the figure below, the system processor is an I^{2}C master to the MPU-60X0. In addition, the MPU-60X0 is an I^{2}C master to the optional external compass sensor. The MPU-60X0 has limited capabilities as an I^{2}C 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 I^{2}C bus pins 23 and 24 (SDA and SCL) directly to the auxiliary sensor I^{2}C 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 I^{2}C master can take control of the sensor I^{2}C bus and gather data from the auxiliary sensors.

For further information regarding I^{2}C master control, please refer to Section 10.


MPU-60X0
Interrupt Status Register
12
INT
I²C Processor Bus: for reading all sensor data from MPU and for configuring external sensors (i.e. compass in this example)
8
/CS
VDD
9
AD0/SDO
VDD or GND
Slave I²C or SPI Serial Interface
23
SCL/SCLK
SCL
24
SDA/SDI
SDA
System Processor
FIFO
Config Register
Sensor Register
Factory Calibration
Sensor I²C Bus: for configuring and reading from external sensors
Sensor Master I²C Serial Interface
Interface Bypass Mux
7
AUX_CL
SCL
6
AUX_DA
SDA
Optional
Compass
Digital Motion Processor (DMP)
Interface bypass mux allows direct configuration of compass by system processor
Bias & LDO
13
18
10
VDD
GND
REGOUT

7.14 MPU-6000 Using SPI Interface

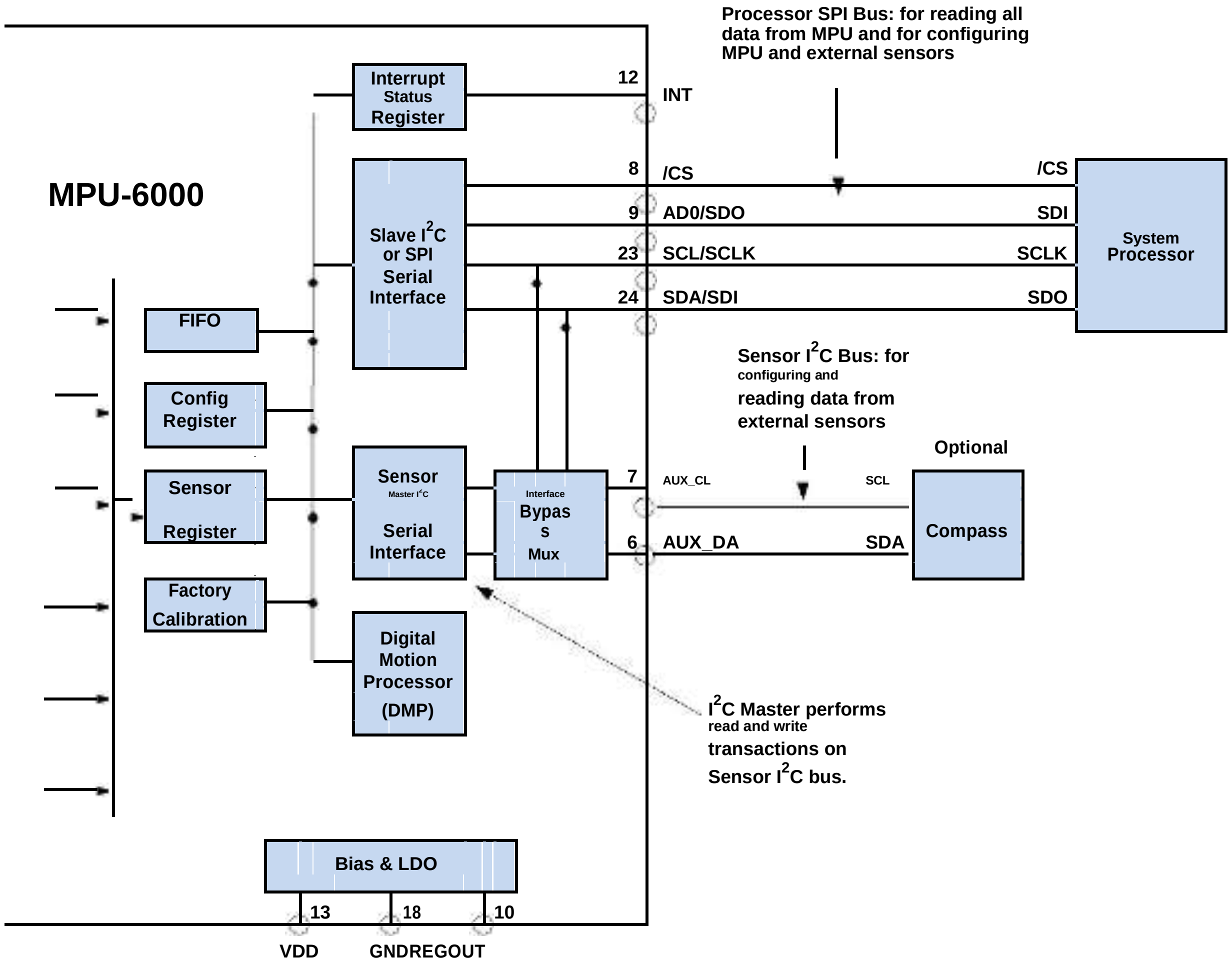
In the figure below, the system processor is an SPI master to the MPU-6000. Pins 8, 9, 23, and 24 are used to support the /CS, SDO, SCLK, and SDI signals for SPI communications. Because these SPI pins are shared with the I^{2}C slave pins (9, 23 and 24), the system processor cannot access the auxiliary I^{2}C bus through the interface bypass multiplexer, which connects the processor I^{2}C interface pins to the sensor I^{2}C interface pins.

Since the MPU-6000 has limited capabilities as an I^{2}C 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 I^{2}C 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 I^{2}C sensor bus can be achieved by using I^{2}C Slaves 0-4 to perform read and write transactions on any device and register on the auxiliary I^{2}C bus. The I^{2}C 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 I^{2}C 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 I^{2}C interface, please refer to the MPU-6000/MPU-6050 Register Map and Register Descriptions document.


Processor SPI Bus: for reading all data from MPU and for configuring MPU and external sensors
MPU-6000
Interrupt Status Register
12
INT
8
/CS
9
AD0/SDO
23
SCL/SCLK
24
SDA/SDI
/CS
SDI
SCLK
SDO
System Processor
Slave I2C or SPI Serial Interface
FIFO
Config Register
Sensor Register
Factory Calibration
Sensor Master I2C Serial Interface
Interface Bypass Mux
Digital Motion Processor (DMP)
Sensor I2C Bus: for configuring and reading data from external sensors
Optional
7
AUX_CL
SCL
6
AUX_DA
SDA
Compass
I2C Master performs read and write transactions on Sensor I2C bus.
Bias & LDO
13
18
10
VDD
GND
REGOUT

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 ±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 I^2C 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 ≤ VDD at all times.
9	Y		AD0 / SDO	I²C Slave Address LSB (AD0); SPI serial data output (SDO)
9		Y	AD0	I²C Slave Address LSB
23	Y		SCL / SCLK	I²C serial clock (SCL); SPI serial clock (SCLK)
23		Y	SCL	I²C serial clock
24	Y		SDA / SDI	I²C serial data (SDA); SPI serial data input (SDI)
24		Y	SDA	I²C 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.

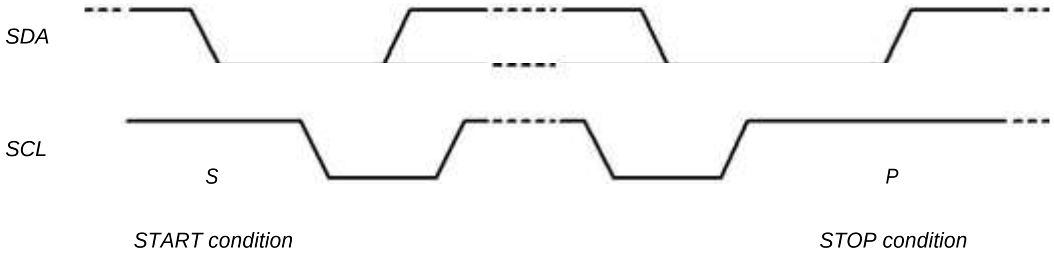
The slave address of the MPU-60X0 is b110100X which is 7 bits long. The LSB bit of the 7 bit address is determined by the logic level on pin AD0. This allows two MPU-60X0s to be connected to the same I²C bus. When used in this configuration, the address of the one of the devices should be b1101000 (pin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

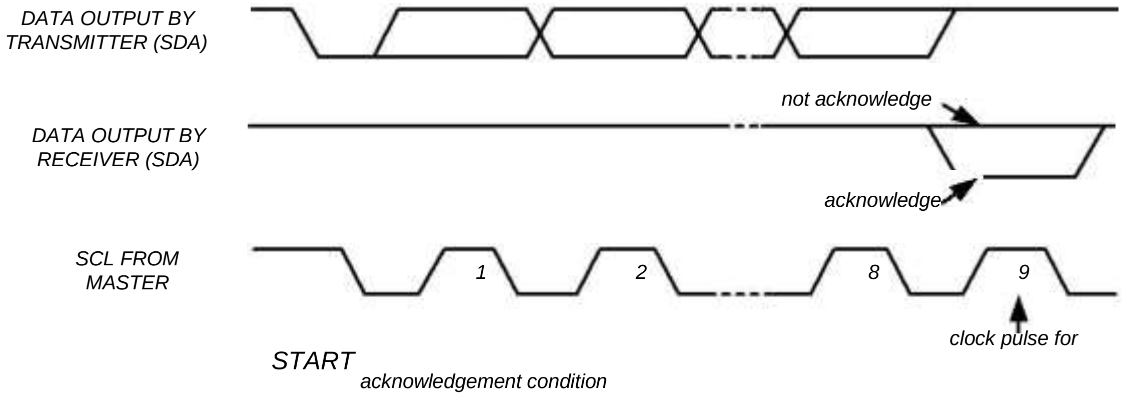


START and STOP Conditions

Data Format / Acknowledge

I^{2}C 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.

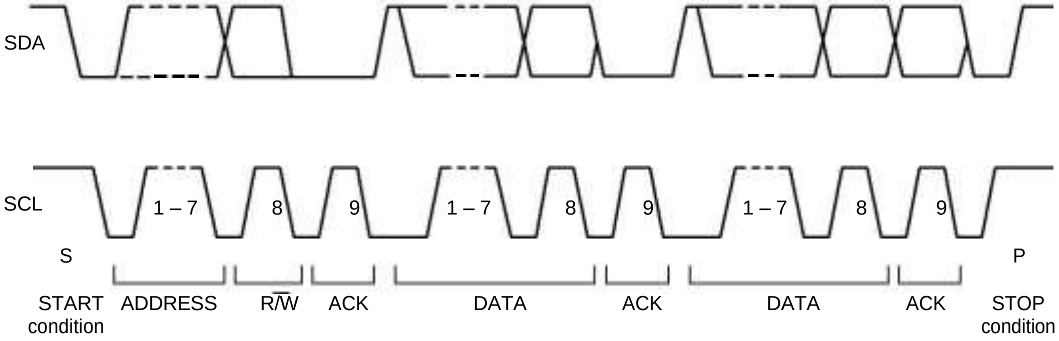
If a slave is busy and cannot transmit or receive another byte of data until some other task has been performed, it can hold SCL LOW, thus forcing the master into a wait state. Normal data transfer resumes when the slave is ready, and releases the clock line (refer to the following figure).



Acknowledge on the I^{2}C 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 I^2C Data Transfer

To write the internal MPU-60X0 registers, the master transmits the start condition (S), followed by the I^2C 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 I^2C 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 9^{th} 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		DAT A		

9.4 I^2C Terms

Signal	Description
S	Start Condition: SDA goes from high to low while SCL is high
AD	Slave I^2C address
W	Write bit (0)
R	Read bit (1)
ACK	Acknowledge: SDA line is low while the SCL line is high at the 9^{tn} 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:

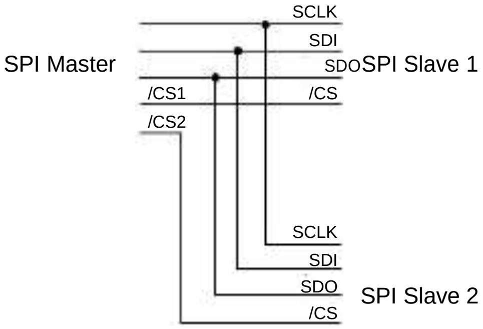
SPI Address format

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 I^{2}C 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*

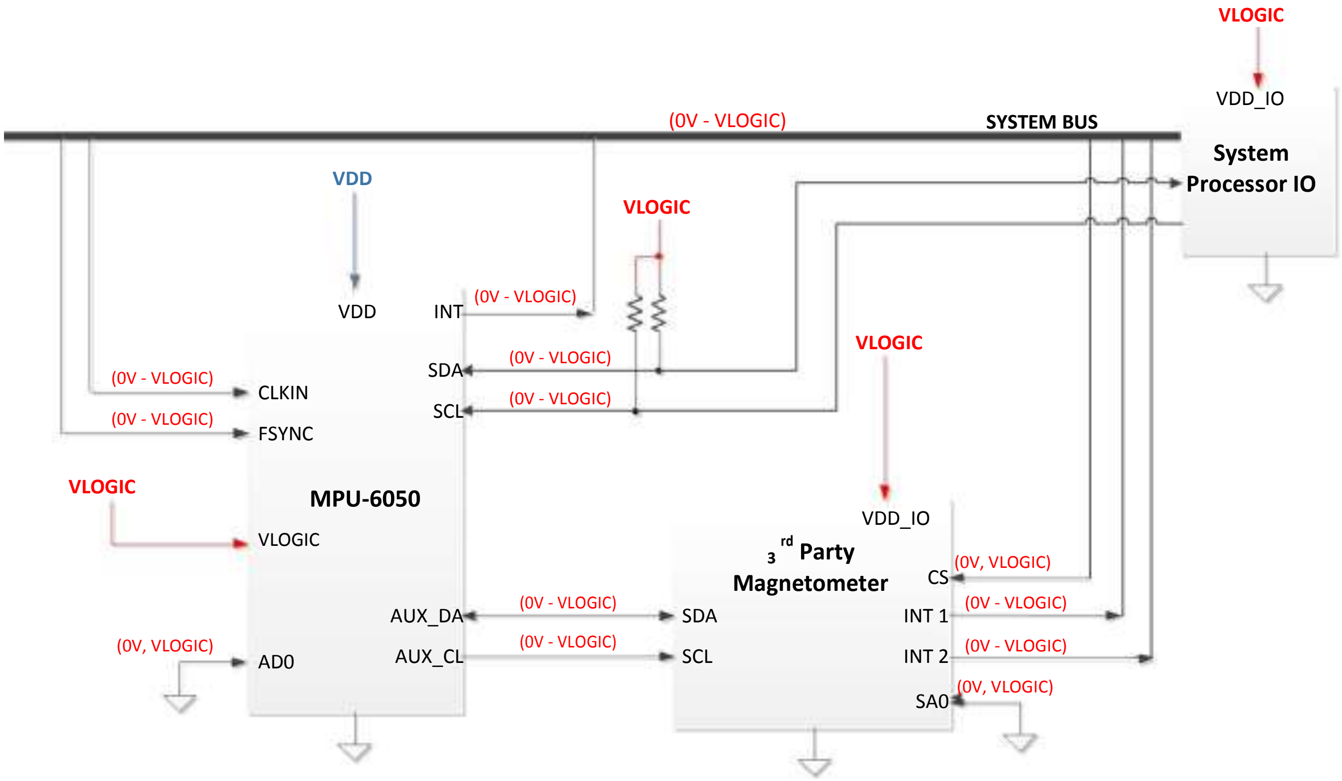
AUX_VDDIO	**MICROPROCESSOR LOGIC LEVELS** (Pins: SDA, SCL, AD0, CLKIN, INT)	**AUXILLARY LOGIC LEVELS** (Pins: AUX_DA, AUX_CL)
0	VLOGIC	VLOGIC

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 I^{2}C 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 I^2C 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:

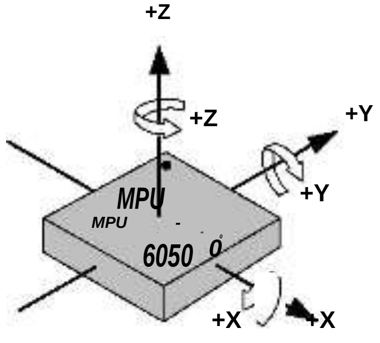
1. AUX_VDDIO determines the IO voltage levels of AUX_DA and AUX_CL (0 = set output levels relative to VLOGIC)
2. All other MPU-6050 logic IOs are referenced to VLOGIC.

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

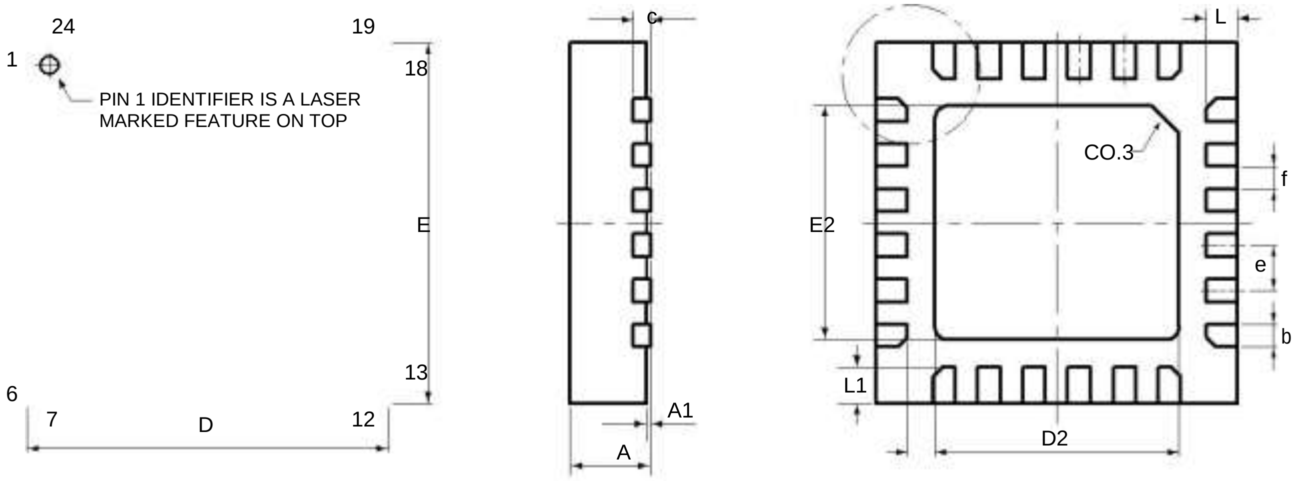
The diagram below shows the orientation of the axes of sensitivity and the polarity of rotation. Note the pin 1 identifier (•) in the figure.

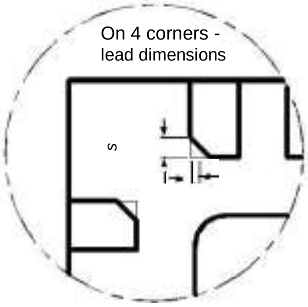


Orientation of Axes of Sensitivity and
Polarity of Rotation

11.2 Package Dimensions

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

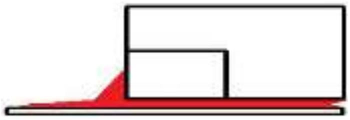




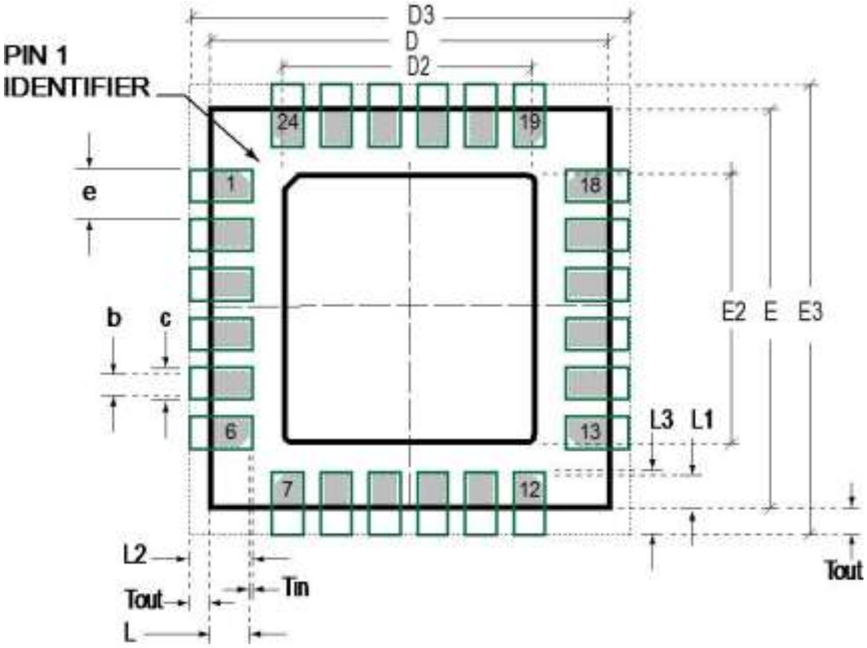
SYMBOLS	DIMENSIONS IN MILLIMETERS		
	MIN	NOM	MAX
A	0.85	0.90	0.95
A1	0.00	0.02	0.05
b	0.18	0.25	0.30
c	---	0.20 REF	---
D	3.90	4.00	4.10
D2	2.65	2.70	2.75
E	3.90	4.00	4.10
E2	2.55	2.60	2.65
e	---	0.50	---
f (e-b)	---	0.25	---
K	0.25	0.30	0.35
L	0.30	0.35	0.40
L1	0.35	0.40	0.45
s	0.05	---	0.15

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

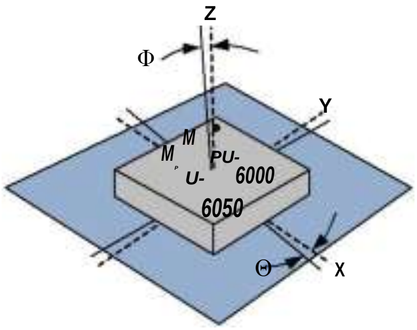
Routing traces or vias under the gyro package such that they run under the exposed die pad is prohibited. Routed active signals may harmonically couple with the gyro MEMS devices, compromising gyro response. These devices are designed with the drive frequencies as follows: X = 33±3Khz, Y = 30±3Khz, and Z=27±3Khz. 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 (sinθ or sinΦ)
0º	0%
0.5º	0.87%
1º	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 10,000*g*. 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 10,000*g* 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 10,000*g.*
- 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.

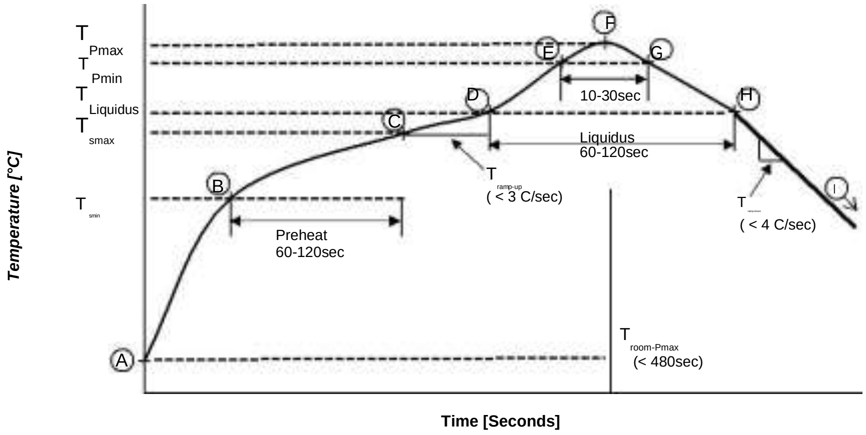
11.4.8 Reflow Specification

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:

SOLDER REFLOW PROFILE FOR QUALIFICATION
LEAD-FREE IR/CONVECTION



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.

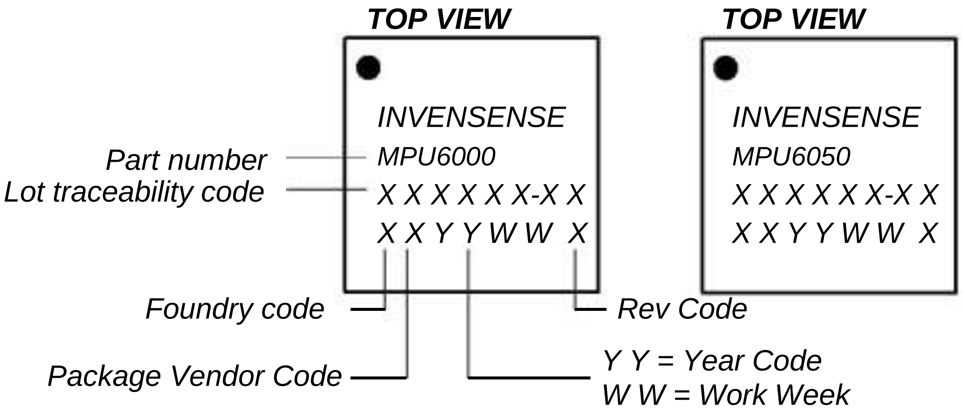
Production Reflow: Check the recommendations of your solder manufacturer. For optimum results, use lead-free solders that have lower specified temperature profiles (Tp_{max} ~ 235°C). Also use lower ramp-up and ramp-down rates than those used in the qualification profile. Never exceed the maximum conditions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

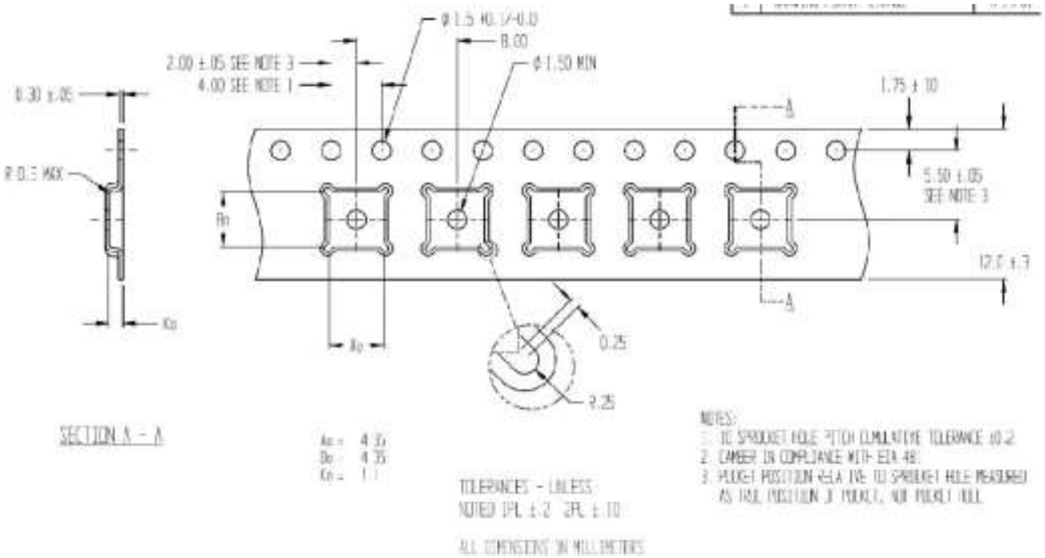
Calculated shelf-life in moisture-sealed bag	12 months -- Storage conditions: <40°C and <90% RH
After opening moisture-sealed bag	168 hours -- Storage conditions: ambient ≤30°C at 60%RH

11.6 Package Marking Specification

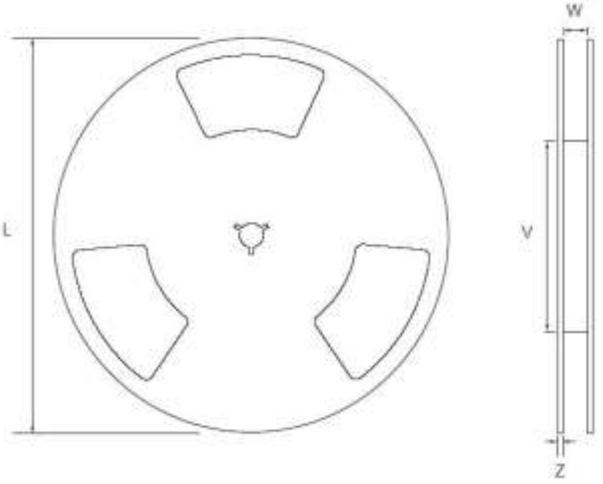


Package Marking Specification

11.7 Tape & Reel Specification



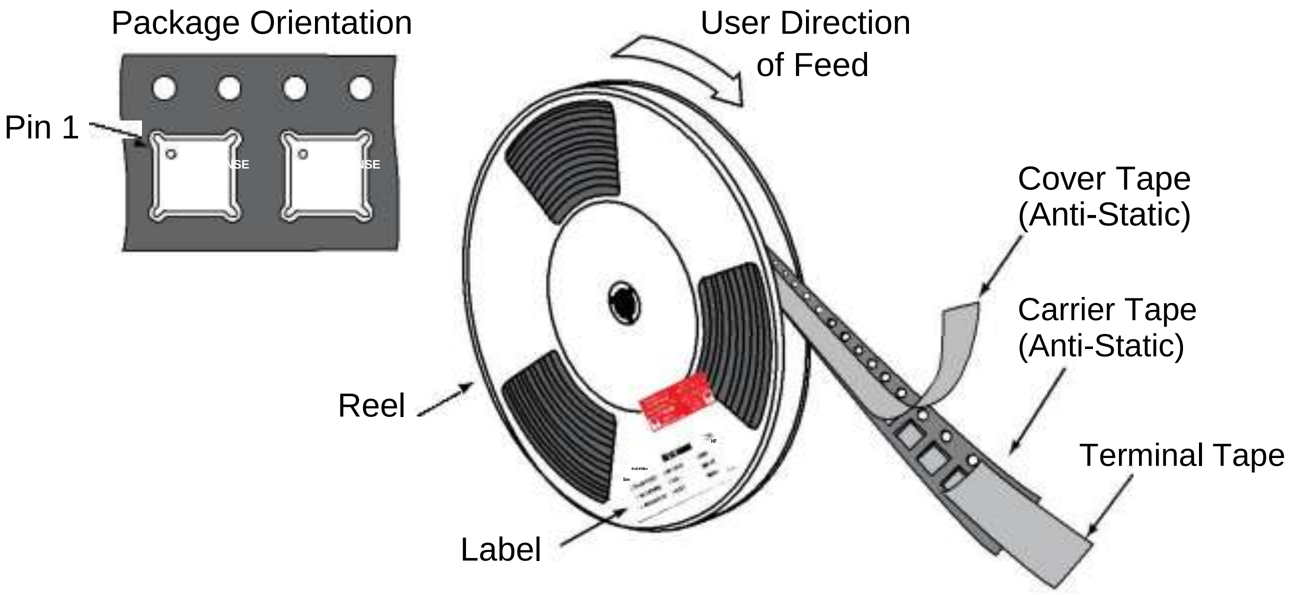
Tape Dimensions



Reel Outline Drawing

Reel Dimensions and Package Size

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

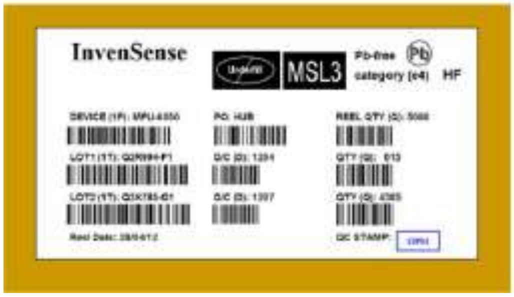


Tape and Reel Specification

Reel Specifications

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

11.8 Label



InvenSense

MSL3 Pb-free category (e4) HF

DEVICE (1P): MPU-6050
PO: HUB
REEL QTY (Q): 5000

LOT1 (1T): Q2R994-F1
D/C (D): 1204
QTY (Q): 613

LOT2 (1T): Q3X785-D1
D/C (D): 1307
QTY (Q): 4385

Reel Date: 28/0412
QC STAMP:

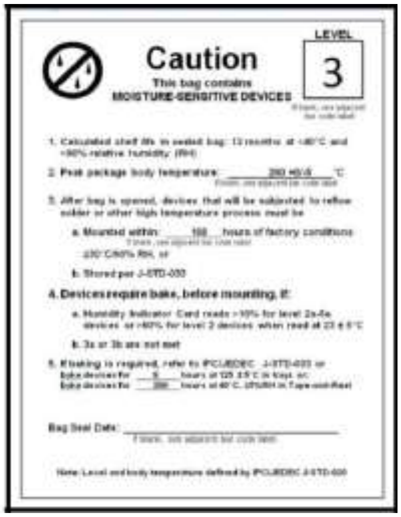
Barcode Label

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



LEVEL 3

Caution

This bag contains
MOISTURE-SENSITIVE DEVICES

If blank, see adjacent bar code label

1. Calculated shelf life in sealed bag: 12 months at <40°C and <90% relative humidity (RH)
2. Peak package body temperature: 260 +0/-5 °C
   If blank, see adjacent bar code label
3. After bag is opened, devices that will be subjected to reflow solder or other high temperature process must be
   a. Mounted within: 168 hours of factory conditions
      If blank, see adjacent bar code label
      ≤30°C/60% RH, or
   b. Stored per J-STD-033
4. **Devices require bake, before mounting, if:**
   a. Humidity Indicator Card reads >10% for level 2a-5a devices or >60% for level 2 devices when read at 23 ± 5°C
   b. 3a or 3b are not met
5. If baking is required, refer to IPC/JEDEC J-STD-033 or
   bake devices for 5 hours at 125 ±5°C in trays, or;
   bake devices for 288 hours at 40°C, ≤5%RH in Tape-and-Reel

Bag Seal Date: ______________________
If blank, see adjacent bar code label

Note: Level and body temperature defined by IPC/JEDEC J-STD-020

MSL3 Label

떨어뜨리지 마시오
Do not drop
충격주의 (깨지기 쉬움)
Fragile
취급주의
Handle with Care
Không làm rơi
勿摔落 * 落下注意
Chú ý va đập (dễ vỡ)
易碎品 * 割物注意
Chú ý khi cầm
小心輕放
取扱注意

Caution Label

ATTENTION
OBSERVE PRECAUTIONS
FOR HANDLING
ELECTROSTATIC
SENSITIVE
DEVICES

ESD Label

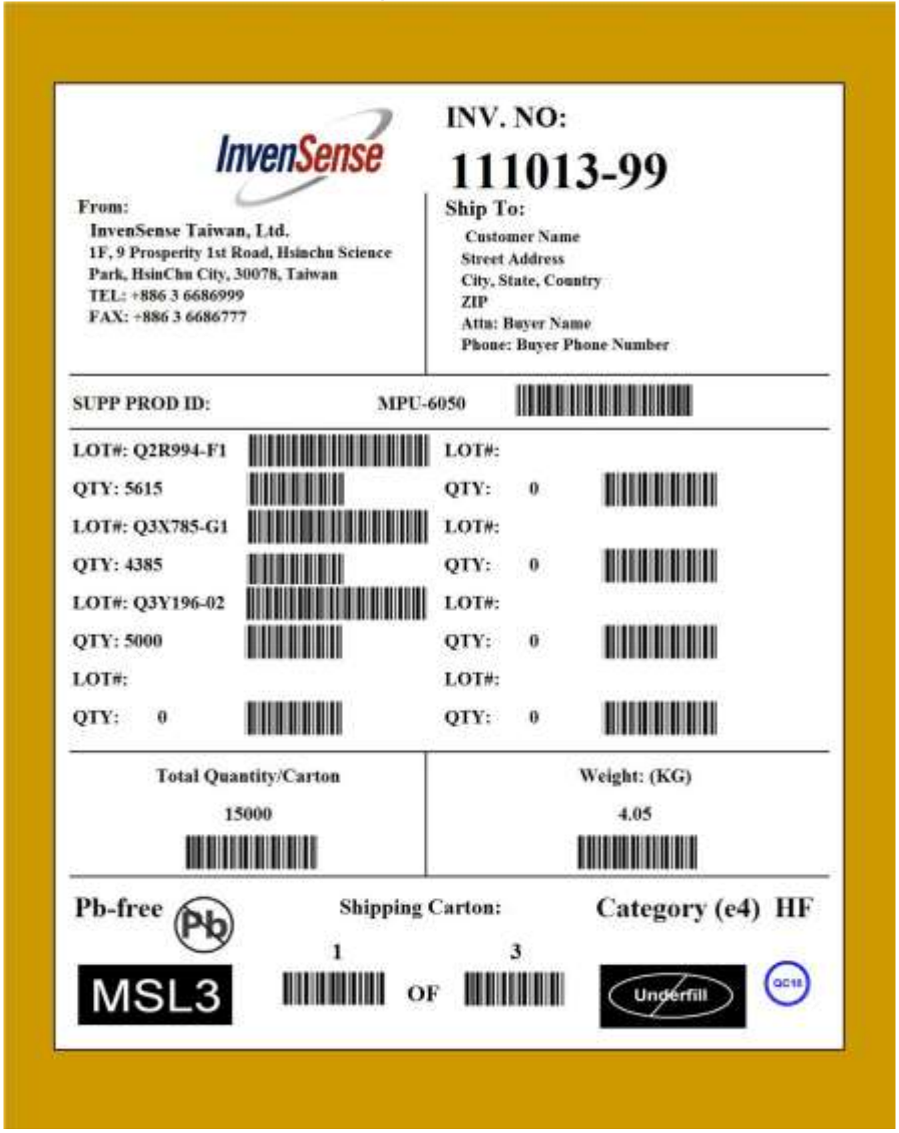
Inner Bubble Wrap

Pizza Box

Pizza Boxes Placed in Foam-Lined Shipper Box

Outer Shipper Label

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	Weight: (KG)
15000	4.05

Pb-free

Shipping Carton: 1 OF 3

Category (e4) HF

MSL3

Underfill

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 JESD47I 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, Tj>125°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; ±100mA	1	6	(0/1)
(MS) Mechanical Shock	JEDEC JESD22-B104C, Mil-Std-883, Method 2002.5, Cond. E, 10,000*g's*, 0.2ms, ±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°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, 10000*g'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°C], Soak mode 2 [5'], 1000 cycles	1	40	(0/1)

[1] Tests are preceded by MSL3 Preconditioning in accordance with JEDEC JESD22-A113F

Ultrasonic Sonar Module

Hardware Manual Rev 1r0

28<R.., , $';@5$3, A(3456'&7$, 6'&54, ;'CA(#, 76, 5, ('F, $'63, 6'(AS'&, T'4, $74$A73, 5@@(7$5S'&6, 3%53, 4#UA74#6, C7635&$#, ;#56A4#;#&36, T4';, 5&, 'VW#$3X, 6A$%, 56, F5((6X, TA4&73A4#X, 5&C,#Y#&,@#36K,26#4,$74$A73,7&7S53#6,5,;#6A4#;#&3,V*,C47Y< 7&), 3%#, 28<R.., 347))#4, 7&@A3, 3', (')7$, 0GZ0K, "%#, 28<R..X, 7&,4#6@'&6#X,F7((,6#&C,5,6%'43,VA4636,'T,A(3456'&7$,6'A&C, F5Y#X, 5&C, 3%#&, 'A3@A36, 5, @A(6#, 56, 6''&, 56, 5, 4#3A4&7&), #$%',76,C#3#$3#CK,"%#,A6#4,$74$A73,4#6'(Y#6,3%#,C7635&$#,V*, ;#56A47&),3%#,@A(6#,F7C3%,'T,3%#,'A3@A3,@A(6#K,I7635&$#6, A@,3',-K>,;#3#46,T4';,3%#,6#&6'4,$5&,V#,;#56A4#CX,F73%, 4#6'(Y7&),4#6'(AS'&,C#@#&C7&),;57&(*,'&,3%#,A6#4,$74$A73K

Features:

!"#$%&'(')*+,-./01,2(3456'&7$,8'&54,

, !8#&6'4,95&)#+,:$;<,=>.$;

, ,,?"#;@#453A4#,B';@#&653#CD,

, !E'F#4,G&@A3+,=H<>HIB,J,=K>;L,"*@

, !"47))#4,G&@A3,+,""M

, !N$%',OA3@A3+,""M

, !EBP,871#+,:.Q->;;

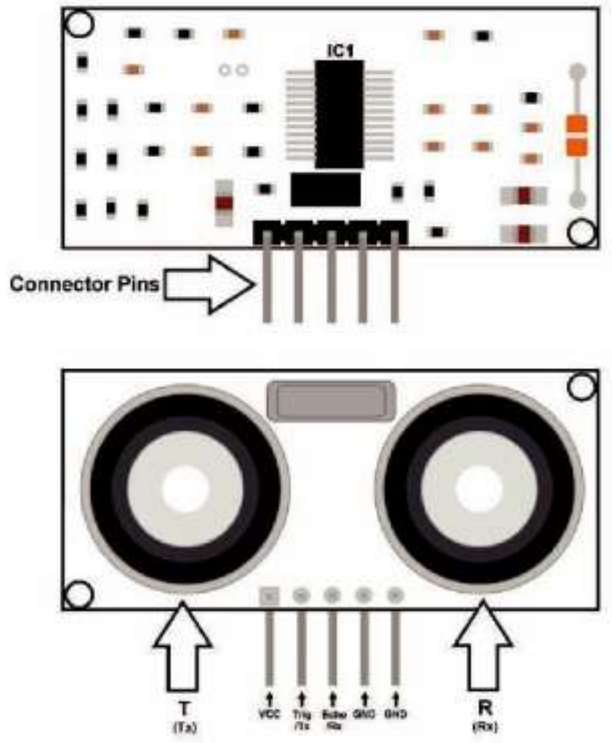


Figure 1. *!"#$%&'()*+,'(%$+)""-&#$%.'(+&/'0)(1+)#&+ major Components.*

Important: *JP1 is normally OPEN (uninstalled).*

Table 1. *!"#$%&'()*+2)&#%(*3+,3(&'$+4)(+5&&)1(63(#&+ %(7+23&*$)8.'(&9*

Pin No.	ID	!"#$%&'()*
R	HBB	>H,E'F#4,G&@A3
:	"47)	"47))#4,G&@A3,5$SY#,0GZ0
=	N$%'	EA(6#,'A3@A3
-	Z[I	)4'A&C
>	Z[I	)4'A&C

NOTE: *:/3+;<=+6%(->%*#-$3$+'>+#/3+!,?@AA+&'(%$+6'7-"3B+>'$+-(C('0(+$3%&'(B+0)""+('#+8$'D)73+-&+%+-&3$+6%(-%"+ '$+%(E+'#/3$+)(>'$6%.'(+>'$+#/%#+6%F3$9+G3(*3B+%""+)(>'$6%.'(+*'(#%)(37+)(+#/)&+7'*-63(#+03$3+'H#%)(37+HE+%*#-%"+ 63%&-$363(#&+%(7+3I83$)63(#%.'(9*

TIMING DIAGRAM

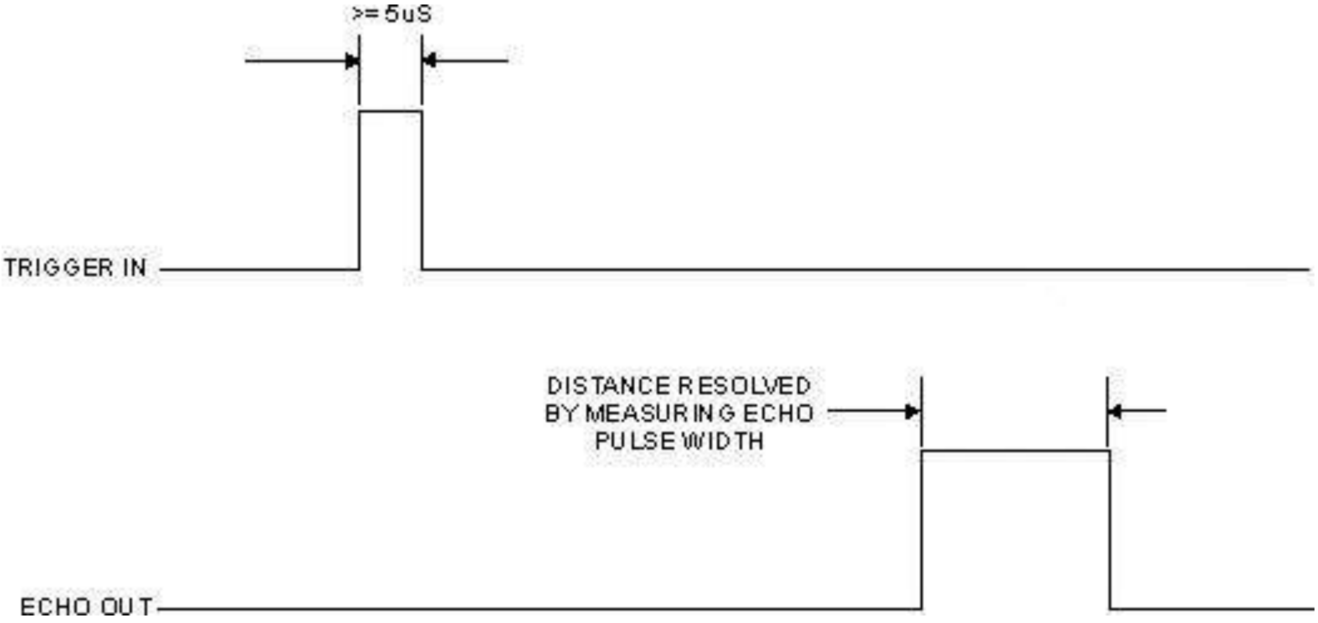


Figure 2. *!,?@AA+:)6)(1+2)%1$%69+2)&#%(*3+63%&-$363(#+)&+)().%#37+HE+&3(&)(1+%+8-"&3+#'+#/3+ #$)113$+)(8-#9+:/3+!,?@AA+0)""+$3&8'(7+0)#/+%+8-"&3+0)#/+%+7-$%.'(+*'$$3&8'(7)(1+#'+#/3+$'-(7+ #$)8+3*/'+.639+*

RESOLVING DISTANCE

!"#$%&#$'(')*#+',*-,(#"%./)#')#0*1-&#2#'331"$-,$&"4#
+'"$,)5&#(&,"1-&(&)$#'"#')'$',$&+#67#,5$'8,$')*#$%&#
$-'**&-#')91$#/'$%#/'$%#,#913"&:#;%&#<=>?@@#/'33#-&>
"9.)+#/'$%#,#&5%.#.1$91$#913"&:#;%&#&5%.#-&"9.)"&#
913"&#/'+$%#5.--&"9.)+"#$.#$%&#$'(&#'$#$,A&"#B.-#$%&#
13$-,".)'5#".1)+#$.#$-,8&3#B-.(#$%&#"&)".-#$.#$%&#
.6C&5$#,)+#6,5A:#D&)5&#$%&#+'"$,)5&#B-.(#$%&#.6C&5$#
5,)#6&#5.(91$&+#67E
-&+15&+#$.E

F'"$,)5&#G#H13"&#I'+$%#J#=9&&+#.B#=.1)+#K#2

!#913"&#/'+$%#8,31&#*-&,$&-#$%,)#L@(=#')+'5,$&"#,)#
.1$#.B#-,)*&#5.)+'$'.):

;%&#,5$1,3#"9&&+#.B#".1)+#+&9&)+"#.)#"&8&-,3#
&)8'-.)(&)$,3#B,5$.-"4#/'$%#$&(9&-,$1-&#%,8')*#$%&#
(."$#9-.).1)5&+#&BB&5$:#;%&#"9&&+#.B#".1)+#')#+-7#
'"#+&$&-(')&+#,99-.M'(,$&37#67E

N#G#OO?:P#Q#@:L;#(K"

R1$#.)&#.B#$%&#B&,$1-&#.B#<=>?@@#'"#'$"#61'3$>')#$&(>
9&-,$1-&#5.(9&)",$'.):#D&)5&4#/'$%#$&(9&-,$1-&#
&BB&5$#.1$#.B#$%&#&S1,$'.)4#$%&#+'"$,)5&#B.-(13,#'"#

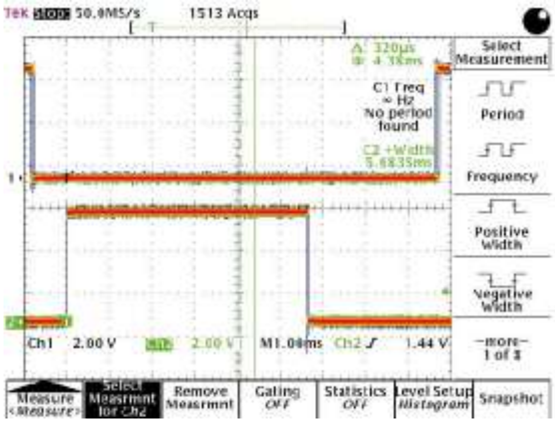
F'"$,)5&#G#H13"&I'+$%#J#?LT:U#(&$&-"

I%&-&E#H13"&I'+$%#')#"&5.)+"

DETECTION WINDOW CONSIDERATIONS

;%&#<=>?@@#%,"#,#+&$&5$'.)#9,$$&-)#$%,$#"9-&,+"#
.1$#B-.(#$%&#"&)".-#(.1$%#,$#V?T#+&*-&&"#,)*3&:#
W)&#.68'.1"#,)+#9-.6,637#1)+&'"-,63&#&BB&5$#.B#$%'"#
5%,-,5$&-'"$'5"#'"#$%&#"&)".-#/'33#%,8&#,)#&BB&5$'8&#
3,-*&-#+&$&5$'.)#/')+./#$%&#B1-$%&-#,/,7#$%&#.6C&5$#
.B#')$&-&"$#'"#B-.(#$%&#"&)".-:#;%'"#/'33#,33./#$%&#"&)>
".-#$.#X"&&Y#(.-&#.6C&5$"4#%&)5&4#/'33#6&#')5-&,"')*#37#
+'"$-,5$&+#67#.$%&-#)&,-67#.6C&5$"4#(,A')*#'$#(.-&#
9-.)&#$.#&--.-:#

Z.)*#+'"$,)5&#+&$&5$'.)#-&S1'-&"#$%&#$,-*&$#.6C&5$#
5-.""#"&5$'.)#6&#3,-*&#&).1*%#B.-#,551-,$&#,)+#-&3'>
,63&#+&$&5$'.):



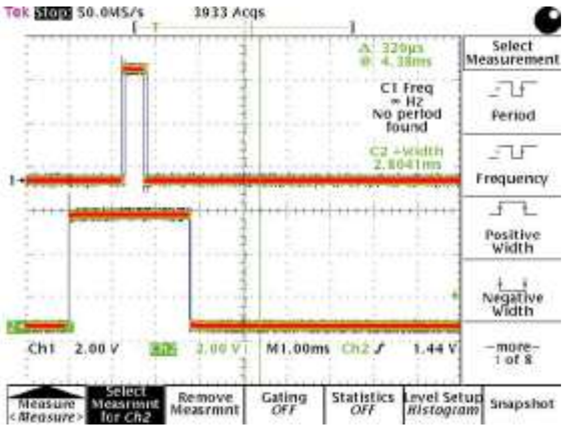
Tek Stop 50.0MS/s 1513 Acqs

Δ: 320µs
@: 4.38ms

C1 Freq
∞ Hz
No period
found

C2 +Width
5.6835ms

Ch1 2.00 V Ch2 2.00 V M1.00ms Ch2 ƒ 1.44 V



Tek Stop 50.0MS/s 3933 Acqs

Δ: 320µs
@: 4.38ms

C1 Freq
∞ Hz
No period
found

C2 +Width
2.8041ms

Ch1 2.00 V Ch2 2.00 V M1.00ms Ch2 ƒ 1.44 V

Figure 3. !,?@AA+ 3*/'+ '-#8-#+ J"'03$+ #$%*3K+ $38'$#&+ %+ **Figure 4.** T)#/+#/3+#3&#+'HO3*#+6'D37+*"'&3$+#'+%H'-#+/%">+
L9MN6&+8-"&3+0)7#/+0/3(+%+#3&#+'HO3*#+)&+8'&).'(+%H'-#+%+ 63#3$B+ #/3+ 8-"&3+ 0)7#/+ %&+ '-#8-#37+ HE+ !,?@AA+)&+ *'$$3?
63#3$+>$'6+#/3+&3(&'$9+P'(D3$.(1+#/)&+#'+3Q-)D%"3(#+7)&? &8'(7)(1"E+$37-*37+)(+/%">+JU9N6&K9+
#%(*3+E)3"7&+%+D%"-3+'>+A9RS69

APPLICATION HINTS

"%#, 28<R.., F7((, F'4/, F73%, @'F#4, 6A@@(*, Y'(35)#6, =H, 3', >HIBK, \'4, V#63, (')7$, (#Y#(, ;53$%7&)X, A6#, 3%#, 65;#, HCC, 6'A4$#,3%#,%'63,$'&34'((#4,76,A67&)K,M'F#4,6A@@(*,Y'(35)#, ;5*,4#6A(3,7&,5,4#CA$S'&,7&,C7635&$#,C#3#$S'&,45&)#K

"%#, 28<R.., 4#UA74#6, '&(*, 3F', G]O, @'43, 3', 7&3#4T5$#, F73%, 5,%'63,$'&34'((#4K,L6,6%'F&,7&,\7)K,,>X,3%#,%'63,$'&34'((#4, ;A63, V#, 5667)&#C, '&#, 'A3@A3, @'43, 3', C47Y#, 3%#, 28<R.., 347))#4,7&@A3X,5&C,'&#,7&@A3,@'43,3',4#5C,5&C,;#56A4#,3%#, #$%',@A(6#,F7C3%,'A3@A3K,

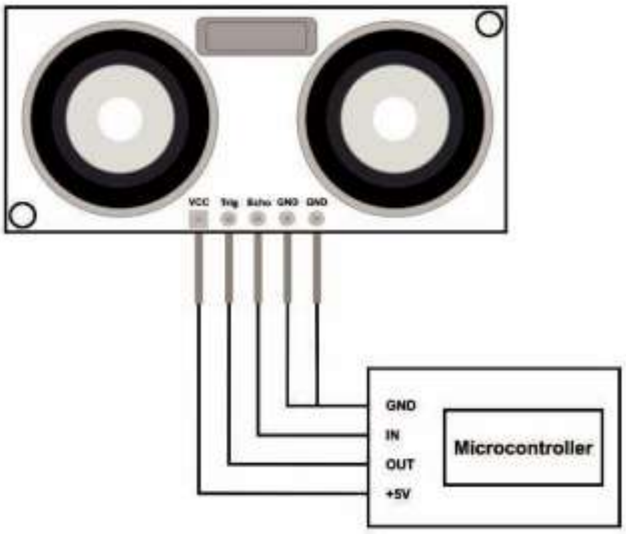


Figure 5. !,?@AA+#'+/'&#+*'(#$'""3$+0)$)(1+3I%68"39

gizDuino Wiring and Demo Code

```
// Demo sketch
// "is sketch will output distance info via the UART port

// port assignment
// change as may be necessary
            const     int trigger=6;
            const     int echo=7;
            #oat      distance;

void setup(){
            Serial.begin(9600);
            pinMode(trigger,OUTPUT);
            pinMode(echo,INPUT);
}

void loop(){

// Trigger US-100 to start measurement
            // Set up trigger
            digitalWrite(trigger,LOW);
            delayMicroseconds(5);
            // Start Measurement
            digitalWrite(trigger,HIGH);
            delayMicroseconds(10);
            digitalWrite(trigger,LOW);
            // Acquire and convert to mtrs
            distance=pulseIn(echo,HIGH);
            distance=distance*0.0001657;
            // send result to UART
            Serial.println(distance);
            delay(50);
}
```

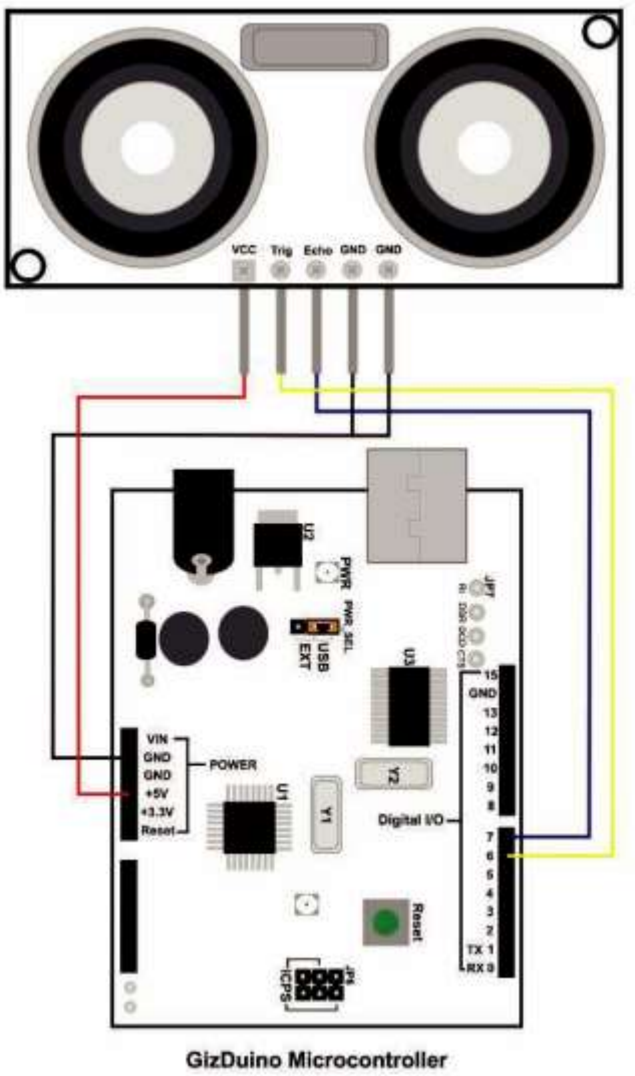


Figure 6. *!,?@AA+!"#$%&'()*+,'(%$+P'((3*#37+#'+%+1)V? 2-)('+=)*$'*'(#$'""3$+C)#9+W(+#/)&+3I%68"3B+#/3+#$)113$+%(7+3*/'+8)(&+%$3+*'((3*#37+#'+8'$#+M+%(7+X+$3&83*?*

.D3"E9+Y'-+%(+*/%(13+#/3+*'((3*.'(+%(7+-&3+'#/3$+ ports instead.*