POINTING ISN'T RUDE: A PROOF-OF-CONCEPT HAB PAYLOAD STABILIZER

Senior Honors Thesis
Submitted in partial fulfillment of requirements to graduate from
the Norbert O. Schedler Honors College
By

Russell Jeffery

University of Central Arkansas
Conway, AR 72035

Spring 2019

Thesis committee:

Advisor:________________________________
William Slaton, Ph.D.
Professor of Physics

Reader:________________________________
Ashley Beyerl, M.S.
Visiting Lecturer of Physics

Director:________________________________
Patricia Smith, Ed.D.
Interim Dean of the Honors College

Table of Contents

Abstract

High-Altitude Balloons (HABs) are relatively cheap, versatile platforms for research projects in physics, meteorology, engineering, and other related fields because of their low cost in comparison with other platforms that offer similar capabilities. One of the major drawbacks of HAB platforms, however, is their instability; lack of any fixed attachment point makes it practically impossible to use directional instruments in any controlled manner. While several HAB payload stabilization methods have already been developed, they all have limitations that leave something to be desired. For example, gyroscopes necessarily take up a large portion of the payload mass, leaving little room for instruments within the legal weight limits; servo motors promise low-cost, high-precision control, but it is difficult to design adequate control algorithms to compensate for the lack of stable attachment points; passive systems can be very cheap and relatively effective but still lack precise orientation control capabilities. A method that does not appear to have been tried, however, is the use of cold-gas thrusters. This thesis details the design and construction of a proof-of-concept, cold-gas-thruster stabilization device and thoughts about the potential value of further development.

Pointing Isn't Rude: A Proof-of-Concept HAB Payload Stabilizer

Introduction

High-altitude balloons (HABs), more commonly known as weather balloons, are latex sacs filled with helium. They are just like party balloons but sized on the order of feet rather than inches. Because they are so much larger than party balloons, they can lift substantially more than their own weight and are therefore used to carry scientific payloads miles into the air--two or three times higher than commercial airlines fly--before popping and falling back to the ground where they are recovered. High-altitude balloons are a low-cost and widely-used platform for researchers at all levels doing projects in physics, meteorology, engineering, and other disciplines because these balloons provide access to near-space conditions at only a tiny fraction of the cost of the spacecraft or highly specialized aircraft that would otherwise be required to carry instruments to such high altitudes. Physicists and meteorologists can directly observe phenomena like the exponential change in pressure with altitude, and engineers can test proof-of-concept systems in conditions that come very close to the freezing, irradiated vacuum of space.

As useful as HABs are, however, they come with their share of drawbacks. One significant problem is that, buffeted by wind and jiggling on a long, dangling rope hanging from a balloon, it is very hard to get a HAB payload to point in any particular direction. This instability is not an issue for measurements like temperature or pressure for which the orientation of the payload doesn't really matter, but directional instruments like cameras need to be held still to reliably capture high-quality data. To address the need for stability, various types of stabilization systems have been developed--each with its own problems and benefits--which will be discussed at length in the literature review below.

This project will involve building a proof-of-concept, cold-gas reaction control system (RCS), meaning that the control will be provided by nozzles that eject compressed gas (much like a conventional chemical rocket engine, but without combustion) which can then be developed into a flight-ready system that can stop a HAB payload from spinning. A cold-gas RCS will be used for several reasons: the technology, though not often applied to HAB payloads, has been widely used and is thoroughly developed and well documented; at small scales, cold-gas RCSs are simpler and more efficient than other propulsion systems; and the gas propellent (often nitrogen, carbon dioxide, or some other common, inert gas) is cheap, readily available, and safe to work with (Anis, 2012, p. 448; Nothnagel, 2011, pp. 37-38).

There are, however, problems with cold-gas propulsion--particularly in HAB applications that require low weight, small size, and low cost. While a lot of parts are available off-the-shelf, they are usually designed for other applications. For example, Nothnagel (2011) writes that a wide range of solenoid valves[1] can be found that meet requirements for speed, pressure, weight, or price, but there are few or none that meet all simultaneously (pp. 37-38). Also, while balloons can be made to carry payloads as heavy as cars well above the troposphere, heavier payloads are subject to more regulations and are more expensive. For most independent projects, Federal Aviation Administration (FAA) regulations require licensure for payloads weighing over 6 pounds, and grant budgets often limit payload cost to a couple of thousand dollars at most. The

[1] A solenoid valve is a kind of valve that can be controlled with an electrical current. They are used in numerous applications--from washing machines to medical devices; from soft drink dispensers to industrial machinery--to control the flow of gas or liquid.

significance of these weight restrictions to this project means that, while the concept may prove to be sound and effective, a need for heavy hardware may still make this type of system impractical.

Literature Review

The main advantages of cold-gas systems over other systems is that they are simpler to implement, safer to work on, and very reliable. The main disadvantage is that, because all the propulsive energy is stored as pressure in the reservoir, they cannot hold nearly as much energy as chemical rockets can. Cold-gas propulsion, while not usually applied to HAB platforms, has been used extensively in small satellites, so there exists a large body of literature on the subject (Furumo, 2013; Nothnagel, 2011). Below, an in-depth investigation of relevant literature explores the broad topics of hardware design, software control, and prototyping of cold-gas systems.

Hardware Design

There are several methods of active and passive stabilization other than cold-gas RCS (Flaten, Gosch, & Habeck, 2015; Nirmal et al., 2016; Quine, Strong, & Wiacek, 2002, pp. 621-624). These other methods of stabilization have their benefits and drawbacks: passive stabilization can dampen and reduce the severity of the payload motion (Flaten et al., 2015), but it provides no controlled pointing ability; rotating servo motors have the potential for high precision, but they require the difficult calculation of stable reference frames from which to measure their movements; monopropellant, liquid fuel, and other types of RCS can be more efficient than cold-gas, but they are also more complicated and dangerous to work with

(Nothnagel, 2011, pp. 33-37). Cold-gas RCSs are not flawless, but in theory, they can provide comparable stabilization, complexity, and safety, at low cost and low weight.

Choice of Propellant

One factor to be considered when designing a cold-gas RCS is the choice of gas to be used as propellent. Anis (2012) and Nothnagel (2011) both went with the popular choice--nitrogen (Anis, 2012, p. 448; Nothnagel, 2011, pp. 37-38). While Nothnagel (2011) describes the specific impulse[2] of nitrogen as "mediocre," this shortcoming is outweighed by nitrogen's comparatively high storage density and low cost. Hydrogen and helium have high specific impulses in cold-gas systems, but they have very low densities and therefore require larger, heavier storage systems. Nothnagel (2011) also notes that helium systems are prone to leaks and the gas itself is more expensive than other options. Nothnagel (2011) considers air, but rejects it as an option due to a lack of the necessary tools for processing it (compressing, filtering, drying, etc.) effectively. Lastly, both Anis (2012) and Nothnagel (2011) considered carbon dioxide (CO_2) but both rejected it: Anis (2012) briefly cites CO_2's "toxic nature" but does not explain this concern further. Nothnagel (2011) rejects CO_2 due to concerns that the sloshing liquid would add either stability issues or tank complexity (p. 37). The concerns about the toxicity and instability of CO_2 may be warranted, but because of CO_2's low cost and the availability of paintball gun hardware designed specifically for delivering bursts of CO_2 gas, CO_2 will be used in this project.

---

[2] Specific impulse is to rocket propulsion systems what fuel economy is to automobiles. It is a measure of efficiency. Here, Nothnagel is discussing the differences in efficiency between similar systems equipped with different gas propellants.

Amount of Propellant Needed

For this project, the RCS will be used exclusively for attitude control--not to change the speed of the payload or keep it aloft. Unfortunately, while Anis (2012), Furumo (2013), and Nothnagel (2011) all discuss the Tsiolkovsky equation for calculating fuel requirements for basic, one-dimensional acceleration, none of them propose a way to calculate the fuel requirements for stabilization (Anis, 2012, pp. 448-449; Furumo, 2013, p. 69; Nothnagel, 2011, pp. 49-51). Furumo (2013) mentions later in the paper that "a more accurate way to characterize the delta-v required for attitude control is needed" (p. 77). The uncertainty that makes it difficult to know the precise propellent requirements is the unpredictability of wind, the main force that the stabilizer will be fighting. Because of this uncertainty, it may be more practical in some cases to simply measure the capabilities of a physical system and base future designs on that information rather than wasting time doing inaccurate math.

Pressure System

Designing the pressure system involves first an estimation of the system parameters; then selection of appropriate parts; and finally, simulation and/or testing of the system. The major parameters for the HAB system have already been mentioned: the weight limit of the system is set by the laws regulating HAB activities; the monetary budget is set by grant funding and likely should not exceed $1000; but it is the final parameter defining the performance requirements of the system that is the real problem. Because of the limitations caused by the lack of a formula to calculate the fuel requirements, the most realistic goal may simply be to maximize the effectiveness of the system.

Furumo (2013) and Nothnagel (2011) designed their systems with little more than the following major parts (including, of course, all necessary piping and such): a pressure tank, a regulator, and the thrusters, each with its own solenoid valve (Furumo, 2013, pp. 70-71; Nothnagel, 2011, pp. 45-46). The main differences are the respective configurations of the thrusters and a few sensors, safety devices, and valves that the designers deemed necessary. Using a unique solenoid for each thruster allowed Furumo (2013) and Nothnagel (2011) to minimize the number of thrusters required to perform all six 3-dimensional maneuvers (translation along and rotation about each axis). For this project, the only motion of concern is rotation about the z-axis, so the HAB payload RCS will likely only include four thrusters and two solenoids--two thrusters and one solenoid for each spin direction.

Anis (2012) delves into the issue of designing and calculating the stresses on a pressure tank (pp. 449-451), but tank design considerations will not be important for this project due to the high availability and low cost of premade pressure vessels, like the refillable CO_2 cartridges made for paintball guns. Nothnagel (2011) discusses the difficulty of finding solenoid valves that meet the appropriate cost, weight, pressure, and flow requirements, but one important difference between this project and Nothnagel's (2011) is that the TALARIS[3] hopper demands much more from its thrusters than the HAB payload will. For that reason, valves that were inadequate for Nothnagel (2011) may be sufficient for this project.

At the end of the pressure system are the thrusters. Both Anis (2012) and Furumo (2013) discuss thruster nozzle design, but because these calculations are so ubiquitous, they are not

[3] The **T**errestrial **A**rtificial **L**unar **A**nd **R**educed grav**I**ty **S**imulator (TALARIS) was a rocket-controlled vehicle created for testing control algorithms (Nothnagel, 2011).

included here. Furthermore, thruster nozzles may be left as an aside because they serve only to increase the efficiency of a rocket. This project is just concerned with the general feasibility of cold-gas RCS for use in HAB payloads. Pursuing optimization prematurely would only be a distraction from the main goals of this project.

Simulation

The last step in the hardware design process is simulation. Furumo (2013) conducted thruster simulations and compared them to analytical calculations as a way to evaluate thruster design. Nothnagel (2011) goes so far as to create an extensive MATLAB model to simulate a whole maneuver of the TALARIS vehicle--including the change in temperature inside the pressure tank and the flow rate of the gas through the system (pp. 51-66). In the case of this project, the only simulation that might be done is described above in the discussion about propellent requirements. In fact, the main reason for doing this project is that this problem is much harder to simulate than to build and test directly.

Software Control

Understanding the Problematic Motion

Before much effort is put into developing the flight controller software, the problematic motion must be characterized and understood as much as possible. After processing data from an attitude sensor during flight, Nirmal et al. (2016) were able to determine the range of motion the payload underwent during flight (p. 10). They also found that there was much more turbulent air in a layer close to the ground (below 19 km), above which there was almost no turbulence at all.

Quine et al. (2002) note that the wind at float altitude can cause "oscillations that can be as large as 1° in both azimuth and elevation" (p. 620). The difference in concern about one-degree oscillations likely reflects a difference in mission objectives.

Quine et al. (2002) explain that their chosen method of stabilization--motors that compensate for the motion of the gondola--actually contributes to the unwanted motion of the gondola, complicating the stabilization procedures and requiring additional hardware to dampen the motion (p. 620). Mao (2014) and Quine et al. (2002) address this issue and present solutions (Mao, 2014, pp. 100-141; Quine et al., 2002, p. 620).

Theoretical Control Model

A significant chunk of Mao's dissertation(2014) is devoted to understanding the mathematics and algorithms involved in controlling a small satellite or HAB payload (pp. 6-50). Also discussing the computational problems of pointing a payload, albeit in far less detail, Nirmal et al. (2016) split the problem of stabilizing the payload into three parts: determining the attitude, calculating the desired attitude, and moving the payload as needed (pp. 4-5). Quine et al. (2002) describe their methods of estimating orientation and correcting any discrepancy using Proportional-Integral-Derivative (PID) algorithms (pp. 625-627). Due to the recent emergence of cheap, remote-controlled quadcopter drones and associated components, a good deal of the work that Quine et al. did to derive their own PID algorithm may not be necessary today.

Hardware Integration

Once the problematic payload motion has been characterized and a theoretical framework for understanding the payload control process has been established, optimized design of the mechanisms and software can begin. Mao (2014) describes the mechanics and control systems used for a HAB payload (pp. 100-141). Again, Nirmal et al. (2016) present a similar but greatly abbreviated solution to the same problem (pp. 4-5). Mao (2014), Nirmal et al. (2016), and Quine et al. (2002) use sun-tracking devices to help establish a reference from which to measure the movement of the payload. Some of them also use inertial measurement units (IMUs) because the sun trackers tend to have a very small field of view (Nirmal et al., 2016, pp. 6-9; Quine et al., 2002, pp. 624-625). Based on the information gathered from the IMU and sun tracker, the attitude of the payload can be calculated.

The next step in the process of correcting the motion of the payload is to calculate the amount of movement necessary to maintain the desired attitude. Nirmal et al. (2016) and Quine et al. (2002) used PID algorithms for this purpose (Nirmal, 2016, pp. 11-14; Quine et al., 2002, p. 627). Quine et al. (2002, pp. 625-627) worked with more outdated technology but explained their algorithms and calculations on this subject in greater depth than did Furumo et al. The final step in the hardware integration process is to send commands to the stabilization system to execute the desired maneuver. Nothnagel (2011, pp. 104-105) notes, however, that executing these commands involves complications--namely that there are inevitable delays between when the signal is sent and when the valves execute the command sent to them.

Prototyping and Testing

The final step in the process of building a working cold-gas RCS stabilization system (before it can be used in the field) is to test a prototype. A prototype or series of prototypes must be built to test and debug the design. This stage may prove to be the longest and most problematic because it marks the transition from the safe, predictable world of theory into complex, chaotic reality. Due to time constraints, the prototyping stage is the final stage encompassed by this project. Perhaps someday an actual HAB payload equipped with some type of cold-gas RCS could actually be flown and used to conduct experiments.

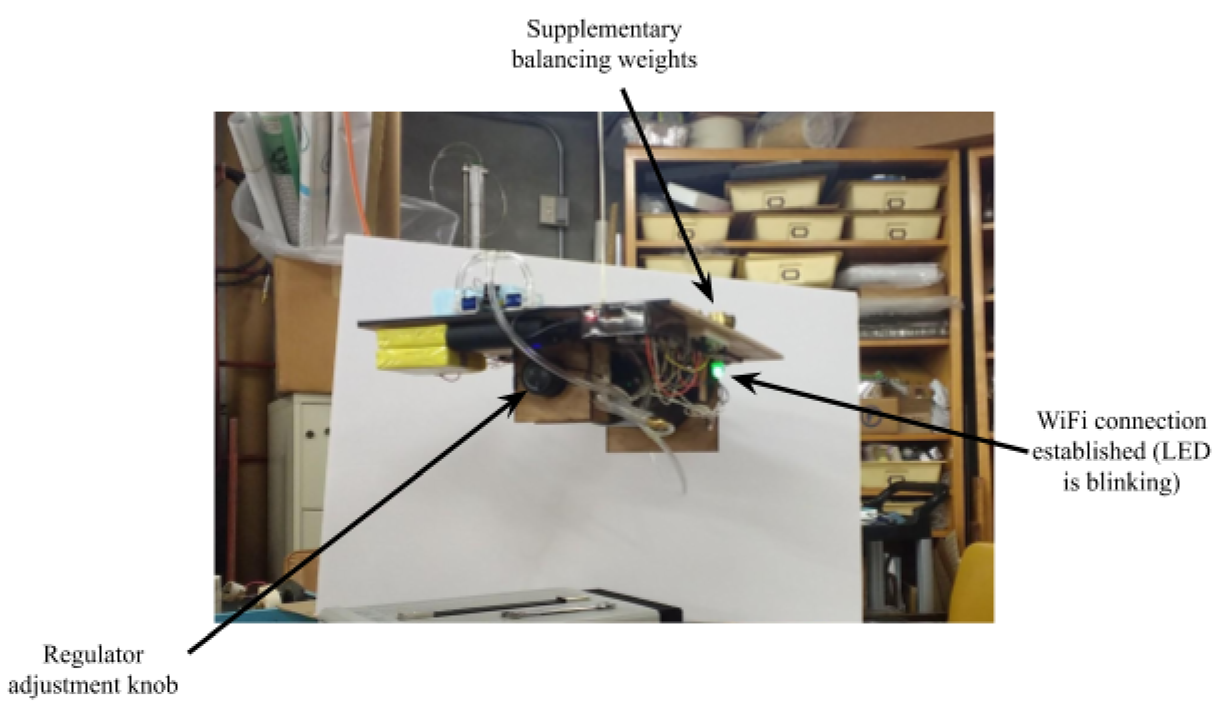
Furumo (2013) ran into several problems during the prototyping phase. The solenoid valves chosen did not allow enough flow to make the thruster nozzles work correctly (Furumo, 2013, p. 76). Also, regarding the creation of a frictionless test stand to simulate flight conditions, Furumo (2013) remarks that "too much time was spent trying to create a frictionless testbed, but in the end, it proved to be impossible" (p. 77). Nirmal et al. (2016) reported fewer challenges (pp. 14-18). The reason is left to speculation, but it may be because Nirmal et al. spent far more time in the planning and design portion of the project than it appears that Furumo (2013) did. Nothnagel (2011) also experienced problems in the prototyping phase. In an early test of the thrusters, the gas cylinder regulator restricted the gas flow and caused problems in much the same way that Furumo's (2013) solenoids caused problems (Nothnagel, 2011, p. 77). Nothnagel (2011) also found that the chosen solenoids had a minimum pulse time which had to be taken into account (p. 82).

One of the main reasons that the prototyping phase is left to the very end is that changing hardware tends to be much harder than changing a calculation or piece of code. Ultimately, the possibility of correcting problems--particularly hardware problems--that arise during the prototyping phase depends entirely on how much time is available for troubleshooting the system, acquiring new parts, etc.

Plans and System Design

There will be three parts of this project that can be separately identified but which are all inherently interrelated: (1) designing the hardware, which mostly consists of the systems required for containing and controlling the propellent; (2) writing control algorithms, which involves understanding the motion to be counteracted and writing code that can interface with the hardware; and (3) prototyping, building, and testing, which will be last steps and the trial-by-fire of the understanding previously gained.

Stabilizer Prototype, Testing Setup

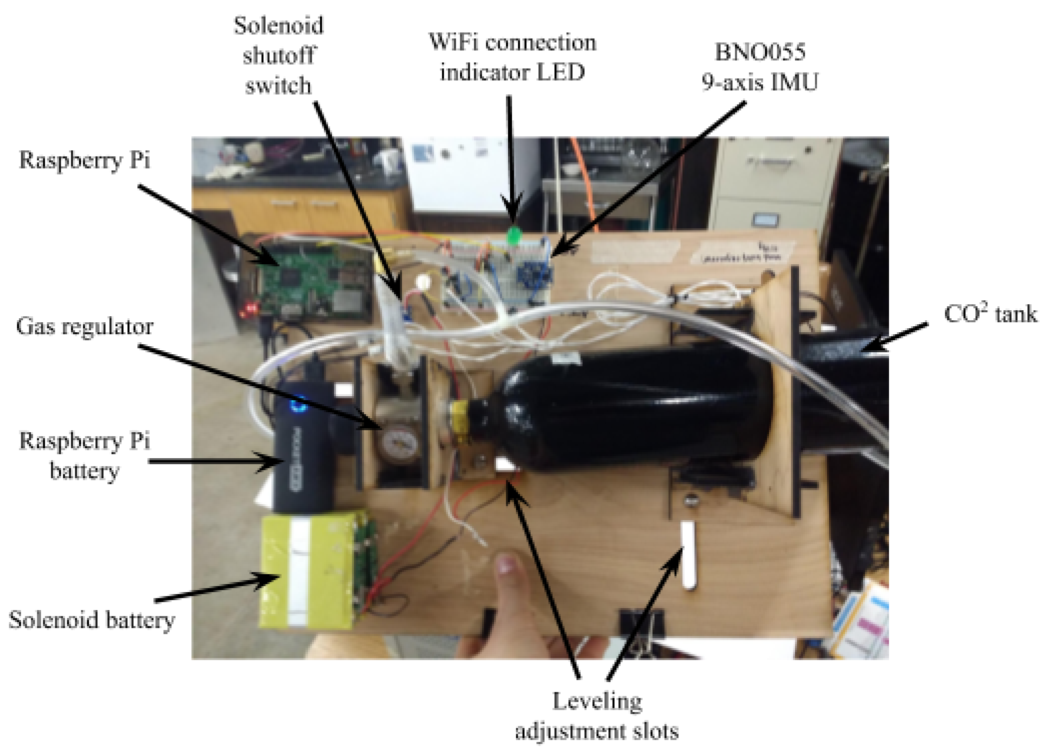


Hardware

Pressure system

In the sport of paintball, small plastic capsules filled with dye are fired from guns with sufficient velocity to rupture the plastic shell and leave a splotch of dye marking the location that was hit. The capsules are propelled by pressurized gas stored in canisters attached to the guns. Because these canisters must be fairly small, light, and inexpensive while safely containing gas at very high pressure (several hundred times atmospheric pressure), they are well suited to the

Stabilizer Prototype, Bottom Side



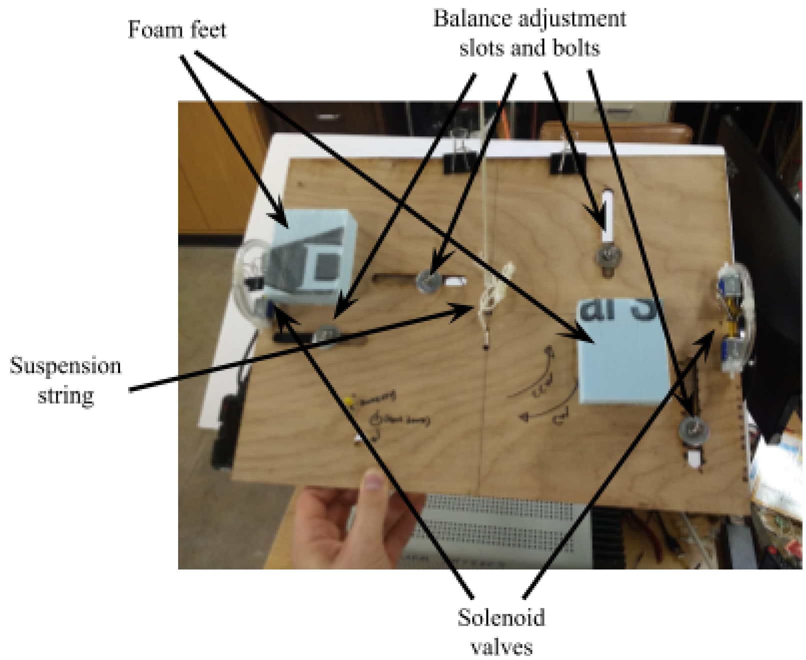
needs of this project. To control the flow of this pressurized gas, small, generic solenoid valves (which are very common and inexpensive) were selected. Due to the small size and low cost of the valves, they cannot handle much pressure without leaking, so a regulator is needed to maintain a low pressure between the output of the gas reservoir and the input of the valves. However, such a regulator--particularly one that will fit on a paintball tank--is difficult to find. Fortunately, because CO_2 is often used by hobbyist brewers to add carbonation to home-brewed beverages, a few home-brewery supply companies carry regulators specifically made to attach to paintball gun tanks. From then on, the only task left was to connect all the pressure components together with 3mm hose and barbed fittings.

After establishing that a paintball CO_2 tank can be used as the pressure reservoir, one must determine what capacity is needed. The most flexible approach for calculating the system's requirements is to derive a formula analytically. The stabilization system functions much like a rocket, the only difference being that the stabilizer changes its *angular* velocity while a conventional rocket changes its *linear* velocity. While this difference is mathematically simple to account for, it is difficult to get into a form that is easy to use in practice. The main problem with deriving a rotational equivalent of the classic rocket equation is that the rotational equivalent of mass, moment of inertia, is difficult to calculate accurately. In practice, considering all the design constraints placed on HAB payloads, it may be more practical to build a series of prototypes with the goal of maximizing the system's capabilities than to try to design the system to meet a precise, precalculated set of requirements.

Frame

During testing, the entire system hung suspended by a string with no other physical connections to transfer data, power, propellent, etc. The frame, therefore, must be able to hold all necessary components. It was made from ¼" fiberboard panelling cut to shape on the University

Stabilizer Prototype, Top Side



of Central Arkansas Physics Department laser cutter[4]. It consisted of a wide, slotted base and two brackets (for holding the pressure reservoir attached to the regulator). The brackets were bolted to the base via the slots and could be adjusted (to change the center of mass of the system) by loosening the bolts and shifting the brackets in the slots. All the other, smaller components were attached to the base using double-sided tape.

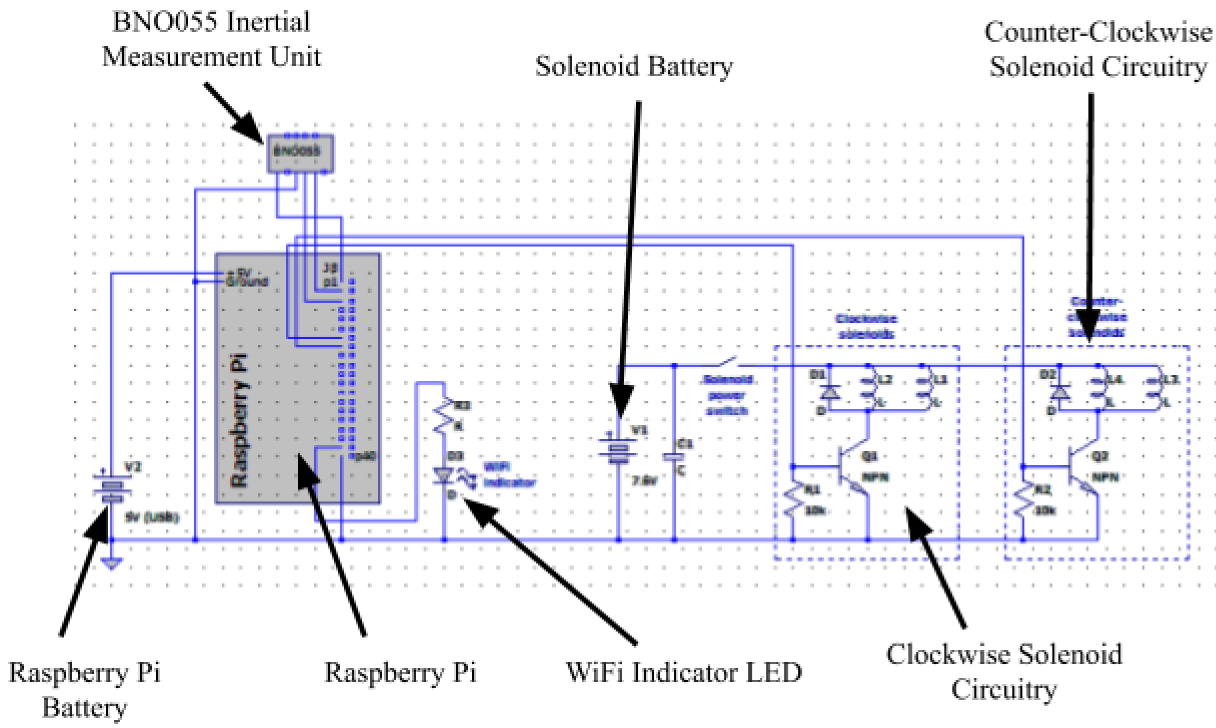
Electrical components

A Raspberry Pi (RPi), a small computer very similar to those in smartphones, was chosen as the main processor because it is provides an easy interface to electrical circuits via

[4] Laser cutter designs are provided in the appendix.

general-purpose input/output (GPIO) pins and a Python module (RPi.GPIO) for controlling

Stabilizer Prototype Circuitry



them. Several attempts were made to power both the RPi and the solenoid valves with the same battery and a ubiquitous linear voltage regulator. Unfortunately, the combination of the load from the solenoids, the dropout[5] of the voltage regulator, and the strict power supply requirements of the RPi necessitated the addition of a second, separate battery for powering the RPi. A total of four solenoid valves were used--two facing opposite directions on both ends of the payload. The solenoids required more power to start than the RPi's GPIO pins could provide,

[5] Linear voltage regulators do not maintain their exact rated voltage on their output. Depending on the supply voltage and the load, the output may drop below its rating by several millivolts. For many projects, this drop in voltage is fine, but in the case of this project it was unacceptable.

so each pair of solenoids was switched with a ZTX1049A power transistor controlled, in turn, by a GPIO pin. Additionally, flyback diodes were added in parallel with the solenoids to protect the other components from harmful voltage spikes caused by the dynamic, inductive loads. Lastly, a decoupling capacitor was added between the RPi's power and ground rails, and an LED was attached to one of the GPIO pins. This LED was used to indicate whether or not the RPi had established a connection to WiFi.

Software

The flight controller was written in Python, and it served the following functions: (1) start the controller and provide minimal feedback while the RPi was running headless[6]; (2) control the solenoid valves to efficiently correct the motion and attitude of the system without too much over-correction; and (3) collect data from the IMU to determine the heading and rate of rotation. All code is reproduced in the appendix.

Headless Setup

While it is helpful for projects like this to be able to use the RPi in headless mode without the burden of bulky human-interface devices, a clear drawback to this mode of operation is that it is difficult to know that the RPi has booted up correctly and the code is running smoothly. To take full advantage of headless mode, a bit of setup must be done first. Three system files on the RPi must be edited: (1) the local WiFi information must be added to wpa_supplicant.conf, the WiFi configuration file; (2) a startup script should be added to rc.local; (3) and an empty file called "ssh" must be added to the folder /boot/ (Setting up a Raspberry Pi headless, n.d.).

[6] The RPi can boot up and run normally without a monitor, keyboard, or mouse. In this case it is said to be running "headless".

Serial Interface

The RPi ran a Python program which collected information from a motion sensor, called an inertial measurement unit (IMU) and controlled the solenoid valves. The BNO055 breakout board[7] from Adafruit Industries was chosen as the IMU because of its wide functionality, the thorough support offered for it by Adafruit, and the ease of interface via an I^2C serial bus. Unfortunately, it proved to be quite difficult to establish a reliable I^2C communication between the RPi and the BNO055. The reason was that the BNO055 implements a function of the I^2C protocol called "clock stretching" which the I^2C driver on the RPi does not support. It proved possible to acquire a single dimension of data with only a few hacks, but in the future, it would be better to find an IMU that uses a different serial protocol like SPI or UART.

The RPi is equipped to communicate natively using several common serial protocols including I^2C, UART, SPI, and PCM. Many of these protocols also have corresponding Python modules (i.e., SMBus for I^2C, and Serial for UART) which allow even easier use. These systems have some limitations, however. For example, while the SPI protocol works, there are not really any good Python modules for implementing it. This lack of a standard SPI module for Python means that the user usually ends up having to bit-bang the protocol themselves, and while manageable, the additional work is irritatingly time consuming.

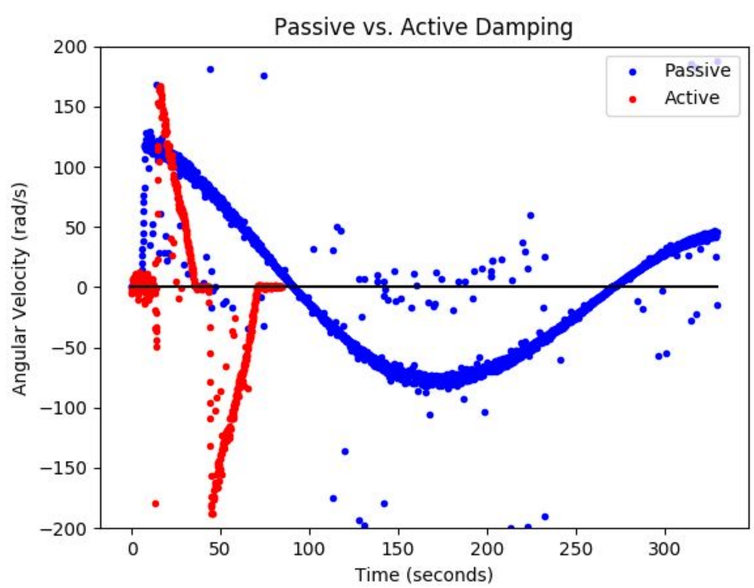
[7] The BNO055 is a microchip. It can be useful for hobbyists, but it is so small that connecting wires can prove to be very difficult. Many hobbyist electronics suppliers offer microchips already soldered to "breakout boards" which make it much easier to use these tiny components.

Controller Program

After everything else was set up, the last part of the last part of the project--writing the controller program--was relatively easy. The controller program simply collected angular position data from the IMU, calculated angular velocity from that data, took the average of several readings to get reliable figures, and decided whether to open solenoid valves to stop any rotation.

Results and Conclusion

After setting the system up as described, the system was able to stabilize its rotation[8] --slowing from a rate of rotation of nearly 30 RPM to a complete stop in around ten seconds. In the chart below, the red points were taken while the stabilization system was turned on, while the



blue points were taken with the stabilization off and the system's motion was damped only by air

[8] A video of the system in action can be found here: https://youtu.be/uUdLrh1b3So

resistance. These results are promising; clearly using a cold-gas RCS stabilization system for use in a small HAB payload is not beyond the realm of possibility. In order to make a system like this to be truly practical for stabilizing a HAB payload, however, a few issues must be solved first: The main problem is the system's mass, a significant portion of which is located in the CO_2 tank. Fortunately, the tank used here was large as paintball tanks go, so pressurized gas tanks around 1% of the mass of the one used here are available fairly cheaply as they are not specialty items. The next heaviest part of the system is the wooden frame, which would likely not be made of wood anyway. Instead, it would be made of rigid foam insulation like most HAB payloads. The next heaviest part after the frame is the battery. The first, most obvious way to cut back here would be to power the entire system with the same battery. This battery would probably be a high-performance Li-Po capable of handling the high current requirements of the solenoids without too much voltage sag. All in all, the prospect of using a low-cost, cold-gas RCS stabilization system on a HAB payload seems well within the realm of possibility. Perhaps this stabilization technique will be able to open doors to new, exciting research possibilities in astronomy or meteorology. Hopefully, this work can serve as a stepping stone to lift, if only by a small amount, those curious people who reach for the stars.

References

Anis, A. (2012). Cold gas propulsion system - An ideal choice for remote sensing small stellites. Retrieved from https://doi.org/10.5772/37149

BNO055 Datasheet. 2016.

Flaten, J., Gosch, C., & Habeck, J. (2015). Techniques for payload stabilization for improved photography during stratospheric balloon flights. Retrieved from http://via.library.depaul.edu/ahac/2015/oral/12/

Furumo, J. G. (2013). Cold-gas propulsion for small satellite attitude control, station keeping, and deorbit. Retrieved from http://www.spacegrant.hawaii.edu/reports/22_SUM12-SP13/S13_JFurumo.pdf

Mao, Y.-T. (2014). *Control theoretic approach for attitude control of high altitude scientific stationary platforms* (Doctoral dissertation). University of California, Berkeley.

Nirmal, K., Sreejith, A. G., Mathew, J., Sarpotdar, M., Suresh, A., Prakash, A., … Murthy, J. (2016). Pointing system for the balloon-borne astronomical payloads. *Journal of Astronomical Telescopes, Instruments, and Systems*, *2*(4), https://doi.org/10.1117/1.JATIS.2.4.047001

Nothnagel, S. L. (2011). *Development of a cold gas propulsion system for the TALARIS hopper* (Master's thesis). Massachusetts Institute of Technology, Boston, MA.

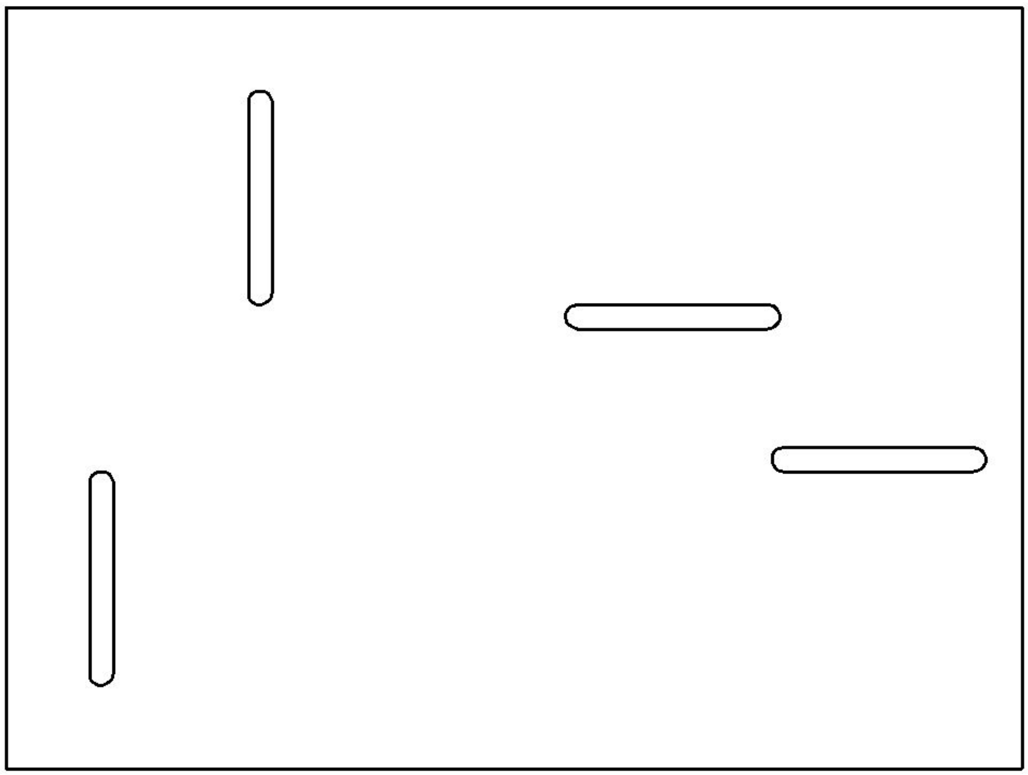
Quine, B. M., Strong, K., & Wiacek, A. (2002). Scanning the earth's limb from a high-altitude balloon: The development and flight of a new balloon-based pointing system. *Journal of*

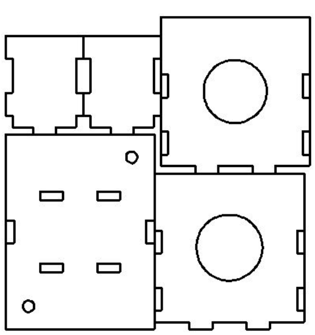
Atmospheric & Oceanic Technology, *19*(5), 618–632. https://doi.org/10.1175/1520-0426(2002)019<0618:STESLF>2.0.CO;2

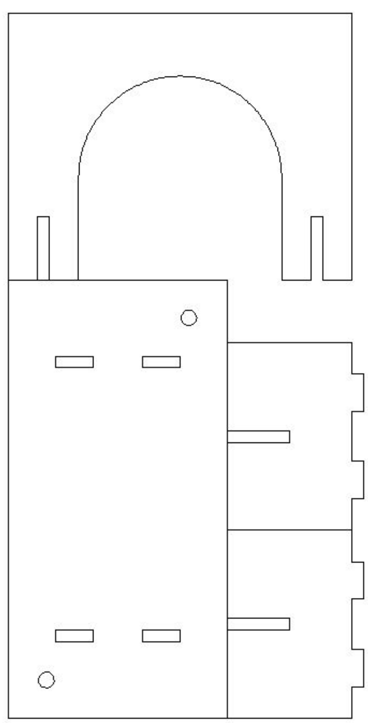
Setting up a raspberry pi headless. (n.d.). Retrieved April 11, 2019, from https://www.raspberrypi.org/documentation/configuration/wireless/headless.md

ZTX1049A Datasheet. 2013.

Appendix A: Laser-Cut Frame Designs (not to scale)







Appendix B: Python Code

Headless Startup Script

```
#!/usr/bin/env python3
'''
Blink an LED when connected to WiFi.
Additionally, if run in a terminal, provide text output.

Russell Jeffery
7 March 2019
'''


import RPi.GPIO as gpio
import os, time


def blink(pin, period):
	gpio.output(pin, True)
	time.sleep(period/2)
	gpio.output(pin, False)
	time.sleep(period/2)

def getip(lines):
	# Parse output of ifconfig to get this machine's ip.
	ip = lines[1]
	ip = ip.split()[1]
	ip = ip.split(sep=':')[1]
	return ip


def main():
	try:
	gpio.setmode(gpio.BOARD)

	led = 37
	gpio.setup(led, gpio.OUT)


	while True:
	#-------------------------------------------------------
```

```
        # Check the connection to Wi-Fi
        #-----------------------------------------------------
        os.system('clear')
        wlan0 = os.popen('ifconfig wlan0')

        # Count the lines; 7 -> not connected, 9 -> connected.
        lines = []
        line = wlan0.readline()
        while line != '':
        lines.append(line)
        line = wlan0.readline()

        if len(lines) == 9:
        print('Connected to WiFi')
        getip(lines)
        blink(led, 0.5)
        else:
        print('Searching for WiFi...')

        blink(led, 0.1)


        finally:
        gpio.cleanup()


if __name__ == "__main__":
        main()
```

Controller Program

```
#!/usr/bin/env python3
'''
Firmware for the HAB payload stabilizer.

Russell Jeffery
7 March 2019
'''

import RPi.GPIO as gpio
import smbus, time, os
```

```
def main():
	try:
		#-------------------------------------------------
		# Set up the interfaces.
		#-------------------------------------------------
		# Set up pins to control the solenoid valves.
		gpio.setmode(gpio.BOARD)
		spin_ccw = 11
		spin_cw = 13
		gpio.setup(spin_cw, gpio.OUT)
		gpio.setup(spin_ccw, gpio.OUT)

		# Setup control interface for BNO055.
		bus = smbus.SMBus(1)
		dev = 0x28 # Device address on the I2C bus.
		bus.write_byte_data(dev, 0x3d, 0x0c) # Set sensor mode in the config register.
		msg = bus.read_byte_data(dev, 0x3d)
		assert msg == 0x0c,"Device returned {} not 0x0c as expected.".format(msg)

		print('Setup complete.')


		#-------------------------------------------------
		# Begin stabilization.
		#-------------------------------------------------
		tupdate = time.time()
		t = 0
		tprev = 0
		tsum = 0
		tct = 0

		x = 0
		xprev = 0
		xsum = 0
		xct = 0

		dx = 0
		dxsum = 0
		dxct = 0

		while True:
```

```
# Get the raw data.
t = time.time()
xlsb = bus.read_byte_data(dev, 0x1a)
xmsb = bus.read_byte_data(dev, 0x1b)

# Calculate time, angle, and angular velocity.
dt = t - tprev
tprev = t
tsum += dt
tct += 1

xprev = x
x = ((xmsb << 8) + xlsb) / 16
xsum += x
xct += 1

dx = (x - xprev) / dt
dxsum += dx
dxct += 1

if (t - tupdate) >= 0.1: # Every tenth of a second, update the control status.
os.system('clear')

# time
tavg = tsum / tct
dtg, dtl = str(round(tavg, 4)).split('.')
print('dt= {dtg: >4}.{dtl: <8}'.format(dtg=dtg, dtl=dtl))
tsum = 0
tct = 0

# angle
xavg = xsum / xct
xg, xl = str(round(xavg, 4)).split('.')
print('x=  {xg: >4}.{xl: <8}'.format(xg=xg, xl=xl))
xsum = 0
xct = 0

# first derivative
dxavg = dxsum / dxct
dxg, dxl = str(round(dxavg, 4)).split('.')
print('dx= {dxg: >4}.{dxl: <8}'.format(dxg=dxg, dxl=dxl))
dxsum = 0
dxct = 0
```

```
        tupdate = time.time()

        # Update the states of the solenoid valves.
        xtarget = 270
        xlim = 10
        xdiff = 0
        dxlim = 7

        if dx > dxlim:
            print('CW solenoid on.')
            gpio.output(spin_cw, True)
        elif dx < -dxlim:
            print('CCW solenoid on.')
            gpio.output(spin_ccw, True)
        else:
            gpio.output(spin_cw, False)
            gpio.output(spin_ccw, False)


        finally:
        print('Maneuver complete. Cleaning up.')
        gpio.cleanup()
        bus.close()


if __name__ == "__main__":
            main()
```

Appendix C: Datasheets

Datasheets included:

- BNO055
- ZTX1049A

Data sheet

BNO055
Intelligent 9-axis absolute orientation sensor

Bosch Sensortec

BOSCH
Invented for life

BNO055: data sheet

Document revision	1.4
Document release date	June 2016
Document number	BST-BNO055-DS000-14
Technical reference code(s)	0 273 141 209
Notes	Data in this document are subject to change without notice. Product photos and pictures are for illustration purposes only and may differ from the real product's appearance.

BNO055

INTELLIGENT ABSOLUTE ORIENTATION SENSOR, 9-AXIS SENSOR FUSION ALL-IN-ONE WINDOWS 8.x COMPLIANT SENSOR HUB

Basic Description

Key features:

• Outputs fused sensor data	Quaternion, Euler angles, Rotation vector, Linear acceleration, Gravity, Heading
• 3 sensors in one device	an advanced triaxial 16bit gyroscope, a versatile, leading edge triaxial 14bit accelerometer and a full performance geomagnetic sensor
• Small package	LGA package 28 pins Footprint 3.8 x 5.2 mm², height 1.13 mm²
• Power Management	Intelligent Power Management: normal, low power and suspend mode available
• Common voltage supplies	V_{DD} voltage range: 2.4V to 3.6V
• Digital interface	HID-I2C (Windows 8 compatible), I²C, UART V_{DDIO} voltage range: 1.7V to 3.6V
• Consumer electronics suite	MSL1, RoHS compliant, halogen-free Operating temperature: -40°C ... +85°C

Key features of integrated sensors:

Accelerometer features

• Programmable functionality	Acceleration ranges ±2g/±4g/±8g/±16g Low-pass filter bandwidths 1kHz - <8Hz Operation modes: - Normal - Suspend - Low power - Standby - Deep suspend
• On-chip interrupt controller	Motion-triggered interrupt-signal generation for - any-motion (slope) detection - slow or no motion recognition - high-g detection

Gyroscope features

• Programmable functionality	Ranges switchable from ±125°/s to ±2000°/s Low-pass filter bandwidths 523Hz - 12Hz Operation modes: - Normal - Fast power up - Deep suspend - Suspend - Advanced power save
• On-chip interrupt controller	Motion-triggered interrupt-signal generation for - any-motion (slope) detection - high rate

Magnetometer features

• Flexible functionality	Magnetic field range typical ±1300μT (x-, y-axis); ±2500μT (z-axis) Magnetic field resolution of ~0.3μT Operating modes: - Low power - Regular - Enhanced regular - High Accuracy Power modes: - Normal - Sleep - Suspend - Force

Typical applications

- Navigation
- Robotics
- Fitness and well-being
- Augmented reality
- Context awareness
- Tablets and ultra-books

General description

The BNO055 is a System in Package (SiP), integrating a triaxial 14-bit accelerometer, a triaxial 16-bit gyroscope with a range of ±2000 degrees per second, a triaxial geomagnetic sensor and a 32-bit cortex M0+ microcontroller running Bosch Sensortec sensor fusion software, in a single package.

The corresponding chip-sets are integrated into one single 28-pin LGA 3.8mm x 5.2mm x 1.1 mm housing. For optimum system integration the BNO055 is equipped with digital bi-directional I^2C and UART interfaces. The I^2C interface can be programmed to run with the HID-I2C protocol turning the BNO055 into a plug-and-play sensor hub solution for devices running the Windows 8.0 or 8.1 operating system.

Contents

Table of Figures

List of Tables

Specification

If not stated otherwise, the given values are over lifetime and full performance temperature and voltage ranges, minimum/maximum values are ±3 sigma.

1.1 Electrical specification

Table 0-1: Electrical parameter specification

Operating Conditions BNO055						
Parameter	**Symbol**	**Condition**	**Min**	**Typ**	**Max**	**Unit**
Supply Voltage (only Sensors)	V_{DD}	--	2.4	--	3.6	V
Supply Voltage (µC and I/O Domain)	V_{DDIO}	--	1.7	--	3.6	V
Voltage Input Low Level (UART, I2C)	V_{DDIO_VIL}	V_{DDIO} = 1.7-2.7V	--	--	0.25 V_{DDIO}	V
		V_{DDIO} = 2.7-3.6V	--	--	0.3 V_{DDIO}	V
Voltage Input High Level (UART, I2C)	V_{DDIO_VIH}	V_{DDIO} = 1.7-2.7V	0.7 V_{DDIO}	--	--	V
		V_{DDIO} = 2.7-3.6V	0.55 V_{DDIO}	--	--	V
Voltage Output Low Level (UART, I2C)	V_{DDIO_VOL}	V_{DDIO} > 3V , I_{OL} =20mA	--	0.1 V_{DDIO}	0.2 V_{DDIO}	V
Voltage Output High Level (UART, I2C)	V_{DDIO_VOH}	V_{DDIO} > 3V , I_{OH} =10mA	0.8 V_{DDIO}	0.9 V_{DDIO}	--	V
POR Voltage threshold on VDDIO-IN rising	V_{DDIO_POT+}	V_{DDIO} falls at 1V/ms or slower	--	1.45	--	V
POR Voltage threshold on VDDIO-IN falling	V_{DDIO_POT-}		--	0.99	--	V
Operating Temperature	T_A	--	-40	--	+85	°C
Total supply current normal mode at T_A (9DOF @100Hz output data rate)	I_{DD} + I_{DDIO}	V_{DD} = 3V, V_{DDIO} = 2.5V	--	--	12.3	mA
Total supply current Low power mode at T_A	I_{DD_LPM}	V_{DD} = 3V, V_{DDIO} = 2.5V	0.33	2.72#		mA
Total supply current suspend mode at T_A	I_{DD_SuM}	V_{DD} = 3V, V_{DDIO} = 2.5V	--	--	0.04*	mA

80% suspend mode and 20% normal mode with 9DOF @100Hz output data rate
* using I2C as communication protocol

1.2 Electrical and physical characteristics, measurement performance

Table 0-2: Electrical characteristics BNO055

OPERATING CONDITIONS BNO055						
Parameter	**Symbol**	**Condition**	**Min**	**Typ**	**Max**	**Unit**
Start-Up time	T_{Sup}	From Off to configuration mode		400		ms
POR time	T_{POR}	From Reset to Config mode		650		ms
Data Rate	DR	s. Par. Fusion Output data rates				
Data rate tolerance 9DOF @100Hz output data rate (if internal oscillator is used)	DR_{tol}			±1		%

OPERATING CONDITIONS ACCELEROMETER						
Parameter	**Symbol**	**Condition**	**Min**	**Typ**	**Max**	**Units**
Acceleration Range	g_{FS2g}	Selectable via serial digital interface		±2		g
	g_{FS4g}			±4		g
	g_{FS8g}			±8		g
	g_{FS16g}			±16		g

OUTPUT SIGNAL ACCELEROMETER (ACCELEROMETER ONLY MODE)						
Parameter	**Symbol**	**Condition**	**Min**	**Typ**	**Max**	**Units**
Sensitivity	S	All g_{FSXg} Values, T_A=25°C		1		LSB/mg
Sensitivity tolerance	S_{tol}	Ta=25°C, g_{FS2g}		±1	±4	%
Sensitivity Temperature Drift	TCS	g_{FS2g}, Nominal V_{DD} supplies, Temp operating conditions		±0.03		%/K
Sensitivity Supply Volt. Drift	S_{VDD}	g_{FS2g}, T_A=25°C, $V_{DD\_min} \le V_{DD} \le V_{DD\_max}$		0.065	0.2	%/V
Zero-g Offset (x,y.z)	Off_{xyz}	g_{FS2g}, T_A=25°C, nominal V_{DD} supplies, over life-time	-150	±80	+150	mg
Zero-g Offset Temperature Drift	TCO	g_{FS2g}, Nominal V_{DD} supplies		±1	+/-3.5	mg/K
Zero-g Offset Supply Volt. Drift	Off_{VDD}	g_{FS2g}, T_A=25°C, $V_{DD\_min} \le V_{DD} \le V_{DD\_max}$		1.5	2.5	mg/V
Bandwidth	bw_8	2nd order filter, bandwidth programmable		8		Hz
	bw_{16}			16		Hz
	bw_{31}			31		Hz
	bw_{63}			63		Hz
	bw_{125}			125		Hz
	bw_{250}			250		Hz
	bw_{500}			500		Hz
	bw_{1000}			1,000		Hz

Nonlinearity	NL	best fit straight line, g_{FS2g}		0.5	2	%FS
Output Noise Density	n_{rms}	g_{FS2g}, T_A=25°C Nominal V_{DD} supplies Normal mode		150	190	µg/√Hz

MECHANICAL CHARACTERISTICS ACCELEROMETER

Parameter	Symbol	Condition	Min	Typ	Max	Units
Cross Axis Sensitivity	CAS	relative contribution between any two of the three axes		1	2	%
Alignment Error	E_A	relative to package outline		0.5	2	°

OPERATING CONDITIONS GYROSCOPE

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Rate Range	R_{FS125}	Selectable via serial digital interface		125		°/s
	R_{FS250}			250		°/s
	R_{FS500}			500		°/s
	R_{FS1000}			1,000		°/s
	R_{FS2000}			2,000		°/s

OUTPUT SIGNAL GYROSCOPE (GYRO ONLY MODE)

Sensitivity via register Map	S	Ta=25°C		16.0 900		LSB/°/s rad/s
Sensitivity tolerance	S_{tol}	Ta=25°C, R_{FS2000}	--	±1	±3	%
Sensitivity Change over Temperature	TCS	Nominal V_{DD} supplies -40°C ≤ T_A ≤ +85°C R_{FS2000}		±0.03	±0.07	%/K
Sensitivity Supply Volt. Drift	S_{VDD}	T_A=25°C, V_{DD_min} ≤ V_{DD} ≤ V_{DD_max}		<0.4		%/V
Nonlinearity	NL	best fit straight line R_{FS1000}, R_{FS2000}		±0.05	±0.2	%FS
Zero-rate Offset	Off Ω_x Ω_y and Ω_z	Nominal V_{DD} supplies T_A=25°C, Slow and fast offset cancellation off	-3	±1	+3	°/s
Zero-Ω Offset Change over Temperature	TCO	Nominal V_{DD} supplies -40°C ≤ T_A ≤ +85°C R_{FS2000}		±0.015	±0.03	°/s per K
Zero-Ω Offset Supply Volt. Drift	OffΩ $_{VDD}$	T_A=25°C, V_{DD_min} ≤ V_{DD} ≤ V_{DD_max}		0.1		°/s /V
Output Noise	n_{rms}	rms, BW=47Hz (@ 0.014°/s/√Hz)		0.1	0.3	°/s

Bandwidth BW	f_{-3dB}			523 230 116 64 47 32 23 12		Hz
MECHANICAL CHARACTERISTICS GYROSCOPE						
Cross Axis Sensitivity	CAS	Sensitivity to stimuli in non-sense-direction		±1	±3	%
OPERATING CONDITIONS MAGNETOMETER (MAGNETOMETER ONLY MODE)						
Parameter	**Symbol**	**Condition**	**Min**	**Typ**	**Max**	**Units**
Magnetic field range[1]	Brg,xy	TA=25°C	±1200	±1300		µT
	Brg,z		±2000	±2500		µT
Magnetometer heading accuracy[2]	As heading	30µT horizontal geomagnetic field component, TA=25°C			±2.5	deg
MAGNETOMETER OUTPUT SIGNAL						
Parameter	**Symbol**	**Condition**	**Min**	**Typ**	**Max**	**Unit**
Device Resolution	$D_{res,m}$	T_A=25°C		0.3		µT
Gain error[3]	$G_{err,m}$	After API compensation T_A=25°C Nominal V_{DD} supplies		±5	±8	%
Sensitivity Temperature Drift	TCS_m	After API compensation -40°C ≤ T_A ≤ +85°C Nominal V_{DD} supplies		±0.01	±0.03	%/K
Zero-B offset	OFF_m	T_A=25°C		±40		µT
Zero-B offset[4]	$OFF_{m,cal}$	After calibration in fusion mode -40°C ≤ T_A ≤ +85°C		±2		µT
Zero-B offset Temperature Drift	TCO_m	-40°C ≤ T_A ≤ +85°C		±0.23	±0.37	µT/K
Full-scale Nonlinearity	$NL_{m, FS}$	best fit straight line			1	%FS

[1] Full linear measurement range considering sensor offsets.
[2] The heading accuracy depends on hardware and software. A fully calibrated sensor and ideal tilt compensation are assumed.
[3] Definition: *gain error = ((measured field after API compensation) / (applied field))* – 1
[4] Magnetic zero-B offset assuming calibration in fusion mode. Typical value after applying calibration movements containing various device orientations (typical device usage).

Output Noise	$n_{rms,lp,m,xy}$	Low power preset x, y-axis, T_A=25°C Nominal V_{DD} supplies		1.0		µT
	$n_{rms,lp,m,z}$	Low power preset z-axis, T_A=25°C Nominal V_{DD} supplies		1.4		µT
	$n_{rms,rg,m}$	Regular preset T_A=25°C Nominal V_{DD} supplies		0.6		µT
	$n_{rms,eh,m}$	Enhanced regular preset T_A=25°C Nominal V_{DD} supplies		0.5		µT
	$n_{rms,ha,m}$	High accuracy preset T_A=25°C Nominal V_{DD} supplies		0.3		µT
Power Supply Rejection Rate	$PSRR_m$	T_A=25°C Nominal V_{DD} supplies		±0.5		µT/V

2. Absolute Maximum Ratings

Table 2-1: Absolute maximum ratings (preliminary target values)

Parameter	Symbol	Condition	Min	Max	Units
Voltage at Supply Pin	V_{DD} Pin		-0.3	4.2	V
	V_{DDIO} Pin		-0.3	3.6	V
Voltage at any Logic Pin	$V_{non\text{-}supply}$ Pin		-0.3	V_{DDIO}+0.3	V
Passive Storage Temp. Range	Trps	≤ 65% rel. H.	-50	+150	°C
Mechanical Shock	$MechShock_{200\mu s}$	Duration ≤ 200µs		10,000	g
	$MechShock_{1ms}$	Duration ≤ 1.0ms		2,000	g
	$MechShock_{freefall}$	Free fall onto hard surfaces		1.8	m
ESD	ESD_{HBM}	HBM, at any Pin		2	kV
	ESD_{CDM}	CDM		500	V
	ESD_{MM}	MM		200	V

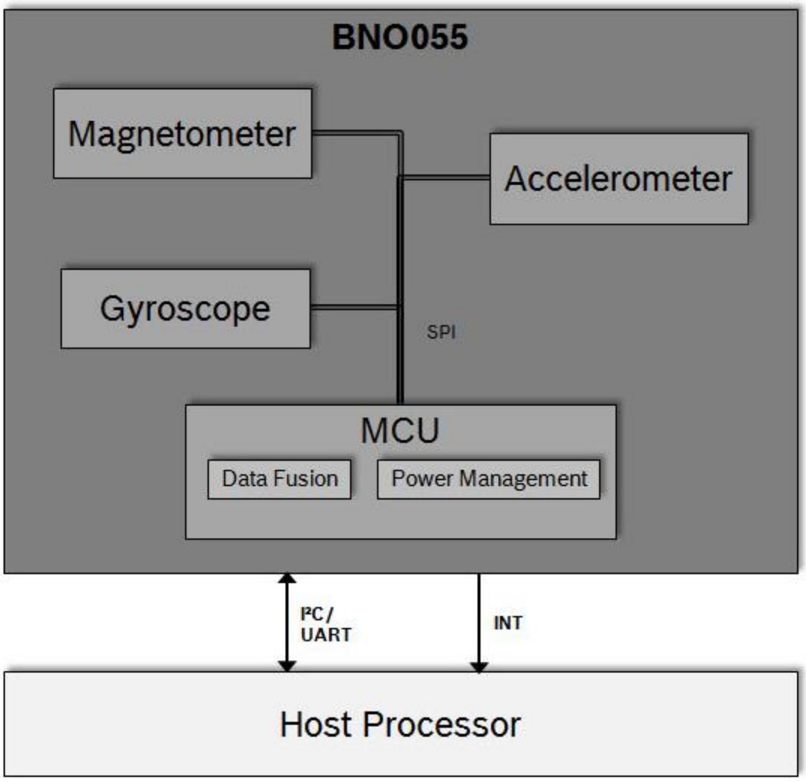
Note:
Stress above these limits may cause damage to the device. Exceeding the specified electrical limits may affect the device reliability or cause malfunction.

3. Functional Description

3.1 Architecture

The following figure shows the basic building blocks of the BNO055 device.

Figure 1: system architecture



3.2 Power management

The BNO055 has two distinct power supply pins:

- V_{DD} is the main power supply for the internal sensors
- V_{DDIO} is a separate power supply pin used for the supply of the µC and the digital interfaces

For the switching sequence of power supply V_{DD} and V_{DDIO} it is mandatory that V_{DD} is powered on and driven to the specified level before or at the same time as V_{DDIO} is powered ON. Otherwise there are no limitations on the voltage levels of both pins relative to each other, as long as they are used within the specified operating range.

The sensor features a power-on reset (POR), initializing the register map with the default values and starting in CONFIG mode. The POR is executed at every power on and can also be triggered either by applying a low signal to the nRESET pin for at least 20ns or by setting the RST_SYS bit in the SYS_TRIGGER register.

The BNO055 can be configured to run in one of the following power modes: normal mode, low power mode, and suspend mode. These power modes are described in more detail in section Power Modes

Power Modes
The BNO055 support three different power modes: Normal mode, Low Power Mode, and Suspend mode.
The power mode can be selected by writing to the PWR_MODE register as defined in the table below. As default at start-up the BNO055 will run in Normal mode.

Table 3-1: power modes selection

Parameter	Value	[Reg Addr]: Reg Value
Power Mode	Normal Mode	[PWR_MODE]: xxxxxx00b
	Low Power Mode	[PWR_MODE]: xxxxxx01b
	Suspend Mode	[PWR_MODE]: xxxxxx10b

3.2.1 Normal Mode

In normal mode all sensors required for the selected operating mode (see section 3.3) are always switched ON. The register map and the internal peripherals of the MCU are always operative in this mode.

3.2.2 Low Power Mode

If no activity (i.e. no motion) is detected for a configurable duration (default 5 seconds), the BNO055 enters the low power mode. In this mode only the accelerometer is active. Once motion is detected (i.e. the accelerometer signals an any-motion interrupt), the system is woken up and normal mode is entered. The following settings are possible.

Table 3-2: Low power modes - Interrupts

Description	Parameter	Value	Reg Value	Restriction
Entering to sleep: NO Motion Interrupt	Detection Type	No Motion	[ACC_NM_SET] : xxxxxxx1b	n/a
		Detection Axis	[ACC_INT_Settings] : bit4-bit2	Shares common bit with Any Motion interrupt axis selection
	Params	Duration	[ACC_NM_SET] : bit6-bit1	n/a
		Threshold	[ACC_NM_THRE] : bit7-bit0	n/a

Description	Parameter	Value	Reg Value
Waking up: Any Motion Interrupt	Detection Type	Detection Axis	[ACC_INT_Settings] : bit4-bit2
	Params	Duration	[ACC_INT_Settings] : bit1-bit0
		Threshold	[ACC_AM_THRES] : bit7-bit0

Additionally, the interrupt pins can also be configured to provide HW interrupt to the host.

The BNO055 is by default configured to have optimum values for entering into sleep and waking up. To restore these values, trigger system reset by setting RST_SYS bit in SYS_TRIGGER register.

There are some limitations to achieve the low power mode performance:

- Only No and Any motion interrupts are applicable and High-G and slow motion interrupts are not applicable in low power mode.
- Low power mode is not applicable where accelerometer is not employed.

3.2.3 Suspend Mode

In suspend mode the system is paused and all the sensors and the microcontroller are put into sleep mode. No values in the register map will be updated in this mode. To exit from suspend mode the mode should be changed by writing to the PWR_MODE register (see Table 3-1).

3.3 Operation Modes

The BNO055 provides a variety of output signals, which can be chosen by selecting the appropriate operation mode. The table below lists the different modes and the available sensor signals.

Table 3-3: Operating modes overview

Operating Mode		Available sensor signals			Fusion Data	
		Accel	**Mag**	**Gyro**	**Relative orientation**	**Absolute orientation**
	CONFIGMODE	-	-	-	-	-
Non-fusionmodes	ACCONLY	X	-	-	-	-
	MAGONLY	-	X	-	-	-
	GYROONLY	-	-	X	-	-
	ACCMAG	X	X	-	-	-
	ACCGYRO	X	-	X	-	-
	MAGGYRO	-	X	X	-	-
	AMG	X	X	X	-	-
Fusion modes	IMU	X	-	X	X	-
	COMPASS	X	X	-	-	X
	M4G	X	X		X	-
	NDOF_FMC_OFF	X	X	X	-	X
	NDOF	X	X	X	-	X

The default operation mode after power-on is CONFIGMODE.

When the user changes to another operation mode, the sensors which are required in that particular sensor mode are powered, while the sensors whose signals are not required are set to suspend mode.

The BNO055 sets the following default settings for the sensors. The user can overwrite these settings in the register map when in CONFIGMODE.

Table 3-4: Default sensor settings

Sensor	Range	Bandwidth
Accelerometer	4G	62.5 Hz
Magnetometer	NA	10 Hz
Gyroscope	2000 dps	32 Hz

In any mode, the sensor data are available in the data register based on the unit selected. The axis of the data is configured based on the axis-remap register configuration.

The operating mode can be selected by writing to the OPR_MODE register, possible register values and the corresponding operating modes are shown in the table below.

Table 3-5: operating modes selection

Parameter	Value	[Reg Addr]: Reg Value
CONFIG MODE	CONFIGMODE	[OPR_MODE]: xxxx0000b
Non-Fusion Mode	ACCONLY	[OPR_MODE]: xxxx0001b
	MAGONLY	[OPR_MODE]: xxxx0010b
	GYROONLY	[OPR_MODE]: xxxx0011b
	ACCMAG	[OPR_MODE]: xxxx0100b
	ACCGYRO	[OPR_MODE]: xxxx0101b
	MAGGYRO	[OPR_MODE]: xxxx0110b
	AMG	[OPR_MODE]: xxxx0111b
Fusion Mode	IMU	[OPR_MODE]: xxxx1000b
	COMPASS	[OPR_MODE]: xxxx1001b
	M4G	[OPR_MODE]: xxxx1010b
	NDOF_FMC_OFF	[OPR_MODE]: xxxx1011b
	NDOF	[OPR_MODE]: xxxx1100b

Table 3-6 below shows the time required to switch between CONFIGMODE and the other operating modes.

Table 3-6: Operating mode switching time

From	To	Switching time
CONFIGMODE	Any operation mode	7ms
Any operation mode	CONFIGMODE	19ms

3.3.1 Config Mode

This mode is used to configure BNO, wherein all output data is reset to zero and sensor fusion is halted. This is the only mode in which all the writable register map entries can be changed. (Exceptions from this rule are the interrupt registers (INT and INT_MSK) and the operation mode register (OPR_MODE), which can be modified in any operation mode.)

As being said, this mode is the default operation mode after power-on or RESET. Any other mode must be chosen to be able to read any sensor data.

3.3.2 Non-Fusion Modes

3.3.2.1 ACCONLY

If the application requires only raw accelerometer data, this mode can be chosen. In this mode the other sensors (magnetometer, gyro) are suspended to lower the power consumption. In this mode, the BNO055 behaves like a stand-alone acceleration sensor.

3.3.2.1 MAGONLY

In MAGONLY mode, the BNO055 behaves like a stand-alone magnetometer, with acceleration sensor and gyroscope being suspended.

3.3.2.2 GYROONLY

In GYROONLY mode, the BNO055 behaves like a stand-alone gyroscope, with acceleration sensor and magnetometer being suspended.

3.3.2.3 ACCMAG

Both accelerometer and magnetometer are switched on, the user can read the data from these two sensors.

3.3.2.4 ACCGYRO

Both accelerometer and gyroscope are switched on; the user can read the data from these two sensors.

3.3.2.5 MAGGYRO

Both magnetometer and gyroscope are switched on, the user can read the data from these two sensors.

3.3.2.6 AMG (ACC-MAG-GYRO)

All three sensors accelerometer, magnetometer and gyroscope are switched on.

3.3.3 Fusion modes

Sensor fusion modes are meant to calculate measures describing the orientation of the device in space. It can be distinguished between non-absolute or relative orientation and absolute orientation. Absolute orientation means orientation of the sensor with respect to the earth and its magnetic field. In other words, absolute orientation sensor fusion modes calculate the direction of the magnetic north pole.
In non-absolute or relative orientation modes, the heading of the sensor can vary depending on how the sensor is placed initially.

All fusion modes provide the heading of the sensor as quaternion data or in Euler angles (roll, pitch and yaw angle). The acceleration sensor is both exposed to the gravity force and to accelerations applied to the sensor due to movement. In fusion modes it is possible to separate the two acceleration sources, and thus the sensor fusion data provides separately linear acceleration (i.e. acceleration that is applied due to movement) and the gravity vector.

3.3.3.1 IMU (Inertial Measurement Unit)

In the IMU mode the relative orientation of the BNO055 in space is calculated from the accelerometer and gyroscope data. The calculation is fast (i.e. high output data rate).

3.3.3.2 COMPASS

The COMPASS mode is intended to measure the magnetic earth field and calculate the geographic direction.
The earth magnetic field is a vector with the horizontal components x,y and the vertical z component. It depends on the position on the globe and natural iron occurrence. For heading calculation (direction of compass pointer) only the horizontal components x and y are used. Therefore the vector components of the earth magnetic field must be transformed in the horizontal plane, which requires the knowledge of the direction of the gravity vector. To summarize, the heading can only be calculated when considering gravity and magnetic field at the same time.

However, the measurement accuracy depends on the stability of the surrounding magnetic field. Furthermore, since the earth magnetic field is usually much smaller than the magnetic fields that occur around and inside electronic devices, the compass mode requires calibration (see chapter 3.10)

3.3.3.3 M4G (Magnet for Gyroscope)

The M4G mode is similar to the IMU mode, but instead of using the gyroscope signal to detect rotation, the changing orientation of the magnetometer in the magnetic field is used. Since the magnetometer has much lower power consumption than the gyroscope, this mode is less power consuming in comparison to the IMU mode. There are no drift effects in this mode which are inherent to the gyroscope.

However, as for compass mode, the measurement accuracy depends on the stability of the surrounding magnetic field.
For this mode no magnetometer calibration is required and also not available.

3.3.3.4 NDOF_FMC_OFF

This fusion mode is same as NDOF mode, but with the Fast Magnetometer Calibration turned ‘OFF’.

3.3.3.5 NDOF

This is a fusion mode with 9 degrees of freedom where the fused absolute orientation data is calculated from accelerometer, gyroscope and the magnetometer. The advantages of combining all three sensors are a fast calculation, resulting in high output data rate, and high robustness from magnetic field distortions. In this mode the Fast Magnetometer calibration is turned ON and thereby resulting in quick calibration of the magnetometer and higher output data accuracy. The current consumption is slightly higher in comparison to the NDOF_FMC_OFF fusion mode.

3.4 Axis remap

The device mounting position should not limit the data output of the BNO055 device. The axis of the device can be re-configured to the new reference axis.

Axis configuration byte: Register Address: **AXIS_MAP_CONFIG**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved		Remapped Z axis value		Remapped Y axis value		Remapped X axis value	

There are two bits are used to configure the axis remap which will define in the following way,

Value	Axis Representation
00	X - Axis
01	Y - Axis
10	Z- Axis
11	Invalid

Also, when user try to configure the same axis to two or more then BNO055 will take this as invalid condition and previous configuration will be restored in the register map. The default value is: X Axis = X, Y Axis = Y and Z Axis = Z (AXIS_REMAP_CONFIG = 0x24).

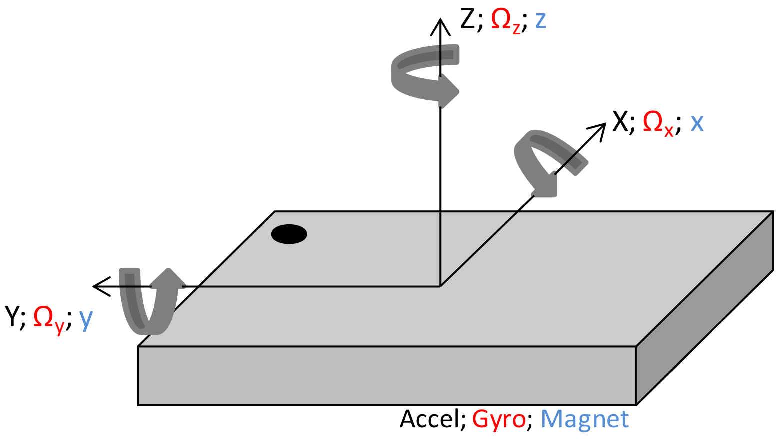
Axis sign configuration byte: Register Address: **AXIS_MAP_SIGN**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved					Remapped X axis sign	Remapped Y axis sign	Remapped Z axis sign

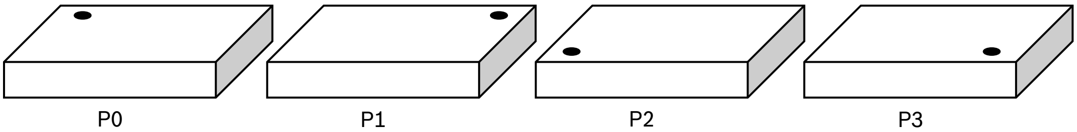
Value	Sign
0	Positive
1	Negative

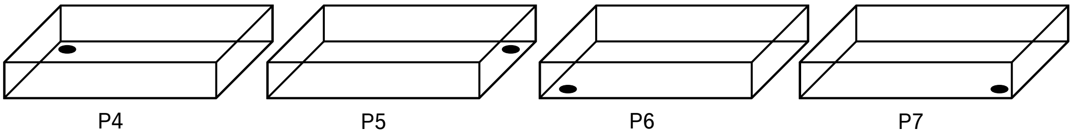
The default value is 0x00.

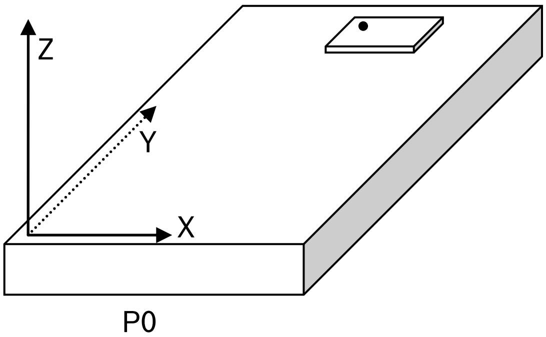
The default values correspond to the following coordinate system



Some example placement for axis vs. register settings:







For the above described placements, following would be the axis configuration parameters.

Placement	AXIS_REMAP_CONFIG	AXIS_REMAP_SIGN
P0	0x21	0x04
P1 (default)	0x24	0x00
P2	0x24	0x06
P3	0x21	0x02
P4	0x24	0x03
P5	0x21	0x01
P6	0x21	0x07
P7	0x24	0x05

3.5 Sensor Configuration

The fusion outputs of the BNO055 are tightly linked with the sensor configuration settings. Due to this fact, the sensor configuration is limited when BNO055 is configured to run in any of the fusion operating mode. In any of the non-fusion modes the configuration settings can be updated by writing to the configuration registers as defined in the following sections.

3.5.1 Default sensor configuration

At power-on the sensors are configured with the default settings as defined in Table 3-8 below.

Table 3-7: Default sensor configuration at power-on

Sensors	Parameters	Value
Accelerometer	Power Mode	NORMAL
	Range	+/- 4g
	Bandwidth	62.5Hz
	Resolution	14 bits
Gyroscope	Power Mode	NORMAL
	Range	2000 °/s
	Bandwidth	32Hz
	Resolution	16 bits
Magnetometer	Power Mode	FORCED
	ODR	20Hz
	XY Repetition	15
	Z Repetition	16
	Resolution x/y/z	13/13/15 bits

3.5.2 Accelerometer configuration

The fusion outputs of the BNO055 are tightly linked with the accelerometer sensor settings. Therefore the configuration possibilities are restricted when running in any of the fusion operating modes. The accelerometer configuration can be changed by writing to the ACC_Config register, Table below shows different Accelerometer configurations

Table 3-8: Accelerometer configurations

Parameter	Values	[Reg Addr]: Reg Value	Restrictions
G Range	2G	[ACC_Config]: xxxxxx00b	Auto controlled in fusion mode
	4G	[ACC_Config]: xxxxxx01b	
	8G	[ACC_Config]: xxxxxx10b	
	16G	[ACC_Config]: xxxxxx11b	
Bandwidth	7.81Hz	[ACC_Config]: xxx000xxb	
	15.63Hz	[ACC_Config]: xxx001xxb	
	31.25Hz	[ACC_Config]: xxx010xxb	
	62.5Hz	[ACC_Config]: xxx011xxb	
	125Hz	[ACC_Config]: xxx100xxb	
	250Hz	[ACC_Config]: xxx101xxb	
	500Hz	[ACC_Config]: xxx110xxb	
	1000Hz	[ACC_Config]: xxx111xxb	
Operation Mode	Normal	[ACC_Config]: 000xxxxxb	
	Suspend	[ACC_Config]: 001xxxxxb	
	Low Power 1	[ACC_Config]: 010xxxxxb	
	Standby	[ACC_Config]: 011xxxxxb	
	Low Power 2	[ACC_Config]: 100xxxxxb	
	Deep Suspend	[ACC_Config]: 101xxxxxb	

The accelerometer sensor operation mode is not configurable by user when BNO power mode is configured as low power mode. BNO rewrites the user configured value to Normal mode when switching from config mode to any BNO operation mode. This used to achieve the BNO low power mode performance.

3.5.3 Gyroscope configuration

The fusion outputs of the BNO055 are tightly linked with the angular rate sensor settings. Therefore the configuration possibilities are restricted when running in any of the fusion operating modes. The gyroscope configuration can be changed by writing to the GYR_Config register, Table below shows different Gyroscope configurations

Table 3-9: Gyroscope configurations

Parameter	Values	[Reg Addr]: Register value	Restrictions
Range	2000 dps	[GYR_Config_0]: xxxxx000b	Auto controlled in fusion mode
	1000 dps	[GYR_Config_0]: xxxxx001b	
	500dps	[GYR_Config_0]: xxxxx010b	
	250 dps	[GYR_Config_0]: xxxxx011b	
	125 dps	[GYR_Config_0]: xxxxx100b	
Bandwidth	523Hz	[GYR_Config_0]: xx000xxxb	
	230Hz	[GYR_Config_0]: xx001xxxb	
	116Hz	[GYR_Config_0]: xx010xxxb	
	47Hz	[GYR_Config_0]: xx011xxxb	
	23Hz	[GYR_Config_0]: xx100xxxb	
	12Hz	[GYR_Config_0]: xx101xxxb	
	64Hz	[GYR_Config_0]: xx110xxxb	
	32Hz	[GYR_Config_0]: xx111xxxb	
Operation Mode	Normal	[GYR_Config_1]: xxxxx000b	
	Fast Power up	[GYR_Config_1]: xxxxx001b	
	Deep Suspend	[GYR_Config_1]: xxxxx010b	
	Suspend	[GYR_Config_1]: xxxxx011b	
	Advanced Powersave	[GYR_Config_1]: xxxxx100b	

3.5.4 Magnetometer configuration

The fusion outputs of the BNO055 are tightly linked with the magnetometer sensor settings. Therefore the configuration possibilities are restricted when running in any of the fusion operating modes. The magnetometer configuration can be changed by writing to the MAG_Config register, Table below shows different Magnetometer configurations.

Table 3-10: Magnetometer configurations

Parameter	Values	[Reg Addr]: Register value	Restrictions
Data output rate	2Hz	[MAG_Config]: xxxxx000b	Auto controlled in fusion mode
	6Hz	[MAG_Config]: xxxxx001b	
	8Hz	[MAG_Config]: xxxxx010b	
	10Hz	[MAG_Config]: xxxxx011b	
	15Hz	[MAG_Config]: xxxxx100b	
	20Hz	[MAG_Config]: xxxxx101b	
	25Hz	[MAG_Config]: xxxxx110b	
	30Hz	[MAG_Config]: xxxxx111b	
Operation Mode	Low Power	[MAG_Config]: xxx00xxxb	
	Regular	[MAG_Config]: xxx01xxxb	
	Enhanced Regular	[MAG_Config]: xxx10xxxb	
	High Accuracy	[MAG_Config]: xxx11xxxb	
Power Mode	Normal	[MAG_Config]: x00xxxxxb	
	Sleep	[MAG_Config]: x01xxxxxb	
	Suspend	[MAG_Config]: x10xxxxxb	
	Force Mode	[MAG_Config]: x11xxxxxb	

3.6 Output data

Depending on the selected operating mode the device will output either un-calibrated sensor data (in non-fusion mode) or calibrated / fused data (in fusion mode), this section describes the output data for each modes.

3.6.1 Unit selection

The measurement units for the various data outputs (regardless of operation mode) can be configured by writing to the UNIT_SEL register as described in Table 3-9.

Table 3-11: unit selection

Data	Units	[Reg Addr]: Register Value
Acceleration, Linear Acceleration, Gravity vector	m/s²	[UNIT_SEL] : xxxxxxx0b
	mg	[UNIT_SEL] : xxxxxxx1b
Magnetic Field Strength	Micro Tesla	NA
Angular Rate	Dps	[UNIT_SEL] : xxxxxx0xb
	Rps	[UNIT_SEL] : xxxxxx1xb
Euler Angles	Degrees	[UNIT_SEL] : xxxxx0xxb
	Radians	[UNIT_SEL] : xxxxx1xxb
Quaternion	Quaternion units	NA
Temperature	°C	[UNIT_SEL] : xxx0xxxxb
	°F	[UNIT_SEL] : xxx1xxxxb

3.6.2 Data output format

The data output format can be selected by writing to the UNIT_SEL register, this allows user to switch between the orientation definition described by Windows and Android operating systems.

Table 3-12: Fusion data output format

Parameter	Values	[Reg Addr]: Register value
Fusion data output format	Windows	[UNIT_SEL]: 0xxxxxxxb
	Android	[UNIT_SEL]: 1xxxxxxxb

The output data format is based on the following convention regarding the rotation angles for roll, pitch and heading / yaw (compare also section 3.4):

Table 3-13: Rotation angle conventions

Rotation angle	Range (Android format)	Range (Windows format)
Pitch	+180° to -180° (turning clockwise decreases values)	-180° to +180° (turning clock-wise increases values)
Roll	-90° to +90° (increasing with increasing inclination)	
Heading / Yaw	0° to 360° (turning clockwise increases values)	

3.6.3 Fusion Output data rates

Table 3-14: Fusion output data rates

BNO055 Operating Mode	Data input rate			Algo calling rate	Data output rate			
	Accel	Mag	Gyro		Accel	Mag	Gyro	Fusion data
IMU	100Hz	NA	100Hz	100Hz	100Hz	NA	100Hz	100Hz
COMPASS	20Hz	20Hz	NA	20Hz	20Hz	20Hz	NA	20Hz
M4G	50Hz	50Hz	NA	50Hz	50Hz	50Hz	NA	50Hz
NDOF_FMC_OFF	100Hz	20Hz	100Hz	100Hz	100Hz	20Hz	100Hz	100Hz
NDOF	100Hz	20Hz	100Hz	100Hz	100Hz	20Hz	100Hz	100Hz

3.6.4 Sensor calibration data

The following section describes the register holding the calibration data of the sensors (see chapter 3.11). The offset and radius data can be read from these registers and stored in the host system, which could be later used to get the correct orientation data after ‘Power on Reset’ of the sensor.

3.6.4.1 Accelerometer offset

The accelerometer offset can be configured in the following registers, shown in the table below. There are 6 bytes required to configure the accelerometer offset (2 bytes for each of the 3 axis X, Y and Z). Configuration will take place only when the user writes the last byte (i.e., ACC_OFFSET_Z_MSB).

Table 3-15: Accelerometer Default-Reg settings

Reg Name	Default Reg Value (Bit 0 – Bit 7)
ACC_OFFSET_X_LSB	0x00
ACC_OFFSET_X_MSB	0x00
ACC_OFFSET_Y_LSB	0x00
ACC_OFFSET_Y_MSB	0x00
ACC_OFFSET_Z_LSB	0x00
ACC_OFFSET_Z_MSB	0x00

The range of the offsets varies based on the G-range of accelerometer sensor.

Table 3-16: Accelerometer G-range settings

Accelerometer G-range	Maximum Offset range in mg
2G	+/- 2000
4G	+/- 4000
8G	+/- 8000
16G	+/- 16000

Table 3-17: Accelerometer Unit settings

Unit	Representation
m/s^2	1 m/s^2 = 100 LSB
mg	1 mg = 1 LSB

3.6.4.2 Magnetometer offset

The magnetometer offset can be configured in the following registers,

Table 3-18: Magnetometer Default-Reg settings

Reg Name	Default Reg Value (Bit 0 – Bit 7)
MAG_OFFSET_X_LSB	0x00
MAG_OFFSET_X_MSB	0x00
MAG_OFFSET_Y_LSB	0x00
MAG_OFFSET_Y_MSB	0x00
MAG_OFFSET_Z_LSB	0x00
MAG_OFFSET_Z_MSB	0x00

There are 6 bytes required to configure the magnetometer offset (bytes (2 bytes for each of the 3 axis X, Y and Z). Configuration will take place only when the user writes the last byte (i.e., MAG_OFFSET_Z_MSB). Therefore the last byte must be written whenever the user wants to changes the configuration. The range of the magnetometer offset is +/-6400 in LSB.

Table 3-19: Magnetometer Unit settings

Unit	Representation
μT	1 μT = 16 LSB

3.6.4.3 Gyroscope offset

The gyroscope offset can be configured in the following registers, shown in the table below

Table 3-20: Gyroscope Default Reg-settings

Reg Name	Default Reg Value (Bit 0 – Bit 7)
GYR_OFFSET_X_LSB	0x00
GYR_OFFSET_X_MSB	0x00
GYR_OFFSET_Y_LSB	0x00
GYR_OFFSET_Y_MSB	0x00
GYR_OFFSET_Z_LSB	0x00
GYR_OFFSET_Z_MSB	0x00

There are 6 bytes required to configure the gyroscope offset (bytes (2 bytes for each of the 3 axis X, Y and Z). Configuration will take place only when the user writes the last byte (i.e., GYR_OFFSET_Z_MSB). Therefore the last byte must be written whenever the user wants to changes the configuration. The range of the offset varies based on the dps-range of gyroscope sensor.

Table 3-21: Gyroscope range settings

Gyroscope dps range	Maximum Offset range in LSB
2000	+/- 32000
1000	+/- 16000
500	+/- 8000
250	+/- 4000
125	+/- 2000

Table 3-22: Gyroscope unit settings

Unit	Representation
Dps	1 Dps = 16 LSB
Rps	1 Rps = 900 LSB

3.6.4.4 Radius

The radius of accelerometer, magnetometer and gyroscope can be configured in the following registers,

Table 3-23: Radius Default-Reg settings

Reg Name	Default Reg Value (Bit 0 – Bit 7)
ACC_RADIUS_LSB	0x00
ACC_RADIUS_MSB	0x00
MAG_RADIUS_LSB	0x00
MAG_RADIUS_MSB	0x00

There are 4 bytes (2 bytes for each accelerometer and magnetometer) to configure the radius. Configuration will take place only when user writes to the last byte (i.e., ACC_RADIUS_MSB and MAG_RADIUS_MSB). Therefore the last byte must be written whenever the user wants to changes the configuration. The range of the radius for accelerometer is +/-1000, magnetometer is +/-960 and Gyroscope is NA.

Table 3-24: Radius range settings

Radius for sensor	Maximum Range
Accelerometer	+/- 1000 LSB
Magnetometer	+/- 960 LSB

3.6.5 Output data registers

3.6.5.1 Acceleration data

In non-fusion mode uncompensated acceleration data for each axis X/Y/Z, can be read from the appropriate ACC_DATA_<axis>_LSB and ACC_DATA_<axis>_MSB registers.

In fusion mode the fusion algorithm output offset compensated acceleration data for each axis X/Y/Z, the output data can be read from the appropriate ACC_DATA_<axis>_LSB and ACC_DATA_<axis>_MSB registers. Refer table below for information regarding the data types for the acceleration data.

Table 3-25: Acceleration data

Parameter	**Data type**	**bytes**
Accel_Data_X	signed	2
Accel_Data_Y	signed	2
Accel_Data_Z	signed	2

3.6.5.2 Magnetic Field Strength

In non-fusion mode uncompensated field strength data for each axis X/Y/Z, can be read from the appropriate MAG_DATA_<axis>_LSB and MAG_DATA_<axis>_MSB registers.

In fusion mode the fusion algorithm output offset compensated magnetic field strength data for each axis X/Y/Z, the output data can be read from the appropriate MAG_DATA_<axis>_LSB and MAG_DATA_<axis>_MSB registers. Refer table below for information regarding the data types for the magnetic field strength.

Table 3-26: Magnetic field strength data

Parameter	**Data type**	**bytes**
Mag_Data_X	signed	2
Mag_Data_Y	signed	2
Mag_Data_Z	signed	2

3.6.5.3 Angular Velocity

In non-fusion mode uncompensated angular velocity (yaw rate) data for each axis X/Y/Z, can be read from the appropriate GYR_DATA_<axis>_LSB and GYR_DATA_<axis>_MSB registers.

In fusion mode the fusion algorithm output offset compensated angular velocity (yaw rate) data for each axis X/Y/Z, the output data can be read from the appropriate GYR_DATA_<axis>_LSB and GYR_DATA_<axis>_MSB registers. Refer table below for information regarding the data types for the angular velocity.

Table 3-27: Yaw rate data

Parameter	**Data type**	**bytes**
Gyr_Data_X	signed	2
Gyr_Data_Y	signed	2
Gyr_Data_Z	signed	2

3.6.5.4 Orientation (Euler angles)

Orientation output only available in fusion operation modes.

The fusion algorithm output offset and tilt compensated orientation data in Euler angles format for each DOF Heading/Roll/Pitch, the output data can be read from the appropriate EUL<dof>_LSB and EUL_<dof>_MSB registers. Refer table below for information regarding the data types and the unit representation for the Euler angle format.

Table 3-28: Compensated orientation data in Euler angles format

Parameter	Data type	bytes
EUL_Heading	Signed	2
EUL_Roll	Signed	2
EUL_Pitch	Signed	2

Table 3-29: Euler angle data representation

Unit	Representation
Degrees	1 degree = 16 LSB
Radians	1 radian = 900 LSB

3.6.5.5 Orientation (Quaternion)

Orientation output only available in fusion operating modes.

The fusion algorithm output offset and tilt compensated orientation data in quaternion format for each DOF w/x/y/z, the output data can be read from the appropriate QUA_DATA_<dof>_LSB and QUA_DATA_<dof>_MSB registers. Refer table below for information regarding the data types and the unit representation for the Orientation output.

Table 3-30: Compensated orientation data in quaternion format

Parameter	Data type	bytes
QUA_Data_w	Signed	2
QUA_Data_x	Signed	2
QUA_Data_y	Signed	2
QUA_Data_z	Signed	2

Table 3-31: Quaternion data representation

Unit	Representation
Quaternion (unit less)	1 Quaternion (unit less) = 2^14 LSB

3.6.5.6 Linear Acceleration

Linear acceleration output only available in fusion operating modes.

The fusion algorithm output linear acceleration data for each axis x/y/z, the output data can be read from the appropriate LIA_DATA_<axis>_LSB and LIA_DATA_<axis>_MSB registers. Refer table below for further information regarding the data types and the unit representation for Linear acceleration

Table 3-32: Linear Acceleration Data

Parameter	Data type	bytes
LIA_Data_X	signed	2
LIA_Data_Y	signed	2
LIA_Data_Z	signed	2

Table 3-33: Linear Acceleration data representation

Unit	Representation
m/s^2	1 m/s^2 = 100 LSB
mg	1 mg = 1 LSB

3.6.5.7 Gravity Vector

Gravity Vector output only available in fusion operating modes.

The fusion algorithm output gravity vector data for each axis x/y/z, the output data can be read from the appropriate GRV_DATA_<axis>_LSB and GRV_DATA_<axis>_MSB registers. Refer table below for further information regarding the data types and the unit representation for the Gravity vector.

Table 3-34: Gravity Vector Data

Parameter	Data type	bytes
GRV_Data_X	signed	2
GRV_Data_Y	signed	2
GRV_Data_Z	signed	2

Table 3-35: Gravity Vector data representation

Unit	Representation
m/s^2	1 m/s^2 = 100 LSB
mg	1 mg = 1 LSB

3.6.5.8 Temperature

The temperature output data can be read from the TEMP register. The table below describes the output data type and data representation (depending on selected unit).
The temperature can be read from one of two sources, the temperature source can be selected by writing to the TEMP_SOURCE register as detailed below.

Table 3-36: Temperature Data

Parameter	Data type	bytes
TEMP	signed	1

Table 3-37: Temperature data representation

Unit	Representation
°C	1°C = 1 LSB
F	2 F = 1 LSB

Table 3-38: Temperature Source Selection

Source	[Reg Addr]: Register Value
Accelerometer	[TEMP_SOURCE]: xxxxxx00b
Gyroscope	[TEMP_SOURCE]: xxxxxx01b

3.7 Data register shadowing

This section describes the two methods to read sensor data from the BNO055 register map. In the first method also called multi byte read (or burst read) the data consistency is ensured by data register shadowing and hence the LSB and MSB of each axis are all referring to the same instance (refer section 4.6 I2C read access)
Whereas in the single byte reads, the MSB may get updated when the data in LSB is read and thereby resulting in data inconsistency.
So depending upon the application, the user may select the type of data read to ensure that the correct data is being read.

3.8 Interrupts

3.8.1 Interrupt Pin

INT is configured as interrupt pin for signaling an interrupt to the host. The interrupt trigger is configured as raising edge and is latched on to the INT pin. Once an interrupt occurs, the INT pin is set to high and will remain high until it is reset by host. This can be done by setting RST_INT in SYS_TRIGGER register.
Interrupts can be enabled by setting the corresponding bit in the interrupt enable register (INT_EN) and disabled when it is cleared.

Interrupt Pin Masking

Interrupts can be routed to the INT pin by setting the corresponding interrupt bit in the INT_MSK register.

Interrupt Status

Interrupt occurrences are stored in the interrupt status register (INT_STA). All bits in this register are cleared on read.

3.8.2 Interrupt Settings

3.8.2.1 Accelerometer Slow/No Motion Interrupt

The slow-motion/no-motion interrupt engine can be configured in two modes.

Slow-motion Interrupt is triggered when the measured slope of at least one enabled axis exceeds the programmable slope threshold for a programmable number of samples. Hence the engine behaves similar to the any-motion interrupt, but with a different set of parameters. In order to suppress false triggers, the interrupt is only generated (cleared) if a certain number *N* of consecutive slope data points is larger (smaller) than the slope threshold given by *slo_no_mot_dur<1:0>*. The number is *N* = *slo_no_mot_dur<1:0>* + 1.

In no-motion mode an interrupt is generated if the slope on all selected axes remains smaller than a programmable threshold for a programmable delay time. Figure 11 shows the timing diagram for the no-motion interrupt. The scaling of the threshold value is identical to that of the slow-motion interrupt. However, in no-motion mode register slo_no_mot_dur defines the delay time before the no-motion interrupt is triggered.

Table 3-39 lists the delay times adjustable with register slo_no_mot_dur. The timer tick period is 1 second. Hence using short delay times can result in considerable timing uncertainty.

If bit *SM/NM* is set to ‘1’ (‘0’), the no-motion/slow-motion interrupt engine is configured in the no-motion (slow-motion) mode. Common to both modes, the engine monitors the slopes of the axes that have been enabled with bits *AM/NM_X_AXIS*, *AM/NM_Y_AXIS*, and *AM/NM_Z_AXIS* for the x-axis, y-axis and z-axis, respectively. The measured slope values are continuously compared against the threshold value defined in register ACC_NM_THRES. The scaling is such that 1 LSB of ACC_NM_THRES corresponds to 3.91 mg in 2g-range (7.81 mg in 4g-range, 15.6 mg in 8g-range and 31.3 mg in 16g-range). Therefore the maximum value is 996 mg in 2g-range (1.99g in 4g-range, 3.98g in 8g-range and 7.97g in 16g-range). The time difference between the successive acceleration samples depends on the selected bandwidth and equates to 1/(2 * bw).

Table 3-39: No-motion time-out periods

slo_no_mot_dur	**Delay time**	***slo_no_mot_dur***	***Delay time***	***slo_no_mot_dur***	***Delay Time***
0	1 s	16	40 s	32	88 s
1	2 s	17	48 s	33	96 s
2	3 s	18	56 s	34	104 s
...	...	19	64 s.	...	...
14	15 s	20	72 s	62	328 s
15	16 s	21	80 s	63	336 s

Note: slo_no_mot_dur values 22 to 31 are not specified

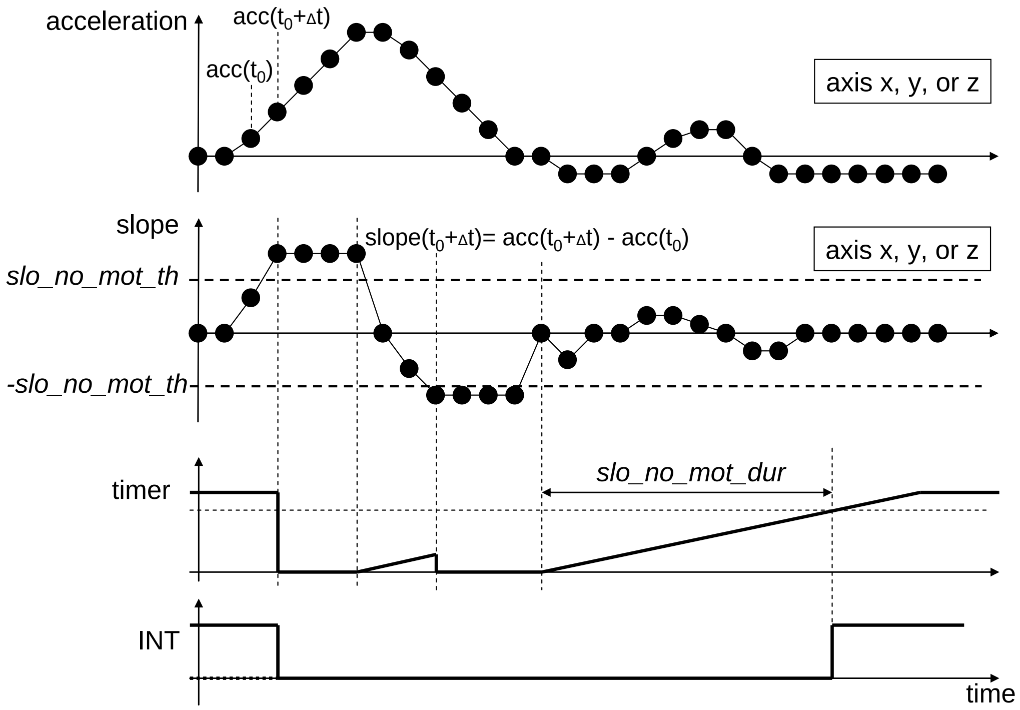


Table 3-40: Timing of No-motion interrupt

Params	**Value**	**[Reg Addr]: Register Value**
Detection Type	No Motion	[ACC_NM_SET]: xxxxxxx1b
	Slow Motion	[ACC_NM_SET]: xxxxxxx0b
Interrupt Parameters	Threshold	[ACC_NM_THRE]: bit7:bit0
	Duration	[ACC_NM_SET]: bit6:bit1

Axis selection	X-axis	[ACC_INT_Settings]: xxxxx1xxb
	Y-axis	[ACC_INT_Settings]: xxxx1xxxb
	Z-axis	[ACC_INT_Settings]: xxx1xxxxb

3.8.2.2 Accelerometer Any Motion Interrupt

The any-motion interrupt uses the slope between successive acceleration signals to detect changes in motion. An interrupt is generated when the slope (absolute value of acceleration difference) exceeds a preset threshold. It is cleared as soon as the slope falls below the threshold. The principle is made clear in Figure 2: Principle of any-motion detection.

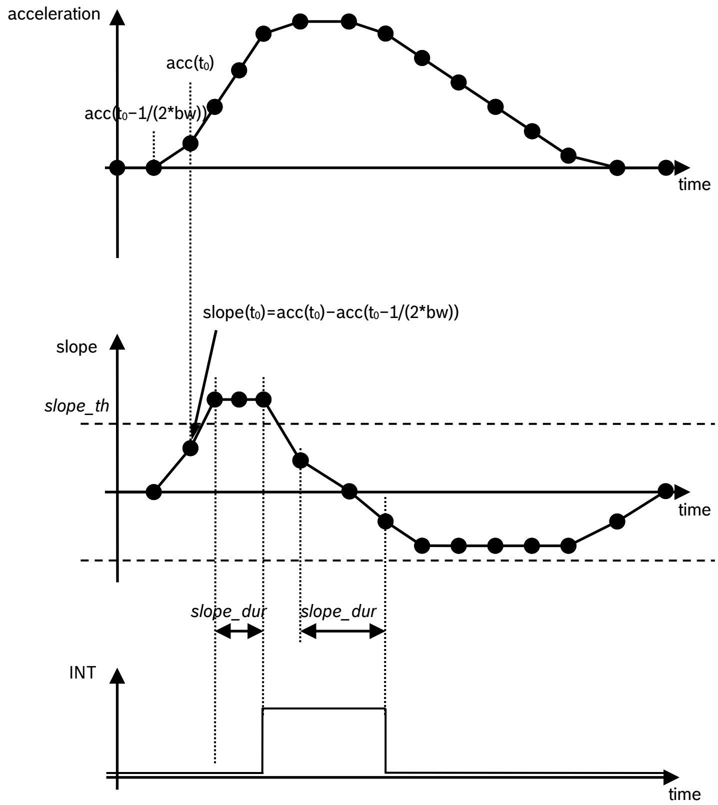


Figure 2: Principle of any-motion detection

The threshold is defined through register ACC_AM_THRES. In terms of scaling 1 LSB of ACC_AM_THRES corresponds to 3.91 mg in 2g-range (7.81 mg in 4g-range, 15.6 mg in 8g-range and 31.3 mg in 16g-range). Therefore the maximum value is 996 mg in 2g-range (1.99g in 4g-range, 3.98g in 8g-range and 7.97g in 16g-range).

The time difference between the successive acceleration signals depends on the selected bandwidth and equates to 1/(2*bandwidth) ()t=1/(2*bw)). In order to suppress false triggers, the interrupt is only generated (cleared) if a certain number *N* of consecutive slope data points is larger (smaller) than the slope threshold given by ACC_AM_THRES. This number is set by the AM_DUR bits. It is N = AM_DUR + 1.

Example: *AM_DUR* = 00b, ..., 11b = 1decimal, ..., 4decimal.

Enabling (disabling) for each axis:

Any-motion detection can be enabled (disabled) for each axis separately by writing ´1´ (´0´) to bits AM/NM_X_AXIS, AM/NM_Y_AXIS, AM/NM_Z_AXIS. The criteria for any-motion detection are fulfilled and the slope interrupt is generated if the slope of any of the enabled axes exceeds the threshold ACC_AM_THRES for [AM_DUR +1] consecutive times. As soon as the slopes of all enabled axes fall or stay below this threshold for [AM_DUR +1] consecutive times the interrupt is cleared unless interrupt signal is latched.

Table 3-41: Any-motion Interrupt parameters and Axis selection

Params	**Value**	**[Reg Addr]: Register Value**
Interrupt Parameters	Threshold	[ACC_AM_THRES]: bit7:bit0
	Duration	[ACC_INT_Settings]: bit1:bit0
Axis selection	X-axis	[ACC_INT_Settings]: xxxxx1xxb
	Y-axis	[ACC_INT_Settings]: xxxx1xxxb
	Z-axis	[ACC_INT_Settings]: xxx1xxxxb

3.8.2.3 Accelerometer High G Interrupt

This interrupt is based on the comparison of acceleration data against a high-g threshold for the detection of shock or other high-acceleration events.

The high-g interrupt is enabled (disabled) per axis by writing ´1´ (´0´) to bits ACC_HIGH_G in the INT_EN register and enabling the axis in with bits HG_X_AXIS, HG_Y_AXIS, and HG_Z_AXIS, respectively in the ACC_INT_Settings register. The high-g threshold is set through the ACC_HG_THRES register. The meaning of an LSB of ACC_HG_THRES depends on the selected g-range: it corresponds to 7.81 mg in 2g-range, 15.63 mg in 4g-range, 31.25 mg in 8g-range, and 62.5 mg in 16g-range (i.e. increment depends from g-range setting).

The high-g interrupt is generated if the absolute value of the acceleration of at least one of the enabled axes (´or´ relation) is higher than the threshold for at least the time defined by the ACC_HG_DURATION register. The interrupt is reset if the absolute value of the acceleration of all enabled axes (´and´ relation) is lower than the threshold for at least the time defined by the ACC_HG_DURATION register. The interrupt status is stored in bit ACC_HIGH_G in the INT_STA register. The relation between the content of ACC_HG_DURATION and the actual delay of the interrupt generation is delay [ms] = [ACC_HG_DURATION + 1] * 2 ms. Therefore, possible delay times range from 2 ms to 512 ms.

Table 3-42: High-G Interrupt parameters and Axis selection

Params	**Value**	**[Reg Addr]: Register Value**
Interrupt Parameters	Threshold	[ACC_HG_THRES]: bit7 : bit0
	Duration	[ACC_HG_DURATION]: bit7 : bit0
Axis selection	X-axis	[ACC_INT_Settings]: xx1xxxxxb
	Y-axis	[ACC_INT_Settings]: x1xxxxxxb
	Z-axis	[ACC_INT_Settings]: 1xxxxxxxb

3.8.2.4 Gyroscope High Rate Interrupt

This interrupt is based on the comparison of angular rate data against a high-rate threshold for the detection of shock or other high-angular rate events. The principle is made clear in Figure 3 below:

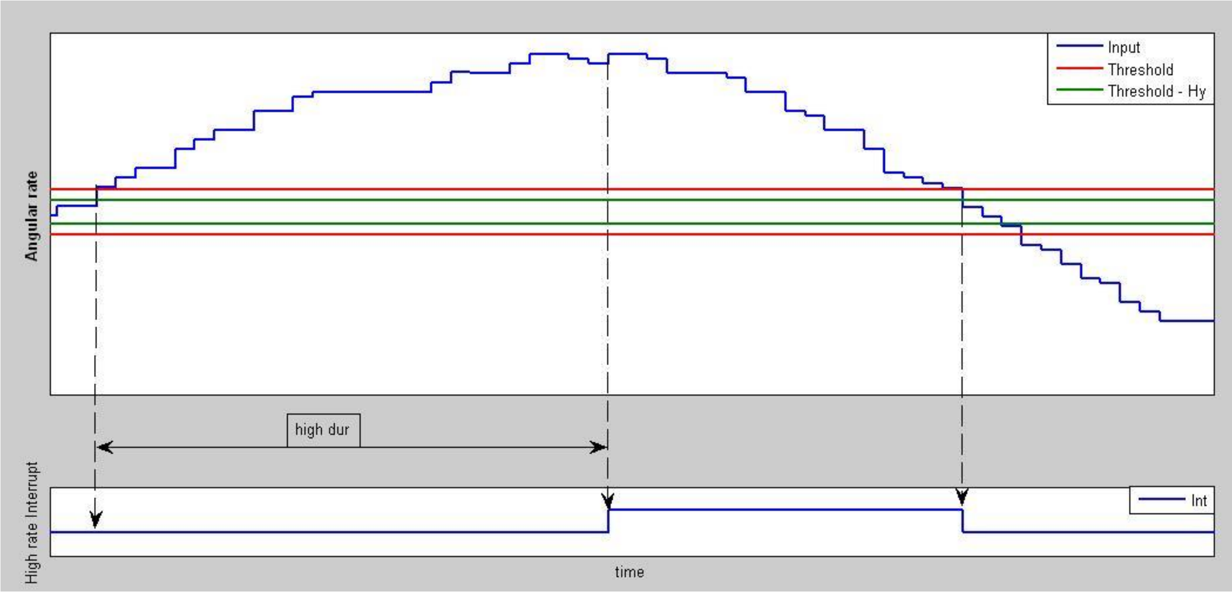


Figure 3: High rate interrupt

The high-rate interrupt is enabled (disabled) per axis by writing ´1´ (´0´) to bits *GYRO_HIGH_RATE* in the *INT_EN* register and for each axis by writing to the *HR_X_AXIS*, *HR_Y_AXIS*, and *HR_Z_AXIS*, respectively in the GYR_INT_SETTING register. The high-rate threshold is set through the *HR_<axis>_*Threshold bits in the appropriate *GYR_HR_<axis>_SET register*. The meaning of an LSB of *HR_<axis>_Threshold* depends on the selected °/s-range: it corresponds to 62.5°/s in 2000°/s-range, 31.25°/s in 1000°/s-range, 15.625°/s in 500°/s -range …). The *HR_<axis>_Threshold* register setting 0 corresponds to 62.26°/s in 2000°/s-range, 31.13°/s in 1000°/s-range, 15.56°/s in 500°/s-range …. Therefore the maximum value is 1999.76°/s in 2000°/s-range (999.87°/s 1000°/s-range, 499.93°/s in 500°/s -range …).
A hysteresis can be selected by setting the *HR_<axis>_THRES_HYST* bits. Analogously to threshold, the meaning of an LSB of *HR_<axis>_THRES_HYST* bits is °/s-range dependent: The *HR_<axis>_THRES_HYST* register setting 0 corresponds to an angular rate difference of 62.26°/s in 2000°/s-range, 31.13°/s in 1000°/s-range, 15.56°/s in 500°/s-range …. The meaning of an LSB of *HR_<axis>_THRES_HYST* depends on the selected °/s-range too: it corresponds to 62.5°/s in 2000°/s-range, 31.25°/s in 1000°/s-range, 15.625°/s in 500°/s -range …).

The high-rate interrupt is generated if the absolute value of the angular rate of at least one of the enabled axes (´or´ relation) is higher than the threshold for at least the time defined by the *GYR_DUR_<axis>* register. The interrupt is reset if the absolute value of the angular rate of all enabled axes (´and´ relation) is lower than the threshold minus the hysteresis. In bit *GYR_HIGH_RATE* in the *INT_STA* the interrupt status is stored. The relation between the content of *GYR_DUR_<axis>* and the actual delay of the interrupt generation is delay [ms] = [*GYR_DUR_<axis>* + 1] * 2.5 ms. Therefore, possible delay times range from 2.5 ms to 640 ms.

Table 3-43: High Rate Interrupt parameters and Axis selection

Params	Value	[Reg Addr]: Register Value
Axis selection	X-axis	[GYR_INT_SETTING]: xxxx1xxxb
	Y-axis	[GYR_INT_SETTING]: xxx1xxxxb
	Z-axis	[GYR_INT_SETTING]: xx1xxxxxb
High Rate Filter settings	Filtered	[GYR_INT_SETTING]: 0xxxxxxxb
	Unfiltered	[GYR_INT_SETTING]: 1xxxxxxxb
Interrupt Settings X-axis	Threshold	[GYR_HR_X_SET]: bit4 : bit0
	Duration	[GYR_DUR_X]: bit7 : bit0
	Hysteresis	[GYR_HR_X_SET]: bit6 : bit5
Interrupt Settings Y-axis	Threshold	[GYR_HR_Y_SET]: bit4 : bit0
	Duration	[GYR_DUR_Y]: bit7 : bit0
	Hysteresis	[GYR_HR_Y_SET]: bit6 : bit5
Interrupt Settings X-axis	Threshold	[GYR_HR_Z_SET]: bit4 : bit0
	Duration	[GYR_DUR_Z]: bit7 : bit0
	Hysteresis	[GYR_HR_Z_SET]: bit6 : bit5

3.8.2.5 Gyroscope Any Motion Interrupt

Any-motion (slope) detection uses the slope between successive angular rate signals to detect changes in motion. An interrupt is generated when the slope (absolute value of angular rate difference) exceeds a preset threshold. It is cleared as soon as the slope falls below the threshold. The principle is made clear in Figure 4.

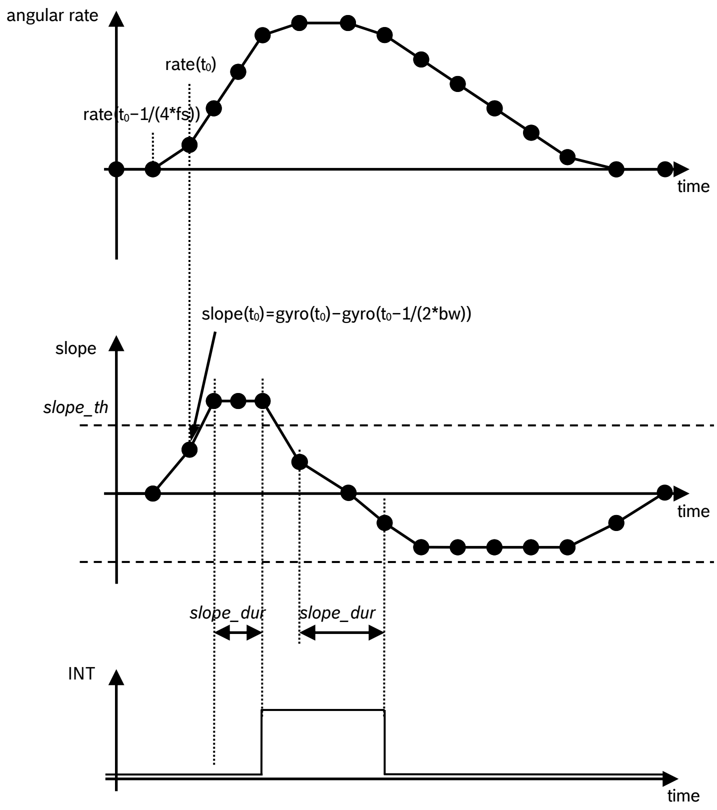


Figure 4: Principle of any-motion detection

The threshold is defined through register GYR_AM_THRES. In terms of scaling 1 LSB of GYR_AM_THRES corresponds to 1 °/s in 2000°/s-range (0.5°/s in 1000°/s-range, 0.25°/s in 500°/s -range ...). Therefore the maximum value is 125°/s in 2000°/s-range (62.5°/s 1000°/s-range, 31.25 in 500°/s -range ...).

The time difference between the successive angular rate signals depends on the selected update rate(fs) which is coupled to the bandwidth and equates to 1/(4*fs) (t=1/(4*fs)). For bandwidth settings with an update rate higher than 400Hz (bandwidth =0,1,2) fs is set to 400Hz.

In order to suppress false triggers, the interrupt is only generated (cleared) if a certain number *N* of consecutive slope data points is larger (smaller) than the slope threshold given by GYR_AM_THRES. This number is set by the Slope Samples bits in the GYR_AM_SET register. It is *N* = [Slope Samples + 1]*4. N is set in samples. Thus the time is scaling with the update rate (fs).

3.8.2.6 Enabling (disabling) for each axis

Any-motion detection can be enabled (disabled) for each axis separately by writing ´1´ (´0´) to bits *AM_X_AXIS, AM_Y_AXIS, AM_Z_AXIS in the GYR_INT_SETTING register*. The criteria for any-motion detection are fulfilled and the Any-Motion interrupt is generated if the slope of any of the enabled axes exceeds the threshold GYR_AM_THRES for [Slope Samples+1]*4 consecutive times. As soon as the slopes of all enabled axes fall or stay below this threshold for [Slope Samples +1]*4 consecutive times the interrupt is cleared unless interrupt signal is latched.

3.8.2.7 Axis of slope / any motion interrupt

The interrupt status is stored in bit *GYRO_AM* in the *INT_EN* register. The Any-motion interrupt supplies additional information about the detected slope.

Table 3-44: Axis selection and any motion interrupt

Params	Value	[Reg Addr]: Register Value
Axis selection	X-axis	[GYR_INT_SETING]: xxxxxxx1b
	Y-axis	[GYR_INT_SETING]: xxxxxx1xb
	Z-axis	[GYR_INT_SETING]: xxxxx1xxb
Any Motion Filter settings	Filtered	[GYR_INT_SETING]: x0xxxxxxb
	Unfiltered	[GYR_INT_SETING]: x1xxxxxxb
Interrupt Settings	Threshold	[GYR_AM_THRES]: bit6 : bit0
	Slope Samples	[GYR_AM_SET]: bit1 : bit0
	Awake Duration	[GYR_AM_SET]: bit3 : bit2

3.9 Self-Test

3.9.1 Power On Self Test (POST)

During the device startup, a power on self test is executed. This feature checks that the connected sensors and microcontroller are responding / functioning correctly. Following tests are executed

Table 3-45: Power on Self Test

Components	Test type
Accelerometer	Verify chip ID
Magnetometer	Verify chip ID
Gyroscope	Verify chip ID
Microcontroller	Memory Build In Self Test

The results of the POST are stored at register ST_RESULT, where a bit set indicates test passed and cleared indicates self test failed.

3.9.2 Build In Self Test (BIST)

The host can trigger a self test from CONFIG MODE. The test can be triggered by setting bit SELF_TEST in the in the SYS_TRIGGER register, the results are stored in the ST_RESULT register. During the execution of the system test, all other features are paused.

Table 3-46: Power on Self Test

Components	Test type
Accelerometer	built in self test
Magnetometer	built in self test
Gyroscope	built in self test
Microcontroller	No test performed

3.10 Boot loader

The boot loader is located at the start of the program memory and it is executed at each reset / power-on sequence. It first checks the status of the nBOOT_LOAD_PIN.

If the nBOOT_LOAD_PIN is pulled low during reset / power-on sequence, it continues execution in boot loader mode. Otherwise the device continues to boot in application mode.

In case there is a firmware update, then an application note would be available in time with the necessary information to upgrade at the host side. Nevertheless it is recommended that the nBOOT_LOAD_PIN is connected as shown in section 5.

3.11 Calibration

Though the sensor fusion software runs the calibration algorithm of all the three sensors (accelerometer, gyroscope and magnetometer) in the background to remove the offsets, some preliminary steps had to be ensured for this automatic calibration[5] to take place.
The accelerometer and the gyroscope are relatively less susceptible to external disturbances, as a result of which the offset is negligible. Whereas the magnetometer is susceptible to external magnetic field and therefore to ensure proper heading accuracy, the calibration steps described below have to be taken.
Depending on the sensors been selected in the fusion mode, the following simple steps had to be taken after every 'Power on Reset' for proper calibration of the device.

3.11.1 Accelerometer Calibration

- Place the device in 6 different stable positions for a period of few seconds to allow the accelerometer to calibrate.
- Make sure that there is slow movement between 2 stable positions
- The 6 stable positions could be in any direction, but make sure that the device is lying at least once perpendicular to the x, y and z axis.
- The register CALIB_STAT can be read to see the calibration status of the accelerometer.

3.11.2 Gyroscope Calibration

- Place the device in a single stable position for a period of few seconds to allow the gyroscope to calibrate
- The register CALIB_STAT can be read to see the calibration status of the gyroscope.

3.11.3 Magnetometer Calibration

Magnetometer in general are susceptible to both hard-iron and soft-iron distortions, but majority of the cases are rather due to the former. And the steps mentioned below are to calibrate the magnetometer for hard-iron distortions.
Nevertheless certain precautions need to be taken into account during the positioning of the sensor in the PCB which is described in our HSMI (Handling, Soldering and Mounting Instructions) application note to avoid unnecessary magnetic influences.

Compass, M4G & NDOF_FMC_OFF:

- Make some random movements (for example: writing the number '8' on air) until the CALIB_STAT register indicates fully calibrated.
- It takes more calibration movements to get the magnetometer calibrated than in the NDOF mode.

NDOF:

- The same random movements have to be made to calibrate the sensor as in the FMC_OFF mode, but here it takes relatively less calibration movements (and slightly higher current consumption) to get the magnetometer calibrated.
- The register CALIB_STAT can be read to see the calibration status of the magnetometer.

[5] It is not possible to disable the automatic calibration which runs in the background

3.11.4 Reuse of Calibration Profile

Once the device is calibrated, the calibration profile can be reused to get the correct orientation data immediately after ‘Power of Reset’ (prior to going through the steps mentioned in the above section). However, once the sensor enters the internal calibration routine, the calibration profile is overwritten with the newly obtained sensor offsets and sensor radius. Depending on the application, necessary steps had to be ensured for proper calibration of the sensor.

Reading Calibration profile
The calibration profile includes sensor offsets and sensor radius. Host system can read the offsets and radius only after a full calibration is achieved and the operation mode is switched to CONFIG_MODE. Refer to sensor offsets and sensor radius registers.

Setting Calibration profile
It is important that the correct offsets and corresponding sensor radius are used. Incorrect offsets may result in unreliable orientation data even at calibration accuracy level 3. To set the calibration profile the following steps need to be taken

1. Select the operation mode to CONFIG_MODE
2. Write the corresponding sensor offsets and radius data
3. Change operation mode to fusion mode

4. Register description

4.1 General Remarks

The entire communication with the device is performed by reading from and writing to registers. Registers have a width of 8 bits. There are several registers which are either completely or partially marked as ‘reserved’. Any reserved bit is ignored when it is written and no specific value is guaranteed when read. It is recommended not to use registers at all which are completely marked as ‘reserved’. Furthermore it is recommended to mask out (logical and with zero) reserved bits of registers which are partially marked as reserved.

Read-Only Registers are marked as shown in Table 4-1: Register Access Coding. Any attempt to write to these registers is ignored.
There are bits within some registers that trigger internal sequences. These bits are configured for write-only access and read as value ´0´.

4.2 Register map

The register map is separated into two logical pages, Page 1 contains sensor specific configuration data and Page 0 contains all other configuration parameters and output data.

At power-on Page 0 is selected, the PAGE_ID register can be used to identify the current selected page and change between page 0 and page 1.

4.2.1 Register map Page 0

Table 4-1: Register Access Coding

read/write	read only	write only	reserved

Table 4-2: Register Map Page 0

Register Address	Register Name	Default Value	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
7F-6B	Reserved	NA								
6A	MAG_RADIUS_MSB		Magnetometer Radius							
69	MAG_RADIUS_LSB		Magnetometer Radius							
68	ACC_RADIUS_MSB		Accelerometer Radius							
67	ACC_RADIUS_LSB		Accelerometer Radius							
66	GYR_OFFSET_Z_MSB	0x00	Gyroscope Offset Z <15:8>							
65	GYR_OFFSET_Z_LSB	0x00	Gyroscope Offset Z <7:0>							
64	GYR_OFFSET_Y_MSB	0x00	Gyroscope Offset Y <15:8>							
63	GYR_OFFSET_Y_LSB	0x00	Gyroscope Offset Y <7:0>							
62	GYR_OFFSET_X_MSB	0x00	Gyroscope Offset X <15:8>							
61	GYR_OFFSET_X_LSB	0x00	Gyroscope Offset X <7:0>							
60	MAG_OFFSET_Z_MSB	0x00	Magnetometer Offset Z <15:8>							
5F	MAG_OFFSET_Z_LSB	0x00	Magnetometer Offset Z <7:0>							
5E	MAG_OFFSET_Y_MSB	0x00	Magnetometer Offset Y <15:8>							
5D	MAG_OFFSET_Y_LSB	0x00	Magnetometer Offset Y <7:0>							
5C	MAG_OFFSET_X_MSB	0x00	Magnetometer Offset X <15:8>							
5B	MAG_OFFSET_X_LSB	0x00	Magnetometer Offset X <7:0>							
5A	ACC_OFFSET_Z_MSB	0x00	Accelerometer Offset Z <15:8>							
59	ACC_OFFSET_Z_LSB	0x00	Accelerometer Offset Z <7:0>							
58	ACC_OFFSET_Y_MSB	0x00	Accelerometer Offset Y <15:8>							
57	ACC_OFFSET_Y_LSB	0x00	Accelerometer Offset Y <7:0>							
56	ACC_OFFSET_X_MSB	0x00	Accelerometer Offset X <15:8>							
55	ACC_OFFSET_X_LSB	0x00	Accelerometer Offset X <7:0>							
43 - 54	Reserved	0x00								

Register Address	Register Name	Default Value	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
42	AXIS_MAP_SIGN	TBD						Remapped X axis sign	Remapped Y axis sign	Remapped Z axis sign
41	AXIS_MAP_CONFIG	TBD			Remapped Z axis value		Remapped Y axis value		Remapped X axis value	
40	TEMP_SOURCE	0x02							TEMP_Source <1:0>	
3F	SYS_TRIGGER	0x00	CLK_SEL	RST_INT	RST_SYS					Self_Test
3E	PWR_MODE	0x00							Power Mode <1:0>	
3D	OPR_MODE	0x1C					Operation Mode <3:0>			
3C	Reserved	0xFF								
3B	UNIT_SEL	0x80	ORI_Android_Windows			TEMP_Unit		EUL_Unit	GYR_Unit	ACC_Unit
3A	SYS_ERR	0x00	System Error Code							
39	SYS_STATUS	0x00	System Status Code							
38	SYS_CLK_STATUS	0x00								ST_MAIN_CLK
37	INT_STA	0x00	ACC_NM	ACC_AM	ACC_HIGH_G		GYR_HIGH_RATE	GYRO_AM		
36	ST_RESULT	0x0F					ST_MCU	ST_GYR	ST_MAG	ST_ACC
35	CALIB_STAT	0x00	SYS Calib Status 0:3		GYR Calib Status 0:3		ACC Calib Status 0:3		MAG Calib Status 0:3	
34	TEMP	0x00	Temperature							
33	GRV_Data_Z_MSB	0x00	Gravity Vector Data Z <15:8>							
32	GRV_Data_Z_LSB	0x00	Gravity Vector Data Z <7:0>							
31	GRV_Data_Y_MSB	0x00	Gravity Vector Data Y <15:8>							
30	GRV_Data_Y_LSB	0x00	Gravity Vector Data Y <7:0>							
2F	GRV_Data_X_MSB	0x00	Gravity Vector Data X <15:8>							
2E	GRV_Data_X_LSB	0x00	Gravity Vector Data X <7:0>							
2D	LIA_Data_Z_MBS	0x00	Linear Acceleration Data Z <15:8>							
2C	LIA_Data_Z_LSB	0x00	Linear Acceleration Data Z <7:0>							
2B	LIA_Data_Y_MBS	0x00	Linear Acceleration Data Y <15:8>							
2A	LIA_Data_Y_LSB	0x00	Linear Acceleration Data Y <7:0>							
29	LIA_Data_X_MBS	0x00	Linear Acceleration Data X <15:8>							
28	LIA_Data_X_LSB	0x00	Linear Acceleration Data X <7:0>							
27	QUA_Data_z_MSB	0x00	Quaternion z Data <15:8>							
26	QUA_Data_z_LSB	0x00	Quaternion z Data <7:0>							
25	QUA_Data_y_MSB	0x00	Quaternion y Data <15:8>							
24	QUA_Data_y_LSB	0x00	Quaternion y Data <7:0>							
23	QUA_Data_x_MSB	0x00	Quaternion x Data <15:8>							
22	QUA_Data_x_LSB	0x00	Quaternion x Data <7:0>							
21	QUA_Data_w_MSB	0x00	Quaternion w Data <15:8>							
20	QUA_Data_w_LSB	0x00	Quaternion w Data <7:0>							
1F	EUL_Pitch_MSB	0x00	Pitch Data <15:8>							

Register Address	Register Name	Default Value	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
1E	EUL_Pitch_LSB	0x00	Pitch Data <7:0>							
1D	EUL_Roll_MSB	0x00	Roll Data <15:8>							
1C	EUL_Roll_LSB	0x00	Roll Data <7:0>							
1B	EUL_Heading_MSB	0x00	Heading Data <15:8>							
1A	EUL_Heading_LSB	0x00	Heading Data <7:0>							
19	GYR_DATA_Z_MSB	0x00	Gyroscope Data Z <15:8>							
18	GYR_DATA_Z_LSB	0x00	Gyroscope Data Z <7:0>							
17	GYR_DATA_Y_MSB	0x00	Gyroscope Data Y <15:8>							
16	GYR_DATA_Y_LSB	0x00	Gyroscope Data Y <7:0>							
15	GYR_DATA_X_MSB	0x00	Gyroscope Data X <15:8>							
14	GYR_DATA_X_LSB	0x00	Gyroscope Data X <7:0>							
13	MAG_DATA_Z_MSB	0x00	Magnetometer Data Z <15:8>							
12	MAG_DATA_Z_LSB	0x00	Magnetometer Data Z <7:0>							
11	MAG_DATA_Y_MSB	0x00	Magnetometer Data Y <15:8>							
10	MAG_DATA_Y_LSB	0x00	Magnetometer Data Y <7:0>							
F	MAG_DATA_X_MSB	0x00	Magnetometer Data X <15:8>							
E	MAG_DATA_X_LSB	0x00	Magnetometer Data X <7:0>							
D	ACC_DATA_Z_MSB	0x00	Acceleration Data Z <15:8>							
C	ACC_DATA_Z_LSB	0x00	Acceleration Data Z <7:0>							
B	ACC_DATA_Y_MSB	0x00	Acceleration Data Y <15:8>							
A	ACC_DATA_Y_LSB	0x00	Acceleration Data Y <7:0>							
9	ACC_DATA_X_MSB	0x00	Acceleration Data X <15:8>							
8	ACC_DATA_X_LSB	0x00	Acceleration Data X <7:0>							
7	Page ID	0x00	Page ID							
6	BL_Rev_ID	NA	Bootloader Version							
5	SW_REV_ID_MSB	0x03[6]	SW Revision ID <15:8>							
4	SW_REV_ID_LSB	0x08[7]	SW Revision ID <7:0>							
3	GYR_ID	0x0F	GYRO chip ID							
2	MAG_ID	0x32	MAG chip ID							
1	ACC_ID	0xFB	ACC chip ID							
0	CHIP_ID	0xA0	BNO055 CHIP ID							

[6] The current software version is 0.3.0.8 and therefore the SW_REV_ID_MSB is 0x03. However the register default value is subject to change with respect to the updated software.
[7] The current software version is 0.3.0.8 and therefore the SW_REV_ID_LSB is 0x08. However the register default value is subject to change with respect to the updated software.

4.2.2 Register map Page 1

Table4-3: Register Map Page 1

Register Address	Register Name	Default Value	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
7F-60	Reserved	0x00								
5F - 50	UNIQUE_ID	n.a.	BNO unique ID							
4F - 20	Reserved	0x00								
1F	GYR_AM_SET	0x0A					Awake Duration <1:0>		Slope Samples <1:0>	
1E	GYR_AM_THRES	0x04		Gyro Any Motion Threshold <6:0>						
1D	GYR_DUR_Z	0x19	HR_Z_Duration							
1C	GYR_HR_Z_SET	0x01		HR_Z_THRES_HYST <1:0>		HR_Z_Threshold <4:0>				
1B	GYR_DUR_Y	0x19	HR_Y_Duration							
1A	GYR_HR_Y_SET	0x01		HR_Y_THRES_HYST <1:0>		HR_Y_Threshold <4:0>				
19	GYR_DUR_X	0x19	HR_X_Duration							
18	GYR_HR_X_SET	0x01		HR_X_THRES_HYST <1:0>		HR_X_Threshold <4:0>				
17	GYR_INT_SETING	0x00	HR_FILT	AM_FILT	HR_Z_AXIS	HR_Y_AXIS	HR_X_AXIS	AM_Z_AXIS	AM_Y_AXIS	AM_X_AXIS
16	ACC_NM_SET	0x0B		NO/SLOW Motion Duration <5:0>						SMNM
15	ACC_NM_THRE	0x0A	Accelerometer NO/SLOW motion threshold							
14	ACC_HG_THRES	0xC0	Accelerometer High G Threshold							
13	ACC_HG_DURATION	0x0F	Accelerometer High G Duration							
12	ACC_INT_Settings	0x03	HG_Z_AXIS	HG_Y_AXIS	HG_X_AXIS	AM/NM_Z_AXIS	AM/NM_Y_AXIS	AM/NM_X_AXIS	AM_DUR <1:0>	
11	ACC_AM_THRES	0x14	Accelerometer Any motion threshold							
10	INT_EN	0x00	ACC_NM	ACC_AM	ACC_HIGH_G		GYR_HIGH_RATE	GYRO_AM		
F	INT_MSK	0x00	ACC_NM	ACC_AM	ACC_HIGH_G		GYR_HIGH_RATE	GYRO_AM		
E	Reserved	0x00								
D	GYR_Sleep_Config	0x00			AUTO_SLP_DURATION <2:0>			SLP_DURATION <2:0>		
C	ACC_Sleep_Config	0x00				SLP_DURATION <3:0>				SLP_MODE
B	GYR_Config_1	0x00						GYR_Power_Mode <2:0>		
A	GYR_Config_0	0x38			GYR_Bandwidth <2:0>			GYR_Range <2:0>		
9	MAG_Config	0x6D		MAG_Power_mode <1:0>		MAG_OPR_Mode <1:0>		MAG_Data_output_rate <2:0>		
8	ACC_Config	0x0D	ACC_PWR_Mode <2:0>			ACC_BW <2:0>			ACC_Range <1:0>	
7	Page ID	0x01	Page ID							
6 - 0	Reserved	n.a.								

4.3 Register description (Page 0)

4.3.1 CHIP_ID 0x00

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	1	0	1	0	0	0	0	0
Content	BNO055 CHIP ID							

DATA	bits	Description
BNO055 CHIP ID	**<7:0>**	Chip identification code, read-only fixed value 0xA0

4.3.2 ACC_ID 0x01

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	R	r
Reset	0xFB							
Content	ACC chip ID							

DATA	bits	Description
ACC chip ID	<7:0>	Chip ID of the Accelerometer device, read-only fixed value 0xFB

4.3.3 MAG_ID 0x02

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	R	r
Reset	0x32							
Content	MAG chip ID							

DATA	bits	Description
MAG chip ID	<7:0>	Chip ID of the Magnetometer device, read-only fixed value 0x32

4.3.4 GYR_ID 0x03

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	R	r
Reset	0x0F							
Content	GRYO chip ID							

DATA	bits	Description
GYRO chip ID	<7:0>	Chip ID of the Gyroscope device, read-only fixed value 0x0F

4.3.5 SW_REV_ID_LSB 0x04

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset								
Content	SW Revision ID <7:0>							

DATA	bits	Description
SW Revision ID <7:0>	<7:0>	Lower byte of SW Revision ID, read-only fixed value depending on SW revision programmed on microcontroller

4.3.6 SW_REV_ID_MSB 0x05

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset								
Content	SW Revision ID <15:8>							

DATA	bits	Description
SW Revision ID <15:8>	<7:0>	Upper byte of SW Revision ID, read-only fixed value depending on SW revision programmed on microcontroller

4.3.7 BL_REV_ID 0x06

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset								
Content	Bootloader Version							

DATA	bits	Description
Bootloader Version	<7:0>	Identifies the version of the bootloader in the microcontroller, read-only

4.3.8 PAGE ID 0x07

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset	0	0	0	0	0	0	0	0
Content	Page ID							

DATA	bits	Description
Page ID	<7:0>	Read: Number of currently selected page Write: Change page, 0x00 or 0x01

4.3.9 ACC_DATA_X_LSB 0x08

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Acceleration Data X <7:0>							

DATA	bits	Description
Acceleration Data X <7:0>	<7:0>	Lower byte of X axis Acceleration data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.10 ACC_DATA_X_MSB 0x09

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Acceleration Data X <15:8>							

DATA	bits	Description
Acceleration Data X <15:8>	<7:0>	Upper byte of X axis Acceleration data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.11 ACC_DATA_Y_LSB 0x0A

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Acceleration Data Y <7:0>							

DATA	bits	Description
Acceleration Data Y <7:0>	<7:0>	Lower byte of Y axis Acceleration data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.12 ACC_DATA_Y_MSB 0x0B

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Acceleration Data Y <15:8>							

DATA	bits	Description
Acceleration Data Y <15:8>	<7:0>	Upper byte of Y axis Acceleration data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.13 ACC_DATA_Z_LSB 0x0C

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Acceleration Data Z <7:0>							

DATA	bits	Description
Acceleration Data Z <7:0>	<7:0>	Lower byte of Z axis Acceleration data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.14 ACC_DATA_Z_MSB 0x0D

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Acceleration Data Z <15:8>							

DATA	bits	Description
Acceleration Data Z <15:8>	<7:0>	Upper byte of Z axis Acceleration data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.15 MAG_DATA_X_LSB 0x0E

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Magnetometer Data X <7:0>							

DATA	bits	Description
Magnetometer Data X <7:0>	<7:0>	Lower byte of X axis Magnetometer data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.16 MAG_DATA_X_MSB 0x0F

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Magnetometer Data X <15:8>							

DATA	bits	Description
Magnetometer Data X <15:8>	<7:0>	Upper byte of X axis Magnetometer data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.17 MAG_DATA_Y_LSB 0x10

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Magnetometer Data Y <7:0>							

DATA	bits	Description
Magnetometer Data Y <7:0>	<7:0>	Lower byte of Y axis Magnetometer data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.18 MAG_DATA_Y_MSB 0x11

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Magnetometer Data Y <15:8>							

DATA	bits	Description
Magnetometer Data Y <15:8>	<7:0>	Upper byte of Y axis Magnetometer data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.19 MAG_DATA_Z_LSB 0x12

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Magnetometer Data Z <7:0>							

DATA	bits	Description
Magnetometer Data Z <7:0>	<7:0>	Lower byte of Z axis Magnetometer data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.20 MAG_DATA_Z_MSB 0x13

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Magnetometer Data Z <15:8>							

DATA	bits	Description
Magnetometer Data Z <15:8>	<7:0>	Upper byte of Z axis Magnetometer data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.21 GYR_DATA_X_LSB 0x14

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Gyroscope Data X <7:0>							

DATA	bits	Description
Gyroscope Data X <7:0>	<7:0>	Lower byte of X axis Gyroscope data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.22 GYR_DATA_X_MSB 0x15

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Gyroscope Data X <15:8>							

DATA	bits	Description
Gyroscope Data X <15:8>	<7:0>	Upper byte of X axis Gyroscope data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.23 GYR_DATA_Y_LSB 0x16

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Gyroscope Data Y <7:0>							

DATA	bits	Description
Gyroscope Data Y <7:0>	<7:0>	Lower byte of Y axis Gyroscope data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.24 GYR_DATA_Y_MSB 0x17

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Gyroscope Data Y <15:8>							

DATA	bits	Description
Gyroscope Data Y <15:8>	<7:0>	Upper byte of Y axis Gyroscope data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.25 GYR_DATA_Z_LSB 0x18

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Gyroscope Data Z <7:0>							

DATA	bits	Description
Gyroscope Data Z <7:0>	<7:0>	Lower byte of Z axis Gyroscope data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.26 GYR_DATA_Z_MSB 0x19

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Gyroscope Data Z <15:8>							

DATA	bits	Description
Gyroscope Data Z <15:8>	<7:0>	Upper byte of Z axis Gyroscope data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.27 EUL_DATA_X_LSB 0x1A

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Heading Data <7:0>							

DATA	bits	Description
Heading Data <7:0>	<7:0>	Lower byte of heading data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.28 EUL_DATA_X_MSB 0x1B

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Heading Data <15:8>							

DATA	bits	Description
Heading Data <15:8>	<7:0>	Upper byte of heading data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.29 EUL_DATA_Y_LSB 0x1C

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Roll Data <7:0>							

DATA	bits	Description
Roll Data <7:0>	<7:0>	Lower byte of roll data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.30 EUL_DATA_Y_MSB 0x1D

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Roll Data <15:8>							

DATA	bits	Description
Roll Data <15:8>	<7:0>	Upper byte of Y axis roll data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.31 EUL_DATA_Z_LSB 0x1E

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Pitch Data <7:0>							

DATA	bits	Description
Pitch Data <7:0>	<7:0>	Lower byte of pitch data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.32 EUL_DATA_Z_MSB 0x1F

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Pitch Data <15:8>							

DATA	bits	Description
Pitch Data <15:8>	<7:0>	Upper byte of pitch data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.33 QUA_DATA_W_LSB 0x20

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Quaternion Data W <7:0>							

DATA	bits	Description
Quaternion Data W <7:0>	<7:0>	Lower byte of w axis Quaternion data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.34 QUA_DATA_W_MSB 0x21

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Quaternion Data W <15:8>							

DATA	bits	Description
Quaternion Data W <15:8>	<7:0>	Upper byte of w axis Quaternion data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.35 QUA_DATA_X_LSB 0x22

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Quaternion Data X <7:0>							

DATA	bits	Description
Quaternion Data X <7:0>	<7:0>	Lower byte of X axis Quaternion data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.36 QUA_DATA_X_MSB 0x23

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Quaternion Data X <15:8>							

DATA	bits	Description
Quaternion Data X <15:8>	<7:0>	Upper byte of X axis Quaternion data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.37 QUA_DATA_Y_LSB 0x24

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Quaternion Data Y <7:0>							

DATA	bits	Description
Quaternion Data Y <7:0>	<7:0>	Lower byte of Y axis Quaternion data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.38 QUA_DATA_Y_MSB 0x25

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Quaternion Data Y <15:8>							

DATA	bits	Description
Quaternion Data Y <15:8>	<7:0>	Upper byte of Y axis Quaternion data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.39 QUA_DATA_Z_LSB 0x26

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Quaternion Data Z <7:0>							

DATA	bits	Description
Quaternion Data Z <7:0>	<7:0>	Lower byte of Z axis Quaternion data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.40 QUA_DATA_Z_MSB 0x27

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Quaternion Data Z <15:8>							

DATA	bits	Description
Quaternion Data Z <15:8>	<7:0>	Upper byte of Z axis Quaternion data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.41 LIA_DATA_X_LSB 0x28

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Linear Acceleration Data X <7:0>							

DATA	bits	Description
Linear Acceleration Data X <7:0>	<7:0>	Lower byte of X axis Linear Acceleration data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.42 LIA_DATA_X_MSB 0x29

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Linear Acceleration Data X <15:8>							

DATA	bits	Description
Linear Acceleration Data X <15:8>	<7:0>	Upper byte of X axis Linear Acceleration data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.43 LIA_DATA_Y_LSB 0x2A

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Linear Acceleration Data Y <7:0>							

DATA	bits	Description
Linear Acceleration Data Y <7:0>	<7:0>	Lower byte of Y axis Linear Acceleration data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.44 LIA_DATA_Y_MSB 0x2B

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Linear Acceleration Data Y <15:8>							

DATA	bits	Description
Linear Acceleration Data Y <15:8>	<7:0>	Upper byte of Y axis Linear Acceleration data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.45 LIA_DATA_Z_LSB 0x2C

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Linear Acceleration Data Z <7:0>							

DATA	bits	Description
Linear Acceleration Data Z <7:0>	<7:0>	Lower byte of Z axis Linear Acceleration data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.46 LIA_DATA_Z_MSB 0x2D

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Linear Acceleration Data Z <15:8>							

DATA	bits	Description
Linear Acceleration Data Z <15:8>	<7:0>	Upper byte of Z axis Linear Acceleration data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.47 GRV_DATA_X_LSB 0x2E

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Gravity Vector Data X <7:0>							

DATA	bits	Description
Gravity Vector Data X <7:0>	<7:0>	Lower byte of X axis Gravity Vector data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.48 GRV_DATA_X_MSB 0x2F

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Gravity Vector Data X <15:8>							

DATA	bits	Description
Gravity Vector Data X <15:8>	<7:0>	Upper byte of X axis Gravity Vector data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.49 GRV_DATA_Y_LSB 0x30

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Gravity Vector Data Y <7:0>							

DATA	bits	Description
Gravity Vector Data Y <7:0>	<7:0>	Lower byte of Y axis Gravity Vector data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.50 GRV_DATA_Y_MSB 0x31

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Gravity Vector Data Y <15:8>							

DATA	bits	Description
Gravity Vector Data Y <15:8>	<7:0>	Upper byte of Y axis Gravity Vector data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.51 GRV_DATA_Z_LSB 0x32

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Gravity Vector Data Z <7:0>							

DATA	bits	Description
Gravity Vector Data Z <7:0>	<7:0>	Lower byte of Z axis Gravity Vector data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.52 GRV_DATA_Z_MSB 0x33

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Gravity Vector Data Z <15:8>							

DATA	bits	Description
Gravity Vector Data Z <15:8>	<7:0>	Upper byte of Z axis Gravity Vector data, read only The output units can be selected using the UNIT_SEL register and data output type can be changed by updating the Operation Mode in the OPR_MODE register, see section 3.3

4.3.53 TEMP 0x34

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Temperature							

DATA	bits	Description
Temperature	<7:0>	Temperature data, read only The output units can be selected using the UNIT_SEL register and data output source can be selected by updating the TEMP_SOURCE register, see section 3.6.5.8

4.3.54 CALIB_STAT 0x35

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	SYS Calib Status <0:1>		GYR Calib Status <0:1>		ACC Calib Status <0:1>		MAG Calib Status <0:1>	

DATA	bits	Description
SYS Calib Status <0:1>	<7:6>	Current system calibration status, depends on status of all sensors, read-only Read: 3 indicates fully calibrated; 0 indicates not calibrated
GYR Calib Status <0:1>	<5:4>	Current calibration status of Gyroscope, read-only Read: 3 indicates fully calibrated; 0 indicates not calibrated
ACC Calib Status <0:1>	<3:2>	Current calibration status of Accelerometer, read-only Read: 3 indicates fully calibrated; 0 indicates not calibrated
MAG Calib Status <0:1>	<1:0>	Current calibration status of Magnetometer, read-only Read: 3 indicates fully calibrated; 0 indicates not calibrated

4.3.55 ST_RESULT 0x36

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset					1	1	1	1
Content	Reserved				ST_MCU	ST_GYR	ST_MAG	ST_ACC

DATA	bits	Description
ST_MCU	3	Microcontroller self test result. Read: 1 indicated test passed; 0 indicates test failed
ST_GYR	2	Gyroscope self test result. Read: 1 indicated test passed; 0 indicates test failed
ST_MAG	1	Magnetometer self test result. Read: 1 indicated test passed; 0 indicates test failed
ST_ACC	0	Accelerometer self test result. Read: 1 indicated test passed; 0 indicates test failed

4.3.56 INT_STA 0x37

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0		0	0		
Content	ACC_NM	ACC_AM	ACC_HIG H_G	Reserved	GYR_HIG H_RATE	GYRO_AM	Reserved	Reserved

DATA	bits	Description
ACC_NM	7	Status of Accelerometer no motion or slow motion interrupt, read only Read: 1 indicates interrupt triggered; 0 indicates no interrupt triggered
ACC_AM	6	Status of Accelerometer any motion interrupt, read only Read: 1 indicates interrupt triggered; 0 indicates no interrupt triggered
ACC_HIGH_G	5	Status of Accelerometer high-g interrupt, read only Read: 1 indicates interrupt triggered; 0 indicates no interrupt triggered
GYR_HIGH_RATE	3	Status of gyroscope high rate interrupt, read only Read: 1 indicates interrupt triggered; 0 indicates no interrupt triggered
GYRO_AM	2	Status of gyroscope any motion interrupt, read only Read: 1 indicates interrupt triggered; 0 indicates no interrupt triggered

4.3.57 SYS_CLK_STATUS 0x38

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	ST_MAIN_ CLK

DATA	bits	Description
0	0	Indicates that, it is Free to configure the CLK SRC (External or Internal)
1	0	Indicates that, it is in Configuration state

4.3.58 SYS_STATUS 0x39

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0
Content	System Status Code							

DATA	bits	Description
System Status Code	<7:0>	Read: 0 System idle, 1 System Error, 2 Initializing peripherals 3 System Initialization 4 Executing selftest, 5 Sensor fusion algorithm running, 6 System running without fusion algorithm

4.3.59 SYS_ERR 0x3A

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset								
Content	System Error Code							

DATA	bits	Description
System Error Code	<7:0>	Read the error status from this register if the SYS_STATUS (0x39) register is SYSTEM ERROR (0x01) Read : 0 No error 1 Peripheral initialization error 2 System initialization error 3 Self test result failed 4 Register map value out of range 5 Register map address out of range 6 Register map write error 7 BNO low power mode not available for selected operation mode 8 Accelerometer power mode not available 9 Fusion algorithm configuration error A Sensor configuration error

4.3.60 UNIT_SEL 0x3B

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset	0			0	0	0	0	0
Content	ORI_Android_Windows	reserved		TEMP_Unit	reserved	EUL_Unit	GYR_Unit	ACC_Unit

DATA	bits	Description
ORI_Android_Windows	7	Read: Current selected orientation mode Write: Select orientation mode 0: Windows orientation 1: Android orientation See section 3.6.2 for more details
TEMP_Unit	5	Read: Current selected temperature units Write: Select temperature units 0: Celsius 1: Fahrenheit See section 3.6.1 for more details
EUL_Unit	3	Read: Current selected Euler units Write: Select Euler units 0: Degrees 1: Radians See section 3.6.1 for more details
GYR_Unit	2	Read: Current selected angular rate units Write: Select angular rate units 0: dps 1: rps See section 3.6.1 for more details
ACC_Unit	1	Read: Current selected acceleration units Write: Select acceleration units 0: m/s^2 1: mg See section 3.6.1 for more details

4.3.61 OPR_MODE 0x3D

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access					r/w	r/w	r/w	r/w
Reset								
Content	Reserved				Operation Mode <3:0>			

DATA	bits	Description
Operation Mode <3:0>	<3:0>	Read: Current selected operation mode Write: Select operation mode See section 3.3 for details

4.3.62 PWR_MODE 0x3E

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access							r/w	r/w
Reset								
Content	Reserved						Power Mode <1:0>	

DATA	bits	Description
Power Mode <1:0>	<1:0>	Read: Current selected power mode Write: Select power mode See section 0 for details

4.3.63 SYS_TRIGGER 0x3F

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	w	w	w					w
Reset	0	0	0					0
Content	CLK_SEL	RST_INT	RST_SYS					Self_Test

DATA	bits	Description
CLK_SEL	7	0: Use internal oscillator 1: Use external oscillator. Set this bit only if external crystal is connected
RST_INT	6	Set to reset all interrupt status bits, and INT output
RST_SYS	5	Set to reset system
Self_Test	0	Set to trigger self test

4.3.64 TEMP_SOURCE 0x40

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access							r/w	r/w
Reset								
Content	Reserved						TEMP_Source <1:0>	

DATA	bits	Description
TEMP_Source <1:0>	<1:0>	See section 3.6.5.8 for details

4.3.65 AXIS_MAP_CONFIG 0x41

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access			r/w	r/w	r/w	r/w	r/w	r/w
Reset								
Content	Reserved		Remapped Z axis value		Remapped Y axis value		Remapped X axis value	

DATA	bits	Description
Remapped Z axis value	<5:4>	See section 3.4 for details
Remapped Y axis value	<3:2>	See section 3.4 for details
Remapped X axis value	<1:0>	See section 3.4 for details

4.3.66 AXIS_MAP_SIGN 0x42

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access						r/w	r/w	r/w
Reset								
Content	Reserved					Remapped X axis sign	Remapped Y axis sign	Remapped Z axis sign

DATA	bits	Description
Remapped X axis sign	2	See section 3.4 for details
Remapped Y axis sign	1	See section 3.4 for details
Remapped Z axis sign	0	See section 3.4 for details

4.3.67 ACC_OFFSET_X_LSB 0x55

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset								
Content	Accelerometer Offset X <7:0>							

DATA	bits	Description
Accelerometer Offset X <7:0>	<7:0>	See section 3.6.4 for details

4.3.68 ACC_OFFSET_X_MSB 0x56

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset								
Content	Accelerometer Offset X <15:8>							

DATA	bits	Description
Accelerometer Offset X <15:8>	<7:0>	See section 3.6.4 for details

4.3.69 ACC_OFFSET_Y_LSB 0x57

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset								
Content	Accelerometer Offset Y <7:0>							

DATA	bits	Description
Accelerometer Offset Y <7:0>	<7:0>	See section 3.6.4 for details

4.3.70 ACC_OFFSET_Y_MSB 0x58

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset								
Content	Accelerometer Offset Y <15:8>							

DATA	bits	Description
Accelerometer Offset Y <15:8>	<7:0>	See section 3.6.4 for details

4.3.71 ACC_OFFSET_Z_LSB 0x59

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset								
Content	Accelerometer Offset Z <7:0>							

DATA	bits	Description
Accelerometer Offset Z <7:0>	<7:0>	See section 3.6.4 for details

4.3.72 ACC_OFFSET_Z_MSB 0x5A

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset								
Content	Accelerometer Offset Z <15:8>							

DATA	bits	Description
Accelerometer Offset Z <15:8>	<7:0>	See section 3.6.4 for details

4.3.73 MAG_OFFSET_X_LSB 0x5B

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset								
Content	Magnetometer Data X <7:0>							

DATA	bits	Description
Magnetometer Offset X <7:0>	<7:0>	See section 3.6.4 for details

4.3.74 MAG_OFFSET_X_MSB 0x56C

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset								
Content	Magnetometer Offset X <15:8>							

DATA	bits	Description
Magnetometer Offset X <15:8>	<7:0>	See section 3.6.4 for details

4.3.75 MAG_OFFSET_Y_LSB 0x5D

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset								
Content	Magnetometer Offset Y <7:0>							

DATA	bits	Description
Magnetometer Offset Y <7:0>	<7:0>	See section 3.6.4 for details

4.3.76 MAG_OFFSET_Y_MSB 0x5E

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset								
Content	Magnetometer Offset Y <15:8>							

DATA	bits	Description
Magnetometer Offset Y <15:8>	<7:0>	See section 3.6.4 for details

4.3.77 MAG_OFFSET_Z_LSB 0x5F

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset								
Content	Magnetometer Offset Z <7:0>							

DATA	bits	Description
Magnetometer Offset Z <7:0>	<7:0>	See section 3.6.4 for details

4.3.78 MAG_OFFSET_Z_MSB 0x60

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset								
Content	Magnetometer Offset Z <15:8>							

DATA	bits	Description
Magnetometer Offset Z <15:8>	<7:0>	See section 3.6.4 for details

4.3.79 GYR_OFFSET_X_LSB 0x61

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset								
Content	Gyroscope Data X <7:0>							

DATA	bits	Description
Gyroscope Offset X <7:0>	<7:0>	See section 3.6.4 for details

4.3.80 GYR_OFFSET_X_MSB 0x62

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset								
Content	Gyroscope Offset X <15:8>							

DATA	bits	Description
Gyroscope Offset X <15:8>	<7:0>	See section 3.6.4 for details

4.3.81 GYR_OFFSET_Y_LSB 0x63

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset								
Content	Gyroscope Offset Y <7:0>							

DATA	bits	Description
Gyroscope Offset Y <7:0>	<7:0>	See section 3.6.4 for details

4.3.82 GYR_OFFSET_Y_MSB 0x64

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset								
Content	Gyroscope Offset Y <15:8>							

DATA	bits	Description
Gyroscope Offset Y <15:8>	<7:0>	See section 3.6.4 for details

4.3.83 GYR_OFFSET_Z_LSB 0x65

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset								
Content	Gyroscope Offset Z <7:0>							

DATA	bits	Description
Gyroscope Offset Z <7:0>	<7:0>	See section 3.6.4 for details

4.3.84 GYR_OFFSET_Z_MSB 0x66

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset								
Content	Gyroscope Offset Z <15:8>							

DATA	bits	Description
Gyroscope Offset Z <15:8>	<7:0>	See section 3.6.4 for details

4.3.85 ACC_RADIUS_LSB 0x67

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset								
Content	Accelerometer Radius <7:0>							

DATA	bits	Description
Gyroscope Offset Z <7:0>	<7:0>	See section 3.6.4for details

4.3.86 ACC_RADIUS_MSB 0x68

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset								
Content	Accelerometer Radius <15:8>							

DATA	bits	Description
Gyroscope Offset Z <15:8>	<7:0>	See section 3.6.4 for details

4.3.87 MAG_RADIUS_LSB 0x69

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset								
Content	Magnetometer Radius <7:0>							

DATA	bits	Description
Gyroscope Offset Z <7:0>	<7:0>	See section 3.6.4 for details

4.3.88 MAG_RADIUS_MSB 0x6A

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset								
Content	Magnetometer Radius <15:8>							

DATA	bits	Description
Gyroscope Offset Z <15:8>	<7:0>	See section 3.6.4 for details

4.4 Register description (Page 1)

4.4.1 Page ID 0x07

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset	0	0	0	0	0	0	0	0
Content	Page ID							

DATA	bits	Description
Page ID	<7:0>	Read: Number of currently selected page Write: Change page, 0x00 or 0x01

4.4.2 ACC_Config 0x08

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset	0	0	0	0	1	1	0	1
Content	ACC_PWR_Mode <2:0>			ACC_BW <2:0>			ACC_Range <1:0>	

DATA	bits	Description
ACC_PWR_Mode <2:0>	<7:5>	Read: current selected power mode Write: can only be changed in sensor mode, see section 3.5.2
ACC_BW <2:0>	<4:3>	Read: current selected bandwidth Write: can only be changed in sensor mode, see section 3.5.2
ACC_Range <1:0>	<2:0>	Read: current selected range Write: can only be changed in sensor mode, see section 3.5.2

4.4.3 MAG_Config 0x09

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset	0	0	0	0	1	0	1	1
Content	reserved	MAG_Power_mode <1:0>		MAG_OPR_Mode <1:0>		MAG_Data_output_rate <2:0>		

DATA	bits	Description
MAG_Power_mode <1:0>	<6:5>	Read: current selected power mode Write: can only be changed in sensor mode, see section 3.5.4
MAG_OPR_Mode <1:0>	<4:3>	Read: current selected operation mode Write: can only be changed in sensor mode, see section 3.5.4
MAG_Data_output_rate <2:0>	<2:0>	Read: current selected data output rate Write: can only be changed in sensor mode, see section 3.5.4

4.4.4 GYR_Config_0 0x0A

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset	0	0	1	1	1	0	0	0
Content	reserved		GYR_Bandwidth <2:0>			GYR_Range <2:0>		

DATA	bits	Description
GYR_Bandwidth <2:0>	<5:3>	Read: current selected bandwidth Write: can only be changed in sensor mode, see section 3.5.3
GYR_Range <2:0>	<2:0>	Read: current selected range Write: can only be changed in sensor mode, see section 3.5.3

4.4.5 GYR_Config_1 0x0B

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset	0	0	0	0	0	0	0	0
Content	reserved					GYR_Power_Mode <2:0>		

DATA	bits	Description
GYR_Power_Mode <2:0>	<2:0>	Read: current selected power mode Write: can only be changed in sensor mode, see section 3.5.3

4.4.6 ACC_Sleep_Config 0x0C

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset								
Content	reserved			SLP_DURATION <3:0>				SLP_MODE

DATA	bits	Description
SLP_DURATION <3:0>	<4:1>	Write: The sleep duration for accelerometer low power mode can be only configured in the sensor operation mode where no fusion library is running. Following sleep phase duration is possible to set.
SLP_MODE	0	The sleep timer mode for accelerometer low power mode can be only configured in the sensor operation mode where no fusion library is running Write 0: use event driven time-base mode 1: use equidistant sampling time-base mode

SLP_DURATION	Accelerometer Sleep Phase Duration
0000b	0.5 ms
0001b	0.5 ms
0010b	0.5 ms
0011b	0.5 ms
0100b	0.5 ms
0101b	0.5 ms
0110b	1 ms
0111b	2 ms
1000b	4 ms
1001b	6 ms
1010b	10 ms
1011b	25 ms
1100b	50 ms
1101b	100 ms
1110b	500 ms
1111b	1000 ms

4.4.7 GYR_Sleep_Config 0x0D

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset								
Content	reserved		AUTO_SLP_DURATION <2:0>			SLP_DURATION <2:0>		

DATA	bits	Description
AUTO_SLP_DURATION <2:0>	<5:3>	The Gyroscope can be configures in the advanced power mode to optimize the power consumption. This can be only done if the selected operation mode in sensor mode. The auto sleep duration is the wake up duration of gyroscope during the duty cycling between normal and fast-power up mode. Possible configuration for auto sleep duration are:
SLP_DURATION <2:0>	<2:0>	The Gyroscope can be configures in the advanced power mode to optimize the power consumption. This can be only done if the selected operation mode in sensor mode. The sleep duration is the sleep time of gyroscope during the duty cycling between normal and fast-power up mode. Possible configuration for sleep duration are:

Auto sleep duration	Time (ms)
000b	Not allowed
001b	4 ms
010b	5 ms
011b	8 ms
100b	10 ms
101b	15 ms
110b	20 ms
111b	40 ms

Sleep duration	Time (ms)
000b	2 ms
001b	4 ms
010b	5 ms
011b	8 ms
100b	10 ms
101b	15 ms
110b	18 ms
111b	20 ms

The only restriction for the use of the power save mode comes from the configuration of the digital filter bandwidth of gyroscope. For each bandwidth configuration, minimum auto sleep duration must be ensured. For example, for bandwidth = 47Hz, the minimum auto sleep duration is 5ms. This is specified in the table below. For sleep duration, there is no restriction.

Gyroscope bandwidth (Hz)	Mini Autosleep duration (ms)
32 Hz	20 ms
64 Hz	10 ms
12 Hz	20 ms
23 Hz	10 ms
47 Hz	5 ms
116 Hz	4 ms
230 Hz	4 ms
Unfiltered (523 Hz)	4 ms

4.4.8 INT_MSK 0x0F

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset	0	0	0		0	0		
Content	ACC_NM	ACC_AM	ACC_HIGH_G	reserved	GYR_HIGH_RATE	GYRO_AM	reserved	reserved

DATA	bits	Description
ACC_NM	7	Masking of Accelerometer no motion or slow motion interrupt, when enabled the interrupt will update the INT_STA register and trigger a change on the INT pin, when disabled only the INT_STA register will be updated. Read: 1: Enabled / 0: Disabled Write: 1: Enable / 0: Disable
ACC_AM	6	Masking of Accelerometer any motion interrupt, when enabled the interrupt will update the INT_STA register and trigger a change on the INT pin, when disabled only the INT_STA register will be updated. Read: 1: Enabled / 0: Disabled Write: 1: Enable / 0: Disable
ACC_HIGH_G	5	Masking of Accelerometer high-g interrupt, when enabled the interrupt will update the INT_STA register and trigger a change on the INT pin, when disabled only the INT_STA register will be updated. Read: 1: Enabled / 0: Disabled Write: 1: Enable / 0: Disable
GYR_HIGH_RATE	3	Masking of gyroscope high rate interrupt, when enabled the interrupt will update the INT_STA register and trigger a change on the INT pin, when disabled only the INT_STA register will be updated. Read: 1: Enabled / 0: Disabled Write: 1: Enable / 0: Disable
GYRO_AM	2	Masking of gyroscope any motion interrupt, when enabled the interrupt will update the INT_STA register and trigger a change on the INT pin, when disabled only the INT_STA register will be updated. Read: 1: Enabled / 0: Disabled Write: 1: Enable / 0: Disable

4.4.9 INT_EN 0x10

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset	0	0	0		0	0		
Content	ACC_NM	ACC_AM	ACC_HIGH_G	reserved	GYR_HIGH_RATE	GYRO_AM	reserved	reserved

DATA	bits	Description
ACC_NM	7	Status of Accelerometer no motion or slow motion interrupt Read: 1: Enabled / 0: Disabled Write: 1: Enable / 0: Disable interrupt
ACC_AM	6	Status of Accelerometer any motion interrupt Read: 1: Enabled / 0: Disabled Write: 1: Enable / 0: Disable interrupt
ACC_HIGH_G	5	Status of Accelerometer high-g interrupt Read: 1: Enabled / 0: Disabled Write: 1: Enable / 0: Disable interrupt
GYR_HIGH_RATE	3	Status of gyroscope high rate interrupt Read: 1: Enabled / 0: Disabled Write: 1: Enable / 0: Disable interrupt
GYRO_AM	2	Status of gyroscope any motion interrupt Read: 1: Enabled / 0: Disabled Write: 1: Enable / 0: Disable interrupt

4.4.10 ACC_AM_THRES 0x11

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset	0	0	0	1	0	1	0	0
Content	Accelerometer Any motion threshold							

DATA	bits	Description
Accelerometer Any motion threshold	<7:0>	Threshold used for the any-motion interrupt. The threshold value is dependent on the accelerometer range selected in the ACC_Config register. 1 LSB = 3.91 mg (2-g range) 1 LSB = 7.81 mg (4-g range) 1 LSB = 15.63 mg (8-g range) 1 LSB = 31.25 mg (16-g range)

4.4.11 ACC_INT_Settings 0x12

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset	0	0	0	0	0	0	1	1
Content	HG_Z_AXIS	HG_Y_AXIS	HG_X_AXIS	AM/NM_Z_AXIS	AM/NM_Y_AXIS	AM/NM_X_AXIS	AM_DUR <1:0>	

DATA	bits	Description
HG_Z_AXIS	7	Select which axis of the accelerometer is used to trigger a high-G interrupt 1: Enabled; 0: Disabled
HG_Y_AXIS	6	Select which axis of the accelerometer is used to trigger a high-G interrupt 1: Enabled; 0: Disabled
HG_X_AXIS	5	Select which axis of the accelerometer is used to trigger a high-G interrupt 1: Enabled; 0: Disabled
AM/NM_Z_AXIS	4	Select which axis of the accelerometer is used to trigger a any motion or no motion interrupt 1: Enabled; 0: Disabled
AM/NM_Y_AXIS	3	Select which axis of the accelerometer is used to trigger a any motion or no motion interrupt 1: Enabled; 0: Disabled
AM/NM_X_AXIS	2	Select which axis of the accelerometer is used to trigger a any motion or no motion interrupt 1: Enabled; 0: Disabled
AM_DUR <1:0>	<1:0>	Any motion interrupt triggers if [AM_DUR<1:0>+1] consecutive data points are above the any motion interrupt threshold define in ACC_AM_THRES register

4.4.12 ACC_HG_DURATION 0x13

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset	0	0	0	0	1	1	1	1
Content	Accelerometer High G Duration							

DATA	bits	Description
Accelerometer High G Duration	<7:0>	The high-g interrupt trigger delay according to [ACC_HG_DURATION + 1] * 2 ms in a range from 2 ms to 512 ms;

4.4.13 ACC_HG_THRES 0x14

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset	1	1	0	0	0	0	0	0
Content	Accelerometer High G Threshold							

DATA	bits	Description
Accelerometer High G Threshold	<7:0>	Threshold used high-g interrupt. The threshold value is dependent on the accelerometer range selected in the ACC_Config register. 1 LSB = 7.81 mg (2-g range 1 LSB = 15.63 mg (4-g range) 1 LSB = 31.25 mg (8-g range) 1 LSB = 62.5 mg (16-g range)

4.4.14 ACC_NM_THRES 0x15

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset	0	0	0	0	1	0	1	0
Content	Accelerometer NO/SLOW motion threshold							

DATA	bits	Description
Accelerometer NO/SLOW motion threshold	<7:0>	Threshold used for the Slow motion or no motion interrupt. The threshold value is dependent on the accelerometer range selected in the ACC_Config register. 1 LSB = 3.91 mg (2-g range) 1 LSB = 7.81 mg (4-g range) 1 LSB = 15.63 mg (8-g range) 1 LSB = 31.25 mg (16-g range)

4.4.15 ACC_NM_SET 0x16

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset		0	0	0	1	0	1	1
Content	reserved	slo_no_mot_dur <5:0>						SMNM

DATA	bits	Description
slo_no_mot_dur <5:0>	<6:1>	Function depends on whether the slow-motion or no-motion interrupt function has been selected. If the slow-motion interrupt function has been enabled (SMNM = ‘0’) then [slo_no_mot_dur<1:0>+1] consecutive slope data points must be above the slow/no-motion threshold (ACC_NM_THRES) for the slow-/no-motion interrupt to trigger. If the no-motion interrupt function has been enabled (SMNM = ‘1’) then slo_no_motion_dur<5:0> defines the time for which no slope data points must exceed the slow/no-motion threshold (ACC_NM_THRES) for the slow/no-motion interrupt to trigger. The delay time in seconds may be calculated according with the following equation: slo_no_mot_dur<5:4>=’b00’ → [slo_no_mot_dur<3:0> + 1] slo_no_mot_dur<5:4>=’b01’ → [slo_no_mot_dur<3:0> * 4 + 20] slo_no_mot_dur<5>=’1’ → [slo_no_mot_dur<4:0> * 8 + 88]
SMNM	0	Select slow motion or no motion interrupt 0: Slow motion; 1: No motion

4.4.16 GYR_INT_SETTING 0x17

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset	0	0	0	0	0	0	0	0
Content	HR_FILT	AM_FILT	HR_Z_AXIS	HR_Y_AXIS	HR_X_AXIS	AM_Z_AXIS	AM_Y_AXIS	AM_X_AXIS

DATA	bits	Description
HR_FILT	7	‘1’ (‘0’) selects unfiltered (filtered) data for high rate interrupt
AM_FILT	6	‘1’ (‘0’) selects unfiltered (filtered) data for any motion interrupt
HR_Z_AXIS	5	1’ (‘0’) enables (disables) high rate interrupt for z-axis
HR_Y_AXIS	4	1’ (‘0’) enables (disables)) high rate interrupt for y-axis
HR_X_AXIS	3	1’ (‘0’) enables (disables)) high rate interrupt for x-axis
AM_Z_AXIS	2	1’ (‘0’) enables (disables) any motion interrupt for z-axis
AM_Y_AXIS	1	1’ (‘0’) enables (disables) any motion interrupt for y-axis
AM_X_AXIS	0	1’ (‘0’) enables (disables) any motion interrupt for x-axis

4.4.17 GYR_HR_X_SET 0x18

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset	0	0	0	0	0	0	0	1
Content	reserved	HR_X_THRES_HYST <1:0>		HR_X_Threshold <4:0>				

DATA	bits	Description
HR_X_THRES_HYST <1:0>	<6:5>	High rate hysteresis for X axis = (255 + 256 * HR_X_THRES_HYST) *4 LSB The high rate value scales with the range setting 1 LSB = 62.26°/s in 2000°/s-range 1 LSB = 31.13°/s in 1000°/s-range 1 LSB = 15.56°/s in 500°/s -range ...
HR_X_Threshold <4:0>	<4:0>	High rate threshold is for the gyroscope X axis. The threshold value is dependent on the gyroscope range selected in the GRY_Config_0 register. 1 LSB = 62.5°/s in 2000°/s-range 1 LSB = 31.25°/s in 1000°/s-range 1 LSB = 15.625°/s in 500°/s -range ...

4.4.18 GYR_DUR_X 0x19

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset	0	0	0	1	1	0	0	1
Content	HR_X_Duration							

DATA	bits	Description
HR_X_Duration	<7:0>	High rate duration = (1 + HR_X_Duration)*2.5ms

4.4.19 GYR_HR_Y_SET 0x1A

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	1
Content	reserved	HR_Y_THRES_HYST <1:0>		HR_Y_Threshold <4:0>				

DATA	bits	Description
HR_Y_THRES_HYST <1:0>	<6:5>	High rate hysteresis for Y axis = (255 + 256 * HR_Y_THRES_HYST) *4 LSB The high rate value scales with the range setting 1 LSB = 62.26°/s in 2000°/s-range 1 LSB = 31.13°/s in 1000°/s-range 1 LSB = 15.56°/s in 500°/s -range ...
HR_Y_Threshold <4:0>	<4:0>	High rate threshold is for the gyroscope Y axis. The threshold value is dependent on the gyroscope range selected in the GRY_Config_0 register. 1 LSB = 62.5°/s in 2000°/s-range 1 LSB = 31.25°/s in 1000°/s-range 1 LSB = 15.625°/s in 500°/s -range ...

4.4.20 GYR_DUR_Y 0x1B

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset	0	0	0	1	1	0	0	1
Content	HR_Y_Duration							

DATA	bits	Description
HR_Y_Duration	<7:0>	High rate duration = (1 + HR_Y_Duration)*2.5ms

4.4.21 GYR_HR_Z_SET 0x1C

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	1
Content	reserved	HR_Z_THRES_HYST <1:0>		HR_Z_Threshold <4:0>				

DATA	bits	Description
HR_Z_THRES_HYST <1:0>	<6:5>	High rate hysteresis for Z axis = (255 + 256 * HR_Z_THRES_HYST) *4 LSB The high rate value scales with the range setting 1 LSB = 62.26°/s in 2000°/s-range 1 LSB = 31.13°/s in 1000°/s-range 1 LSB = 15.56°/s in 500°/s -range ...
HR_Z_Threshold <4:0>	<4:0>	High rate threshold is for the gyroscope Z axis. The threshold value is dependent on the gyroscope range selected in the GRY_Config_0 register. 1 LSB = 62.5°/s in 2000°/s-range 1 LSB = 31.25°/s in 1000°/s-range 1 LSB = 15.625°/s in 500°/s -range ...

4.4.22 GYR_DUR_Z 0x1D

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset	0	0	0	1	1	0	0	1
Content	HR_Z_Duration							

DATA	bits	Description
HR_Z_Duration	<7:0>	High rate duration = (1 + HR_Z_Duration)*2.5ms

4.4.23 GYR_AM_THRES 0x1E

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset	0	0	0	0	0	1	0	0
Content	reserved	Gyro Any Motion Threshold <6:0>						

DATA	bits	Description
Gyro Any Motion Threshold <6:0>	<6:0>	Any motion threshold is for the gyroscope any motion interrupt. The threshold value is dependent on the gyroscope range selected in the GRY_Config_0 register. 1 LSB = 1 °/s in 2000°/s-range 1 LSB = 0.5°/s in 1000°/s-range 1 LSB = 0.25°/s in 500°/s -range ...

4.4.24 GYR_AM_SET 0x1F

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Access	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
Reset	0	0	0	0	1	0	1	0
Content	reserved				Awake Duration <1:0>		Slope Samples <1:0>	

DATA	bits	Description
Awake Duration <1:0>	<3:2>	0=8 samples, 1=16 samples, 2=32 samples, 3=64 samples
Slope Samples <1:0>	<1:0>	Any motion interrupt triggers if [Slope Samples + 1]*4 consecutive data points are above the any motion interrupt threshold define in GYRO_AM_THRES register

4.5 Digital Interface

The BNO055 supports two digital interfaces for communication between the salve and host device: I^2C which supports the HID-I2C protocol and I2C Standard and Fast modes; and the UART interface.

The active interface is selected by the state of the protocol select pins (PS1 and PS0), Table 4-4 shows the mapping between the protocol select pins and the selected interface mode.

Table 4-4: protocol select pin mapping

PS1	PS0	Functionality
0	0	Standard/Fast I2C Interface
0	1	HID over I2C
1	0	UART Interface
1	1	Reserved

It is not allowed to keep the protocol select pins floating.

Both digital interfaces share partially the same pins, the pin mapping for each interface is shown in Table 4-5.

Table 4-5: Mapping of digital interface pins

PIN	I2C Interfaces (PS1=0b0)	UART Interface (PS1.PS0=0b10)
COM0	SDA	Tx
COM1	SCL	Rx
COM2	GNDIO	
COM3	I2C address select	

The following table shows the electrical specifications of the interface pins:

Table 4-6: Electrical specification of the interface pins

Parameter	Symbol	Condition	Min	Typ	Max	Units
Pull-up Resistance, COM3 pin	R_{up}	Internal Pull-up Resistance to VDDIO	20	40	60	kΩ
Input Capacitance	C_{in}			5	10	pF
I^2C Bus Load Capacitance (max. drive capability)	C_{I2C_Load}				400	pF

4.6 I2C Protocol

The I²C bus uses SCL (= SCx pin, serial clock) and SDA (= SDx pin, serial data input and output) signal lines. Both lines are connected to V_{DDIO} externally via pull-up resistors so that they are pulled high when the bus is free.

The I²C interface of the BNO055 is compatible with the I²C Specification UM10204 Rev. 03 (19 June 2007), available at http://www.nxp.com. The BNO055 supports I²C standard mode and fast mode, only 7-bit address mode is supported. The BNO055 I²C interface uses clock stretching.

The default I²C address of the BNO055 device is 0101001b (0x29). The alternative address 0101000b (0x28), in I2C mode the input pin COM3 can be used to select between the primary and alternative I2C address as shown in Table 4-7.

Table 4-7: I2C address selection

I2C configuration	COM3_state	I2C address
Slave	HIGH	0x29
Slave	LOW	0x28
HID-I2C	X	0x40

The timing specification for I²C of the BNO055 is given in Table 4-8: I²C timings:

Table 4-8: I²C timings

Parameter	Symbol	Condition	Min	Max	Units
Clock Frequency	f_{SCL}			400	kHz
SCL Low Period	t_{LOW}		1.3		µs
SCL High Period	t_{HIGH}		0.6		
SDA Setup Time	t_{SUDAT}		0.1		
SDA Hold Time	t_{HDDAT}		0.0		
Setup Time for a repeated Start Condition	t_{SUSTA}		0.6		
Hold Time for a Start Condition	t_{HDSTA}		0.6		
Setup Time for a Stop Condition	t_{SUSTO}		0.6		
Time before a new Transmission can start	t_{BUF}		1.3		
Idle time between write accesses, normal mode, standby mode, low-power mode 2	$t_{IDLE\_wacc\_nm}$		2		µs
Idle time between write accesses, suspend mode, low-power mode 1	$t_{IDLE\_wacc\_sum}$		450		µs

Figure 5: I²C timing diagram shows the definition of the I²C timings given in Table 4-8:

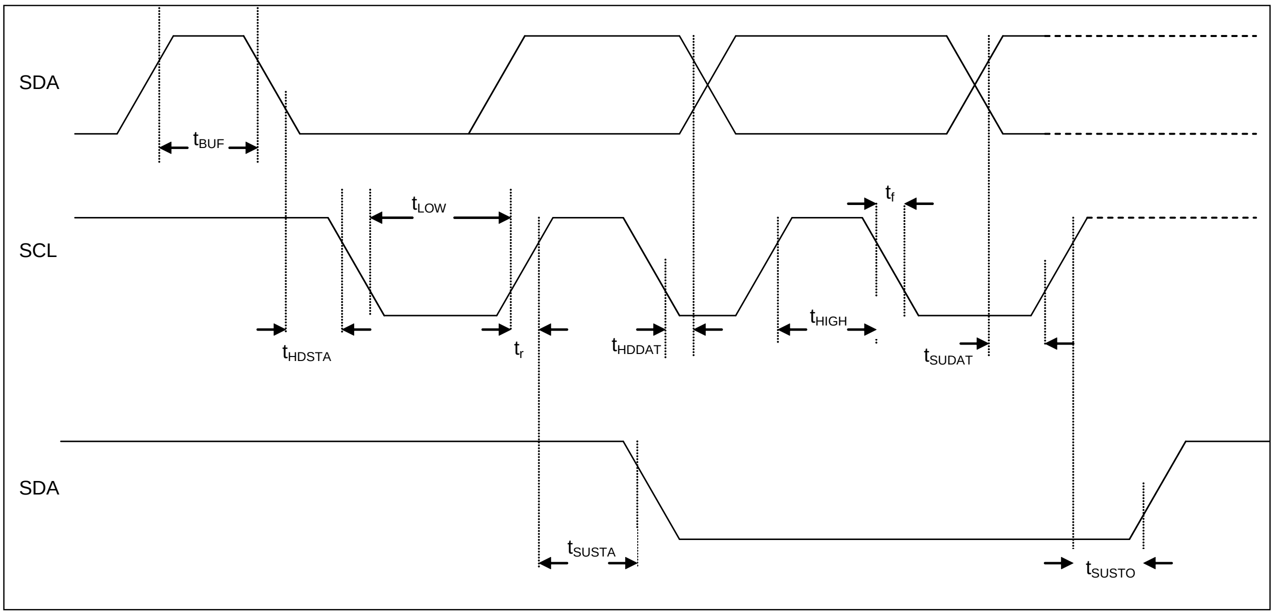


Figure 5: I²C timing diagram

The I²C protocol works as follows:

START: Data transmission on the bus begins with a high to low transition on the SDA line while SCL is held high (start condition (S) indicated by I²C bus master). Once the START signal is transferred by the master, the bus is considered busy.

STOP: Each data transfer should be terminated by a Stop signal (P) generated by master. The STOP condition is a low to HIGH transition on SDA line while SCL is held high.

ACK: Each byte of data transferred must be acknowledged. It is indicated by an acknowledge bit sent by the receiver. The transmitter must release the SDA line (no pull down) during the acknowledge pulse while the receiver must then pull the SDA line low so that it remains stable low during the high period of the acknowledge clock cycle.

In the following diagrams these abbreviations are used:

S	Start
P	Stop
ACKS	Acknowledge by slave
ACKM	Acknowledge by master
NACKM	Not acknowledge by master
RW	Read / Write

A START immediately followed by a STOP (without SCL toggling from ´VDDIO´ to ´GND´) is not supported. If such a combination occurs, the STOP is not recognized by the device.

I²C write access:
I²C write access can be used to write a data byte in one sequence. The sequence begins with start condition generated by the master, followed by 7 bits slave address and a write bit (RW = 0). The slave sends an acknowledge bit (ACK = 0) and releases the bus. Then the master sends the one byte register address. The slave again acknowledges the transmission

and waits for the 8 bits of data which shall be written to the specified register address. After the slave acknowledges the data byte, the master generates a stop signal and terminates the writing protocol.

Example of an I²C write access to the BNO055 (i2c address in this case: 0101000b = 0x28):

Start	Slave address							RW	ACKS	dummy	Register address (0x00 .. 0x7F)							ACKS	Data								ACKS	Stop
S	0	1	0	1	0	0	0	0	A	x	x	x	x	x	x	x	x	A	x	x	x	x	x	x	x	x	A	P

Figure 6: I²C write

I²C read access:

I²C read access also can be used to read one or multiple data bytes in one sequence. A read sequence consists of a one-byte I²C write phase followed by the I²C read phase. The two parts of the transmission must be separated by a repeated start condition (Sr). The I²C write phase addresses the slave and sends the register address to be read. After slave acknowledges the transmission, the master generates again a start condition and sends the slave address together with a read bit (RW = 1). Then the master releases the bus and waits for the data bytes to be read out from slave. After each data byte the master has to generate an acknowledge bit (ACK = 0) to enable further data transfer. A NACKM (ACK = 1) from the master stops the data being transferred from the slave. The slave releases the bus so that the master can generate a STOP condition and terminate the transmission.

The register address is automatically incremented and, therefore, more than one byte can be sequentially read out. Once a new data read transmission starts, the start address will be set to the register address specified in the latest I²C write command. By default the start address is set at 0x00. In this way repetitive multi-bytes reads from the same starting address are possible.

Example of an I²C read access to the BNO055:

Start	Slave address							RW	ACKS	dummy	Register address (0x08)							ACKS
S	0	1	0	1	0	0	0	0	A	x	0	0	0	1	0	0	0	A

Start	Slave address							RW	ACKS	Read data (0x08)								ACKM	Read data (0x09)								ACKM	
Sr	0	1	0	1	0	0	0	1	A	x	x	x	x	x	x	x	x	A	x	x	x	x	x	x	x	x	A	...

	ACKS	Read data (0x0A)								ACKM	Read data (0x0B)								ACKM	
...	A	x	x	x	x	x	x	x	x	A	x	x	x	x	x	x	x	x	A	...

	ACKS	Read data (0x0C)								ACKM	Read data (0x0D)								NACKM	Stop
...	A	x	x	x	x	x	x	x	x	A	x	x	x	x	x	x	x	x	NA	P

Figure 7: I²C multiple read

4.7 UART Protocol

The BNO055 supports UART interface with the following settings: 115200 bps, 8N1 (8 data bits, no parity bit, one stop bit). The maximum length support for read and write is 128 Byte. The packet structure for register read and write are described below.

Register write

Command:

Byte 1	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**.....**	**Byte (n+4)**
Start Byte	Write	Reg addr	Length	Data 1	.....	Data n
0xAA	0x00	<..>	<..>	<..>	.....	<..>

Acknowledge Response:

Byte 1	**Byte 2**
Response Header	Status
0xEE	0x01: WRITE_SUCCESS 0x03: WRITE_FAIL 0x04: REGMAP_INVALID_ADDRESS 0x05: REGMAP_WRITE_DISABLED 0x06: WRONG_START_BYTE 0x07: BUS_OVER_RUN_ERROR 0X08: MAX_LENGTH_ERROR 0x09: MIN_LENGTH_ERROR 0x0A: RECEIVE_CHARACTER_TIMEOUT

Register read

Command:

Byte 1	**Byte 2**	**Byte 2**	**Byte 3**
Start Byte	Read	Reg addr	Length
0xAA	0x01	<..>	<..>

Read Success Response:

Byte 1	**Byte 2**	**Byte 3**	**.....**	**Byte (n+2)**
ResponseByte	length	Data 1	.....	Data n
0xBB	<..>			

Read Failure or Acknowledge Response:

Byte 1	**Byte 2**
Response Header	Status
0xEE	0x02: READ_FAIL 0x04: REGMAP_INVALID_ADDRESS 0x05: REGMAP_WRITE_DISABLED 0x06: WRONG_START_BYTE 0x07: BUS_OVER_RUN_ERROR 0X08: MAX_LENGTH_ERROR 0x09: MIN_LENGTH_ERROR 0x0A: RECEIVE_CHARACTER_TIMEOUT

4.8 HID over I2C

HID over I2C is a standard interface protocol to connect devices with hosts via I2C. The main advantage of HID is that there exist generic drivers for different input devices (such as sensors) which can be used with sensors that implement the corresponding well defined HID profiles. HID over I2C describes how messages (reports and events) are exchanged between the device and the host. A descriptor of the structure of these reports is provided by the device and read by the host during initialization of the device at host system start.
For detailed information on HID please refer to the HID over I2C documentation from Microsoft.

5. Pin-out and connection diagram

5.1 Pin-out

The pin-out of the LGA package is shown in Figure 8 and the pin function is described in Table 5-1.

Figure 8: Pin-out bottom view

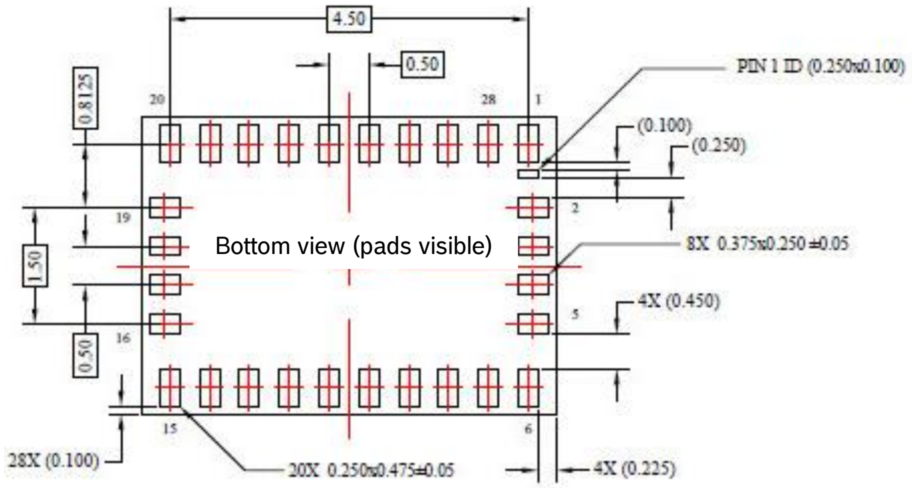
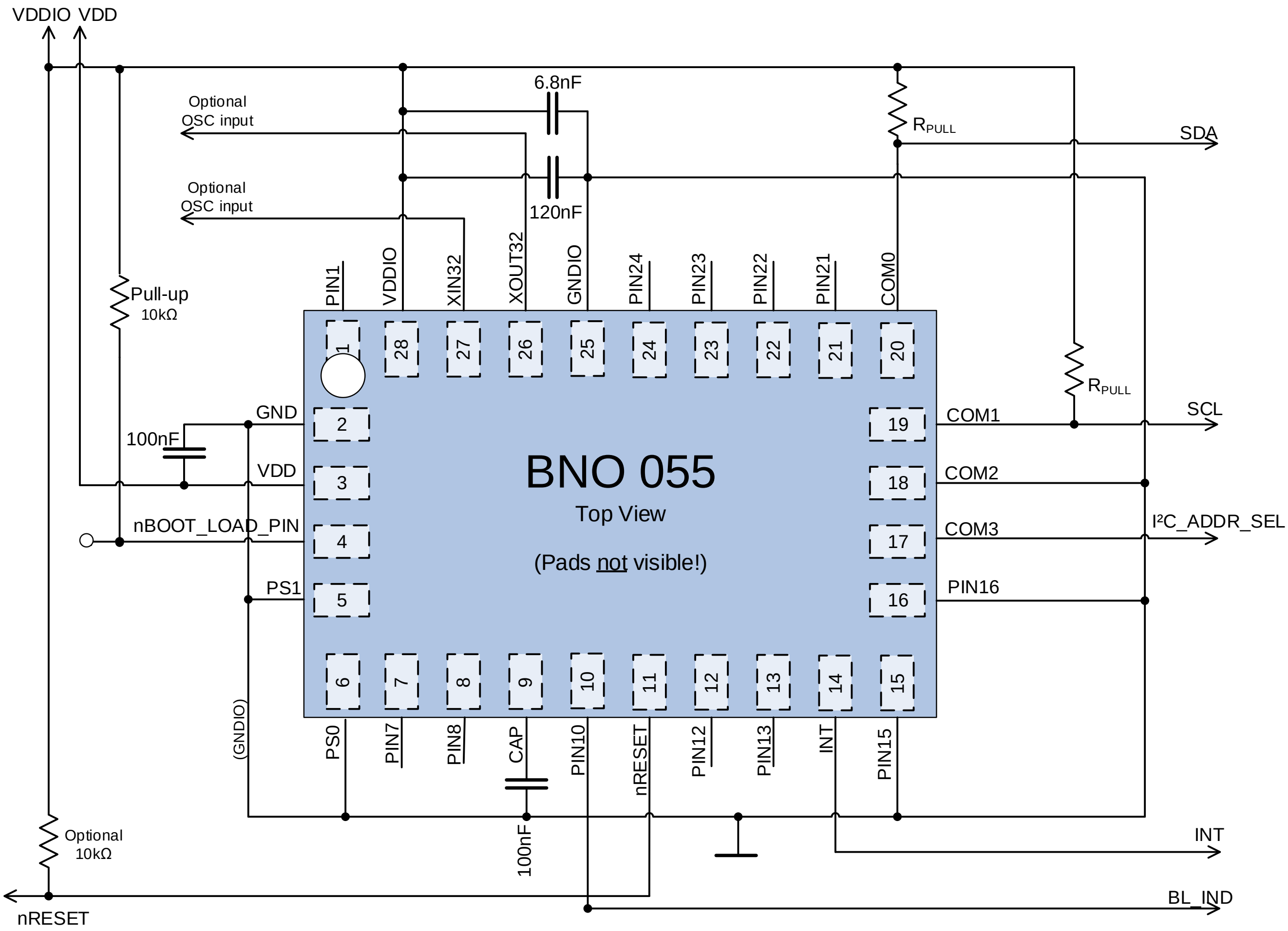


Table 5-1: Pin description

Pin #	Name	I/O Type	Description	Function		
				I2C	UART	HID-I2C
1	PIN1	--	Do not connect	DNC		
2	GND	Ground	GND	GND		
3	VDD	Supply	VDD	VDD		
4	nBOOT_LOAD_PIN	Digital In	Bootloader mode select pin (active low)	nBOOT_LOAD_PIN		
5	PS1	Digital In	Protocol select pin 1	GNDIO	VDDIO	GNDIO
6	PS0	Digital In	Protocol select pin 2	GNDIO	GNDIO	VDDIO
7	PIN7	--	Do not connect	DNC		
8	PIN8	--	Do not connect	DNC		
9	CAP	--	External capacitor	CAP		
10	BL_IND	Digital Out	Boot loader indicator	DNC		
11	nRESET	--	Reset pin (active low)	nRESET		
12	PIN12	--	Do not connect	DNC		
13	PIN13	--	Do not connect	DNC		
14	INT	Digital Out	Interrupt output	Interrupt		
15	PIN15	Ground	Connect to GNDIO	GNDIO		
16	PIN16	Ground	Connect to GNDIO	GNDIO		
17	COM3	Digital In	Digital interface pin 3	I2C address select	GNDIO	GNDIO
18	COM2	Digital I/O	Digital interface pin 2	GNDIO		
19	COM1	Digital I/O	Digital interface pin 1	SCL	Rx	SCL
20	COM0	Digital I/O	Digital interface pin 0	SDA	Tx	SDA
21	PIN21	--	Do not connect	DNC		
22	PIN22	--	Do not connect	DNC		
23	PIN23	--	Do not connect	DNC		
24	PIN24	--	Do not connect	DNC		
25	GNDIO	Ground	GNDIO	GNDIO		
26	XOUT32	Digital Out	Optional OSC port	OSC Output		
27	XIN32	Digital In	Optional OSC port	OSC Input		
28	VDDIO	Supply	VDDIO	VDDIO		

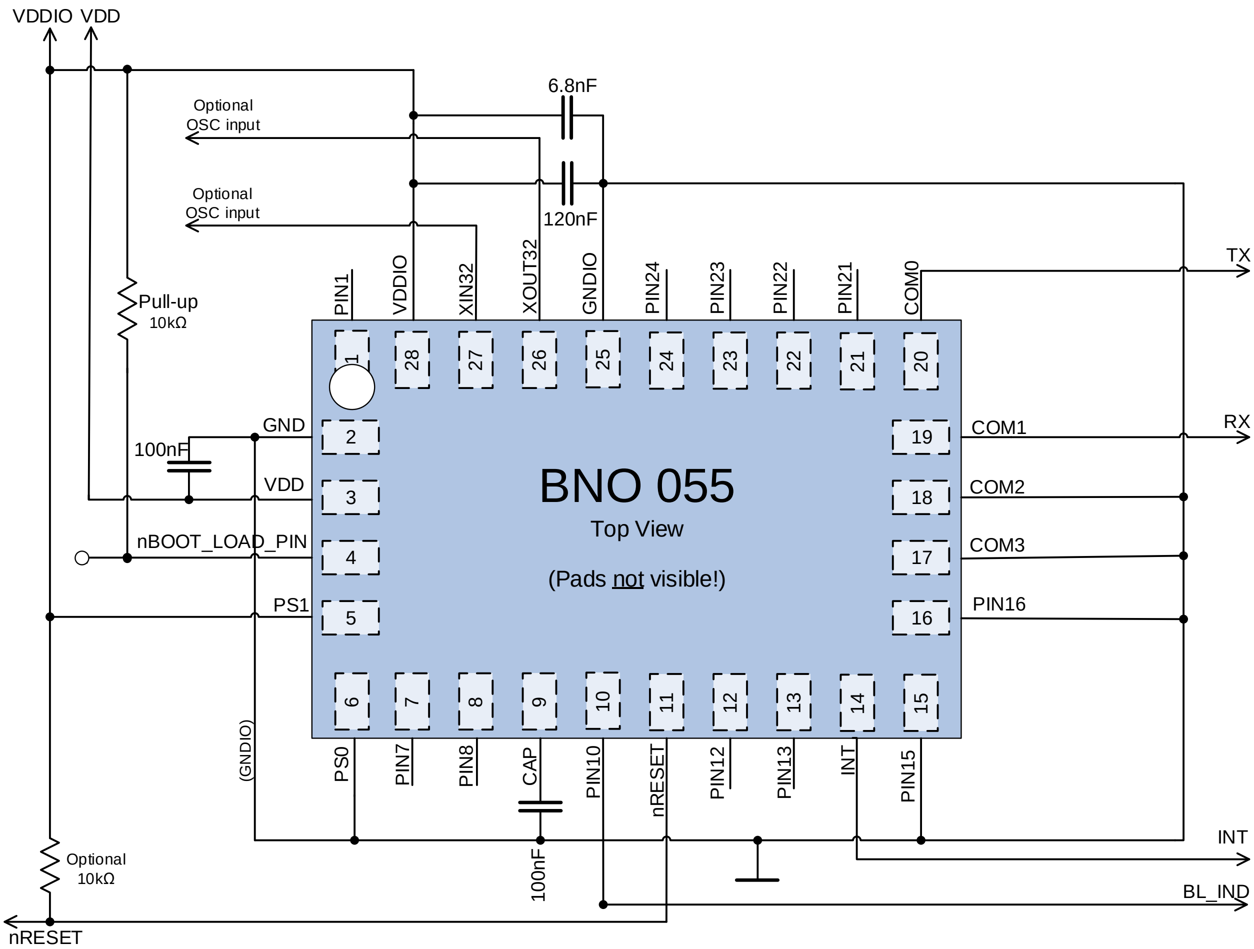
5.2 Connection diagram I²C

Figure 9: I²C connection diagram



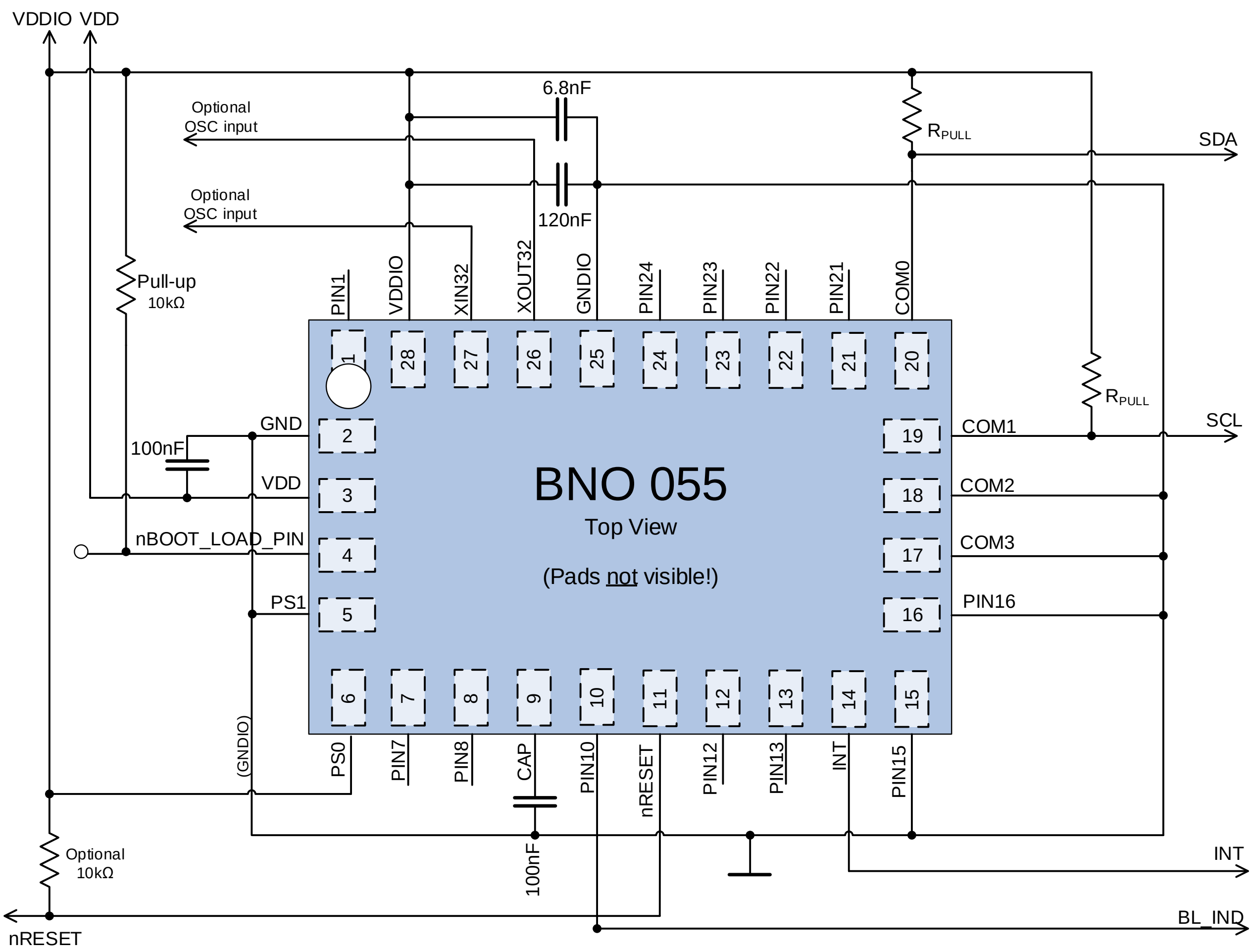
5.3 Connection diagram UART

Figure 10: UART connection diagram



5.4 Connection diagram HID-I2C

Figure 11 : HID via IC connection diagram



5.5 XOUT32 & XIN32 Connections

The BNO055 can run from an internal or external 32 KHz clock source. By default, the internal clock is selected.
An External clock can be selected by setting bit CLK_SEL in the SYSTEM_TRIGGER register. An external 32 KHz crystal oscillator has to be connected to the pins XIN32 and XOUT32 as shown below.

To get the best performance out of BNO055, it is recommended to use the external crystal.

5.5.1 External 32kHz Crystal Oscillator

Figure 12 : External 32kHz Crystal Oscillator with Load Capacitor

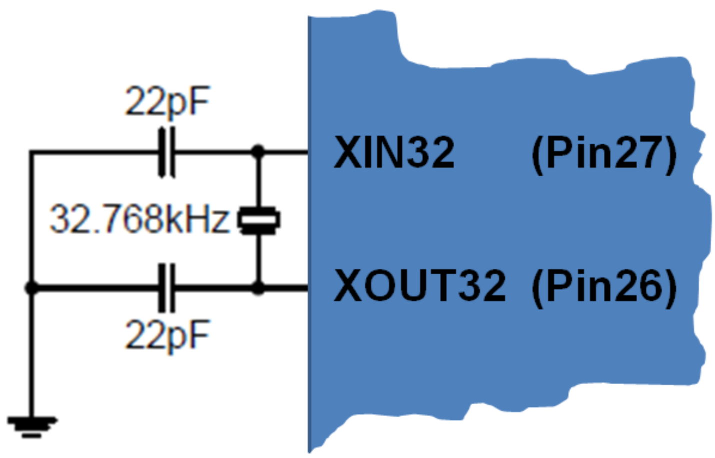


Table 5-2: Crystal Oscillator Source Connections

Pin Name	Recommended Pin Connection	Description
XIN32	Load capacitor 22pF[8][9]	Timer oscillator input
XOUT32	Load capacitor 22pF[8][9]	Timer oscillator output

5.5.2 Internal clock mode

The internal clock can be selected by clearing bit CLK_SEL in the SYSTEM_TRIGGER register. When an internal clock is used, both pins XIN32 and XOUT32 can be left open. The internal clock of the BNO055 can have clock deviation up to $\pm$3%

[8] These values are given only as typical example.
[9] Decoupling capacitor should be placed close to the device for each supply pin pair in the signal group.

6. Package

6.1 Outline dimensions

The sensor package is a standard LGA package; dimensions are shown in the following diagram. Units are in mm. Note: Unless otherwise specified tolerance = decimal ±0.1mm.
The chapter 3.5 provides information regarding the sensor axis orientation.

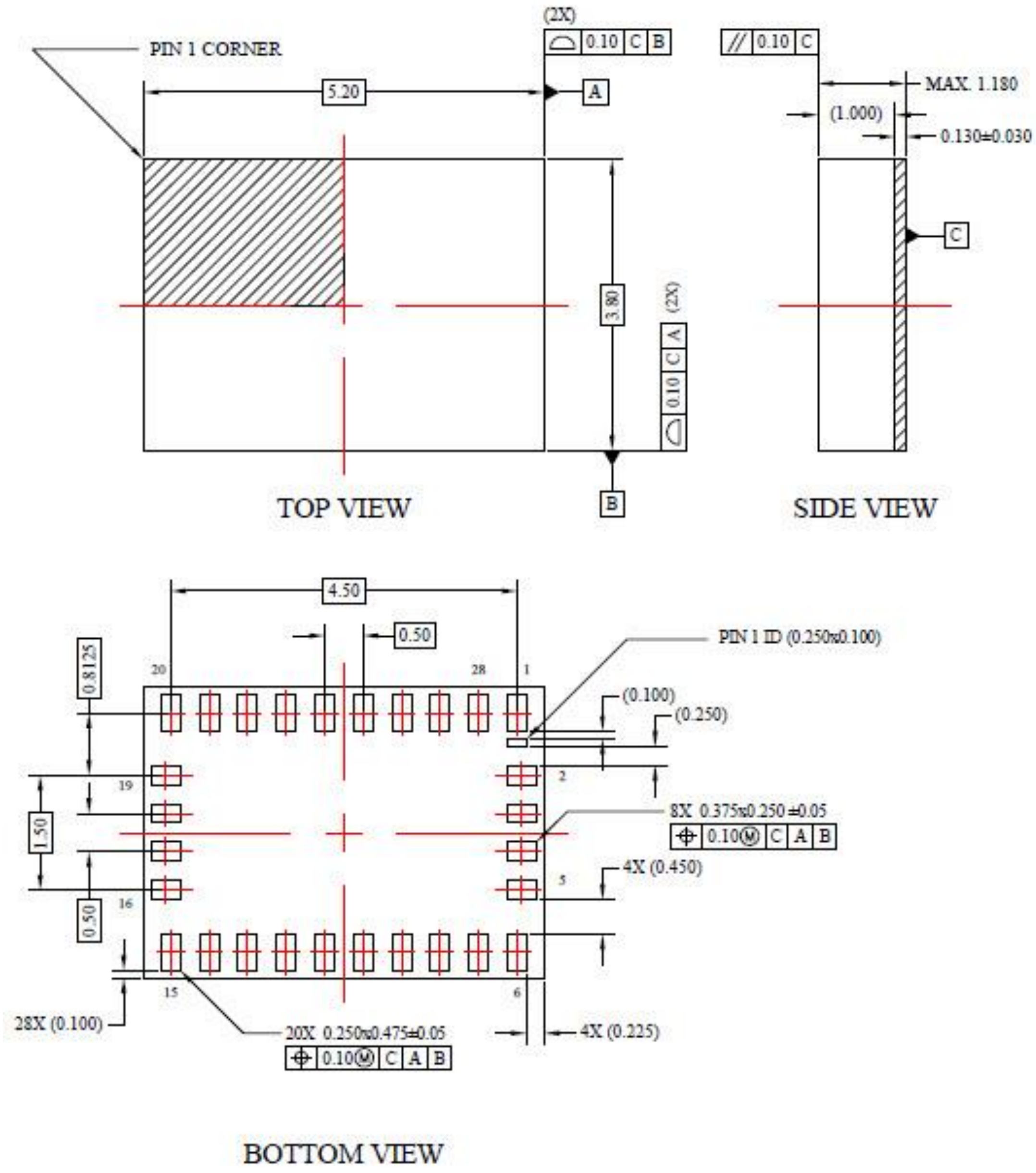
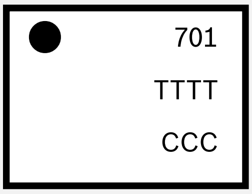


Figure 13: Outline dimensions

6.2 Marking

Table 6-1: Marking of mass production parts

Labeling	Name	Symbol	Remark
	Pin 1 identifier	•	---
	Product number	701	3 numeric digits, internal identification for product type
	Second Row	T	Internal use
	Third Row	C	Numerical counter

6.3 Soldering Guidelines

The moisture sensitivity level of the BNO055 sensors corresponds to JEDEC Level 1, see also

- IPC/JEDEC J-STD-020C "Joint Industry Standard: Moisture/Reflow Sensitivity Classification for non-hermetic Solid State Surface Mount Devices"
- IPC/JEDEC J-STD-033A "Joint Industry Standard: Handling, Packing, Shipping and Use of Moisture/Reflow Sensitive Surface Mount Devices"

The sensor fulfils the lead-free soldering requirements of the above-mentioned IPC/JEDEC standard, i.e. reflow soldering with a peak temperature up to 260°C.

6.4 Handling instructions

Micromechanical sensors are designed to sense acceleration with high accuracy even at low amplitudes and contain highly sensitive structures inside the sensor element. The MEMS sensor can tolerate mechanical shocks up to several thousand g's. However, these limits might be exceeded in conditions with extreme shock loads such as e.g. hammer blow on or next to the sensor, dropping of the sensor onto hard surfaces etc.

We recommend avoiding g-forces beyond the specified limits during transport, handling and mounting of the sensors in a defined and qualified installation process.

This device has built-in protections against high electrostatic discharges or electric fields (e.g. 2kV HBM); however, anti-static precautions should be taken as for any other CMOS component. Unless otherwise specified, proper operation can only occur when all terminal voltages are kept within the supply voltage range. Unused inputs must always be tied to a defined logic voltage level.

For more details on recommended handling, soldering and mounting please contact your local Bosch Sensortec sales representative and ask for the “Handling, soldering and mounting instructions” document.

6.5 Tape and reel specification

The BNO055 is shipped in a standard cardboard box. For details please refer to the 'Shipment packaging details' document.

6.6 Environmental safety

The BNO055 sensor meets the requirements of the EC restriction of hazardous substances (RoHS and RoHS2) directive, see also:

Directive 2002/95/EC of the European Parliament and of the Council of 27 January 2003
on the restriction of the use of certain hazardous substances in electrical and electronic equipment.

6.6.1 Halogen content

The BNO055 is halogen-free. For more details on the analysis results please contact your Bosch Sensortec representative.

6.6.2 Internal package structure

Within the scope of Bosch Sensortec's ambition to improve its products and secure the mass product supply, Bosch Sensortec qualifies additional sources (e.g. 2nd source) for the LGA package of the BNO055.

While Bosch Sensortec took care that all of the technical packages parameters are described above are 100% identical for all sources, there can be differences in the chemical content and the internal structural between the different package sources.

However, as secured by the extensive product qualification process of Bosch Sensortec, this has no impact to the usage or to the quality of the BMNO55 product.

7. Legal disclaimer

7.1 Engineering samples

Engineering Samples are marked with an asterisk (*) or (e) or (E). Samples may vary from the valid technical specifications of the product series contained in this data sheet. They are therefore not intended or fit for resale to third parties or for use in end products. Their sole purpose is internal client testing. The testing of an engineering sample may in no way replace the testing of a product series. Bosch Sensortec assumes no liability for the use of engineering samples. The Purchaser shall indemnify Bosch Sensortec from all claims arising from the use of engineering samples.

7.2 Product use

Bosch Sensortec products are developed for the consumer goods industry. They may only be used within the parameters of this product data sheet. They are not fit for use in life-sustaining or security sensitive systems. Security sensitive systems are those for which a malfunction is expected to lead to bodily harm or significant property damage. In addition, they are not fit for use in products which interact with motor vehicle systems.

The resale and/or use of products are at the purchaser's own risk and his own responsibility. The examination of fitness for the intended use is the sole responsibility of the Purchaser.

The purchaser shall indemnify Bosch Sensortec from all third party claims arising from any product use not covered by the parameters of this product data sheet or not approved by Bosch Sensortec and reimburse Bosch Sensortec for all costs in connection with such claims.

The purchaser must monitor the market for the purchased products, particularly with regard to product safety, and inform Bosch Sensortec without delay of all security relevant incidents.

7.3 Application examples and hints

With respect to any examples or hints given herein, any typical values stated herein and/or any information regarding the application of the device, Bosch Sensortec hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights or copyrights of any third party. The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics. They are provided for illustrative purposes only and no evaluation regarding infringement of intellectual property rights or copyrights or regarding functionality, performance or error has been made.

8. Document history and modifications

Rev. No	Chapter	Description of modification/changes	Date
0.1		Initial version	2013-09-02
0.2		Completely revised version (BMF055 added)	2013-10-15
0.9		Preliminary version with feature set of Firmware version 0.2.B.0	2014-04-25
1.0		Complete review	2014-07-11
1.1	3	Rearrangement of subsections in chapter 3 for better readability.	2014-11-05
	3.3	Table 3.1 is updated for better readability and all the operation modes are elaborated	
	3.11	Chapter on calibration included	
	3.7, 3.10	Update	
	4.2	The default value of the UNIT_SEL register is updated	
	4.6	I2C communication example figures are updated.	
	5.1, 5.2, 5.3, 5.4	Included table 5.1 Pin description. Connection diagram updated	
1.2	5	Updated pin description and connection diagram	2014-11-30
	6.1	Updated outline dimensions	
	6.2	Chapter removed and the respective information is updated in the Handling, soldering and mounting instructions application note.	
1.3	1.1	Supply current in low power mode is updated	2015-08-19
	1.2	Table 0-2 is updated for POR time description	
	3.5	Accelerometer restrictions updated in table 3.8	
	3.7	New section called 'Data Register Shadowing' is included to explain the concept shadowing	
	4.4.15	The SMNM bit field for Slow motion and no motion updated in the register description	
1.4	1.1	Representation of voltage in the table 0-1 is updated	2016-06-02
	2	The max value for ESD is updated	
	4.4.6	ACC_Sleep_Config register is updated for Accelerometer Sleep Phase Duration	
	5.1	Table 5-1: Pin description together with all the 3 connection diagrams are updated.	

Bosch Sensortec GmbH
Gerhard-Kindler-Strasse 9
72770 Reutlingen / Germany

contact@bosch-sensortec.com
www.bosch-sensortec.com

Modifications reserved | Printed in Germany
Specifications subject to change without notice
Document number: BST-BNO055-DS000-14
Revision_1.4_201606

Green

ZTX1049A

25V NPN MEDIUM POWER TRANSISTOR IN E-LINE

Features

- BV_{CEO} > 25V
- I_C = 4A High Continuous Collector Current
- I_{CM} = 20A Peak Pulse Current
- T_J up to 200°C for High Temperature Operation
- Low Saturation Voltage < 75mV @ 1A
- P_D = 1W Power dissipation
- **Lead-Free Finish; RoHS compliant (Note 1 & 2)**
- **Halogen and Antimony Free. "Green" Device (Note 3)**
- **Qualified to AEC-Q101 Standards for High Reliability**

Mechanical Data

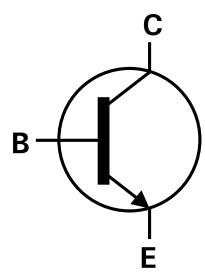
- Case: E-Line (TO-92 Compatible)
- Case Material: molded plastic, "Green" Molding Compound
- UL Flammability Classification Rating 94V-0
- Terminals: Finish – Matte Tin Plated Leads, Solderable per MIL-STD-202, Method 208 (e3)
- Weight: 0.159 grams (approximate)

Applications

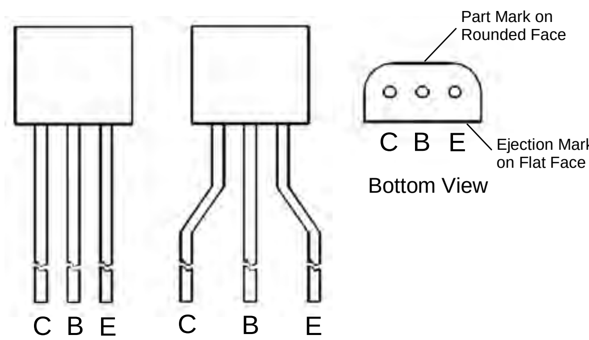
- LCD Backlight Converters
- Emergency Lighting
- DC-DC Converters

E-Line
(TO-92 Compatible)

Flat Face View



Device Symbol



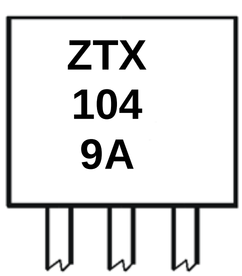
Pin-Out Configuration

Ordering Information (Note 4)

Part Number	Marking	Case	Leads	Quantity
ZTX1049ASTZ	ZTX1049A	E-Line	Joggled	2,000 taped per Ammo Box
ZTX1049A	ZTX1049A	E-Line	Straight	4,000 loose in a Box

Notes:
1. EU Directive 2002/95/EC (RoHS) & 2011/65/EU (RoHS 2) compliant. All applicable RoHS exemptions applied.
2. See http://www.diodes.com/quality/lead_free.html for more information about Diodes Incorporated's definitions of Halogen- and Antimony-free, "Green" and Lead-free.
3. Halogen- and Antimony-free "Green" products are defined as those which contain <900ppm bromine, <900ppm chlorine (<1500ppm total Br + Cl) and <1000ppm antimony compounds.
4. For packaging details, go to our website at http://www.diodes.com/products/packages.html.

Marking Information



Rounded Face View

ZTX1049A = Product type Marking Code

Maximum Ratings (@T_A = +25°C, unless otherwise specified.)

Characteristic	Symbol	Value	Unit
Collector-Base Voltage	V_{CBO}	80	V
Collector-Emitter Voltage	V_{CEO}	25	V
Emitter-Base Voltage	V_{EBO}	5	V
Continuous Collector Current	I_C	4	A
Peak Pulse Current	I_{CM}	20	A
Base Current	I_B	500	mA

Thermal Characteristics (@T_A = +25°C, unless otherwise specified.)

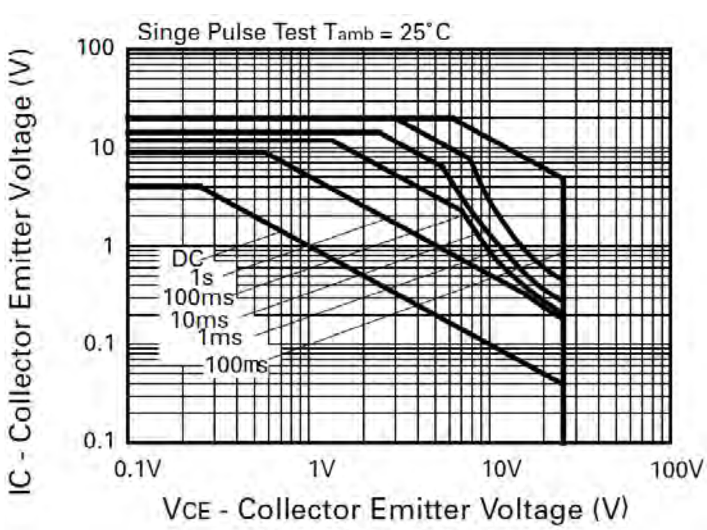
Characteristic	Symbol	Value	Unit
Power Dissipation (Note 5)	P_D	1.5	W
Power Dissipation (Note 6)	P_D	1	W
Thermal Resistance Junction to Ambient (Note 5)	$R_{\theta JA}$	116	°C/W
Thermal Resistance Junction to Ambient (Note 6)	$R_{\theta JA}$	175	°C/W
Thermal Resistance Junction to Lead (Note 7)	$R_{\theta JL}$	63.75	°C/W
Operating and Storage Temperature Range	T_J, T_{STG}	-55 to +200	°C

ESD Ratings (Note 8)

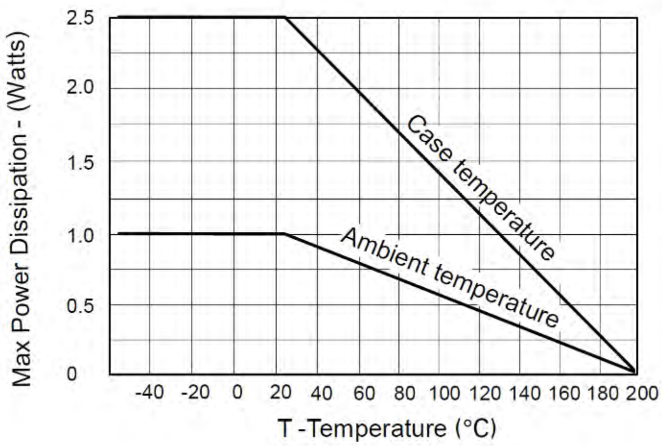
Characteristic	Symbol	Value	Unit	JEDEC Class
Electrostatic Discharge - Human Body Model	ESD HBM	≥ 4,000	V	3A
Electrostatic Discharge - Machine Model	ESD MM	≥ 400	V	C

Notes:
5. For a through-hole device mounted at the seating plane (2.5mm lead length) with the collector lead on 25mm X 25mm 1oz weight copper that is on a single-sided FR4 PCB; device is measured under still air conditions whilst operating in a steady-state.
6. Same as note (5), except the device is mounted on minimum recommended pad layout with 12mm lead length from the bottom of package to the board.
7. Thermal resistance from junction to solder-point at the seating plane (2.5mm from the bottom of package along the collector lead).
8. Refer to JEDEC specification JESD22-A114 and JESD22-A115.

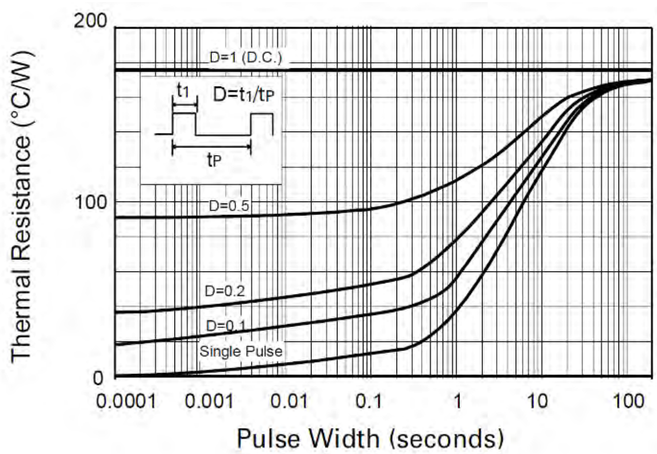
Thermal Characteristics and Derating Information



Safe Operating Area



Derating curve



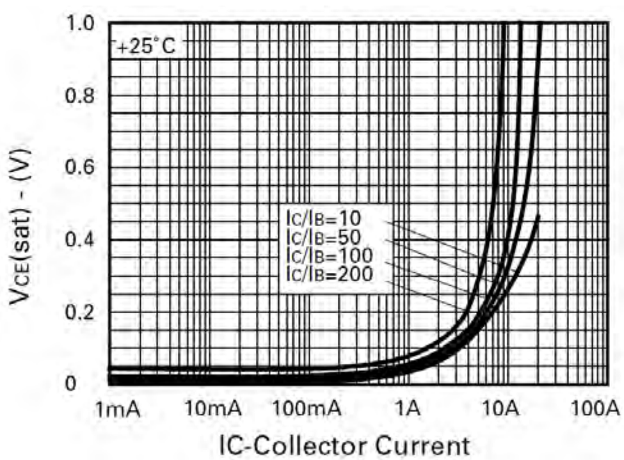
Maximum transient thermal impedance

Electrical Characteristics (@T_A = +25°C, unless otherwise specified.)

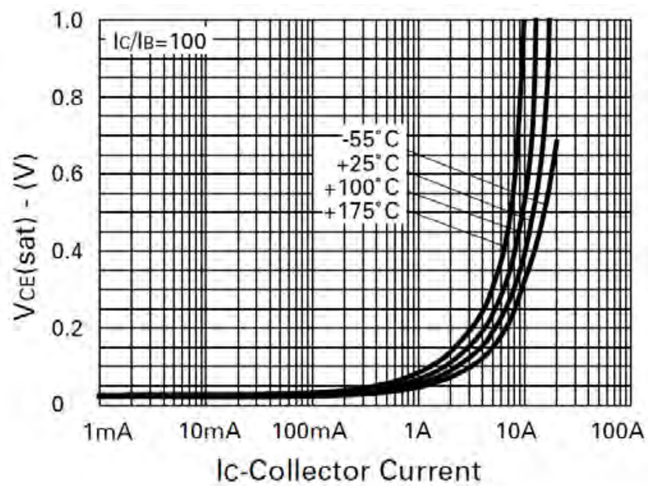
Characteristic	Symbol	Min	Typ	Max	Unit	Test Condition
Collector-Base Breakdown Voltage	BV_{CBO}	80	120	—	V	I_C = 100µA
Collector-Emitter Breakdown Voltage	BV_{CES}	80	120	—	V	I_C = 100µA
Collector-Emitter Breakdown Voltage (Note 9)	BV_{CEO}	25	30	—	V	I_C = 10mA
Collector-Emitter Breakdown Voltage	BV_{CEV}	80	120	—	V	I_C = 100µA, V_{EB} = 1V
Emitter-Base Breakdown Voltage	BV_{EBO}	5	8.75	—	V	I_E = 100µA
Collector Cut-off Current	I_{CBO}	—	0.3	10	nA	V_{CB} = 50V
Collector Emitter Cut-off Current	I_{CES}	—	0.3	10	nA	V_{CES} = 50V
Emitter Cut-off Current	I_{EBO}	—	0.3	10	nA	V_{EB} = 4V
Collector-Emitter Saturation Voltage (Note 9)	$V_{CE(sat)}$	—	30 60 125 155	45 80 180 220	mV	I_C = 500mA, I_B = 10mA I_C =1A, I_B = 10mA I_C =2A, I_B = 10mA I_C =4A, I_B = 50mA
Base-Emitter Saturation Voltage (Note 9)	$V_{BE(sat)}$	—	890	950	mV	I_C =4A, I_B = 50mA
Base-Emitter Turn-On Voltage (Note 9)	$V_{BE(on)}$	—	820	900	mV	I_C = 4A, V_{CE} = 2V
DC Current Gain (Note 9)	h_{FE}	250 300 300 200 35	430 450 450 350 70	— — 1200 — —	—	I_C = 10mA, V_{CE} = 2V I_C = 0.5A, V_{CE} = 2V I_C = 1A, V_{CE} = 2V I_C = 4A, V_{CE} = 2V I_C = 20A, V_{CE} = 2V
Current Gain-Bandwidth Product (Note 9)	f_T	—	180	—	MHz	V_{CE} = 10V, I_C = 50mA f = 50MHz
Output Capacitance (Note 9)	C_{obo}	—	45	60	pF	V_{CB} = 10V. f = 1MHz
Turn-On Times	t_{on}	—	125	—	ns	I_C = 4A, I_B = 40mA, V_{CC} = 10V
Turn-Off Times	t_{off}	—	380	—	ns	I_C = 4A, I_B = 40mA, V_{CC} = 10V

Notes: 9. Measured under pulsed conditions. Pulse width ≤ 300 µs. Duty cycle ≤ 2%

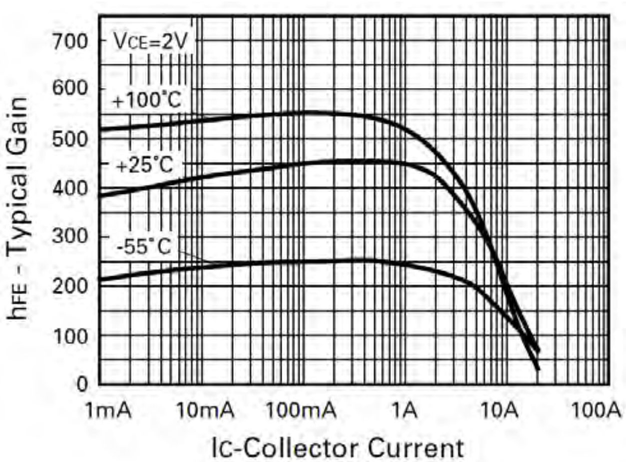
Typical Electrical Characteristics (@T_A = +25°C, unless otherwise specified.)



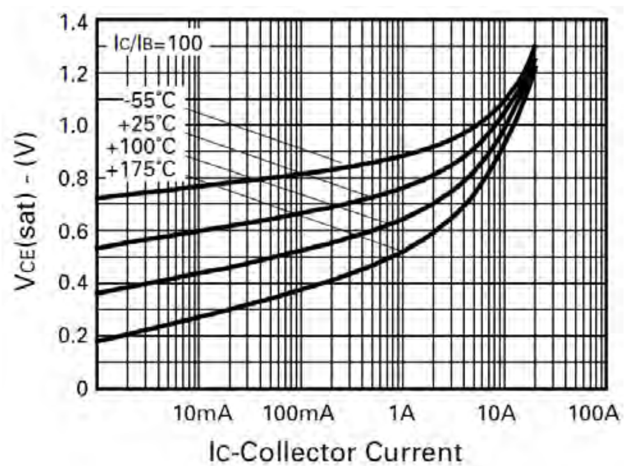
VCE(sat) v IC



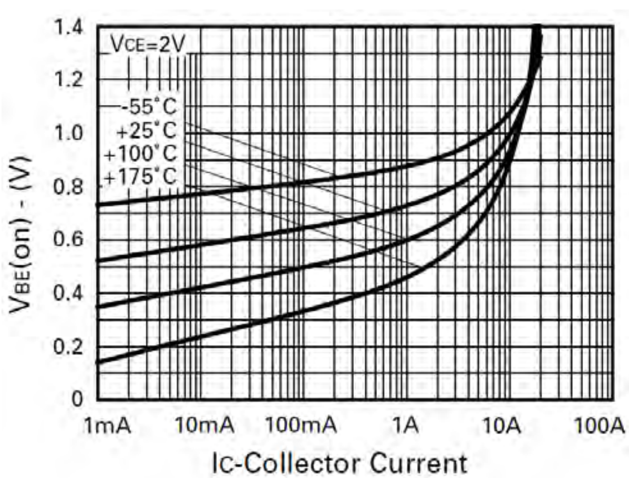
VCE(sat) v IC



hFE V IC



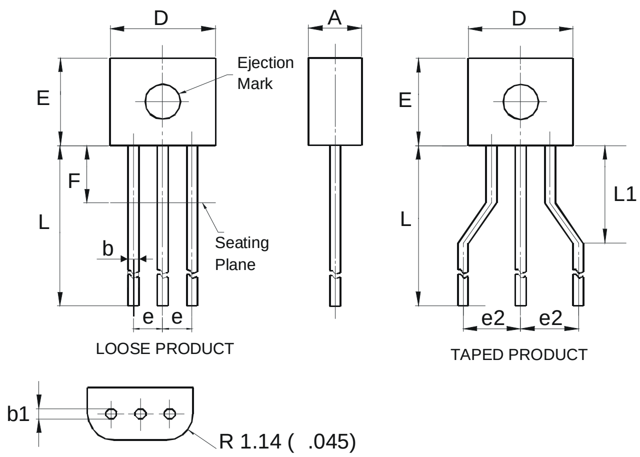
VBE(sat) v IC



VBE(on) v IC

Package Outline Dimensions

Please see AP02002 at http://www.diodes.com/datasheets/ap02002.pdf for latest version.



E-Line			
Dim	**Min**	**Max**	**Typ**
A	2.16	2.41	–
b	0.41	0.495	–
b1	0.41	0.495	–
D	4.37	4.77	–
E	3.61	4.01	–
e	–	–	1.27
e2	–	–	2.54
F	–	2.50	–
L	13.00	13.97	–
L1	2.50	3.50	–
All Dimensions in mm			