BRAIN COMPUTER INTERFACE

A PROJECT REPORT

SUBMITTED BY

ARSHAD NAZIR

ER : 7011812061612161

TO

Capital University

in partial fulfillment of the requirements for the award of Degree

OF

BACHELOR OF TECHNOLOGY

IN

ELECTRICAL AND ELECTRONICS ENGINEERING

CAPITAL UNIVERSITY

- Patna Rd, Near Subhash chowk, Chitragupt Nagar, Koderma

Jharkhand 825410

SEPTEMBER 2022

DECLARATION

I undersigned hereby declare that the project report entitled **"Brain Computer Interface."**, submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology from Capital University ,Jharkhand is a bonafide work done by me under the supervision. This submission represents our ideas in my own words and where ideas or works of others have been included, I have adequately and accurately cited and referred the original sources. I also declare that I have adhered to the ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Koderma	**Name and Signature**
Date: 20/07/2022	**ARSHAD NAZIR**

Department of Electrical & Electronics Engineering

Capital University ,Koderma

(Established in 2018 by the Government of Jharkhand)

CERTIFICATE

Certified that this report entitled **'Brain Computer Interface'** is the report of seminar presented by **Arshad Nazir, ER: 7011812061612161** during **2019-2020** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Electrical and Electronics Engineering of Capital University.

ACKNOWLEDGEMENT

I take this opportunity to express my deep sense of gratitude to the Almighty and sincere thanks to all who helped me to complete the project successfully. I express my sincere gratitude to our Principal, for providing me with all the necessary facilities and support for doing the project.

I am extremely grateful to **o**ur professor, Head of the Department, Department of Electrical and Electronics Engineering, for her constructive guidance, advises, constant support and technical guidance provided throughout the making of this seminar. Without her intellectual support and apt suggestions at the perfect time, this seminar work would not be possible.

I am also forever indebted to Associate Professor and Senior Advisor, Department of Electrical and Electronics Engineering, for her constant support and technical guidance provided throughout my undergraduate studies and especially during the project work.

I am greatly obliged to Assistant Professor and Seminar Guide, for her constructive criticism, guidance and suggestions.

I extend my immense gratitude to all Faculties and Technical Staffs in the Department of Electrical and Electronics Engineering, for their help and necessary facilities to complete this work. My humble gratitude and heartiest thanks also goes to my parents and friends, who have supported and helped me during the course of this work.

Place: Koderma **Arshad Nazir**

Date: 20/07/2022

ABSTRACT

My Aim is to uplift paralyzed people making them able to do anything with this technology. This device integrates Medical EEG scanner with Artificial Intelligence for a prediction of their behavior. Brain-Computer Interface (BCI) is a device that allows direct communication path between central nervous system and external devices without peripheral nerves dependency. Daily Activities are in fact limited for paraplegic and quadreplegic people. The system uses Supervised Reinforcement learning for the precise prediction.

Key Words: Brain Computer Interface, Artificial Intelligence, Machine Learning.

Contents

List of Figures

List of Tables

LIST OF ABBREVIATION

BCI Brain Computer Interface

CV Computer Vision

EEG Electroencephalography

HSV Hue Saturation Value

ML Machine Language

USB Universal Serial Bus

List of Tables

Chapter 1

INTRODUCTION

. Man machine interface has been one of the growing fields of research and development in recent years. Most of the effort has been dedicated to the design of user- friendly or ergonomic systems by means of innovative interfaces such as voice recognition, virtual reality. A direct brain-computer interface would add a new dimension to man-machine interaction.[1] A brain-computer interface, sometimes called a direct neural interface or a brain machine interface, is a direct communication pathway between a human or animal brain(or brain cell culture) and an external device. In one BCIs, computers either accept commands from the brain or send signals to it but not both. Two way BCIs will allow brains and external devices to exchange information in both directions but have yet to be successfully implanted in animals or humans. Brain-Computer interface is a staple of science fiction writing. In its earliest incarnations no mechanism was thought necessary, as the technology seemed so far fetched that no explanation was likely. As more became known about the brain however, the possibility has become more real and the science fiction more technically sophisticated. Recently, the cyberpunk movement has adopted the idea of 'jacking in', sliding 'biosoft' chips into slots implanted in the skull(Gibson, W.1984).Although such biosofts are still science fiction, there have been several recent steps toward interfacing the brain and computers. In this definition, the word brain means the brain or nervous system of an organic life form rather than the mind. Computer means any processing or computational device, from simple circuits to silicon chips (including hypothetical future technologies like quantum computing). Research on BCIs has been going on for more than 30 years but from the mid 1990's there has been dramatic increase working experimental implants[2][3].

1.1 Organization of Report

This device makes decisions when user focus on something that needs to happen. It mainly works in three stages Trigger signal Understanding Environmental variables Execution and learning process. These three steps will be explained briefly in the following chapters

Chapter 2

LITERATURE REVIEW

Bin et al., (2009) presented an online multi-channel SSVEP-oriented BCI systems along with Canonical Correlation Analysis (CCA) method to extract frequency information relayed to SSVEP. Significant parameters, window length, channel location, and harmonics number were examined through offline data, and consequences directed the online system architecture. Consequence proved that this novel BCI technique had optimal performance with 95.3% precision and information transferring rate of 58 ± 9.6 bit min1. This proposed technique's positive features were the channel selection and the parameter optimizations were not required since in addition with the harmonic frequency probable application, least user variation, and easy organisation.

Luo and Sullivan (2010) presented a user-friendly SSVEP oriented BCI system. Single-channel EEG was recorded with lower noise dry electrodes. Compared to the conservative gel oriented multi-sensor EEG system, dry sensors were appropriate, suitable and inexpensive. The author explained a novel Stimulus-Locked Inter-trace Correlation (SLIC) method for SSVEP classifications with EEG timelocked for stimuli onset. By SLIC system, the average accuracy was 75.8% with least error rate.

Sellers et al., (2010) developed a novel non-invasive EEG-based BCI system through persons with ALS and without assistive communication equipment. The initial results indicated that this novel BCI system is useful for home usage and considerably improved lives of people with strict motor disabilities.

Cecotti (2010) presented a BCI interaction system on neural function intended to provide a new output channel to deliberate the required brain control. This scheme had proposed a new self-paced BCI speller based on SSVEP detection. Speller initiated selection on the basis of decision tree was authenticated on 8 healthy participants lacking application knowledge. Accuracy and information transferring rate average were 92.25% and 37.62 bits per min respectively.

Mugler et al, (2010) developed an EEG BCI internet browser and used 10 healthy participants and 3 individual patients affected by advanced Amyotrophic Lateral Sclerosis (ALS). ALS 16 participants attained 73% average accuracy and consequent Information Transfer Rate (ITR) of 8.6 bits/min and volunteers without any BCI knowledge had over 90% accuracy and an ITR of 14.4 bits/min.

Zickleret al., (2011) proposed a P300 BCI incorporated with trade AT-software providing complete communication set and environmental control functions. Its usability was authenticated through potential end-user with several jobs in a regulated assessment process in an actual world atmosphere.

Kaufmann et al., (2011) presented Event-Related Potential (ERP)-based spelling equipment termed as P300-Speller, a basic BCI to improve impaired speech or motor function in patient‘s communication.

Rivet et al., (2009) proposed a novel unsupervised technique to improve evoked response through target stimuli in oddball concept. It proved to enhance evoked responses quality taking into consideration signal and noise, as conflicting to Principal Component Analysis (PCA) that measured the signal. This technique employed to improve P300 subspace prior to BCI classification.

Miglani and Gupta (2013) proposed a BCI called a brain machine intermediate for the people with energetic motor disability. Ang et al., (2011) proposed techniques for helping stroke survivor through BCI for communicating with the exterior world. It aimed to reinstate motor functions through signifying movement oriented brain elasticity.

Prasad et al., (2010) recommended the usage of rehabilitation protocol with Motor Imagery (MI) preparation with Physical Practice (PP) in directed rehabilitation among the stroke patients. The BCI neuro-feedback output had been computed based on MI task Classification Accuracy (CA) rate. Output collections assessments with the Action Research Arm Test (ARAT) and Grip Strength (GS) were used in assessing upper limb activity recovery.

Tan and Nijholt (2010) proved that cognitive neuroscience and brain imaging methods initiated the ability to converse directly with human brains. This is also probable through sensors that controls physical procedures in the brain related to a particular thoughts. The authors employed this method to build BCI, communication system that is brain‘s usual output pathway independent of peripheral nerve or muscle.

Anderson and Kirby (2003) proposed a 6-channel EEG data recording system using subjects for two mental jobs. Signals were converted through Karhunen-Loéve 17 transformation and classified through quadratic discriminate analysis and correctness was authenticated for 6 EEG channel subset.

Bamdadian et al., (2011) enhanced MIbased BCI performance through identifying the best free kernel parameters of SVM classifier. A real-coded GA determined SVM‘s free kernel parameters. This techniques performance had been assessed through publicly accessible BCI Competition IV dataset IIa for right/left hand motor imagery tasks.

Chapter 3

TRIGGER SIGNAL

Trigger signal is the first step in taking action by this BCI device. To obtain trigger signal the EEG Scanner fixed installed this device an electrode which is in contact with the for head .The electrons finds the voltage fluctuations resulting from ionic current within the neurons of the brain.
Four Types of waves are constantly recorded in device
Delta waves (.5 to 3 Hz) – Meditation
Theta waves (3 to 8 Hz) – Sleep
Alpha waves (8 to 12 Hz) – Idle
Beta waves (12 to 38 Hz) – Attention
Gamma waves (38 to 42 Hz) – Hyper Active / Anxiety

The sensor detects all the neuro facial spikes and is recorded therefore eye blinks can be recorded in this device, this is later used in device for more triggers .The device is connected to Raspberry Pi via bluetooth and data is streamed to the microprocessor. The Microprocessor filters the data extracting the attention and blink values of the user . If the Attention and Blink values in exceed a threshold value. The system triggers to give a new command

3.1 Electroencephalography (EEG)

Electroencephalography (EEG) is an electrophysiological monitoring method to record electrical activity of the brain. It is typically noninvasive, with the electrodes placed along the scalp, although invasive electrodes are sometimes used, as in electrocorticography.[4] EEG measures voltage fluctuations resulting from ionic current within the neurons of the brain. Clinically, EEG refers to the recording of the brain's spontaneous electrical activity over a period of time, as recorded from multiple electrodes placed on the scalp

3.1.1 Working

An EEG tracks and records brain wave patterns. Small metal discs with thin wires (electrodes) are placed on the scalp, and then send signals to a computer to record the results. Normal electrical activity in the brain makes a recognizable pattern. Through an EEG, doctors can look for abnormal patterns that indicate seizures and other problems.EEG isn't uncomfortable, and people do not feel any shocks on the scalp or elsewhere. Still, having electrodes pasted to the scalp can be a little stressful for kids.EEGs are very safe as it does not emit any radiation.[5]

Chapter 4

UNDERSTANDING ENVIRONMENTAL VARIABLES

When a trigger for a command is executed the microprocessor simultaneously does record all the available environmental variables such as
1) Intensity of Light in the area from the image in camera
2) Common Objects Detected in the image from the camera
3) Head position rotation and activity from gyroscope and accelerometer
4) Availible Wifi and its Signal Strength
5) Time and Date of command
6) Previous Commands

With the help of Scikit-learn and Tensorflow Machine Learning libraries , A perfect match of output from the Log table is taken as given as an integer output value[6].

4.1 Image Processing

4.1.1 Hardware

Figure 4.1: Pi Camera

The v2 Camera Module has a Sony IMX219 8-megapixel sensor.The Camera Module can be used to take high-definition video, as well as stills photographs. The camera works with all models of Raspberry Pi 1, 2, 3 and 4. It can be accessed through the MMAL and V4L APIs, and there are numerous third-party libraries built for it, including the Picamera Python library.

4.1.2 Software

OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel.OpenCV is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though extensive older C interface. There are bindings in Python, Java and MATLAB/OCTAVE[7].

4.1.2.1 Intensity of Light in the area

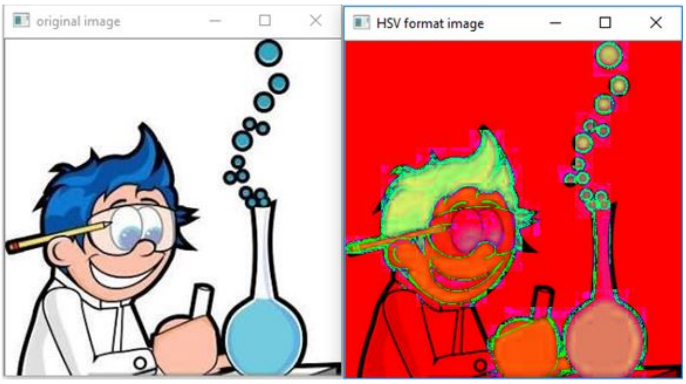


Figure 4.2: HSV Result

With the help of OpenCV libraries and PiCamera Python Libraries . The RGB Image taken in the camera is converted to HSV in which the V value is the intensity of light in the for each pixel.The total Average Mean of the Intensity values is calculated to a single value

The intensity of the light is an environmental varible wihich can determine whether the person is outside/inside home or in a dark/luminant room.

4.1.2.2 Common Objects Detected in the image

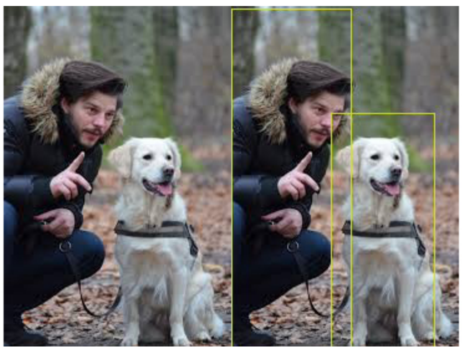


Figure 4.3: Common Objects Detected

With the help of OpenCv and Tensorflow data, Common household and external object shaped are detected in order. The process is lengthy and takes time to produce result

Common Objects detected can determine whether there is any humans, household objects ,office objects , outdoor objects , objects that are specified to certain locations etc[8]

4.2 Rotational Activity of User's Head

4.2.1 Hardware

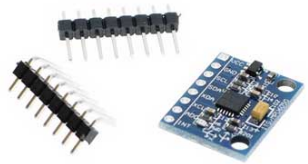


Figure 4.4: MPU6050

The MPU6050 IMU has both 3-Axis accelerometer and 3-Axis gyroscope integrated on a single chip.

The gyroscope measures rotational velocity or rate of change of the angular position over time, along the X, Y and Z axis. It uses MEMS technology and the Coriolis Effect for measuring

4.2.2 Software

4.2.2.1 Head Rotation and Change in Head Rotation(Velocity)

```
print (" Reading Data of Gyroscope and Accelerometer")

while True:

    #Read Accelerometer raw value
    acc_x = read_raw_data(ACCEL_XOUT_H)
    acc_y = read_raw_data(ACCEL_YOUT_H)
    acc_z = read_raw_data(ACCEL_ZOUT_H)

    #Read Gyroscope raw value
    gyro_x = read_raw_data(GYRO_XOUT_H)
    gyro_y = read_raw_data(GYRO_YOUT_H)
    gyro_z = read_raw_data(GYRO_ZOUT_H)

    #Full scale range +/- 250 degree/C as per sensitivity scale factor
    Ax = acc_x/16384.0
    Ay = acc_y/16384.0
    Az = acc_z/16384.0

    Gx = gyro_x/131.0
    Gy = gyro_y/131.0
    Gz = gyro_z/131.0


    print ("Gx=%.2f" %Gx, u'\u00b0'+ "/s", "\tGy=%.2f" %Gy, u'\u00b0'+ "/s", "\tGz=%.2f"
    %Gz, u'\u00b0'+ "/s", "\tAx=%.2f g" %Ax, "\tAy=%.2f g" %Ay, "\tAz=%.2f g" %Az)
    sleep(1)
```

Figure 4.5: Gyroscope and Accelerator Code

On Serial Communication with MPU6050 IMU it returns an irrational value for the angle of X, Y ,Z and Acceleration of X ,Y ,Z.

The Gyroscope and Accelerometer values can determine whether user's activity whether the user is sleeping, idle or travelling

4.3 Wireless Fidelity

```
proc = subprocess.Popen(["iwlist", interface, "scan"],stdout=subprocess.PIPE, universal_newlines=True)
out, err = proc.communicate()

for line in out.split("\n"):
    cell_line = match(line,"Cell ")
    if cell_line != None:
        cells.append([])
        line = cell_line[-27:]
    cells[-1].append(line.rstrip())

cells=cells[1:]

for cell in cells:
    parsed_cells.append(parse_cell(cell))

sort_cells(parsed_cells)

print_cells(parsed_cells)
```

Figure 4.6: Wi-Fi code

Raspberry Pi had a built in system in order to connect to the Wifi network. We use that system for finding the nearby Networks ans their signal strength

The details of Wireless Fidelity can determine whether the user is traveling or his rough location can be estimated, Similar to Google Maps.

4.4 Time of Command

```
now = datetime.now()
time = now.strftime("%H")
day = now.strftime("%D")
```

Figure 4.7: Time code

Raspberry Pi has a built in CMOS clock for returning the Time,Date,Geographical Season at the time of command

This details can extract the Sort the Activity, Weekend Conditions , Seasonal Conditions etc.

4.5 Study of Previous Command

The last Command is retrived from the Log table in order to find the recent activity and action the user has been going through

This details has a major importance in estimating the activity of the user.

Chapter 5

EXECUTION AND LEARNING

With Scikit-learn and Tensorflow Machine Learning libraries , A perfect match of output from the Log table is taken as given as an integer output value. Execution can be done in many ways. Since a raspberry Pi processor has an Wireless Fidelity , Bluetooth RFCOMM and USB Communication.[9]

The system waits 3 seconds for a reject trigger. If the user has given a reject trigger .The current action will be reversed. After 3 seconds if the current user has not rejected the execution .The environmental variables and the output will be saved into the Log which is further used for learning and predicting.

5.1 Scikit and Tensorflow Libraries

Scikit-learn (formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language.[10] It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks.[11] It is used for both research and production at Google.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache License 2.0 on November 9, 2015.

5.2 Training table

Inty	COD	GX	GY	GZ	AX	AY	AZ	Wifi	SS	Time	Day	LC	Out
23	2	33	22	11	25	42	71	12	3	6	7	3	4
63	3	42	63	21	16	22	48	10	2	12	5	3	3
52	1	59	31	53	17	33	60	12	1	5	1	3	6

Table 5.1: Example Training Log Table

Inty - Intensity

COD - Common Objects Detected

GX - GyroScope X

GY - GyroScope X

GZ - GyroScope X

AX - Accelero X

AY - Acceleration Y

AZ - Acceleration Z

SS - Signal Strength

LC - Last Command

Out - Output

This is the example data saved inside the trainer log.The outputs of every data is converted to a number for simplicity and data management.

Chapter 6

CONCLUSION

The work is set to open new opportunities for paraplegic and quadreplegic patients with the help of Artificial Inteligence.[12]This system can bring differently abled people bring back to their respective privileges. The system empowers users to remotely monitor and control devices. BCI EEG CV HSV USB ML

In the field of BCI system designing the main aim is to revolutionizing the human computer interaction future with exponentially increasing the outcomes in many number of fields. In this literature review the discussion of novel system like ECoG and EEG based BCI and the methodology with some limitation is done, also the methodology is not restricted to the flow of modules of only one BCI application it is explore with the nowadays system criteria. There are some system which discussed are based on the evoked potential such as SSVEP based and VCP based which has some difficulties in user training and lack of gaze control issues.[13] As compare to other module the signal acquisition and signal processing are relatively simple to implement with the help of classifier which are widely used. There are some desires from the user and BCI system such as long term training of the user for estimated EEG signal, system should have the improve signal processing unit to handle low quality signal and development of accurate EEG signal block such that it can allow the simulation based analysis.

REFERENCES

[1] Y Mastanamma . ”Electric Vehicle Mathematical Modelling and Simulation Using MATLAB-Simulink.” IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE) 12.4 (2017): 47-53.

[2] Hybrid Brain Computer Interface Based on EEG and NIRS moralities by Min Ho Lee and Jan ,IEEE Paper 2018.

[3] Stabilisation of Brain Computer Interface with low dimensional spaces of Neural Activity by Alan D degenhart

[4] Machine translation of cortical activity to text with an encoder–decoder framework by Joseph G Makin

[5] Shangkai Gao, Yijun Wang, Xiaorong Gao, and Bo Hong, “Visual and auditory brain–computer interfaces,”IEEE Transactions On Biomedical Engineering, vol. 61, pp. 1436-1447, May 2014.

[6] Paul S. Hammon and Virginia R. de Sa, “Preprocessing and metaclassification forbrain-computer interfaces,” IEEE Transactions On Biomedical Engineering, vol. 54, pp.518-525, March 2007.

[7] G. Schalk, Dennis J. McFarland, T. Hinterberger, N. Birbaumer, and Jonathan R.Wolpaw, "BCI2000: A general-purpose brain-computer interface (BCI) system," IEEE Transactions On Biomedical Engineering, vol. 51, pp. 1034-1043, June 2004.

[8] Melody M. Moore, "Real-world applications for brain– computerinterface technology," IEEE Transactions On Neural Systems And Rehabilitation Engineering, vol. 11, pp. 162-165, June 2003.

[9] Gerwin Schalk, and Eric C. Leuthardt, "Brain-computer interfaces using electrocorticographic signals," IEEE Reviews In Biomedical Engineering, vol. 4,pp. 140-154, 2011.

[10] Jinyi Long, Yuanqing Li, Hongtao Wang, Tianyou Yu, Jiahui Pan, and Feng Li, "A hybrid brain computer interface to controlthe direction and speed of a simulated or real wheelchair," IEEE Transactions On Neural Systems And Rehabilitation Engineering, vol. 20, pp. 720-729, September 2012.

[11] David Marshall, Damien Coyle, Shane Wilson, and Michael Callaghan, "Games, gameplay, and BCI: the state of the art,"IEEE Transactions On Computational Intelligence And Ai In Games, vol. 5, pp. 88-99, June 2013.

[12] C. Guger, H. Ramoser, and G. Pfurtscheller, "Real-time eeg analysis with subject-specificspatial patterns for a brain–computer interface (BCI)," IEEE Transactions On Rehabilitation Engineering, vol. 8, pp. 447-456, December 2000.

[13] Cecilia L. Maeder, Claudia Sannelli, Stefan Haufe, and Benjamin Blankertz, "Pre-stimulus sensorimotor rhythms influencebrain–computer interface classification performance," IEEE Transactions On Neural Systems And Rehabilitation Engineering, vol. 20, no. 5, September 2012.

APPENDIX

Program

```
#Made by Arshad on 3/3/2020
import mindwave, time                                   #Library for EEG Scanner
from os import system
import RPi.GPIO as GPIO
import smbus             #import SMBus module of I2C
from time import sleep
import sys
import subprocess
from picamera.array import PiRGBArray                   #Library for PiCAM
from picamera import PiCamera
import cv2                                              #Library for OpenCV
from sklearn import tree                                #Library for Machine Learning(Plotting Tree)
from sklearn.externals.six import StringIO
from IPython.display import Image                       #Library for displaying Image
from sklearn.tree import export_graphviz
import pydotplus                                        #Library for Graphical Output
from datetime import datetime                           #Library for Accessing Time and Date



PWR_MGMT_1   = 0x6B                                     #Registar Values of Acceleromeeter GyroScope
SMPLRT_DIV   = 0x19
CONFIG       = 0x1A
GYRO_CONFIG  = 0x1B
INT_ENABLE   = 0x38
ACCEL_XOUT_H = 0x3B
ACCEL_YOUT_H = 0x3D
ACCEL_ZOUT_H = 0x3F
GYRO_XOUT_H  = 0x43
GYRO_YOUT_H  = 0x45
GYRO_ZOUT_H  = 0x47
interface = "wlan0"



acc_x = 0                                               #Initialising all the Current Situation Variables
acc_y = 0
```

```
acc_z = 0
gyro_x = 0
gyro_y = 0
gyro_z = 0
Ax = 0
Ay = 0
Az = 0
Gx = 0
Gy = 0
Gz = 0
NearestWifi=""
CommonObjectsLabel = []
LightIntensity = 0

dot_data = StringIO()


                                                            #Setting a Sample Data, Sample Output for User

situation = [[20,1,0,0,3,9,0]]
output = [0]
opnames=["Room Light On", "Room Light Off" ,"AC On","AC Off", "Music On","Music Off"]




                                                            # initialise PWM pin
GPIO.setmode(GPIO.BOARD)                                    # BOARD numbering

LED = 11                                                    # BOARD Pin 11 == BCM Pin 17
GPIO.setup(LED, GPIO.OUT)

pwm_LED = GPIO.PWM(LED, 100)
pwm_LED.start(0)

                                                            # setup headset
headset = mindwave.Headset('/dev/ttyUSB0', '1E5F')
```

```
time.sleep(2)

headset.connect()
print "Connecting..."

while headset.status != 'connected':
    time.sleep(0.5)
    if headset.status == 'standby':
        headset.connect()
        print "Retrying connect..."
print "Connected."

bus = smbus.SMBus(1)
Device_Address = 0x68                                        # MPU6050 device address

MPU_Init()


while True:
    time.sleep(.5)
    print "Attention: %s, Meditation: %s" % (headset.attention, headset.meditation)
    if headset.attention > 30:
        pwm_LED.ChangeDutyCycle(min(100, headset.attention-30))

                                                            #Read Accelerometer raw value
        acc_x = read_raw_data(ACCEL_XOUT_H)
        acc_y = read_raw_data(ACCEL_YOUT_H)
        acc_z = read_raw_data(ACCEL_ZOUT_H)

                                                            #Read Gyroscope raw value
        gyro_x = read_raw_data(GYRO_XOUT_H)
        gyro_y = read_raw_data(GYRO_YOUT_H)
        gyro_z = read_raw_data(GYRO_ZOUT_H)

                                                            #Full scale range +/- 250 degree/C as per sensitivity scale
        Ax = acc_x/16384.0
        Ay = acc_y/16384.0
```

```python
        Az = acc_z/16384.0

        Gx = gyro_x/131.0
        Gy = gyro_y/131.0
        Gz = gyro_z/131.0



        cells=[[]]
        parsed_cells=[]

        proc = subprocess.Popen(["iwlist", interface, "scan"],stdout=subprocess.PIPE, universal_newlines=True)
        out, err = proc.communicate()

        for line in out.split("\n"):
            cell_line = match(line,"Cell ")
            if cell_line != None:
                cells.append([])
                line = cell_line[-27:]
            cells[-1].append(line.rstrip())

        cells=cells[1:]

        for cell in cells:
            parsed_cells.append(parse_cell(cell))

        sort_cells(parsed_cells)

        print_cells(parsed_cells)


        camera = PiCamera()
        rawCapture = PiRGBArray(camera)
        # allow the camera to warmup
        time.sleep(0.1)
        # grab an image from the camera
        camera.capture(rawCapture, format="bgr")
```

```
        image = rawCapture.array



        bbox, label, conf = cv.detect_common_objects(image)
        CommonObjectsLabel = label


        int brightness = 0
        cv::Mat mat = get_image_from_device()

        cv::Mat hsv
        cv::cvtColor(mat, hsv, CV_BGR2HSV)
        const auto result = cv::mean(hsv)

        LightIntensity = result[2];

        now = datetime.now()
        time = now.strftime("%H")
        day = now.strftime("%D")

        clf  = tree.DecisionTreeClassifier()
        clf.fit(situation,output)



        result = (clf.predict([[LightIntensity,CommonObjectsLabel,acc_x,acc_y,acc_z,gyro_x,gyro_y,gyro_z,Ax,Ay,Az,Gx,
        NearestWifi,time,day]]))

        print(opnames[result[0]])
        print("Accept - 1, Reject - 0")
        res = input()
        if res == '1':
            print("Accept")
            situation.append([LightIntensity,CommonObjectsLabel,acc_x,acc_y,acc_z,gyro_x,gyro_y,gyro_z,Ax,Ay,Az,Gx,Gy
            NearestWifi,time,day])

```

```
                output.append(result[0])
        if res == '0':
            print("Reject")
            for x in range(len(opnames)):
                if x != result[0]:
                    print(x)
                    situation.append([inty,cod,hr,wifi,ss,time,lc])
                    output.append(x)
        input()




def MPU_Init():
    #write to sample rate register
    bus.write_byte_data(Device_Address, SMPLRT_DIV, 7)

    #Write to power management register
    bus.write_byte_data(Device_Address, PWR_MGMT_1, 1)

    #Write to Configuration register
    bus.write_byte_data(Device_Address, CONFIG, 0)

    #Write to Gyro configuration register
    bus.write_byte_data(Device_Address, GYRO_CONFIG, 24)

    #Write to interrupt enable register
    bus.write_byte_data(Device_Address, INT_ENABLE, 1)

def read_raw_data(addr):
    #Accelero and Gyro value are 16-bit
        high = bus.read_byte_data(Device_Address, addr)
        low = bus.read_byte_data(Device_Address, addr+1)
```

```python

        #concatenate higher and lower value
        value = ((high << 8) | low)

        #to get signed value from mpu6050
        if(value > 32768):
                value = value - 65536
        return value



def get_name(cell):
    return matching_line(cell,"ESSID:")[1:-1]

def get_quality(cell):
    quality = matching_line(cell,"Quality=").split()[0].split('/')
    return str(int(round(float(quality[0]) / float(quality[1]) * 100))).rjust(3) + " %"


def get_signal_level(cell):

    return matching_line(cell,"Quality=").split("Signal level=")[1]








rules={"Name":get_name,
       "Quality":get_quality,
       "Signal":get_signal_level
       }


```

```
def sort_cells(cells):
    sortby = "Quality"
    reverse = True
    cells.sort(None, lambda el:el[sortby], reverse)



columns=["Name","Quality","Signal"]





def matching_line(lines, keyword):
    """Returns the first matching line in a list of lines. See match()"""
    for line in lines:
        matching=match(line,keyword)
        if matching!=None:
            return matching
    return None

def match(line,keyword):
    """If the first part of line (modulo blanks) matches keyword,
    returns the end of that line. Otherwise returns None"""
    line=line.lstrip()
    length=len(keyword)
    if line[:length] == keyword:
        return line[length:]
    else:
        return None

def parse_cell(cell):
    parsed_cell={}
    for key in rules:
        rule=rules[key]
        parsed_cell.update({key:rule(cell)})
    return parsed_cell
```

```
def print_table(table):
    widths=map(max,map(lambda l:map(len,l),zip(*table))) #functional magic

    justified_table = []
    for line in table:
        justified_line=[]
        for i,el in enumerate(line):
            justified_line.append(el.ljust(widths[i]+2))
        justified_table.append(justified_line)

    for line in justified_table:
        for el in line:
            print el,
        NearestWifi = el
        print

def print_cells(cells):
    table=[columns]
    for cell in cells:
        cell_properties=[]
        for column in columns:
            cell_properties.append(cell[column])
        table.append(cell_properties)
    print_table(table)
```

DataSheet

6.0.1 Raspberry Pi

3 Block Diagram

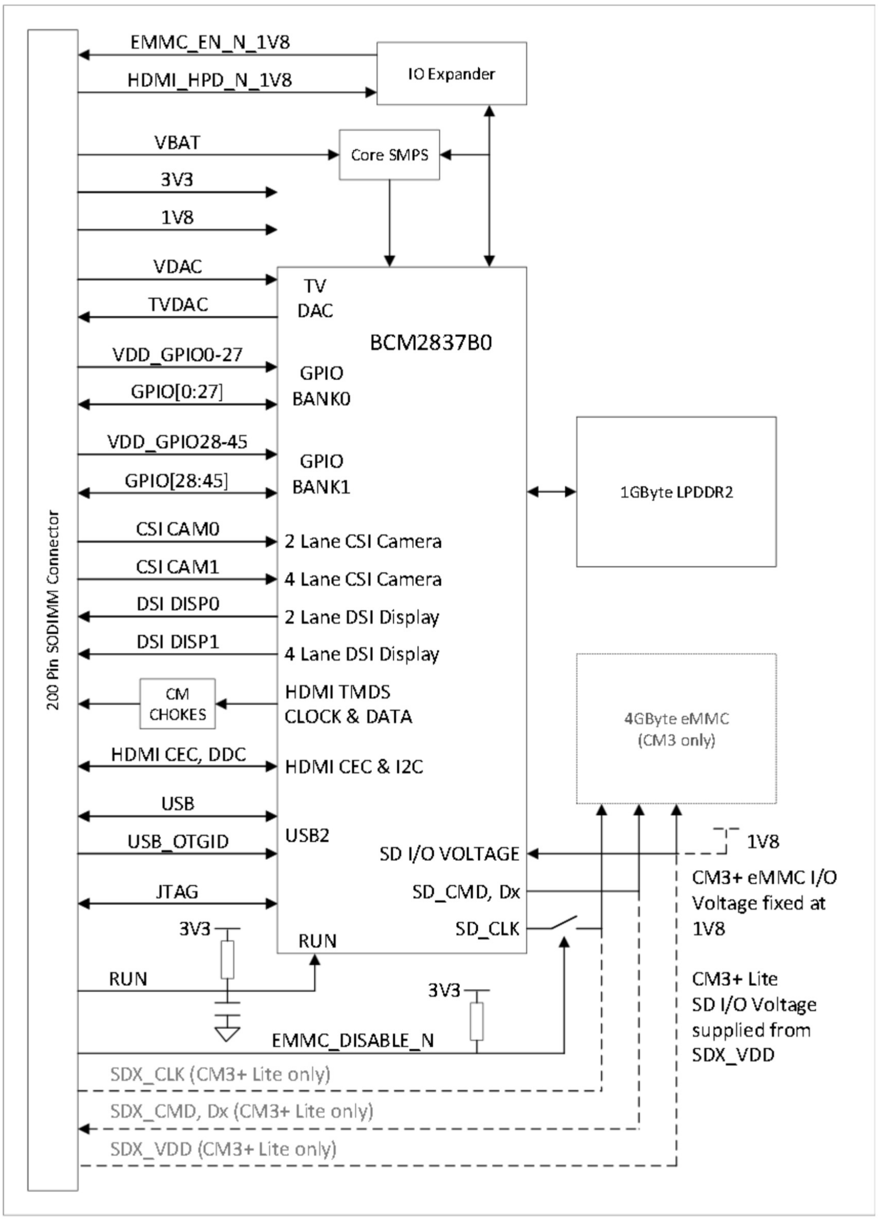


Figure 1: CM3+ Block Diagram

4 Mechanical Specification

The CM3+ modules conform to JEDEC MO-224 mechanical specification for 200 pin DDR2 (1.8V) SODIMM modules and therefore should work with the many DDR2 SODIMM sockets available on the market. **(Please note that the pinout of the Compute Module is not the same as a DDR2 SODIMM module; they are not electrically compatible.)**

The SODIMM form factor was chosen as a way to provide the 200 pin connections using a standard, readily available and low cost connector compatible with low cost PCB manufacture.

The maximum component height on the underside of the Compute Module is 1.2mm.

The maximum component height on the top side of the Compute Module is 2.5mm.

The Compute Module PCB thickness is 1.0mm +/- 0.1mm.

Note that the location and arrangement of components on the Compute Module may change slightly over time due to revisions for cost and manufacturing considerations; however, maximum component heights and PCB thickness will be kept as specified.

Figure 2 gives the CM3+ mechanical dimensions.

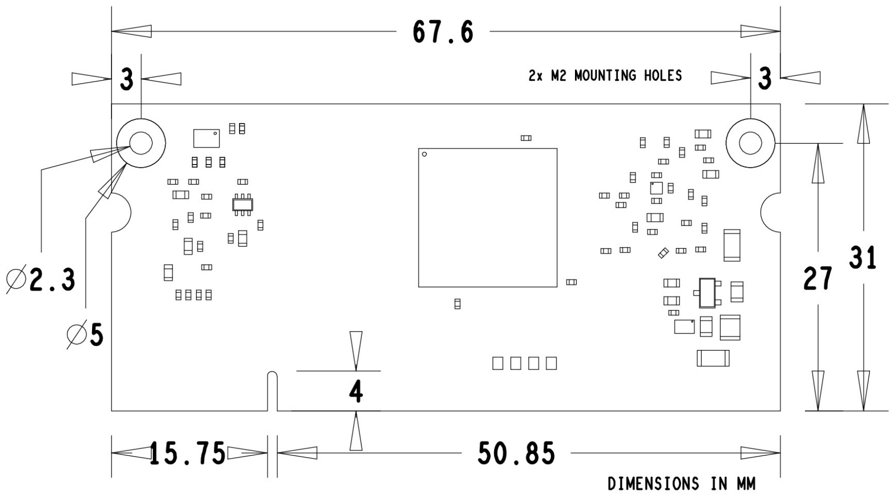


Figure 2: CM3+ Mechanical Dimensions

5 Pin Assignments

CM3+	CM3+ Lite	PIN		PIN	CM3+	CM3+ Lite
GND		1		2	EMMC_DISABLE_N	
GPIO0		3		4	NC	SDX_VDD
GPIO1		5		6	NC	SDX_VDD
GND		7		8	GND	
GPIO2		9		10	NC	SDX_CLK
GPIO3		11		12	NC	SDX_CMD
GND		13		14	GND	
GPIO4		15		16	NC	SDX_D0
GPIO5		17		18	NC	SDX_D1
GND		19		20	GND	
GPIO6		21		22	NC	SDX_D2
GPIO7		23		24	NC	SDX_D3
GND		25		26	GND	
GPIO8		27		28	GPIO28	
GPIO9		29		30	GPIO29	
GND		31		32	GND	
GPIO10		33		34	GPIO30	
GPIO11		35		36	GPIO31	
GND		37		38	GND	
GPIO0-27_VDD		39		40	GPIO0-27_VDD	
		KEY				
GPIO28-45_VDD		41		42	GPIO28-45_VDD	
GND		43		44	GND	
GPIO12		45		46	GPIO32	
GPIO13		47		48	GPIO33	
GND		49		50	GND	
GPIO14		51		52	GPIO34	
GPIO15		53		54	GPIO35	
GND		55		56	GND	
GPIO16		57		58	GPIO36	
GPIO17		59		60	GPIO37	
GND		61		62	GND	
GPIO18		63		64	GPIO38	
GPIO19		65		66	GPIO39	
GND		67		68	GND	
GPIO20		69		70	GPIO40	
GPIO21		71		72	GPIO41	
GND		73		74	GND	
GPIO22		75		76	GPIO42	
GPIO23		77		78	GPIO43	
GND		79		80	GND	
GPIO24		81		82	GPIO44	
GPIO25		83		84	GPIO45	
GND		85		86	GND	
GPIO26		87		88	HDMI_HPD_N_1V8	
GPIO27		89		90	EMMC_EN_N_1V8	
GND		91		92	GND	
DSI0_DN1		93		94	DSI1_DP0	
DSI0_DP1		95		96	DSI1_DN0	
GND		97		98	GND	
DSI0_DN0		99		100	DSI1_CP	
DSI0_DP0		101		102	DSI1_CN	
GND		103		104	GND	
DSI0_CN		105		106	DSI1_DP3	
DSI0_CP		107		108	DSI1_DN3	
GND		109		110	GND	
HDMI_CLK_N		111		112	DSI1_DP2	
HDMI_CLK_P		113		114	DSI1_DN2	
GND		115		116	GND	
HDMI_D0_N		117		118	DSI1_DP1	
HDMI_D0_P		119		120	DSI1_DN1	
GND		121		122	GND	
HDMI_D1_N		123		124	NC	
HDMI_D1_P		125		126	NC	
GND		127		128	NC	
HDMI_D2_N		129		130	NC	
HDMI_D2_P		131		132	NC	
GND		133		134	GND	
CAM1_DP3		135		136	CAM0_DP0	
CAM1_DN3		137		138	CAM0_DN0	
GND		139		140	GND	
CAM1_DP2		141		142	CAM0_CP	
CAM1_DN2		143		144	CAM0_CN	
GND		145		146	GND	
CAM1_CP		147		148	CAM0_DP1	
CAM1_CN		149		150	CAM0_DN1	
GND		151		152	GND	
CAM1_DP1		153		154	NC	
CAM1_DN1		155		156	NC	
GND		157		158	NC	
CAM1_DP0		159		160	NC	
CAM1_DN0		161		162	NC	
GND		163		164	GND	
USB_DP		165		166	TVDAC	
USB_DM		167		168	USB_OTGID	
GND		169		170	GND	
HDMI_CEC		171		172	VC_TRST_N	
HDMI_SDA		173		174	VC_TDI	
HDMI_SCL		175		176	VC_TMS	
RUN		177		178	VC_TDO	
VDD_CORE (DO NOT CONNECT		179		180	VC_TCK	
GND		181		182	GND	
1V8		183		184	1V8	
1V8		185		186	1V8	
GND		187		188	GND	
VDAC		189		190	VDAC	
3V3		191		192	3V3	
3V3		193		194	3V3	
GND		195		196	GND	
VBAT		197		198	VBAT	
VBAT		199		200	VBAT	

Table 2: Compute Module 3+ SODIMM Connector Pinout

Table 2 gives the Compute Module 3+ pinout and Table 3 gives the pin functions.

Pin Name	DIR	Voltage Ref	PDN[a] State	If Unused	Description/Notes
RUN and Boot Control (see text for usage guide)					
RUN	I	3V3[b]	Pull High	Leave open	Has internal 10k pull up
EMMC_DISABLE_N	I	3V3[b]	Pull High	Leave open	Has internal 10k pull up
EMMC_EN_N_1V8	O	1V8	Pull High	Leave open	Has internal 2k2 pull up
GPIO					
GPIO[27:0]	I/O	GPIO0-27_VDD	Pull or Hi-Z[c]	Leave open	GPIO Bank 0
GPIO[45:28]	I/O	GPIO28-45_VDD	Pull or Hi-Z[c]	Leave open	GPIO Bank 1
Primary SD Interface[d,e]					
SDX_CLK	O	SDX_VDD	Pull High	Leave open	Primary SD interface CLK
SDX_CMD	I/O	SDX_VDD	Pull High	Leave open	Primary SD interface CMD
SDX_Dx	I/O	SDX_VDD	Pull High	Leave open	Primary SD interface DATA
USB Interface					
USB_Dx	I/O	-	Z	Leave open	Serial interface
USB_OTGID	I	3V3		Tie to GND	OTG pin detect
HDMI Interface					
HDMI_SCL	I/O	3V3[b]	Z[f]	Leave open	DDC Clock (5.5V tolerant)
HDMI_SDA	I/O	3V3[b]	Z[f]	Leave open	DDC Data (5.5V tolerant)
HDMI_CEC	I/O	3V3	Z	Leave open	CEC (has internal 27k pull up)
HDMI_CLKx	O	-	Z	Leave open	HDMI serial clock
HDMI_Dx	O	-	Z	Leave open	HDMI serial data
HDMI_HPD_N_1V8	I	1V8	Pull High	Leave open	HDMI hotplug detect
CAM0 (CSI0) 2-lane Interface					
CAM0_Cx	I	-	Z	Leave open	Serial clock
CAM0_Dx	I	-	Z	Leave open	Serial data
CAM1 (CSI1) 4-lane Interface					
CAM1_Cx	I	-	Z	Leave open	Serial clock
CAM1_Dx	I	-	Z	Leave open	Serial data
DSI0 (Display 0) 2-lane Interface					
DSI0_Cx	O	-	Z	Leave open	Serial clock
DSI0_Dx	O	-	Z	Leave open	Serial data
DSI1 (Display 1) 4-lane Interface					
DSI1_Cx	O	-	Z	Leave open	Serial clock
DSI1_Dx	O	-	Z	Leave open	Serial data
TV Out					
TVDAC	O	-	Z	Leave open	Composite video DAC output
JTAG Interface					
TMS	I	3V3	Z	Leave open	Has internal 50k pull up
TRST_N	I	3V3	Z	Leave open	Has internal 50k pull up
TCK	I	3V3	Z	Leave open	Has internal 50k pull up
TDI	I	3V3	Z	Leave open	Has internal 50k pull up
TDO	O	3V3	O	Leave open	Has internal 50k pull up

[a] The PDN column indicates power-down state (when RUN pin LOW)
[b] Must be driven by an open-collector driver
[c] GPIO have software enabled pulls which keep state over power-down
[d] Only available on Lite variants
[e] The CM will always try to boot from this interface first
[f] Requires external pull-up resistor to 5V as per HDMI spec

Table 3: Pin Functions

6 Electrical Specification

Caution! Stresses above those listed in Table 4 may cause permanent damage to the device. This is a stress rating only; functional operation of the device under these or any other conditions above those listed in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Symbol	Parameter	Minimum	Maximum	Unit
VBAT	Core SMPS Supply	-0.5	6.0	V
3V3	3V3 Supply Voltage	-0.5	4.10	V
1V8	1V8 Supply Voltage	-0.5	2.10	V
VDAC	TV DAC Supply	-0.5	4.10	V
GPIO0-27_VDD	GPIO0-27 I/O Supply Voltage	-0.5	4.10	V
GPIO28-45_VDD	GPIO28-45 I/O Supply Voltage	-0.5	4.10	V
SDX_VDD	Primary SD/eMMC Supply Voltage	-0.5	4.10	V

Table 4: Absolute Maximum Ratings

DC Characteristics are defined in Table 5

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{IL}	Input low voltage[a]	VDD_IO = 1.8V	-	-	0.6	V
		VDD_IO = 2.7V	-	-	0.8	V
		VDD_IO = 3.3V	-	-	0.9	V
V_{IH}	Input high voltage[a]	VDD_IO = 1.8V	1.0	-	-	V
		VDD_IO = 2.7V	1.3	-	-	V
		VDD_IO = 3.3V	1.6	-	-	V
I_{IL}	Input leakage current	TA = +85°C	-	-	5	μA
C_{IN}	Input capacitance	-	-	5	-	pF
V_{OL}	Output low voltage[b]	VDD_IO = 1.8V, IOL = -2mA	-	-	0.2	V
		VDD_IO = 2.7V, IOL = -2mA	-	-	0.15	V
		VDD_IO = 3.3V, IOL = -2mA	-	-	0.14	V
V_{OH}	Output high voltage[b]	VDD_IO = 1.8V, IOH = 2mA	1.6	-	-	V
		VDD_IO = 2.7V, IOH = 2mA	2.5	-	-	V
		VDD_IO = 3.3V, IOH = 2mA	3.0	-	-	V
I_{OL}	Output low current[c]	VDD_IO = 1.8V, VO = 0.4V	12	-	-	mA
		VDD_IO = 2.7V, VO = 0.4V	17	-	-	mA
		VDD_IO = 3.3V, VO = 0.4V	18	-	-	mA
I_{OH}	Output high current[c]	VDD_IO = 1.8V, VO = 1.4V	10	-	-	mA
		VDD_IO = 2.7V, VO = 2.3V	16	-	-	mA
		VDD_IO = 3.3V, VO = 2.3V	17	-	-	mA
R_{PU}	Pullup resistor	-	50	-	65	kΩ
R_{PD}	Pulldown resistor	-	50	-	65	kΩ

[a] Hysteresis enabled
[b] Default drive strength (8mA)
[c] Maximum drive strength (16mA)

Table 5: DC Characteristics

AC Characteristics are defined in Table 6 and Fig. 3.

Pin Name	Symbol	Parameter	Minimum	Typical	Maximum	Unit
Digital outputs	t_{rise}	10-90% rise time[a]	-	1.6	-	ns
Digital outputs	t_{fall}	90-10% fall time[a]	-	1.7	-	ns
GPCLK	t_{JOSC}	Oscillator-derived GPCLK cycle-cycle jitter (RMS)	-	-	20	ps
GPCLK	t_{JPLL}	PLL-derived GPCLK cycle-cycle jitter (RMS)	-	-	48	ps

[a] Default drive strength, CL = 5pF, VDD_IOx = 3.3V

Table 6: Digital I/O Pin AC Characteristics

Figure 3: Digital IO Characteristics

7 Power Supplies

The Compute Module 3+ has six separate supplies that must be present and powered at all times; you cannot leave any of them unpowered, even if a specific interface or GPIO bank is unused. The six supplies are as follows:

1. VBAT is used to power the BCM2837 processor core. It feeds the SMPS that generates the chip core voltage.
2. 3V3 powers various BCM2837 PHYs, IO and the eMMC Flash.
3. 1V8 powers various BCM2837 PHYs, IO and SDRAM.
4. VDAC powers the composite (TV-out) DAC.
5. GPIO0-27_VREF powers the GPIO 0-27 IO bank.
6. GPIO28-45_VREF powers the GPIO 28-45 IO bank.

Supply	Descripion	Minimum	Typical	Maximum	Unit
VBAT	Core SMPS Supply	2.5	-	5.0 + 5%	V
3V3	3V3 Supply Voltage	3.3 - 5%	3.3	3.3 + 5%	V
1V8	1V8 Supply Voltage	1.8 - 5%	1.8	1.8 + 5%	V
VDAC	TV DAC Supply[a]	2.5 - 5%	2.8	3.3 + 5%	V
GPIO0-27_VDD	GPIO0-27 I/O Supply Voltage	1.8 - 5%	-	3.3 + 5%	V
GPIO28-45_VDD	GPIO28-45 I/O Supply Voltage	1.8 - 5%	-	3.3 + 5%	V
SDX_VDD	Primary SD/eMMC Supply Voltage	1.8 - 5%	-	3.3 + 5%	V

[a] Requires a clean 2.5-2.8V supply if TV DAC is used, else connect to 3V3

Table 7: Power Supply Operating Ranges

7.1 Supply Sequencing

Supplies should be staggered so that the highest voltage comes up first, then the remaining voltages in descending order. This is to avoid forward biasing internal (on-chip) diodes between supplies, and causing latch-up. Alternatively supplies can be synchronised to come up at exactly the same time as long as at no point a lower voltage supply rail voltage exceeds a higher voltage supply rail voltage.

7.2 Power Requirements

Exact power requirements will be heavily dependent upon the individual use case. If an on-chip subsystem is unused, it is usually in a low power state or completely turned off. For instance, if your application does not use 3D graphics then a large part of the core digital logic will never turn on and need power. This is also the case for camera and display interfaces, HDMI, USB interfaces, video encoders and decoders, and so on.

Powerchain design is critical for stable and reliable operation of the Compute Module 3+. We strongly recommend that designers spend time measuring and verifying power requirements for their particular use case and application, as well as paying careful attention to power supply sequencing and maximum supply voltage tolerance.

Table 8 specifies the recommended minimum power supply outputs required to power the Compute Module 3+.

Supply	Minimum Requirement	Unit
VBAT (CM1)	2000[a]	mW
VBAT (CM3,3L)	3500[a]	mW
3V3	250	mA
1V8	250	mA
VDAC	25	mA
GPIO0-27_VDD	50[b]	mA
GPIO28-45_VDD	50[b]	mA
SDX_VDD	50[b]	mA

[a] Recommended minimum. Actual power drawn is very dependent on use-case
[b] Each GPIO can supply up to 16mA, aggregate current per bank must not exceed 50mA

Table 8: Mimimum Power Supply Requirements

8 Booting

The eMMC Flash device on CM3+ is directly connected to the primary BCM2837 SD/eMMC interface. These connections are not accessible on the module pins. On CM3+ Lite this SD interface is available on the SDX_ pins.

When initially powered on, or after the RUN pin has been held low and then released, the BCM2837 will try to access the primary SD/eMMC interface. It will then look for a file called bootcode.bin on the primary partition (which must be FAT) to start booting the system. If it cannot access the SD/eMMC device or the boot code cannot be found, it will fall back to waiting for boot code to be written to it over USB; in other words, its USB port is in slave mode waiting to accept boot code from a suitable host.

A USB boot tool is available on Github which allows a host PC running Linux to write the BCM2837 boot code over USB to the module. That boot code then runs and provides access to the SD/eMMC as a USB mass storage device, which can then be read and written using the host PC. Note that a Raspberry Pi can be used as the host machine. For those using Windows a precompiled and packeged tool is available. For more information see here.

The Compute Module has a pin called EMMC_DISABLE_N which when shorted to GND will disable the SD/eMMC interface (by physically disconnecting the SD_CMD pin), forcing BCM2837 to boot from USB. Note that when the eMMC is disabled in this way, it takes a couple of seconds from powering up for the processor to stop attempting to talk to the SD/eMMC device and fall back to booting from USB.

Note that once booted over USB, BCM2837 needs to re-enable the SD/eMMC device (by releasing EMMC_DISABLE_N) to allow access to it as mass storage. It expects to be able to do this by driving the EMMC_EN_N_1V8 pin LOW, which at boot is initially an input with a pull up to 1V8. If an end user wishes to add the ability to access the SD/eMMC over USB in their product, similar circuitry to that used on the Compute Module IO Board to enable/disable the USB boot and SD/eMMC must be used; that is, EMMC_DISABLE_N pulled low via MOSFET(s) and released again by MOSFET, with the gate controlled by EMMC_EN_N_1V8. **Ensure you use MOSFETs suitable for switching at 1.8V (i.e. use a device with gate threshold voltage, Vt, suitable for 1.8V switching).**

9 Peripherals

9.1 GPIO

BCM2837 has in total 54 GPIO lines in 3 separate voltage banks. All GPIO pins have at least two alternative functions within the SoC. When not used for the alternate peripheral function, each GPIO pin may be set as an input (optionally as an interrupt) or an output. The alternate functions are usually peripheral I/Os, and most peripherals appear twice to allow flexibility on the choice of I/O voltage.

GPIO bank2 is used on the module to connect to the eMMC device and for an on-board I2C bus (to talk to the core SMPS and control the special function pins). On CM3+ Lite most of bank2 is exposed to allow a user to connect their choice of SD card or eMMC device (if required).

Bank0 and 1 GPIOs are available for general use. GPIO0 to GPIO27 are bank0 and GPIO28-45 make up bank1. GPIO0-27_VDD is the power supply for bank0 and GPIO28-45_VDD is the power supply for bank1. SDX_VDD is the supply for bank2 on CM3+ Lite. These supplies can be in the range 1.8V-3.3V (see Table 7) and are not optional; each bank must be powered, even when none of the GPIOs for that bank are used.

Note that the HDMI_HPD_N_1V8 and EMMC_EN_N_1V8 pins are 1.8V IO and are used for special functions (HDMI hot plug detect and boot control respectively). Please do not use these pins for any other purpose, as the software for the module will always expect these pins to have these special functions. If they are unused please leave them unconnected.

All GPIOs except GPIO28, 29, 44 and 45 have weak in-pad pull-ups or pull-downs enabled when the device is powered on. It is recommended to add off-chip pulls to GPIO28, 29, 44 and 45 to make sure they never float during power on and initial boot.

9.1.1 GPIO Alternate Functions

GPIO	Default Pull	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5
0	High	SDA0	SA5	PCLK	-	-	-
1	High	SCL0	SA4	DE	-	-	-
2	High	SDA1	SA3	LCD_VSYNC	-	-	-
3	High	SCL1	SA2	LCD_HSYNC	-	-	-
4	High	GPCLK0	SA1	DPI_D0	-	-	ARM_TDI
5	High	GPCLK1	SA0	DPI_D1	-	-	ARM_TDO
6	High	GPCLK2	SOE_N	DPI_D2	-	-	ARM_RTCK
7	High	SPI0_CE1_N	SWE_N	DPI_D3	-	-	-
8	High	SPI0_CE0_N	SD0	DPI_D4	-	-	-
9	Low	SPI0_MISO	SD1	DPI_D5	-	-	-
10	Low	SPI0_MOSI	SD2	DPI_D6	-	-	-
11	Low	SPI0_SCLK	SD3	DPI_D7	-	-	-
12	Low	PWM0	SD4	DPI_D8	-	-	ARM_TMS
13	Low	PWM1	SD5	DPI_D9	-	-	ARM_TCK
14	Low	TXD0	SD6	DPI_D10	-	-	TXD1
15	Low	RXD0	SD7	DPI_D11	-	-	RXD1
16	Low	FL0	SD8	DPI_D12	CTS0	SPI1_CE2_N	CTS1
17	Low	FL1	SD9	DPI_D13	RTS0	SPI1_CE1_N	RTS1
18	Low	PCM_CLK	SD10	DPI_D14	-	SPI1_CE0_N	PWM0
19	Low	PCM_FS	SD11	DPI_D15	-	SPI1_MISO	PWM1
20	Low	PCM_DIN	SD12	DPI_D16	-	SPI1_MOSI	GPCLK0
21	Low	PCM_DOUT	SD13	DPI_D17	-	SPI1_SCLK	GPCLK1
22	Low	SD0_CLK	SD14	DPI_D18	SD1_CLK	ARM_TRST	-
23	Low	SD0_CMD	SD15	DPI_D19	SD1_CMD	ARM_RTCK	-
24	Low	SD0_DAT0	SD16	DPI_D20	SD1_DAT0	ARM_TDO	-
25	Low	SD0_DAT1	SD17	DPI_D21	SD1_DAT1	ARM_TCK	-
26	Low	SD0_DAT2	TE0	DPI_D22	SD1_DAT2	ARM_TDI	-
27	Low	SD0_DAT3	TE1	DPI_D23	SD1_DAT3	ARM_TMS	-

Table 9: GPIO Bank0 Alternate Functions

GPIO	Default Pull	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5
28	None	SDA0	SA5	PCM_CLK	FL0	-	-
29	None	SCL0	SA4	PCM_FS	FL1	-	-
30	Low	TE0	SA3	PCM_DIN	CTS0	-	CTS1
31	Low	FL0	SA2	PCM_DOUT	RTS0	-	RTS1
32	Low	GPCLK0	SA1	RING_OCLK	TXD0	-	TXD1
33	Low	FL1	SA0	TE1	RXD0	-	RXD1
34	High	GPCLK0	SOE_N	TE2	SD1_CLK	-	-
35	High	SPI0_CE1_N	SWE_N	-	SD1_CMD	-	-
36	High	SPI0_CE0_N	SD0	TXD0	SD1_DAT0	-	-
37	Low	SPI0_MISO	SD1	RXD0	SD1_DAT1	-	-
38	Low	SPI0_MOSI	SD2	RTS0	SD1_DAT2	-	-
39	Low	SPI0_SCLK	SD3	CTS0	SD1_DAT3	-	-
40	Low	PWM0	SD4	-	SD1_DAT4	SPI2_MISO	TXD1
41	Low	PWM1	SD5	TE0	SD1_DAT5	SPI2_MOSI	RXD1
42	Low	GPCLK1	SD6	TE1	SD1_DAT6	SPI2_SCLK	RTS1
43	Low	GPCLK2	SD7	TE2	SD1_DAT7	SPI2_CE0_N	CTS1
44	None	GPCLK1	SDA0	SDA1	TE0	SPI2_CE1_N	-
45	None	PWM1	SCL0	SCL1	TE1	SPI2_CE2_N	-

Table 10: GPIO Bank1 Alternate Functions

Table 9 and Table 10 detail the default pin pull state and available alternate GPIO functions. Most of these alternate peripheral functions are described in detail in the Broadcom Peripherals Specification document and have Linux drivers available.

9.1.2 Secondary Memory Interface (SMI)

The SMI peripheral is an asynchronous NAND type bus supporting Intel mode80 type transfers at 8 or 16 bit widths and available in the ALT1 positions on GPIO banks 0 and 1 (see Table 9 and Table 10). It is not publicly documented in the Broadcom Peripherals Specification but a Linux driver is available in the Raspberry Pi Github Linux repository (bcm2835_smi.c in linux/drivers/misc).

9.1.3 Display Parallel Interface (DPI)

A standard parallel RGB (DPI) interface is available on bank 0 GPIOs. This up-to-24-bit parallel interface can support a secondary display. Again this interface is not documented in the Broadcom Peripherals Specification but documentation can be found here.

9.1.4 SD/SDIO Interface

The BCM283x supports two SD card interfaces, SD0 and SD1.

The first (SD0) is a proprietary Broadcom controller that does not support SDIO and is the primary interface used to boot and talk to the eMMC or SDX_x signals.

The second interface (SD1) is standards compliant and can interface to SD, SDIO and eMMC devices; for example on a Raspberry Pi 3 B+ it is used to talk to the on-board CYW43455 WiFi device in SDIO mode.

Both interfaces can support speeds up to 50MHz single ended (SD High Speed Mode).

9.2 CSI (MIPI Serial Camera)

Currently the CSI interface is not openly documented and only CSI camera sensors supported by the official Raspberry Pi firmware will work with this interface. Supported sensors are the OmniVision OV5647 and Sony IMX219.

It is recommended to attach other cameras via USB.

9.3 DSI (MIPI Serial Display)

Currently the DSI interface is not openly documented and only DSI displays supported by the official Raspberry Pi firmware will work with this interface.

Displays can also be added via the parallel DPI interface which is available as a GPIO alternate function - see Table 9 and Section 9.1.3

9.4 USB

The BCM2837 USB port is On-The-Go (OTG) capable. If using either as a fixed slave or fixed master, please tie the USB_OTGID pin to ground.

The USB port (Pins USB_DP and USB_DM) must be routed as 90 ohm differential PCB traces.

Note that the port is capable of being used as a true OTG port however there is no official documentation. Some users have had success making this work.

9.5 HDMI

BCM283x supports HDMI V1.3a.

It is recommended that users follow a similar arrangement to the Compute Module IO Board circuitry for HDMI output.

The HDMI CK_P/N (clock) and D0-D2_P/N (data) pins must each be routed as matched length 100 ohm differential PCB traces. It is also important to make sure that each differential pair is closely phase matched. Finally, keep HDMI traces well away from other noise sources and as short as possible.

Failure to observe these design rules is likely to result in EMC failure.

9.6 Composite (TV Out)

The TVDAC pin can be used to output composite video (PAL or NTSC). Please route this signal away from noise sources and use a 75 ohm PCB trace.

Note that the TV DAC is powered from the VDAC supply which must be a clean supply of 2.5-2.8V. It is recommended users generate this supply from 3V3 using a low noise LDO.

If the TVDAC output is not used VDAC can be connected to 3V3, but it must be powered even if the TV-out functionality is unused.

10 Thermals

The BCM2837 SoC employs DVFS (Dynamic Voltage and Frequency Scaling) on the core voltage. When the processor is idle (low CPU utilisation), it will reduce the core frequency and voltage to reduce current draw and heat output. When the core utilisation exceeds a certain threshold the core votlage is increased and the core frequency is boosted to the maximum working frerquency of 1.2GHz. The voltage and frequency are throttled back when the CPU load reduces back to an 'idle' level OR when the silicon temperature as mesured by the on-chip temperature sensor exceeds 80C (thermal throttling).

A designer must pay careful attention to the thermal design of products using the CM3+ so that performance is not artificially curtailed due to the processor thermal throttling, as the Quad ARM complex in the BCM2837 can generate significant heat output under load.

10.1 Temperature Range

The operating temperature range of the module is set by the lowest maximum and highest minimum of any of the components used.

The eMMC and LPDDR2 have the narrowest range, these are rated for -25 to +80 degrees Celsius. Therefore the nominal range for the CM3+ and CM3+ Lite is -25C to +80C.

However, this range is the maximum for the silicon die; therefore, users would have to take into account the heat generated when in use and make sure this does not cause the temperature to exceed 80 degrees Celsius.

11 Availability

Raspberry Pi guarantee availability of CM3+ and CM3+ Lite until at least January 2026.

12 Support

For support please see the hardware documentation section of the Raspberry Pi website and post questions to the Raspberry Pi forum.

MPU6050

6.2 Accelerometer Specifications

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

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

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

6.3 Electrical and Other Common Specifications

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

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

6.4 Electrical Specifications, Continued

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

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

6.5 Electrical Specifications, Continued

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

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

6.6 Electrical Specifications, Continued

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

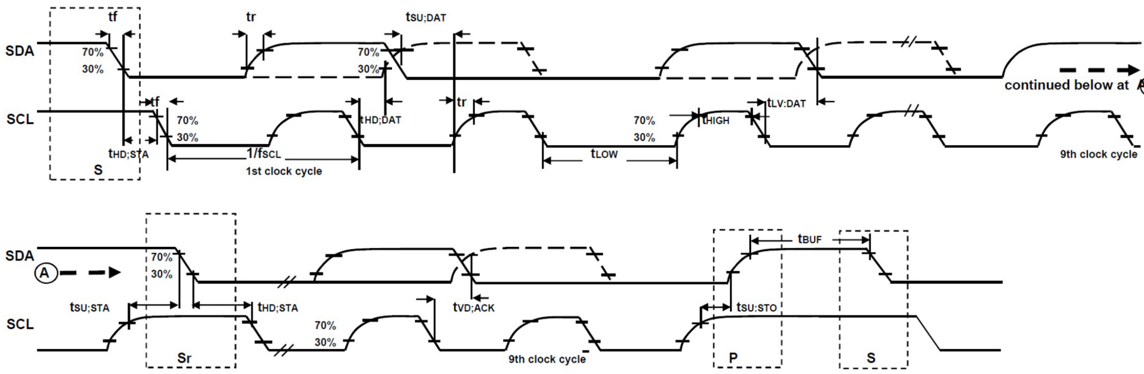
Parameters	Conditions	Min	Typical	Max	Units	Notes
INTERNAL CLOCK SOURCE	**CLK_SEL=0,1,2,3**					
Gyroscope Sample Rate, Fast	DLPFCFG=0 SAMPLERATEDIV = 0		8		kHz	
Gyroscope Sample Rate, Slow	DLPFCFG=1,2,3,4,5, or 6 SAMPLERATEDIV = 0		1		kHz	
Accelerometer Sample Rate			1		kHz	
Clock Frequency Initial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

6.7 I²C Timing Characterization

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

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

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

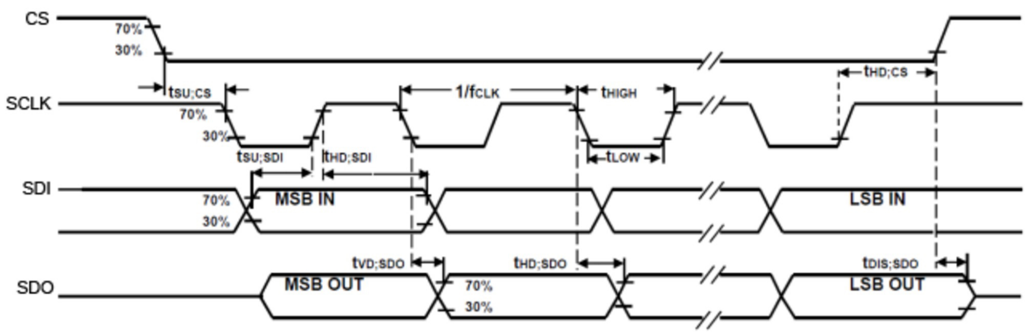


I²C Bus Timing Diagram

6.8 SPI Timing Characterization (MPU-6000 only)

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

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



SPI Bus Timing Diagram

6.9 Absolute Maximum Ratings

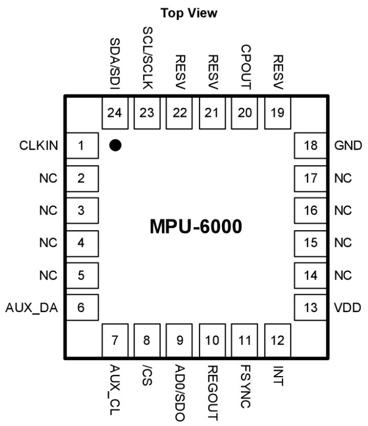
Stress above those listed as “Absolute Maximum Ratings” may cause permanent damage to the device. These are stress ratings only and functional operation of the device at these conditions is not implied. Exposure to the absolute maximum ratings conditions for extended periods may affect device reliability.

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

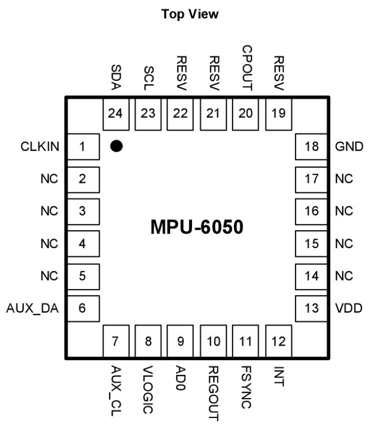
7 Applications Information

7.1 Pin Out and Signal Description

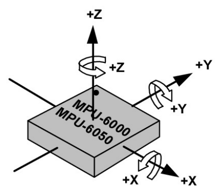
Pin Number	MPU-6000	MPU-6050	Pin Name	Pin Description
1	Y	Y	CLKIN	Optional external reference clock input. Connect to GND if unused.
6	Y	Y	AUX_DA	I^{2}C master serial data, for connecting to external sensors
7	Y	Y	AUX_CL	I^{2}C Master serial clock, for connecting to external sensors
8	Y		/CS	SPI chip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



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

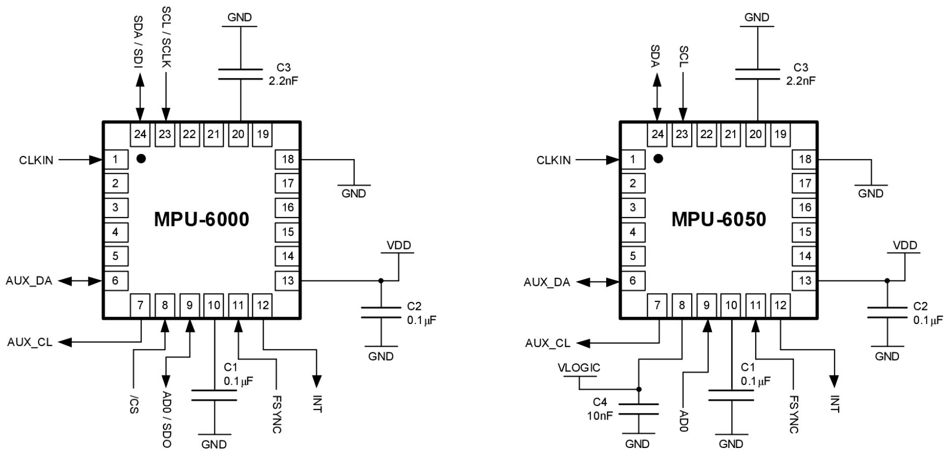


QFN Package
24-pin, 4mm x 4mm x 0.9mm



Orientation of Axes of Sensitivity and Polarity of Rotation

7.2 Typical Operating Circuit



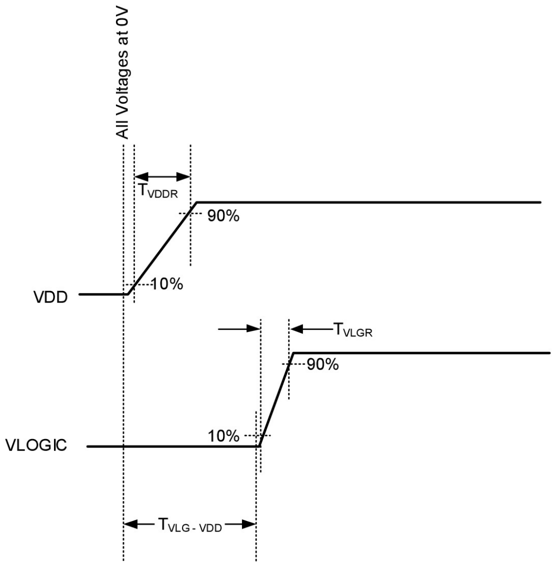
Typical Operating Circuits

7.3 Bill of Materials for External Components

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

* MPU-6050 Only.

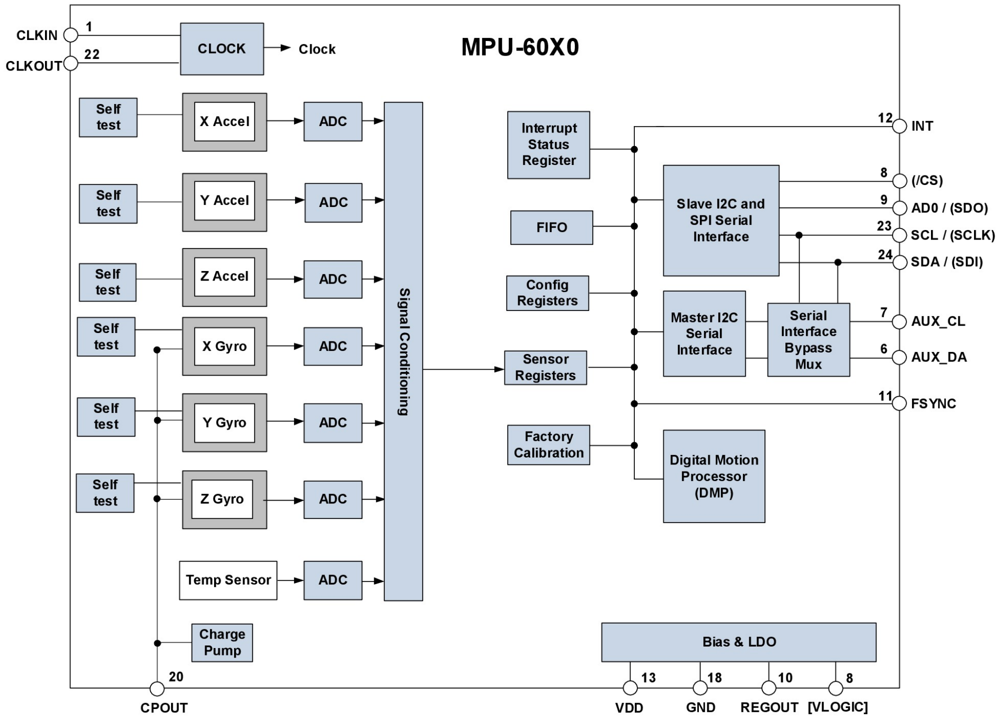
7.4 Recommended Power-on Procedure



Power-Up Sequencing

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

7.5 Block Diagram



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

7.6 Overview

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

- Three-axis MEMS rate gyroscope sensor with 16-bit ADCs and signal conditioning
- Three-axis MEMS accelerometer sensor with 16-bit ADCs and signal conditioning
- Digital Motion Processor (DMP) engine
- Primary I^{2}C and SPI (MPU-6000 only) serial communications interfaces
- Auxiliary I^{2}C serial interface for 3rd party magnetometer & other sensors
- Clocking
- Sensor Data Registers
- FIFO
- Interrupts
- Digital-Output Temperature Sensor
- Gyroscope & Accelerometer Self-test
- Bias and LDO
- Charge Pump

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

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

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

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

7.9 Digital Motion Processor

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

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

7.10 Primary I^{2}C and SPI Serial Communications Interfaces

The MPU-60X0 communicates to a system processor using either a SPI (MPU-6000 only) or an I^{2}C serial interface. The MPU-60X0 always acts as a slave when communicating to the system processor. The LSB of the of the I^{2}C slave address is set by pin 9 (AD0).

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

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

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

7.11 Auxiliary I^{2}C Serial Interface

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

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

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

- I^{2}C Master Mode: Allows the MPU-60X0 to directly access the data registers of external digital sensors, such as a magnetometer. In this mode, the MPU-60X0 directly obtains data from auxiliary sensors, allowing the on-chip DMP to generate sensor fusion data without intervention from the system applications processor.

  For example, In I^{2}C Master mode, the MPU-60X0 can be configured to perform burst reads, returning the following data from a magnetometer:

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

  The I^{2}C Master can be configured to read up to 24 bytes from up to 4 auxiliary sensors. A fifth sensor can be configured to work single byte read/write mode.

- Pass-Through Mode: Allows an external system processor to act as master and directly communicate to the external sensors connected to the auxiliary I^{2}C bus pins (AUX_DA and AUX_CL). In this mode, the auxiliary I^{2}C bus control logic (3rd party sensor interface block) of the MPU-60X0 is disabled, and the auxiliary I^{2}C pins AUX_DA and AUX_CL (Pins 6 and 7) are connected to the main I^{2}C bus (Pins 23 and 24) through analog switches.

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

  In Pass-Through Mode the system processor can still access MPU-60X0 data through the I^{2}C interface.

Auxiliary I^{2}C Bus IO Logic Levels

- MPU-6000: The logic level of the auxiliary I^{2}C bus is VDD
- MPU-6050: The logic level of the auxiliary I^{2}C bus can be programmed to be either VDD or VLOGIC

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

7.12 Self-Test

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

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

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

The self-test response is defined as follows:

Self-test response = Sensor output with self-test enabled – Sensor output without self-test enabled

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

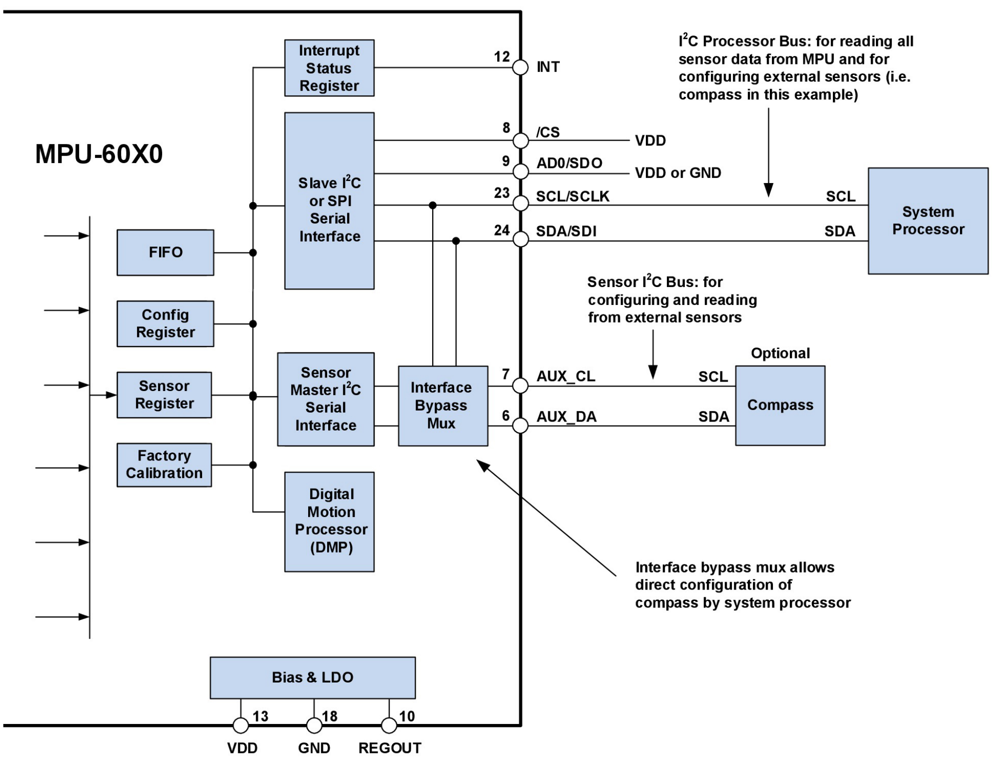
When the value of the self-test response is within the min/max limits of the product specification, the part has passed self test. When the self-test response exceeds the min/max values, the part is deemed to have failed self-test. Code for operating self test code is included within the MotionApps software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

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

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

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

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



7.14 MPU-6000 Using SPI Interface

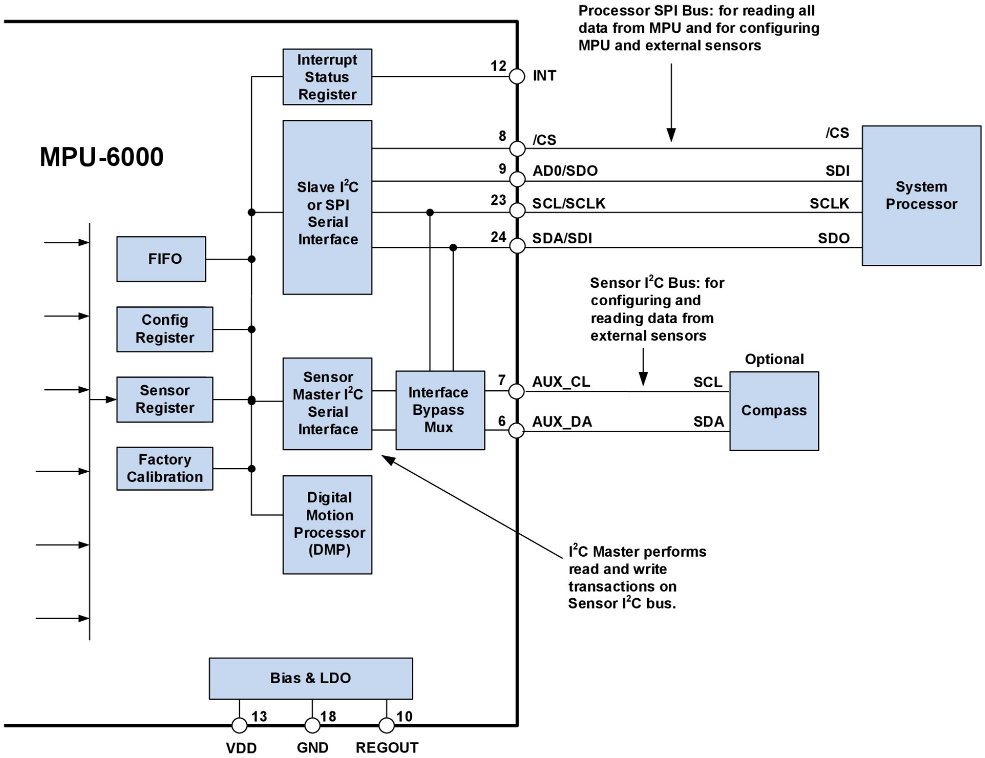
In the figure below, the system processor is an SPI master to the MPU-6000. Pins 8, 9, 23, and 24 are used to support the /CS, SDO, SCLK, and SDI signals for SPI communications. Because these SPI pins are shared with the I²C slave pins (9, 23 and 24), the system processor cannot access the auxiliary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

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

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

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

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



7.15 Internal Clock Generation

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

Allowable internal sources for generating the internal clock are:

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

Allowable external clocking sources are:

- 32.768kHz square wave
- 19.2MHz square wave

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

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

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

7.16 Sensor Data Registers

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

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

7.17 FIFO

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

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

7.18 Interrupts

Interrupt functionality is configured via the Interrupt Configuration register. Items that are configurable include the INT pin configuration, the interrupt latching and clearing method, and triggers for the interrupt. Items that can trigger an interrupt are (1) Clock generator locked to new reference oscillator (used when switching clock sources); (2) new data is available to be read (from the FIFO and Data registers); (3) accelerometer event interrupts; and (4) the MPU-60X0 did not receive an acknowledge from an auxiliary sensor on the secondary I^2C bus. The interrupt status can be read from the Interrupt Status register.

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

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

7.19 Digital-Output Temperature Sensor

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

7.20 Bias and LDO

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

7.21 Charge Pump

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