Laboratorio de Microcomputadoras - 66.09

Controlador de Posición Angular

Profesor:	Ing. Guillermo Campiglio
Cuatrimestre/Año:	1º/2017
Turno de las clases prácticas	Miércoles
Jefe de trabajos prácticos:	Ing. Ricardo Arias
Docente guía:	Ing. Ricardo Arias

Autores			Seguimiento del proyecto								
Nombre	Apellido	Padrón									
Ignacio	de Pedro Mermier	97793									
Marco	Torcasso	97792									

Observaciones:

....
....
....
....
....
....
....
....
....

Fecha de aprobación				Firma J.T.P

Coloquio	
Nota final	
Firma profesor	

Índice

1. Objetivos del Proyecto **1**

2. Descripción del Proyecto **2**

2.1. Bloque MicroControlador uC 2

2.2. Bloque LCD 2

2.3. Bloque IMU 2

2.4. Bloque Potenciómetros 2

2.5. Bloque Servomotores 3

2.6. Bloque Bluetooth 3

2.7. Bloque Selección 3

3. Diagrama de conexión en bloques **4**

4. Software - Diagrama de Flujo **5**

4.1. Diagramas de Flujo 5

4.2. Código de Programa 8

4.3. Interrupciones 8

4.4. Rutinas 8

5. Circuito Esquemático **11**

6. Resultados **12**

7. Conclusiones **13**

8. Apéndice **14**

8.1. Manual de uso 14

8.2. Listado de componentes y Presupuesto 15

8.3. Hojas de Datos 15

8.4. Código de Programa 15

1. Objetivos del Proyecto

El siguiente proyecto se define como un dispositivo capaz de controlar la posición angular con dos grados de libertad (elevación y alabeo) mediante el uso de dos servomotores. Este dispositivo contará con diversos modos de funcionamiento, acorde al objetivo que se persiga. Las diferentes funciones implicarán el uso de distintas entradas de control y periféricos propios del MicroControlador.

El proyecto posee dos modo de operación: Modo de operación Directo y modo de operación Realimentado.

- **Modo de operación Directo**: Por medio de 2 potenciómetros (uno para cada grado de libertad de la posición angular) o un sensor IMU (unidad de medición inercial) controlar el ángulo de inclinación de una tabla directamente.

- **Modo de operación Realimentado**: Haciendo uso de la unidad de medición inercial, realizar un control realimentado que permita, por ejemplo, mantener una tabla con una inclinación fija frente a movimientos externos del dispositivo.

El proyecto lo tanto presenta distintas funcionalidades:

- **Control de Estabilidad**: Para el caso del modo de operación realimentado, un caso de aplicación es el control de estabilidad: rotando 180 grados la base que sostiene los dos servomotores, los movimientos realizados por la unidad de medición inercial (IMU) se traducirán de forma contraria por sobre la base de apoyo controlada por los servomotores, generando que la posición de la base de apoyo en cuestión no se modifique desde un observador externo al dispositivo. Un ejemplo de aplicación podría ser para vehículos utilizados en investigación aeroespacial, donde el terreno suele ser engorroso y es necesario realizar un control de su estabilidad.

- **Control Angular**: Para el caso del modo de operación Directo, un caso de aplicación es el control de una posición angular determinada. Por medio de los potenciómetros es factible lograr establecer un ángulo deseado en la base de apoyo y sin modificar los valores de los mismos (potenciómetros) el ángulo establecido no se modificará. Un ejemplo de aplicación podría ser para simulación de trayectorias de distintos elementos en planos inclinados, o la medición de la posición angular de un objeto en el cual no sea sencilla su medición.

2. Descripción del Proyecto

El proyecto realizado consta de distintos bloques de operación, cada uno con una finalidad distinta, descriptos a continuación:

2.1. Bloque MicroControlador uC

El núcleo central del proyecto en cuestión es el MicroControlador ATmega88pa, el cual se encarga de adquirir, procesar y devolver los datos necesarios para el correcto funcionamiento del proyecto. En primer medida se encarga de inicializar los distintos bloques conectados al mismo, y luego de ”correr” el programa encargado de controlar cada bloque, así como ejecutar la función de operación deseada por el usuario.
Para realizar el control de los distintos bloques se utilizaron los distintos periféricos que el MicroControlador posee integrado: Los puertos de entrada/salida, 2 bloques de timer, el puerto de comunicación serie UART, 2 conversores analógico-digital y el puerto de comunicación serie I2C.

2.2. Bloque LCD

Con el fin de visualizar los datos obtenidos por el MicroControlador, se utilizó un LCD de 2 lineas y 16 columnas. Comunicado con el MicroControlador por medio de los puertos de entrada/salida, el mismo se encarga de escribir la posición angular de cada uno de los servomotores, dentro del rango de -90 y 90; junto con la función implementada por el MicroControlador en ese momento, siendo:

- **MPU**: la función en la cual el control de posición es realizado por medio de la unidad de medición inercial .
- **ADC**: la función en la cual el control de posición es realizado por medio del movimiento de los potenciómetros.

2.3. Bloque IMU

Una de las funciones que posee el proyecto es medir la posición angular por medio de un sensor IMU y reflejarla en la base de apoyo a controlar. Este sensor es capaz de medir la aceleración, por medio de un acelerómetro, y la velocidad angular, por medio de un giroscopio. El mismo se comunica con el MicroControlador a través del protocolo I2C y este último procesa ambos datos (aceleración y velocidad angular), obteniendo así el resultado de la posición angular.

2.4. Bloque Potenciómetros

La otra función que posee el proyecto es controlar la posición angular de la base de apoyo de forma manual, por medio de dos potenciómetros. Utilizando el conversor analógico-digital propio del MicroControlador, se toma el valor de la tensión en el punto medio de los potenciómetros para luego convertir dicho valor obtenido en una posición angular determinada.

2.5. Bloque Servomotores

Para realizar el control de la base de apoyo se utilizaron dos servomotores, dado que los mismos permiten de forma sencilla realizar un control de posición de sus ejes. Por medio del bloque de Timer 1 (TC1) en modo 10 bits y fast PWM, se generó dos señales de PWM de aproximadamente 70 Hz (frecuencia de operación de los servomotores) de pulso variable entre 1ms y 2ms en función de la posición angular deseada (tanto para la función ADC como para la función MPU).

2.6. Bloque Bluetooth

Como método de comunicación con una terminal externa se decidió utilizar un módulo BlueTooth conectado con el MicroControlador por medio del módulo UART. Este bloque se encarga de transmitir los datos de la posición angular del sensor IMU, cuando el dispositivo se encuentra dentro de la función MPU.

2.7. Bloque Selección

Encargado de controlar las distintas funciones del dispositivo, se implemento un pulsador normal abierto conectado a un puerto de entrada/salida del MicroControlador.

3. Diagrama de conexión en bloques

La siguiente Figura muestra el diagrama de conexión en bloques del proyecto realizado

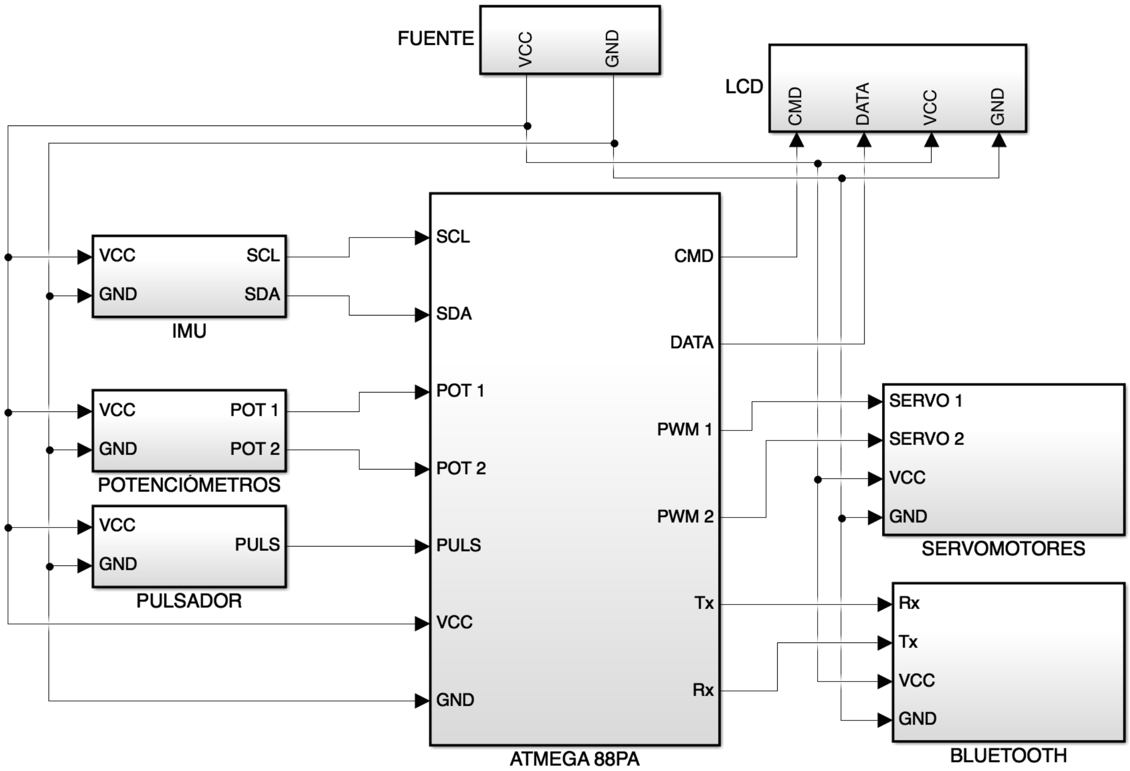


Figura 1. *Diagrama de Conexión en Bloques*

4. Software - Diagrama de Flujo

4.1. Diagramas de Flujo

En las figuras siguientes se describirán los diagramas de flujo correspondientes para la compresión del programa realizado.

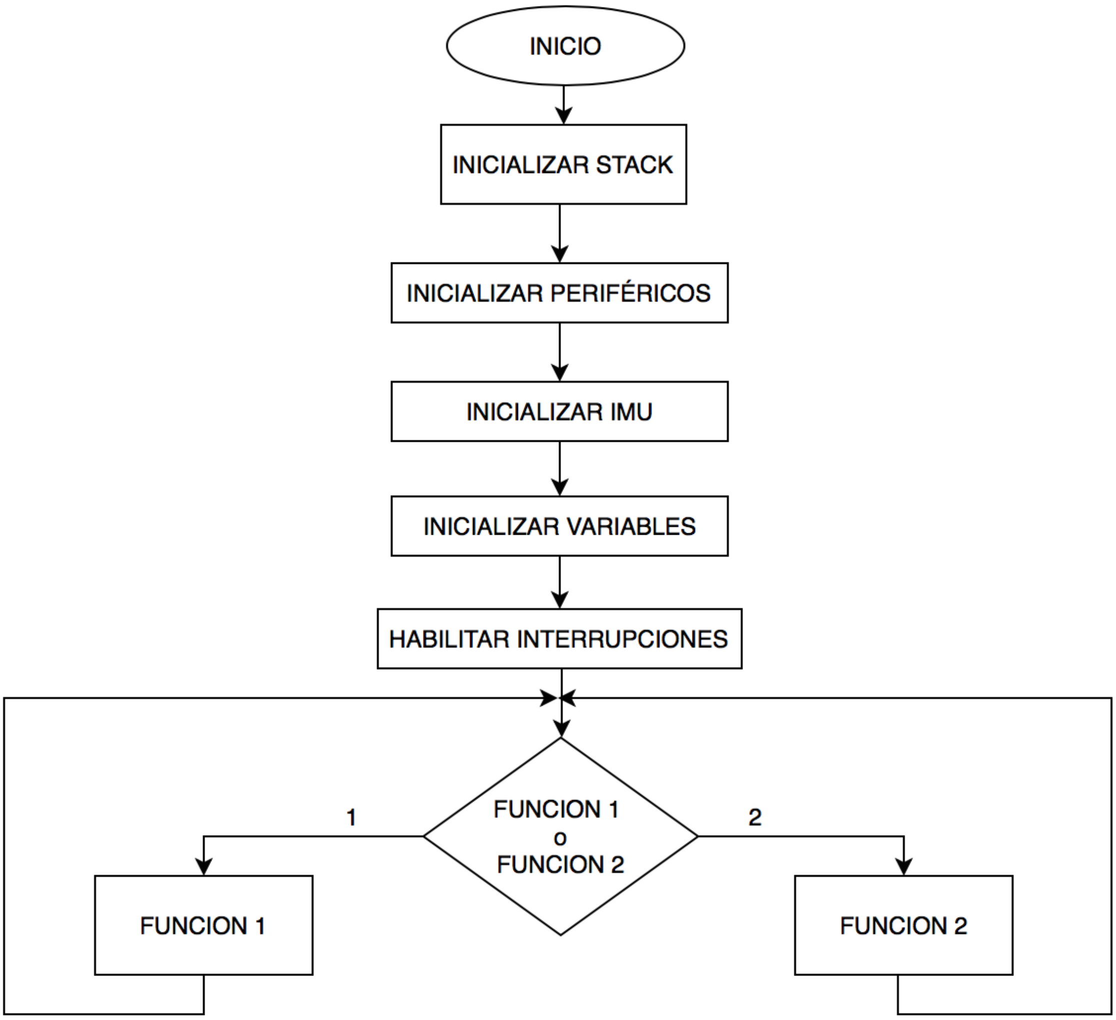


Figura 2. *Diagrama de Flujo Programa Principal*

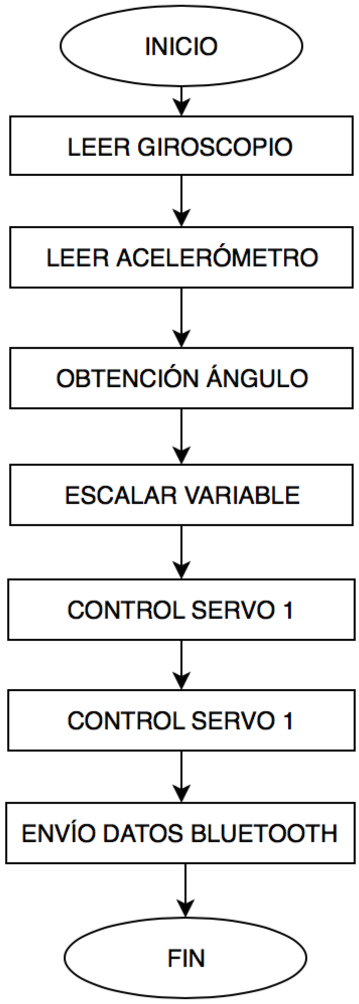


Figura 3. *Diagrama de Flujo Función MPU*

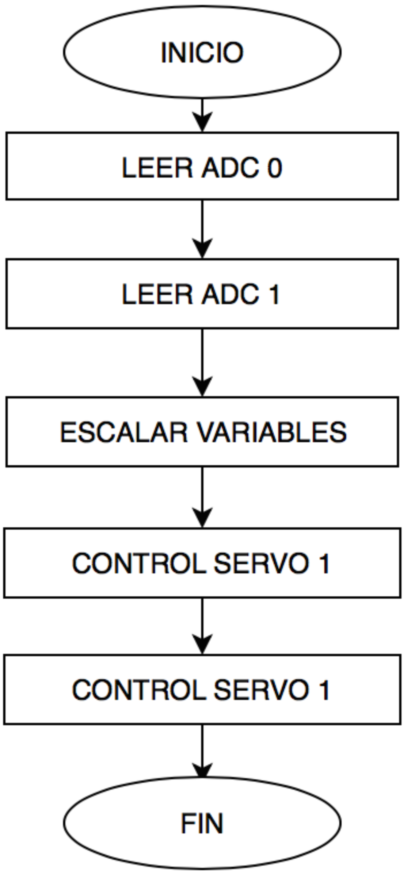


Figura 4. *Diagrama de Flujo Función ADC*

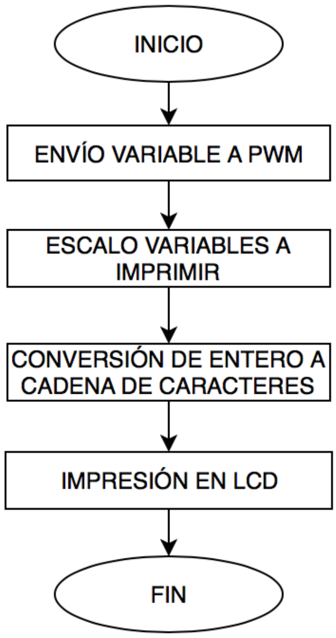


Figura 5. *Diagrama de Flujo Control Servo*

4.2. Código de Programa

Por cuestiones de prolijidad, el código del programa se encuentra en el Apéndice.

4.3. Interrupciones

En el programa implementado para el funcionamiento del dispositivo, se utilizó una interrupción de caracter temporal, es decir, una interrupción por Timer, la cual se activa cada 8ms. El objetivo de la misma es poder realizar una adquisición y posterior procesamiento de datos provenientes del sensor IMU, dado que de esta manera me permite integrar los valores del giroscopio y acelerómetro de forma correcta.

4.4. Rutinas

Dentro del programa se implementaron las siguientes rutinas:

- Inicializar Stack: Coloca el Stack al final de la memoria RAM, evitando la reescritura de un registro ubicado en la memoria RAM.
- Configure Ports: Configura los puertos de entrada/salida según la función de cada pin del MicroControlador.
- Configure LCD: Configura el LCD en modo de 4 bits para utilizar menos pines de DATA.

- Print CAL: imprime en pantalla el mensaje "CALIBRANDO SENSOR".
- Configure I2C: Configura el módulo I2C del MicroControlador en función de las especificaciones del sensor IMU.
- Configure UART: Configura el módulo UART el MicroControlador en función de las especificaciones del módulo BlueTooth.
- Configure timer: Configura el TIMER 0 del MicroControlador para realizar el control de interrupción por Timer.
- Configura PWM: Configura el TIMER 1 del MicroControlador para realizar el control de los servomotores.
- Configure ADC: Configura el módulo ADC del MicroControlador para realizar la adquisición de datos de los potenciómetros.
- Init MPU: Configura el sensor IMU para la recepción y transmisión de datos desde y hacia el MicroControlador.
- Calibrar MPU: Calibra el sensor IMU para obtener la posición inicial según la inclinación que posea en ese momento.
- Inicializar Variables: Limpia todas las variables utilizadas en el control del sensor IMU.
- CLEAR LCD: Limpia la pantalla del LCD.
- PRINT MSG: Imprime en LCD mensajes "SERVO 1: " y "SERVO 2: ".
- MSG FUN: Imprime en LCD la función realizada por el MicroControlador en un momento dado.
- LCD POS: Imprime en LCD los valores de la posición angular de los servomotores.
- PWM TO LCD: Escalamiento de los registros de PWM para obtener el valor correspondiente entre -90 y +90 grados.
- INT TO STR: Conversion de un número de 3 cifras a una cadena de caracteres en ASCII.
- Read Gyro: Obtengo los valores del giroscopio del sensor IMU.
- Read Acc: Obtengo los valores del acelerómetro del sensor IMU.
- Angulo Acc: obtengo el valor de la posición angular por medio de los transmitidos por el sensor IMU.
- Escalo pr: escalamiento de las variables de Pitch y Roll dentro de un rango de -90 y +90.
- Timer Interrupt: Handler de la interrupción por Timer para el procesamiento de datos enviados por sensor IMU.
- PWM CHANGE 0 y 1: Escalamiento de variables Ptich y Roll para colocar en registros de PWM y posterior escritura de registros.
- SEND PWM 0 y 1: Envío de valores obtenidos en "Escalar ADC" hacia los registros de PWM.
- Write UART: Envio por el módulo UART el valor de posición de un servomotor hacia el módulo BlueTooth.

- Read ADC: Adquisición de los valores de conversor analógico-digital de los puertos ADC 0 y ADC 1.
- Escalar ADC: Escalamiento de los valores obtenidos por el ADC para colocar en registros de PWM.
- POL: Lectura del pulsador de selección de función.
- DELAYS: Encargadas de realizar retardos.
- MULT: Realiza el producto entre dos registros de 8 bits y obtención de la parte alta de dicho resultado,
- SUM S16 S8: Suma 2 registros con signo de 16 y 8 bits.
- MUL SIGNED16: Realiza el producto entre 1 registros con signo de 16 bits y un registro sin signo de 8 bits.

5. Circuito Esquemático

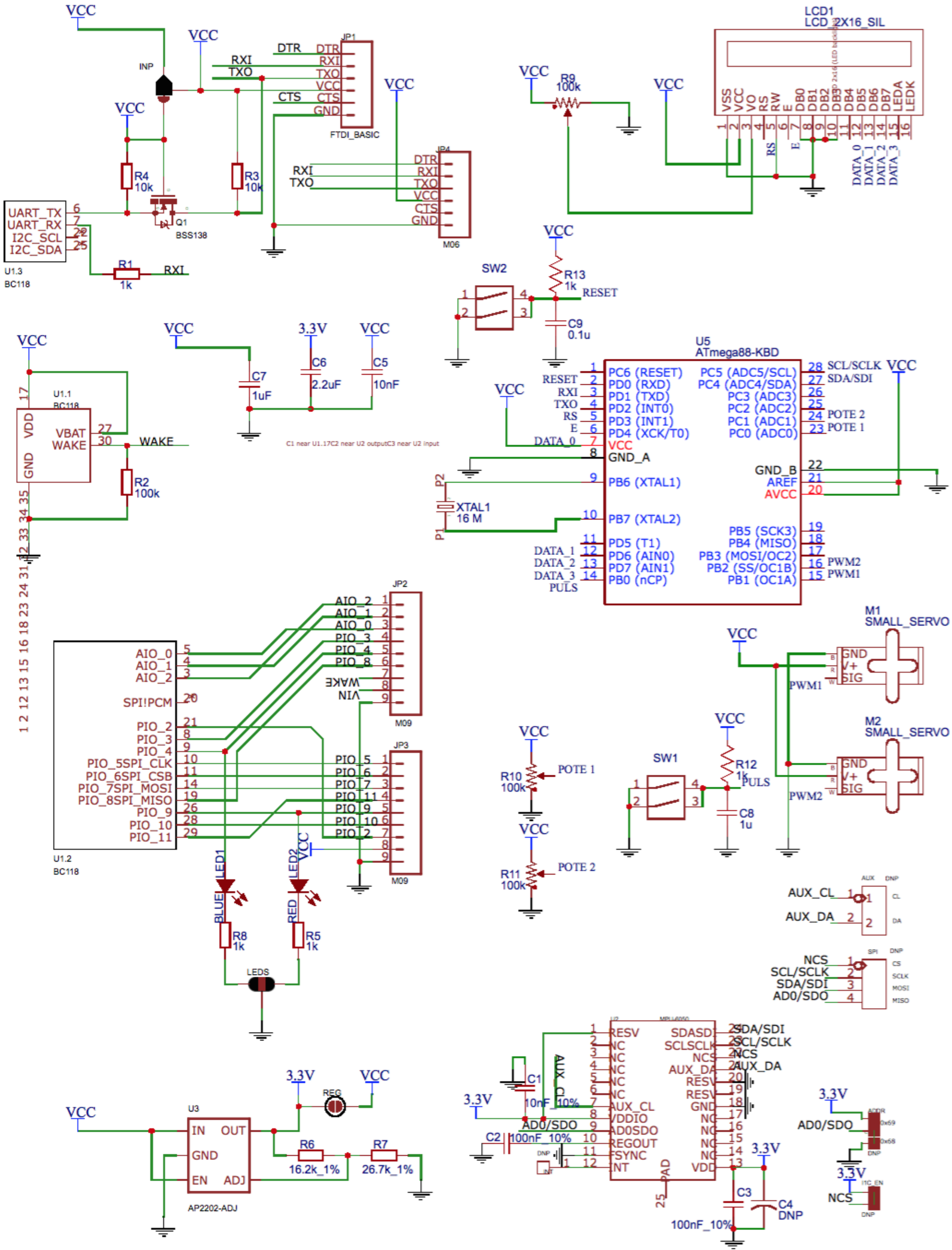


Figura 6. *Circuito Esquemático del Proyecto*

6. Resultados

Los resultados obtenidos en el proyecto son consistentes con los objetivos determinados al comienzo de el presente informe. Se observa de forma clara un correcto funcionamiento del dispositivo en cuestión, y consecuentemente dentro de cada una de las funciones.

Se observa un correcto funcionamiento de la función MPU, dado los servomotores copian la trayectoria realizada por el sensor IMU (con una cota de error de aproximadamente 5 %).

Se observa un correcto funcionamiento de la función ADC, dado que los servomotores se mueven de forma esperada con el movimiento de los potenciómetros (con una cota de error de aproximadamente 2 %).

Se observa un correcto funcionamiento del LCD, indicando la posición angular de los servomotores, independientemente de la función implementada.

Se observa un correcto funcionamiento del pulsador de selección de función, sin presencia de rebote de pulsador.

Se observa un correcto funcionamiento de la comunicación BlueTooth, dado que envía a tiempo real el valor de la posición de cada uno de los servomotores.

Se observa un error de posición dentro de la función MPU, para los valores superiores a 60 grados e inferiores a -60 grados. Esto se debe ya que para la obtención de la posición angular, es necesario implementar funciones matemática, tales como el arcotangente; y, al estar trabajando en lenguaje Assembler, sin la oportunidad de una librería matemática, es muy complicado lograr la implementación de dichas funciones. Por lo que se decidió interpolar las funciones matemáticas a realizar dentro de un rango de valores entre -60 y +60 grados.

7. Conclusiones

El dispositivo logrado, cumple con las expectativas y objetivos previamente informados, dado que se en los resultados se evidencia un correcto funcionamiento.

Dentro de los aciertos podemos destacar:

- Correcto uso del sensor IMU y posterior procesamiento de datos.
- Correcto uso del modulo Bluetooth.
- Correcto seguimiento entre los valores indicados por pantalla y realizados por los servomotores.
- Correcto uso del MicroControlador, realización de rutinas y uso de periféricos.
- Correcto uso del LCD.

Dentro de los errores obtenidos podemos destacar:

- Error en el procesamiento de los datos de posición angular para valores inferiores a -60 y superiores a +60.

Acerca de las sugerencias sobre futuras mejores en el proyecto, cabe destacar la posibilidad de realizar un seguimiento de la posición angular según la luz del sol, con el fin de aprovechar de forma óptima la luz solar (ejemplo: panel solar). Para la implementación del mismo, sería necesaria la colocación de sensores de luz sobre la base de apoyo (ejemplo: LDR), realizar un control por medio de conversores analógicos-digitales, con el fin de mantener siempre la resistencia maxima posible dentro de cada sensor de luz.

8. Apéndice

8.1. Manual de uso

Para la correcta utilización del proyecto realizado es necesario seguir una serie de pasos detallados a continuación:

- Previo a la conexión del dispositivo es necesario prescindir de una fuente de corriente continua de 5V, capaz de suministrar una corriente de 1 Ampere aproximadamente, con el fin de controlar correctamente los servomotores.
- Conexión de dos cables banana-banana entre la fuente de alimentación y los conectores hembras ubicados en la cara lateral derecha.
- Realizada la conexión de alimentación se observará en en LCD el mensaje "CALIBRANDO SENSOR", no mover el dispositivo durante etapa, para una correcta calibración del sensor IMU.
- Se observará en pantalla el mensaje de posición angular inicial de los dos servomotores "+ 000"; y en la zona superior derecha el mensaje "MPU", indicando que nos encontramos dentro de dicha función. Por lo que los servomotores, se moverán en función del movimiento realizado por el sensor IMU, ubicado dentro de la caja contenedora.
- Para cambiar hacia la función ADC, presionar el pulsados colocado en la zona inferior izquierda de la caja contenedora. Al presionar se observará un posible cambio de posición de los servomotores y el mensaje "ADC" en la zona superior derecha del LCD (reemplazando al mensaje "MPU"). Dentro de esta función la posición de los servomotores estará controlada por medio de la posición de los potenciómetros ubicados en la zona inferior izquierda de la caja contenedora.
- Para realizar el modo de operación Realimentado es necesario colocar la base de apoyo controlada por los servomotores sobre la zona superior de la caja contenedora rotada 180 grados respecto de su posición inicial.

8.2. Listado de componentes y Presupuesto

Para el desarrollo y armado del proyecto se utilizaron los siguientes materiales, indicando su correspondiente precio de mercado:

- 1 x Kit Atmega88pa LABI (200$ ARS).
- 1 x Kit USBTiny LABI (120$ ARS)
- 1 x LCD Winstar Wh1602 (149$ ARS).
- 1 x Sensor IMU: MPU 6050, I2C (120$ ARS).
- 1 x Módulo BlueTooth: HC06 (150$ ARS).
- 2 x ServoMotor SG90 (5$ USD por unidad).
- 2 x Potenciómetros lineales 50 K (15$ ARS por unidad).
- 1 x Pulsador (15$ ARS).
- 5 x capactores electrolíticos (valores varios) (47 $ ARS total).
- 1 x Placa Multiperforada (88$ ARS).
- 1 x Kit de cables Arduino (159$ ARS).
- 1 x Cable 0,7mm (1 m) (30$ ARS).
- 2 x Ficha Banana Hembra (10$ ARS por unidad).
- Materiales varios para el armado (200$ ARS).

Presupuesto Total aproximado: 1400 $ ARS

8.3. Hojas de Datos

8.4. Código de Programa

```
.include "m88pdef.inc"


.def aux = r16
.def aux2= r17
.def aux3= r18
.def aux4= r19
.def aux5= r21
.def com_reg = r20

.equ slv_r = 0b11010001
.equ slv_w = 0b11010000
.equ xacch = 59
.equ xaccl = 60
.equ yacch = 61
.equ yaccl = 62
.equ zacch = 63
.equ zaccl = 64
.equ xgyroh = 67
.equ xgyrol = 68
.equ ygyroh = 69
.equ ygyrol = 70
.equ zgyroh = 71
.equ zgyrol = 72


.dseg
.org    SRAM_START
;variables gyro/acc
gyro_xh:    .byte 1
gyro_xl:    .byte 1
gyro_yh:    .byte 1
gyro_yl:    .byte 1
gyro_zh:    .byte 1
gyro_zl:    .byte 1
acc_xh: .byte 1
acc_xl: .byte 1
acc_yh: .byte 1
acc_yl: .byte 1
acc_zh: .byte 1
acc_zl: .byte 1
roll_h: .byte 1
roll_l: .byte 1
pitch_l: .byte 1
pitch_h: .byte 1
roll_acc: .byte 1
pitch_acc: .byte 1
roll: .byte 1
pitch: .byte 1
PWM_value: .byte 1

;calibracion del gyro/acc
off_gyro_xh:    .byte 1
off_gyro_xl:    .byte 1
off_gyro_yh:    .byte 1
off_gyro_yl:    .byte 1
off_gyro_zh:    .byte 1
off_gyro_zl:    .byte 1
off_acc_xh: .byte 1
off_acc_xl: .byte 1
off_acc_yh: .byte 1
off_acc_yl: .byte 1
off_acc_zh: .byte 1
off_acc_zl: .byte 1
```

```
FUNCTION:   .byte 1
GRAD_STR:   .byte 3
GRAD_SGN:   .byte 1
ADC_value_0: .byte 1
ADC_value_1: .byte 1



.cseg
.org 0x0000
    rjmp    inicio
.org 0x00E ;timer0 compa interrupt
    rjmp    timer_interrupt

.org INT_VECTORS_SIZE

tabla_atan:
.db 0,1,2,3,4,4,5,6,7,8,9,10,11,11,12,13,14,15,16,17,17,18,19,20,21,21,22,23,24,24,25,26,2
7,27,28,29,29,30,31,31,32,33,33,34,35,35,36,36,37,37,38,39,39,40,40,41,41,42,42,43,43,44,4
4,45,45,45

inicio:

; Inicializar Stack
    ldi  aux, low(RAMEND)
    out  SPL, aux
    ldi  aux, high(RAMEND)
    out  SPH, aux

    ldi     aux,1
    sts     FUNCTION,aux

    ; Inicializar Perisfericos
    rcall configure_ports
    rcall configure_LCD
    rcall PRINT_CAL
    rcall configure_I2C
    rcall configure_UART
    rcall configure_timer
    rcall configure_PWM
    rcall configure_ADC

    ; Inicializar MPU-6050
    rcall init_mpu
    rcall calibrar_mpu
    ;limpio LCD
    rcall CLEAR_LCD

    ; Inicializar variables
    ldi aux,0
    sts roll_h,aux
    sts roll_l,aux
    sts pitch_h,aux
    sts pitch_l,aux
    sts pitch_acc,aux
    sts roll_acc,aux
    sts roll,aux
    sts pitch,aux

    sei ;habilito interrupciones

    rcall PRINT_MSG
    rcall MSG_FUN
```

```
main_loop:


end:
    rcall   POL
    lds     aux,FUNCTION
    cpi     aux,1
    breq    FUN_1
    cpi     aux,2
    breq    FUN_2
    rjmp    end


FUN_1:
    rcall read_gyro
    rcall read_acc
    rcall angulo_acc
    rcall escalo_pr
    rcall PWM_CHANGE_0
    rcall PWM_CHANGE_1
    brtc nomando
    lds com_reg,pitch
    rcall write_uart
    clt
nomando:
    rjmp end

FUN_2:
    push aux3
    ldi aux2,0
    rcall read_ADC
    lds aux,ADC_value_0
    lsr aux
    ldi aux3,57
    add aux,aux3
    rcall SEND_PWM_0
    lds aux,ADC_value_1
    lsr aux
    ldi aux3,57
    add aux,aux3
    rcall SEND_PWM_1
    pop aux3
    rjmp end



POL:
    push    aux
    sbis    PINB,0
    rjmp    INC_FUN
END_POL:
    pop aux
    ret


INC_FUN:
    sbis    PINB,0
    rjmp    INC_FUN
    lds     aux,FUNCTION
    inc     aux
    cpi     aux,3
    breq    RST_FUN
END_INC_FUN:
```

```
    sts     FUNCTION,aux
    rcall   MSG_FUN
    rjmp    END_POL



RST_FUN:
    ldi     aux,1
    rjmp    END_INC_FUN


.include "MULT.inc"
.include "DELAYS.inc"
.include "HDLR_TIMER.inc"
.include "PWM.inc"
.include "SUM_S16_S8.inc"
.include "GYRO.inc"
.include "PORTS.inc"
.include "MPU.inc"
.include "I2C.inc"
.include "TIMER0.inc"
.include "UART.inc"
.include "LCD.inc"
.include "PWM_TO_LCD.inc"
.include "INT_TO_STR.inc"
.include "ADC.inc"
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;BEGIN ADC;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
configure_ADC:
    push    aux
    ldi     aux,(1 << REFS0)|(1 << ADLAR)
    sts     ADMUX, aux
    ldi     aux,(1 << ADEN)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);hablita ADC, single convertio
n, disable autotrigger, disable ADC interrupt
    sts     ADCSRA, aux
    pop     aux
    ret
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
read_ADC:

    push    aux

    lds     aux,ADMUX
    andi    aux,0xF8 ;enable ADC0
    sts     ADMUX,aux

    lds     aux, ADCSRA
    ori     aux, (1 << ADSC) ;enable single convertion
    sts     ADCSRA,aux

loop_read_ADC_0:

    lds     aux,ADCSRA
    sbrs    aux,ADIF ;skip si termino la conversion
    rjmp    loop_read_ADC_0

    lds     aux,ADCSRA
    ori     aux,(1<<ADIF)|(1<<ADEN) ;limpio ADIF seteandolo
    sts     ADCSRA, aux

    lds     aux,ADCL
    lds     aux,ADCH
    sts     ADC_value_0,aux ; guardo el resultado en registro RAM

    lds     aux,ADMUX
    andi    aux,0xF8
    ori     aux,(1<<MUX0);habilito ADC1
    sts     ADMUX,aux

    lds     aux, ADCSRA
    ori     aux, (1 << ADSC) ;habilito single convertion
    sts     ADCSRA,aux

loop_read_ADC_1:

    lds     aux,ADCSRA
    sbrs    aux,ADIF        ;skip si termino la conversion
    rjmp    loop_read_ADC_1

    lds     aux,ADCSRA
    ori     aux,(1<<ADIF)|(1<<ADEN);limpio flag
    sts     ADCSRA, aux

    lds     aux,ADCL
    lds     aux,ADCH
    sts     ADC_value_1,aux ;guardo resultado en registro RAM

    pop     aux
    ret
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;BEGIN ADC;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
configure_ADC:
    push    aux
    ldi     aux,(1 << REFS0)|(1 << ADLAR)
    sts     ADMUX, aux
    ldi     aux,(1 << ADEN)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);hablita ADC, single convertio
n, disable autotrigger, disable ADC interrupt
    sts     ADCSRA, aux
    pop     aux
    ret
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
read_ADC:

    push    aux

    lds     aux,ADMUX
    andi    aux,0xF8 ;enable ADC0
    sts     ADMUX,aux

    lds     aux, ADCSRA
    ori     aux, (1 << ADSC) ;enable single convertion
    sts     ADCSRA,aux

loop_read_ADC_0:

    lds     aux,ADCSRA
    sbrs    aux,ADIF ;skip si termino la conversion
    rjmp    loop_read_ADC_0

    lds     aux,ADCSRA
    ori     aux,(1<<ADIF)|(1<<ADEN) ;limpio ADIF seteandolo
    sts     ADCSRA, aux

    lds     aux,ADCL
    lds     aux,ADCH
    sts     ADC_value_0,aux ; guardo el resultado en registro RAM

    lds     aux,ADMUX
    andi    aux,0xF8
    ori     aux,(1<<MUX0);habilito ADC1
    sts     ADMUX,aux

    lds     aux, ADCSRA
    ori     aux, (1 << ADSC) ;habilito single convertion
    sts     ADCSRA,aux

loop_read_ADC_1:

    lds     aux,ADCSRA
    sbrs    aux,ADIF        ;skip si termino la conversion
    rjmp    loop_read_ADC_1

    lds     aux,ADCSRA
    ori     aux,(1<<ADIF)|(1<<ADEN);limpio flag
    sts     ADCSRA, aux

    lds     aux,ADCL
    lds     aux,ADCH
    sts     ADC_value_1,aux ;guardo resultado en registro RAM

    pop     aux
    ret
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;DELAYS;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
delay_1s:
    push aux
    eor     aux, aux
loop_retardo_1s:
    rcall   delay_100ms
    inc     aux
    cpi     aux,10
    brne    loop_retardo_1s
    pop aux
    ret
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
delay_100ms:
    push aux
    eor     aux, aux
loop_retardo_100ms:
    rcall   delay_1ms
    inc     aux
    cpi     aux,100
    brne    loop_retardo_100ms
    pop aux
    ret
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
delay_10ms:
    push aux
    eor     aux, aux
loop_retardo_10ms:
    rcall   delay_1ms
    inc     aux
    cpi     aux,10
    brne    loop_retardo_10ms
    pop aux
    ret
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
delay_1ms:
    push aux
    eor     aux, aux
loop_retardo_1ms:
    rcall   delay_10us
    inc     aux
    cpi     aux,100
    brne    loop_retardo_1ms
    pop aux
    ret
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
delay_10us:
    push aux
    eor     aux, aux
loop_retardo_10us:
    inc     aux
    cpi     aux,180
    brne    loop_retardo_10us
    pop aux
    ret

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
delay_50000:
    push    r18
    push    r19

    ldi     r18, 65
    ldi     r19, 239
L1: dec     r19
    brne    L1
```

```
    dec      r18
    brne    L1

    pop     r19
    pop     r18
    ret
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
delay_50:

    push    aux

    ldi     aux, 50
L2: dec     aux
    brne    L2

    pop     aux
    ret
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
;;;;;;;;;;;;;;;;;;;;;AJUSTE OFFSET ACCEL/GYRO;;;;;;;;;;;;;;;;;;
calibrar_mpu:
    push    aux
    push    aux2
    push    aux3
    push    aux4
    push    aux5

    clr     aux
    clr     aux2
    clr     aux4
    clr     aux5
    ldi     aux3,16 ;Contador de muestras
sum_xacc:
    ldi com_reg,xacch
    rcall read_reg
    subi    com_reg,128
    add     aux,com_reg
    clr     com_reg
    adc     aux2,com_reg
    ldi com_reg,xaccl
    rcall read_reg
    add     aux4,com_reg
    clr     com_reg
    adc     aux5,com_reg
    rcall delay_10ms
    dec     aux3
    brne    sum_xacc
    ldi     aux3,4
div_xacc:
    lsr     aux2
    ror     aux
    lsr     aux5
    ror     aux4
    dec     aux3
    brne    div_xacc
    subi aux,128
    sts off_acc_xh, aux
    sts off_acc_xl, aux4
    clr     aux
    clr     aux2
    clr     aux4
    clr     aux5
    ldi     aux3,16 ;Contador de muestras
sum_yacc:
    ldi com_reg,yacch
    rcall read_reg
    subi    com_reg,128
    add     aux,com_reg
    clr     com_reg
    adc     aux2,com_reg
    ldi com_reg,yaccl
    rcall read_reg
    add     aux4,com_reg
    clr     com_reg
    adc     aux5,com_reg
    rcall delay_10ms
    dec     aux3
    brne    sum_yacc
    ldi     aux3,4
div_yacc:
    lsr     aux2
    ror     aux
```

```
    lsr     aux5
    ror     aux4
    dec     aux3
    brne    div_yacc
    subi aux,128
    sts off_acc_yh, aux
    sts off_acc_yl, aux4

    clr     aux
    clr     aux2
    clr     aux4
    clr     aux5
    ldi     aux3,16 ;Contador de muestras
sum_zacc:
    ldi com_reg,zacch
    rcall read_reg
    subi    com_reg,128
    add     aux,com_reg
    clr     com_reg
    adc     aux2,com_reg
    ldi com_reg,zaccl
    rcall read_reg
    add     aux4,com_reg
    clr     com_reg
    adc     aux5,com_reg
    rcall delay_10ms
    dec     aux3
    brne    sum_zacc
    ldi     aux3,4
div_zacc:
    lsr     aux2
    ror     aux
    lsr     aux5
    ror     aux4
    dec     aux3
    brne    div_zacc
    subi aux,192 ;128(para expresar como signed)+64 porque el eje z lee 1G en pos. neutral
    sts off_acc_zh, aux
    sts off_acc_zl, aux4

    clr     aux
    clr     aux2
    clr     aux4
    clr     aux5
    ldi     aux3,16 ;Contador de muestras
sum_xgyro:
    ldi com_reg,xgyroh
    rcall read_reg
    subi    com_reg,128
    add     aux,com_reg
    clr     com_reg
    adc     aux2,com_reg
    ldi com_reg,xgyrol
    rcall read_reg
    add     aux4,com_reg
    clr     com_reg
    adc     aux5,com_reg
    rcall delay_10ms
    dec     aux3
    brne    sum_xgyro
    ldi     aux3,4
div_xgyro:
    lsr     aux2
    ror     aux
```

```
    lsr     aux5
    ror     aux4
    dec     aux3
    brne    div_xgyro
    subi aux,128
    sts off_gyro_xh, aux
    sts off_gyro_xl, aux4
    clr     aux
    clr     aux2
    clr     aux4
    clr     aux5
    ldi     aux3,16 ;Contador de muestras
sum_ygyro:
    ldi com_reg,ygyroh
    rcall read_reg
    subi    com_reg,128
    add     aux,com_reg
    clr     com_reg
    adc     aux2,com_reg
    ldi com_reg,ygyrol
    rcall read_reg
    add     aux4,com_reg
    clr     com_reg
    adc     aux5,com_reg
    rcall delay_10ms
    dec     aux3
    brne    sum_ygyro
    ldi     aux3,4
div_ygyro:
    lsr     aux2
    ror     aux
    lsr     aux5
    ror     aux4
    dec     aux3
    brne    div_ygyro
    subi aux,128
    sts off_gyro_yh, aux
    sts off_gyro_yl, aux4
    clr     aux
    clr     aux2
    clr     aux4
    clr     aux5
    ldi     aux3,16 ;Contador de muestras
sum_zgyro:
    ldi com_reg,zgyroh
    rcall read_reg
    subi    com_reg,128
    add     aux,com_reg
    clr     com_reg
    adc     aux2,com_reg
    ldi com_reg,zgyrol
    rcall read_reg
    add     aux4,com_reg
    clr     com_reg
    adc     aux5,com_reg
    rcall delay_10ms
    dec     aux3
    brne    sum_zgyro
    ldi     aux3,4
div_zgyro:
    lsr     aux2
    ror     aux
    lsr     aux5
    ror     aux4
```

```
    dec     aux3
    brne    div_zgyro
    subi aux,128
    sts off_gyro_zh, aux
    sts off_gyro_zl, aux4
    pop     aux5
    pop     aux4
    pop     aux3
    pop     aux2
    pop     aux
    ret
;;;;;;;;;;;;;;;;;;;LEER GYRO/ACC;;;;;;;;;;;;;;;;;;
read_gyro:
    push com_reg
    push aux
    ldi com_reg,xgyroh
    rcall read_reg
    lds aux,off_gyro_xh
    sub com_reg,aux
    sts gyro_xh,com_reg
    ldi com_reg,ygyroh
    rcall read_reg
    lds aux,off_gyro_yh
    sub com_reg,aux
    sts gyro_yh,com_reg
    pop aux
    pop com_reg
    ret
read_acc:
    push com_reg
    push aux
    ldi com_reg,xacch
    rcall read_reg
    lds aux,off_acc_xh
    sub com_reg,aux
    sts acc_xh,com_reg
    ldi com_reg,yacch
    rcall read_reg
    lds aux,off_acc_yh
    sub com_reg,aux
    sts acc_yh,com_reg
    ldi com_reg,zacch
    rcall read_reg
    lds aux,off_acc_zh
    sub com_reg,aux
    sts acc_zh,com_reg
    pop aux
    pop com_reg
    ret

;;;;;;;;;;;;;;;;;;;;;OBTENCION ANGULO ACC;;;;;;;;;;;;;;;;;;;;;;
angulo_acc:
    push aux
    push aux2
    push aux3
    lds aux3,acc_zh ;z siempre pos para el rango usado
    lds aux,acc_xh
    rcall calculo_atan
    sts roll_acc,aux
    lds aux3,acc_zh ;z siempre pos para el rango usado
    lds aux,acc_yh
    rcall calculo_atan
    sts pitch_acc,aux
    pop aux3
```

```
    pop aux2
    pop aux
    ret

;;;;;;;;;;;;;Calculo atan por tabla;;;;;;;;;;;;;;;;
calculo_atan:
    push aux4
    clr aux4
    sbrs aux,7 ;salteo si es negativo
    rjmp incl_pos
    neg aux
    ldi aux4,1 ;seteo flag negativo
incl_pos:
    cp aux,aux3
    breq roll45 ;si son iguales el angulo es 45
    brlo xmenorz
    mov aux2,aux    ;si angulo>45
    mov aux,aux3    ;swapeo reg (ver prop. tan)
    mov aux3,aux2
    ori aux4,2  ;seteo flag angulo>45
xmenorz:
    ldi aux2,64
    mul aux,aux2
    mov aux,r0
    mov aux2,r1
    rcall div_unsigned16; res: r1:r0 (solo me importa parte baja)
    ldi zl,low(tabla_atan<<1)
    ldi zh,high(tabla_atan<<1)
    add zl,r0
    clr r0
    adc zh,r0
    lpm aux,z ;extraigo atan de la tabla
    rjmp fin_tan
roll45:
    ldi aux,45
fin_tan:
    sbrs aux4,1
    rjmp no_swap
    ldi aux2,90 ;si hubo swap, hago 90-angulo
    sub aux2,aux
    mov aux,aux2
no_swap:
    sbrc aux4,0
    neg aux ;niego aux si el angulo es negativo
    pop aux4
    ret
;;;;;;;;;;;;Escalar angulo en unidades del gyro a grados;;;;;;;;;;;;;;;;
escalo_pr:
;t_muestreo=8ms
;Escalo pitch(°)=pitch*t_muestreo*(250/128)=pitch*(4/256)
;multiplico por 4 y para dividir por 256, simplemente
;utilizo solo los 8bit mas significativos
;Nota: esto es mas rapido que dividir por 64 desplazando
;Nota2: no hay problema de overflow al multiplicar, ya
;que el rango de valores del gyro es mucho menor que
;2^14
    push aux3
    push aux2
    push aux

    lds aux3,pitch_h
    lds aux2,pitch_l
    ldi aux,4
    rcall mul_signed16
```

```
    sts pitch,aux3

;Idem pitch, escalo roll
    lds aux3,roll_h
    lds aux2,roll_l
    ldi aux,4
    rcall mul_signed16
    sts roll,aux3
    pop aux
    pop aux2
    pop aux3
    ret

;;;;;;;;;;;;;;Div unsigned 16 / unsigned 8;;;;;;;;;;;;;;;;
div_unsigned16: ;[r1,r0]<-[aux2,aux]/aux3
    push aux4
    clr aux4
    clr r1
    clr r0
    inc r0

div8a:
    clc
    rol aux
    rol aux2
    rol aux4
    brcs div8b
    cp aux4,aux3
    brcs div8c
div8b:
    sub aux4,aux3
    sec
    rjmp div8d
div8c:
    clc
div8d:
    rol r0
    rol r1
    brcc div8a
    pop aux4
    ret
```

```
;;;;;;;;;;;;;;;;;;Handler interrupcion timer;;;;;;;;;;;;;;;;
timer_interrupt:
    push aux
    push aux2
    push aux3
    push aux4
    push aux5
    push r0
    push r1
    ldi aux,0
    out TCNT0,aux ;reseteo contador timer
    in aux5,sreg

;hallo pitch integrando gyro x
    lds aux,gyro_xh
    lds aux2,pitch_l
    lds aux3,pitch_h
    rcall suma_s16s8


;corrijo deriva con acelerometro
;filtro complementario: promedio ponderado (pgyro*7+pacc)/8
    ldi aux,7
    rcall mul_signed16  ;pgyro*7
    lds aux, pitch_acc
    neg aux ;mismo sentido que gyro
    ldi aux4,64
    muls aux,aux4 ;escalo a "angulo" de gyro
    add aux2,r0 ;pgyro*7+pacc(escalado)
    adc aux3,r1
    ldi aux4,3
div_pitch:  ;divido por 8
    asr aux3    ;asr porque es signed
    ror aux2
    dec aux4
    brne div_pitch

    sts pitch_h,aux3
    sts pitch_l,aux2

;hallo roll integrando gyro y
    lds aux,gyro_yh
    lds aux2,roll_l
    lds aux3,roll_h
    rcall suma_s16s8

;idem pitch para roll
    ldi aux,7
    rcall mul_signed16
    lds aux, roll_acc
    neg aux
    ldi aux4,64
    muls aux,aux4
    add aux2,r0
    adc aux3,r1
    ldi aux4,3
div_roll:
    asr aux3
    ror aux2
    dec aux4
    brne div_roll

    sts roll_h,aux3
    sts roll_l,aux2
```

```
    out sreg,aux5
    set
    pop r1
    pop r0
    pop aux5
    pop aux4
    pop aux3
    pop aux2
    pop aux
    reti
```

```
;;;;;;;;;;;;;;;;;;;;;;;;CONFIG I2C;;;;;;;;;;;;;;;;;;;;;;;;;
configure_I2C:
    push aux
    ldi aux,0
    sts TWSR, aux
    ldi aux,15 ;bitrate 400KHz
    sts TWBR, aux
    ldi aux, (1<<TWEN)
    sts TWCR, aux
    pop aux
    ret
start_i2c:
    push aux
    ldi aux,(1<<TWINT)|(1<<TWSTA)|(1<<TWEN)
    sts TWCR, aux
    waitstart:
    lds aux,TWCR
    sbrs aux,TWINT
    rjmp waitstart
    pop aux
    ret
stop_i2c:
    push aux
    ldi aux,(1<<TWINT)|(1<<TWEN)|(1<<TWSTO)
    sts TWCR, aux
    pop aux
    ret

;;;;;;;;;;;;;;;;;;;;;;;;R/W I2C;;;;;;;;;;;;;;;;;;;;;;;;;
write_i2c:
    push aux
    sts  TWDR,com_reg
    ldi aux,(1<<TWINT)|(1<<TWEN)
    sts TWCR,aux
    waitwi2c:
    lds aux,TWCR
    sbrs aux,TWINT
    rjmp waitwi2c
    pop aux
    ret
read_i2c:
    push aux
    ldi aux,(1<<TWINT)|(1<<TWEN)
    sts TWCR,aux
    waitri2c:
    lds aux,TWCR
    sbrs aux,TWINT
    rjmp waitri2c
    lds com_reg,TWDR
    pop aux
    ret
```

```
INT_TO_STR:
    push    aux
    push    r18
    push    r19
    push    r20

    clr     r18 ;limpio registros
    clr     r19
    clr     r20

    push    r21 ;auxiliar para el 100 y el 10 para

    ldi     r21,100
loop_division_100:

    sub     aux,r21
    brlo    fin_100
    inc     r18             ;centena
    rjmp    loop_division_100

fin_100:
    add     aux,r21
    ldi     r21,10

loop_division_10:
    sub     aux,r21
    brlo    fin_10
    inc     r19             ;decenas
    rjmp    loop_division_10

fin_10:
    add     aux,r21
    mov     r20,aux         ;unidades
    ldi     aux,48          ;los shifteo ascii
    ldi     xl,low(GRAD_STR)    ;guardo las variables en RAM
    ldi     xh,high(GRAD_STR)
    add     r18,aux
    st      x+,r18
    add     r19,aux
    st      x+,r19
    add     r20,aux
    st      x,r20


    pop     r21
    pop     r20
    pop     r19
    pop     r18
    pop     aux
    ret
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;BEGIN LCD;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

configure_LCD:
    push    aux

    rcall   delay_50000
    rcall   delay_50000
    rcall   delay_50000

    ldi     aux,0x30
    rcall   write_LCD
    rcall   delay_50000
    ldi     aux,0x30
    rcall   write_LCD
    rcall   delay_50000
    ldi     aux,0x30
    rcall   write_LCD
    rcall   delay_50000

;INICIALIZACION LCD EN MODO 4 BITS

    ldi     aux,0x20
    rcall   write_LCD
    rcall   delay_50000

    ldi     aux,0x28
    rcall   send_LCD_CMND
    rcall   delay_50000 ;

    ldi     aux,0x08
    rcall   send_LCD_CMND
    rcall   delay_50000

    ldi     aux,0x01
    rcall   send_LCD_CMND
    rcall   delay_50000

    ldi     aux,0x0C
    rcall   send_LCD_CMND
    rcall   delay_50000

    ldi     aux,0x06
    rcall   send_LCD_CMND
    rcall   delay_50000

    pop     aux
    ret

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
write_LCD:

    push    aux2

    in      aux2,PORTD
    andi    aux2,0x0F
    or      aux2,aux
    out     PORTD,aux2

    sbi     PORTD,3         ;SET E
    cbi     PORTD,3         ;CLR E

    rcall   delay_50

    pop     aux2
```

```
    ret
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

send_LCD_CMND:

    cbi     PORTD,2

    push    aux
    andi    aux,0xF0
    rcall   write_LCD
    pop     aux

    swap    aux
    andi    aux,0xF0
    rcall   write_LCD

    cbi     PORTD,2
    ret
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
send_LCD_DATA:

    sbi     PORTD,2

    push    aux
    andi    aux,0xF0
    rcall   write_LCD
    pop     aux

    swap    aux
    andi    aux,0xF0
    rcall   write_LCD

    cbi     PORTD,2
    ret
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

PRINT_CAL:                   ;IMPRIMIR MENSAJE CALIBRANDO SENSOR EN LCD
    push    aux
    ldi     aux,0x83
    rcall   send_LCD_CMND
    rcall   delay_50000

    ldi     zl,LOW(msg_cal_1<<1)     ;EN ROM
    ldi     zh,HIGH(msg_cal_1<<1)
LOOP_PRINT_MSG_CAL_1:
    lpm     aux,z+
    cpi     aux,0
    breq    fin_PRINT_MSG_CAL_1
    rcall   SEND_LCD_DATA
    rcall   delay_50000
    rjmp    LOOP_PRINT_MSG_CAL_1

FIN_PRINT_MSG_CAL_1:

    ldi     aux,0xC5
    rcall   send_LCD_CMND
    rcall   delay_50000


    ldi     zl,LOW(msg_cal_2<<1)     ;EN ROM
    ldi     zh,HIGH(msg_cal_2<<1)
LOOP_PRINT_MSG_CAL_2:
    rcall   delay_50000
    lpm     aux,z+
```

```
    cpi     aux,0
    breq    fin_PRINT_MSG_CAL_2
    rcall   SEND_LCD_DATA
    rjmp    LOOP_PRINT_MSG_CAL_2

FIN_PRINT_MSG_CAL_2:

    pop     aux
    ret
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

MSG_FUN:
    push    aux
    lds     aux,FUNCTION
    cpi     aux,1
    breq    FUN_1_MSG
    cpi     aux,2
    breq    FUN_2_MSG
END_MSG_FUN:
    pop     aux
    ret

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

FUN_1_MSG:                                      ;MENSAJE FUNCION

    ldi     aux,0x8D
    rcall   send_LCD_CMND
    rcall   delay_50000

    ldi     zl,LOW(msg_fun_1<<1)
    ldi     zh,HIGH(msg_fun_1<<1)
LOOP_FUN_MSG_1:
    rcall   delay_50000
    lpm     aux,z+
    cpi     aux,0
    breq    END_MSG_FUN
    rcall   SEND_LCD_DATA
    rjmp    LOOP_FUN_MSG_1

FUN_2_MSG:

    ldi     aux,0x8D
    rcall   send_LCD_CMND
    rcall   delay_50000

    ldi     zl,LOW(msg_fun_2<<1)
    ldi     zh,HIGH(msg_fun_2<<1)
LOOP_FUN_MSG_2:
    rcall   delay_50000
    lpm     aux,z+
    cpi     aux,0
    breq    END_MSG_FUN
    rcall   SEND_LCD_DATA
    rjmp    LOOP_FUN_MSG_1


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

PRINT_MSG:                                      ;MENSAJE SERVO 1: SERVO 2:
    push    aux

    ldi     aux,0x80
    rcall   send_LCD_CMND
```

```
    rcall   delay_50000

    ldi     zl,LOW(msg_1<<1)
    ldi     zh,HIGH(msg_1<<1)
LOOP_PRINT_MSG_1:
    rcall   delay_50000
    lpm     aux,z+
    cpi     aux,0
    breq    fin_PRINT_MSG_1
    rcall   SEND_LCD_DATA
    rjmp    LOOP_PRINT_MSG_1

FIN_PRINT_MSG_1:

    ldi     aux,0xC0
    rcall   send_LCD_CMND
    rcall   delay_50000

    ldi     zl,LOW(msg_2<<1)
    ldi     zh,HIGH(msg_2<<1)
LOOP_PRINT_MSG_2:
    rcall   delay_50000
    lpm     aux,z+
    cpi     aux,0
    breq    fin_PRINT_MSG_2
    rcall   SEND_LCD_DATA
    rjmp    LOOP_PRINT_MSG_2

FIN_PRINT_MSG_2:

    pop     aux
    ret

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


CLEAR_LCD:
    push    aux
    ldi     aux,0x01
    rcall   send_LCD_CMND
    rcall   delay_50000
    pop     aux
    ret
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
LCD_POS:                                            ;IMPRIME POSICION ANGULAR

    push    aux
    ldi     xl,low(GRAD_SGN)
    ldi     xh,high(GRAD_SGN)
    ld      aux,x
    rcall   SEND_LCD_DATA
    rcall   delay_50000

    ldi     xl,low(GRAD_STR)
    ldi     xh,high(GRAD_STR)
    ld      aux,x+
    rcall   SEND_LCD_DATA
    rcall   delay_50000
    ld      aux,x+
    rcall   SEND_LCD_DATA
    rcall   delay_50000
    ld      aux,x
    rcall   SEND_LCD_DATA
    rcall   delay_50000
```

```
    pop     aux
    ret
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

msg_1:  .db "SERVO 1:",0
msg_2:  .db "SERVO 2:",0
msg_cal_1:  .db "CALIBRANDO",0
msg_cal_2:  .db "SENSOR",0
msg_fun_1:  .db "MPU",0
msg_fun_2:  .db "ADC",0
```

```
;;;;;;;;;;;;INIT MPU;;;;;;;;;;;;;;;;;;;;;;;;
init_mpu:
    rcall start_i2c
    ldi com_reg,slv_w
    rcall write_i2c
    ldi com_reg,0x6b    ;power magnament
    rcall write_i2c
    ldi com_reg,0b00000000
    rcall write_i2c
    rcall stop_i2c

    rcall start_i2c
    ldi com_reg,slv_w
    rcall write_i2c
    ldi com_reg,26  ;configuro el pasa-bajo
    rcall write_i2c
    ldi com_reg,2
    rcall write_i2c
    rcall stop_i2c

    rcall start_i2c
    ldi com_reg,slv_w
    rcall write_i2c
    ldi com_reg,27  ;configuro gyro
    rcall write_i2c
    ldi com_reg,0
    rcall write_i2c
    rcall stop_i2c

    rcall start_i2c
    ldi com_reg,slv_w
    rcall write_i2c
    ldi com_reg,28  ;configuro acc
    rcall write_i2c
    ldi com_reg,0
    rcall write_i2c
    rcall stop_i2c

    cbi PORTC,3
    ldi com_reg,0x75
    rcall read_reg
    cpi com_reg,0x68
    brne no_com
    sbi PORTC,3 ;led testigo comunicacion
    no_com:

    ret



;;;;;;;;;;;;;;;;;;;;READ MPU;;;;;;;;;;;;;;;;;;;;;;;;
read_reg:
    push com_reg
    rcall start_i2c
    ldi com_reg,slv_w
    rcall write_i2c
    pop com_reg
    rcall write_i2c
    rcall start_i2c
    ldi com_reg,slv_r
    rcall write_i2c
    rcall read_i2c
    rcall stop_i2c
    ret
```

```
MULT:
    push    r0  ;MULTIPLICA 2 BYTES Y SOLO ME QUEDO CON PARTE ALTA
    push    r1
loop_mul_45:
    mul     aux,aux3
    mov     aux,r1
    pop     r1
    pop     r0
    ret
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;CONFIG PORTS;;;;;;;;;;;;;;;;;;;;;;;;
configure_ports:
    push    aux

    ldi     aux,(1 << DDB1) | (1 << DDB2)
    out     DDRB,aux

    ldi     aux,(0 << DDC1) | (1 << DDC3)
    out     DDRC,aux

    ldi     aux,(1 << DDD4) | (1 << DDD5) | (1 << DDD6) | (1 << DDD7) ;LCD data_PIN
    ori     aux,(1 << DDD2) ;RS
    ori     aux,(1 << DDD3) ;E
    out     DDRD,aux

    pop     aux
    ret
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;Actualizo PWM;;;;;;;;;;;;;;
PWM_change_0:
    push    aux2
    push    aux
    push    aux3
    lds     aux,pitch           ;ESCALAR PITCH A PWM
    ldi     aux2,90
    add     aux,aux2
    ldi     aux3,180
    rcall   MULT
    ldi     aux2,58
    add     aux,aux2            ;ESCALAR PITCH A PWM
    ldi     aux2,0
    rcall   SEND_PWM_0
    pop     aux3
    pop     aux
    pop     aux2
    ret

PWM_change_1:
    push    aux2
    push    aux
    push    aux3
    lds     aux,roll            ;ESCALAR ROLL A PWM
    ldi     aux2,90
    add     aux,aux2
    ldi     aux3,180
    rcall   MULT
    ldi     aux2,58
    add     aux,aux2            ;ESCALAR ROLL A PWM
    ldi     aux2,0
    rcall   SEND_PWM_1
    pop     aux3
    pop     aux
    pop     aux2
    ret
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
SEND_PWM_0:                 ;ENVIO VALOR A PWM E IMPRIMO EN LCD

    sts     OCR1AH,aux2
    sts     OCR1AL,aux
    rcall   PWM_TO_LCD
    rcall   INT_TO_STR
    ldi     aux,0x88
    rcall   send_LCD_CMND
    rcall   delay_50000
    rcall   LCD_POS
    ret

SEND_PWM_1:

    sts     OCR1BH,aux2
    sts     OCR1BL,aux
    rcall   PWM_TO_LCD
    rcall   INT_TO_STR
    ldi     aux,0xC8
    rcall   send_LCD_CMND
    rcall   delay_50000
    rcall   LCD_POS
    ret

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
configure_PWM:
    push    aux
    ldi     aux,1
    sts     TCNT1H,aux
    sts     TCNT1L,aux
    ldi     aux,0xE8
    sts     ICR1H,aux
    ldi     aux,0x0B
    sts     ICR1L,aux
    ldi     aux,(1<<COM1A1)|(1<<COM1B1)|(1<<WGM11)|(1<<WGM10)
    sts     TCCR1A,aux

    ldi     aux,(1<<WGM12)|(1<<CS22)|(0<<CS21)|(0<<CS20)
    ;si bien es redundante orear 0's, lo hago para que la funcion este lista
    ;para futuras modificaciones
    sts     TCCR1B,aux

    ldi     aux,0               ;valor inicial PWM
    sts     OCR1AH,aux
    ldi     aux,110
    sts     OCR1AL,aux
    ldi     aux,0
    sts     OCR1BH,aux
    ldi     aux,110
    sts     OCR1BL,aux
    pop     aux

    ret


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
PWM_TO_LCD:
                ;TRANSFORMA VALOR DE PWM A STRING PARA IMPIRMIR EN LCD
    push aux3
    subi aux,57         ;ESCALA PWM A 0 180
    ldi aux3,181

    push    r0
    push    r1
    mul     aux,aux3
    rol     r0
    rol     r1
    mov     aux,r1
    pop     r1
    pop     r0

    cpi aux,90
    brlo NEG_PWM_TO_LCD         ;DETECTO SIGNO
    ldi  aux3,0x2B
    sts  GRAD_SGN,aux3
    subi aux,90

END_PWM_TO_LCD:
    pop aux3
    ret


NEG_PWM_TO_LCD:
    ldi aux3,0x2D
    sts GRAD_SGN,aux3
    ldi aux3,90
    sub aux3,aux
    mov aux,aux3
    rjmp END_PWM_TO_LCD
```

```
;;;;;;;;;;;;;;;;;;;;;;Suma signed 16bit + signed8bit;;;;;;;;;;
suma_s16s8: ; [aux3,aux2]<-[aux3,aux2]+aux
    sbrc aux,7 ;salteo si es positivo
    rjmp negativo
    add aux2,aux
    clr aux
    adc aux3,aux
    rjmp listosum
negativo:
    add aux2,aux
    ldi aux,0xff
    adc aux3,aux
listosum:
    ret

;;;;;;;;;;;;;;;;;;;;;;Mul signed 16bit x unsigned 8bit;;;;;;;;;;
mul_signed16:
    mulsu aux3,aux
    mov aux3,r0 ;desprecio parte alta ya que nunca llega
    mul aux2,aux
    mov aux2,r0
    add aux3,r1
    ret
```

```
;;;;;;;;;;;;;;;;;;;;;;;CONFIG TIMER 0;;;;;;;;;;;;;;;;;;;;;;;;;;
;calculo output compare: (frec/preescaler)*t_timer=OCR0A
;nota: seteo a 8ms por conveniencia para escalar los valores
;del gyro y acc
configure_timer:
    push aux
    ldi aux,0
    out TCCR0A, aux
    ldi aux, (1<<CS02)|(1<<CS00) ; preescaler 1024
    out TCCR0B,aux
    ldi aux,144; 144 timer a 8ms
    out OCR0A,aux
    ldi aux, (1<<OCIE0A)
    sts TIMSK0,aux
    ldi aux,0
    out TCNT0,aux
    pop aux
    ret
```

```
;;;;;;;;;;;;;;;;;CONFIG UART;;;;;;;;;;;;;;;;;;;;;
configure_UART:
    push aux
    ldi aux,0
    sts UBRR0H, aux
    ldi aux,119
    sts UBRR0L, aux
    ldi aux, (1<<UCSZ00)|(1<<UCSZ01)
    sts UCSR0C,aux
    ldi aux, (1<<TXEN0)|(1<<RXEN0)
    sts UCSR0B,aux
    ldi aux, 0
    sts UCSR0A, aux
    pop aux
    ret
write_uart:
    push aux
    waitwuart:
    lds aux,UCSR0A
    sbrs aux,UDRE0
    rjmp waitwuart
    sts UDR0,com_reg
    pop aux
    ret
read_uart:
    push aux
    lds aux,UCSR0A
    sbrs aux,RXC0
    rjmp no_data
    lds com_reg,UDR0
    no_data:
    pop aux
    ret
```

HC Serial Bluetooth Products

User Instructional Manual

1 Introduction

HC serial Bluetooth products consist of Bluetooth serial interface module and Bluetooth adapter, such as:

(1) Bluetooth serial interface module:

Industrial level: HC-03, HC-04(HC-04-M, HC-04-S)

Civil level: HC-05, HC-06(HC-06-M, HC-06-S)

HC-05-D, HC-06-D (with baseboard, for test and evaluation)

(2) Bluetooth adapter:

HC-M4

HC-M6

This document mainly introduces Bluetooth serial module. Bluetooth serial module is used for converting serial port to Bluetooth. These modules have two modes: master and slaver device. The device named after even number is defined to be master or slaver when out of factory and can’t be changed to the other mode. But for the device named after odd number, users can set the work mode (master or slaver) of the device by AT commands.

HC-04 specifically includes:

Master device: HC-04-M, M=master

Slave device: HC-04-S, S=slaver

The default situation of HC-04 is slave mode. If you need master mode, please state it clearly or place an order for HC-O4-M directly.The naming rule of HC-06 is same.

When HC-03 and HC-05 are out of factory, one part of parameters are set for activating the device. The work mode is not set, since user can set the mode of HC-03, HC-05 as they want.

The main function of Bluetooth serial module is replacing the serial port line, such as:

1. There are two MCUs want to communicate with each other. One connects to Bluetooth master device while the other one connects to slave device. Their connection can be built once the pair is made. This Bluetooth connection is equivalently liked to a serial port line connection including RXD, TXD

signals. And they can use the Bluetooth serial module to communicate with each other.

2. When MCU has Bluetooth salve module, it can communicate with Bluetooth adapter of computers and smart phones. Then there is a virtual communicable serial port line between MCU and computer or smart phone.

3. The Bluetooth devices in the market mostly are salve devices, such as Bluetooth printer, Bluetooth GPS. So, we can use master module to make pair and communicate with them.

Bluetooth Serial module's operation doesn't need drive, and can communicate with the other Bluetooth device who has the serial. But communication between two Bluetooth modules requires at least two conditions:

(1) The communication must be between master and slave.

(2) The password must be correct.

However, the two conditions are not sufficient conditions. There are also some other conditions basing on different device model. Detailed information is provided in the following chapters.

In the following chapters, we will repeatedly refer to Linvor's (Formerly known as Guangzhou HC Information Technology Co., Ltd.) material and photos.

2 Selection of the Module

The Bluetooth serial module named even number is compatible with each other; The salve module is also compatible with each other. In other word, the function of HC-04 and HC-06, HC-03 and HC-05 are mutually compatible with each other. HC-04 and HC-06 are former version that user can't reset the work mode (master or slave). And only a few AT commands and functions can be used, like reset the name of Bluetooth (only the slaver), reset the password, reset the baud rate and check the version number. The command set of HC-03 and HC-05 are more flexible than HC-04 and HC-06's. Generally, the Bluetooth of HC-03/HC-05 is recommended for the user.

Here are the main factory parameters of HC-05 and HC-06. Pay attention to the differences:

HC-05	HC-06
Master and slave mode can be switched	Master and slave mode can't be switched
Bluetooth name: HC-05	Bluetooth name: linvor
Password:1234	Password:1234

Master role: have no function to remember the last paired salve device. It can be made paired to any slave device. In other words, just set AT+CMODE=1 when out of factory. If you want HC-05 to remember the last paired slave device address like HC-06, you can set AT+CMODE=0 after paired with the other device. Please refer the command set of HC-05 for the details.	Master role: have paired memory to remember last slave device and only make pair with that device unless KEY (PIN26) is triggered by high level. The default connected PIN26 is low level.
Pairing: The master device can not only make pair with the specified Bluetooth address, like cell-phone, computer adapter, slave device, but also can search and make pair with the slave device automatically. Typical method: On some specific conditions, master device and slave device can make pair with each other automatically. (This is the default method.)	Pairing: Master device search and make pair with the slave device automatically. Typical method: On some specific conditions, master and slave device can make pair with each other automatically.
Multi-device communication: There is only point to point communication for modules, but the adapter can communicate with multi-modules.	Multi-device communication: There is only point to point communication for modules, but the adapter can communicate with multi-modules.
AT Mode 1: After power on, it can enter the AT mode by triggering PIN34 with high level. Then the baud rate for setting AT command is equal to the baud rate in communication, for example: 9600. AT mode 2: First set the PIN34 as high level, or while on powering the module set the PIN34 to be high level, the Baud rate used here is 38400 bps. Notice: All AT commands can be operated only	AT Mode: Before paired, it is at the AT mode. After paired it’s at transparent communication.

when the PIN34 is at high level. Only part of the AT commands can be used if PIN34 doesn't keep the high level after entering to the AT mode. Through this kind of designing, set permissions for the module is left to the user's external control circuit, that makes the application of HC-05 is very flexible.	
During the process of communication, the module can enter to AT mode by setting PIN34 to be high level. By releasing PIN34, the module can go back to communication mode in which user can inquire some information dynamically. For example, to inquire the pairing is finished or not.	During the communication mode, the module can't enter to the AT mode.
Default communication baud rate: 9600, 4800-1.3M are settable.	Default communication baud rate: 9600, 1200-1.3M are settable.
KEY: PIN34, for entering to the AT mode.	KEY: PIN26, for master abandons memory.
LED1: PIN31, indicator of Bluetooth mode. Slow flicker (1Hz) represents entering to the AT mode2, while fast flicker(2Hz) represents entering to the AT mode1 or during the communication pairing. Double flicker per second represents pairing is finished, the module is communicable. LED2: PIN32, before pairing is at low level, after the pairing is at high level. The using method of master and slaver's indicator is the same. Notice: The PIN of LED1 and LED2 are connected with LED+.	LED: The flicker frequency of slave device is 102ms. If master device already has the memory of slave device, the flicker frequency during the pairing is 110ms/s. If not, or master has emptied the memory, then the flicker frequency is 750m/s. After pairing, no matter it's a master or slave device, the LED PIN is at high level. Notice: The LED PIN connects to LED+ PIN.
Consumption: During the pairing, the current is	Consumption: During the pairing, the current is

fluctuant in the range of 30-40mA. The mean current is about 25mA. After paring, no matter processing communication or not, the current is 8mA. There is no sleep mode. This parameter is same for all the Bluetooth modules.	fluctuant in the range of 30-40 m. The mean current is about 25mA. After paring, no matter processing communication or not, the current is 8mA. There is no sleep mode. This parameter is same for all the Bluetooth modules.
Reset: PIN11, active if it's input low level. It can be suspended in using.	Reset: PIN11, active if it's input low level. It can be suspended in using.
Level: Civil	Level: Civil

The table above that includes main parameters of two serial modules is a reference for user selection.

HC-03/HC-05 serial product is recommended.

3. Information of Package

The PIN definitions of HC-03, HC-04, HC-05 and HC-06 are kind of different, but the package size is the same: 28mm * 15mm * 2.35mm.

The following figure 1 is a picture of HC-06 and its main PINs. Figure 2 is a picture of HC-05 and its main PINs. Figure 3 is a comparative picture with one coin. Figure 4 is their package size information. When user designs the circuit, you can visit the website of Guangzhou HC Information Technology Co., Ltd. (www.wavesen.com) to download the package library of protle version.

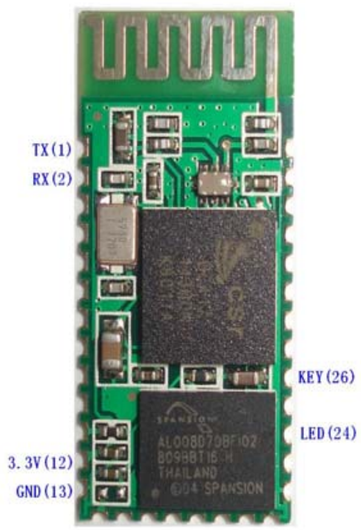


Figure 1 HC-06

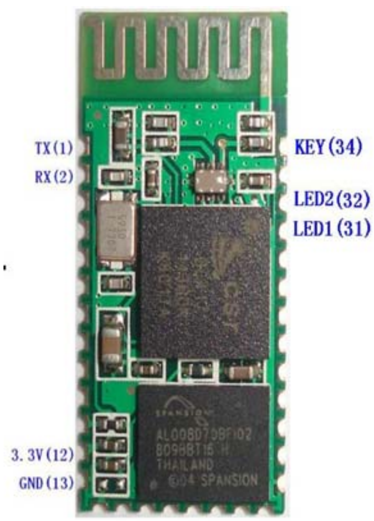


Figure 2 HC-05

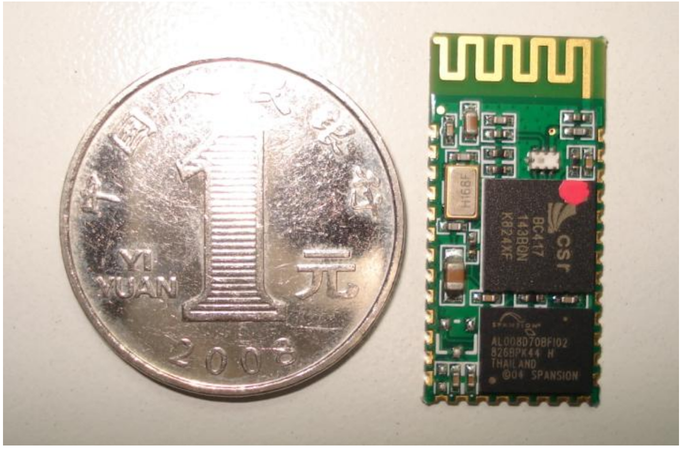
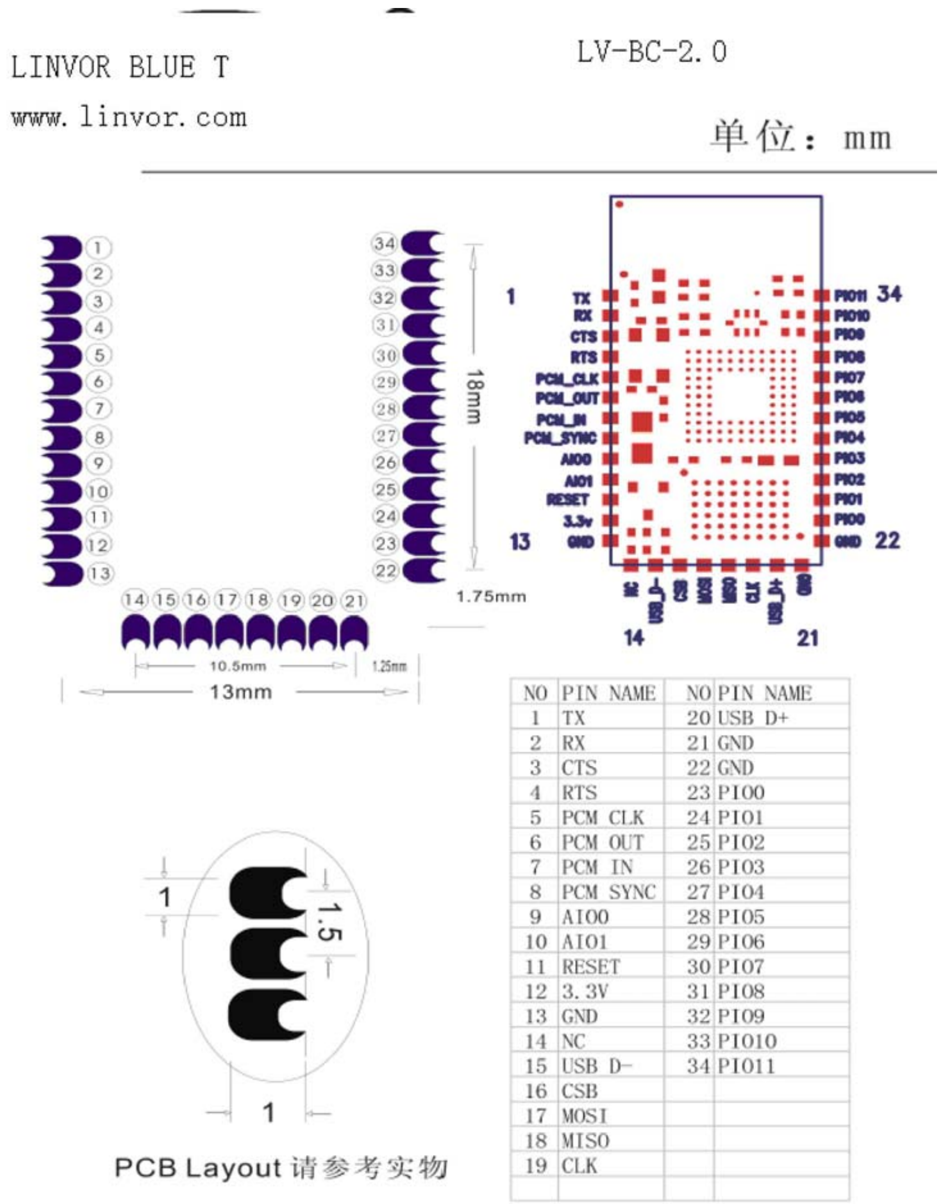


Figure 3 Comparative picture with one coin



NO	PIN NAME	NO	PIN NAME
1	TX	20	USB D+
2	RX	21	GND
3	CTS	22	GND
4	RTS	23	PIO0
5	PCM CLK	24	PIO1
6	PCM OUT	25	PIO2
7	PCM IN	26	PIO3
8	PCM SYNC	27	PIO4
9	AIO0	28	PIO5
10	AIO1	29	PIO6
11	RESET	30	PIO7
12	3.3V	31	PIO8
13	GND	32	PIO9
14	NC	33	PIO10
15	USB D-	34	PIO11
16	CSB		
17	MOSI		
18	MISO		
19	CLK		

Figure 4 Package size information

4. The Using and Testing Method of HC-06 for the First Time

This chapter will introduce the using method of HC-06 in detail. User can test the module according to this chapter when he or she uses the module at the first time.

PINs description:

PIN1	UART_TXD , TTL/CMOS level, UART Data output
PIN2	UART_RXD, TTL/COMS level, s UART Data input
PIN11	RESET, the reset PIN of module, inputting low level can reset the module, when the module is in using, this PIN can connect to air.
PIN12	VCC, voltage supply for logic, the standard voltage is 3.3V, and can work at 3.0-4.2V
PIN13	GND
PIN22	GND
PIN24	LED, working mode indicator Slave device: Before paired, this PIN outputs the period of 102ms square wave. After paired, this PIN outputs high level. Master device: On the condition of having no memory of pairing with a slave device, this PIN outputs the period of 110ms square wave. On the condition of having the memory of pairing with a slave device, this PIN outputs the period of 750ms square wave. After paired, this PIN outputs high level.
PIN26	For master device, this PIN is used for emptying information about pairing. After emptying, master device will search slaver randomly, then remember the address of the new got slave device. In the next power on, master device will only search this address.

(1) The circuit 1 (connect the module to 3.3V serial port of MCU) is showed by figure 5.

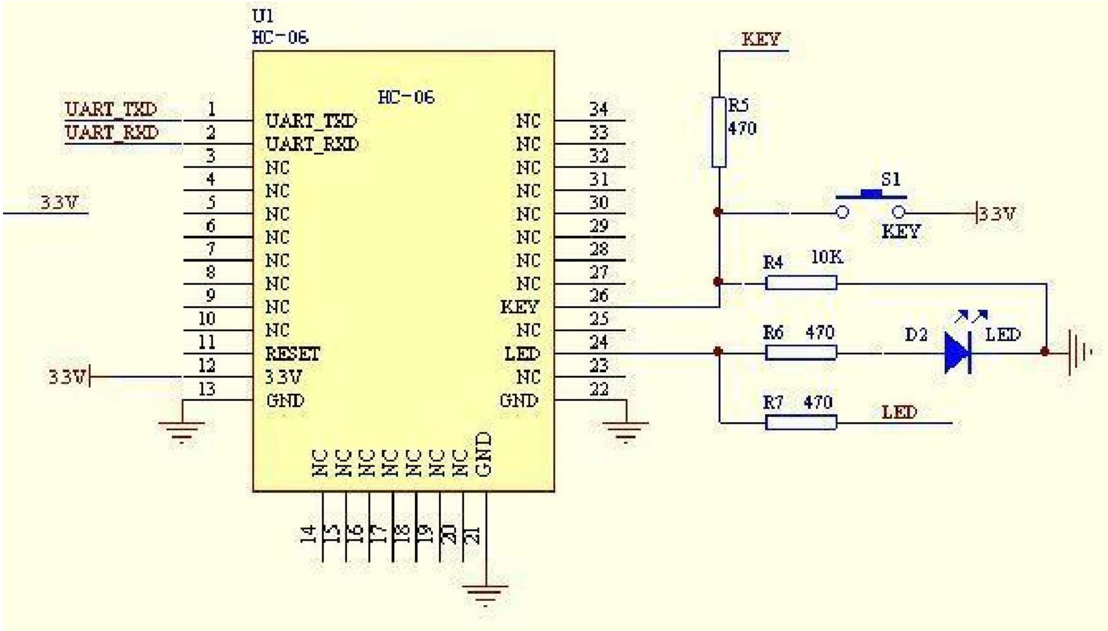


Figure 5 The circuit 1

In principle, HC-06 can work when UART_TXD, UART_RXD, VCC and GND are connected. However, for better testing results, connecting LED and KEY are recommended (when testing the master).

Where, the 3.3V TXD of MCU connects to HC-06's UART_RXD, the 3.3V RXD of MCU connects to HC-06's UART_TXD, and 3.3V power and GND should be connected. Then the minimum system is finished.

Note that, the PIN2:UART_RXD of Bluetooth module has no pull-up resistor. If the MCU TXD doesn't have pull-up function, then user should add a pull-up resistor to the UART_RXD. It may be easy to be ignored.

If there are two MCU which connect to master and slave device respectively, then before paired(LED will flicker) user can send AT commands by serial port when the system is power on. Please refer to HC-04 and HC-06's data sheet for detailed commands. In the last chapter, the command set will be introduced. Please pay attention to that the command of HC-04/HC-06 doesn't have terminator. For example, consider the call command, sending out AT is already enough, need not add the CRLF (carriage return line feed).

If the LED is constant lighting, it indicates the pairing is finished. The two MCUs can communicate with each other by serial port. User can think there is a serial port line between two MCUs.

(2) The circuit 2 (connect the module to 5V serial port of MCU) is showed by figure 6.

Figure 6 is the block diagram of Bluetooth baseboard. This kind of circuit can amplify Bluetooth module’s operating voltage to 3.1-6.5V. In this diagram, the J1 port can not only be connected with MCU system of 3.3V and 5V, but also can be connected with computer serial port.

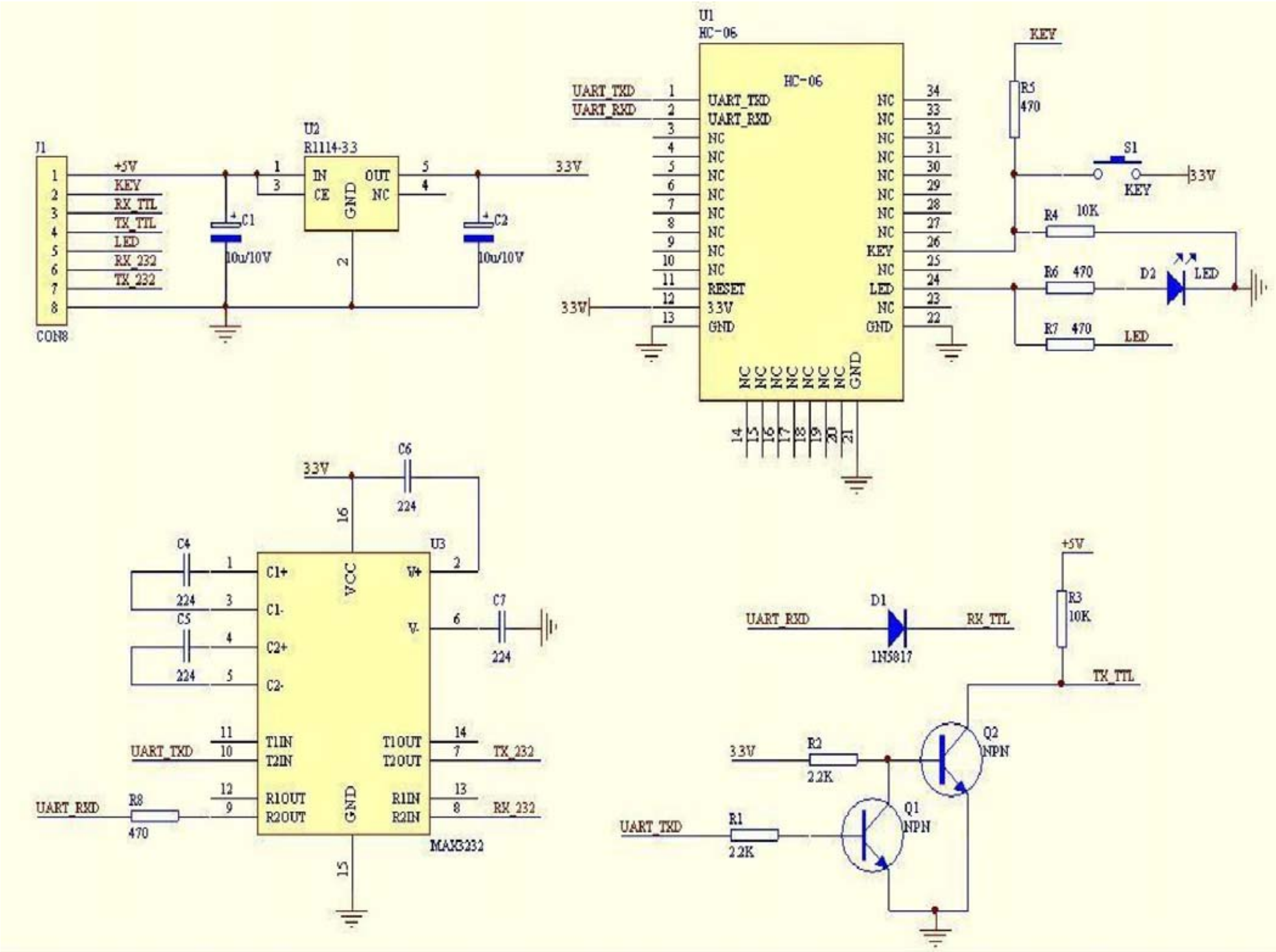


Figure 6 The circuit 2

(3) AT command test

Before paired, the mode of HC-04 and HC-06 are AT mode.

On the condition of 9600N81, OK will be received when user send the two letters AT. Please refer to the last chapter of datasheet for other commands of HC-06. Please pay attention to that sending out AT is already enough, need not add the CRLF (carriage return line feed).

The command set of Version V1.4 doesn’t include parity. The version V1.5 and its later version have parity function. Moreover, there are three more commands of V1.5 than V1.4. They are:

No parity (default) AT+PN

Odd parity AT+PO

Even parity AT+PE

Do not let the sending frequency of AT command of HC-06 exceed 1Hz, because the command of HC-06 end or not is determined by the time interval.

(4) Pairing with adapter

User can refer to the download center of the company's website for "The Introduction of IVT" that introduces the Bluetooth module makes pair with computer adapter. That document taking HC-06-D for example introduces how the serial module makes pair with the adapter. That method is like to make pair with cell-phone. But the difference is that cell-phone need a third-party communication software to help. It's liked the kind of PC serial helper of and the hyper terminal. A software named "PDA serial helper" provided by our company is suitable for WM system. It has been proven that this serial module is supported by many smart phone systems' Bluetooth, such as, sybian, android, windows mobile and etc.

(5) Pairing introduction

HC-06 master device has no memory before the first use. If the password is correct, the mater device will make pair with the slave device automatically in the first use. In the following use, the master device will remember the Bluetooth address of the last paired device and search it. The searching won't stop until the device is found. If master device's PIN26 is input high level, the device will lose the memory. In that occasion, it'll search the proper slave device like the first use. Based on this function, the master device can be set to make pair with the specified address or any address by user.

(6) Reset new password introduction

User can set a new password for the HC-06 through AT+PINxxxx command. But the new password will become active after discharged all the energy of the module. If the module still has any energy, the old one is still active. In the test, for discharging all the system energy and activating the new password, we can connect the power supply PIN with GND about 20 seconds after the power is cut off. Generally, shutting down the device for 30 minutes also can discharge the energy, if there is no peripheral circuit helps discharge energy. User should make the proper way according to the specific situation.

(7) Name introduction

If the device has no name, it's better that user doesn't try to change the master device name. The name should be limited in 20 characters.

Summary: The character of HC-06: 1 not many command 2 easy for application 3 low price. It's good for some specific application. HC-04 is very similar with HC-06. Their only one difference is HC-04 is for industry, HC-06 is for civil. Except this, they don't have difference.

The following reference about HC-04 and HC-06 can be downloaded from company website www.wavesen.com:

HC-06 datasheet .pdf	(the command set introduction is included)
HC-04 datasheet .pdf	(the command set introduction is included)
IVT BlueSoleil-2.6	(IVT Bluetooth drive test version)
Bluetooth FAQ.pdf	
HC-04-D(HD-06-D)datasheet(English).pdf	
HC-06-AT command software (test version)	(some commands in V1.5 is not supported by V1.4)
PCB package of Bluetooth key modules	(PCB package lib in protel)
IVT software manual.pdf	(introduce how to operate the modern and make pair with Bluetooth module)
PDA serial test helper.exe	(serial helper used for WM system)

5 manual for the first use of HC-05

This chapter will introduce how to test and use the HC-05 if it's the first time for user to operate it.

(1) PINs description

PIN1	UART_TXD, Bluetooth serial signal sending PIN, can connect with MCU's RXD PIN
PIN2	UART_RXD, Bluetooth serial signal receiving PIN, can connect with the MCU's TXD PIN, there is no pull-up resistor in this PIN. But It needs to be added an eternal pull-up resistor.
PIN11	RESET, the reset PIN of module, inputting low level can reset the module, when the module is in using, this PIN can connect to air.
PIN12	VCC, voltage supply for logic, the standard voltage is 3.3V, and can work at 3.0-4.2V
PIN13	GND

PIN31	LED1, indicator of work mode. Has 3 modes: When the module is supplied power and PIN34 is input high level, PIN31 output 1Hz square wave to make the LED flicker slowly. It indicates that the module is at the AT mode, and the baud rate is 38400; When the module is supplied power and PIN34 is input low level, PIN31 output 2Hz square wave to make the LED flicker quickly. It indicates the module is at the pairable mode. If PIN34 is input high level, then the module will enter to AT mode, but the output of PIN31 is still 2Hz square wave. After the pairing, PIN31 output 2Hz square ware. Note: if PIN34 keep high level, all the commands in the AT command set can be in application. Otherwise, if just excite PIN34 with high level but not keep, only some command can be used. More information has provided at chapter 2.
PIN32	Output terminal. Before paired, it output low level. Once the pair is finished, it output high level.
PIN34	Mode switch input. If it is input low level, the module is at paired or communication mode. If it's input high level, the module will enter to AT mode. Even though the module is at communication, the module can enter to the AT mode if PIN34 is input high level. Then it will go back to the communication mode if PIN34 is input low level again.

(2) Application circuit 1 (connect to the 3.3V system)

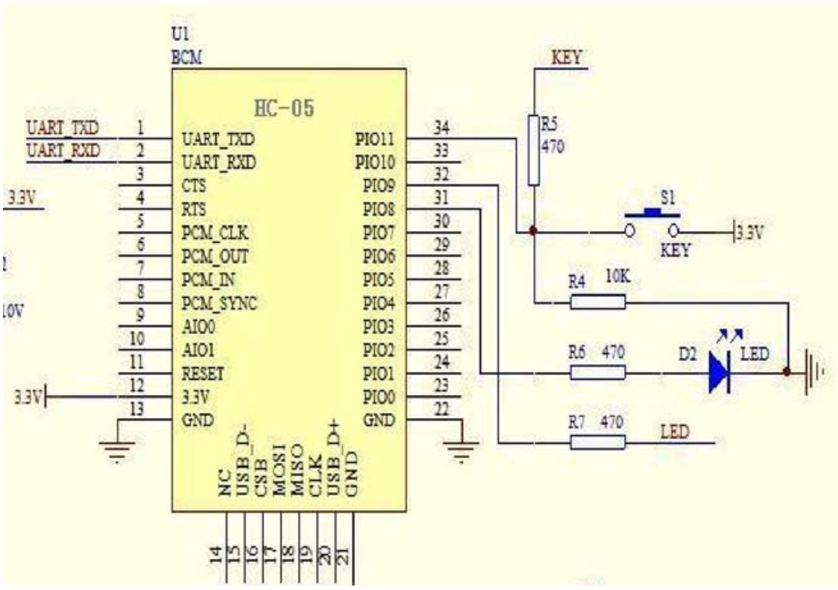


Figure 7 Application 1

(3) Application circuit 2 (connect to 5V serial system or PC serial)

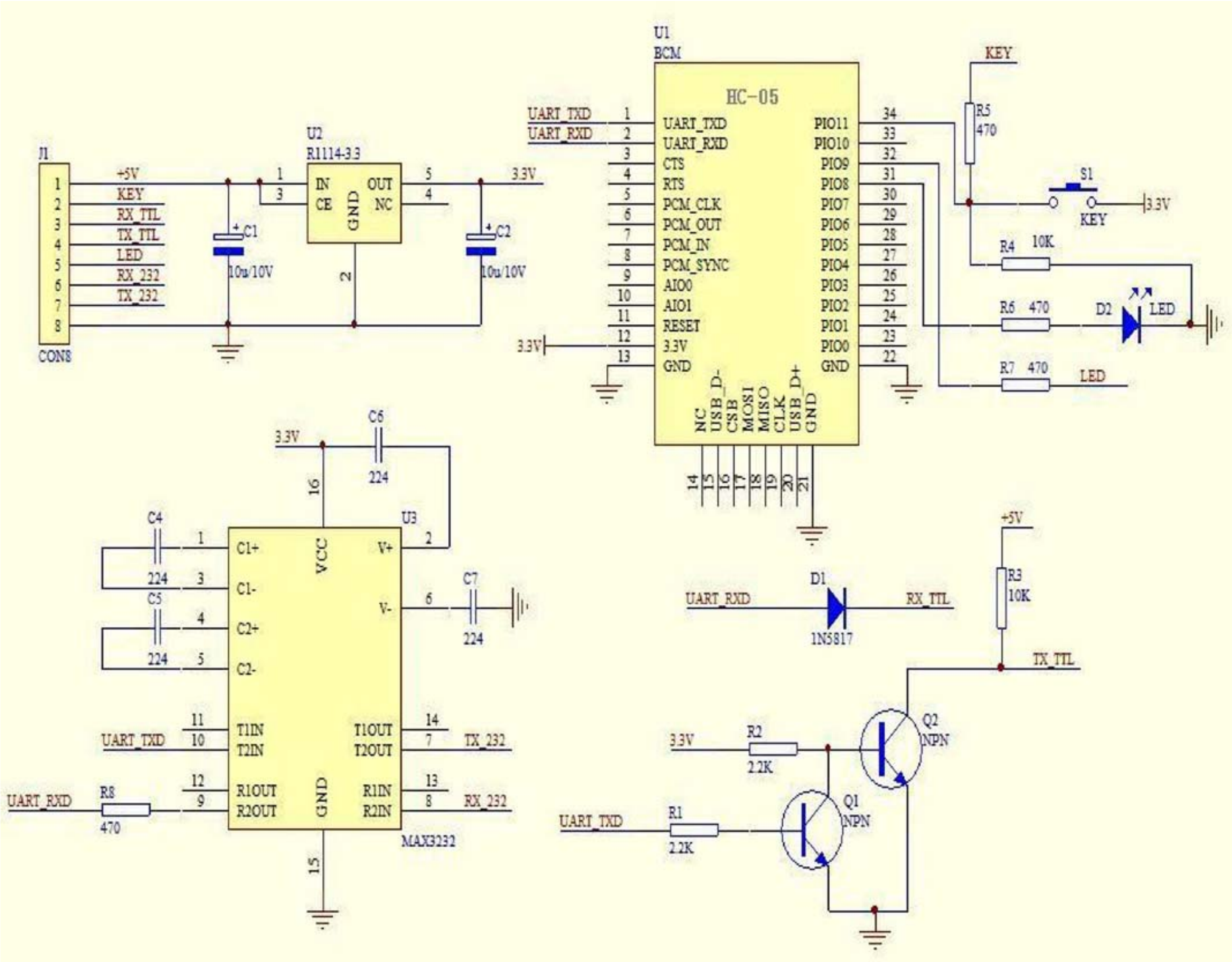


Figure 8 Application circuit 2

(4) AT command test

This chapter introduces some common commands in use. The detail introduction about HC-05 command is in HC-0305 AT command set.

Enter to AT mode:

Way1: Supply power to module and input high level to PIN34 at the same time, the module will enter to AT mode with the baud rate-38400.

Way2: In the first step, supply power to module; In the second step, input high level to PIN34. Then the module will enter to AT mode with the baud rate-9600. Way1 is recommended.

Command structure: all command should end up with “\r\n” (Hex: 0X0D X0A) as the terminator. If the serial helper is installed, user just need enter “ENTER” key at the end of command.

Reset the master-slave role command:

AT+ROLE=0 ----Set the module to be salve mode. The default mode is salve.

AT+ROLE=1 ----Set the module to be master mode.

Set memory command:

AT+CMODE=1

Set the module to make pair with the other random Bluetooth module (Not specified address). The default is this mode.

AT+CMODE=1

Set the module to make pair with the other Bluetooth module (specified address). If set the module to make pair with random one first, then set the module to make pair with the Bluetooth module has specified address. Then the module will search the last paired module until the module is found.

Reset the password command

AT+PSWD=XXXX

Set the module pair password. The password must be 4-bits.

Reset the baud rate

AT+UART== <Param>,<Param2>,<Param3>.

More information is provided at HC-0305 command set

Example:

AT+UART=9600,0,0 ----set the baud rate to be 9600N81

Reset the Bluetooth name

AT+NAME=XXXXX

Summary:

HC-05 has many functions and covers all functions of HC-06. The above commands are the most common ones. Besides this, HC-05 leaves lots of space for user. So HC-05 is better than HC-06 and recommended. HC-03 is similar with HC-05. The above introduction also suits HC-03

The following reference about HC-03 and HC-05 can be downloaded from company website www.wavesen.com:

HC-03 datasheet .pdf (the command set introduction is included)

HC-05 datasheet .pdf (the command set introduction is included)

IVT BlueSoleil-2.6 (IVT Bluetooth drive test version)

Bluetooth FAQ.pdf

PCB package of Bluetooth key modules (PCB package lib in protel)

IVT software manual.pdf (introduce how to operate the modern and make pair with Bluetooth module)

PDA serial test helper.exe (serial helper used for WM system)

HC-03/05 Bluetooth serial command set.pdf

6. Ordering information

The website of Guangzhou HC Information Technology Co., Ltd is www.wavesen.com The contact information is provided at the company website.

Order Way: If you want our product, you can give order to the production center of our company directly or order it in Taobao. There is a link to Taobao in our company website.

Package: 50 pieces chips in an anti-static blister package. The weight of a module is about 0.9g. The weight of a package is about 50g.

Please provide the product’s model when you order:

HC-04-M HC-04 master module

HC-04-S HC-04 slave module

HC-06-M HC-06 master module

HC-06-S HC-06 slave module

HC-03

HC-05 HC-03/05 can be preset to be master module or slave module.

MPU-6000 and MPU-6050 Product Specification Revision 3.4

CONTENTS

1 REVISION HISTORY 5
2 PURPOSE AND SCOPE 6
3 PRODUCT OVERVIEW 7
3.1 MPU-60X0 OVERVIEW 7
4 APPLICATIONS....................9
5 FEATURES....................10
5.1 GYROSCOPE FEATURES....................10
5.2 ACCELEROMETER FEATURES 10
5.3 ADDITIONAL FEATURES....................10
5.4 MOTIONPROCESSING....................11
5.5 CLOCKING 11
6 ELECTRICAL CHARACTERISTICS....................12
6.1 GYROSCOPE SPECIFICATIONS 12
6.2 ACCELEROMETER SPECIFICATIONS....................13
6.3 ELECTRICAL AND OTHER COMMON SPECIFICATIONS....................14
6.4 ELECTRICAL SPECIFICATIONS, CONTINUED 15
6.5 ELECTRICAL SPECIFICATIONS, CONTINUED 16
6.6 ELECTRICAL SPECIFICATIONS, CONTINUED 17
6.7 I²C TIMING CHARACTERIZATION....................18
6.8 SPI TIMING CHARACTERIZATION (MPU-6000 ONLY) 19
6.9 ABSOLUTE MAXIMUM RATINGS 20
7 APPLICATIONS INFORMATION....................21
7.1 PIN OUT AND SIGNAL DESCRIPTION....................21
7.2 TYPICAL OPERATING CIRCUIT....................22
7.3 BILL OF MATERIALS FOR EXTERNAL COMPONENTS....................22
7.4 RECOMMENDED POWER-ON PROCEDURE 23
7.5 BLOCK DIAGRAM 24
7.6 OVERVIEW 24
7.7 THREE-AXIS MEMS GYROSCOPE WITH 16-BIT ADCS AND SIGNAL CONDITIONING....................25
7.8 THREE-AXIS MEMS ACCELEROMETER WITH 16-BIT ADCS AND SIGNAL CONDITIONING 25
7.9 DIGITAL MOTION PROCESSOR 25
7.10 PRIMARY I²C AND SPI SERIAL COMMUNICATIONS INTERFACES 25
7.11 AUXILIARY I²C SERIAL INTERFACE 26

7.12 SELF-TEST 27
7.13 MPU-60X0 SOLUTION FOR 9-AXIS SENSOR FUSION USING I^{2}C INTERFACE 28
7.14 MPU-6000 USING SPI INTERFACE 29
7.15 INTERNAL CLOCK GENERATION 30
7.16 SENSOR DATA REGISTERS 30
7.17 FIFO 30
7.18 INTERRUPTS 30
7.19 DIGITAL-OUTPUT TEMPERATURE SENSOR 31
7.20 BIAS AND LDO 31
7.21 CHARGE PUMP 31

8 PROGRAMMABLE INTERRUPTS 32

9 DIGITAL INTERFACE 33

9.1 I^{2}C AND SPI (MPU-6000 ONLY) SERIAL INTERFACES 33
9.2 I^{2}C INTERFACE 33
9.3 I^{2}C COMMUNICATIONS PROTOCOL 33
9.4 I^{2}C TERMS 36
9.5 SPI INTERFACE (MPU-6000 ONLY) 37

10 SERIAL INTERFACE CONSIDERATIONS (MPU-6050) 38

10.1 MPU-6050 SUPPORTED INTERFACES 38
10.2 LOGIC LEVELS 38
10.3 LOGIC LEVELS DIAGRAM FOR AUX_*VDDIO* = 0 39

11 ASSEMBLY 40

11.1 ORIENTATION OF AXES 40
11.2 PACKAGE DIMENSIONS 41
11.3 PCB DESIGN GUIDELINES 42
11.4 ASSEMBLY PRECAUTIONS 43
11.5 STORAGE SPECIFICATIONS 46
11.6 PACKAGE MARKING SPECIFICATION 46
11.7 TAPE & REEL SPECIFICATION 47
11.8 LABEL 48
11.9 PACKAGING 49
11.10 REPRESENTATIVE SHIPPING CARTON LABEL 50

12 RELIABILITY 51

12.1 QUALIFICATION TEST POLICY 51

12.2 QUALIFICATION TEST PLAN 51
13 ENVIRONMENTAL COMPLIANCE..52

1 Revision History

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

2 Purpose and Scope

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

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

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

3 Product Overview

3.1 MPU-60X0 Overview

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

The MPU-60X0 is the world's first integrated 6-axis MotionTracking device that combines a 3-axis gyroscope, 3-axis accelerometer, and a Digital Motion Processor™ (DMP) all in a small 4x4x0.9mm package. With its dedicated I^2C sensor bus, it directly accepts inputs from an external 3-axis compass to provide a complete 9-axis MotionFusion™ output. The MPU-60X0 MotionTracking device, with its 6-axis integration, on-board MotionFusion™, and run-time calibration firmware, enables manufacturers to eliminate the costly and complex selection, qualification, and system level integration of discrete devices, guaranteeing optimal motion performance for consumers. The MPU-60X0 is also designed to interface with multiple non-inertial digital sensors, such as pressure sensors, on its auxiliary I^2C port. The MPU-60X0 is footprint compatible with the MPU-30X0 family.

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

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

Communication with all registers of the device is performed using either I^2C at 400kHz or SPI at 1MHz (MPU-6000 only). For applications requiring faster communications, the sensor and interrupt registers may be read using SPI at 20MHz (MPU-6000 only). Additional features include an embedded temperature sensor and an on-chip oscillator with ±1% variation over the operating temperature range.

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

For power supply flexibility, the MPU-60X0 operates from VDD power supply voltage range of 2.375V-3.46V. Additionally, the MPU-6050 provides a VLOGIC reference pin (in addition to its analog supply pin: VDD), which sets the logic levels of its I^2C interface. The VLOGIC voltage may be 1.8V±5% or VDD.

The MPU-6000 and MPU-6050 are identical, except that the MPU-6050 supports the I^2C serial interface only, and has a separate VLOGIC reference pin. The MPU-6000 supports both I^2C and SPI interfaces and has a single supply pin, VDD, which is both the device's logic reference supply and the analog supply for the part. The table below outlines these differences:

Primary Differences between MPU-6000 and MPU-6050

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

4 Applications

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

5 Features

5.1 Gyroscope Features

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

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

5.2 Accelerometer Features

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

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

5.3 Additional Features

The MPU-60X0 includes the following additional features:

- 9-Axis MotionFusion by the on-chip Digital Motion Processor (DMP)
- Auxiliary master I^2C bus for reading data from external sensors (e.g., magnetometer)
- 3.9mA operating current when all 6 motion sensing axes and the DMP are enabled
- VDD supply voltage range of 2.375V-3.46V
- Flexible VLOGIC reference voltage supports multiple I^2C interface voltages (MPU-6050 only)
- Smallest and thinnest QFN package for portable devices: 4x4x0.9mm
- Minimal cross-axis sensitivity between the accelerometer and gyroscope axes
- 1024 byte FIFO buffer reduces power consumption by allowing host processor to read the data in bursts and then go into a low-power mode as the MPU collects more data
- Digital-output temperature sensor
- User-programmable digital filters for gyroscope, accelerometer, and temp sensor
- 10,000 *g* shock tolerant
- 400kHz Fast Mode I^2C for communicating with all registers
- 1MHz SPI serial interface for communicating with all registers (MPU-6000 only)
- 20MHz SPI serial interface for reading sensor and interrupt registers (MPU-6000 only)

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

5.4 MotionProcessing

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

5.5 Clocking

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

6 Electrical Characteristics

6.1 Gyroscope Specifications

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

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

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

6.2 Accelerometer Specifications

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

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

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

6.3 Electrical and Other Common Specifications

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

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

6.4 Electrical Specifications, Continued

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

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

6.5 Electrical Specifications, Continued

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

Parameters	Conditions	Typical	Units	Notes
Primary 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

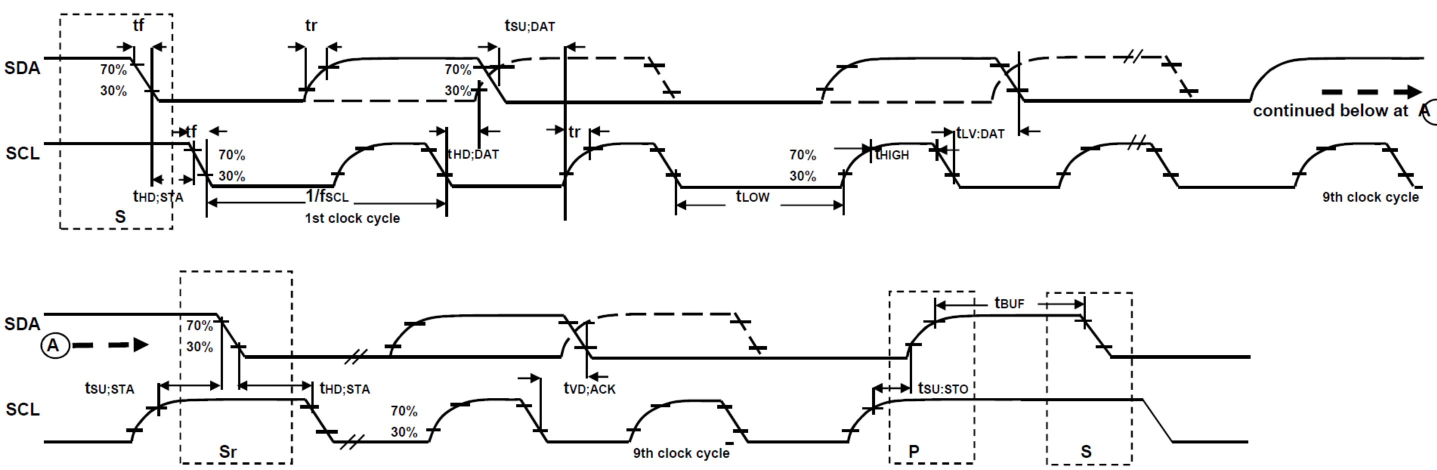
Parameters	Conditions	Min	Typical	Max	Units	Notes
INTERNAL CLOCK SOURCE	**CLK_SEL=0,1,2,3**					
Gyroscope Sample Rate, Fast	DLPFCFG=0 SAMPLERATEDIV = 0		8		kHz	
Gyroscope Sample Rate, Slow	DLPFCFG=1,2,3,4,5, or 6 SAMPLERATEDIV = 0		1		kHz	
Accelerometer Sample Rate			1		kHz	
Clock Frequency Initial 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

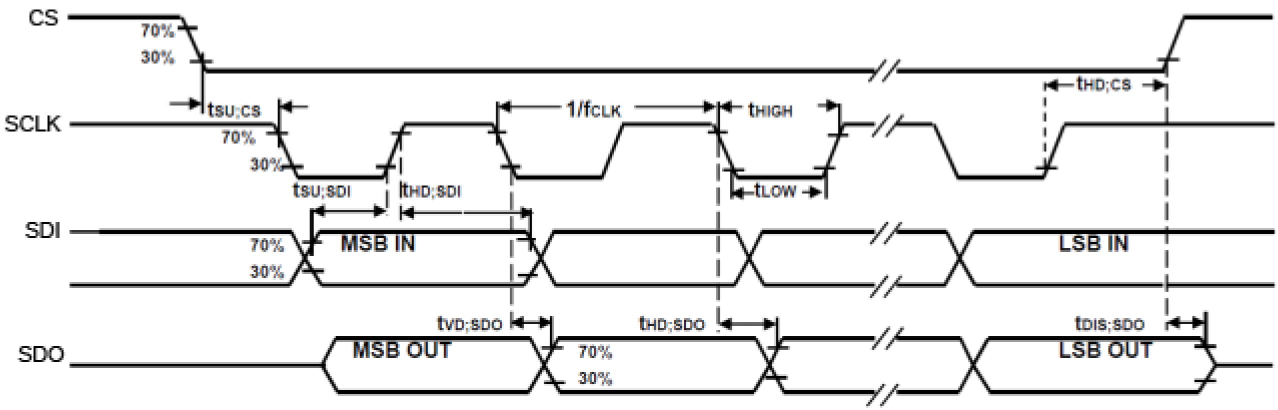


I²C Bus Timing Diagram

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

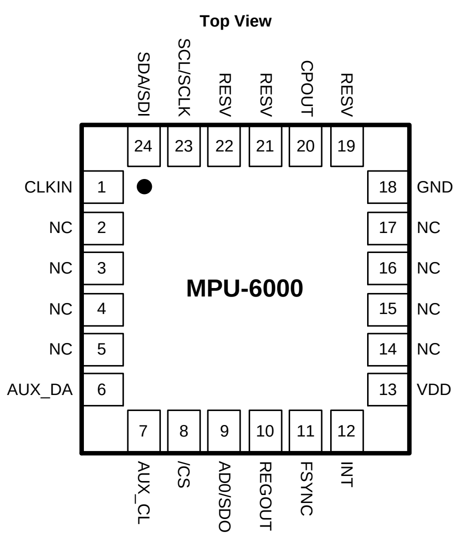
Stress above those listed as "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only and functional operation of the device at these conditions is not implied. Exposure to the absolute maximum ratings conditions for extended periods may affect device reliability.

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

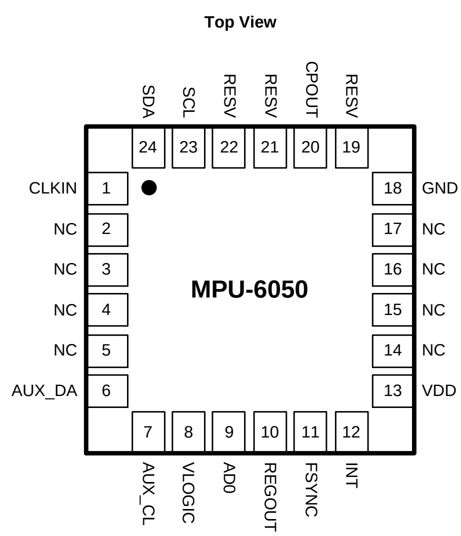
7 Applications Information

7.1 Pin Out and Signal Description

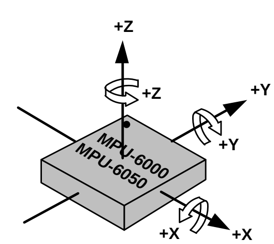
Pin Number	MPU-6000	MPU-6050	Pin Name	Pin Description
1	Y	Y	CLKIN	Optional external reference clock input. Connect to GND if unused.
6	Y	Y	AUX_DA	I^2C master serial data, for connecting to external sensors
7	Y	Y	AUX_CL	I^2C Master serial clock, for connecting to external sensors
8	Y		/CS	SPI chip select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



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

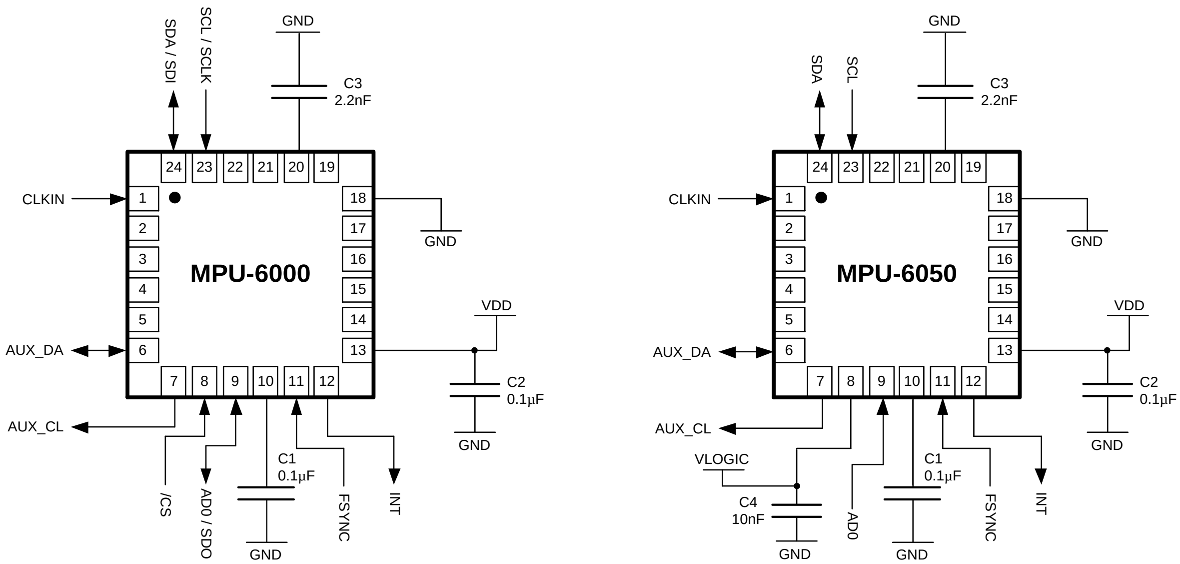


QFN Package
24-pin, 4mm x 4mm x 0.9mm



Orientation of Axes of Sensitivity and Polarity of Rotation

7.2 Typical Operating Circuit



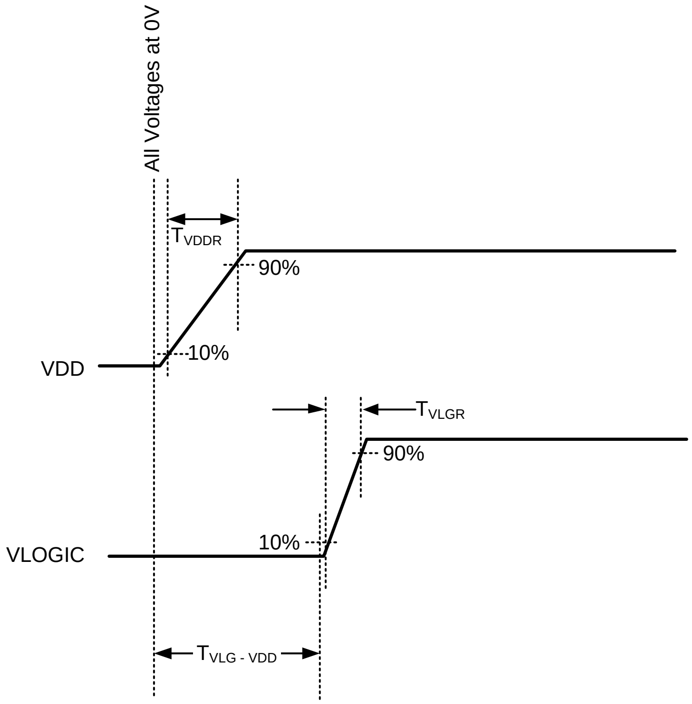
Typical Operating Circuits

7.3 Bill of Materials for External Components

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

* MPU-6050 Only.

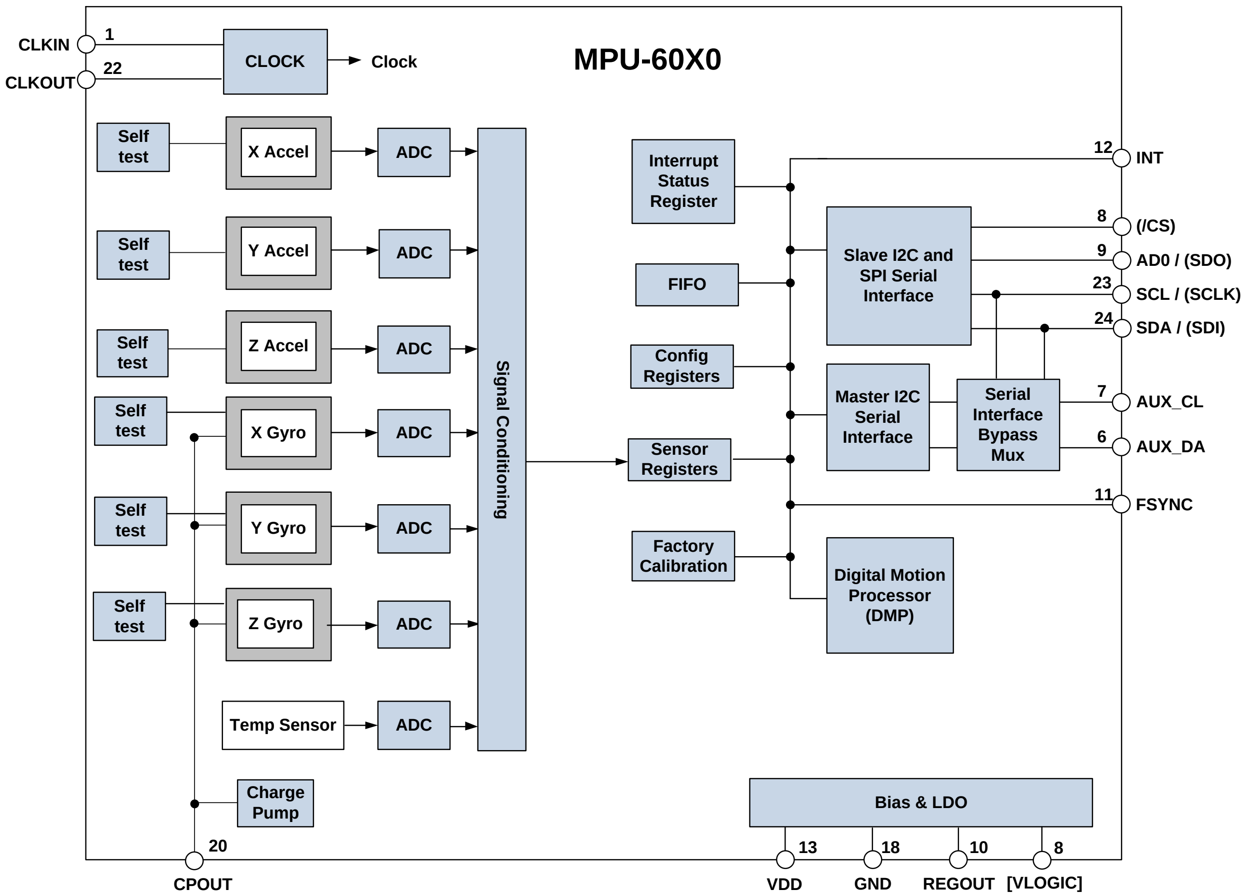
7.4 Recommended Power-on Procedure



Power-Up Sequencing

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

7.5 Block Diagram



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

7.6 Overview

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

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

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

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

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

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

7.9 Digital Motion Processor

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

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

7.10 Primary I^2C and SPI Serial Communications Interfaces

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

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

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

Auxiliary I^2C Bus Modes of Operation:

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

Auxiliary I^2C Bus IO Logic Levels

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

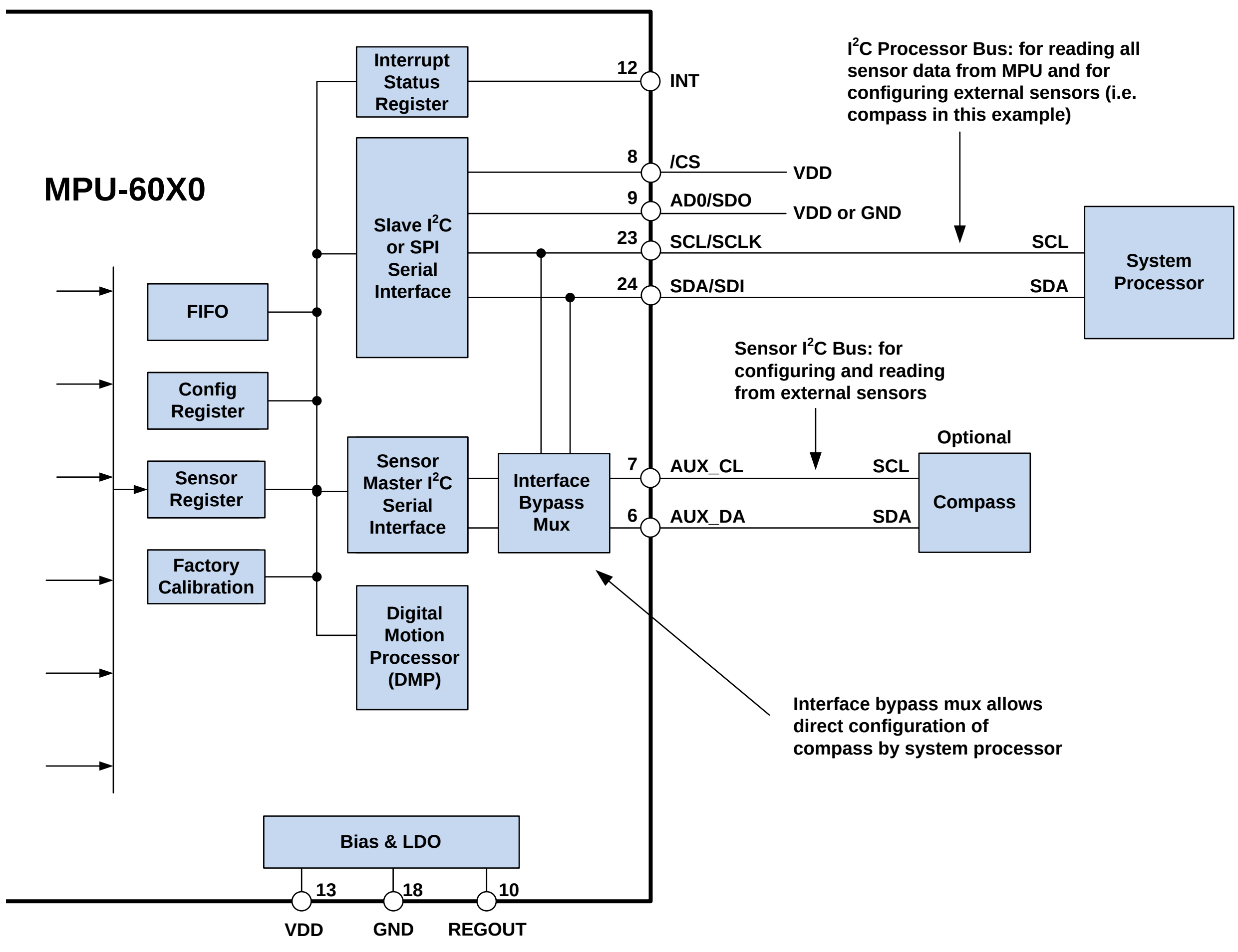
When the value of the self-test response is within the min/max limits of the product specification, the part has passed self test. When the self-test response exceeds the min/max values, the part is deemed to have failed self-test. Code for operating self test code is included within the MotionApps software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

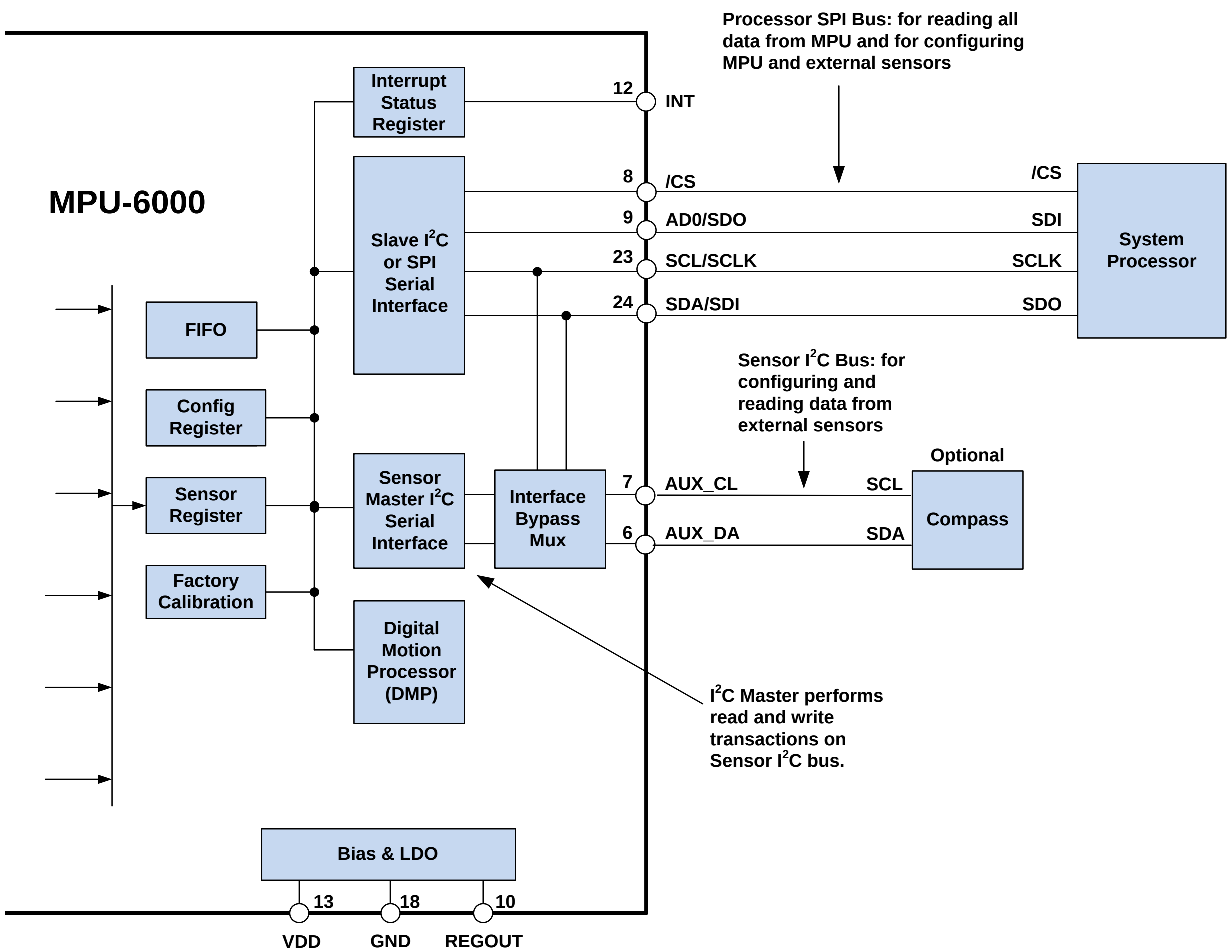
In the figure below, the system processor is an SPI master to the MPU-6000. Pins 8, 9, 23, and 24 are used to support the /CS, SDO, SCLK, and SDI signals for SPI communications. Because these SPI pins are shared with the I^2C slave pins (9, 23 and 24), the system processor cannot access the auxiliary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

Since the MPU-6000 has limited capabilities as an I^2C Master, and depends on the system processor to manage the initial configuration of any auxiliary sensors, another method must be used for programming the sensors on the auxiliary sensor I^2C bus pins 6 and 7 (AUX_DA and AUX_CL).

When using SPI communications between the MPU-6000 and the system processor, configuration of devices on the auxiliary I^2C sensor bus can be achieved by using I^2C Slaves 0-4 to perform read and write transactions on any device and register on the auxiliary I^2C bus. The I^2C Slave 4 interface can be used to perform only single byte read and write transactions.

Once the external sensors have been configured, the MPU-6000 can perform single or multi-byte reads using the sensor I^2C bus. The read results from the Slave 0-3 controllers can be written to the FIFO buffer as well as to the external sensor registers.

For further information regarding the control of the MPU-60X0's auxiliary I^2C 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).

8 Programmable Interrupts

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

Table of Interrupt Sources

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

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

9 Digital Interface

9.1 I^2C and SPI (MPU-6000 only) Serial Interfaces

The internal registers and memory of the MPU-6000/MPU-6050 can be accessed using either I^2C at 400 kHz or SPI at 1MHz (MPU-6000 only). SPI operates in four-wire mode.

Serial Interface

Pin Number	MPU-6000	MPU-6050	Pin Name	Pin Description
8	Y		/CS	SPI chip select (0=SPI enable)
8		Y	VLOGIC	Digital I/O supply voltage. VLOGIC must be ≤ VDD at all times.
9	Y		AD0 / SDO	I^2C Slave Address LSB (AD0); SPI serial data output (SDO)
9		Y	AD0	I^2C Slave Address LSB
23	Y		SCL / SCLK	I^2C serial clock (SCL); SPI serial clock (SCLK)
23		Y	SCL	I^2C serial clock
24	Y		SDA / SDI	I^2C serial data (SDA); SPI serial data input (SDI)
24		Y	SDA	I^2C serial data

Note:

To prevent switching into I^2C mode when using SPI (MPU-6000), the I^2C interface should be disabled by setting the *I2C_IF_DIS* configuration bit. Setting this bit should be performed immediately after waiting for the time specified by the "Start-Up Time for Register Read/Write" in Section 6.3.

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

9.2 I^2C Interface

I^2C is a two-wire interface comprised of the signals serial data (SDA) and serial clock (SCL). In general, the lines are open-drain and bi-directional. In a generalized I^2C interface implementation, attached devices can be a master or a slave. The master device puts the slave address on the bus, and the slave device with the matching address acknowledges the master.

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

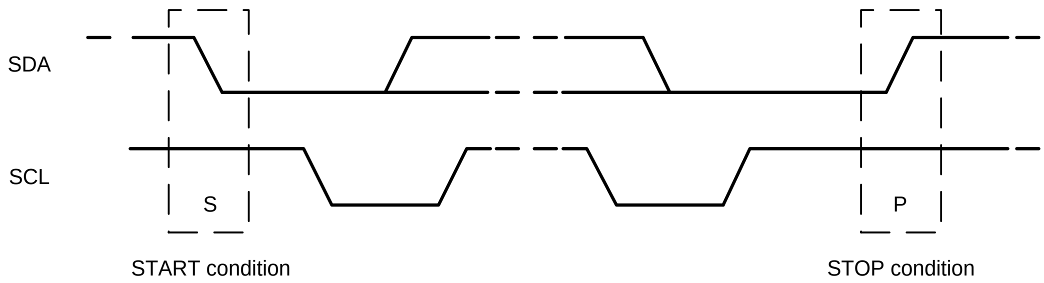
The slave address of the MPU-60X0 is b110100X which is 7 bits long. The LSB bit of the 7 bit address is determined by the logic level on pin AD0. This allows two MPU-60X0s to be connected to the same I^2C bus. When used in this configuration, the address of the one of the devices should be b1101000 (pin AD0 is logic low) and the address of the other should be b1101001 (pin AD0 is logic high).

9.3 I^2C Communications Protocol

START (S) and STOP (P) Conditions

Communication on the I^2C bus starts when the master puts the START condition (S) on the bus, which is defined as a HIGH-to-LOW transition of the SDA line while SCL line is HIGH (see figure below). The bus is considered to be busy until the master puts a STOP condition (P) on the bus, which is defined as a LOW to HIGH transition on the SDA line while SCL is HIGH (see figure below).

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

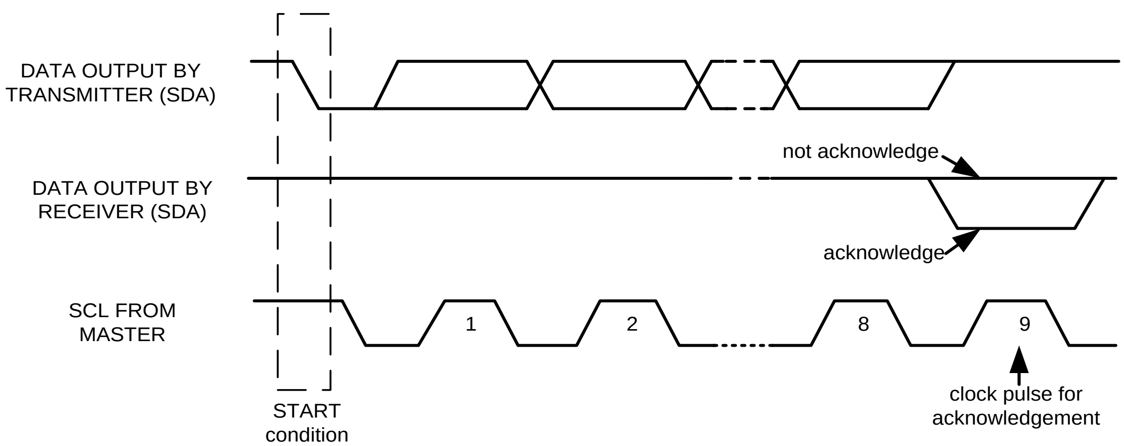


START and STOP Conditions

Data Format / Acknowledge

I^2C data bytes are defined to be 8-bits long. There is no restriction to the number of bytes transmitted per data transfer. Each byte transferred must be followed by an acknowledge (ACK) signal. The clock for the acknowledge signal is generated by the master, while the receiver generates the actual acknowledge signal by pulling down SDA and holding it low during the HIGH portion of the acknowledge clock pulse.

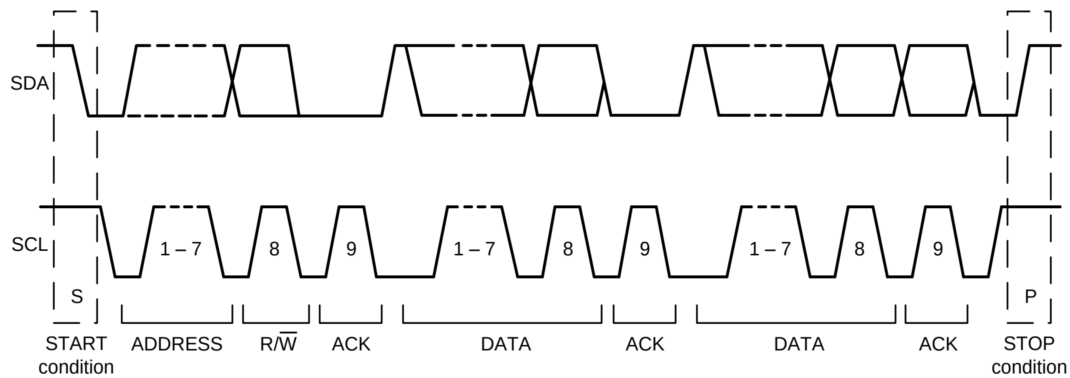
If a slave is busy and cannot transmit or receive another byte of data until some other task has been performed, it can hold SCL LOW, thus forcing the master into a wait state. Normal data transfer resumes when the slave is ready, and releases the clock line (refer to the following figure).



Acknowledge on the I^2C Bus

Communications

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



Complete I^{2}C Data Transfer

To write the internal MPU-60X0 registers, the master transmits the start condition (S), followed by the I^{2}C address and the write bit (0). At the 9th clock cycle (when the clock is high), the MPU-60X0 acknowledges the transfer. Then the master puts the register address (RA) on the bus. After the MPU-60X0 acknowledges the reception of the register address, the master puts the register data onto the bus. This is followed by the ACK signal, and data transfer may be concluded by the stop condition (P). To write multiple bytes after the last ACK signal, the master can continue outputting data rather than transmitting a stop signal. In this case, the MPU-60X0 automatically increments the register address and loads the data to the appropriate register. The following figures show single and two-byte write sequences.

Single-Byte Write Sequence

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

Burst Write Sequence

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

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

Single-Byte Read Sequence

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

Burst Read Sequence

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

9.4 I²C Terms

Signal	Description
S	Start Condition: SDA goes from high to low while SCL is high
AD	Slave I²C address
W	Write bit (0)
R	Read bit (1)
ACK	Acknowledge: SDA line is low while the SCL line is high at the 9th clock cycle
NACK	Not-Acknowledge: SDA line stays high at the 9th clock cycle
RA	MPU-60X0 internal register address
DATA	Transmit or received data
P	Stop condition: SDA going from low to high while SCL is high

9.5 SPI Interface (MPU-6000 only)

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

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

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

SPI Operational Features

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

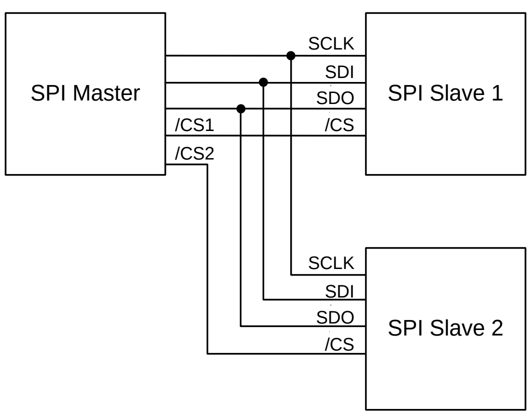
SPI Address format

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

SPI Data format

MSB							**LSB**
D7	D6	D5	D4	D3	D2	D1	D0

6. Supports Single or Burst Read/Writes.



Typical SPI Master / Slave Configuration

10 Serial Interface Considerations (MPU-6050)

10.1 MPU-6050 Supported Interfaces

The MPU-6050 supports I^2C communications on both its primary (microprocessor) serial interface and its auxiliary interface.

10.2 Logic Levels

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

I/O Logic Levels vs. *AUX_VDDIO*

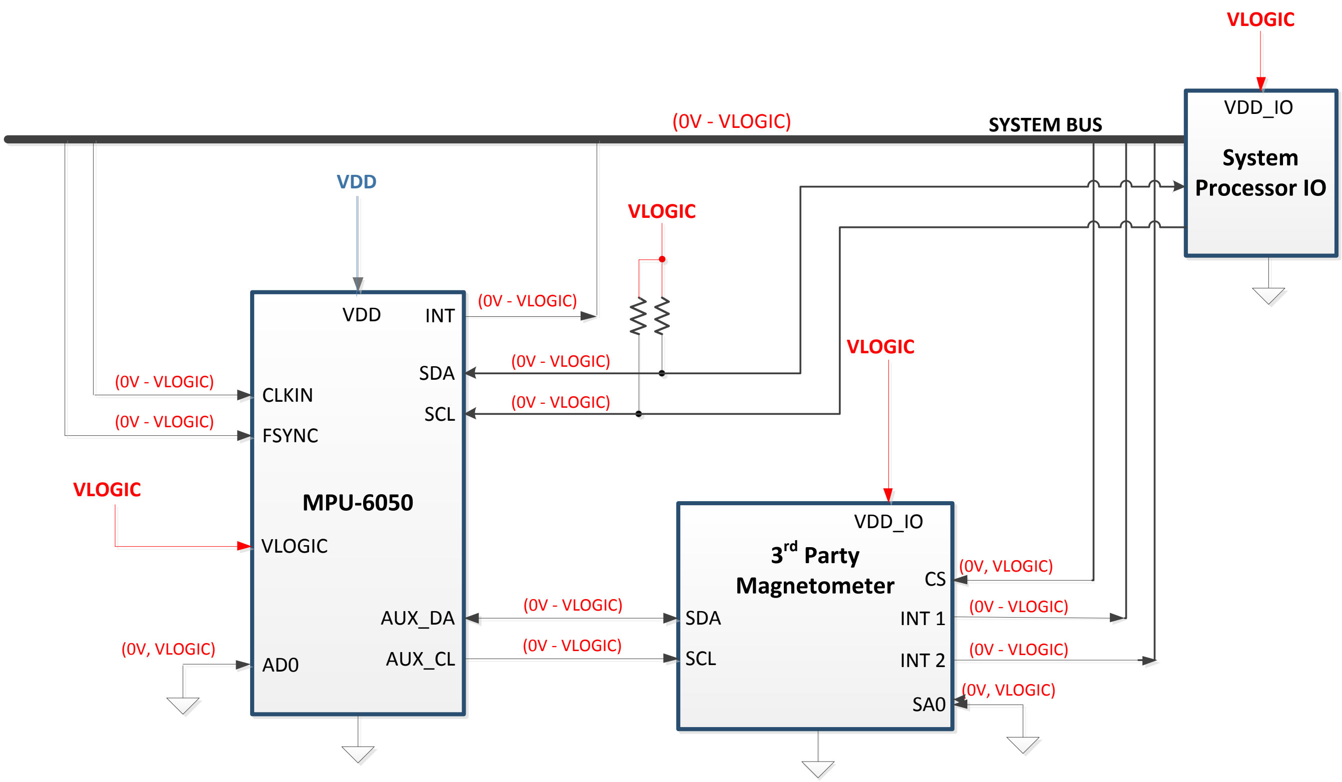
AUX_VDDIO	**MICROPROCESSOR LOGIC LEVELS** (Pins: SDA, SCL, AD0, CLKIN, INT)	**AUXILLARY LOGIC LEVELS** (Pins: AUX_DA, AUX_CL)
0	VLOGIC	VLOGIC

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

When *AUX_VDDIO* is set to 0 (its power-on-reset value), VLOGIC is the power supply voltage for both the microprocessor system bus and the auxiliary I^2C bus, as shown in the figure of Section 10.3.

10.3 Logic Levels Diagram for AUX_*VDDIO* = 0

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



I/O Levels and Connections for *AUX_VDDIO* = 0

Notes:

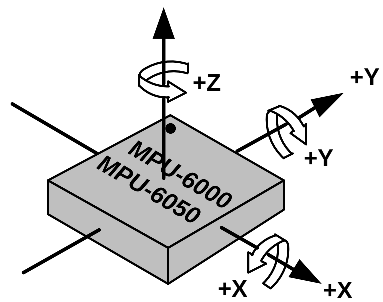
1. AUX_VDDIO determines the IO voltage levels of AUX_DA and AUX_CL
   (0 = set output levels relative to VLOGIC)
2. All other MPU-6050 logic IOs are referenced to VLOGIC.

11 Assembly

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

11.1 Orientation of Axes

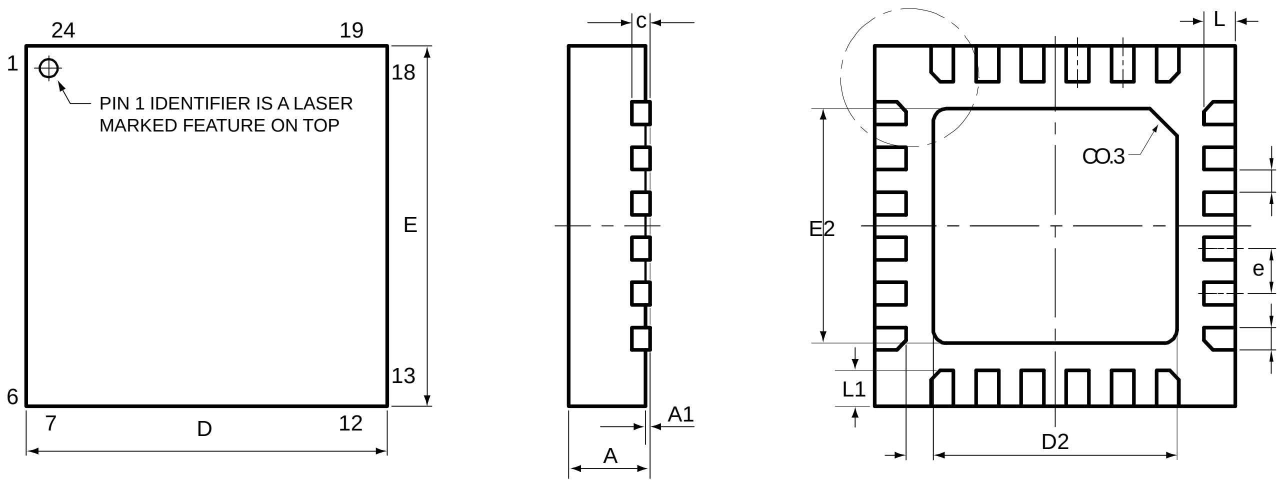
The diagram below shows the orientation of the axes of sensitivity and the polarity of rotation. Note the pin 1 identifier (•) in the figure.

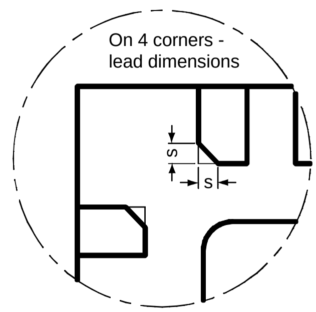


Orientation of Axes of Sensitivity and Polarity of Rotation

11.2 Package Dimensions

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

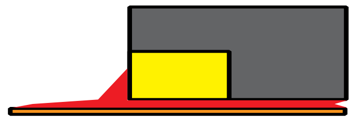




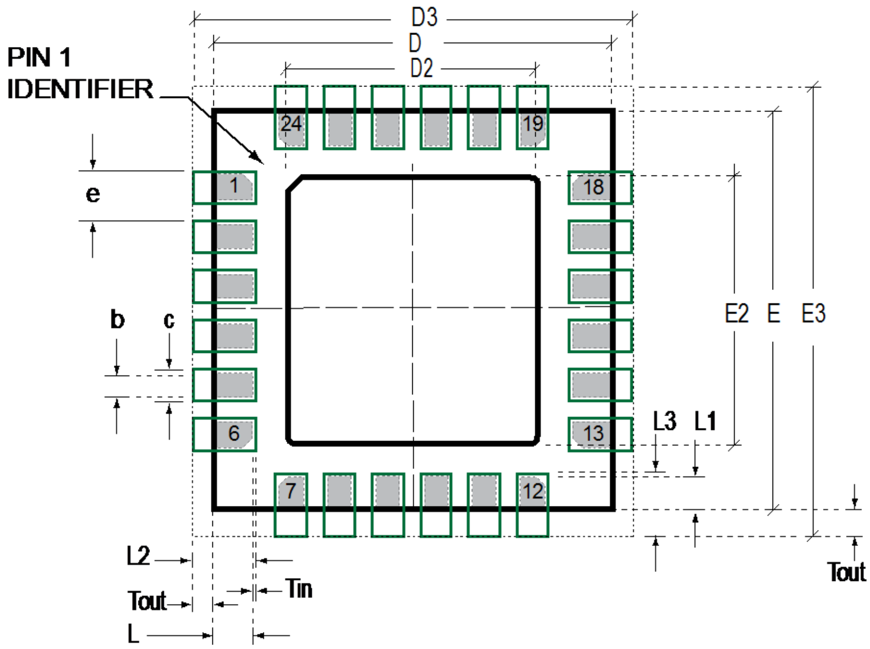
SYMBOLS	DIMENSIONS IN MILLIMETERS		
	MIN	NOM	MAX
A	0.85	0.90	0.95
A1	0.00	0.02	0.05
b	0.18	0.25	0.30
c	---	0.20 REF	---
D	3.90	4.00	4.10
D2	2.65	2.70	2.75
E	3.90	4.00	4.10
E2	2.55	2.60	2.65
e	---	0.50	---
f (e-b)	---	0.25	---
K	0.25	0.30	0.35
L	0.30	0.35	0.40
L1	0.35	0.40	0.45
s	0.05	---	0.15

11.3 PCB Design Guidelines

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



JEDEC type extension with solder rising on outer edge



PCB Layout Diagram

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

PCB Dimensions Table (for PCB Lay-out Diagram)

11.4 Assembly Precautions

11.4.1 Gyroscope Surface Mount Guidelines

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

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

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

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

11.4.2 Exposed Die Pad Precautions

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

11.4.3 Trace Routing

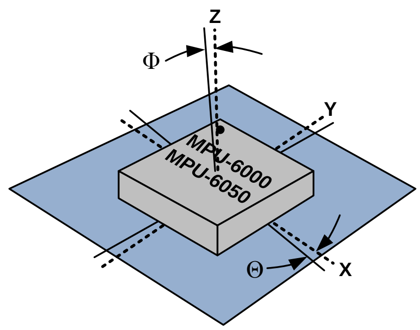
Routing traces or vias under the gyro package such that they run under the exposed die pad is prohibited. Routed active signals may harmonically couple with the gyro MEMS devices, compromising gyro response. These devices are designed with the drive frequencies as follows: X = 33±3Khz, Y = 30±3Khz, and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

11.4.4 Component Placement

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

11.4.5 PCB Mounting and Cross-Axis Sensitivity

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



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

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

Cross-Axis Sensitivity vs. Orientation Error

Orientation Error (θ or Φ)	Cross-Axis Sensitivity (sinθ or sinΦ)
0º	0%
0.5º	0.87%
1º	1.75%

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

11.4.6 MEMS Handling Instructions

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

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

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

11.4.7 ESD Considerations

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

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

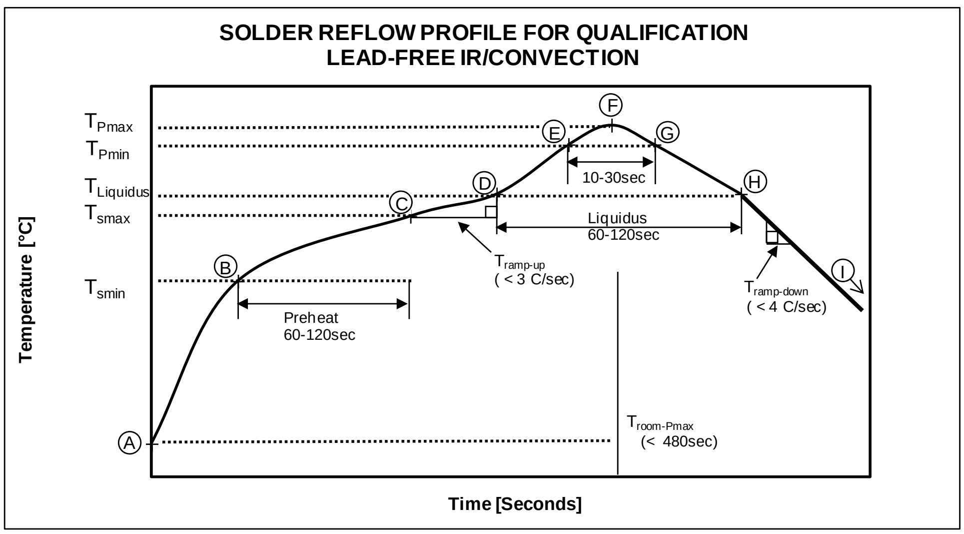
Restrict all device handling to ESD protected work areas that measure less than 200V static charge. Ensure that all workstations and personnel are properly grounded to prevent ESD.

11.4.8 Reflow Specification

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

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

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



Temperature Set Points Corresponding to Reflow Profile Above

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

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

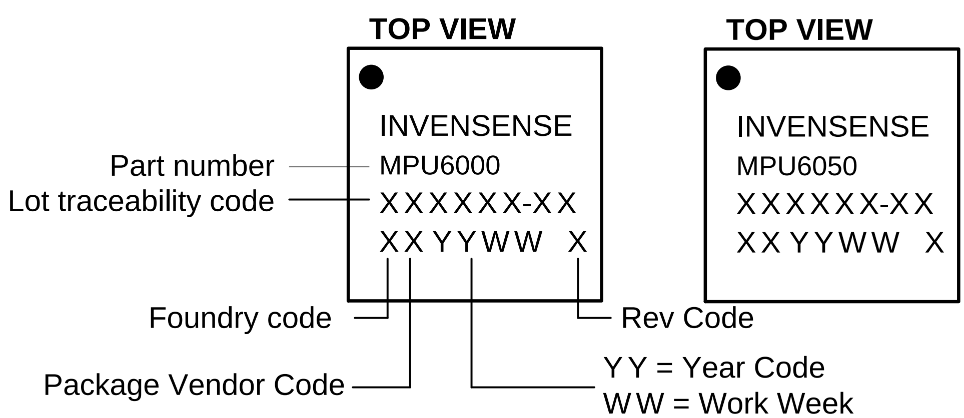
Production Reflow: Check the recommendations of your solder manufacturer. For optimum results, use lead-free solders that have lower specified temperature profiles (Tp_{max} ~ 235°C). Also use lower ramp-up and ramp-down rates than those used in the qualification profile. Never exceed the maximum conditions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

11.5 Storage Specifications

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

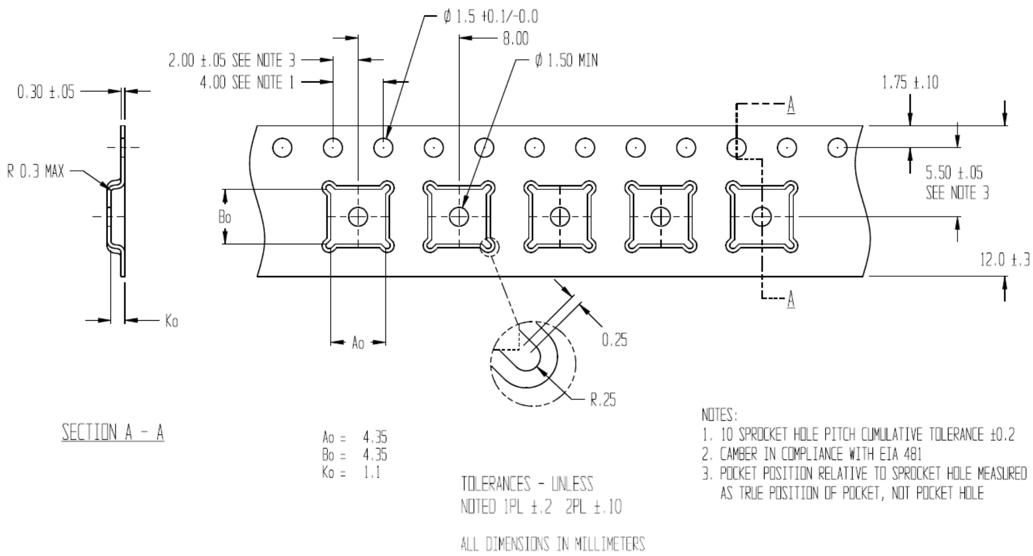
Calculated shelf-life in moisture-sealed bag	12 months -- Storage conditions: <40°C and <90% RH
After opening moisture-sealed bag	168 hours -- Storage conditions: ambient ≤30°C at 60%RH

11.6 Package Marking Specification

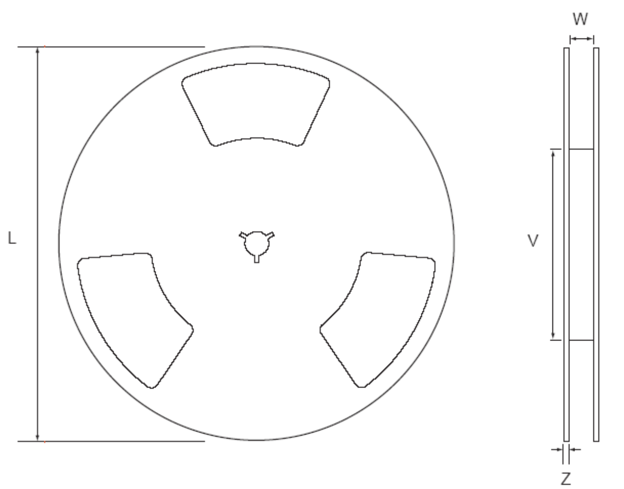


Package Marking Specification

11.7 Tape & Reel Specification



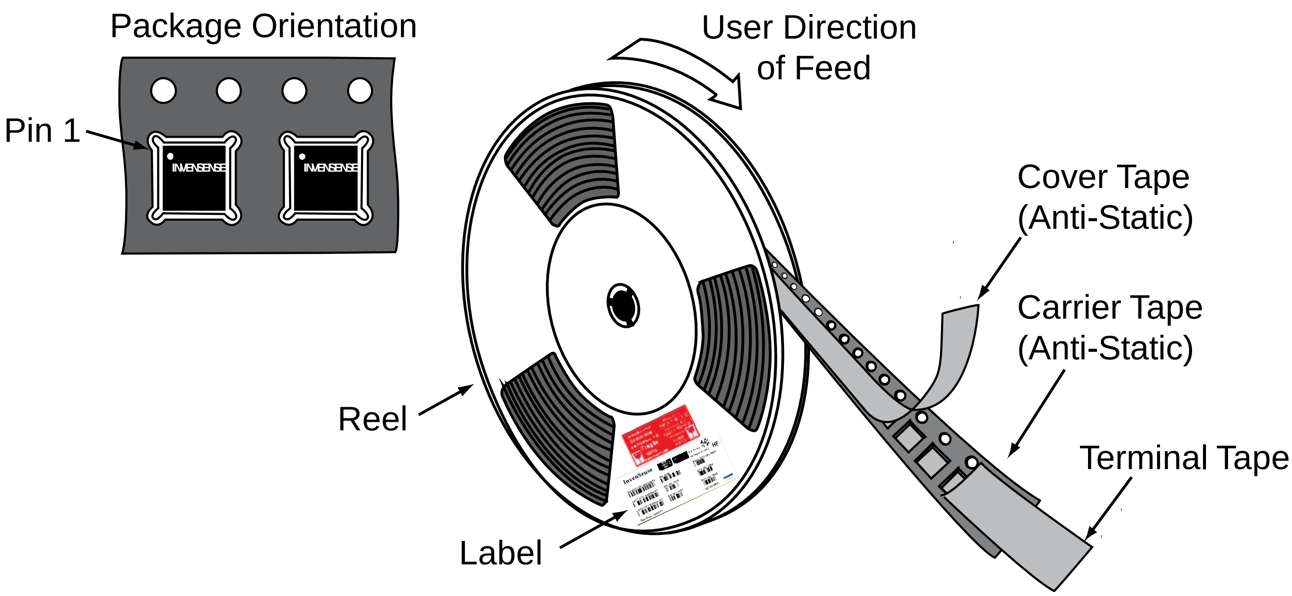
Tape Dimensions



Reel Outline Drawing

Reel Dimensions and Package Size

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



Tape and Reel Specification

Reel Specifications

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

11.8 Label

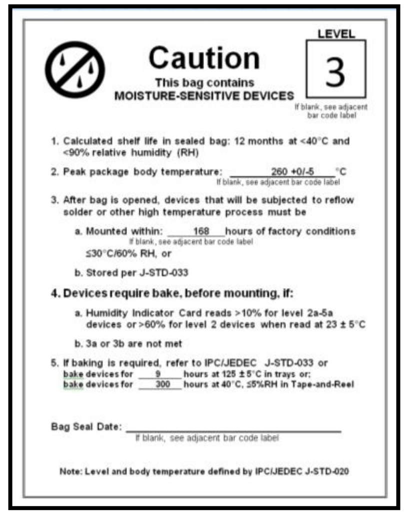
Barcode Label

Location of Label on Reel

11.9 Packaging

REEL – with Barcode & Caution labels

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



MSL3 Label

Caution Label

ESD Label

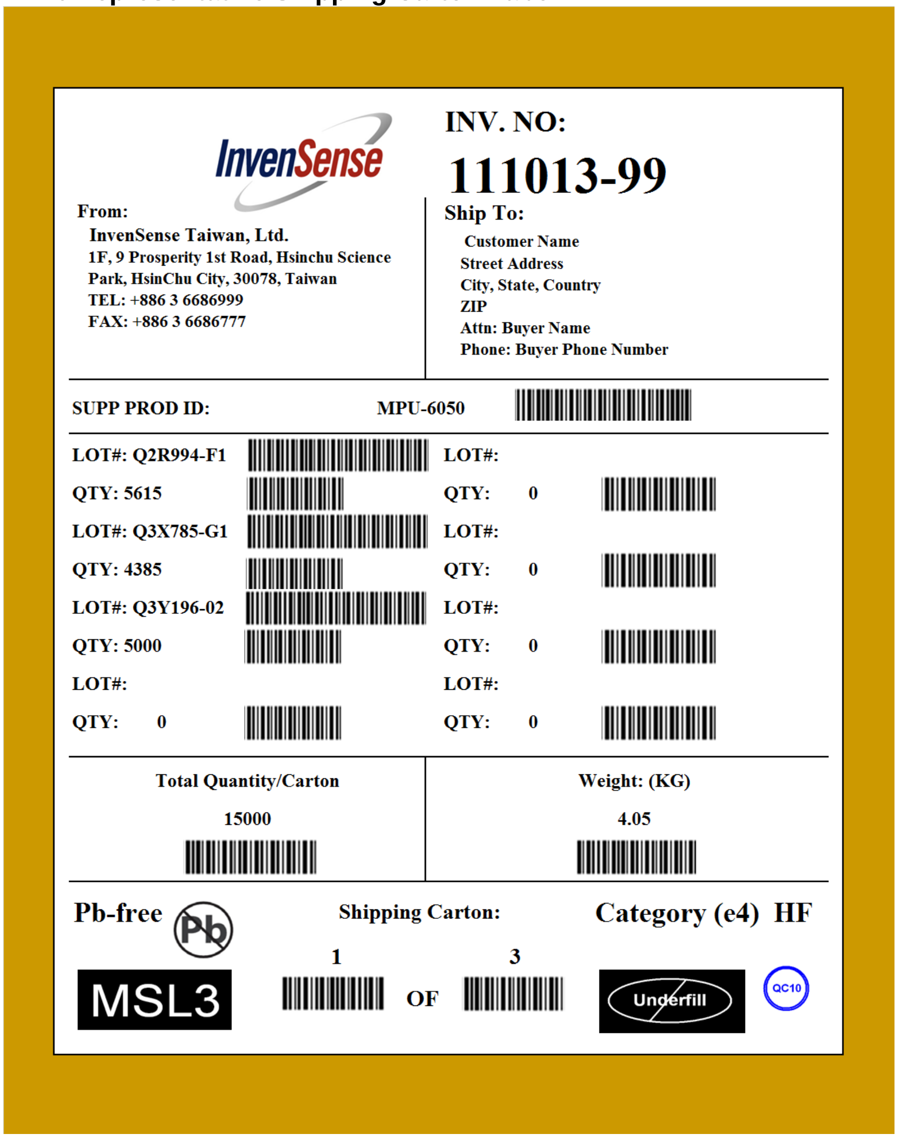
Inner Bubble Wrap

Pizza Box

Pizza Boxes Placed in Foam-Lined Shipper Box

Outer Shipper Label

11.10 Representative Shipping Carton Label


InvenSense

INV. NO:

111013-99

From:

InvenSense Taiwan, Ltd.

1F, 9 Prosperity 1st Road, Hsinchu Science Park, HsinChu City, 30078, Taiwan

TEL: +886 3 6686999

FAX: +886 3 6686777

Ship To:

Customer Name

Street Address

City, State, Country

ZIP

Attn: Buyer Name

Phone: Buyer Phone Number

SUPP PROD ID: MPU-6050

LOT#: Q2R994-F1	LOT#:
QTY: 5615	QTY: 0
LOT#: Q3X785-G1	LOT#:
QTY: 4385	QTY: 0
LOT#: Q3Y196-02	LOT#:
QTY: 5000	QTY: 0
LOT#:	LOT#:
QTY: 0	QTY: 0

Total Quantity/Carton	Weight: (KG)
15000	4.05

Pb-free

Shipping Carton:

1 OF 3

Category (e4) HF

MSL3

Underfill

QC10

12 Reliability

12.1 Qualification Test Policy

InvenSense's products complete a Qualification Test Plan before being released to production. The Qualification Test Plan for the MPU-60X0 followed the JESD47I Standards, "Stress-Test-Driven Qualification of Integrated Circuits," with the individual tests described below.

12.2 Qualification Test Plan

Accelerated Life Tests

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

Device Component Level Tests

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

Board Level Tests

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

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

13 Environmental Compliance

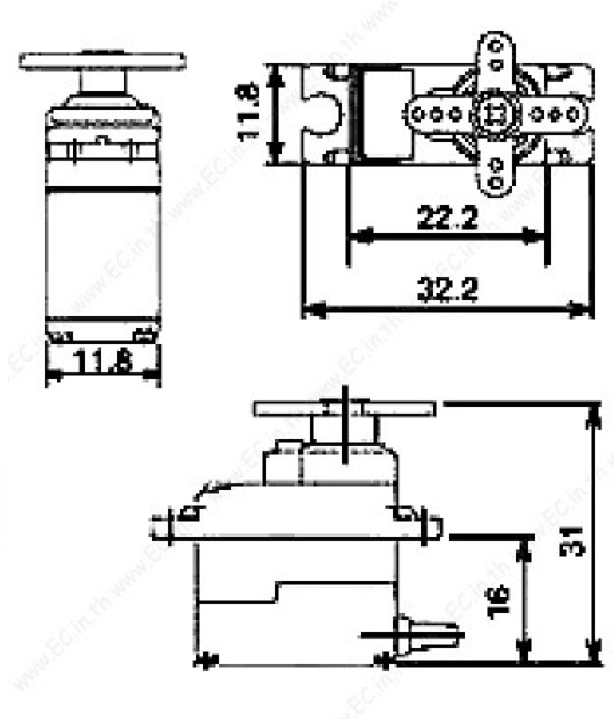
The MPU-6000/MPU-6050 is RoHS and Green compliant.

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

Environmental Declaration Disclaimer:

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

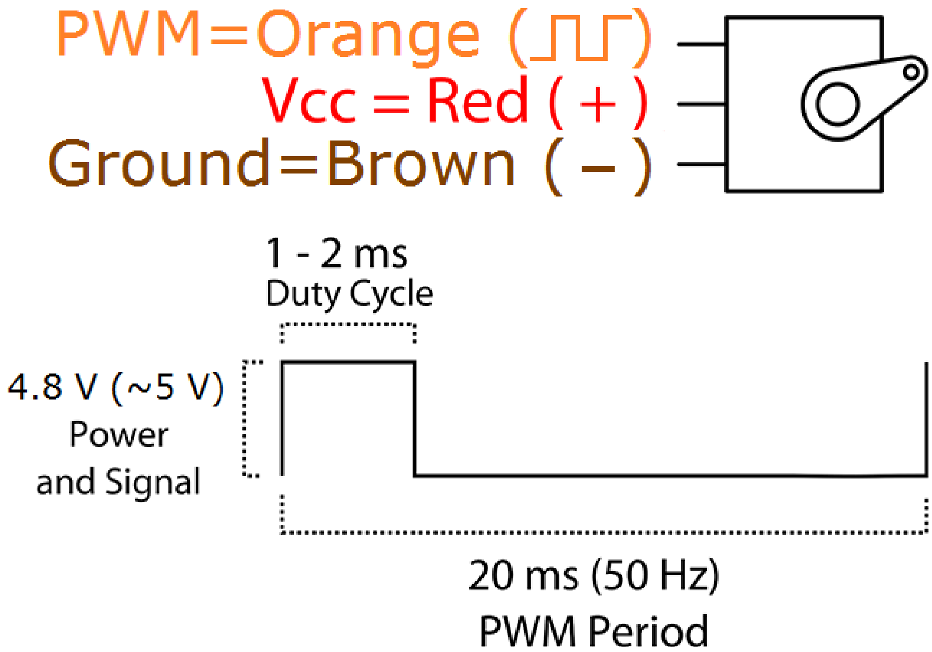
SG90 9 g Micro Servo



Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but *smaller*. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.

Specifications

- Weight: 9 g
- Dimension: 22.2 x 11.8 x 31 mm approx.
- Stall torque: 1.8 kgf·cm
- Operating speed: 0.1 s/60 degree
- Operating voltage: 4.8 V (~5V)
- Dead band width: 10 µs
- Temperature range: 0 ºC – 55 ºC



Position "0" (1.5 ms pulse) is middle, "90" (~2 ms pulse) is all the way to the right, "-90" (~1 ms pulse) is all the way to the left.

Winstar Display Co., LTD

CUSTOMER		
MODEL	WH1602B-YGK-CP	
APPROVAL	BY:	DATA:

SALES BY	APPROVED BY	CHECKED BY	PREPARED BY

華凌光電股份有限公司

住址：台中市中清路 163 號

NO.163 CHUNG CHING RD.
TAICHUNG, TAIWAN, R.O.C

TEL：886-4-4262208　FAX：886-4-4262207

Contents

1.Module Classification Information

2.Precautions in use of LCD Modules

3.General Specification

4.Absolute Maximum Ratings

5.Electrical Characteristics

6.Optical Characteristics

7.Interface Pin Function

8.Contour Drawing & Block Diagram

9.Function Description

10.Character Generator ROM Pattern

11.Instruction Table

12.Timing Characteristics

13.Initializing of LCM

14.Quality Assurance

15.Reliability

16.Backlight Information

1.Module Classification Information

W	H	1602	B	-	Y	G	K	-	CP
①	②	③	④		⑤	⑥	⑦		⑧

①　Brand：WINSTAR DISPLAY CORPORATION

②　Display Type：H→ Character Type, G→ Graphic Type

③　Display Font：Character 16 words, 2Lines.

④　Model serials no.

⑤	Backlight Type：	N→ Without backlight	A→ LED, Amber
		B→ EL, Blue green	R→ LED, Red
		D→ EL, Green	O→ LED, Orange
		W→ EL, White	G→ LED, Green
		F→ CCFL, White	
		Y→ LED, Yellow Green	
⑥	LCD Mode：	B→ TN Positive, Gray	T→ FSTN Negative
		N→ TN Negative,	
		G→ STN Positive, Gray	
		Y→ STN Positive, Yellow Green	
		M→ STN Negative, Blue	
		F→ FSTN Positive	
⑦	LCD Polarize Type/ Temperature range/ View direction	A→ Reflective, N.T, 6:00	H→ Transflective, W.T,6:00
		D→ Reflective, N.T, 12:00	K→ Transflective, W.T,12:00
		G→ Reflective, W. T, 6:00	C→ Transmissive, N.T,6:00
		J→ Reflective, W. T, 12:00	F→ Transmissive, N.T,12:00
		B→ Transflective, N.T,6:00	I→ Transmissive, W. T, 6:00
		E→ Transflective, N.T.12:00	L→ Transmissive, W.T,12:00
⑧	Special Code	CP : English and Cyrillic standard font	

2.Precautions in use of LCD Modules

(1)Avoid applying excessive shocks to the module or making any alterations or modifications to it.
(2)Don’t make extra holes on the printed circuit board, modify its shape or change the components of LCD module.
(3)Don’t disassemble the LCM.
(4)Don’t operate it above the absolute maximum rating.
(5)Don’t drop, bend or twist LCM.
(6)Soldering: only to the I/O terminals.
(7)Storage: please storage in anti-static electricity container and clean environment.

3.General Specification

Item	Dimension	Unit
Number of Characters	16 characters x 2 Lines	-
Module dimension	80.0 x 36.0 x 13.2(MAX)	mm
View area	66.0 x 16.0	mm
Active area	56.21 x 11.5	mm
Dot size	0.56 x 0.66	mm
Dot pitch	0.60 x 0.70	mm
Character size	2.96 x 5.56	mm
Character pitch	3.55 x 5.94	mm
LCD type	STN, Positive, Transflective, Grey	
Duty	1/16	
View direction	12 o’clock	
Backlight Type	LED Yellow Green	

4.Absolute Maximum Ratings

Item	Symbol	Min	Typ	Max	Unit
Operating Temperature	T_{OP}	0	-	+50	℃
Storage Temperature	T_{ST}	-20	-	+60	℃
Input Voltage	V_I	V_{SS}	-	V_{DD}	V
Supply Voltage For Logic	$V_{DD}-V_{SS}$	-0.3	-	7	V
Supply Voltage For LCD	$V_{DD}-V_0$	-0.3	-	13	V

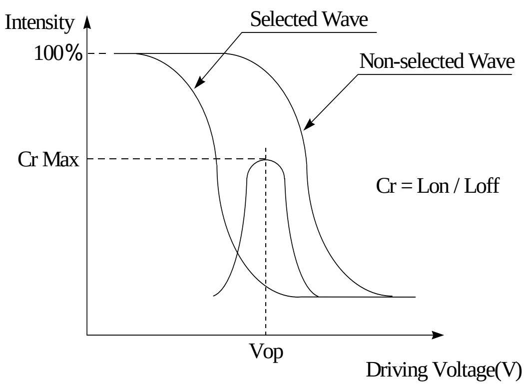
5.Electrical Characteristics

Item	Symbol	Condition	Min	Typ	Max	Unit
Supply Voltage For Logic	$V_{DD}-V_{SS}$	-	4.5	-	5.5	V
Supply Voltage For LCD	$V_{DD}-V_0$	Ta=-20℃ Ta=25℃ Ta=70℃	- - 3.4	3.8	5.2 - -	V V V
Input High Volt.	V_{IH}	-	2.2	-	V_{DD}	V
Input Low Volt.	V_{IL}	-	-	-	0.6	V
Output High Volt.	V_{OH}	-	2.4	-	-	V
Output Low Volt.	V_{OL}	-	-	-	0.4	V
Supply Current	I_{DD}	V_{DD}=5V	-	1.2	-	mA

6.Optical Characteristics

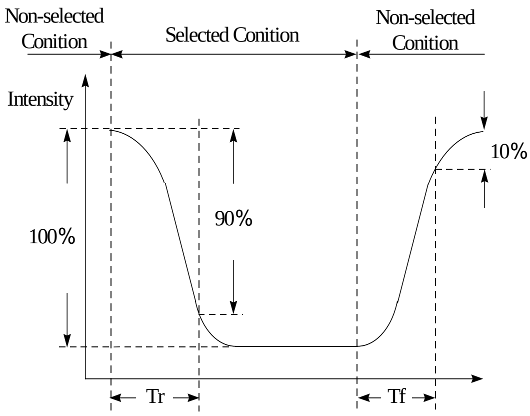
Item	Symbol	Condition	Min	Typ	Max	Unit
View Angle	(V)θ	CR≧ 2	10	-	105	deg
	(H)φ	CR≧ 2	-40	-	40	deg
Contrast Ratio	CR	-	-	3	-	-
Response Time	T rise	-	-	150	200	ms
	T fall	-	-	150	200	ms

Definition of Operation Voltage (Vop)



[positive type]

Definition of Response Time (Tr , Tf)



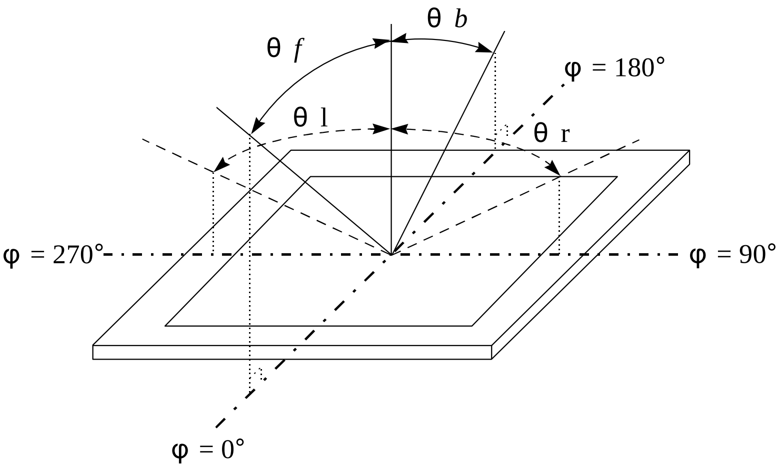
[positive type]

Conditions :

Operating Voltage : Vop   Viewing Angle(θ , φ) : 0° , 0°

Frame Frequency : 64 HZ   Driving Waveform : 1/N duty , 1/a bias

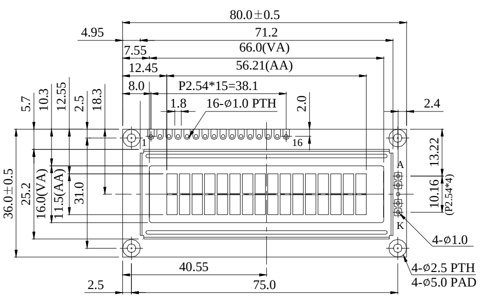
Definition of viewing angle(CR≧ 2)

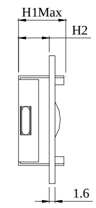


7.Interface Pin Function

Pin No.	Symbol	Level	Description
1	V_{SS}	0V	Ground
2	V_{DD}	5.0V	Supply Voltage for logic
3	VO	(Variable)	Operating voltage for LCD
4	RS	H/L	H: DATA, L: Instruction code
5	R/W	H/L	H: Read(MPU→ Module) L: Write(MPU→ Module)
6	E	H,H→ L	Chip enable signal
7	DB0	H/L	Data bit 0
8	DB1	H/L	Data bit 1
9	DB2	H/L	Data bit 2
10	DB3	H/L	Data bit 3
11	DB4	H/L	Data bit 4
12	DB5	H/L	Data bit 5
13	DB6	H/L	Data bit 6
14	DB7	H/L	Data bit 7
15	A	-	LED +
16	K	-	LED -

8.Contour Drawing &Block Diagram

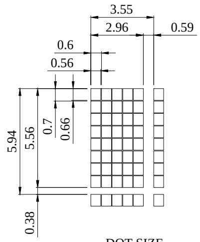




PIN NO.	SYMBOL
1	Vss
2	Vdd
3	Vo
4	RS
5	R/$\overline{W}$
6	E
7	DB0
8	DB1
9	DB2
10	DB3
11	DB4
12	DB5
13	DB6
14	DB7
15	A/Vee
16	K

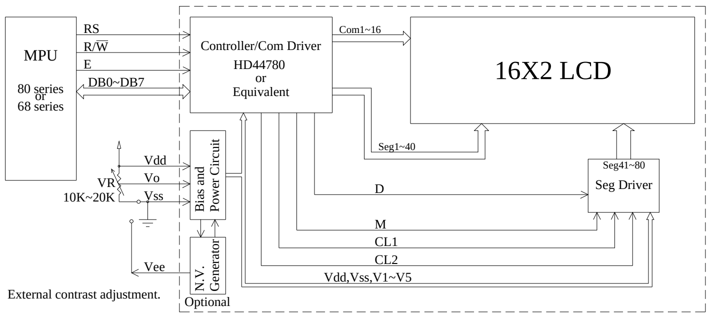
LED H/L B/L

	High	
H1	13.2	
H2	8.6	



DOT SIZE

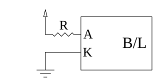
The non-specified tolerance of dimension is ±0.3mm.



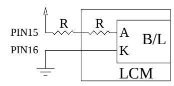
External contrast adjustment.

LED B/L Drive Method

1.Drive from A,K

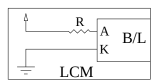


2.Drive from pin15, pin16



(Will never get Vee output from pin15)

3.Drive from Vdd,Vss



(Contrast performance may go down.)

Recommanded Value

V_{LED}= 4.2V, I_{LED}= 130mA

R= 4.7Ω (1/2 Watt)

Character located	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DDRAM address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
DDRAM address	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F

9.Function Description

The LCD display Module is built in a LSI controller, the controller has two 8-bit registers, an instruction register (IR) and a data register (DR).
The IR stores instruction codes, such as display clear and cursor shift, and address information for display data RAM (DDRAM) and character generator (CGRAM). The IR can only be written from the MPU. The DR temporarily stores data to be written or read from DDRAM or CGRAM. When address information is written into the IR, then data is stored into the DR from DDRAM or CGRAM. By the register selector (RS) signal, these two registers can be selected.

RS	R/W	Operation
0	0	IR write as an internal operation (display clear, etc.)
0	1	Read busy flag (DB7) and address counter (DB0 to DB7)
1	0	Write data to DDRAM or CGRAM (DR to DDRAM or CGRAM)
1	1	Read data from DDRAM or CGRAM (DDRAM or CGRAM to DR)

Busy Flag (BF)
When the busy flag is 1, the controller LSI is in the internal operation mode, and the next instruction will not be accepted. When RS=0 and R/W=1, the busy flag is output to DB7. The next instruction must be written after ensuring that the busy flag is 0.

Address Counter (AC)
The address counter (AC) assigns addresses to both DDRAM and CGRAM

Display Data RAM (DDRAM)
This DDRAM is used to store the display data represented in 8-bit character codes. Its extended capacity is 80×8 bits or 80 characters. Below figure is the relationships between DDRAM

High bits Low bits

AC (hexadecimal)

AC6	AC5	AC4	AC3	AC2	AC1	AC0

Example: DDRAM addresses 4E

1	0	0	1	1	1	0

addresses and positions on the liquid crystal display.

Display position DDRAM address

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F

2-Line by 16-Character Display

Character Generator ROM (CGROM)

The CGROM generate 5×8 dot or 5×10 dot character patterns from 8-bit character codes. See Table 2.

Character Generator RAM (CGRAM)

In CGRAM, the user can rewrite character by program.   For 5×8 dots, eight character patterns can be written, and for 5×10 dots, four character patterns can be written.

Write into DDRAM the character code at the addresses shown as the left column of table 1. To show the character patterns stored in CGRAM.

Relationship between CGRAM Addresses, Character Codes (DDRAM) and Character patterns

Table 1.

For 5 * 8 dot character patterns

Character Codes (DDRAM data)	CGRAM Address	Character Patterns (CGRAM data)	
7 6 5 4 3 2 1 0 (High … Low)	5 4 3 2 1 0 (High … Low)	7 6 5 4 3 2 1 0 (High … Low)	
	0 0 0	* * * ■ ■ ■ ■ 0	
	0 0 1	* * * ■ 0 0 0 ■	
	0 1 0	* * * ■ 0 0 0 ■	
	0 1 1	* * * ■ ■ ■ ■ 0	Character pattern(1)
0 0 0 0 * 0 0 0	0 0 0 1 0 0	* * * ■ 0 ■ 0 0	
	1 0 1	* * * ■ 0 0 ■ 0	
	1 1 0	* * * ■ 0 0 0 ■	
	1 1 1	* * * 0 0 0 0 0	Cursor pattern
	0 0 0	* * * ■ 0 0 0 ■	
	0 0 1	* * * 0 ■ 0 ■ 0	
	0 1 0	* * * ■ ■ ■ ■ ■	
	0 1 1	* * * 0 0 ■ 0 0	Character pattern(2)
0 0 0 0 * 0 0 1	0 0 1 1 0 0	* * * ■ ■ ■ ■ ■	
	1 0 1	* * * 0 0 ■ 0 0	
	1 1 0	* * * 0 0 ■ 0 0	
	1 1 1	* * * 0 0 0 0 0	Cursor pattern
	0 0 0	* * *	
	0 0 1		
0 0 0 0 * 1 1 1	1 1 1 1 0 0		
	1 0 1		
	1 1 0		
	1 1 1	* * *	

For 5 * 10 dot character patterns

Character Codes (DDRAM data)	CGRAM Address	Character Patterns (CGRAM data)	
7 6 5 4 3 2 1 0 (High … Low)	5 4 3 2 1 0 (High … Low)	7 6 5 4 3 2 1 0 (High … Low)	
	0 0 0 0	* * * 0 0 0 0 0	
	0 0 0 1	* * * 0 0 0 0 0	
	0 0 1 0	* * * ■ 0 ■ ■ 0	
	0 0 1 1	* * * ■ ■ 0 0 ■	
	0 1 0 0	* * * ■ 0 0 0 ■	
0 0 0 0 * 0 0 0	0 0 0 1 0 1	* * * ■ 0 0 0 ■	
	0 1 1 0	* * * ■ ■ ■ ■ 0	Character pattern
	0 1 1 1	* * * ■ 0 0 0 0	
	1 0 0 0	* * * ■ 0 0 0 0	
	1 0 0 1	* * * ■ 0 0 0 0	
	1 0 1 0	* * * 0 0 0 0 0	Cursor pattern
	1 1 1 1	* * * * * * * *	

■ : " High "

10.Character Generator ROM Pattern

Table.2

Upper 4 bit / Lower 4 bit	LLLL	LLLH	LLHL	LLHH	LHLL	LHLH	LHHL	LHHH	HLLL	HLLH	HLHL	HLHH	HHLL	HHLH	HHHL	HHHH
LLLL	CG RAM (1)			0	@	P	`	p			Б	Ю	ч	.	Д	[illegible]
LLLH	CG RAM (2)		!	1	A	Q	a	q			Г	Я	ш	[illegible]	Ц	[illegible]
LLHL	CG RAM (3)		"	2	B	R	b	r			Ё	б	ъ	[illegible]	Щ	[illegible]
LLHH	CG RAM (4)		#	3	C	S	c	s			Ж	в	ы	[illegible]	д	[illegible]
LHLL	CG RAM (5)		$	4	D	T	d	t			З	г	ь	[illegible]	ф	[illegible]
LHLH	CG RAM (6)		%	5	E	U	e	u			И	ё	э	[illegible]	ц	[illegible]
LHHL	CG RAM (7)		&	6	F	V	f	v			Й	ж	ю	[illegible]	щ	[illegible]
LHHH	CG RAM (8)		'	7	G	W	g	w			Л	з	я	[illegible]	´	[illegible]
HLLL	CG RAM (1)		(	8	H	X	h	x			П	и	«	Π	¨	[illegible]
HLLH	CG RAM (2)		)	9	I	Y	i	y			У	й	»	†	[illegible]	[illegible]
HLHL	CG RAM (3)		*	:	J	Z	j	z			Ф	к	„	[illegible]	é	[illegible]
HLHH	CG RAM (4)		+	;	K	[	k	10			Ч	л	"	[illegible]	[illegible]	[illegible]
HHLL	CG RAM (5)		,	<	L	¥	l	12			Ш	м	№	[illegible]	ü	[illegible]
HHLH	CG RAM (6)		-	=	M	]	m	15			Ъ	н	¿	[illegible]	[illegible]	[illegible]
HHHL	CG RAM (7)		.	>	N	^	n	[illegible]			Ы	п	ƒ	[illegible]	¤	[illegible]
HHHH	CG RAM (8)		/	?	O	_	o	[illegible]			Э	т	£	[illegible]	○	█

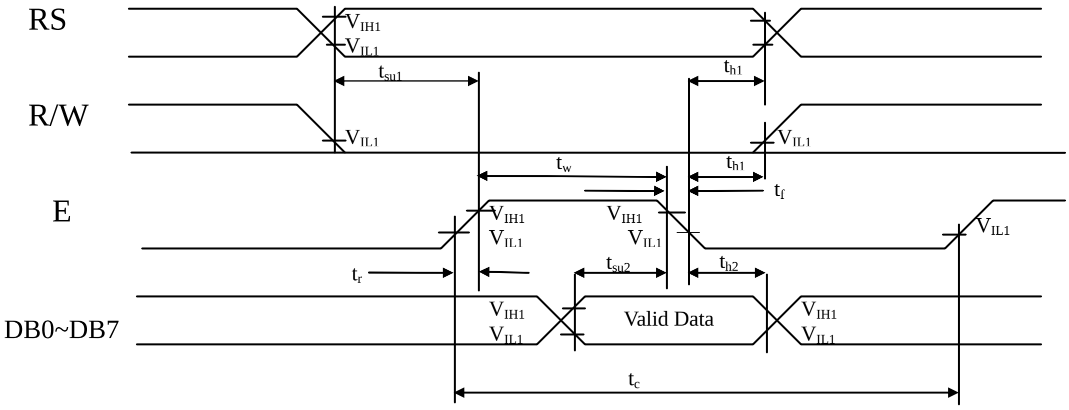
11.Instruction Table

Instruction	Instruction Code										Description	Execution time (fosc=270Khz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Write “00H” to DDRAM and set DDRAM address to “00H” from AC	1.53ms
Return Home	0	0	0	0	0	0	0	0	1	-	Set DDRAM address to “00H” from AC and return cursor to its original position if shifted. The contents of DDRAM are not changed.	1.53ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	SH	Assign cursor moving direction and enable the shift of entire display.	39μ s
Display ON/OFF Control	0	0	0	0	0	0	1	D	C	B	Set display (D), cursor (C), and blinking of cursor (B) on/off control bit.	39μ s
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	-	-	Set cursor moving and display shift control bit, and the direction, without changing of DDRAM data.	39μ s
Function Set	0	0	0	0	1	DL	N	F	-	-	Set interface data length (DL:8-bit/4-bit), numbers of display line (N:2-line/1-line)and, display font type (F:5×11 dots/5×8 dots)	39μ s
Set CGRAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter.	39μ s
Set DDRAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in address counter.	39μ s
Read Busy Flag and Address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Whether during internal operation or not can be known by reading BF. The contents of address counter can also be read.	0μ s
Write Data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM).	43μ s
Read Data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM).	43μ s

* ” - ” : don’t care

12.Timing Characteristics

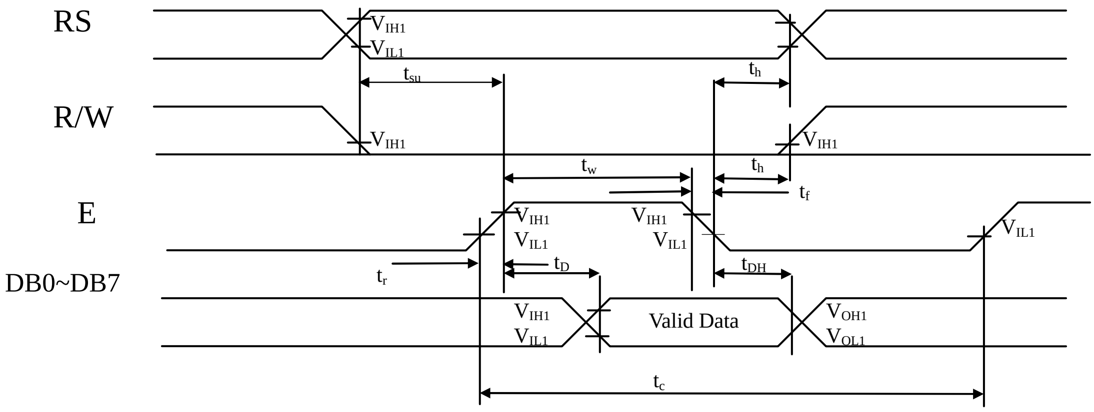
12.1 Write Operation



(V_{DD}=4.5V~5.5V , Ta=-30~+85℃)

Mode	Characteristic	Symbol	Min.	Typ.	Max.	Unit
Write Mode	**E cycle Time**	**tc**	**500**	**-**	**-**	**ns**
	E Rise/Fall Time	**t_R, t_F**	**-**	**-**	**20**	
	E Pulse Width (High, Low)	**tw**	**230**	**-**	**-**	
	R/W and RS Setup Time	**tsu1**	**40**	**-**	**-**	
	R/W and RS Hold Time	**t_{H1}**	**10**	**-**	**-**	
	Data Setup Time	**tsu2**	**80**	**-**	**-**	
	Data Hold Time	**t_{H2}**	**10**	**-**	**-**	

12.2 Read Operation



(V_{DD}=4.5V~5.5V , Ta=-30~+85℃)

Mode	Characteristic	Symbol	Min.	Typ.	Max.	Unit
Read Mode	**E cycle Time**	**tc**	**500**	**-**	**-**	**ns**
	E Rise/Fall Time	**t_R, t_F**	**-**	**-**	**20**	
	E Pulse Width (High, Low)	**tw**	**230**	**-**	**-**	
	R/W and RS Setup Time	**tsu**	**40**	**-**	**-**	
	R/W and RS Hold Time	**t_H**	**10**	**-**	**-**	
	Data Output Delay Time	**t_D**	**-**	**-**	**120**	
	Data Hold Time	**t_{DH}**	**5**	**-**	**-**	

13.Initializing of LCM

Power on

Wait for more than 15 ms after V_{CC} rises to 4.5 V

BF can not be checked before this instruction.

RS	R/$\overline{W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	*	*	*	*

Function set (Interface is 8 bits long.)

Wait for more than 4.1 ms

BF can not be checked before this instruction.

RS	R/$\overline{W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	*	*	*	*

Function set (Interface is 8 bits long.)

Wait for more than 100 μs

BF can not be checked before this instruction.

RS	R/$\overline{W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	*	*	*	*

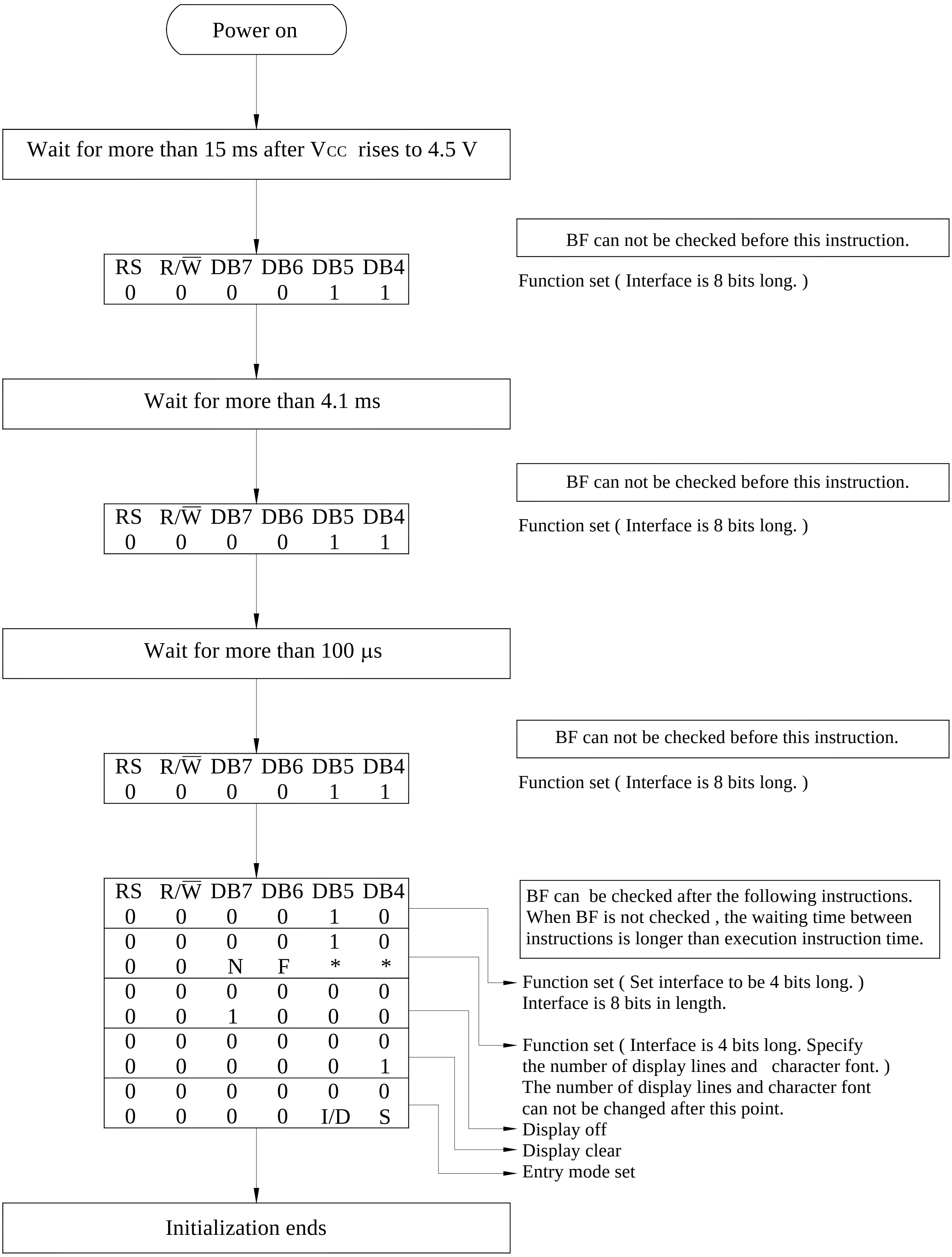
Function set (Interface is 8 bits long.)

BF can be checked after the following instructions.
When BF is not checked , the waiting time between instructions is longer than execution instruction time.

RS	R/$\overline{W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
0	0	0	0	1	1	N	F	*	*	Function set (Interface is 8 bits long. Specify the number of display lines and font.) The number of display lines and character font can not be changed after this point.
0	0	0	0	0	0	1	0	0	0	Display off
0	0	0	0	0	0	0	0	0	1	Display clear
0	0	0	0	0	0	0	1	I/D	S	Entry mode set

Initialization ends

8-Bit Ineterface



4-Bit Ineterface

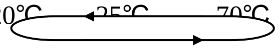
14.Quality Assurance

Screen Cosmetic Criteria

Item	Defect	Judgment Criterion	Partition
1	Spots	A)Clear Size: d mm — Acceptable Qty in active area d ≦ 0.1 — Disregard 0.1<d≦ 0.2 — 6 0.2<d≦ 0.3 — 2 0.3<d — 0 Note: Including pin holes and defective dots which must be within one pixel size. B)Unclear Size: d mm — Acceptable Qty in active area d ≦ 0.2 — Disregard 0.2<d≦ 0.5 — 6 0.5<d≦ 0.7 — 2 0.7<d — 0	Minor
2	Bubbles in Polarize	Size: d mm — Acceptable Qty in active area d≦ 0.3 — Disregard 0.3<d≦ 1.0 — 3 1.0<d≦ 1.5 — 1 1.5<d — 0	Minor
3	Scratch	In accordance with spots cosmetic criteria. When the light reflects on the panel surface, the scratches are not to be remarkable.	Minor
4	Allowable Density	Above defects should be separated more than 30mm each other.	Minor
5	Coloration	Not to be noticeable coloration in the viewing area of the LCD panels. Back-light type should be judged with back-light on state only.	Minor

15.Reliability

Content of Reliability Test

Environmental Test			
Test Item	**Content of Test**	**Test Condition**	**Applicable Standard**
High Temperature storage	Endurance test applying the high storage temperature for a long time.	80℃ 200hrs	——
Low Temperature storage	Endurance test applying the high storage temperature for a long time.	-30℃ 200hrs	——
High Temperature Operation	Endurance test applying the electric stress (Voltage & Current) and the thermal stress to the element for a long time.	70℃ 200hrs	——
Low Temperature Operation	Endurance test applying the electric stress under low temperature for a long time.	-20℃ 200hrs	——
High Temperature/ Humidity Storage	Endurance test applying the high temperature and high humidity storage for a long time.	80℃,90%RH 96hrs	——
High Temperature/ Humidity Operation	Endurance test applying the electric stress (Voltage & Current) and temperature / humidity stress to the element for a long time.	70℃,90%RH 96hrs	——
Temperature Cycle	Endurance test applying the low and high temperature cycle.  -20℃ 25℃ 70℃ 30min 5min 30min 1 cycle	-20℃/70℃ 10 cycles	——
Mechanical Test			
Vibration test	Endurance test applying the vibration during transportation and using.	10~22Hz→ 1.5mmp-p 22~500Hz→ 1.5G Total 0.5hrs	——
Shock test	Constructional and mechanical endurance test applying the shock during transportation.	50G Half sign wave 11 msedc 3 times of each direction	——
Atmospheric pressure test	Endurance test applying the atmospheric pressure during transportation by air.	115mbar 40hrs	——
Others			
Static electricity test	Endurance test applying the electric stress to the terminal.	VS=800V,RS=1.5kΩ CS=100pF 1 time	——

***Supply voltage for logic system=5V. Supply voltage for LCD system =Operating voltage at 25℃

16.Backlight Information

Specification

PARAMETER	SYMBOL	MIN	TYP	MAX	UNIT	TEST CONDITION
Supply Current	**ILED**	–	**130**	–	**mA**	**V=4.2V**
Supply Voltage	**V**	-	**4.2**	**4.6**	**V**	-
Reverse Voltage	**VR**	-	-	**8**	**V**	-
Luminous Intensity	**IV**	**10**	-	-	**CD/M^2**	**ILED=130mA**
Wave Length	**λ p**	–	**573**	-	**nm**	**ILED=130mA**
Life Time	-	-	**100000**	-	**Hr.**	**V≦ 4.6V**
Color	**Yellow Green**					