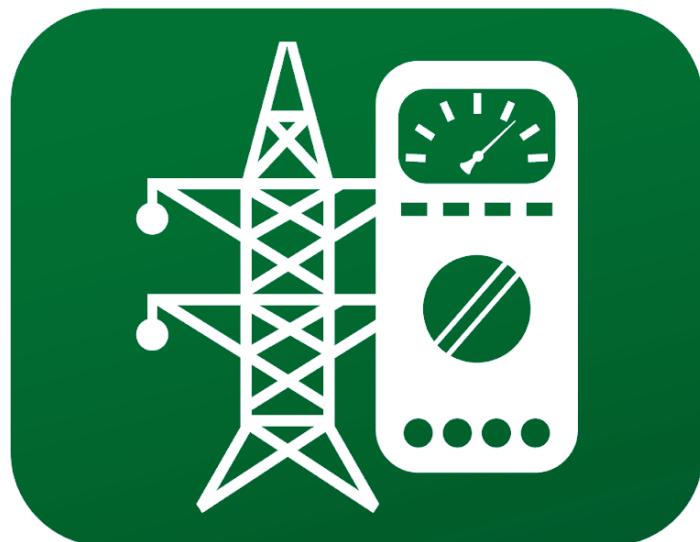


INSTITUTO LUIS BUENO CRESPO

DEPARTAMENTO DE ELECTRÓNICA



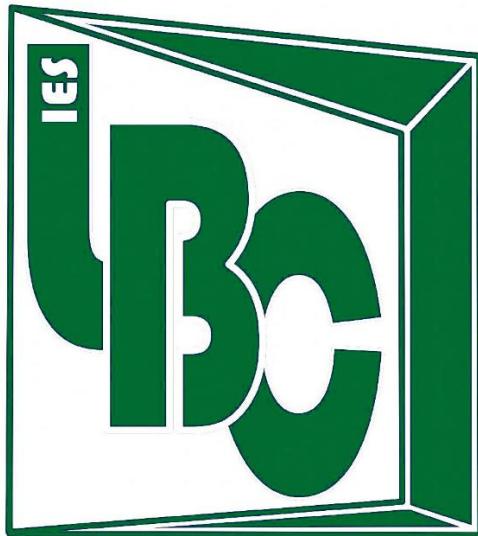
Sorena (Analizador de acústica)

Para acceder al Título de

**Técnico Superior en Sistemas de Telecomunicaciones e
Informáticos.**

Autor: Juan Miguel Acosta Ortega
Director: Jose Luis Vázquez Castro

06/2021



PROYECTO FIN DE CICLO

AUTOR: Juan Miguel Acosta Ortega

TÍTULO: Sorena (Analizador de acústica).

TRIBUNAL:

Don . Jose Luis Vázquez Castro
Don . Jose Antonio Ruiz Fernández
Don . Rafael García Jiménez
Don . Gerardo Avelino Benítez Gómez

CALIFICACIÓN: _____

Presentado en Armilla a 23 de junio de 2021.

Evaluado en Armilla a 23 de junio de 2021.

El Presidente

El Vocal

El Secretario



Índice

<u>ÍNDICE</u>	<u>5</u>
<u>ÍNDICE DE ILUSTRACIÓN</u>	<u>7</u>
<u>ÍNDICE DE TABLAS</u>	<u>8</u>
<u>RESUMEN</u>	<u>9</u>
<u>ABSTRACT</u>	<u>10</u>
<u>REFERENCIAS</u>	<u>11</u>
<u>GLOSARIO</u>	<u>12</u>
<u>CAPITULO 1. INTRODUCCIÓN</u>	<u>14</u>
PLANTEAMIENTO DEL PROBLEMA	<u>14</u>
JUSTIFICACIÓN	<u>14</u>
OBJETIVOS	<u>14</u>
ASPECTOS HISTÓRICOS	<u>15</u>
<u>CAPITULO 2. ESTADO DEL ARTE</u>	<u>16</u>
<u>CAPITULO 3. DESARROLLO DEL SISTEMA</u>	<u>18</u>
ALCANCE DEL PROYECTO	<u>18</u>
DESCRIPCIÓN DEL FUNCIONAMIENTO	<u>18</u>
<u>CAPITULO 4. PRUEBAS Y RESULTADOS</u>	<u>27</u>
<u>PRUEBAS Y SIMULACIONES</u>	<u>27</u>
<u>PRESUPUESTO</u>	<u>31</u>
<u>PLIEGO DE CONDICIONES</u>	<u>32</u>
PLIEGO DE CONDICIONES TÉCNICAS PARTICULARES.	<u>32</u>
PRESCRIPCIONES SOBRE LOS MATERIALES.	<u>32</u>

**Proyecto Fin de Ciclo****Índice****PLANOS****40****APÉNDICE****48****MANUAL DE INSTRUCCIONES****48****PROGRAMACIÓN FUENTE SORENA DIVIDIDA EN ARCHIVOS:****50****ANEXO (DATASHEETS)****77**



Índice de Ilustración

ILUSTRACIÓN 1: CONFIGURACIÓN TÍPICA ANALIZADOR DE ACÚSTICA.....	16
ILUSTRACIÓN 2: APLICACIÓN MÓVIL SONÓMETRO	17
ILUSTRACIÓN 3: SONÓMETRO	17
ILUSTRACIÓN 4: GENERADOR DE SEÑALES ORIENTADO A MICROCONTROLADORES.....	17
ILUSTRACIÓN 5: GENERADOR DE SEÑALES DE TALLER.....	17
ILUSTRACIÓN 6: LOGO LIQUID-MENU.....	18
ILUSTRACIÓN 7: DIAGRAMA DE FLUJO DE CALCULADORA DE TIEMPO DE REVERBERACIÓN.....	19
ILUSTRACIÓN 8: DIAGRAMA DE FLUJO DE GENERADOR DE TONOS	20
ILUSTRACIÓN 9: DIAGRAMA DE FLUJO DE SONÓMETRO	21
ILUSTRACIÓN 10: CONEXIÓN DE LOS COMPONENTES EN PROTOBOARD	24
ILUSTRACIÓN 11: INTERIOR DE LA CARCASA.....	25
ILUSTRACIÓN 12: EXTERIOR DE LA CARCASA.....	25
ILUSTRACIÓN 13: JERARQUÍA DE OBJETOS DE LA LIBRERÍA LIQUID-MENU.....	27
ILUSTRACIÓN 14: ARMÓNICOS DE UNA Onda CUADRADA	28
ILUSTRACIÓN 15: FRECUENCIAS COMPONENTES DE UNA SEÑAL	29
ILUSTRACIÓN 16: DIAGRAMA DE BLOQUES AD9833	33
ILUSTRACIÓN 17: ESKUEMA ELECTRÓNICO LM836.....	34
ILUSTRACIÓN 18: DIAGRAMA DE BLOQUES ADS1115	35
ILUSTRACIÓN 19: PINES Y SUS FUNCIONES ARDUINO MEGA	37
ILUSTRACIÓN 20: DIAGRAMA DE BLOQUES ARDUINO MEGA	38
ILUSTRACIÓN 21: DIAGRAMA DE BLOQUES PANTALLA LCD	39



Índice de Tablas

TABLA 1: TIEMPO DE REVERBERACIÓN SEGÚN RECINTO.....	14
TABLA 2: PINES UTILIZADOS ARDUINO	22
TABLA 3: SOFTWARE UTILIZADO DURANTE EL DESARROLLO	26
TABLA 4: RESUMEN DE PRESUPUESTO.....	31



Resumen

El analizador de acústica, también llamada “Sorena”, pretende ejercer las funciones de analizador de acústica, en este caso ofrece funcionalidades relacionadas con el sonido como lo son la función de sonómetro, de generador de tonos puros, de calculadora de tiempo de reverberación teórico, y próximamente de calculadora de tiempo de reverberación real.

El proyecto estará basado en el microcontrolador “Arduino”, y constará de diferentes módulos compatibles con esta tecnología que conseguirán un aparato con una interfaz amigable para el usuario, y plenamente funcional que además constará, entre otros componentes, de una pantalla LCD que mostrará principalmente un menú por estados que desembocará en los resultados pertinentes.



Abstract

The acoustics analyzer, also called "Sorena", performs the functions of an acoustic analyzer, in this case it offers functionalities related to sound such as the sound level meter function, the pure tone generator, the theoretical reverberation time calculator, and coming soon from real reverberation time calculator.

The project will be based on the “Arduino” microcontroller, and will consist of different modules compatible with this technology that will achieve a device with a user-friendly interface, and fully functional that will also consist, among other components, of an LCD screen that will mainly seek a menu by states that will lead to the relevant results.



Referencias

- [1] <https://guillermomaroto.thinkific.com/courses/take/cplusplusarduino/lessons/15293347-que-es-arduino-por-que-tengo-que-aprender-c> (Curso de Guillermo Maroto, es un curso bastante completo con clases gratuitas, fue muy útil para comprender cómo funciona un menu por estados, y cómo hacer los arreglos pertinentes para un uso sencillo y amigable).
- [2] <https://forum.arduino.cc/index.php?topic=505482.0> (Arduino Forum, es una herramienta muy útil en la que la gente deja reflejadas dudas y soluciones de errores específicos y generales de Arduino: Medir decibelios, la función map, qué librerías utilizar...).
- [3] <https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library/tree/master/examples> (Ejemplos varios de cómo utilizar la pantalla LCD con la tecnología I2C (Master and slave), además de las respectivas librerías y cómo integrarlas a Arduino).
- [4] <https://aprendiendoarduino.wordpress.com/2017/07/09/i2c/> (Información adicional sobre cómo funciona I2C).
- [5] <https://www.youtube.com/watch?v=dKPURQHRQ4k&list=WL&index=107&t=709s> (Ejemplo de sonómetro y analizador de espectro con Arduino).
- [6] https://www.youtube.com/watch?v=149TXc_bPeA&list=LL&index=3 (Otro ejemplo de sonómetro en Arduino).
- [7] <https://www.youtube.com/watch?v=LhQxDZDoeFY&list=LL&index=9> (Generador de funciones DDS con AD9833).
- [8] https://www.serina.es/empresas/cede_muestra/302/TEMA%20MUESTRA.pdf (Información acerca de sonorización en el presente y en el pasado).
- [9] <https://www.youtube.com/watch?v=lxlumFsKpg0&t=247s> (Introducción al uso de la librería "LiquidMenu").
- [10] <https://www.alldatasheet.com/view.jsp?Searchword=AD9833&sField=4> (Especificaciones técnicas sobre el modulo AD9833 → Generador de señales).
- [11] [https://www.ariat-tech.es/parts/ADI\(Analog-Devices,Inc.\)/AD9833BRMZ](https://www.ariat-tech.es/parts/ADI(Analog-Devices,Inc.)/AD9833BRMZ) (Página del fabricante del generador de señales AD9833 (DATASHEET)).
- [12] <https://www.diffusionmagazine.com/index.php/biblioteca/categorias/historia/331-historia-de-la-reverberacion> (Marco histórico de la reverberación en recintos cerrados).



Glosario

A

- **AD9833 DDS (Generador de funciones):** Como su nombre lo indica, un generador de funciones es un dispositivo que puede generar una forma de onda específica con una frecuencia específica en el momento de la configuración. Por ejemplo, considere que tiene un filtro LC para el que desea probar su respuesta de frecuencia de salida, puede hacerlo fácilmente con la ayuda de un generador de funciones.

C

- **Conversor ADC:** Un conversor o convertidor de señal analógica a digital (Conversor Analógico Digital, CAD; Analog-to-Digital Converter, ADC) es un dispositivo electrónico capaz de convertir una señal analógica, ya sea de tensión o corriente, en una señal digital mediante un cuantificador y codificándose en muchos casos en un código binario en particular. Donde un código es la representación única de los elementos, en este caso, cada valor numérico binario hace corresponder a un solo valor de tensión o corriente.

I

- **I2C:** Es un protocolo síncrono. I2C usa solo 2 cables, uno para el reloj (SCL) y otro para el dato (SDA). Esto significa que el maestro y el esclavo envían datos por el mismo cable, el cual es controlado por el maestro, que crea la señal de reloj. I2C no utiliza selección de esclavo, sino direccionamiento.

L

- **LCD:** Una **pantalla de cristal líquido** o **LCD** (sigla del inglés *liquid-crystal display*) es una pantalla delgada y plana formada por un número de píxeles en color o monocromos colocados delante de una fuente de luz o reflectora. A menudo se utiliza en dispositivos electrónicos de pilas, ya que utiliza cantidades muy pequeñas de energía eléctrica.

R

- **Rotary encoder:** Un codificador rotatorio, también llamado codificador del eje o generador de pulsos, suele ser un dispositivo electromecánico usado para convertir la posición angular de un eje a un código digital, lo que lo convierte en una clase de transductor.



S

- **SPI:** El Bus SPI (del inglés Serial Peripheral Interface) es un estándar de comunicaciones, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos. El bus de interfaz de periféricos serie o bus SPI es un estándar para controlar casi cualquier dispositivo electrónico digital que acepte un flujo de bits serie regulado por un reloj (comunicación sincrónica).

T

- **T60:** Se define el tiempo de reverberación como el tiempo necesario para que la intensidad de un sonido disminuya a la millonésima parte de su valor inicial o, lo que es lo mismo, que el nivel de intensidad acústica disminuya 60 decibelios por debajo del valor inicial del sonido.
En la práctica también se miden el T20 Y T30 (derivado del momento en que la curva de descomposición alcanza por primera vez 5 dB y 25 Db/5 y 35 dB por debajo del nivel inicial).



Capítulo 1. Introducción

Planteamiento del problema

Existen diversos problemas a la hora de sonorizar un recinto. En especial si es en interior se ha de tener en cuenta si los materiales del habitáculo son lo suficientemente absorbentes o reflectantes para la tarea que se quiera desempeñar dentro de el.

Es por ello que se puede obtener un dato con el que comprender la necesidad de cambiar o mejorar los materiales del recinto según su absorción o reflexión del sonido.

Este es el **tiempo de reverberación**.

Justificación

Sorena pretende paliar este problema ofreciendo una manera práctica y otra teórica de obtener ese dato.

Tabla 1: Tiempo de reverberación según recinto

Tiempo de reverberación normalizado en segundos (según tipo de recinto)	
Sala de conferencias	0.7-1.0
Cine	1.0-1.2
Teatro de ópera	1.2-1.5
Sala de conciertos (música de cámara)	1.3-1.7
Iglesia	2.0-3.0
Locutorio de radio	0.2-0.4

Objetivos

Sorena se centra por lo tanto en conseguir un aparato físico que consiga proporcionar al usuario todos los datos necesarios para comprobar si el recinto de adecua a las especificaciones requeridas:

- Sonómetro
- Generador de tonos
- Tiempo de reverberación práctico (INCOMPLETO)
- Tiempo de reverberación teórico

**MEMORIA**

Introducción

Aspectos históricos

La reverberación es un fenómeno acústico generado por la suma total de las reflexiones de un recinto cerrado que son percibidas por el oyente en una variación de tiempo, su duración y coloración depende de la distancia que hay entre el oyente y la fuente sonora y de las superficies que conforman el espacio.

La reverberación ha existido durante muchos años en ambientes cerrados. En el año 20 A.C., el arquitecto romano Marco Vitruvio Polión describe en su única obra de arquitectura varios diseños con el fin de mejorar la acústica de los antiguos teatros romanos. Una de sus propuestas era el uso de vasijas de bronce afinadas ubicadas debajo de las sillas, las cuales actuaban como resonadores sintonizados para frecuencias bajas y medias.

Luego, en 1640, el matemático francés Marin Mersenne halló el tiempo de retorno del sonido en el aire por medio del eco. Lo logró calculando la velocidad del sonido con un 10% de error. En el año de 1895, el Sr. Wallace Clement Sabine investigó la reverberación para mejorar las características acústicas de la sala Fogg Lecture Hall de la Universidad de Harvard. Ésta famosa, por poseer una acústica muy deficiente para el trabajo discursivo que se necesitaba allí; fue el escenario de trabajo para Sabine y su equipo durante tres años. Sabine, logró disminuir la reverberación del Fogg Lecture Hall al adherir en las paredes materiales absorbentes, lo que mejoró la acústica del lugar para el uso asignado.

Posterior a su triunfo en Harvard, Sabine asesoró la construcción del Boston Symphony Hall donde estableció la fórmula para predecir el tiempo de reverberación en recintos cerrados. Lastimosamente, en el año de 1900, en la inauguración de la sala, Sabine descubrió que el tiempo de reverberación final no era el que había pronosticado mediante su ecuación. Debido a este fracaso Sabine fue severamente criticado por los medios. Esto lo llevó a desistir en sus investigaciones y a recluirse en las aulas de clase como profesor. Cincuenta años más tarde las investigaciones de Sabine fueron revisadas exhaustivamente y se pudo comprobar que los datos del investigador eran correctos. Actualmente, el Boston Symphony Hall es una de las mejores salas del mundo en materia de acústica.

Varios investigadores, como Eyring y Millington, han tratado de mejorar la ecuación de Sabine sin modificar ampliamente la original. Lo que se ha logrado es agregar algunas variantes como la **temperatura y la humedad** para hallar los factores de absorción. En los años veinte, en materia de producción, muchos técnicos de grabación intentaron capturar la reverberación natural de teatros e iglesias, su acústica producía sonidos más claros, especialmente para la música clásica.

Hoy en día el tiempo de reverberación se calcula dentro del recinto utilizando un analizador acústico y (normalmente) una fuente de sonido omnidireccional para más precisión.

En conclusión, la reverberación es una característica crucial en la mayoría de los recintos cerrados, y que además estén relacionados con tema de acústica (colegios, cines, óperas...), y cuya medición se ha perfeccionado hasta el día de hoy.



Capítulo 2. Estado del arte

En el caso de analizadores de acústica, normalmente se sigue un estándar a la hora de hacer la medición práctica, es decir, se usa uno portátil, un amplificador, y un altavoz normalmente omnidireccional en forma de dodecaedro que permite una medición mucho más precisa.

Las diferencias entre los diferentes proveedores suelen ser pequeños despuntes en el hardware (forma del analizador acústico, calidad del amplificador y altavoz...), y el software, es decir, si integran generador de los diferentes tipos de ruido, si poseen indicadores de la calidad de medición, si pueden medir los espectros de nivel en la sala de emisión y recepción, si pueden o no generar informes de medición precisos, etc.

En general los analizadores de acústica suelen aspirar a un diseño ligero y ergonómico que facilita su agarre, sujeción y utilización con una sola mano, y a que sus aplicaciones abarquen medición del sonido y vibraciones, además de la evaluación del ruido ambiental.



Ilustración 1: Configuración típica analizador de acústica

De entre las empresas más punteras en el sector se han de destacar NTI Audio y Brüel y Kjaer, dos de las empresas más competentes en este ámbito y que poseen ambas los modelos más demandados de analizadores de acústica de recintos.

Además, el caso de los sonómetros es similar, la mayoría de los que se encuentran en el mercado son portátiles, pero en este caso, aunque se pierde calidad y precisión, también se pueden usar aplicaciones de móvil o software similar que ofrecen resultados similares.

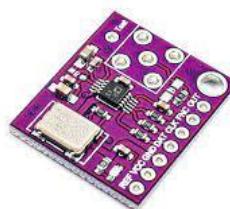
**MEMORIA**

Estado del Arte

**Ilustración 3:** Sonómetro**Ilustración 2:**
Aplicación móvil
sonómetro

En el mercado se encuentran muchos tipos de generadores de señales puras, que ofrecen señales eléctricas. En concreto, se utiliza para obtener señales periódicas (la tensión varía periódicamente en el tiempo) controlando su periodo (tiempo en que se realiza una oscilación completa) y su amplitud (máximo valor que toma la tensión de la señal). Los que se suelen usar en taller suelen costar los 100 euros.

Por último, en el caso de Sorena, el componente generador de señales es uno específico para este tipo de proyectos, pequeño y con las funcionalidades justas.

**Ilustración 5:** Generador de señales de taller**Ilustración 4:** Generador de señales orientado a microcontroladores



Capítulo 3. Desarrollo del sistema

Alcance del proyecto

Al estar finalizada la parte funcional del proyecto, Sorena será capaz de dar valores aproximados al técnico instalador sobre las características de los materiales de la sala (Tiempo de reverberación), como de calcular el T60 teóricamente, además de generar tonos a cualquier frecuencia y forma de onda posible, y medir el ruido ambiente en decibelios (dB).

Descripción del funcionamiento

Al inicio del desarrollo de un proyecto se han de plasmar las ideas y definir los objetivos a conseguir. En este caso “Sorena” pretende ser un aparato sencillo de manejar y funcional, a la par que portátil, que es una de sus propuestas de valor (ser un analizador de acústica de bajo precio con amplificador de sonido y altavoz y micrófono integrados).

Este proyecto en concreto consta de varias “piezas” que se han de ensamblar al final, es decir, está formado por una parte de **programación, de electrónica, y de diseño:**

- El pilar fundamental de Sorena es la programación, pues es única y creada desde cero especialmente para este proyecto.

A lo largo de todo el desarrollo se ha de elaborar el **menú** anteriormente nombrado por el que se desenvolverá la mayor parte del tiempo el usuario, y por el que llegará a los diferentes bloques de interés. Este se ha desarrollado finalmente usando la jerarquía de objetos (Sistema, menú, pantalla, línea) proporcionada por una librería prefabricada y con clases desarrolladas llamada “**LiquidMenu**”.



Ilustración 6: Logo liquid-menu



MEMORIA

Desarrollo del sistema

Las tareas que ha de desempeñar el aparato han de ser la de **Calculadora de Tiempo de reverberación, Generador de tonos puros, Sonómetro y Medidor del tiempo de reverberación real (INCOMPLETO)**.

Es conclusión, un aparato enfocado a la facilitación de recolección de datos en el ámbito de la acústica de interiores.

Para ello se necesitan diferentes programas ensamblados en el del menú:

1º Tiempo de reverberación teórico: Este programa es el más sencillo a nivel de lógica, pero más largo debido al modo de inserción de dígitos, las variables utilizadas, y las diferentes operaciones realizadas a lo largo del mismo.

Sigue una variable bandera para ir insertando datos mediante un método que involucra un contador y un encoder rotativo. De esta manera se evita poner una botonera con los diferentes posibles dígitos.

Cuando el usuario inserta todos los dígitos se calcula el T60 directamente, y se muestra el resultado por pantalla.

$$T60 = 0.161 * (\text{Volumen} / \text{Absorción total})$$

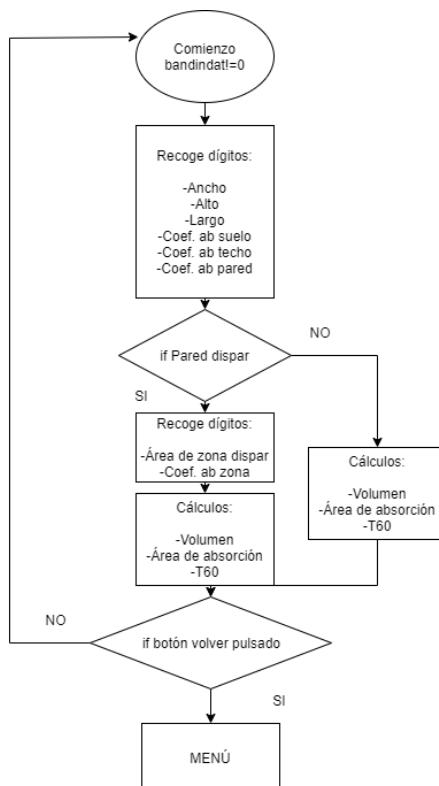


Ilustración 7: Diagrama de flujo de calculadora de tiempo de reverberación



MEMORIA

Desarrollo del sistema

La complejidad de este tipo de programas surge con la síntesis de la programación y la electrónica. Por ejemplo se han de evitar oscilaciones innecesarias en pantalla, que el encoder sume muy rápido y en dirección contraria a la esperada...

2º Generador de tonos: Este al contrario que el anterior, requiere un código más rebuscado, pues sumado a la muestra de datos por la pantalla de la LCD se ha de hacer una función que limpie dígitos no deseados constantemente, y que sólo se actualice la pantalla cuando haya alguna actualización de uno de los valores (multiplicador, función de onda, frecuencia...).

Además, se le han de sumar funciones que sólo aportan escrituras de registro que se encuentran en el Data-sheet del módulo generador de ondas (AD9833).

Todo lo demás relacionado con este programa tiene que ver con cómo seleccionar la frecuencia, la forma de onda, y el multiplicador anteriormente nombrados.

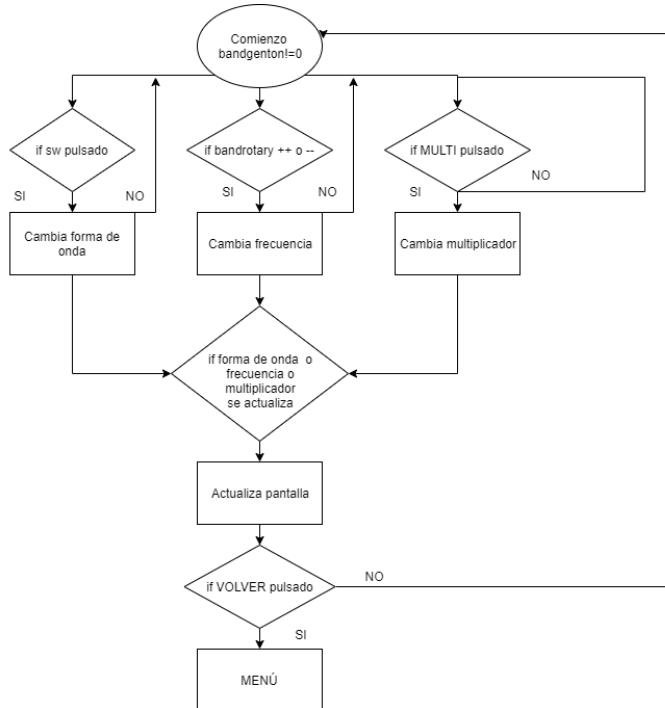


Ilustración 8: Diagrama de flujo de generador de tonos

3º Sonómetro: En este programa se llegan a los decibelios (sensación sonora) haciendo cuentas sencillas con las especificaciones técnicas de Arduino y las del micrófono.

Cuando manejamos sonidos lo primero que debemos entender es que la señal que obtenemos varía rápidamente y además tiene mucho ruido. De esta forma, el valor puntual de una medición no suele tener importancia, y deberemos realizar la integración en un

**MEMORIA**

Desarrollo del sistema

determinado periodo. Para ello lo habitual es trabajar con ventanas temporales, es decir, un intervalo en el que realizamos la medición.

Para ello, definimos una ventana temporal de 50 ms (SampleWindow), equivalente a una frecuencia de 20 Hz ($T=1/f$), y calculamos el máximo y mínimo registrado dentro de la ventana. A continuación mostramos el valor registrado por pantalla.

Se actualizan los máximos y los mínimos de la señal bruta recogida por el micrófono y se mapea entre 0-1023 valores que son los que Arduino puede recoger mediante sus entradas “analógicas”.

Mediante estas operaciones se llega a los decibelios, una unidad de medida del sonido no lineal:

```
picoapico = Max - Min; // Amplitud del sonido
```

```
volts = (picoapico * 5.0) / 1024; // Convertir a tensión
```

```
db = (20. * log(10)) * (volts); //Convertir a dB
```

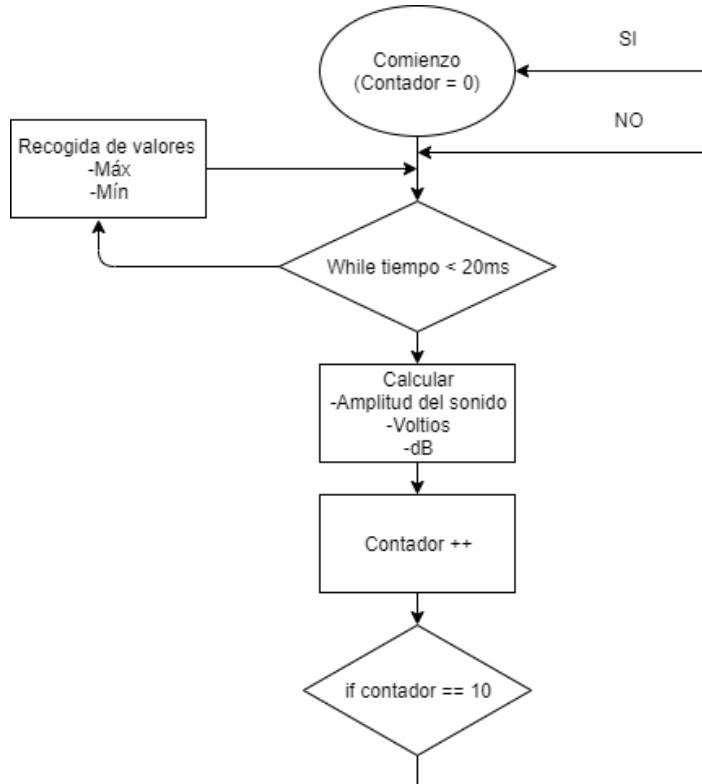


Ilustración 9: Diagrama de flujo de sonómetro

**MEMORIA**

Desarrollo del sistema

4º Tiempo de reverberación práctico (INCOMPLETO) :

Este programa pretendía calcular los tiempos de reverberación (T60, T30 Y T20) para las diferentes octavas del rango audible humano (64, 125, 250, 500, 1000, 2000, 4000, 8000 y 16000), sin embargo, el conversor analógico nativo del microcontrolador posee poca resolución y velocidad para esta tarea, y además el propio Arduino tarda demasiado en registrar los distintos valores recogidos en memoria.

Por lo tanto se necesitaría un conversor analógico-digital con más resolución y además más velocidad, propio de grabadores de voz.

Los demás programas son intuitivos y de muestra de datos por pantalla.

- El segundo pilar fundamental de este proyecto es el de la electrónica, sobre todo enfocada a los microcontroladores y su funcionamiento.

Por ejemplo, el diseño del menú estrictamente programado no podría ser el mismo para un proyecto que involucre también electrónica, pues los resultados, como más adelante se demostrarán en el apartado siguiente, podrían cambiar drásticamente.

El núcleo de este proyecto es el microcontrolador *Arduino*, y será utilizado como base para utilizar diferentes módulos orientados a él:

Tabla 2: Pines utilizados Arduino

Módulo electrónico	Pines utilizados
Rotary encoder	5 / 6 / 7 2/10/12
Pantalla LCD 20x4 (Módulo I2C)	SCL / SDA
Ad9833	38 / 52 / 51
Micrófono (Sound Sensor Module)	A0
Botones	8 / 9

En primer lugar, se usa una pantalla LCD de 20x4 con módulo I2C (para usar ese protocolo de comunicación “master-slave” y reducir el número de pines que necesita).

**MEMORIA**

Desarrollo del sistema

Esta es fácil de programar, y además solo necesita 2 pines más los de alimentación, sin ella el usuario no podría ver los resultados de las diferentes funciones ni moverse por ninguna interfaz sin el uso de un ordenador.

En segundo lugar para desenvolverse por el menú, usaremos 2 encoders rotativos (uno para el menú principal, y otro para los submenús que requieran cambiar o introducir valores), y dos botones (MULTI y VOLVER).

Se escoje esta opción porque es la más fácil y funcional de implementar para no tener que añadir una botonera con diferentes dígitos y flechas.

Y además Sorena cuenta con un micrófono electret para la función de sonómetro y escucha de los tonos puros, un altavoz direccional de 3W/8Ω, un conector Jack de 6.5mm con desconexión, un amplificador de baja potencia , un generador de tonos puros enfocado a este tipo de proyectos , resistencias para los pulsadores (1kΩ) y un posible conversor analógico digital para mayor calidad y resolución.

Como protocolos de comunicación usados a destacar están el protocolo I2C (Pines SDA-SCL), y SPI (Pines MOSI-SCK (51-52).

I2C utiliza dos cables para todo el proceso: SDA (datos en serie) y SCL (reloj en serie). El protocolo **I2C** puede admitir múltiples dispositivos esclavos, pero a **diferencia de SPI**, que solo admite un dispositivo maestro, **I2C** también puede admitir múltiples dispositivos maestros.

En conclusión, se han utilizado componentes enfocados para microcontroladores que faciliten las tareas de, por ejemplo, etapa de amplificación, moverse por el menú...



MEMORIA

Desarrollo del sistema

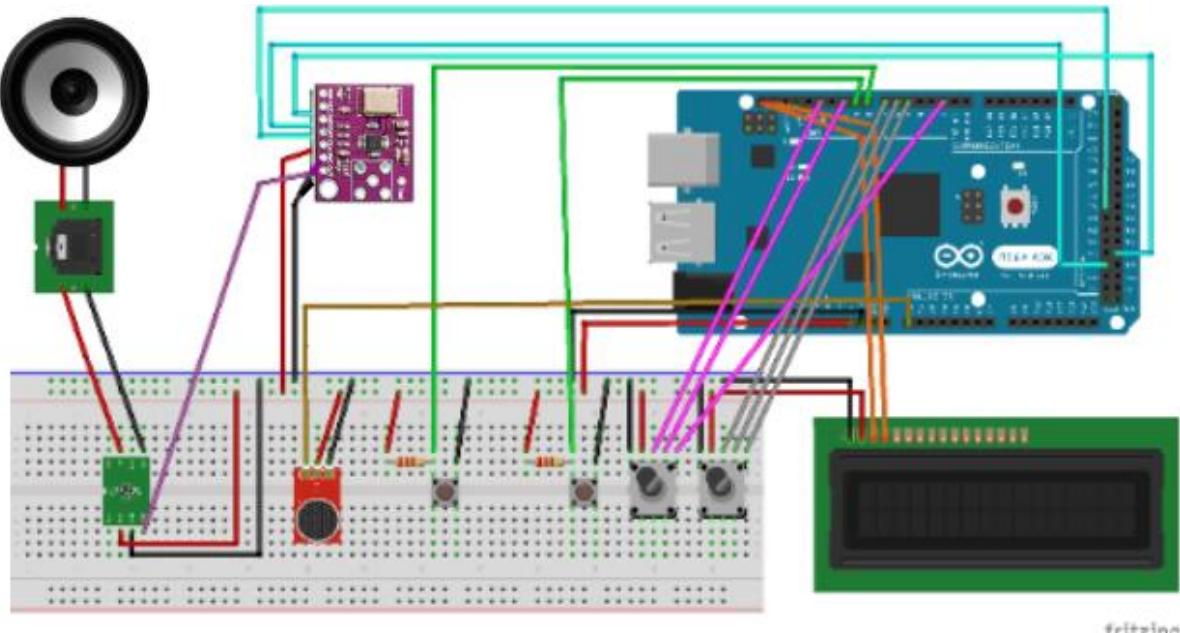


Ilustración 10: Conexión de los componentes en protoboard

- Por último pero no menos importante, se encuentra la etapa de diseño, donde se han de ensamblar todos los componentes en una carcasa propia y original para un uso más amigable.

Estos irán casi todos atornillados a la carcasa, o insertados a presión y fijados con silicona para que quede un producto compacto y profesional.

El único componente soldado y compuesto a medida para la carcasa fue el de los botones para que quedara con menos pines y más fijo.

La carcasa será realizada en un software específico de modelaje 3D, y estará fabricada pensando en todos los componentes:

**MEMORIA**

Desarrollo del sistema

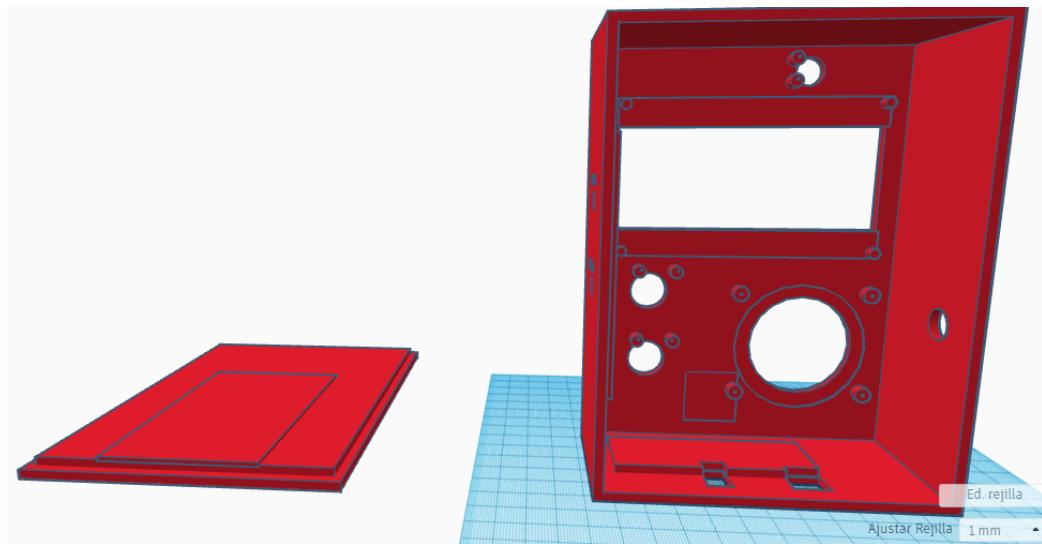


Ilustración 11: Interior de la carcasa

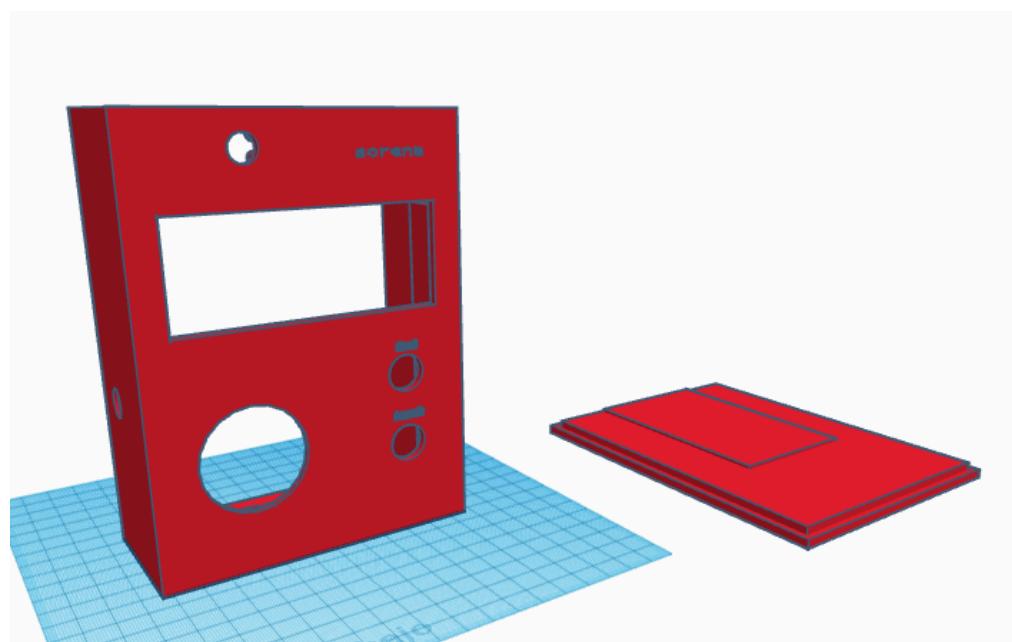


Ilustración 12: Exterior de la carcasa

**MEMORIA**

Desarrollo del sistema

Para poder terminar el proyecto se han utilizado diferentes software específicos para cada tarea:

Tabla 3: Software utilizado durante el desarrollo

Programas utilizados durante el desarrollo	
Programación	Arduino / Code-blocks
Modelaje y simulación de circuitos	Thinkercad
Planos (Pines)	Autocad Web
Imagen esquema conexión protoboard (Componentes)	Fritzing
Memoria y presentación	Word, PowerPoint



Capítulo 4. Pruebas y resultados

Pruebas y simulaciones

Durante el planteamiento y realización de este proyecto, han variado diferentes aspectos del mismo, tanto de programación, electrónica y componentes como de resultado final.

En la **primera etapa** del desarrollo, surgieron diferentes problemas provenientes de ámbitos dispares:

El marco principal de la programación de Sorena, además de poder medir el tiempo de reverberación, es la del propio menú, apartado que afecta directamente a la experiencia de uso del usuario, y esta precisamente, es la que ha sufrido cambios más drásticos durante el desarrollo.

En el inicio y planteamiento del sistema se quiso plantear un menú rápido y poco eficiente usando la estructura “switch case” o “if/else if”, sin embargo, al realizar pruebas con el rotary encoder y la pantalla LCD surgieron diversos problemas que más tarde desembocaron en un cambio radical de estructura del menú.

El primer fallo era que al pasar de caso en caso con una variable bandera para ello, la pantalla LCD parpadeaba indefinidamente por “la velocidad de dibujo que repetir el caso” suponía. Para solucionarlo se usó un “if” sumado a una variable específica para cada caso, especialmente para que el caso sólo se dibujara una vez en pantalla, pero desgraciadamente esto suponía no poder actualizar por pantalla una variable contador necesaria para la inserción de datos por parte del usuario.

Por todo esto se usó finalmente una librería ya hecha especial para pantallas LCD con módulo I2C y Arduino, que además funcionaba mediante jerarquía de objetos, es decir, creamos un objeto “menú”, que posee objetos arraigados llamados “pantalla”, que a su vez se componen de objetos

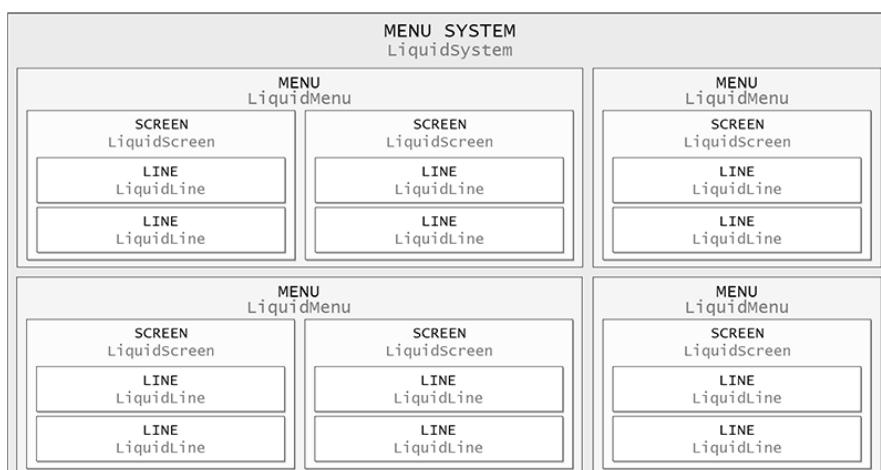


Ilustración 13:
Jerarquía de objetos
de la librería liquid-
menu

**MEMORIA**

Desarrollo del sistema

llamados “líneas”. Esta manera de realizar un menú es bastante más compleja, pero a su vez muchísimo más funcional.

El segundo cambio que se realizó en mitad del desarrollo fue el de soldar el módulo I2C a la pantalla LCD de 20x4 para así poder reducir los pines utilizados en Arduino, deshacernos de jumpers innecesarios en la simulación, además del potenciómetro, y conseguir así que la pantalla LCD funcionara debidamente.

El tercer error de implementación estuvo relacionado con **el micrófono** que se implementó en el sistema (electret). Es menos sensible que un micrófono profesional, y es por eso que se contempla la posibilidad de medir también el T20 (Tiempo de reverberación hasta que la señal cae 20 dB), que sería más preciso que el anterior.

El cuarto test fallido, es obviamente la poca potencia que el microcontrolador puede proporcionar directamente al altavoz direccional, es por eso que se le implementó una etapa de amplificación , y además un conector provisto de “desconector” para que cupiera la posibilidad de implementar un altavoz omnidireccional más preciso para la tarea que ocupa, medir el tiempo de reverberación.

La quinta implementación no pensada a priori es la de decidir **no utilizar sólo las características de Arduino** para crear tonos, pues Arduino es puramente digital, y sus ondas están compuestas por infinitos armónicos:

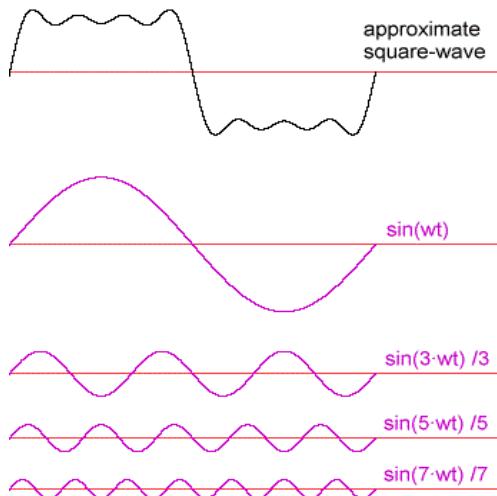


Ilustración 14: Armónicos de una onda cuadrada

Una alternativa para solucionar este problema sería el de utilizar filtros (pasabanda o pasobajo), o una caja de filtros, pero esto acarrearía tener un filtro disponible para cada frecuencia seleccionada, y más componentes para este fin, y no es una decisión sostenible.



MEMORIA

Desarrollo del sistema

Es por eso que se optó por utilizar un generador de señales, como lo es el AD9833 que además de ser más barato, es plenamente funcional y más sencillo de implementar.

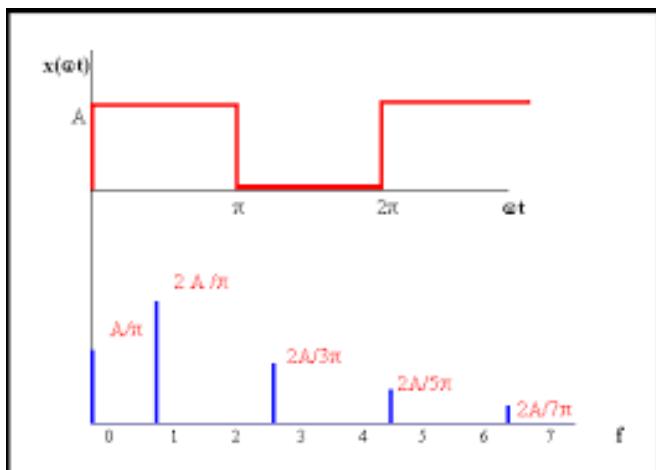


Ilustración 15: Frecuencias componentes de una señal

Además, **durante el proyecto y durante el ensamblamiento de todos los programas en uno**, es decir, después de añadir más variables al programa estructurado del menú, Arduino UNO de quedó sin memoria suficiente para desempeñar las diferentes funciones el programa con normalidad.

Es por eso que se usará finalmente el microcontrolador Arduino MEGA rev 3.

Como error de planteamiento se encuentra el de tener intención de realizar **un programa de tantas líneas en un mismo archivo** (organización nula). Esto se debe a que si hay algún error de compilación o de organización sería mucho más complicado de solventar que si el programa se dividiera en distintos archivos (facciones) que contuvieran partes específicas de cada uno de los pilares del programa principal.

Por ejemplo, **Sorena** se divide en tres archivos diferentes:

- Sorena.ino (Es el programa principal, y se encarga tanto de llamar a las funciones creadas en los diferentes archivos, como de soportar el void loop y void setup).
- clmenu.h (Todo lo relacionado con la jerarquía de objetos de la librería LiquidMenu y la creación de las funciones de cada línea y submenú).
- Gensen.h (Posee variables, funciones, y valores de registros necesarios para el generador de tonos).

En la recta final del desarrollo del proyecto hubo cambios decisivos sobre todo en el hardware del producto:

**MEMORIA**

Desarrollo del sistema

-Se cambió el altavoz que se eligió en primera instancia ($1W/8\Omega$) para colocar otro de ($3W/8\Omega$) que supuso una gran mejora en la calidad de ciertas mediciones. Por ende se tuvo que hacer otro diseño de la carcasa que cambiaría en medidas generales y en las del altavoz.

-Además para mejorar la resolución de Arduino (ADC → 10 bits “0-1023 valores de muestreo”), se puede implementar un ADC que sustituyera el nativo de 4 canales y 16 bits (0-65536 valores), y así la curva de decaimiento de los decibelios en la medida del tiempo de reverberación será menos abrupta.

Aunque la resolución del ADS1115 es de 16-bit, la realidad es que **no todos bits de los 16-bit se utilizan para expresar el valor de voltaje**.

La salida que obtienes del ADS1115 se conoce como **entero con signo** es decir, **uno de los bits** de la palabra de 16-bit se utiliza para **establecer el signo, positivo o negativo**.
Por lo tanto, **solo se utilizan 15-bit de los 16-bit** lo que significa que hay **32.768 valores** posibles (2^{15}).

El problema es que de esta manera **no se podrían medir los tiempos de reverberación mayores a 860Hz**. Así que finalmente no se implementa y **se sacrifica calidad por funcionalidad**.

Es por las limitaciones de resolución y velocidad de registro de valores en memoria que el apartado de tiempo de reverberación práctica queda incompleto para esta versión de Sorena.

Este proyecto se finalizó el día 23 de junio de 2021.

**PRESUPUESTO**

Presupuesto

Presupuesto

Tabla 4: Resumen de presupuesto

Presupuesto de componentes	
Arduino MEGA R3 (Elegoo)	10,89€
Pila + Cable con conector	3,25€
Rotary encoder + Adaptador KY-040	4,20€
Jumpers (Macho-Macho/Macho-Hembra)	3,34€
Pantalla LCD 20x4 1602A	4,89€
Módulo I2C para pantalla LCD serie PCF8574	2,87€
Botones	2,99€
Generador de señales AD9833	4,49€
Tornillería roscada variada (2mm,4mm...)	1,48€
Módulo amplificador de potencia de audio LM386	0,34€
Sound sensor module max4466	0,78€
Altavoz 3W/8Ohm	1,23€
Placas de circuito impreso	2,59€
Estaño circuitería	1,29€
Conversor analógico-digital ADS1115	4,65

Total: 49,28€



Pliego de Condiciones

Pliego de condiciones técnicas particulares.

Prescripciones sobre los materiales.

AD9833:

DESCRIPCIÓN GENERAL

El AD9833 es un generador de formas de onda programable de baja potencia capaz de producir salidas de onda sinusoidal, triangular y cuadrada. La generación de formas de onda es necesaria en varios tipos de aplicaciones de detección, actuación y reflectometría en el dominio del tiempo (TDR). La frecuencia y la fase de salida son programables por software, lo que permite una fácil sintonización. No se necesitan componentes externos. Los registros de frecuencia tienen 28 bits de ancho: con una frecuencia de reloj de 25 MHz, se puede lograr una resolución de 0,1 Hz; con una frecuencia de reloj de 1 MHz, el AD9833 se puede sintonizar a una resolución de 0,004 Hz.

El AD9833 se escribe a través de una interfaz en serie de 3 cables. Esta interfaz en serie funciona a velocidades de reloj de hasta 40 MHz y es compatible con los estándares de DSP y microcontroladores. El dispositivo funciona con una fuente de alimentación de 2,3 V a 5,5 V. El AD9833 tiene una función de apagado (SLEEP). Esta función permite apagar las secciones del dispositivo que no se están utilizando, minimizando así el consumo de corriente de la pieza. Por ejemplo, el DAC se puede apagar cuando se genera una salida de reloj.

El AD9833 está disponible en un paquete MSOP de 10 derivaciones.

CARACTERÍSTICAS

Frecuencia y fase programables digitalmente Consumo de energía de 12,65 mW a 3 V Rango de frecuencia de salida de 0 MHz a 12,5 MHz Resolución de 28 bits: 0,1 Hz a reloj de referencia de 25 MHz.

Salidas de onda sinusoidal, triangular y cuadrada Fuente de alimentación de 2,3 V a 5,5 V No se requieren componentes externos Interfaz SPI de 3 cables.

Rango de temperatura extendido: -40°C a $+105^{\circ}\text{C}$ Opción de apagado Paquete MSOP de 10 derivaciones AEC-Q100 calificado para aplicaciones automotrices.

**Pliego de Condiciones**

Pliego de condiciones

APLICACIONES

- Generación de estímulos de frecuencia / forma de onda
- Medición de flujo de líquido y gas
- Aplicaciones sensoriales: detección de proximidad, movimiento y defectos
- Pérdida / atenuación de línea
- Equipo médico y de prueba
- Generadores de barrido / reloj
- Aplicaciones de reflectometría en el dominio del tiempo (TDR)

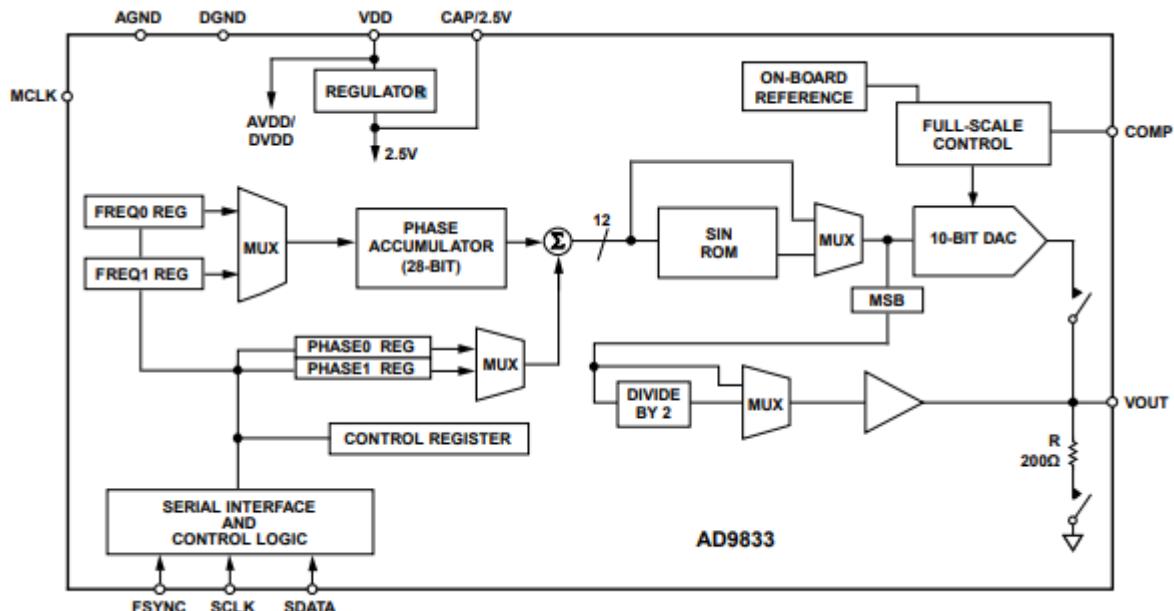


Ilustración 16: Diagrama de bloques AD9833

LM3861:**DESCRIPCIÓN GENERAL**

El LM836 UTC es un amplificador de potencia, diseñado para el uso de aplicaciones de bajo consumo genéricas. La ganancia es internamente establecida a 20 para mantener la parte exterior al mínimo, además de una resistencia y un condensador externo entre el pin 1 y el 8 que podrán incrementar la ganancia de cualquier valor desde 20 hasta 200.

Las entradas están conectadas a una tierra única mientras que las salidas automáticamente bajan a la mitad de la tensión de alimentación.

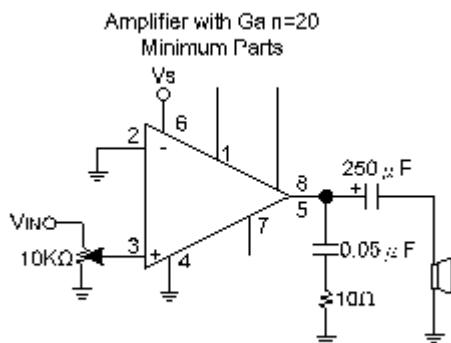
El consumo de energía en reposo es solo de 24 miliWatts cuando opera desde un soporte de 6 voltios, haciendo al ideal de la operación de batería del LM386.

**Pliego de Condiciones**

Pliego de condiciones

APLICACIONES

- Baterías
- Partes externas mínimas
- Rango de voltaje amplio (4V-12V)
- Baja potencia 315 mW de media

TYPICAL APPLICATIONS CIRCUIT**Ilustración 17: Esquema electrónico LM836****ADS1115:****DESCRIPCIÓN GENERAL**

Los ADS1113, ADS1114 y ADS1115 (ADS111x) son conversores analógicos a digitales (ADC) de precisión, baja potencia, 16 bits, compatibles con I2C, que se ofrecen en un paquete X2QFN-10 ultrapequeño y sin cables. Un paquete VSSOP-10. Los dispositivos ADS111x incorporan una referencia de voltaje de baja deriva y un oscilador. El ADS1114 y ADS1115 también incorporan un amplificador de ganancia programable (PGA) y un comparador digital. Estas características, junto con una amplia gama de suministros operativos, hacen que el ADS111x sea muy adecuado para aplicaciones de medición de sensores con limitaciones de espacio y energía.

El ADS111x convierte a velocidades de datos de hasta 860 muestras por segundo (SPS). El PGA ofrece rangos de entrada de ± 256 mV a ± 6.144 V, lo que permite mediciones precisas de señales grandes y pequeñas. El ADS1115 cuenta con un multiplexor de entrada (MUX) que permite dos medidas de entrada diferenciales o cuatro de un solo extremo. Utilice el comparador digital en el ADS1114 y ADS1115 para la detección de subtensión y sobretensión.

El ADS111x funciona en modo de conversión continua o en modo de disparo único. Los dispositivos se apagan automáticamente después de la conversión al modo de disparo único; por lo tanto, el consumo de energía se reduce significativamente durante los períodos de inactividad.



PARÁMETROS

- Resolución (bits): 16
- Número de canales de entrada: 4
- Frecuencia de muestreo (máx.) (KSPS): 0,86
- Tipo de interfaz: I2C
- Arquitectura: Delta-Sigma
- Tipo de entrada: Diferencial, de un solo extremo
- Configuración: multicanal multiplexada
- Catálogo: de calificaciones
- Modo de referencia: Int
- Rango de entrada: (Max) (V) 5,5
- Rango de entrada: (Min) (V) 0
- Características :comparador, oscilador, PGA

CARACTERÍSTICAS

- Paquete X2QFN ultra pequeño:
- 2 mm × 1,5 mm × 0,4 mm
- Amplio rango de suministro: 2,0 V a 5,5 V
- Bajo consumo de corriente: 150 µA
- (Modo de conversión continua)
- Tasa de datos programable:
- 8 SPS a 860 SPS
- Asentamiento de ciclo único
- Referencia interna de voltaje de baja deriva
- Oscilador interno

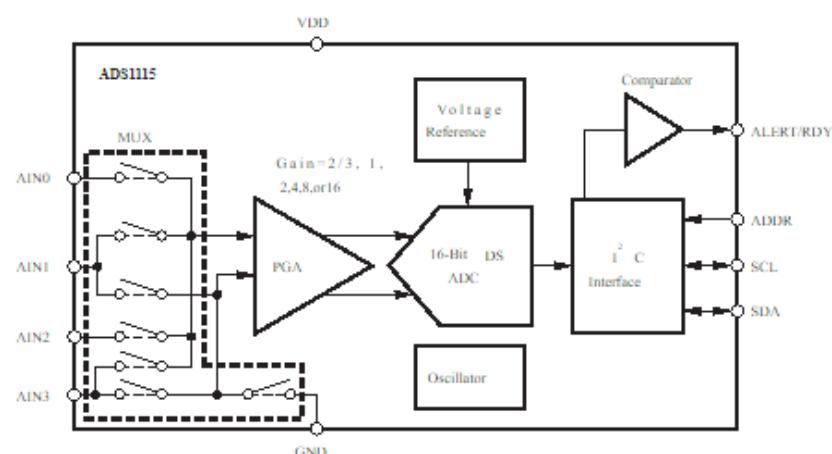


Ilustración 18: Diagrama de bloques ADS1115

**Pliego de Condiciones**

Pliego de condiciones

ARDUINO MEGA R3:**DESCRIPCIÓN GENERAL**

El Arduino Mega 2560 es una placa de microcontrolador basada en el ATmega2560 . Tiene 54 pines de entrada / salida digital (de los cuales 15 se pueden usar como salidas PWM), 16 entradas analógicas, 4 UART (puertos serie de hardware), un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, un encabezado ICSP, y un botón de reinicio. Contiene todo lo necesario para soportar el microcontrolador; simplemente conéctelo a una computadora con un cable USB o enciéndalo con un adaptador de CA a CC o una batería para comenzar. La placa Mega 2560 es compatible con la mayoría de los escudos diseñados para la Uno y las antiguas placas Duemilanove o Diecimila. El Mega 2560 es una actualización del Arduino Mega , al que reemplaza.

PARÁMETROS

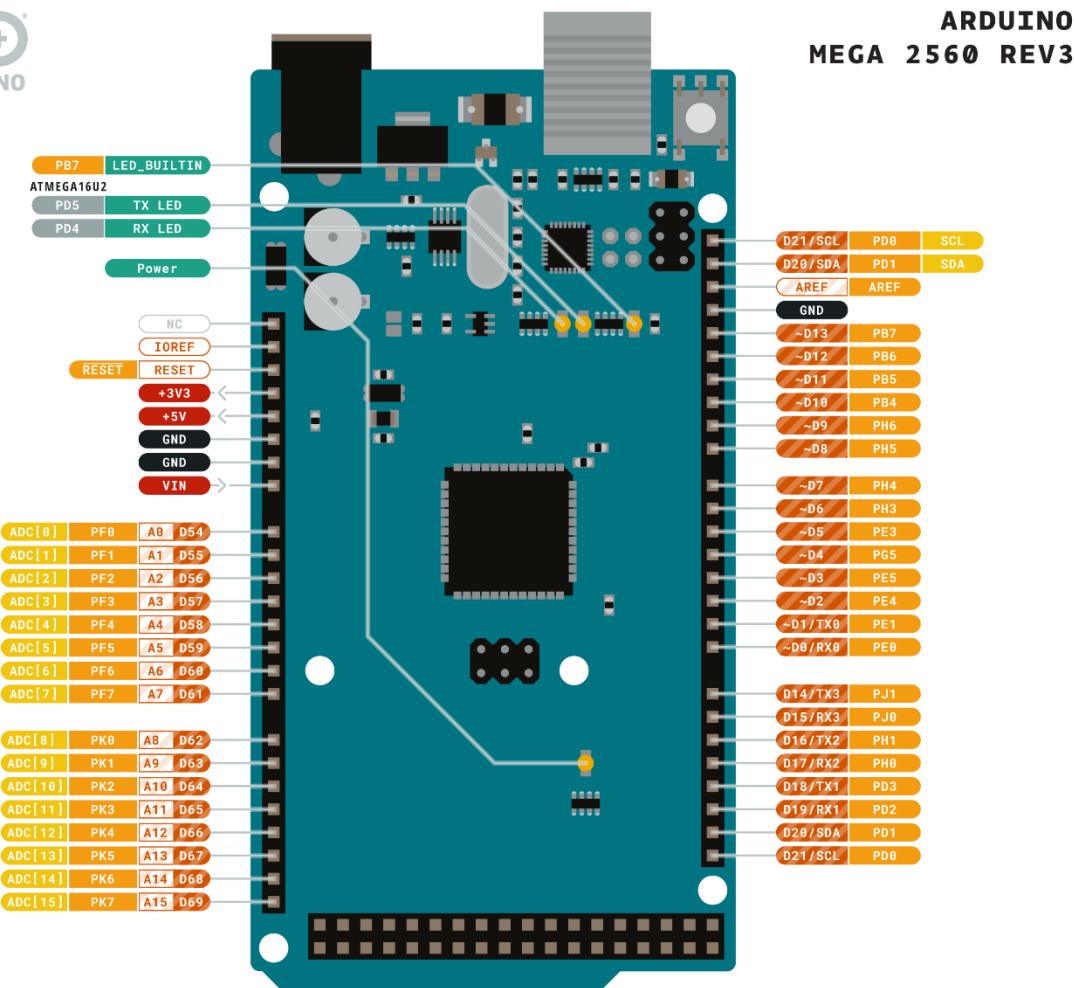
- Microcontrolador: ATmega2560
- Tensión de funcionamiento: 5V
- Voltaje de entrada (recomendado): 7-12 V
- Voltaje de entrada (límite): 6-20 V
- Pines de E / S digitales: 54 (de los cuales 15 proporcionan salida PWM)
- Pines de entrada analógica: dieciséis
- Corriente CC por pin de E / S: 20 mA
- Corriente CC para pin de 3.3V: 50 mA
- Memoria flash: 256 KB de los cuales 8 KB utiliza el gestor de arranque
- SRAM8 KB
- EEPROM: 4 KB
- Velocidad de reloj: 16 MHz
- LED_BUILTIN: 13
- Largo: 101,52 milímetros
- Ancho: 53,3 milímetros
- Peso: 37 g



Pliego de Condiciones

Pliego de condiciones

PINES



■ Ground ■ Internal Pin ■ Digital Pin ■ Microcontroller's Port
■ Power ■ SWD Pin ■ Analog Pin
■ LED ■ Other Pin ■ Default

ARDUINO .CC



This work is licensed under the Creative Commons
Attribution-ShareAlike 4.0 International License. To view
a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/> or send a letter to Creative
Commons, PO Box 1866, Mountain View, CA 94042, USA.

Ilustración 19: Pines y sus funciones ARDUINO MEGA



Pliego de Condiciones

Pliego de condiciones

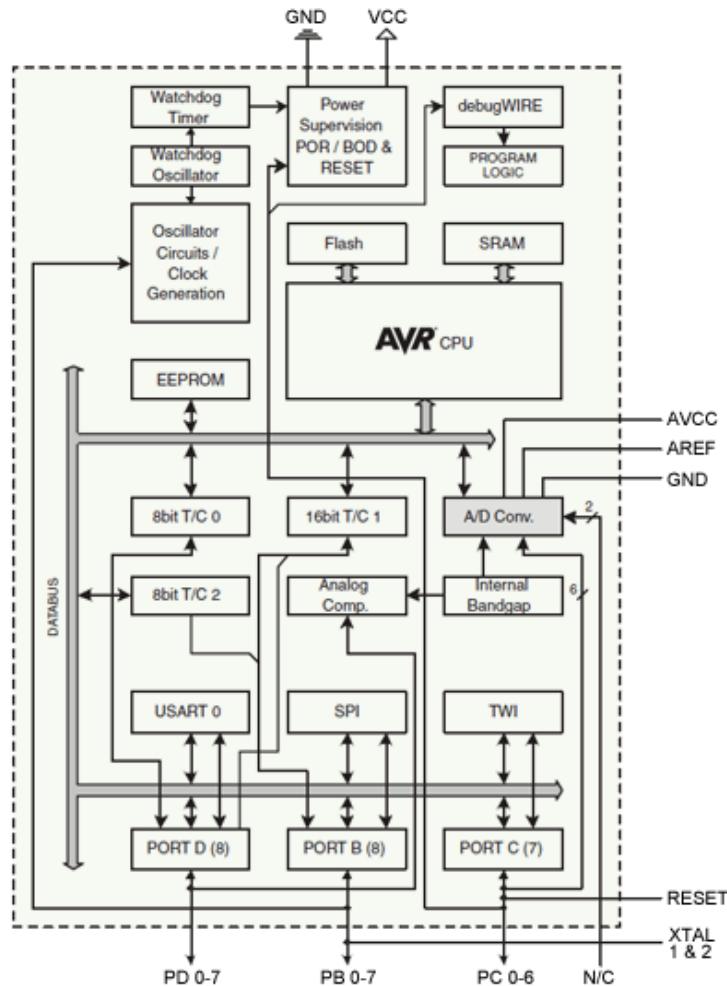


Ilustración 20: Diagrama de bloques Arduino MEGA

LCD 20X4 1602A:

DESCRIPCIÓN GENERAL

Los módulos LCD se utilizan con mucha frecuencia en la mayoría de los proyectos integrados, debido a su bajo precio, disponibilidad y compatibilidad con el programador. La mayoría de nosotros nos habríamos encontrado con estas pantallas en nuestro día a día, ya sea en PCO o en calculadoras. La apariencia y los pines ya se han visualizado arriba, ahora vamos a ponernos un poco técnicos.

La pantalla LCD de 16×2 se llama así porque; tiene 16 columnas y 2 filas. Hay muchas combinaciones disponibles como 8×1 , 8×2 , 10×2 , 16×1 , etc. pero la más utilizada es la LCD de 20×4 . Entonces, tendrá ($20 \times 4 = 80$) 80 caracteres en total y cada personaje estará compuesto por 5×8 Pixel Dots. En la imagen de abajo se muestra un solo carácter con todos sus píxeles.

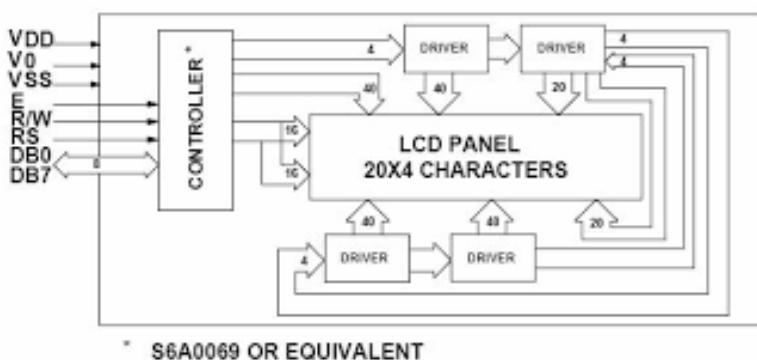
**Pliego de Condiciones**

Pliego de condiciones

Ahora, sabemos que cada “cuadro” tiene ($5 \times 8 = 40$) 40 píxeles y para 80 caracteres tendremos (80×40) 3200 píxeles. Además, la pantalla LCD también debería recibir instrucciones sobre la posición de los píxeles. Por lo tanto, será una tarea agitada manejar todo con la ayuda de MCU, por lo que se utiliza una interfaz IC como HD44780, que está montada en la parte posterior del módulo LCD. La función de este IC es obtener los comandos y los datos de la MCU y procesarlos para mostrar información significativa en nuestra pantalla LCD. Puede aprender a conectar una pantalla LCD utilizando los enlaces mencionados anteriormente. Si es un programador avanzado y le gustaría crear su propia biblioteca para conectar su microcontrolador con este módulo LCD, entonces debe comprender que el HD44780 IC está funcionando y los comandos que se pueden encontrar en su hoja de datos.

PARÁMETROS

- El voltaje de funcionamiento es de 4,7 V a 5,3 V
- El consumo de corriente es de 1 mA sin luz de fondo
- Módulo de pantalla LCD alfanumérico, lo que significa que puede mostrar alfabetos y números
- Consta de dos filas y cada fila puede imprimir 16 caracteres.
- Cada personaje está construido por una caja de 5×8 píxeles.
- Puede funcionar tanto en modo de 8 bits como de 4 bits
- También puede mostrar cualquier carácter generado personalizado
- Disponible en luz de fondo verde y azul

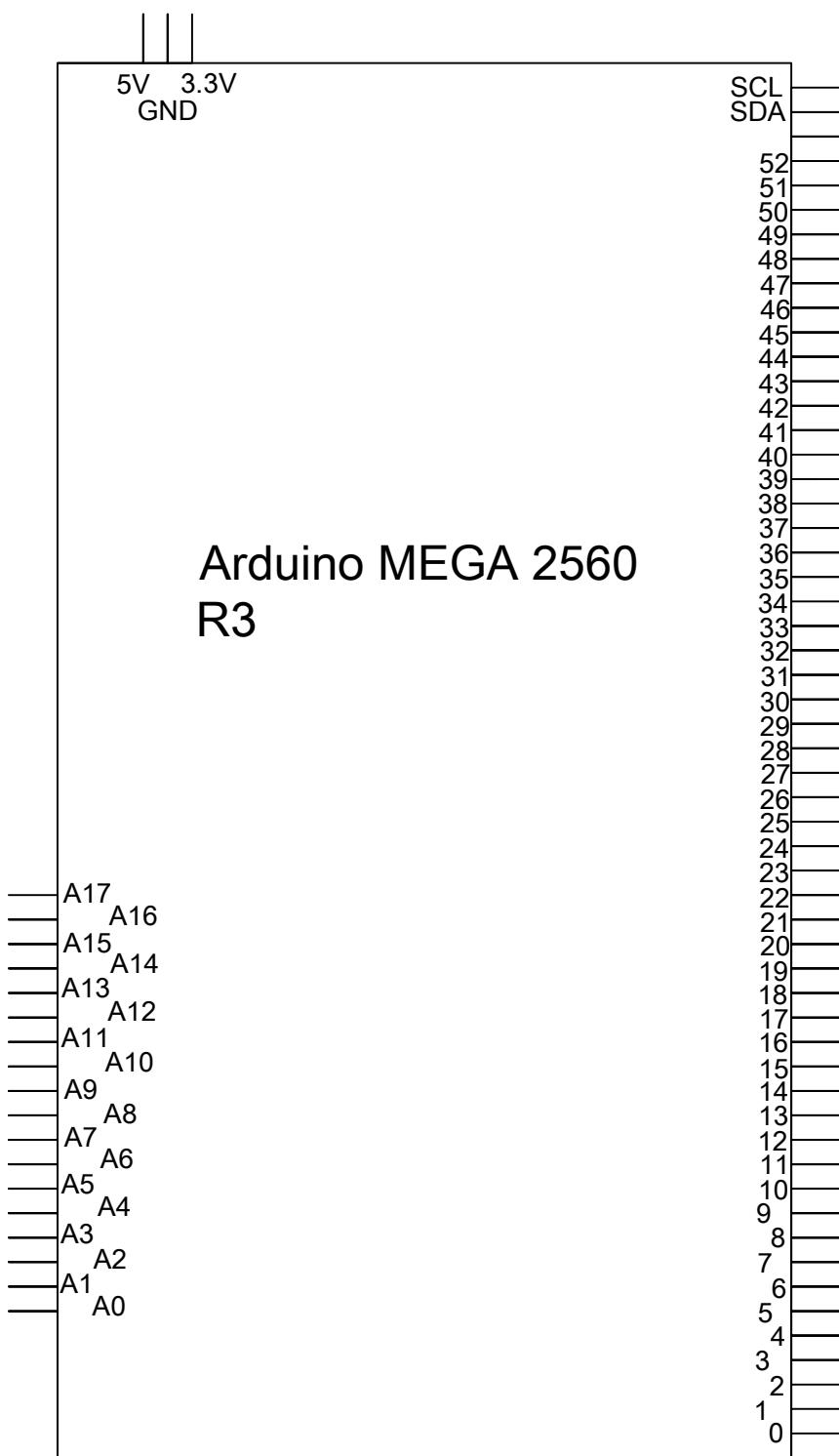
**Ilustración 21: Diagrama de bloques pantalla LCD**

**PLANOS**

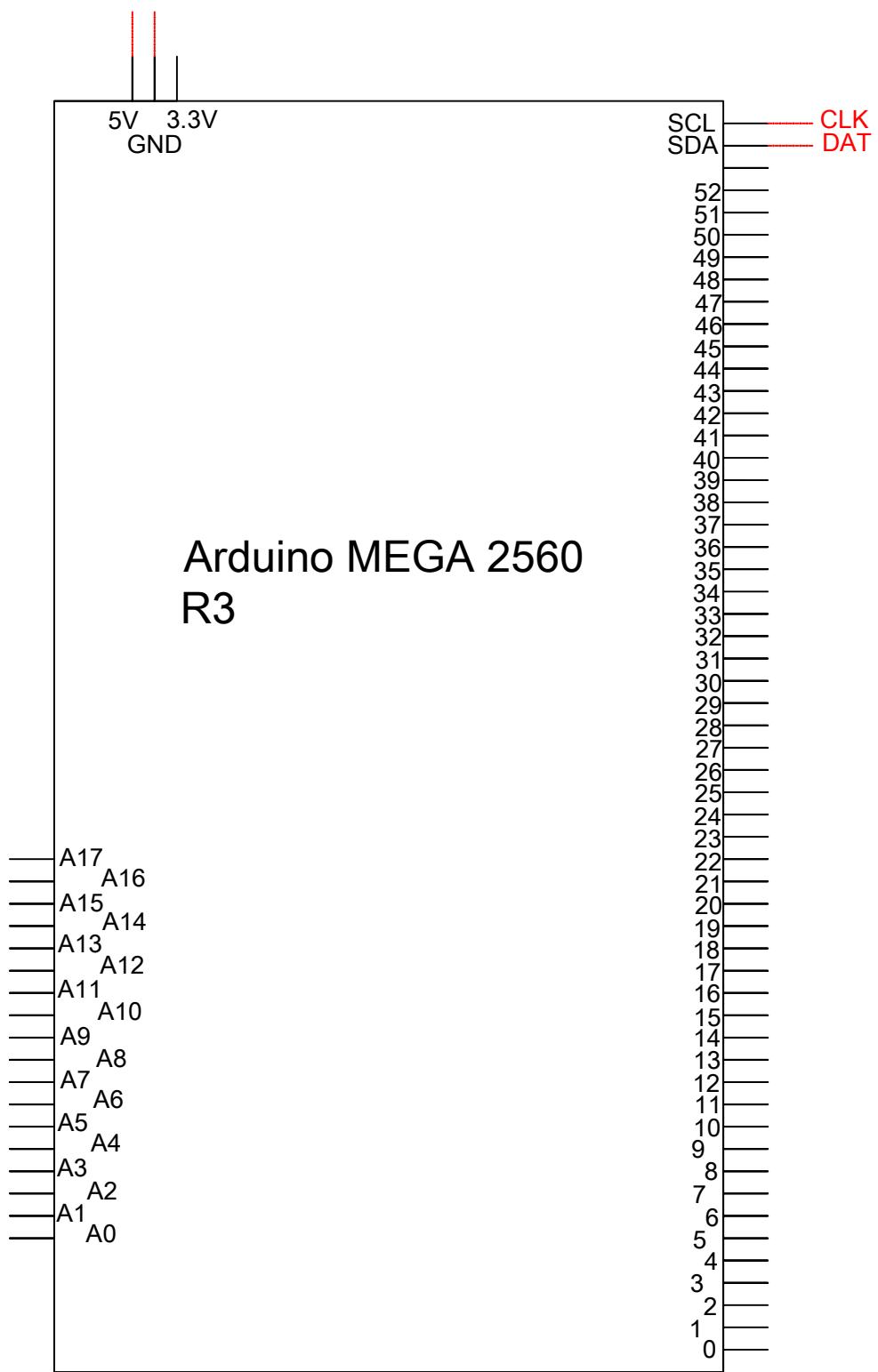
Planos

Planos

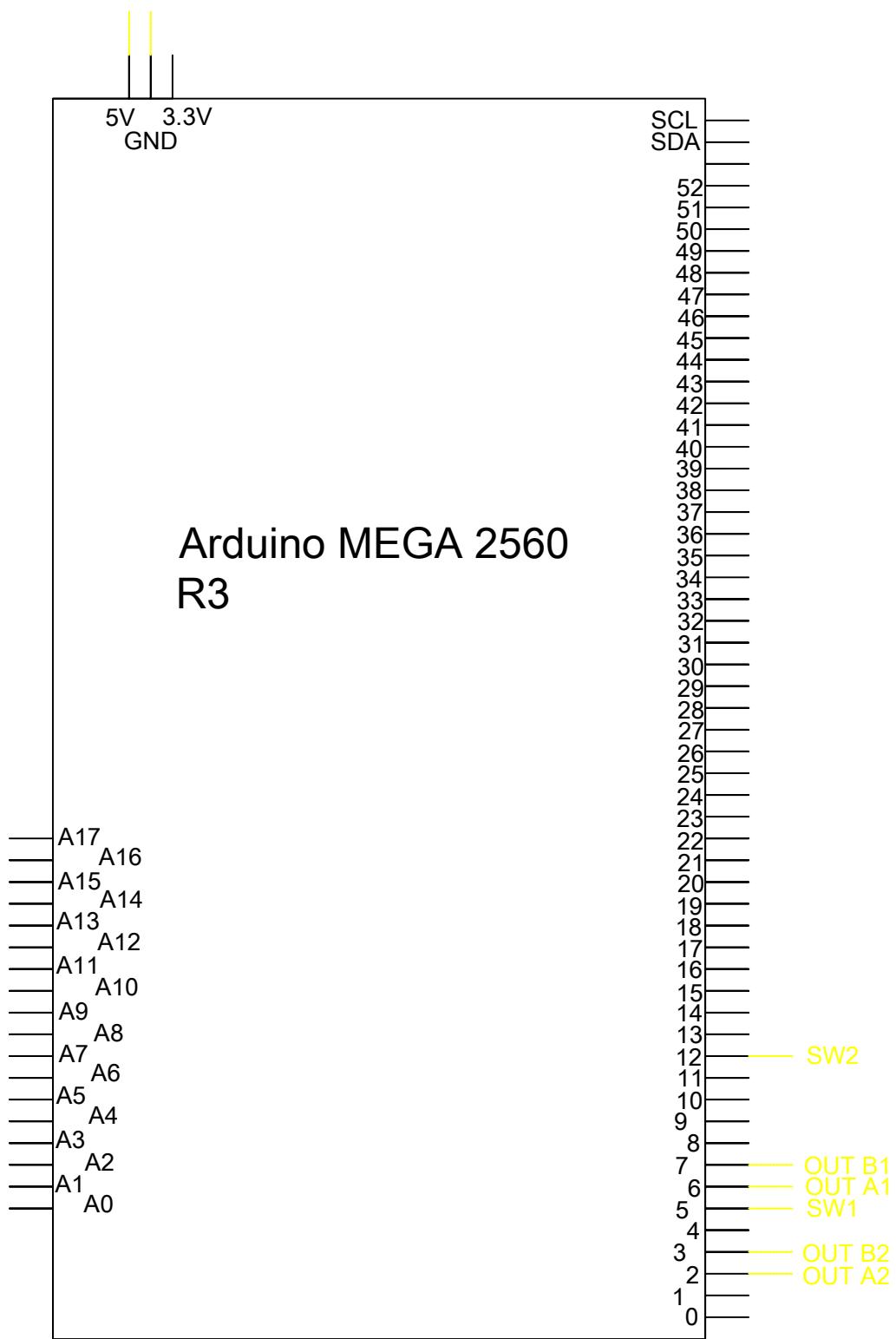
- Plano 1º: Esquema pines Arduino MEGA
- Plano 2º: Esquema pines LCD+I2C
- Plano 3º: Esquema pines Rotary Encoder
- Plano 4º: Esquema pines AD9833
- Plano 5º: Esquema pines botones
- Plano 6º: Esquema pines micrófono
- Plano 7º: Esquema pines LM386



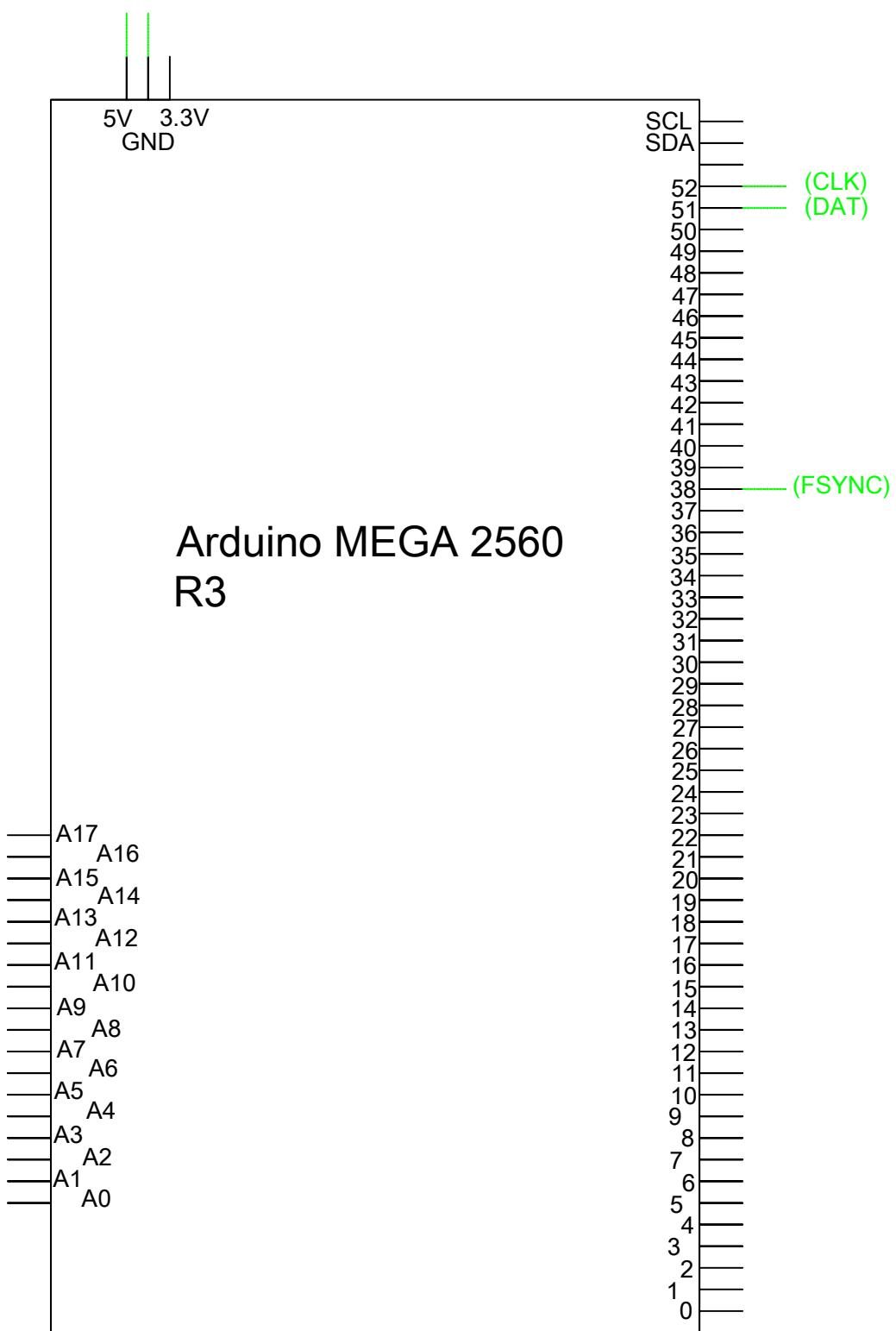
	Fecha	Nombre	Firma	IES LUIS BUENO CRESPO
Dibujado	14/06/21	J.Acosta		
Comprobado	21/06/21	J.L.Vázquez		
Escala	Esquema Arduino Mega R3			Número 1 Sustituye a Sustituido por



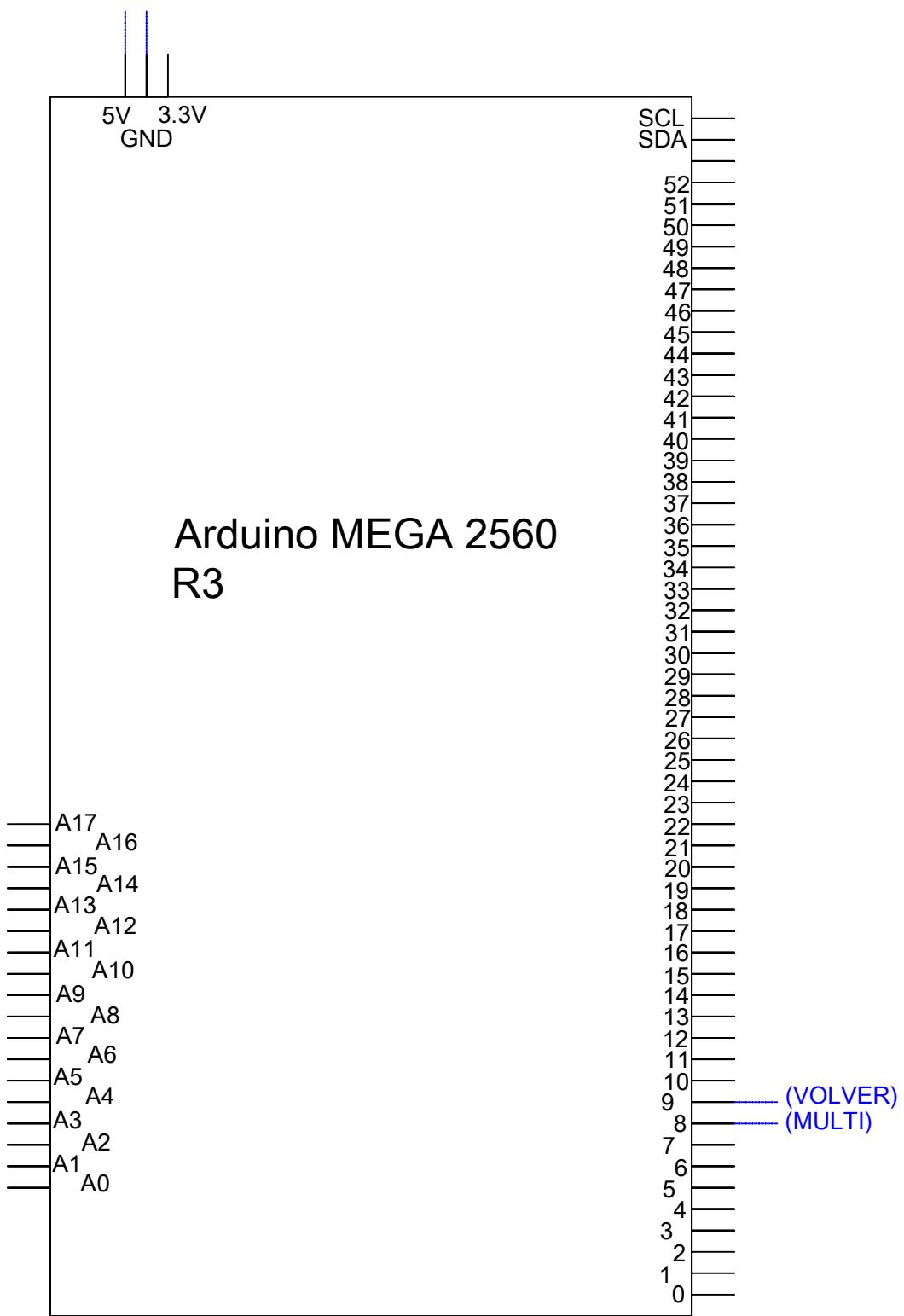
	Fecha	Nombre	Firma	IES LUIS BUENO CRESPO
Dibujado	14/06/21	J.Acosta		
Comprobado	21/06/21	J.L.Vázquez		
Escala	Esquema Pantalla LCD+I2C			Número 2
				Sustituye a
				Sustituido por



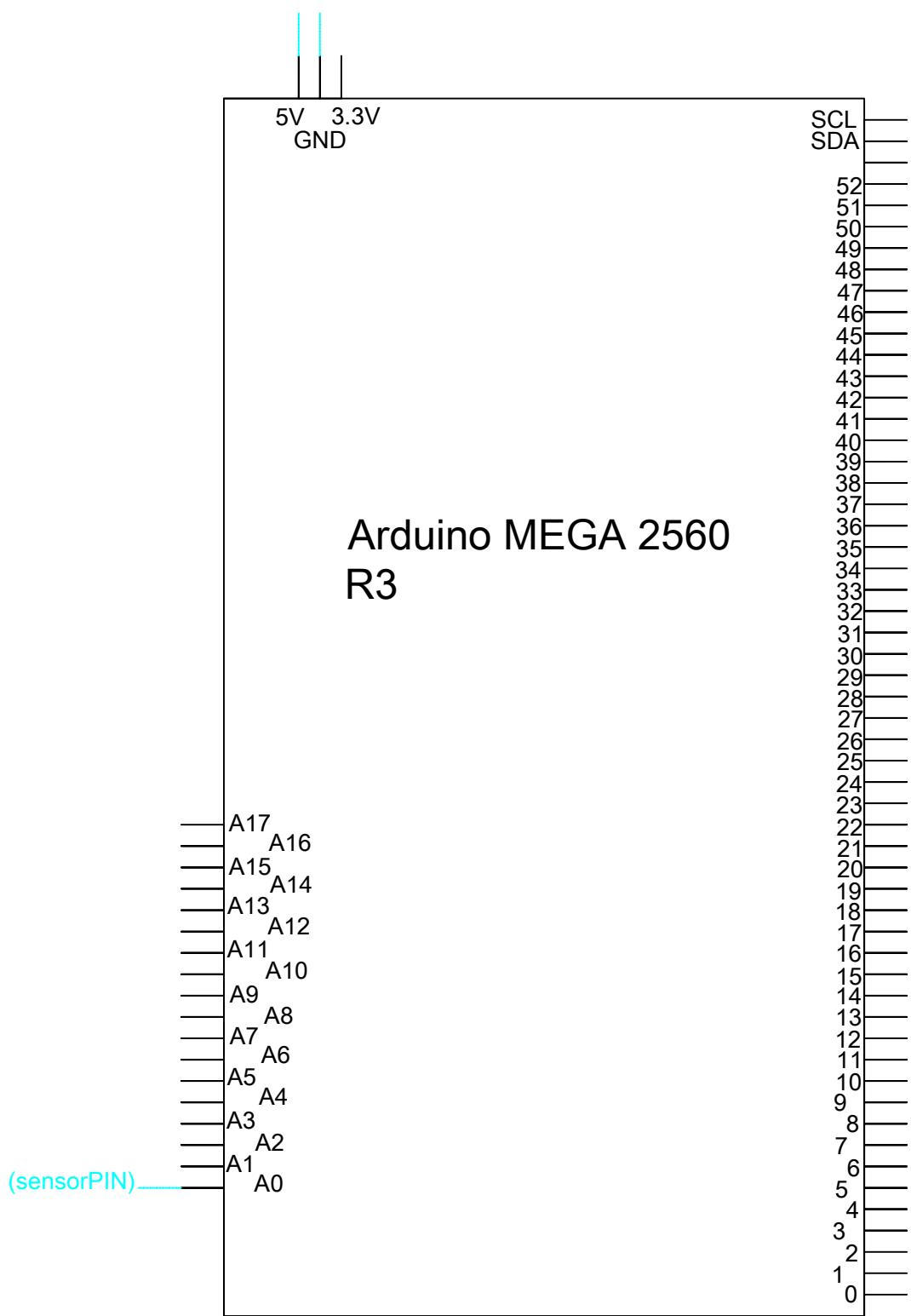
	Fecha	Nombre	Firma	IES LUIS BUENO CRESPO
Dibujado	14/06/21	J.Acosta		
Comprobado	21/06/21	J.L.Vázquez		
Escala	Esquema Rotary Encoder			Número 3
				Sustituye a
				Sustituido por



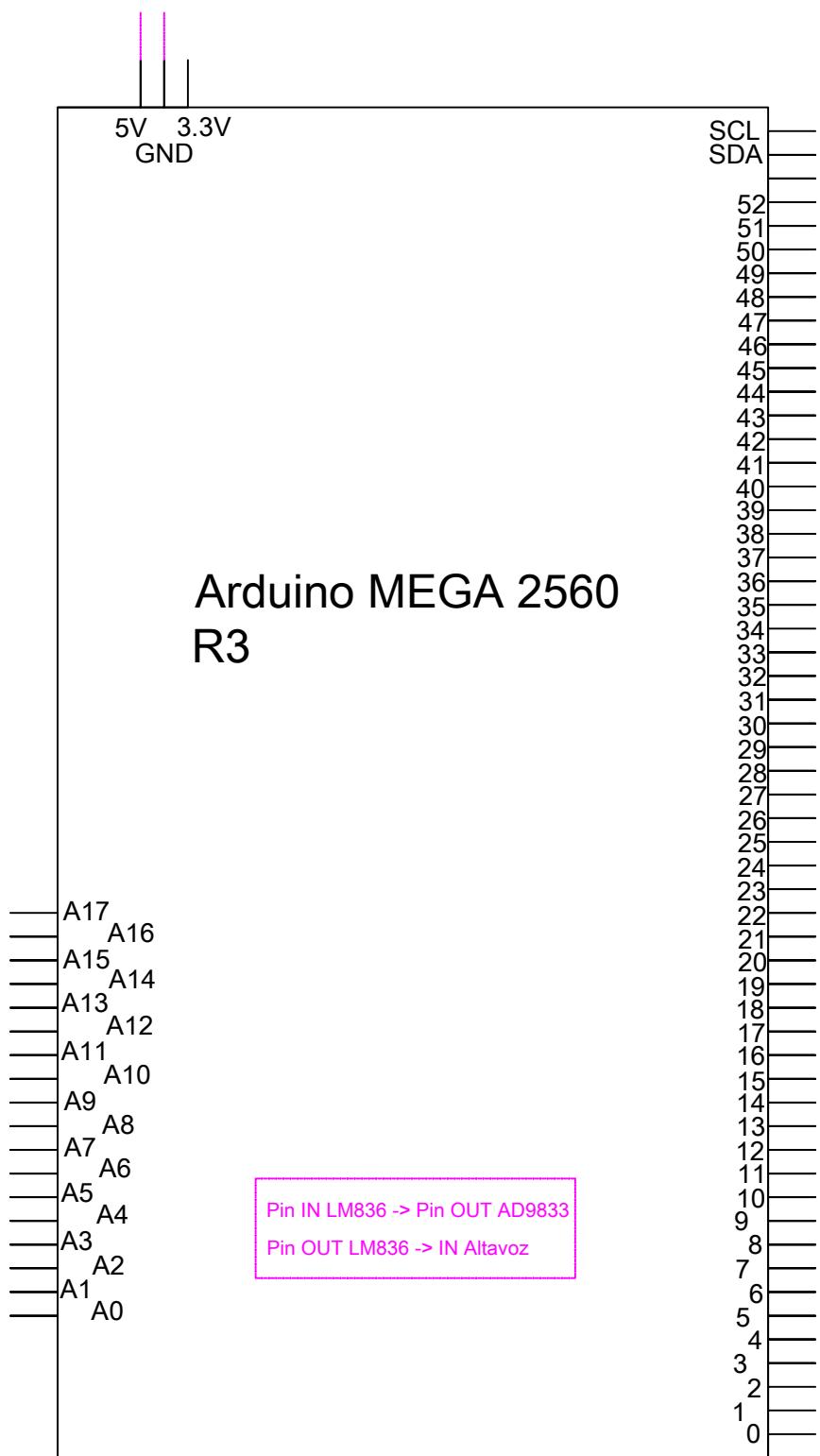
	Fecha	Nombre	Firma	IES LUIS BUENO CRESPO
Dibujado	14/06/21	J.Acosta		
Comprobado	21/06/21	J.L.Vázquez		
Escala	Esquema AD9833			Número 4
				Sustituye a
				Sustituido por



	Fecha	Nombre	Firma	IES LUIS BUENO CRESPO
Dibujado	14/06/21	J.Acosta		
Comprobado	21/06/21	J.L.Vázquez		
Escala	Esquema Pines Botones			Número 5 Sustituye a Sustituido por



	Fecha	Nombre	Firma	IES LUIS BUENO CRESPO
Dibujado	14/06/21	J.Acosta		
Comprobado	21/06/21	J.L.Vázquez		
Escala	Esquema Pines Micrófono			Número 6 Sustituye a Sustituido por



	Fecha	Nombre	Firma	IES LUIS BUENO CRESPO
Dibujado	14/06/21	J.Acosta		
Comprobado	21/06/21	J.L.Vázquez		
Escala	Esquema Pines LM836		Número 7	
			Sustituye a	
			Sustituido por	



Apéndice

Manual de instrucciones

Sorena es un analizador de acústica portátil, se le puede implementar una batería o pila, o conectarlo a la corriente con su debido transformador, pero también se puede alimentar desde cualquier ordenador o toma usb que le suministre 5V.

De esta manera el dispositivo se encenderá y de forma seguida mostrará la marca, la versión, y una pequeña transición de carga.

Una vez encendido mostrará el **menú principal** y sus posibilidades:

-**Med. Teórica** (Donde se podrá calcular el tiempo de reverberación teórico, el resultado anterior obtenido del mismo, y se podrá hacer uso de la función de sonómetro).

-**Med. Práctica** (Donde se podrá usar la función de generador de tonos puros, la medición del tiempo de reverberación práctico, y también se podrá ver el resultado anterior).

-**Soporte** (Aquí se dejan reflejadas las redes sociales oficiales de Sorena y del creador del analizador de acústica, Juan Miguel Acosta Ortega).

Por este menú, al igual que por sus submenús, el usuario se podrá desenvolver con el “Rotary encoder” llamado menú, es decir, si se gira en sentido horario la flecha (focus del menú) subirá de opción, y si gira en sentido antihorario bajará. Entrará en el submenú escogido cuando se pulse el botón del mismom Rotary.

Una vez en cualquiera de los submenús, el procedimiento para escoger cualquiera de las funciones deseadas será el mismo. Si se pulsa la función “Atrás”, el usuario volverá al menú anterior, en este caso el principal.

El **primer submenú** presenta las funciones comentadas anteriormente:

-**Inserción de datos:** En esta función se insertarán datos con el Rotary encoder llamado “frecuencia”, de modo que cuando gire en sentido horario el valor aumentará de 0.5 en 0.5 en caso de ser una longitud y de 0.01 en 0.01 en caso de ser coeficiente de absorción, y cuando gire en sentido antihorario decrementará, .

Al acabar de insertar todos los datos necesarios, el valor del tiempo de reverberación teórico se mostrará por pantalla durante 4 segundos.

Los valores a insertar son el ancho alto y largo de la sala, y el coeficiente de absorción de las paredes, el techo y el suelo.

**Proyecto Fin de Ciclo****Apéndice**

Ejemplo → Ancho*Largo*Alto=Volumen;

Se puede escoger el coeficiente de absorción para cualquier frecuencia deseada.

Para salir en cualquier momento de esta función se puede pulsar el botón “volver”.

-Resultado ant. : Esta función mostrará el resultado calculado anteriormente durante 4 segundos por pantalla.

-Sonómetro: El sonómetro sin embargo mostrará 1 valor cada segundo durante 10 segundos acerca de la sensación sonora del ambiente.

El **segundo submenú** por otro lado presenta las siguientes funciones:

-Gen. Tonos: Esta función tiene diversos controles. Con el encoder rotativo llamado “frecuencia” se cambia efectivamente la frecuencia, con el botón “multi” varía el multiplicador, es decir, si la frecuencia suma de uno en uno, de 10 en 10, de 100 en 100... y con el botón del rotary varía la forma de onda.

Para salir en cualquier momento de esta función se puede pulsar el botón “volver”.

-T. Reverberación: INCOMPLETO Al entrar en esta función deberemos pulsar el botón MULTI, esto es debido a que a la hora de realizar el tiempo de reverberación se ha de esperar a un momento idóneo en el que no haya ruido.

Después de esto al pulsar el borón del encoder secundario iremos pasando de octava en octava para medir su tiempo de reverberación, y se podrá volver al menú principal pulsando el botón volver.

-Resultados anteriores: En esta función simplemente se muestran por pantalla los resultados anteriores del tiempo de reverberación práctico.

El **tercer y último submenú** deja varias formas de contactar con el soporte técnico en caso de avería.



Programación Fuente Sorena dividida en archivos:

Sorena.ino

```
#include <LiquidCrystal_I2C.h>
#include <LiquidMenu.h>
#include <SPI.h>
#include <MD_cmdProcessor.h>
#include <MD_AD9833.h>

#include "clmenu.h"
#include "Gensen.h"

//Bandera del rotary encoder secundario
int banderarotary2=0;

//valores usados en el tiempo de reverberación práctico
long int valor[20],mayor,menor; //tiempo[500];

int dat; //Variable necesaria para el generador de ondas

bool pulsant=LOW, puls nueva=LOW, pulsacion=0,seleccion=0; //Para controlar el flanco de bajada, flanco de subida, y opciones del t60 teorico

unsigned long inicio=0, fin=0, transcurrido=0;//variables para obtener el paso del tiempo en la funcion de t.rev. practico

long int paso=9; //variable bandera para el tiempo de reverberación teórico

float contador=0; //para la inserción de dígitos

float
volumen=0,areasuelo=0,areapared13=0,areapared24=0,areatecho=0,ancho=0,largo=0,alto=0,areapared=0 ,coefabsuelo=0;
float coefabpared=0,coefabtecho=0,area=0,areaabsuelo=0,areaabparedes=0,areaabtecho=0,contadorfloat=0,
areaventana=0,areaabventana=0,coefabventana=0;
//variables usadas en el cálculo teórico del T60

void selectOption(); //llamamos a todas las funciones en el archivo principal
void fn_atras();
void fn_t60teo();
void fn_t60pra();
void fn_sop();
void fn_indat();
void fn_resant();
void fn_sonom();
void fn_genton();
void fn_t60();
void fn_reant();
```

**Proyecto Fin de Ciclo**

Apéndice

```
void fn_gmail();
void fn_insta();
void fn_linkedin();
void inicializamenu();

void AD9833setup();
void AD9833reset();
void AD9833setFrequency(long frequency, int Waveform);
void WriteRegister(int dat);
void actualiza_func();
void limpiaDigitos();

void setup()
{
    Serial.begin(9600);
    pinMode(FSYNC, OUTPUT); //escrituras de registro e inserción de valores necesarias para el generador de
    tonos AD9833 encontradas en su datasheet
    digitalWrite(FSYNC, HIGH);
    SPI.begin();
    delay(50);
    WriteRegister(0x100);
    delay(10);
    SPI.setDataMode(SPI_MODE2);
    digitalWrite(FSYNC, LOW);
    delayMicroseconds(10);
    SPI.transfer(dat>>8);
    SPI.transfer(dat&0xFF);
    digitalWrite(FSYNC, HIGH);
    SPI.setDataMode(SPI_MODE0);

    pinMode(sw,INPUT_PULLUP);
    pinMode(back_func,INPUT_PULLUP);
    pinMode(MULTI,INPUT_PULLUP);
    pinMode(sw2,INPUT_PULLUP);

    lcd.begin();
    lcd.backlight();

    lcd.clear();

    lcd.setCursor(7,0); lcd.print("Sorena");
    lcd.setCursor(0,2); lcd.print("Ver 0.1  ");

    lcd.setCursor(0,3);
    for( int i=0 ; i<20 ; i++ )
    {
        lcd.print(".");
        delay(50);
    }
    delay(50);
```

**Proyecto Fin de Ciclo**

Apéndice

```
menu.init(); //Esta inicialización solo es necesaria si se usa el módulo I2C
inicializamenu();
menu.setFocusedLine(0);
menu.update();
AD9833setup();
}

void loop()
{
    selectOption();
    //rotary del menu
    aState = digitalRead(outputA);
    if (aState != aLastState)
    {
        if (digitalRead(outputB) != aState)
        {

            menu.switch_focus(true);
        } else {

            menu.switch_focus(false);
        }
        menu.update();
        aLastState = aState;
    }

    //rotary para cambiar datos
    aState2 = digitalRead(outputA2);
    if (aState2 != aLastState2)
    {
        if (digitalRead(outputB2) != aState2)
        {
            banderarotary2=-1;
            Serial.println("sentido antihorario");
            Serial.println(banderarotary2);

        } else
        {
            banderarotary2=1;
            Serial.println("sentido horario");
            Serial.println(banderarotary2);

        }
        aLastState2 = aState2;
    }

    puls nueva=digitalRead(sw2);
    if (pulsant==HIGH && puls nueva==LOW && (bandindat!=0 || dentro==1 || bandt60!=0))//flanco de bajada
{
```

**Proyecto Fin de Ciclo**

Apéndice

```
pulsant=LOW;
delay(50);
pulsacion=0;
banderarotary2=0;
}

if (pulsant==LOW && puls nueva==HIGH && (bandindat!=0 || dentro==1))//flanco de subida, que solo
funcione en funciones concretas.
{
    pulsant=HIGH;
    delay(50);
    pulsacion=1;
    banderarotary2=0;
}

//variable bandera para entrar a la función de generador de tonos
if(bandgenton!=0)
{
    //Para que sólo se ejecute esta fracción de código una sola vez por entrada a la función
    if(primerag==1)
    {
        counter=500;
        func=0;
        digitalWrite(FSYNC, HIGH);
        lcd.clear();
        lcd.setCursor(0,0);
        primerag=0;
    }
    //Leo el multiplicador de frecuencia (botón MULTI)
    if(!digitalRead(MULTI))
    {
        delay(50);
        if(!digitalRead(MULTI))
        {
            multi=multi+1;

            Serial.println("Multi -> ");Serial.println(multi);
            if(multi>=7) multi=0;
        }
    }
}

//leo el rotary para modificar al frecuencia
if (banderarotary2!=0)
{
    if(banderarotary2>0)
    {
        counter=counter+pow(10,multi);
        if(counter>100000) counter=1000000;
        banderarotary2=0;
    }
}
```

**Proyecto Fin de Ciclo**

Apéndice

```
Serial.println("frecuencia= ");
Serial.println(counter);
}
if(banderarotary2<0)
{
    counter=counter-pow(10,multi);
    if(counter<0) counter=0;
    banderarotary2=0;
    Serial.println("frecuencia= ");
    Serial.println(counter);
}
//Leo el pulsador para cambiar el tipo de onda
if(!digitalRead(sw2))
{
    delay(100);
    if(!digitalRead(sw2))
    {

        if(func_ant == 0){
            func = 1;
            Serial.println("sw1º");
        }
        if(func_ant == 1){
            func = 2;
            Serial.println("sw2º");
        }
        if(func_ant == 2){
            func = 0;
            Serial.println("sw3º");
        }
    }
}

//si hay alguna modificación en cualquiera de estos valores se actualiza la onda y la pantalla, antes no.
if((counter_ant != counter) || (function_ant != function) || (multi!=multi_ant) || (func_ant != func))
{
    lcd.setCursor(0, 3);
    lcd.print("10^");
    lcd.print(multi);
    lcd.setCursor(17, 0);
    lcd.print("Hz");
    lcd.setCursor(0, 0);
    lcd.print(counter);

    Serial.println("Escribiendo...");

    if(func == 0){
        lcd.setCursor(10, 3);
        lcd.print(" SENOIDAL");
        function = WAVE_SINE;
    }
}
```

**Proyecto Fin de Ciclo**

Apéndice

```
if(func == 1){  
lcd.setCursor(10, 3);  
lcd.print("TRIANGULAR");  
function = WAVE_TRIANGLE;  
}  
if(func == 2){  
lcd.setCursor(10, 3);  
lcd.print(" CUADRADA");  
function = WAVE_SQUARE;  
}  
Serial.println("Funcion");  
Serial.println(function,HEX);  
limpiaDigitos();  
AD9833setFrequency(counter, function);  
Serial.println("Actualizo");  
  
counter_ant = counter;  
function_ant = function;  
multi_ant=multi;  
func_ant = func;  
}  
  
//Leo el pulsador para salir de este menu  
if(digitalRead(back_func)==LOW)  
{  
  digitalWrite(FSYNC, LOW);  
  counter=0;  
  menu.change_screen(3);  
  menu.set_focusedLine(0);  
  Serial.println("Salir genton");  
  bandgenton=0;  
  AD9833setFrequency(counter, function);  
}  
}  
  
//en este momento se entra a la funcion de tiempo de reverberación práctico  
if(bandt60!=0)  
{  
  
  if(primerat==1) //hacemos esto cada vez que se entra a la función  
  {  
    lcd.clear();  
    primerat=0;  
  }  
  
  if(dentro==1)  
  {  
    //Esta función no esta terminada, pues después de hacer pruebas con el ADC nativo y el ADS1115 de  
    Adafruit se determinó que hacía falta uno de mucha más calidad, y no había tiempo para realizar más  
    pruebas.  
  }  
}
```

**Proyecto Fin de Ciclo**

Apéndice

```
lcd.setCursor(1,0);
lcd.print(" Manten VOLVER  ");
Serial.println("1000Hz");

counter=1000;
function=WAVE_SINE;
AD9833setFrequency(counter, function);
delay(1000);

counter=0;
AD9833setFrequency(counter, function);

/*
for(int i=0;i<200;i++)
{
valor[i] = analogRead(sensorPIN);
tiempo[i] = millis();

if (valor < menor) menor=valor;
if (valor > mayor) mayor=valor;
}
*/

if(digitalRead(back_func)==LOW) //para poder salir y reestablecer ciertos valores a la normalidad
{

menu.change_screen(3);
menu.set_focusedLine(0);
Serial.println("Salir t60");
bandt60=0;
dentro=0;
primerat=1;
counter=0;
function=WAVE_SINE;
AD9833setFrequency(counter, function);

}

}

}

//entramos a la inserción de datos; casi todas las funciones tienen su variable bandera
if (bandindat!=0)
{

if(primeraindat==1) //paso a paso se introducen los valores
{
lcd.clear();
paso=9;
pulsacion=0;
lcd.setCursor(0,0);
```

**Proyecto Fin de Ciclo**

Apéndice

```
lcd.print("Insercion datos:");
lcd.setCursor(0,1);
lcd.print("Ancho= ");
primeraindat=0;
}

if (pulsacion==1 && paso==9)
{
paso=8;
ancho=contador-0.5;
Serial.println("ancho=");
Serial.println(ancho);
contador=0;
lcd.setCursor(0,2);
lcd.print("Largo= ");
Serial.println("ancho=contador");

pulsacion=0;
}

if ((banderarotary2!=0) && (paso==9) && (pulsacion==0))
{
lcd.setCursor(0,1);
lcd.print("Ancho= ");
lcd.setCursor(9,1);
lcd.print(contador);
lcd.setCursor(14,1);
lcd.print("m");
Serial.println("Ancho");
Serial.println(contador);

if(banderarotary2>0)
{
contador=contador+0.5;
banderarotary2=0;
}
if(banderarotary2<0)
{
contador=contador-0.5;
banderarotary2=0;
if(contador<=0) contador=0;
}
banderarotary2=0;
}

if (pulsacion==1 && paso ==8)
{
paso=7;
```

**Proyecto Fin de Ciclo**

Apéndice

```
largo=contador-0.5;
Serial.println("largo=");
Serial.println(largo);
contador=0;
pulsacion=0;
lcd.setCursor(0,3);
lcd.print("Alto= ");
}

if ((banderarotary2!=0) && (paso==8))
{
    Serial.println("Paso");
    Serial.println(paso);
    lcd.setCursor(0,2);
    lcd.print("Largo= ");
    lcd.setCursor(9,2);
    lcd.print(contador);
    lcd.setCursor(14,2);
    lcd.print("m");
    Serial.println("largo");
    Serial.println(contador);

    if(banderarotary2>0)
    {
        contador=contador+0.5;
        banderarotary2=0;
    }
    if(banderarotary2<0)
    {
        contador=contador-0.5;
        banderarotary2=0;
        if(contador<=0) contador=0;
    }
}

if (pulsacion==1 && paso ==7)
{
    paso=6;
    alto=contador-0.5;
    Serial.println("ancho");Serial.println(ancho);
    Serial.println("largo");Serial.println(largo);
    Serial.println("alto");Serial.println(alto);
    volumen=alto*ancho*largo;
    areapared13=2*(largo*alto);
    areapared24=2*(ancho*alto);
    areapared=areapared13+areapared24;
    areasuelo=ancho*largo;
    areatecho=ancho*largo;
    contador=0;
    delay(20);
}
```

**Proyecto Fin de Ciclo****Apéndice**

```
lcd.clear();
lcd.setCursor(0,1);
lcd.print("Coef. ab suelo= ");
Serial.println("volumen");Serial.println(volumen);
Serial.println("pared");Serial.println(areapared);
Serial.println("suelo");Serial.println(areasuelo);
Serial.println("techo");Serial.println(areatecho);
pulsacion=0;
}

if ((banderarotary2!=0) && (paso==7))
{
    lcd.setCursor(0,3);
    lcd.print("Alto= ");
    lcd.setCursor(9,3);
    lcd.print(contador);
    lcd.setCursor(14,3);
    lcd.print("m");

    Serial.println(contador);

    if(banderarotary2>0)
    {
        contador=contador+0.5;
        banderarotary2=0;
    }
    if(banderarotary2<0)
    {
        contador=contador-0.5;
        banderarotary2=0;
        if(contador<=0) contador=0;
    }
}

if (pulsacion==1 && paso ==6)
{
    paso=5;
    coefabsuelo=contadorfloat-0.01;
    areaabsuelo=coefabsuelo*areasuelo;
    contadorfloat=0;
    pulsacion=0;
    lcd.clear();
    lcd.setCursor(0,1);
    lcd.print("Coef. ab. pared= ");
}

if ((banderarotary2!=0) && (paso==6))
{
    Serial.println("Paso");
    Serial.println(paso);
```

**Proyecto Fin de Ciclo**

Apéndice

```
lcd.setCursor(0,2);
lcd.print(contadorfloat);

Serial.println("c. a. suelo");
Serial.println(contadorfloat);

if(banderarotary2>0)
{
    contadorfloat=contadorfloat+0.01;
    banderarotary2=0;

}
if(banderarotary2<0)
{
    contadorfloat=contadorfloat-0.01;
    banderarotary2=0;
    if(contadorfloat<=0) contadorfloat=0;
}
}

if (pulsacion==1 && paso ==5)
{
    paso=4;
    coefabpared=contadorfloat-0.01;
    contadorfloat=0;
    delay(20);
    lcd.clear();
    lcd.setCursor(0,1);
    lcd.print("Coef. ab. techo= ");
    pulsacion=0;
}

if ((banderarotary2!=0) && (paso==5))
{

lcd.setCursor(0,2);
lcd.print(contadorfloat);
Serial.println(contadorfloat);

if(banderarotary2>0)
{
    contadorfloat=contadorfloat+0.01;
    banderarotary2=0;

}
if(banderarotary2<0)
{
    contadorfloat=contadorfloat-0.01;
    banderarotary2=0;
    if(contadorfloat<=0) contadorfloat=0;
```

**Proyecto Fin de Ciclo**

Apéndice

```
}

if (pulsacion==1 && paso ==4)
{
    paso=3;
    coefabtecho=contadorfloat-0.01;
    areaaabtecho=coefabtecho*areatecho;
    contadorfloat=0;
    pulsacion=0;
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Alguna pared dispar:");
    lcd.setCursor(1,1);
    lcd.print("SI");
    lcd.setCursor(6,1);
    lcd.print("NO");
    Serial.println("c.a suelo");Serial.println(coefabsuelo);
    Serial.println("c.a techo");Serial.println(coefabtecho);
    Serial.println("c.a pared");Serial.println(coefabpared);
}

if ((banderarotary2!=0) && (paso==4))
{
    Serial.println("Paso");
    Serial.println(paso);

    lcd.setCursor(0,2);
    lcd.print(contadorfloat);

    Serial.println("a");
    Serial.println(contadorfloat);

    if(banderarotary2>0)
    {
        contadorfloat=contadorfloat+0.01;
        banderarotary2=0;

    }
    if(banderarotary2<0)
    {
        contadorfloat=contadorfloat-0.01;
        banderarotary2=0;
        if(contadorfloat<=0) contadorfloat=0;
    }
}

if (pulsacion==1 && paso ==3 && seleccion==1)
{
    paso=2;
```

**Proyecto Fin de Ciclo**

Apéndice

```
banderarotary2=0;
pulsacion=0;
contador=0;
contadorfloat=0;
Serial.println("si");
lcd.setCursor(0,2);
lcd.print("Superficie= ");
}

if (pulsacion==1 && paso==3 && seleccion==0)
{
    paso=0;
    banderarotary2=0;
    pulsacion=0;
    contadorfloat=0;
    contador=0;
    Serial.println("no");
}

if ((banderarotary2!=0) && (paso==3))
{
    if(banderarotary2>0)
    {
        lcd.setCursor(0,1);
        lcd.print(">SI NO ");
        seleccion=1;
        banderarotary2=0;
    }
    if(banderarotary2<0)
    {
        lcd.setCursor(0,1);
        lcd.print(" SI >NO ");
        areaabparedes=coefabpared*areapared;
        seleccion=0;
        banderarotary2=0;
    }
}

if (pulsacion==1 && paso==2)
{
    paso=1;
    areaventana=contador-0.5;
    areapared=areapared-areaventana;
    areaabparedes=coefabpared*areapared;
    pulsacion=0;
    contador=0;
    contadorfloat=0;
    lcd.setCursor(0,3);
    lcd.print("co.ab.dispar= ");
}
```

**Proyecto Fin de Ciclo**

Apéndice

```
if ((banderarotary2!=0) && (paso==2))
{
    lcd.setCursor(12,2);
    lcd.print(contador);
    lcd.setCursor(17,2);
    lcd.print("m^2");

    if(banderarotary2>0)
    {
        contador=contador+0.5;
        banderarotary2=0;
    }
    if(banderarotary2<0)
    {
        contador=contador-0.5;
        banderarotary2=0;
        if(contador<=0) contador=0;
    }
}

if (pulsacion==1 && paso==1)
{
    paso=0;
    coefabventana=contador-0.01;
    areaabventana=coefabventana*areaventana;
    areaabparedes=areaabparedes+areaabventana;
    pulsacion=0;
    contador=0;
    contadorfloat=0;
}

if ((banderarotary2!=0) && (paso==1))
{
    lcd.setCursor(13,3);
    lcd.print(contador);

    if(banderarotary2>0)
    {
        contador=contador+0.01;
        banderarotary2=0;
    }
    if(banderarotary2<0)
    {
        contador=contador-0.01;
        banderarotary2=0;
        if(contador<=0) contador=0;
    }
}
```

**Proyecto Fin de Ciclo****Apéndice**

```
}

if (paso==0)
{
    Serial.println("areaabtecho");Serial.println(areaabtecho);
    Serial.println("areaabsuelo");Serial.println(areaabsuelo);
    Serial.println("areaabparedes");Serial.println(areaabparedes);

    area=areaabtecho+areaabparedes+areaabsuelo;
    trev60teo=(0.161*volumen)/area;
    Serial.println("volumen");Serial.println(volumen);
    Serial.println("area");Serial.println(area);
    Serial.println("trevt60teo");Serial.println(trev60teo);
    lcd.clear();
    lcd.print(" T. Reverberacion ");
    lcd.setCursor(0,1);
    lcd.print(" teorico ");
    lcd.setCursor(7,3);
    lcd.print(trev60teo); lcd.print(" s");
    delay(4000);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Insercion datos:");
    lcd.setCursor(0,1);
    lcd.print("Ancho= ");
    paso=9;
}

if(digitalRead(back_func)==LOW)
{
    menu.change_screen(2);
    menu.set_focusedLine(0);
    Serial.println("salir indat");
    bandindat=0;
    paso=9;
}
}
```

void AD9833setFrequency(long frequency, int Waveform) //funcion que establece la frecuencia y forma de onda necesaria en el archivo principal.

```
{
    long FreqWord = (frequency * pow(2, 28)) / 25.0E6;
    int MSB = (int)((FreqWord & 0xFFFFC000) >> 14);
    int LSB = (int)(FreqWord & 0x3FFF);
    LSB |= 0x4000;
    MSB |= 0x4000;
    WriteRegister(0x2100);
    WriteRegister(LSB);
```

**Proyecto Fin de Ciclo****Apéndice**

```
    WriteRegister(MSB);
    WriteRegister(0xC000);
    WriteRegister(Waveform);
}
```

Gensen.h

```
#define FSYNC 38

#define WAVE_SINE 0x2000
#define WAVE_SQUARE 0x2028
#define WAVE_TRIANGLE 0x2002

#define outputA 6 //Data (Sentido antihorario)
#define outputB 7 //Clock (Sentido horario)
#define sw 5
#define MULTI 8

#define sw2 12
#define outputA2 2
#define outputB2 3
```

```
void AD9833setup();
void AD9833reset();
void AD9833setFrequency(long frequency, int Waveform);
void WriteRegister(int dat);

void limpiaDigitos();
```

```
void AD9833setup() //de nuevo escrituras de registro y setup del AD9833
{
    pinMode(FSYNC, OUTPUT);
    digitalWrite(FSYNC, HIGH);
    SPI.begin();
    delay(50);
    AD9833reset();
}

void AD9833reset()
{
    WriteRegister(0x100);
    delay(10);
}
```

```
void WriteRegister(int dat)
```

**Proyecto Fin de Ciclo**

Apéndice

```
{  
    SPI.setDataMode(SPI_MODE2);  
    digitalWrite(FSYNC, LOW);  
    delayMicroseconds(10);  
    SPI.transfer(dat>>8);  
    SPI.transfer(dat&0xFF);  
    digitalWrite(FSYNC, HIGH);  
    SPI.setDataMode(SPI_MODE0);  
}  
  
void limpiaDigitos() //limpia los dígitos sobrantes en pantalla  
{  
    if(counter < 100)  
    {  
        lcd.setCursor(2, 0);  
        lcd.print("    ");  
    }else  
    {  
        if(counter < 1000)  
        {  
            lcd.setCursor(3, 0);  
            lcd.print("    ");  
        }else  
        {  
            if(counter < 10000)  
            {  
                lcd.setCursor(4, 0);  
                lcd.print("    ");  
            }else  
            {  
                if(counter < 100000)  
                {  
                    lcd.setCursor(5, 0);  
                    lcd.print("    ");  
                }else  
                {  
                    if(counter < 1000000)  
                    {  
                        lcd.setCursor(6, 0);  
                        lcd.print("    ");  
                    }else  
                    {  
                        if(counter < 10000000)  
                        {  
                            lcd.setCursor(7, 0);  
                            lcd.print("    ");  
                        }  
                    }  
                }  
            }  
        }  
    }  
}
```

**Proyecto Fin de Ciclo****Apéndice**{
}
}**Clmenu.h**

```
#define outputA 6 //Data (Sentido antihorario)
#define outputB 7 //Clock (Sentido horario)
#define sw 5
#define outputA2 2
#define outputB2 3
#define sw2 12
#define MULTI 8
#define back_func 9
#define sensorPIN A0

LiquidCrystal_I2C lcd(0x27, 20, 4);

const int sampleWindow = 50; // Ancho ventana en mS (50 mS = 20Hz)
unsigned long startMillis= millis();
unsigned int Max = 0;
unsigned int Min = 1024;
    // Recopilar durante la ventana
unsigned int sample;
int contadorsonom=0;

float t64=0,t250=0,t500=0,t1000=0,t2000=0,t4000=0,t8000=0,t16000=0;//diferentes resultados del t60 para
diferentes octavas
float trev60teo=0;

int aState;      //Para detectar la posición de los pines del encoder respecto a las placas metálicas internas.
int aLastState;

int aState2;     //Para detectar la posición de los pines del encoder respecto a las placas metálicas internas
del rotary secundario
int aLastState2;

int dentro=0;   //para empezar a medir en el t.rev práctico

extern int banderarotary2;

int multi=0;//variables usadas en la función de generador de ondas
int multi_ant=0;
long int counter = 500;
long int counter_ant = 0;
long int function = 0;
long int function_ant = 0;
int func = 0;
int func_ant = 0;
```

**Proyecto Fin de Ciclo**

Apéndice

```
int bandgenton=0; //variables bandera de diferentes funciones de los submenús
int primerag=0;
int bandt60=0;
int primerat=0;
int bandindat=0;
int primeraindat=0;

LiquidLine linea10(1, 0, "Med. teorica"); //Se inicializan en el segundo bit para dejar hueco para el cursor (flecha), se crean los objetos antes de utilizarlos.
LiquidLine linea20(1, 1, "Med. practica");
LiquidLine linea30(1, 3, "Soporte");
LiquidScreen pantalla1(linea10,linea20,linea30);

LiquidLine linea11(1, 0, "Insercion de datos");
LiquidLine linea12(1, 1, "Resultado ant.");
LiquidLine linea13(1, 2, "Sonometro");
LiquidLine linea14(1, 3, "Atras");
LiquidScreen pantalla2(linea11,linea12,linea13,linea14);

LiquidLine linea21(1, 0, "Gen. tonos");
LiquidLine linea22(1, 1, "T. Reverberacion");
LiquidLine linea23(1, 2, "Resultado ant.");
LiquidLine linea24(1, 3, "Atras");
LiquidScreen pantalla3(linea21,linea22,linea23,linea24);

LiquidLine linea31(1, 0, "Correo electronico");
LiquidLine linea32(1, 1, "Instagram");
LiquidLine linea33(1, 2, "Linkedin");
LiquidLine linea34(1, 3, "Atras");
LiquidScreen pantalla4(linea31,linea32,linea33,linea34);

LiquidMenu menu(lcd,pantalla1,pantalla2,pantalla3,pantalla4);

void fn_atras();
void fn_t60teo();
void fn_t60pra();
void fn_sop();
void fn_indat();
void fn_resant();
void fn_sonom();
void fn_genton();
void fn_t60();
void fn_reant();
void fn_gmail();
void fn_insta();
void fn_linkedin();

//Funciones generales-----
---
void inicializamenu()
```

**Proyecto Fin de Ciclo****Apéndice**

{

```
linea10.set_focusPosition(Position::LEFT); //Posición de la flecha en el menú (Izquierda)
linea20.set_focusPosition(Position::LEFT);
linea30.set_focusPosition(Position::LEFT);
```

```
linea10.attach_function(1, fn_t60teo); // Se le añade una función a la selección de cada línea
linea20.attach_function(1, fn_t60pra);
linea30.attach_function(1, fn_sop);
```

```
menu.add_screen(pantalla1); //Añadimos las pantalla al menú
```

```
linea11.set_focusPosition(Position::LEFT);
linea12.set_focusPosition(Position::LEFT);
linea13.set_focusPosition(Position::LEFT);
linea14.set_focusPosition(Position::LEFT);
```

```
linea11.attach_function(1, fn_indat);
linea12.attach_function(1, fn_resant);
linea13.attach_function(1, fn_sonom);
linea14.attach_function(1, fn_atras);
```

```
menu.add_screen(pantalla2);
```

```
linea21.set_focusPosition(Position::LEFT);
linea22.set_focusPosition(Position::LEFT);
linea23.set_focusPosition(Position::LEFT);
linea24.set_focusPosition(Position::LEFT);
```

```
linea21.attach_function(1, fn_genton);
linea22.attach_function(1, fn_t60);
linea23.attach_function(1, fn_reant);
linea24.attach_function(1, fn_atras);
```

```
menu.add_screen(pantalla3);
```

```
linea31.set_focusPosition(Position::LEFT);
linea32.set_focusPosition(Position::LEFT);
linea33.set_focusPosition(Position::LEFT);
linea34.set_focusPosition(Position::LEFT);
```

```
linea31.attach_function(1, fn_gmail);
linea32.attach_function(1, fn_insta);
linea33.attach_function(1, fn_linkedin);
linea34.attach_function(1, fn_atras);
```

```
menu.add_screen(pantalla4);
```

```
pantalla1.set_displayLineCount(3);
pantalla2.set_displayLineCount(4);
```

**Proyecto Fin de Ciclo**

Apéndice

```
pantalla3.set_displayLineCount(4);
pantalla4.set_displayLineCount(4);
}
```

//Funciones generales-----

```
void selectOption() //para seleccionar las opciones con el encoder principal.
{
  if(digitalRead(sw) == LOW)
  {
    menu.call_function(1);
    delay(500);
  }
}
```

```
void fn_atras() //para volver en el menú principal
{
  menu.change_screen(1);
  menu.set_focusedLine(0);
}
```

//Funciones pantalla 1-----

```
void fn_t60teo() //para moverse por las pantallas del menú principal
{
  menu.change_screen(2);
  menu.set_focusedLine(0);
}
```

```
void fn_t60pra()
{
  menu.change_screen(3);
  menu.set_focusedLine(0);
}
```

```
void fn_sop()
{
  menu.change_screen(4);
  menu.set_focusedLine(0);
}
```

//Funciones pantalla 2-----

```
void fn_indat()
{
```

```
  bandindat=1;
```

**Proyecto Fin de Ciclo****Apéndice**

```
primeraindat=1;
Serial.println("Primera INDAT");

}

void fn_resant() //muestra resultados por pantalla
{
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(" T. Reverberacion ");
    lcd.setCursor(0,1);
    lcd.print(" teorico   ");
    lcd.setCursor(7,3);
    lcd.print(trev60teo); lcd.print(" s");
    delay(4000);

    menu.change_screen(2);
    menu.set_focusedLine(0);

}

void fn_sonom()
{
    const int sampleWindow = 50; // Ancho ventana en mS (50 mS = 20Hz)
    unsigned long startMillis= millis();
    unsigned int Max = 0;
    unsigned int Min = 1024;
    // Recopilar durante la ventana
    unsigned int sample;
    int contadorsonom=0;

    lcd.clear();
    lcd.setCursor(0,0);
    lcd.println(" Sonometro   ");
    lcd.setCursor(0,2);
    lcd.print("(      ) xxxx ");

    while (contadorsonom < 10) {

        while (millis() - startMillis < sampleWindow) {

            Serial.println("Medidas");
            sample = analogRead(sensorPIN);

            if (sample < 1024) {

                if (sample > Max) {
```

**Proyecto Fin de Ciclo**

Apéndice

```
Max = sample; // Actualizar máximo
}

else if (sample < Min) {

    Min = sample; // Actualizar mínimo
}
}

Serial.println("Imprimiendo");
unsigned int picoapico = Max - Min; // Amplitud del sonido
double volts = (picoapico * 5.0) / 1024; // Convertir a tensión
double db = (20. * log(10)) * (volts); //Convertir a dB

lcd.setCursor(0,0);
lcd.println(" Sonometro ");
lcd.setCursor(0,1);
lcd.println(" ");
lcd.setCursor(0,2);
lcd.print("( ) ");
lcd.setCursor(0,3);lcd.print("Sen. sono=");
lcd.print(db);lcd.print("dB ");
lcd.setCursor(18,3);
lcd.print(" ");
Serial.println(db);Serial.println("dB");

if ((db>=1) && (db<=10))
{
lcd.setCursor(0,2);
lcd.print("(.) BAJA ");
}

if ((db>10) && (db<=20))
{
lcd.setCursor(0,2);
lcd.print("(..) BAJA ");
}

if ((db>20) && (db<=30))
{
lcd.setCursor(0,2);
lcd.print("(...) BAJA ");
}

if ((db>30) && (db<=40))
{
lcd.setCursor(0,2);
```

**Proyecto Fin de Ciclo**

Apéndice

```
lcd.print("....    ) MEDIA  ");
}

if ((db>40) && (db<=50))
{
  lcd.setCursor(0,2);
  lcd.print(".....    ) MEDIA  ");
}

if ((db>50) && (db<=60))
{
  lcd.setCursor(0,2);
  lcd.print(".....    ) MEDIA  ");
}

if ((db>60) && (db<=70))
{
  lcd.setCursor(0,2);
  lcd.print(".....    ) ALTA  ");
}

if ((db>70) && (db<=80))
{
  lcd.setCursor(0,2);
  lcd.print(".....    ) ALTA  ");
}

if ((db>80) && (db<=90))
{
  lcd.setCursor(0,2);
  lcd.print(".....    ) ALTA  ");
}

if ((db>90) && (db<=100))
{
  lcd.setCursor(0,2);
  lcd.print(".....    ) M.ALTA  ");
}

Serial.println("Medidas");

delay(1000);
contadorsonom++;
startMillis=millis();
Max=0;
Min=1024;
Serial.println(contadorsonom);

}
```

**Proyecto Fin de Ciclo**

Apéndice

```
menu.change_screen(2);
menu.set.FocusedLine(0);
contadorsonom==0;

}

//Funciones pantalla 3-----
---

void fn_genton()
{
    bandgenton=1;
    primerag=1;
    Serial.println("Primera Genton");
}

void fn_t60()
{
    lcd.clear();
    bandt60=1;
    primerat=1;
    dentro=0;
    //mensaje pulsa la tecla multi para empezar la medida
    do{
        lcd.setCursor(4,1);
        lcd.print("PULSE MULTI");
        Serial.println("Pulse MULTI");
        //esperando la pulsacion de una tecla
    }while(digitalRead(MULTI)!=LOW);
    //genero tono
    dentro=1;
    Serial.println("A");
    //mido

    //imprimo resultados
}

void fn_reant()
{
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("T.Rev.64: ");lcd.print(t64);lcd.print("s");
    lcd.setCursor(0,1);
    lcd.print("T.Rev.250: ");lcd.print(t250);lcd.print("s");
    lcd.setCursor(0,2);
    lcd.print("T.Rev.500: ");lcd.print(t500);lcd.print("s");
}
```

**Proyecto Fin de Ciclo**

Apéndice

```
lcd.setCursor(0,3);
lcd.print("T.Rev.1000: ");lcd.print(t1000);lcd.print("s");
delay(3000);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("T.Rev.2000: ");lcd.print(t2000);lcd.print("s");
lcd.setCursor(0,1);
lcd.print("T.Rev.4000: ");lcd.print(t4000);lcd.print("s");
lcd.setCursor(0,2);
lcd.print("T.Rev.8000: ");lcd.print(t8000);lcd.print("s");
lcd.setCursor(0,3);
lcd.print("T.Rev.16000: ");lcd.print(t16000);lcd.print("s");
delay(3000);

menu.change_screen(3);
menu.set_focusedLine(0);
}

//Funciones pantalla 4-----
---

void fn_gmail()
{
lcd.clear();
lcd.setCursor(0,0);
lcd.print(" sorena2021 ");
lcd.setCursor(0,1);
lcd.print(" @gmail.com ");
lcd.setCursor(14,3);
lcd.print("Ver1.0");
delay(3000);
menu.change_screen(4);
menu.set_focusedLine(0);
}

void fn_insta()
{
lcd.clear();
lcd.setCursor(0,0);
lcd.print(" <User> ");
lcd.setCursor(0,1);
lcd.print(" @Sorena2021 ");
lcd.setCursor(14,3);
lcd.print("Ver1.0");
delay(3000);
menu.change_screen(4);
menu.set_focusedLine(0);
}

void fn_linkedin()
```

**Proyecto Fin de Ciclo**

Apéndice

```
{  
lcd.clear();  
lcd.setCursor(0,0);  
lcd.print("    <User>    ");  
lcd.setCursor(0,1);  
lcd.print(" Juan Miguel    ");  
lcd.setCursor(0,2);  
lcd.print(" Acosta Ortega  ");  
lcd.setCursor(14,3);  
lcd.print("Ver1.0");  
delay(3000);  
menu.change_screen(4);  
menu.set_focusedLine(0);  
}
```



ANEXO

(Datasheets)

- 1 AD9833 (GENERADOR DE SEÑALES)**
- 2 LM836 (AMPLIFICADOR)**
- 3 ARDUINO MEGA R3**
- 4 HD44780U (PANTALLA LCD 20X4)**
- 5 PCA8754 (MÓDULO I2C)**
- 6 KY-040 (ROTARY ENCODER)**

FEATURES

Digitally programmable frequency and phase
12.65 mW power consumption at 3 V
0 MHz to 12.5 MHz output frequency range
28-bit resolution: 0.1 Hz at 25 MHz reference clock
Sinusoidal, triangular, and square wave outputs
2.3 V to 5.5 V power supply
No external components required
3-wire SPI interface
Extended temperature range: -40°C to +105°C
Power-down option
10-lead MSOP package
Qualified for automotive applications

APPLICATIONS

Frequency stimulus/waveform generation
Liquid and gas flow measurement
**Sensory applications: proximity, motion,
and defect detection**
Line loss/attenuation
Test and medical equipment
Sweep/clock generators
Time domain reflectometry (TDR) applications

GENERAL DESCRIPTION

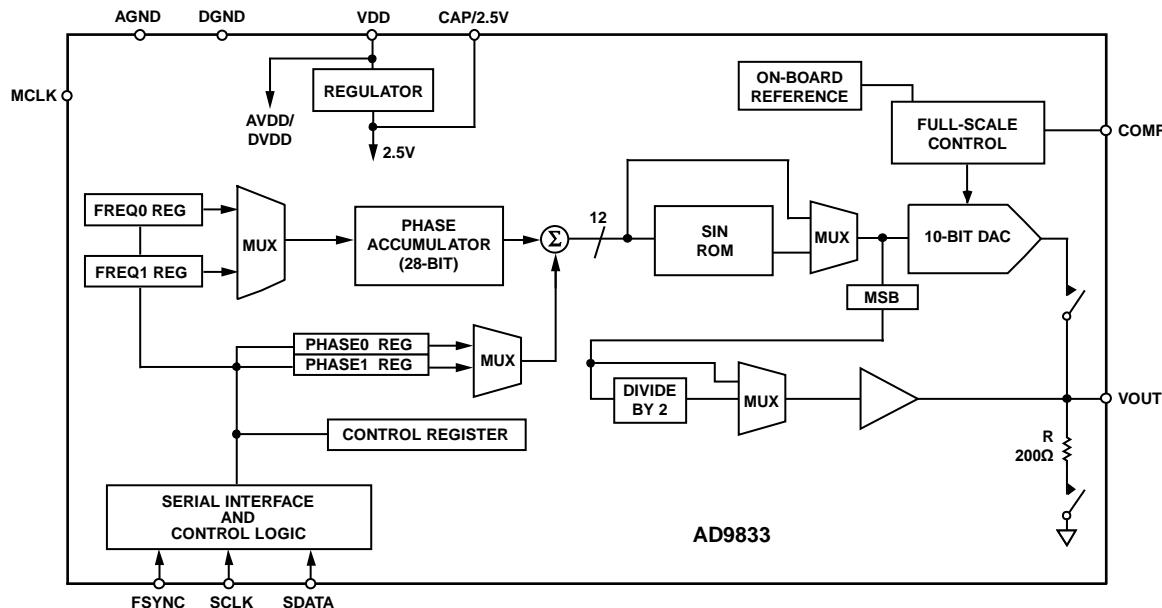
The AD9833 is a low power, programmable waveform generator capable of producing sine, triangular, and square wave outputs. Waveform generation is required in various types of sensing, actuation, and time domain reflectometry (TDR) applications. The output frequency and phase are software programmable, allowing easy tuning. No external components are needed. The frequency registers are 28 bits wide: with a 25 MHz clock rate, resolution of 0.1 Hz can be achieved; with a 1 MHz clock rate, the AD9833 can be tuned to 0.004 Hz resolution.

The AD9833 is written to via a 3-wire serial interface. This serial interface operates at clock rates up to 40 MHz and is compatible with DSP and microcontroller standards. The device operates with a power supply from 2.3 V to 5.5 V.

The AD9833 has a power-down function (SLEEP). This function allows sections of the device that are not being used to be powered down, thus minimizing the current consumption of the part. For example, the DAC can be powered down when a clock output is being generated.

The AD9833 is available in a 10-lead MSOP package.

FUNCTIONAL BLOCK DIAGRAM



02/04/001

Figure 1.

Rev. F

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

Document Feedback

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
 Tel: 781.329.4700 ©2003–2018 Analog Devices, Inc. All rights reserved.
Technical Support www.analog.com

TABLE OF CONTENTS

Features	1
Applications	1
General Description	1
Functional Block Diagram	1
Revision History	2
Specifications.....	3
Timing Characteristics	4
Absolute Maximum Ratings.....	5
ESD Caution.....	5
Pin Configuration and Function Descriptions.....	6
Typical Performance Characteristics	7
Terminology	10
Theory of Operation	11
Circuit Description.....	12
Numerically Controlled Oscillator Plus Phase Modulator ...	12
Sin ROM	12
Digital-to-Analog Converter (DAC)	12
Regulator.....	12
Functional Description.....	13
Serial Interface	13
Powering Up the AD9833	13
Latency Period	13
Control Register	13
Frequency and Phase Registers	15
Reset Function.....	16
Sleep Function	16
VOUT Pin	16
Applications Information	17
Grounding and Layout	17
Interfacing to Microprocessors.....	20
AD9833 to 68HC11/68L11 Interface.....	20
AD9833 to 80C51/80L51 Interface	20
AD9833 to DSP56002 Interface	20
Outline Dimensions.....	21
Ordering Guide	21
Automotive Products	21

REVISION HISTORY

4/2018—Rev. E to Rev. F

Updated Format.....	Universal
Changes to AD9833 to 68HC11/68L11 Interface Section	20
Deleted Evaluation Board Section and Figure 32 to Figure 37;	
Renumbered Sequentially.....	21
Changes to Ordering Guide	21

9/2012—Rev. D to Rev. E

Changed Input Current, I_{INH}/I_{INL} from 10 mA to 10 μ A.....	3
--	---

4/2011—Rev. C to Rev. D

Change to Figure 13	8
Changes to Table 9.....	15
Deleted AD9833 to ADSP-2101/ADSP-2103 Interface	
Section.....	20
Changes to Evaluation Board Section.....	21
Added System Demonstration Platform Section, AD9833	
to SPORT Interface Section, and Evaluation Kit Section.....	21
Changes to Crystal Oscillator vs. External Clock Section	
and Power Supply Section	21
Added Figure 32 and Figure 33; Renumbered Figures	
Sequentially	21
Deleted Prototyping Area Section and Figure 33.....	22

Added Evaluation Board Schematics Section, Figure 34,	
and Figure 35.....	22
Deleted Table 16	23
Added Evaluation Board Layout Section, Figure 36,	
Figure 37, and Figure 38	23
Changes to Ordering Guide	24

9/2010—Rev. B to Rev. C

Changed 20 mW to 12.65 mW in Data Sheet Title	
and Features List.....	1
Changes to Figure 6 Caption and Figure 7.....	7

6/2010—Rev. A to Rev. B

Changes to Features Section	1
Changes to Serial Interface Section.....	13
Changes to VOUT Pin Section	16
Changes to Grounding and Layout Section	17
Updated Outline Dimensions	24
Changes to Ordering Guide	24
Added Automotive Products Section	24

6/2003—Rev. 0 to Rev. A

Updated Ordering Guide	4
------------------------------	---

SPECIFICATIONS

VDD = 2.3 V to 5.5 V, AGND = DGND = 0 V, TA = T_{MIN} to T_{MAX}, R_{SET} = 6.8 kΩ for VOUT, unless otherwise noted.

Table 1.

Parameter ¹	Min	Typ	Max	Unit	Test Conditions/Comments
SIGNAL DAC SPECIFICATIONS					
Resolution		10		Bits	
Update Rate			25	MSPS	
VOUT Maximum		0.65		V	
VOUT Minimum		38		mV	
VOUT Temperature Coefficient		200		ppm/°C	
DC Accuracy					
Integral Nonlinearity		±1.0		LSB	
Differential Nonlinearity		±0.5		LSB	
DDS SPECIFICATIONS (SFDR)					
Dynamic Specifications					
Signal-to-Noise Ratio (SNR)	55	60		dB	f _{MCLK} = 25 MHz, f _{OUT} = f _{MCLK} /4096
Total Harmonic Distortion (THD)		−66	−56	dBc	f _{MCLK} = 25 MHz, f _{OUT} = f _{MCLK} /4096
Spurious-Free Dynamic Range (SFDR)					
Wideband (0 to Nyquist)		−60		dBc	f _{MCLK} = 25 MHz, f _{OUT} = f _{MCLK} /50
Narrow-Band (±200 kHz)		−78		dBc	f _{MCLK} = 25 MHz, f _{OUT} = f _{MCLK} /50
Clock Feedthrough		−60		dBc	
Wake-Up Time		1		ms	
LOGIC INPUTS					
Input High Voltage, V _{INH}	1.7			V	2.3 V to 2.7 V power supply
	2.0			V	2.7 V to 3.6 V power supply
	2.8			V	4.5 V to 5.5 V power supply
Input Low Voltage, V _{INL}		0.5		V	2.3 V to 2.7 V power supply
		0.7		V	2.7 V to 3.6 V power supply
		0.8		V	4.5 V to 5.5 V power supply
Input Current, I _{INH} /I _{INL}		10		µA	
Input Capacitance, C _{IN}		3		pF	
POWER SUPPLIES					
VDD	2.3		5.5	V	f _{MCLK} = 25 MHz, f _{OUT} = f _{MCLK} /4096
I _{DD}		4.5	5.5	mA	I _{DD} code dependent; see Figure 7
Low Power Sleep Mode		0.5		mA	DAC powered down, MCLK running

¹ Operating temperature range is −40°C to +105°C; typical specifications are at +25°C.

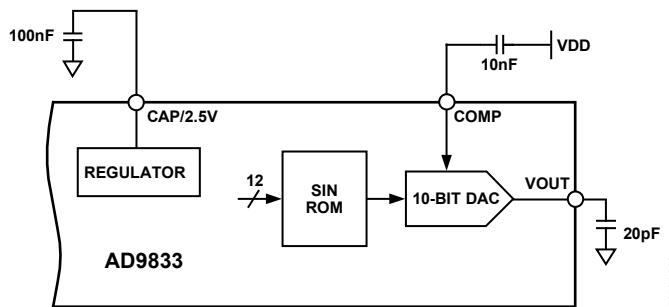


Figure 2. Test Circuit Used to Test Specifications

02704-002

TIMING CHARACTERISTICS

VDD = 2.3 V to 5.5 V, AGND = DGND = 0 V, unless otherwise noted.¹

Table 2.

Parameter	Limit at T _{MIN} to T _{MAX}	Unit	Description
t ₁	40	ns min	MCLK period
t ₂	16	ns min	MCLK high duration
t ₃	16	ns min	MCLK low duration
t ₄	25	ns min	SCLK period
t ₅	10	ns min	SCLK high duration
t ₆	10	ns min	SCLK low duration
t ₇	5	ns min	FSYNC to SCLK falling edge setup time
t _{8 min}	10	ns min	FSYNC to SCLK hold time
t _{8 max}	t ₄ – 5	ns max	
t ₉	5	ns min	Data setup time
t ₁₀	3	ns min	Data hold time
t ₁₁	5	ns min	SCLK high to FSYNC falling edge setup time

¹ Guaranteed by design, not production tested.

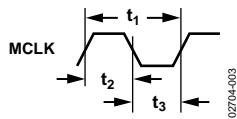
Timing Diagrams

Figure 3. Master Clock

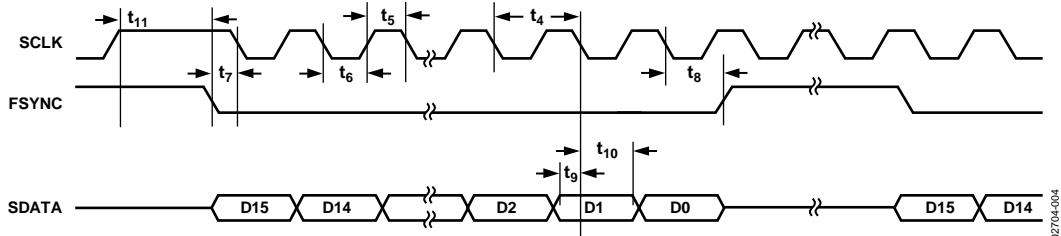


Figure 4. Serial Timing

ABSOLUTE MAXIMUM RATINGS

T_A = 25°C, unless otherwise noted.

Table 3.

Parameter	Rating
VDD to AGND	-0.3 V to +6 V
VDD to DGND	-0.3 V to +6 V
AGND to DGND	-0.3 V to +0.3 V
CAP/2.5V	2.75 V
Digital I/O Voltage to DGND	-0.3 V to VDD + 0.3 V
Analog I/O Voltage to AGND	-0.3 V to VDD + 0.3 V
Operating Temperature Range Industrial (B Version)	-40°C to +105°C
Storage Temperature Range	-65°C to +150°C
Maximum Junction Temperature	150°C
MSOP Package	
θ _{JA} Thermal Impedance	206°C/W
θ _{JC} Thermal Impedance	44°C/W
Lead Temperature, Soldering (10 sec)	300°C
IR Reflow, Peak Temperature	220°C

Stresses at or above those listed under Absolute Maximum Ratings may cause permanent damage to the product. This is a stress rating only; functional operation of the product at these or any other conditions above those indicated in the operational section of this specification is not implied. Operation beyond the maximum operating conditions for extended periods may affect product reliability.

ESD CAUTION



ESD (electrostatic discharge) sensitive device.

Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.

PIN CONFIGURATION AND FUNCTION DESCRIPTIONS

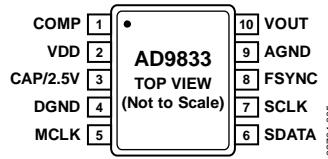


Figure 5. Pin Configuration

Table 4. Pin Function Descriptions

Pin No.	Mnemonic	Description
1	COMP	DAC Bias Pin. This pin is used for decoupling the DAC bias voltage.
2	VDD	Positive Power Supply for the Analog and Digital Interface Sections. The on-board 2.5 V regulator is also supplied from VDD. VDD can have a value from 2.3 V to 5.5 V. A 0.1 μ F and a 10 μ F decoupling capacitor should be connected between VDD and AGND.
3	CAP/2.5V	The digital circuitry operates from a 2.5 V power supply. This 2.5 V is generated from VDD using an on-board regulator when VDD exceeds 2.7 V. The regulator requires a decoupling capacitor of 100 nF typical, which is connected from CAP/2.5V to DGND. If VDD is less than or equal to 2.7 V, CAP/2.5V should be tied directly to VDD.
4	DGND	Digital Ground.
5	MCLK	Digital Clock Input. DDS output frequencies are expressed as a binary fraction of the frequency of MCLK. This clock determines the output frequency accuracy and phase noise.
6	SDATA	Serial Data Input. The 16-bit serial data-word is applied to this input.
7	SCLK	Serial Clock Input. Data is clocked into the AD9833 on each falling edge of SCLK.
8	FSYNC	Active Low Control Input. FSYNC is the frame synchronization signal for the input data. When FSYNC is taken low, the internal logic is informed that a new word is being loaded into the device.
9	AGND	Analog Ground.
10	VOUT	Voltage Output. The analog and digital output from the AD9833 is available at this pin. An external load resistor is not required because the device has a 200 Ω resistor on-board.

TYPICAL PERFORMANCE CHARACTERISTICS

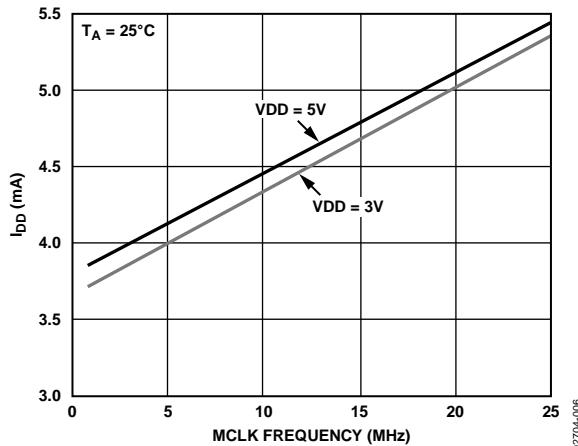


Figure 6. Typical Current Consumption (I_{DD}) vs. MCLK Frequency for $f_{OUT} = MCLK/10$

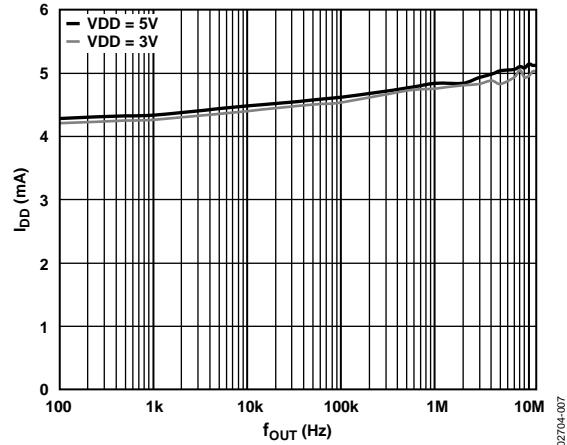


Figure 7. Typical I_{DD} vs. f_{OUT} for $f_{MCLK} = 25\text{ MHz}$

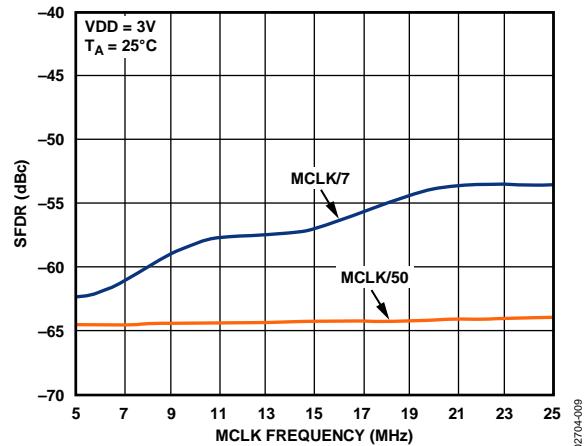


Figure 9. Wideband SFDR vs. MCLK Frequency

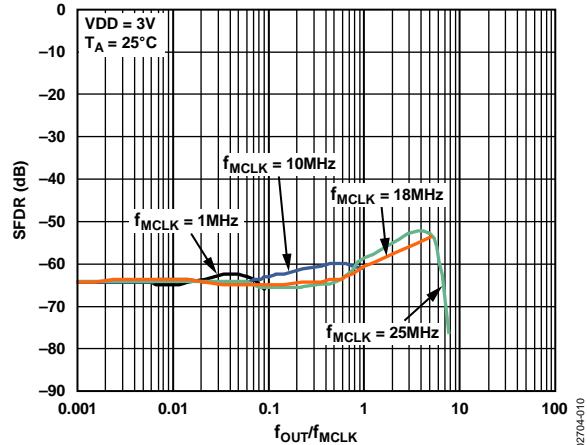


Figure 10. Wideband SFDR vs. f_{OUT}/f_{MCLK} for Various MCLK Frequencies

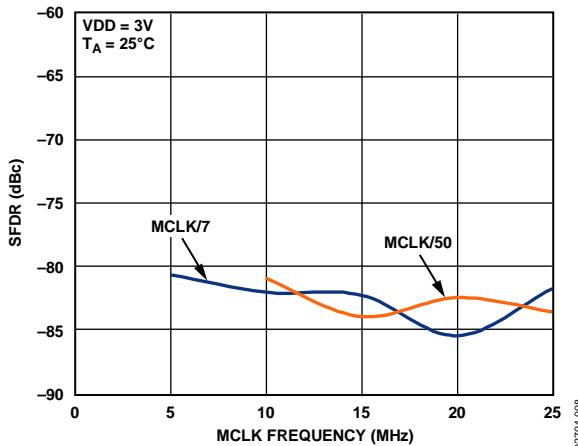


Figure 8. Narrow-Band SFDR vs. MCLK Frequency

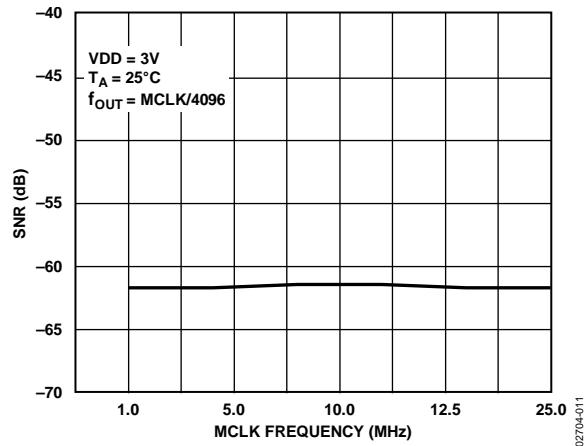


Figure 11. SNR vs. MCLK Frequency

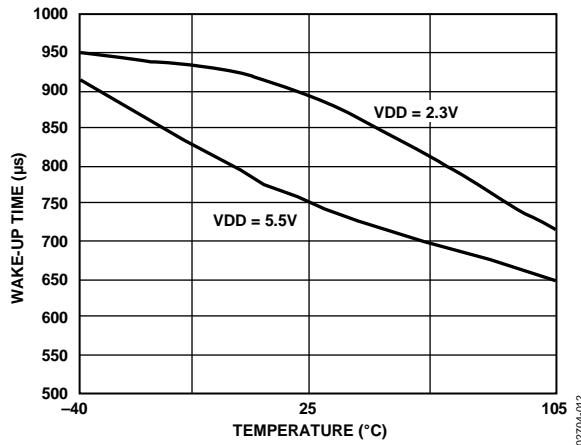
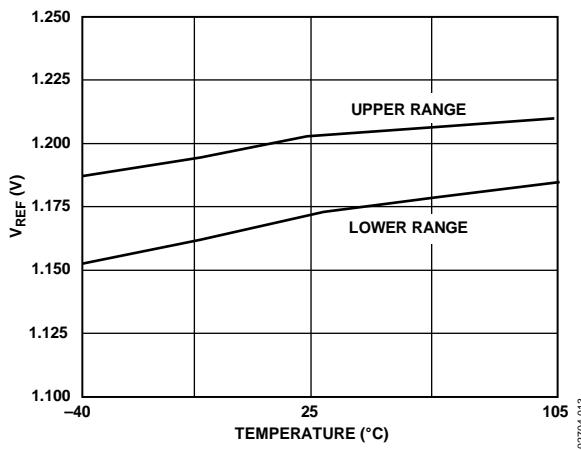
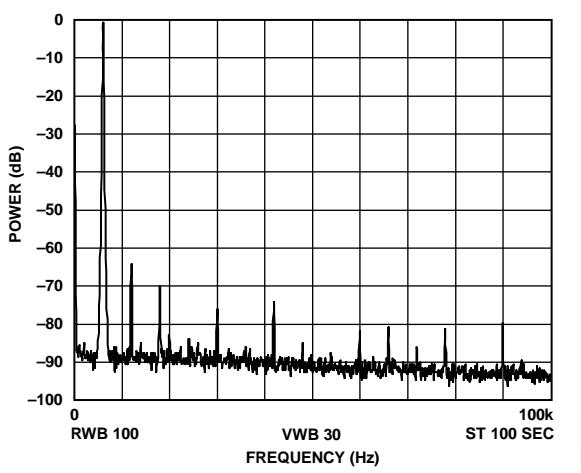
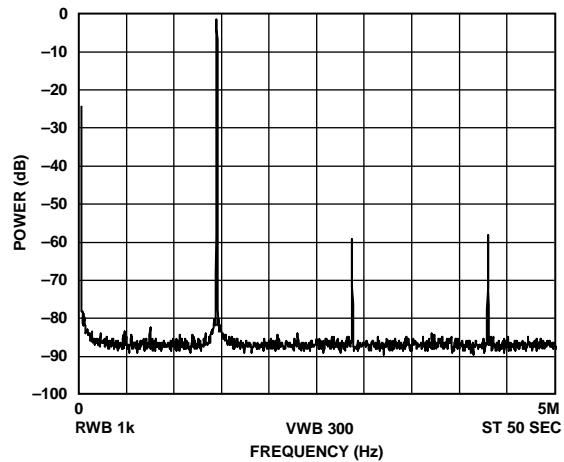
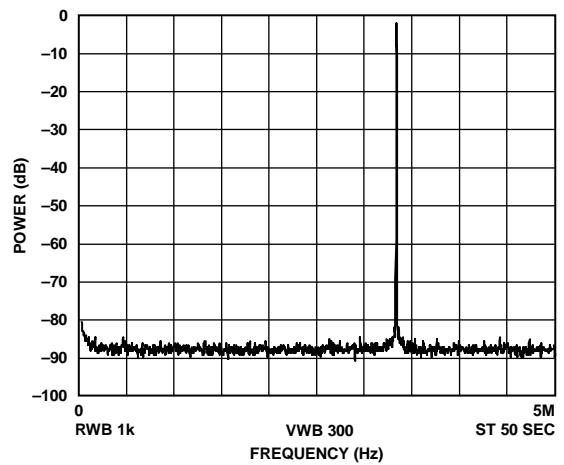
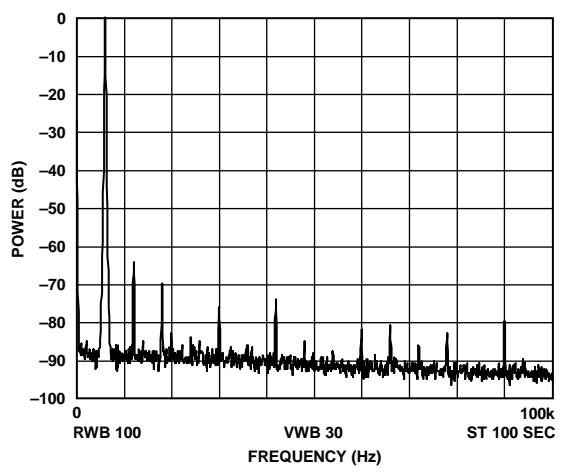


Figure 12. Wake-Up Time vs. Temperature

Figure 13. V_{REF} vs. TemperatureFigure 14. Power vs. Frequency, $f_{MCLK} = 10 \text{ MHz}$, $f_{OUT} = 2.4 \text{ kHz}$, Frequency Word = 0x000FBA9Figure 15. Power vs. Frequency, $f_{MCLK} = 10 \text{ MHz}$, $f_{OUT} = 1.43 \text{ MHz} = f_{MCLK}/7$, Frequency Word = 0x2492492Figure 16. Power vs. Frequency, $f_{MCLK} = 10 \text{ MHz}$, $f_{OUT} = 3.33 \text{ MHz} = f_{MCLK}/3$, Frequency Word = 0x5555555Figure 17. Power vs. Frequency, $f_{MCLK} = 25 \text{ MHz}$, $f_{OUT} = 6 \text{ kHz}$, Frequency Word = 0x000FBA9

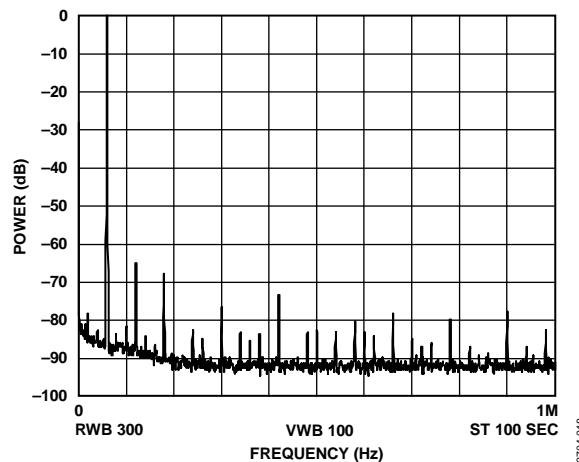


Figure 18. Power vs. Frequency, $f_{MCLK} = 25 \text{ MHz}$, $f_{OUT} = 60 \text{ kHz}$,
Frequency Word = 0x009D495

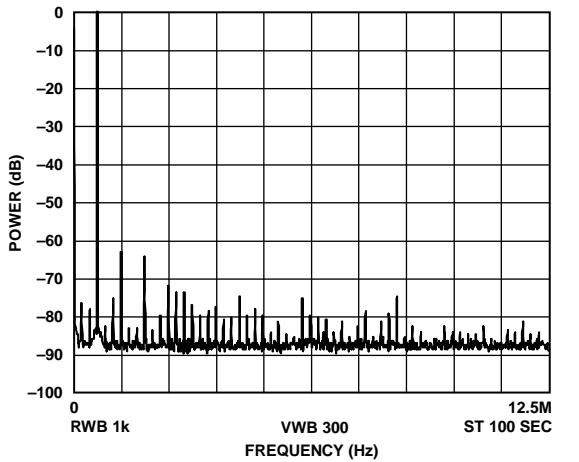


Figure 19. Power vs. Frequency, $f_{MCLK} = 25 \text{ MHz}$, $f_{OUT} = 600 \text{ kHz}$,
Frequency Word = 0x0624DD3

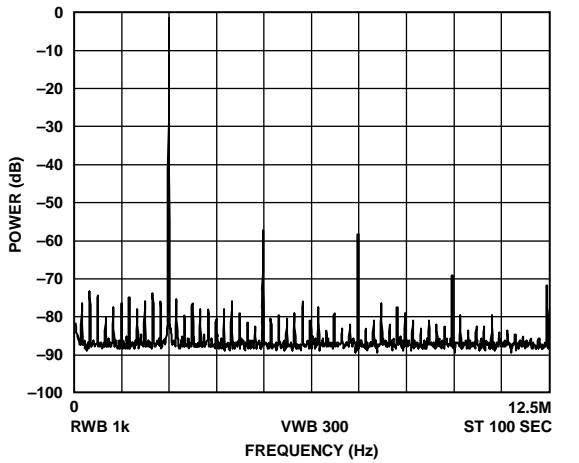


Figure 20. Power vs. Frequency, $f_{MCLK} = 25 \text{ MHz}$, $f_{OUT} = 2.4 \text{ MHz}$,
Frequency Word = 0x189374D

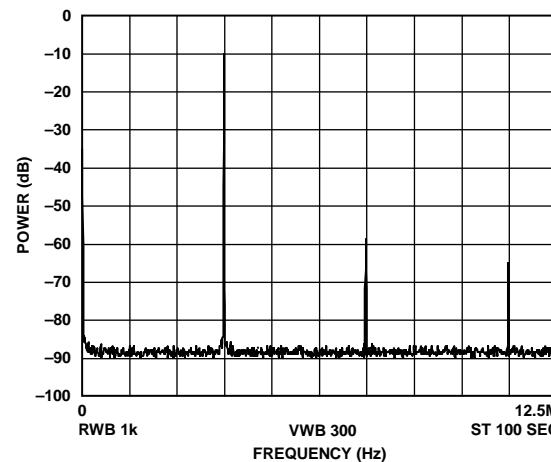


Figure 21. Power vs. Frequency, $f_{MCLK} = 25 \text{ MHz}$, $f_{OUT} = 3.857 \text{ MHz} = f_{MCLK}/7$,
Frequency Word = 0x2492492

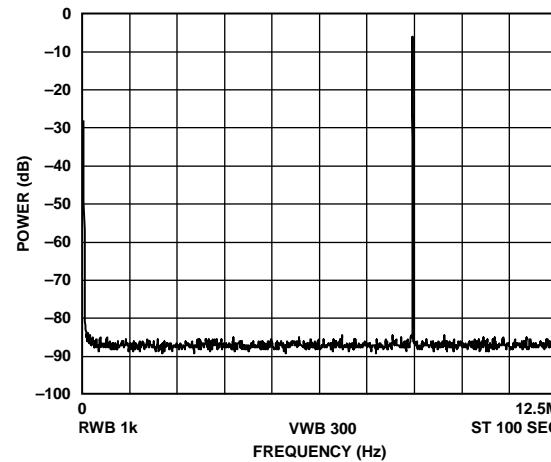


Figure 22. Power vs. Frequency, $f_{MCLK} = 25 \text{ MHz}$, $f_{OUT} = 8.333 \text{ MHz} = f_{MCLK}/3$,
Frequency Word = 0x5555555

TERMINOLOGY

Integral Nonlinearity (INL)

INL is the maximum deviation of any code from a straight line passing through the endpoints of the transfer function. The endpoints of the transfer function are zero scale, a point 0.5 LSB below the first code transition (000 ... 00 to 000 ... 01), and full scale, a point 0.5 LSB above the last code transition (111 ... 10 to 111 ... 11). The error is expressed in LSBs.

Differential Nonlinearity (DNL)

DNL is the difference between the measured and ideal 1 LSB change between two adjacent codes in the DAC. A specified DNL of ± 1 LSB maximum ensures monotonicity.

Output Compliance

Output compliance refers to the maximum voltage that can be generated at the output of the DAC to meet the specifications. When voltages greater than that specified for the output compliance are generated, the AD9833 may not meet the specifications listed in the data sheet.

Spurious-Free Dynamic Range (SFDR)

Along with the frequency of interest, harmonics of the fundamental frequency and images of these frequencies are present at the output of a DDS device. SFDR refers to the largest spur or harmonic present in the band of interest. The wideband SFDR gives the magnitude of the largest spur or harmonic relative to the magnitude of the fundamental frequency in the zero to Nyquist bandwidth. The narrow-band SFDR gives the attenuation of the largest spur or harmonic in a bandwidth of ± 200 kHz about the fundamental frequency.

Total Harmonic Distortion (THD)

THD is the ratio of the rms sum of harmonics to the rms value of the fundamental. For the AD9833, THD is defined as

$$\text{THD} = 20 \log \sqrt{\frac{V_2^2 + V_3^2 + V_4^2 + V_5^2 + V_6^2}{V_1^2}}$$

where:

V_1 is the rms amplitude of the fundamental.

V_2, V_3, V_4, V_5 , and V_6 are the rms amplitudes of the second through sixth harmonics.

Signal-to-Noise Ratio (SNR)

SNR is the ratio of the rms value of the measured output signal to the rms sum of all other spectral components below the Nyquist frequency. The value for SNR is expressed in decibels.

Clock Feedthrough

There is feedthrough from the MCLK input to the analog output. Clock feedthrough refers to the magnitude of the MCLK signal relative to the fundamental frequency in the output spectrum of the AD9833.

THEORY OF OPERATION

Sine waves are typically thought of in terms of their magnitude form: $a(t) = \sin(\omega t)$. However, these sine waves are nonlinear and not easy to generate except through piecewise construction. On the other hand, the angular information is linear in nature. That is, the phase angle rotates through a fixed angle for each unit of time. The angular rate depends on the frequency of the signal by the traditional rate of $\omega = 2\pi f$.

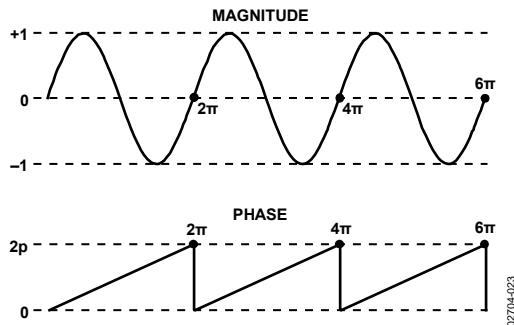


Figure 23. Sine Wave

Knowing that the phase of a sine wave is linear and given a reference interval (clock period), the phase rotation for that period can be determined.

$$\Delta\text{Phase} = \omega\Delta t$$

Solving for ω ,

$$\omega = \Delta\text{Phase}/\Delta t = 2\pi f$$

Solving for f and substituting the reference clock frequency for the reference period ($1/f_{\text{MCLK}} = \Delta t$)

$$f = \Delta\text{Phase} \times f_{\text{MCLK}}/2\pi$$

The AD9833 builds the output based on this simple equation. A simple DDS chip can implement this equation with three major subcircuits: numerically controlled oscillator (NCO) and phase modulator, SIN ROM, and digital-to-analog converter (DAC).

Each subcircuit is described in the Circuit Description section.

CIRCUIT DESCRIPTION

The AD9833 is a fully integrated direct digital synthesis (DDS) chip. The chip requires one reference clock, one low precision resistor, and decoupling capacitors to provide digitally created sine waves up to 12.5 MHz. In addition to the generation of this RF signal, the chip is fully capable of a broad range of simple and complex modulation schemes. These modulation schemes are fully implemented in the digital domain, allowing accurate and simple realization of complex modulation algorithms using DSP techniques.

The internal circuitry of the AD9833 consists of the following main sections: a numerically controlled oscillator (NCO), frequency and phase modulators, SIN ROM, a DAC, and a regulator.

NUMERICALLY CONTROLLED OSCILLATOR PLUS PHASE MODULATOR

This consists of two frequency select registers, a phase accumulator, two phase offset registers, and a phase offset adder. The main component of the NCO is a 28-bit phase accumulator. Continuous time signals have a phase range of 0 to 2π . Outside this range of numbers, the sinusoid functions repeat themselves in a periodic manner. The digital implementation is no different. The accumulator simply scales the range of phase numbers into a multibit digital word. The phase accumulator in the AD9833 is implemented with 28 bits. Therefore, in the AD9833, $2\pi = 2^{28}$. Likewise, the Δ Phase term is scaled into this range of numbers:

$$0 < \Delta\text{Phase} < 2^{28} - 1$$

With these substitutions, the previous equation becomes

$$f = \Delta\text{Phase} \times f_{MCLK} / 2^{28}$$

where $0 < \Delta\text{Phase} < 2^{28} - 1$.

The input to the phase accumulator can be selected from either the FREQ0 register or the FREQ1 register and is controlled by the FSELECT bit. NCOs inherently generate continuous phase signals, thus avoiding any output discontinuity when switching between frequencies.

Following the NCO, a phase offset can be added to perform phase modulation using the 12-bit phase registers. The contents of one of these phase registers are added to the most significant bits of the NCO. The AD9833 has two phase registers; their resolution is $2\pi/4096$.

SIN ROM

To make the output from the NCO useful, it must be converted from phase information into a sinusoidal value. Because phase information maps directly into amplitude, the SIN ROM uses the digital phase information as an address to a lookup table and converts the phase information into amplitude. Although the NCO contains a 28-bit phase accumulator, the output of the NCO is truncated to 12 bits. Using the full resolution of the phase accumulator is impractical and unnecessary, because this would require a lookup table of 2^{28} entries. It is necessary only to have sufficient phase resolution such that the errors due to truncation are smaller than the resolution of the 10-bit DAC. This requires that the SIN ROM have two bits of phase resolution more than the 10-bit DAC.

The SIN ROM is enabled using the mode bit (D1) in the control register (see Table 15).

DIGITAL-TO-ANALOG CONVERTER (DAC)

The AD9833 includes a high impedance, current source 10-bit DAC. The DAC receives the digital words from the SIN ROM and converts them into the corresponding analog voltages.

The DAC is configured for single-ended operation. An external load resistor is not required because the device has a $200\ \Omega$ resistor on-board. The DAC generates an output voltage of typically 0.6 V p-p.

REGULATOR

VDD provides the power supply required for the analog section and the digital section of the AD9833. This supply can have a value of 2.3 V to 5.5 V.

The internal digital section of the AD9833 is operated at 2.5 V. An on-board regulator steps down the voltage applied at VDD to 2.5 V. When the applied voltage at the VDD pin of the AD9833 is less than or equal to 2.7 V, the CAP/2.5V and VDD pins should be tied together, thus bypassing the on-board regulator.

FUNCTIONAL DESCRIPTION

SERIAL INTERFACE

The AD9833 has a standard 3-wire serial interface that is compatible with the SPI, QSPI™, MICROWIRE®, and DSP interface standards.

Data is loaded into the device as a 16-bit word under the control of a serial clock input, SCLK. The timing diagram for this operation is given in Figure 4.

The FSYNC input is a level-triggered input that acts as a frame synchronization and chip enable. Data can be transferred into the device only when FSYNC is low. To start the serial data transfer, FSYNC should be taken low, observing the minimum FSYNC-to-SCLK falling edge setup time, t_7 . After FSYNC goes low, serial data is shifted into the input shift register of the device on the falling edges of SCLK for 16 clock pulses. FSYNC may be taken high after the 16th falling edge of SCLK, observing the minimum SCLK falling edge to FSYNC rising edge time, t_8 . Alternatively, FSYNC can be kept low for a multiple of 16 SCLK pulses and then brought high at the end of the data transfer. In this way, a continuous stream of 16-bit words can be loaded while FSYNC is held low; FSYNC goes high only after the 16th SCLK falling edge of the last word loaded.

The SCLK can be continuous, or it can idle high or low between write operations. In either case, it must be high when FSYNC goes low (t_{11}).

For an example of how to program the AD9833, see the [AN-1070 Application Note](#) on the Analog Devices, Inc., website.

POWERING UP THE AD9833

The flowchart in Figure 26 shows the operating routine for the AD9833. When the AD9833 is powered up, the part should be reset. This resets the appropriate internal registers to 0 to provide an analog output of midscale.

To avoid spurious DAC outputs during AD9833 initialization, the reset bit should be set to 1 until the part is ready to begin generating an output. A reset does not reset the phase, frequency, or control registers. These registers will contain invalid data and, therefore, should be set to known values by the user. The reset bit should then be set to 0 to begin generating an output. The data appears on the DAC output seven or eight MCLK cycles after the reset bit is set to 0.

LATENCY PERIOD

A latency period is associated with each asynchronous write operation in the AD9833. If a selected frequency or phase register is loaded with a new word, there is a delay of seven or eight MCLK cycles before the analog output changes. The delay can be seven or eight cycles, depending on the position of the MCLK rising edge when the data is loaded into the destination register.

CONTROL REGISTER

The AD9833 contains a 16-bit control register that allows the user to configure the operation of the AD9833. All control bits other than the mode bit are sampled on the internal falling edge of MCLK.

Table 6 describes the individual bits of the control register. The different functions and the various output options of the AD9833 are described in more detail in the Frequency and Phase Registers section.

To inform the AD9833 that the contents of the control register will be altered, D15 and D14 must be set to 0, as shown in Table 5.

Table 5. Control Register Bits

D15	D14	D13	D0
0	0		Control Bits

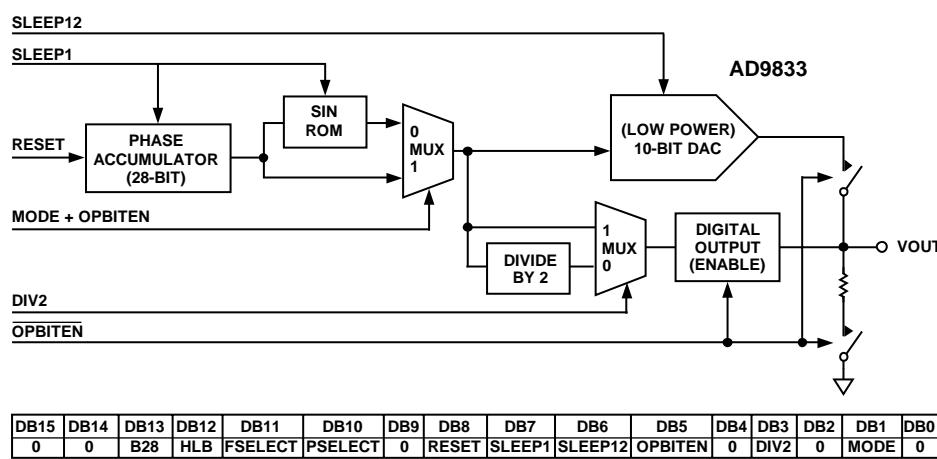


Figure 24. Function of Control Bits

Table 6. Description of Bits in the Control Register

Bit	Name	Function
D13	B28	Two write operations are required to load a complete word into either of the frequency registers. B28 = 1 allows a complete word to be loaded into a frequency register in two consecutive writes. The first write contains the 14 LSBs of the frequency word, and the next write contains the 14 MSBs. The first two bits of each 16-bit word define the frequency register to which the word is loaded, and should therefore be the same for both of the consecutive writes. See Table 8 for the appropriate addresses. The write to the frequency register occurs after both words have been loaded; therefore, the register never holds an intermediate value. An example of a complete 28-bit write is shown in Table 9. When B28 = 0, the 28-bit frequency register operates as two 14-bit registers, one containing the 14 MSBs and the other containing the 14 LSBs. This means that the 14 MSBs of the frequency word can be altered independent of the 14 LSBs, and vice versa. To alter the 14 MSBs or the 14 LSBs, a single write is made to the appropriate frequency address. The control bit D12 (HLB) informs the AD9833 whether the bits to be altered are the 14 MSBs or 14 LSBs.
D12	HLB	This control bit allows the user to continuously load the MSBs or LSBs of a frequency register while ignoring the remaining 14 bits. This is useful if the complete 28-bit resolution is not required. HLB is used in conjunction with D13 (B28). This control bit indicates whether the 14 bits being loaded are being transferred to the 14 MSBs or 14 LSBs of the addressed frequency register. D13 (B28) must be set to 0 to be able to change the MSBs and LSBs of a frequency word separately. When D13 (B28) = 1, this control bit is ignored. HLB = 1 allows a write to the 14 MSBs of the addressed frequency register. HLB = 0 allows a write to the 14 LSBs of the addressed frequency register.
D11	FSELECT	The FSELECT bit defines whether the FREQ0 register or the FREQ1 register is used in the phase accumulator.
D10	PSELECT	The PSELECT bit defines whether the PHASE0 register or the PHASE1 register data is added to the output of the phase accumulator.
D9	Reserved	This bit should be set to 0.
D8	Reset	Reset = 1 resets internal registers to 0, which corresponds to an analog output of midscale. Reset = 0 disables reset. This function is explained further in Table 13.
D7	SLEEP1	When SLEEP1 = 1, the internal MCLK clock is disabled, and the DAC output remains at its present value because the NCO is no longer accumulating. When SLEEP1 = 0, MCLK is enabled. This function is explained further in Table 14.
D6	SLEEP12	SLEEP12 = 1 powers down the on-chip DAC. This is useful when the AD9833 is used to output the MSB of the DAC data. SLEEP12 = 0 implies that the DAC is active. This function is explained further in Table 14.
D5	OPBITEN	The function of this bit, in association with D1 (mode), is to control what is output at the VOUT pin. This is explained further in Table 15. When OPBITEN = 1, the output of the DAC is no longer available at the VOUT pin. Instead, the MSB (or MSB/2) of the DAC data is connected to the VOUT pin. This is useful as a coarse clock source. The DIV2 bit controls whether it is the MSB or MSB/2 that is output. When OPBITEN = 0, the DAC is connected to VOUT. The mode bit determines whether it is a sinusoidal or a ramp output that is available.
D4	Reserved	This bit must be set to 0.
D3	DIV2	DIV2 is used in association with D5 (OPBITEN). This is explained further in Table 15. When DIV2 = 1, the MSB of the DAC data is passed directly to the VOUT pin. When DIV2 = 0, the MSB/2 of the DAC data is output at the VOUT pin.
D2	Reserved	This bit must be set to 0.
D1	Mode	This bit is used in association with OPBITEN (D5). The function of this bit is to control what is output at the VOUT pin when the on-chip DAC is connected to VOUT. This bit should be set to 0 if the control bit OPBITEN = 1. This is explained further in Table 15. When mode = 1, the SIN ROM is bypassed, resulting in a triangle output from the DAC. When mode = 0, the SIN ROM is used to convert the phase information into amplitude information, which results in a sinusoidal signal at the output.
D0	Reserved	This bit must be set to 0.

FREQUENCY AND PHASE REGISTERS

The AD9833 contains two frequency registers and two phase registers, which are described in Table 7.

Table 7. Frequency and Phase Registers

Register	Size	Description
FREQ0	28 bits	Frequency Register 0. When the FSELECT bit = 0, this register defines the output frequency as a fraction of the MCLK frequency.
FREQ1	28 bits	Frequency Register 1. When the FSELECT bit = 1, this register defines the output frequency as a fraction of the MCLK frequency.
PHASE0	12 bits	Phase Offset Register 0. When the PSELECT bit = 0, the contents of this register are added to the output of the phase accumulator.
PHASE1	12 bits	Phase Offset Register 1. When the PSELECT bit = 1, the contents of this register are added to the output of the phase accumulator.

The analog output from the AD9833 is

$$f_{MCLK}/2^{28} \times FREQREG$$

where *FREQREG* is the value loaded into the selected frequency register. This signal is phase shifted by

$$2\pi/4096 \times PHASEREG$$

where *PHASEREG* is the value contained in the selected phase register. Consideration must be given to the relationship of the selected output frequency and the reference clock frequency to avoid unwanted output anomalies.

The flowchart in Figure 28 shows the routine for writing to the frequency and phase registers of the AD9833.

Writing to a Frequency Register

When writing to a frequency register, Bit D15 and Bit D14 give the address of the frequency register.

Table 8. Frequency Register Bits

D15	D14	D13	D0
0	1	MSB 14 FREQ0 REG bits	LSB
1	0	MSB 14 FREQ1 REG bits	LSB

If the user wants to change the entire contents of a frequency register, two consecutive writes to the same address must be performed because the frequency registers are 28 bits wide. The first write contains the 14 LSBs, and the second write contains the 14 MSBs. For this mode of operation, the B28 (D13) control bit should be set to 1. An example of a 28-bit write is shown in Table 9.

Table 9. Writing 0xFFFFC000 to the FREQ0 Register

SDATA Input	Result of Input Word
0010 0000 0000 0000	Control word write (D15, D14 = 00), B28 (D13) = 1, HLB (D12) = X
0100 0000 0000 0000	FREQ0 register write (D15, D14 = 01), 14 LSBs = 0x0000
0111 1111 1111 1111	FREQ0 register write (D15, D14 = 01), 14 MSBs = 0x3FFF

In some applications, the user does not need to alter all 28 bits of the frequency register. With coarse tuning, only the 14 MSBs are altered, while with fine tuning, only the 14 LSBs are altered. By setting the B28 (D13) control bit to 0, the 28-bit frequency register operates as two, 14-bit registers, one containing the 14 MSBs and the other containing the 14 LSBs. This means that the 14 MSBs of the frequency word can be altered independent of the 14 LSBs, and vice versa. Bit HLB (D12) in the control register identifies which 14 bits are being altered. Examples of this are shown in Table 10 and Table 11.

Table 10. Writing 0x3FFF to the 14 LSBs of the FREQ1 Register

SDATA Input	Result of Input Word
0000 0000 0000 0000	Control word write (D15, D14 = 00), B28 (D13) = 0; HLB (D12) = 0, that is, LSBs
1011 1111 1111 1111	FREQ1 REG write (D15, D14 = 10), 14 LSBs = 0x3FFF

Table 11. Writing 0x00FF to the 14 MSBs of the FREQ0 Register

SDATA Input	Result of Input Word
0001 0000 0000 0000	Control word write (D15, D14 = 00), B28 (D13) = 0, HLB (D12) = 1, that is, MSBs
0100 0000 1111 1111	FREQ0 REG write (D15, D14 = 01), 14 MSBs = 0x00FF

Writing to a Phase Register

When writing to a phase register, Bit D15 and Bit D14 are set to 11. Bit D13 identifies which phase register is being loaded.

Table 12. Phase Register Bits

D15	D14	D13	D12	D11	D0
1	1	0	X	MSB 12 PHASE0 bits	LSB
1	1	1	X	MSB 12 PHASE1 bits	LSB

RESET FUNCTION

The reset function resets appropriate internal registers to 0 to provide an analog output of midscale. Reset does not reset the phase, frequency, or control registers. When the AD9833 is powered up, the part should be reset. To reset the AD9833, set the reset bit to 1. To take the part out of reset, set the bit to 0. A signal appears at the DAC to output eight MCLK cycles after reset is set to 0.

Table 13. Applying the Reset Function

Reset Bit	Result
0	No reset applied
1	Internal registers reset

SLEEP FUNCTION

Sections of the AD9833 that are not in use can be powered down to minimize power consumption. This is done using the sleep function. The parts of the chip that can be powered down are the internal clock and the DAC. The bits required for the sleep function are outlined in Table 14.

Table 14. Applying the Sleep Function

SLEEP1 Bit	SLEEP12 Bit	Result
0	0	No power-down
0	1	DAC powered down
1	0	Internal clock disabled
1	1	Both the DAC powered down and the internal clock disabled

DAC Powered Down

This is useful when the AD9833 is used to output the MSB of the DAC data only. In this case, the DAC is not required; therefore, it can be powered down to reduce power consumption.

Internal Clock Disabled

When the internal clock of the AD9833 is disabled, the DAC output remains at its present value because the NCO is no longer accumulating. New frequency, phase, and control words can be written to the part when the SLEEP1 control bit is active. The synchronizing clock is still active, which means that the selected frequency and phase registers can also be changed using the control bits. Setting the SLEEP1 bit to 0 enables the MCLK. Any changes made to the registers while SLEEP1 is active will be seen at the output after a latency period.

VOUT PIN

The AD9833 offers a variety of outputs from the chip, all of which are available from the VOUT pin. The choice of outputs is the MSB of the DAC data, a sinusoidal output, or a triangle output.

The OPBITEN (D5) and mode (D1) bits in the control register are used to decide which output is available from the AD9833.

MSB of the DAC Data

The MSB of the DAC data can be output from the AD9833. By setting the OPBITEN (D5) control bit to 1, the MSB of the DAC data is available at the VOUT pin. This is useful as a coarse clock source. This square wave can also be divided by 2 before being output. The DIV2 (D3) bit in the control register controls the frequency of this output from the VOUT pin.

Sinusoidal Output

The SIN ROM is used to convert the phase information from the frequency and phase registers into amplitude information that results in a sinusoidal signal at the output. To have a sinusoidal output from the VOUT pin, set the mode (D1) bit to 0 and the OPBITEN (D5) bit to 0.

Triangle Output

The SIN ROM can be bypassed so that the truncated digital output from the NCO is sent to the DAC. In this case, the output is no longer sinusoidal. The DAC will produce a 10-bit linear triangular function. To have a triangle output from the VOUT pin, set the mode (D1) bit = 1.

Note that the SLEEP12 bit must be 0 (that is, the DAC is enabled) when using this pin.

Table 15. Outputs from the VOUT Pin

OPBITEN Bit	Mode Bit	DIV2 Bit	VOUT Pin
0	0	X ¹	Sinusoid
0	1	X ¹	Triangle
1	0	0	DAC data MSB/2
1	0	1	DAC data MSB
1	1	X ¹	Reserved

¹ X = don't care.

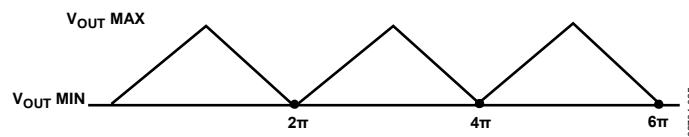


Figure 25. Triangle Output

APPLICATIONS INFORMATION

Because of the various output options available from the part, the AD9833 can be configured to suit a wide variety of applications.

One of the areas where the AD9833 is suitable is in modulation applications. The part can be used to perform simple modulation, such as FSK. More complex modulation schemes, such as GMSK and QPSK, can also be implemented using the AD9833.

In an FSK application, the two frequency registers of the AD9833 are loaded with different values. One frequency represents the space frequency, while the other represents the mark frequency. Using the FSELECT bit in the control register of the AD9833, the user can modulate the carrier frequency between the two values.

The AD9833 has two phase registers, which enables the part to perform PSK. With phase-shift keying, the carrier frequency is phase shifted, the phase being altered by an amount that is related to the bit stream being input to the modulator.

The AD9833 is also suitable for signal generator applications. Because the MSB of the DAC data is available at the VOUT pin, the device can be used to generate a square wave.

With its low current consumption, the part is suitable for applications in which it can be used as a local oscillator.

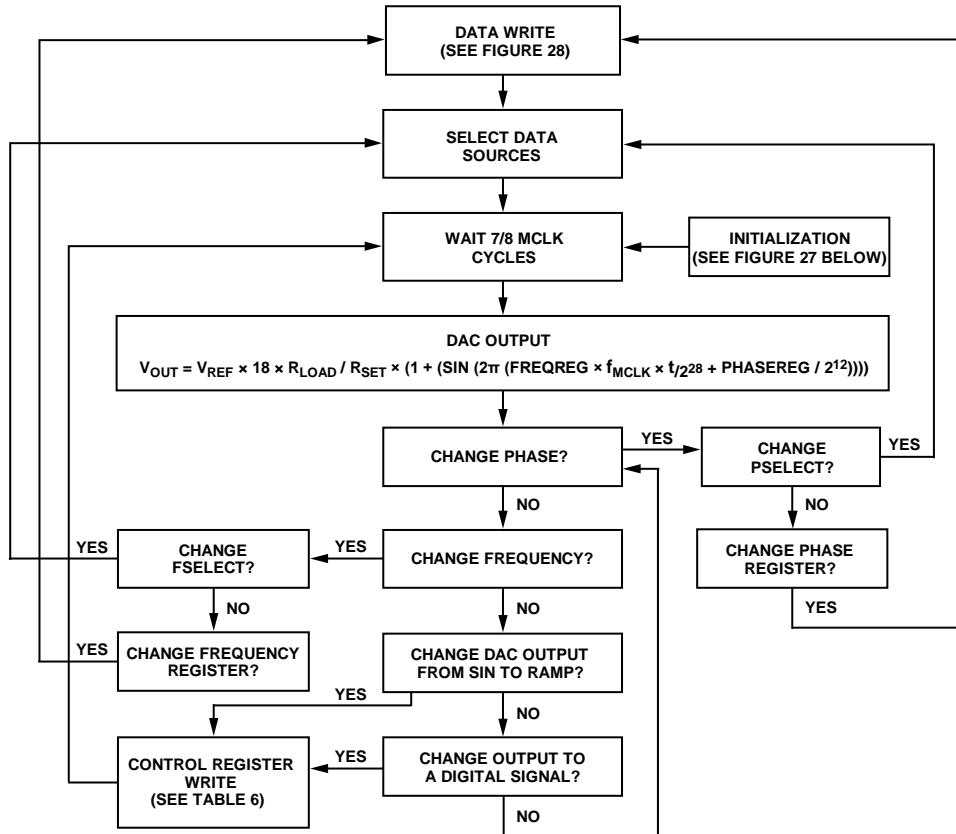
GROUNDING AND LAYOUT

The printed circuit board (PCB) that houses the AD9833 should be designed so that the analog and digital sections are separated and confined to certain areas of the board. This facilitates the use of ground planes that can be separated easily. A minimum etch technique is generally best for ground planes because it gives the best shielding. Digital and analog ground planes should be joined in one place only. If the AD9833 is the only device requiring an AGND-to-DGND connection, then the ground planes should be connected at the AGND and DGND pins of the AD9833. If the AD9833 is in a system where multiple devices require AGND-to-DGND connections, the connection should be made at one point only, a star ground point that should be established as close as possible to the AD9833.

Avoid running digital lines under the device as these couple noise onto the die. The analog ground plane should be allowed to run under the AD9833 to avoid noise coupling. The power supply lines to the AD9833 should use as large a track as possible to provide low impedance paths and reduce the effects of glitches on the power supply line. Fast switching signals, such as clocks, should be shielded with digital ground to avoid radiating noise to other sections of the board.

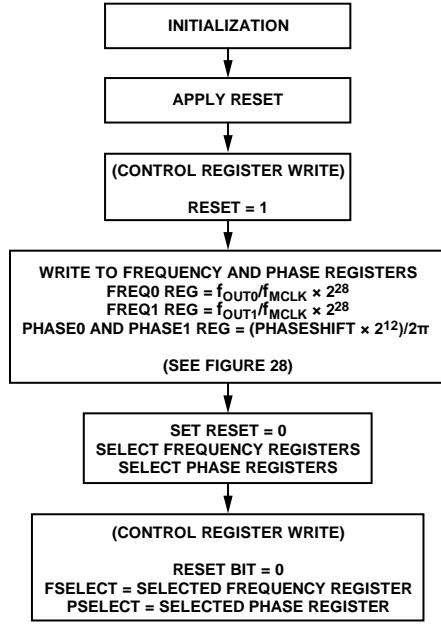
Avoid crossover of digital and analog signals. Traces on opposite sides of the board should run at right angles to each other. This reduces the effects of feedthrough through the board. A microstrip technique is by far the best, but it is not always possible with a double-sided board. In this technique, the component side of the board is dedicated to ground planes, and signals are placed on the other side.

Good decoupling is important. The AD9833 should have supply bypassing of $0.1\ \mu\text{F}$ ceramic capacitors in parallel with $10\ \mu\text{F}$ tantalum capacitors. To achieve the best performance from the decoupling capacitors, they should be placed as close as possible to the device, ideally right up against the device.



02704-026

Figure 26. Flowchart for AD9833 Initialization and Operation



02704-027

Figure 27. Flowchart for Initialization

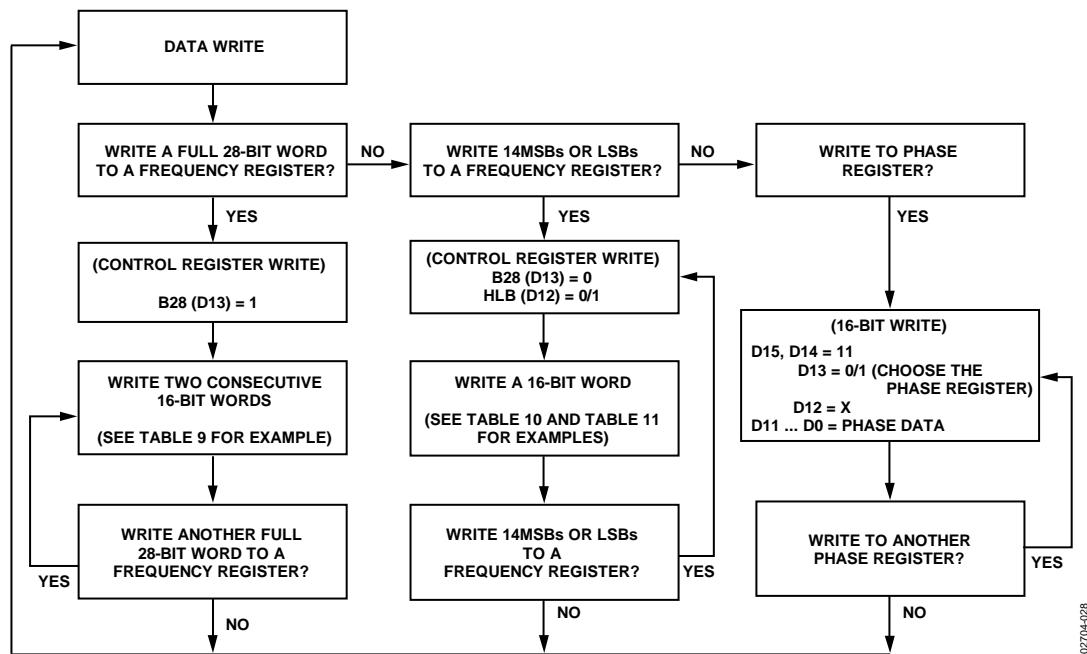


Figure 28. Flowchart for Data Writes

INTERFACING TO MICROPROCESSORS

The AD9833 has a standard serial interface that allows the part to interface directly with several microprocessors. The device uses an external serial clock to write the data or control information into the device. The serial clock can have a frequency of 40 MHz maximum. The serial clock can be continuous, or it can idle high or low between write operations. When data or control information is written to the AD9833, FSYNC is taken low and is held low until the 16 bits of data are written into the AD9833. The FSYNC signal frames the 16 bits of information that are loaded into the AD9833.

AD9833 TO 68HC11/68L11 INTERFACE

Figure 29 shows the serial interface between the AD9833 and the 68HC11/68L11 microcontroller. The microcontroller is configured as the master by setting the MSTR bit in the SPCR to 1. This setting provides a serial clock on SCK; the MOSI output drives the serial data line SDATA. Because the microcontroller does not have a dedicated frame sync pin, the FSYNC signal is derived from a port line (PC7). The setup conditions for correct operation of the interface are as follows:

- SCK idles high between write operations (CPOL = 1)
- Data is valid on the SCK falling edge (CPHA = 0)

When data is being transmitted to the AD9833, the FSYNC line is taken low (PC7). Serial data from the 68HC11/68L11 is transmitted in 8-bit bytes with only eight falling clock edges occurring in the transmit cycle. Data is transmitted MSB first. To load data into the AD9833, PC7 is held low after the first eight bits are transferred, and a second serial write operation is performed to the AD9833. Only after the second eight bits are transferred should FSYNC be taken high again.

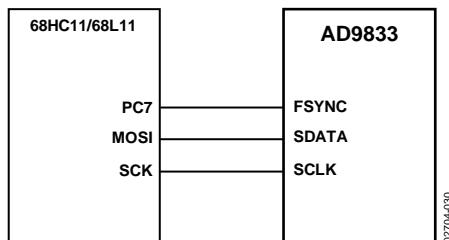


Figure 29. 68HC11/68L11 to AD9833 Interface

AD9833 TO 80C51/80L51 INTERFACE

Figure 30 shows the serial interface between the AD9833 and the 80C51/80L51 microcontroller. The microcontroller is operated in Mode 0 so that TxD of the 80C51/80L51 drives SCLK of the AD9833, and RxD drives the serial data line SDATA. The FSYNC signal is derived from a bit programmable pin on the port (P3.3 is shown in Figure 30).

When data is to be transmitted to the AD9833, P3.3 is taken low. The 80C51/80L51 transmits data in 8-bit bytes, thus only eight falling SCLK edges occur in each cycle. To load the remaining eight bits to the AD9833, P3.3 is held low after the first eight bits are transmitted, and a second write operation is initiated to transmit the second byte of data. P3.3 is taken high following the completion of the second write operation. SCLK should idle high between the two write operations.

The 80C51/80L51 outputs the serial data in a format that has the LSB first. The AD9833 accepts the MSB first (the four MSBs are the control information, the next four bits are the address, and the eight LSBs contain the data when writing to a destination register). Therefore, the transmit routine of the 80C51/80L51 must take this into account and rearrange the bits so that the MSB is output first.

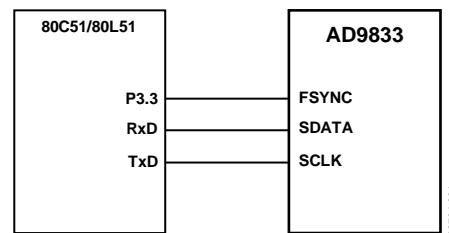


Figure 30. 80C51/80L51 to AD9833 Interface

AD9833 TO DSP56002 INTERFACE

Figure 31 shows the interface between the AD9833 and the DSP56002. The DSP56002 is configured for normal mode asynchronous operation with a gated internal clock (SYN = 0, GCK = 1, SCKD = 1). The frame sync pin is generated internally (SC2 = 1), the transfers are 16 bits wide (WL1 = 1, WL0 = 0), and the frame sync signal frames the 16 bits (FSL = 0). The frame sync signal is available on the SC2 pin, but it must be inverted before it is applied to the AD9833. The interface to the DSP56000/DSP56001 is similar to that of the DSP56002.

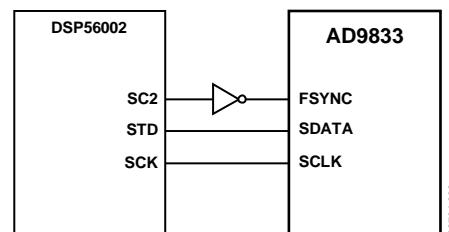
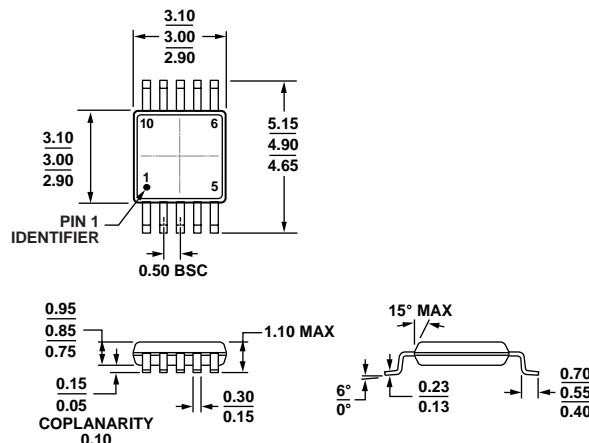


Figure 31. DSP56002 to AD9833 Interface

OUTLINE DIMENSIONS



COMPLIANT TO JEDEC STANDARDS MO-187-BA

091709-A

Figure 32. 10-Lead Mini Small Outline Package [MSOP]
(RM-10)
Dimensions shown in millimeters

ORDERING GUIDE

Model ^{1, 2, 3}	Temperature Range	Package Description	Package Option	Marking Code
AD9833BRM	-40°C to +105°C	10-Lead MSOP	RM-10	DJB
AD9833BRM-REEL7	-40°C to +105°C	10-Lead MSOP	RM-10	DJB
AD9833BRMZ	-40°C to +105°C	10-Lead MSOP	RM-10	D68
AD9833BRMZ-REEL	-40°C to +105°C	10-Lead MSOP	RM-10	D68
AD9833BRMZ-REEL7	-40°C to +105°C	10-Lead MSOP	RM-10	D68
AD9833WBRMZ-REEL	-40°C to +105°C	10-Lead MSOP	RM-10	D68
EVAL-AD9833SDZ		Evaluation Board		

¹ Z = RoHS Compliant Part.² W = Qualified for Automotive Applications.³ The evaluation board for the AD9833 requires the system demonstration platform (SDP) board, which is sold separately.

AUTOMOTIVE PRODUCTS

The AD9833WBRMZ-REEL model is available with controlled manufacturing to support the quality and reliability requirements of automotive applications. Note that this automotive model may have specifications that differ from the commercial models; therefore, designers should review the Specifications section of this data sheet carefully. Only the automotive grade product shown is available for use in automotive applications. Contact your local Analog Devices account representative for specific product ordering information and to obtain the specific Automotive Reliability reports for these models.

I²C refers to a communications protocol originally developed by Philips Semiconductors (now NXP Semiconductors).

LM386 Low Voltage Audio Power Amplifier

1 Features

- Battery Operation
- Minimum External Parts
- Wide Supply Voltage Range: 4 V–12 V or 5 V–18 V
- Low Quiescent Current Drain: 4 mA
- Voltage Gains from 20 to 200
- Ground-Referenced Input
- Self-Centering Output Quiescent Voltage
- Low Distortion: 0.2% ($A_V = 20$, $V_S = 6$ V, $R_L = 8 \Omega$, $P_O = 125$ mW, $f = 1$ kHz)
- Available in 8-Pin MSOP Package

2 Applications

- AM-FM Radio Amplifiers
- Portable Tape Player Amplifiers
- Intercoms
- TV Sound Systems
- Line Drivers
- Ultrasonic Drivers
- Small Servo Drivers
- Power Converters

3 Description

The LM386M-1 and LM386MX-1 are power amplifiers designed for use in low voltage consumer applications. The gain is internally set to 20 to keep external part count low, but the addition of an external resistor and capacitor between pins 1 and 8 will increase the gain to any value from 20 to 200.

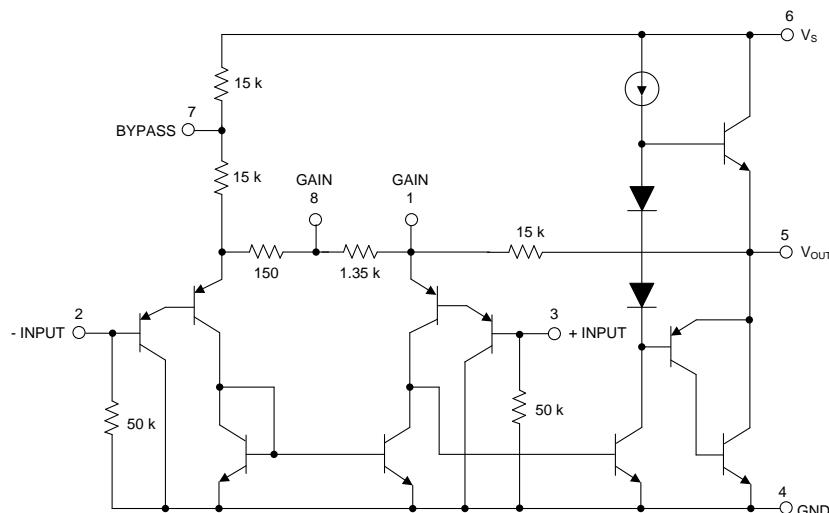
The inputs are ground referenced while the output automatically biases to one-half the supply voltage. The quiescent power drain is only 24 mW when operating from a 6-V supply, making the LM386M-1 and LM386MX-1 ideal for battery operation.

Device Information⁽¹⁾

PART NUMBER	PACKAGE	BODY SIZE (NOM)
LM386N-1	PDIP (8)	9.60 mm × 6.35 mm
LM386N-3	PDIP (8)	9.60 mm × 6.35 mm
LM386N-4	PDIP (8)	9.60 mm × 6.35 mm
LM386M-1	SOIC (8)	4.90 mm × 3.90 mm
LM386MX-1	SOIC (8)	4.90 mm × 3.90 mm
LM386MMX-1	VSSOP (8)	3.00 mm × 3.00 mm

(1) For all available packages, see the orderable addendum at the end of the data sheet.

Schematic



An IMPORTANT NOTICE at the end of this data sheet addresses availability, warranty, changes, use in safety-critical applications, intellectual property matters and other important disclaimers. PRODUCTION DATA.

Table of Contents

1 Features	1	9 Application and Implementation	8
2 Applications	1	9.1 Application Information.....	8
3 Description	1	9.2 Typical Application	8
4 Revision History.....	2	10 Power Supply Recommendations	15
5 Pin Configuration and Functions	3	11 Layout.....	16
6 Specifications.....	3	11.1 Layout Guidelines	16
6.1 Absolute Maximum Ratings	3	11.2 Layout Examples.....	16
6.2 ESD Ratings	3	12 Device and Documentation Support	18
6.3 Recommended Operating Conditions	4	12.1 Device Support.....	18
6.4 Thermal Information	4	12.2 Documentation Support	18
6.5 Electrical Characteristics.....	4	12.3 Related Links	18
6.6 Typical Characteristics	5	12.4 Receiving Notification of Documentation Updates	18
7 Parameter Measurement Information	6	12.5 Community Resources.....	18
8 Detailed Description	7	12.6 Trademarks	18
8.1 Overview	7	12.7 Electrostatic Discharge Caution	18
8.2 Functional Block Diagram	7	12.8 Glossary	18
8.3 Feature Description	7	13 Mechanical, Packaging, and Orderable	19
8.4 Device Functional Modes.....	7	Information	

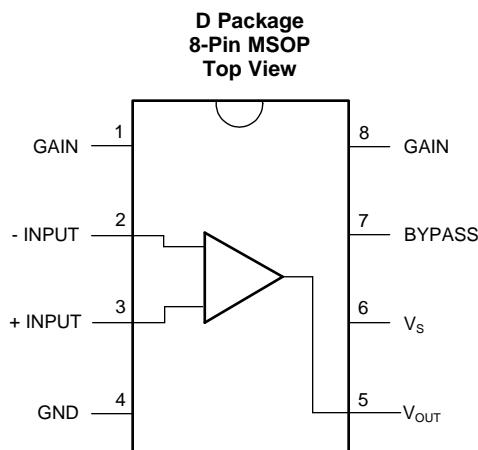
4 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Revision B (March 2017) to Revision C	Page
• Changed devices LM386M-1/LM386MX-1 To: LM386 in the data sheet title	1
• Changed From: LM386N-4 To: Speaker Impedance in the <i>Recommended Operating Conditions</i> table	4
• Changed From: 5 Ω to 12 Ω To: 5 V to 12 V for Supply Voltage in Table 1	8
• Changed kW To: k Ω in the <i>Gain Control</i> section	8
• Changed kW To: k Ω in the <i>Input Biasing</i> section.....	9
• Changed Figure 11	9
• Changed From: 5 Ω to 12 Ω To: 5 V to 12 V for Supply Voltage in Table 2	10
• Changed Figure 13	10
• Changed From: 5 Ω to 12 Ω To: 5 V to 12 V for Supply Voltage in Table 3	11
• Changed Figure 15	11
• Changed From: 5 Ω to 12 Ω To: 5 V to 12 V for Supply Voltage in Table 4	12
• Changed Figure 17	12
• Changed From: 5 Ω to 12 Ω To: 5 V to 12 V for Supply Voltage in Table 5	13
• Changed From: 5 Ω to 12 Ω To: 5 V to 12 V for Supply Voltage in Table 6	14
• Changed Figure 21	14
• Changed From: 5 Ω to 12 Ω To: 5 V to 12 V for Supply Voltage in Table 7	15
• Changed Figure 23	15

Changes from Revision A (May 2004) to Revision B	Page
• Added LM386MX-1 device to the data sheet.	1
• Added Device Information, Application and Implementation, Power Supply Recommendation, Layout, and Device and Documentation Support sections.....	1
• Inserted Functional Block Diagram.....	7

5 Pin Configuration and Functions



Pin Functions

PIN		TYPE	DESCRIPTION
NAME	NO.		
GAIN	1	—	Gain setting pin
-INPUT	2	I	Inverting input
+INPUT	3	I	Noninverting input
GND	4	P	Ground reference
V _{OUT}	5	O	Output
V _S	6	P	Power supply voltage
BYPASS	7	O	Bypass decoupling path
GAIN	8	—	Gain setting pin

6 Specifications

6.1 Absolute Maximum Ratings

over operating free-air temperature range (unless otherwise noted)⁽¹⁾

		MIN	MAX	UNIT
Supply Voltage, V _{CC}	LM386N-1/-3, LM386M-1		15	V
	LM386N-4		22	
Package Dissipation	LM386N		1.25	W
	LM386M		0.73	
	LM386MM-1		0.595	
Input Voltage, V _I		-0.4	0.4	V
Storage temperature, T _{stg}		-65	150	°C

- (1) Stresses beyond those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. These are stress ratings only, which do not imply functional operation of the device at these or any other conditions beyond those indicated under *Recommended Operating Conditions*. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

6.2 ESD Ratings

		VALUE	UNIT
V _(ESD)	Electrostatic discharge	Human-body model (HBM), per ANSI/ESDA/JEDEC JS-001 ⁽¹⁾	±1000
		Charged-device model (CDM), per JEDEC specification JESD22-C101 ⁽²⁾	±1000

(1) JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process.

(2) JEDEC document JEP157 states that 250-V CDM allows safe manufacturing with a standard ESD control process.

6.3 Recommended Operating Conditions

over operating free-air temperature range (unless otherwise noted)

		MIN	NOM	MAX	UNIT
V _{CC}	Supply Voltage	4		12	V
	LM386N-4	5		18	V
Speaker Impedance		4			Ω
VI		-0.4		0.4	V
TA		0		70	°C

6.4 Thermal Information

THERMAL METRIC ⁽¹⁾		LM386	LM386	LM386	UNIT
		D (SOIC)	DGK (VSSOP)	P (PDIP)	
		8	8	8	
R _{θJA}	Junction-to-ambient thermal resistance	115.7	169.3	53.4	°C/W
R _{θJC(top)}	Junction-to-case (top) thermal resistance	59.7	73.1	42.1	°C/W
R _{θJB}	Junction-to-board thermal resistance	56.2	100.2	30.6	°C/W
Ψ _{JT}	Junction-to-top characterization parameter	12.4	9.2	19.0	°C/W
Ψ _{JB}	Junction-to-board characterization parameter	55.6	99.1	50.5	°C/W

- (1) For more information about traditional and new thermal metrics, see the [Semiconductor and IC Package Thermal Metrics](#) application report.

6.5 Electrical Characteristics

over operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
V _S	LM386N-1, -3, LM386M-1, LM386MM-1	4		12	V
	LM386N-4	5		18	
I _Q	V _S = 6 V, V _{IN} = 0		4	8	mA
P _{OUT}	V _S = 6 V, R _L = 8 Ω, THD = 10% (LM386N-1, LM386M-1, LM386MM-1)	250	325		mW
	V _S = 9 V, R _L = 8 Ω, THD = 10% (LM386N-3)	500	700		
	V _S = 16 V, R _L = 32 Ω, THD = 10% (LM386N-4)	700	100		
A _V	V _S = 6 V, f = 1 kHz		26		dB
	10 μF from Pin 1 to 8		46		
BW	V _S = 6 V, Pins 1 and 8 Open		300		kHz
THD	V _S = 6 V, R _L = 8 Ω, POUT = 125 mW f = 1 kHz, Pins 1 and 8 Open		0.2%		
PSRR	V _S = 6 V, f = 1 kHz, CBYPASS = 10 μF Pins 1 and 8 Open, Referred to Output		50		dB
R _{IN}			50		kΩ
I _{BIAS}	V _S = 6 V, Pins 2 and 3 Open		250		nA

6.6 Typical Characteristics

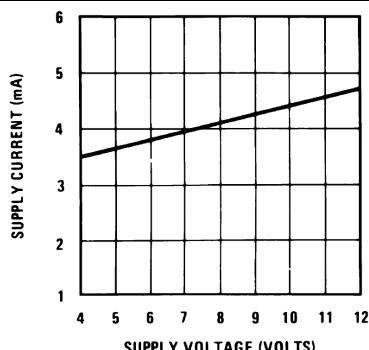


Figure 1. Supply Current vs Supply Voltage

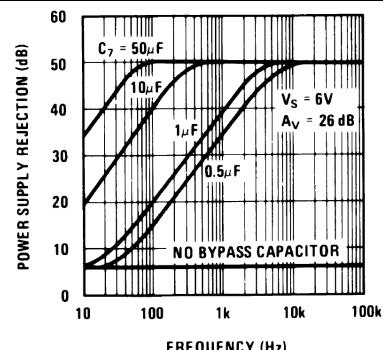


Figure 2. Power Supply Rejection vs Frequency

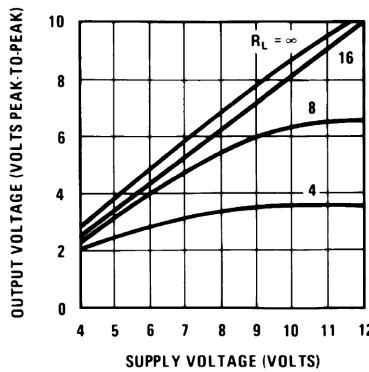


Figure 3. Output Voltage vs Supply Voltage

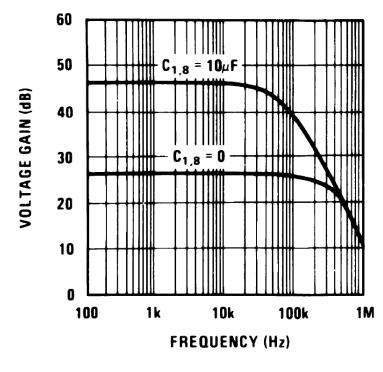


Figure 4. Voltage Gain vs Frequency

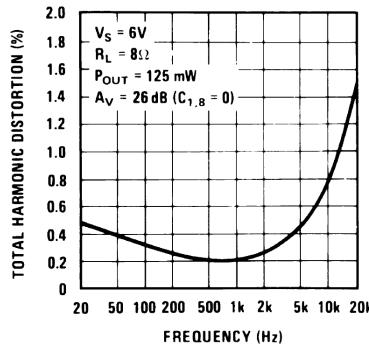


Figure 5. Total Harmonic Distortion vs Frequency

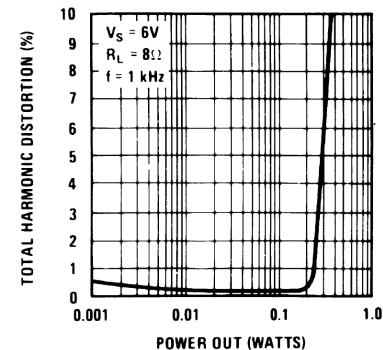


Figure 6. Total Harmonic Distortion vs Power Out

Typical Characteristics (continued)

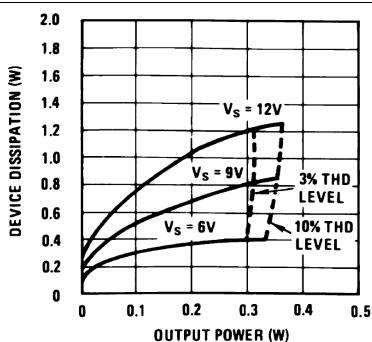


Figure 7. Device Dissipation vs Output Power

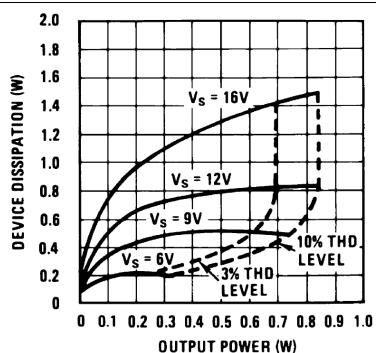


Figure 8. Device Dissipation vs Output Power

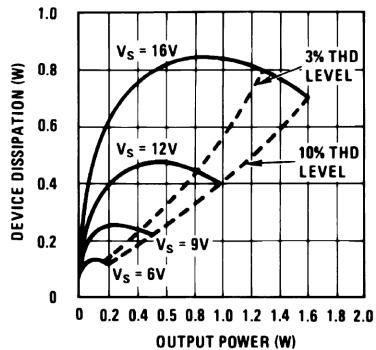


Figure 9. Device Dissipation vs Output Power

7 Parameter Measurement Information

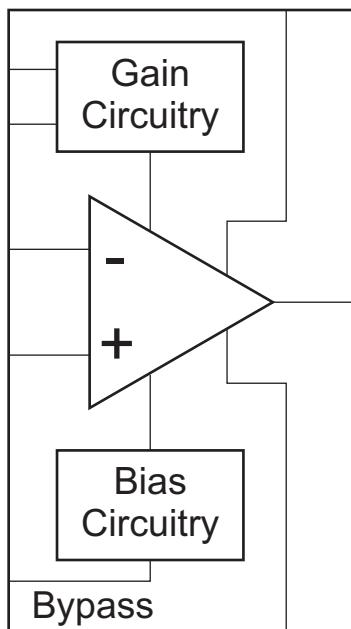
All parameters are measured according to the conditions described in the [Specifications](#) section.

8 Detailed Description

8.1 Overview

The LM386 is a mono low voltage amplifier that can be used in a variety of applications. It can drive loads from $4\ \Omega$ to $32\ \Omega$. The gain is internally set to 20 but it can be modified from 20 to 200 by placing a resistor and capacitor between pins 1 and 8. This device comes in three different 8-pin packages as PDIP, SOIC and VSSOP to fit in different applications.

8.2 Functional Block Diagram



8.3 Feature Description

There is an internal $1.35\text{-K}\Omega$ resistor that sets the gain of this device to 20. The gain can be modified from 20 to 200. Detailed information about gain setting can be found in the [Detailed Design Procedure](#) section.

8.4 Device Functional Modes

As this is an Op Amp it can be used in different configurations to fit in several applications. The internal gain setting resistor allows the LM386 to be used in a very low part count system. In addition a series resistor can be placed between pins 1 and 5 to modify the gain and frequency response for specific applications.

9 Application and Implementation

NOTE

Information in the following applications sections is not part of the TI component specification, and TI does not warrant its accuracy or completeness. TI's customers are responsible for determining suitability of components for their purposes. Customers should validate and test their design implementation to confirm system functionality.

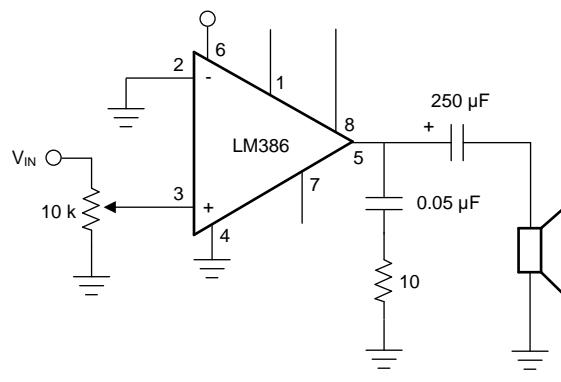
9.1 Application Information

Below are shown different setups that show how the LM386 can be implemented in a variety of applications.

9.2 Typical Application

9.2.1 LM386 with Gain = 20

Figure 10 shows the minimum part count application that can be implemented using LM386. Its gain is internally set to 20.



Copyright © 2017, Texas Instruments Incorporated

Figure 10. LM386 with Gain = 20

9.2.1.1 Design Requirements

Table 1. Design Parameters

DESIGN PARAMETER	EXAMPLE VALUE
Load Impedance	4 Ω to 32 Ω
Supply Voltage	5 V to 12 V

9.2.1.2 Detailed Design Procedure

9.2.1.2.1 Gain Control

To make the LM386 a more versatile amplifier, two pins (1 and 8) are provided for gain control. With pins 1 and 8 open the 1.35-kΩ resistor sets the gain at 20 (26 dB). If a capacitor is put from pin 1 to 8, bypassing the 1.35-kΩ resistor, the gain will go up to 200 (46 dB). If a resistor is placed in series with the capacitor, the gain can be set to any value from 20 to 200. Gain control can also be done by capacitively coupling a resistor (or FET) from pin 1 to ground.

Additional external components can be placed in parallel with the internal feedback resistors to tailor the gain and frequency response for individual applications. For example, we can compensate poor speaker bass response by frequency shaping the feedback path. This is done with a series RC from pin 1 to 5 (paralleling the internal 15-kΩ resistor). For 6 dB effective bass boost: $R \approx 15 \text{ k}\Omega$, the lowest value for good stable operation is $R = 10 \text{ k}\Omega$ if pin 8 is open. If pins 1 and 8 are bypassed then R as low as 2 kΩ can be used. This restriction is because the amplifier is only compensated for closed-loop gains greater than 9.

9.2.1.2.2 Input Biasing

The schematic shows that both inputs are biased to ground with a $50\text{ k}\Omega$ resistor. The base current of the input transistors is about 250 nA , so the inputs are at about 12.5 mV when left open. If the dc source resistance driving the LM386 is higher than $250\text{ k}\Omega$ it will contribute very little additional offset (about 2.5 mV at the input, 50 mV at the output). If the dc source resistance is less than $10\text{ k}\Omega$, then shorting the unused input to ground will keep the offset low (about 2.5 mV at the input, 50 mV at the output). For dc source resistances between these values we can eliminate excess offset by putting a resistor from the unused input to ground, equal in value to the dc source resistance. Of course all offset problems are eliminated if the input is capacitively coupled.

When using the LM386 with higher gains (bypassing the $1.35\text{ k}\Omega$ resistor between pins 1 and 8) it is necessary to bypass the unused input, preventing degradation of gain and possible instabilities. This is done with a $0.1\text{ }\mu\text{F}$ capacitor or a short to ground depending on the dc source resistance on the driven input.

9.2.1.3 Application Curve

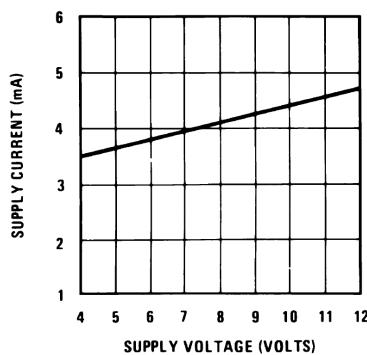
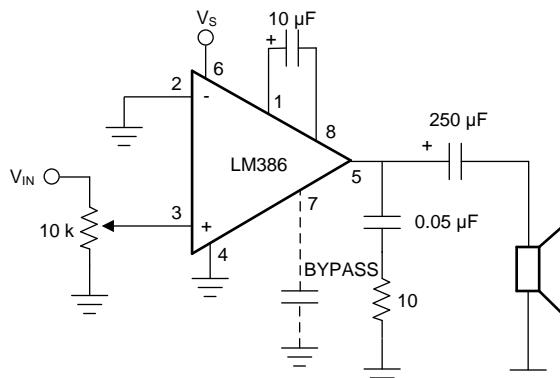


Figure 11. Supply Current vs Supply Voltage

9.2.2 LM386 with Gain = 200



Copyright © 2017, Texas Instruments Incorporated

Figure 12. LM386 with Gain = 200

9.2.2.1 Design Requirements

Table 2. Design Parameters

DESIGN PARAMETER	EXAMPLE VALUE
Load Impedance	4 Ω to 32 Ω
Supply Voltage	5 V to 12 V

9.2.2.2 Detailed Design Procedure

The Detailed Design Procedure can be found in the [Detailed Design Procedure](#) section.

9.2.2.3 Application Curve

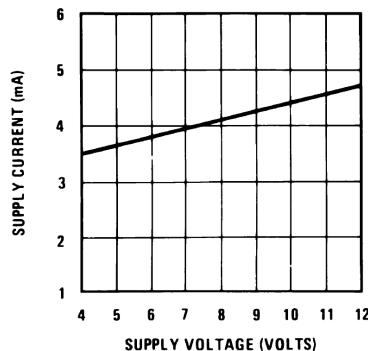
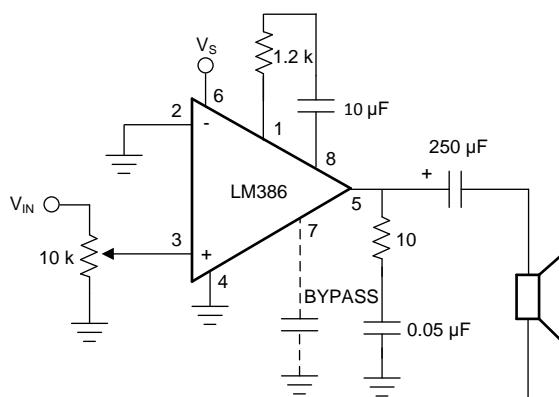


Figure 13. Supply Current vs Supply Voltage

9.2.3 LM386 with Gain = 50



Copyright © 2017, Texas Instruments Incorporated

Figure 14. LM386 with Gain = 50

9.2.3.1 Design Requirements

Table 3. Design Parameters

DESIGN PARAMETER	EXAMPLE VALUE
Load Impedance	4 Ω to 32 Ω
Supply Voltage	5 V to 12 V

9.2.3.2 Detailed Design Procedure

The Detailed Design Procedure can be found in the [Detailed Design Procedure](#) section.

9.2.3.3 Application Curve

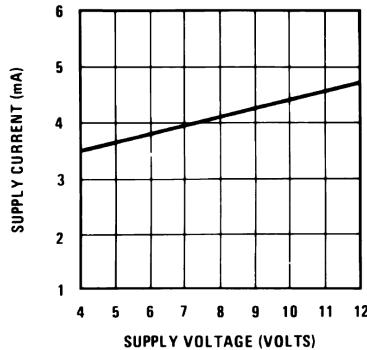
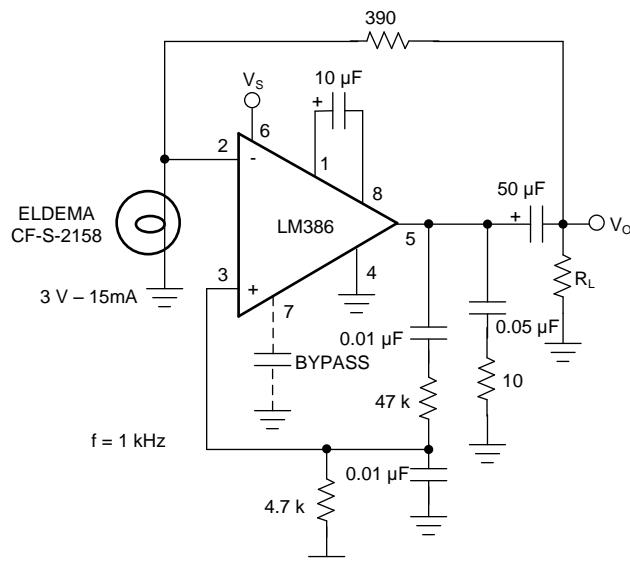


Figure 15. Supply Current vs Supply Voltage

9.2.4 Low Distortion Power Wienbridge Oscillator



Copyright © 2017, Texas Instruments Incorporated

Figure 16. Low Distortion Power Wienbridge Oscillator

9.2.4.1 Design Requirements

Table 4. Design Parameters

DESIGN PARAMETER	EXAMPLE VALUE
Load Impedance	4 Ω to 32 Ω
Supply Voltage	5 V to 12 V

9.2.4.2 Detailed Design Procedure

The Detailed Design Procedure can be found in the [Detailed Design Procedure](#) section.

9.2.4.3 Application Curve

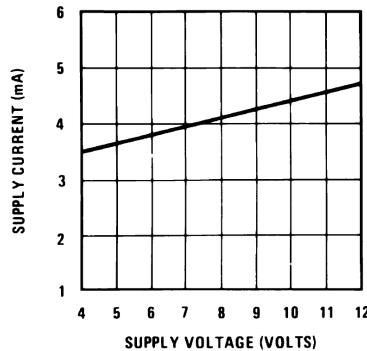
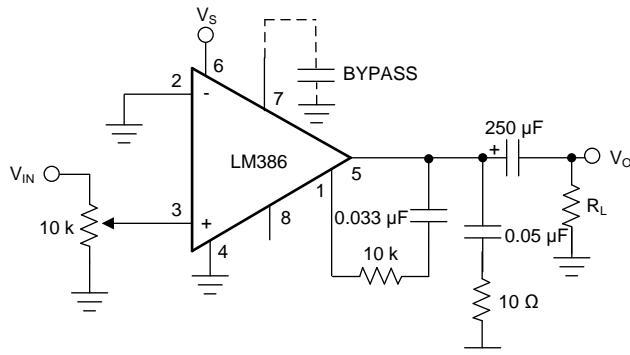


Figure 17. Supply Current vs Supply Voltage

9.2.5 LM386 with Bass Boost



Copyright © 2017, Texas Instruments Incorporated

Figure 18. LM386 with Bass Boost

9.2.5.1 Design Requirements

Table 5. Design Parameters

DESIGN PARAMETER	EXAMPLE VALUE
Load Impedance	4 Ω to 32 Ω
Supply Voltage	5 V to 12 V

9.2.5.2 Detailed Design Procedure

The Detailed Design Procedure can be found in the [Detailed Design Procedure](#) section.

9.2.5.3 Application Curve

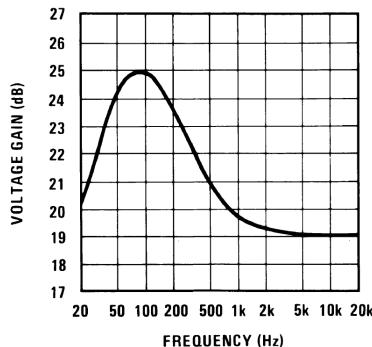


Figure 19. Voltage Gain vs Frequency

9.2.6 Square Wave Oscillator

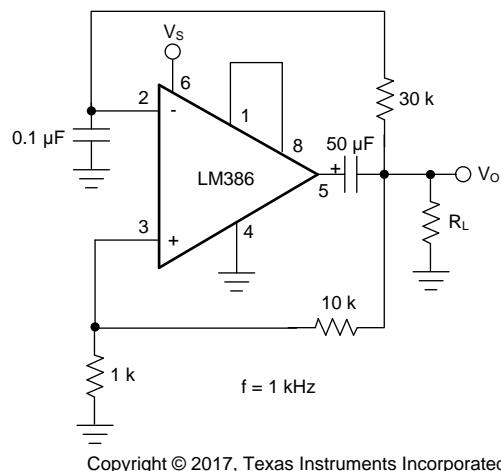


Figure 20. Square Wave Oscillator

Table 6. Design Parameters

DESIGN PARAMETER	EXAMPLE VALUE
Load Impedance	4 Ω to 32 Ω
Supply Voltage	5 V to 12 V

9.2.6.1 Detailed Design Procedure

The Detailed Design Procedure can be found in the [Detailed Design Procedure](#) section.

9.2.6.2 Application Curve

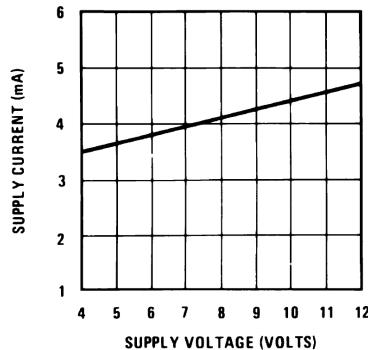
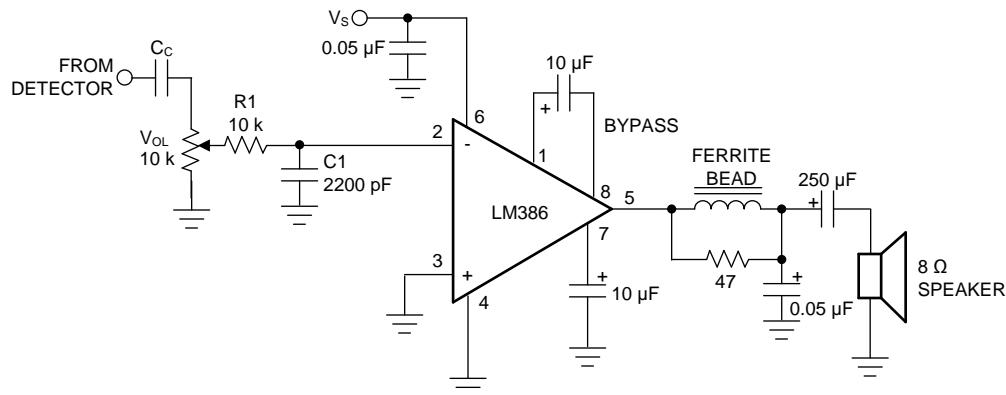


Figure 21. Supply Current vs Supply Voltage

9.2.7 AM Radio Power Amplifier



Copyright © 2017, Texas Instruments Incorporated

Figure 22. AM Radio Power Amplifier

9.2.7.1 Design Requirements

Table 7. Design Parameters

DESIGN PARAMETER	EXAMPLE VALUE
Load Impedance	4 Ω to 32 Ω
Supply Voltage	5 V to 12 V

9.2.7.2 Detailed Design Procedure

The Detailed Design Procedure can be found in the [Detailed Design Procedure](#) section.

9.2.7.3 Application Curve

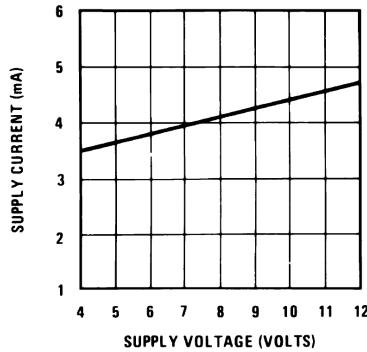


Figure 23. Supply Current vs Supply Voltage

10 Power Supply Recommendations

The LM386 is specified for operation up to 12 V or 18 V. The power supply should be well regulated and the voltage must be within the specified values. It is recommended to place a capacitor to GND close to the LM386 power supply pin.

11 Layout

11.1 Layout Guidelines

Place all required components as close as possible to the device. Use short traces for the output to the speaker connection. Route the analog traces far from the digital signal traces and avoid crossing them.

11.2 Layout Examples

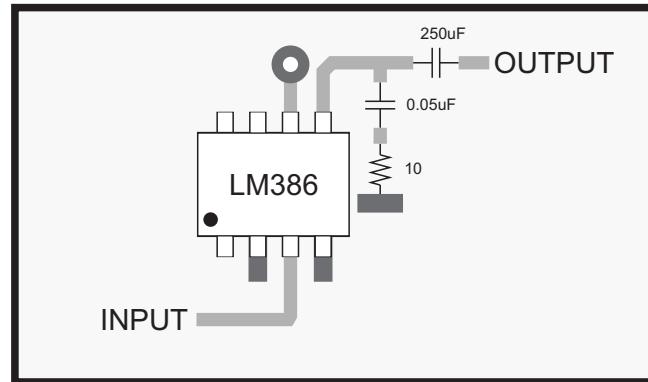


Figure 24. Layout Example for Minimum Parts Gain = 20 dB on PDIP package

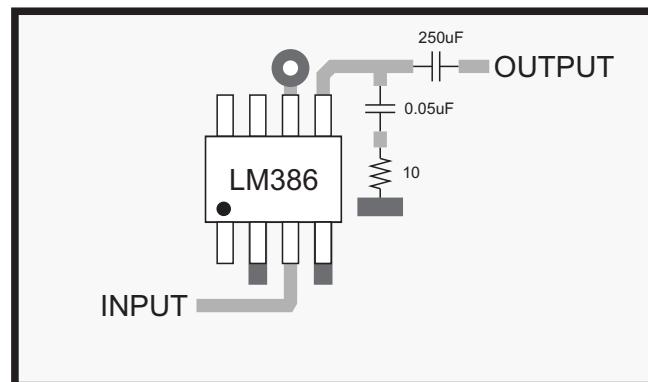
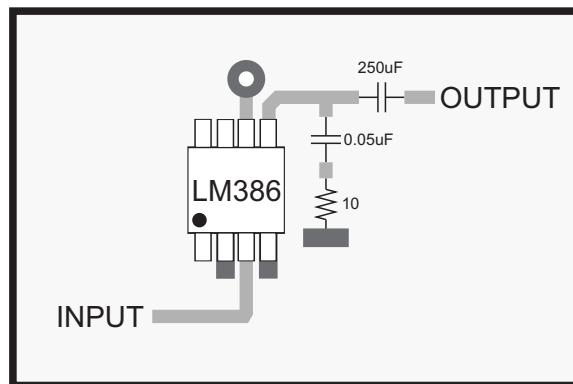


Figure 25. Layout Example for Minimum Parts Gain = 20 dB on SOIC package

Layout Examples (continued)



■ Connection to ground plane ● Connection to power 5V
— Top layer traces □ Top layer ground plane

Figure 26. Layout Example for Minimum Parts Gain = 20 dB on VSSOP package

12 Device and Documentation Support

12.1 Device Support

12.1.1 Development Support

12.2 Documentation Support

12.3 Related Links

The table below lists quick access links. Categories include technical documents, support and community resources, tools and software, and quick access to order now.

Table 8. Related Links

PARTS	PRODUCT FOLDER	ORDER NOW	TECHNICAL DOCUMENTS	TOOLS & SOFTWARE	SUPPORT & COMMUNITY
LM386M-1	Click here				
LM386MX-1	Click here				

12.4 Receiving Notification of Documentation Updates

To receive notification of documentation updates — go to the product folder for your device on ti.com. In the upper right-hand corner, click the *Alert me* button to register and receive a weekly digest of product information that has changed (if any). For change details, check the revision history of any revised document.

12.5 Community Resources

The following links connect to TI community resources. Linked contents are provided "AS IS" by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's [Terms of Use](#).

TI E2E™ Online Community **TI's Engineer-to-Engineer (E2E) Community.** Created to foster collaboration among engineers. At e2e.ti.com, you can ask questions, share knowledge, explore ideas and help solve problems with fellow engineers.

Design Support **TI's Design Support** Quickly find helpful E2E forums along with design support tools and contact information for technical support.

12.6 Trademarks

E2E is a trademark of Texas Instruments.

All other trademarks are the property of their respective owners.

12.7 Electrostatic Discharge Caution



These devices have limited built-in ESD protection. The leads should be shorted together or the device placed in conductive foam during storage or handling to prevent electrostatic damage to the MOS gates.

12.8 Glossary

SLYZ022 — TI Glossary.

This glossary lists and explains terms, acronyms, and definitions.

13 Mechanical, Packaging, and Orderable Information

The following pages include mechanical, packaging, and orderable information. This information is the most current data available for the designated devices. This data is subject to change without notice and revision of this document. For browser-based versions of this data sheet, refer to the left-hand navigation.

PACKAGING INFORMATION

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead finish/ Ball material (6)	MSL Peak Temp (3)	Op Temp (°C)	Device Marking (4/5)	Samples
LM386M-1/NOPB	ACTIVE	SOIC	D	8	95	RoHS & Green	SN	Level-1-260C-UNLIM	0 to 70	LM386 M-1	Samples
LM386MMX-1/NOPB	ACTIVE	VSSOP	DGK	8	3500	RoHS & Green	SN	Level-1-260C-UNLIM	0 to 70	Z86	Samples
LM386MX-1/NOPB	ACTIVE	SOIC	D	8	2500	RoHS & Green	SN	Level-1-260C-UNLIM	0 to 70	LM386 M-1	Samples
LM386N-1/NOPB	ACTIVE	PDIP	P	8	40	RoHS & Green	Call TI SN	Level-1-NA-UNLIM	0 to 70	LM 386N-1	Samples
LM386N-3/NOPB	ACTIVE	PDIP	P	8	40	RoHS & Green	SN	Level-1-NA-UNLIM	0 to 70	LM 386N-3	Samples
LM386N-4/NOPB	ACTIVE	PDIP	P	8	40	RoHS & Green	Call TI SN	Level-1-NA-UNLIM	0 to 70	LM 386N-4	Samples

(1) The marketing status values are defined as follows:

ACTIVE: Product device recommended for new designs.

LIFEBUY: TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.

NRND: Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.

PREVIEW: Device has been announced but is not in production. Samples may or may not be available.

OBSOLETE: TI has discontinued the production of the device.

(2) **RoHS:** TI defines "RoHS" to mean semiconductor products that are compliant with the current EU RoHS requirements for all 10 RoHS substances, including the requirement that RoHS substance do not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, "RoHS" products are suitable for use in specified lead-free processes. TI may reference these types of products as "Pb-Free".

RoHS Exempt: TI defines "RoHS Exempt" to mean products that contain lead but are compliant with EU RoHS pursuant to a specific EU RoHS exemption.

Green: TI defines "Green" to mean the content of Chlorine (Cl) and Bromine (Br) based flame retardants meet JS709B low halogen requirements of <=1000ppm threshold. Antimony trioxide based flame retardants must also meet the <=1000ppm threshold requirement.

(3) MSL, Peak Temp. - The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

(4) There may be additional marking, which relates to the logo, the lot trace code information, or the environmental category on the device.

(5) Multiple Device Markings will be inside parentheses. Only one Device Marking contained in parentheses and separated by a "~" will appear on a device. If a line is indented then it is a continuation of the previous line and the two combined represent the entire Device Marking for that device.



www.ti.com

PACKAGE OPTION ADDENDUM

10-Dec-2020

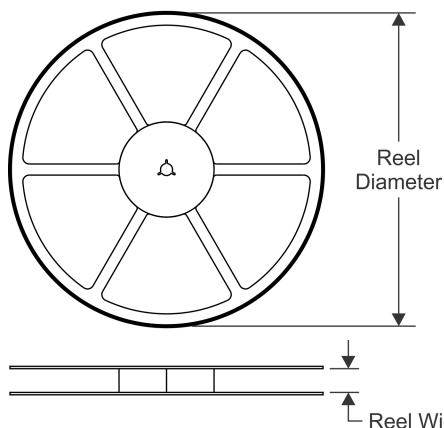
- (6) Lead finish/Ball material - Orderable Devices may have multiple material finish options. Finish options are separated by a vertical ruled line. Lead finish/Ball material values may wrap to two lines if the finish value exceeds the maximum column width.

Important Information and Disclaimer: The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

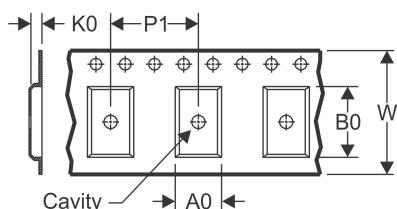
In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

TAPE AND REEL INFORMATION

REEL DIMENSIONS

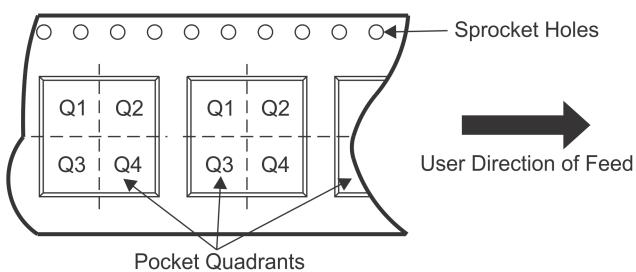


TAPE DIMENSIONS



A0	Dimension designed to accommodate the component width
B0	Dimension designed to accommodate the component length
K0	Dimension designed to accommodate the component thickness
W	Overall width of the carrier tape
P1	Pitch between successive cavity centers

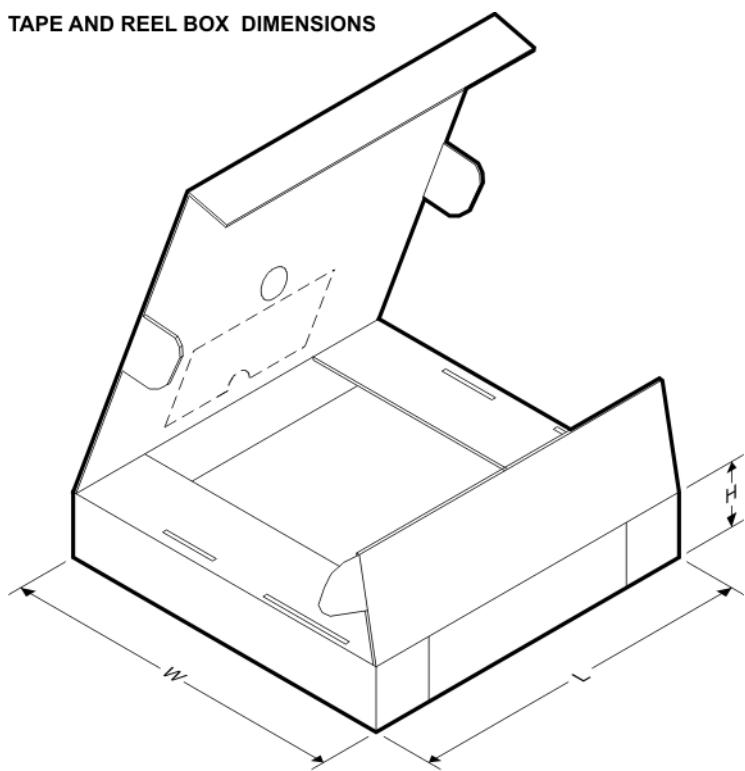
QUADRANT ASSIGNMENTS FOR PIN 1 ORIENTATION IN TAPE



*All dimensions are nominal

Device	Package Type	Package Drawing	Pins	SPQ	Reel Diameter (mm)	Reel Width W1 (mm)	A0 (mm)	B0 (mm)	K0 (mm)	P1 (mm)	W (mm)	Pin1 Quadrant
LM386MMX-1/NOPB	VSSOP	DGK	8	3500	330.0	12.4	5.3	3.4	1.4	8.0	12.0	Q1
LM386MX-1/NOPB	SOIC	D	8	2500	330.0	12.4	6.5	5.4	2.0	8.0	12.0	Q1

TAPE AND REEL BOX DIMENSIONS



*All dimensions are nominal

Device	Package Type	Package Drawing	Pins	SPQ	Length (mm)	Width (mm)	Height (mm)
LM386MMX-1/NOPB	VSSOP	DGK	8	3500	367.0	367.0	35.0
LM386MX-1/NOPB	SOIC	D	8	2500	367.0	367.0	35.0

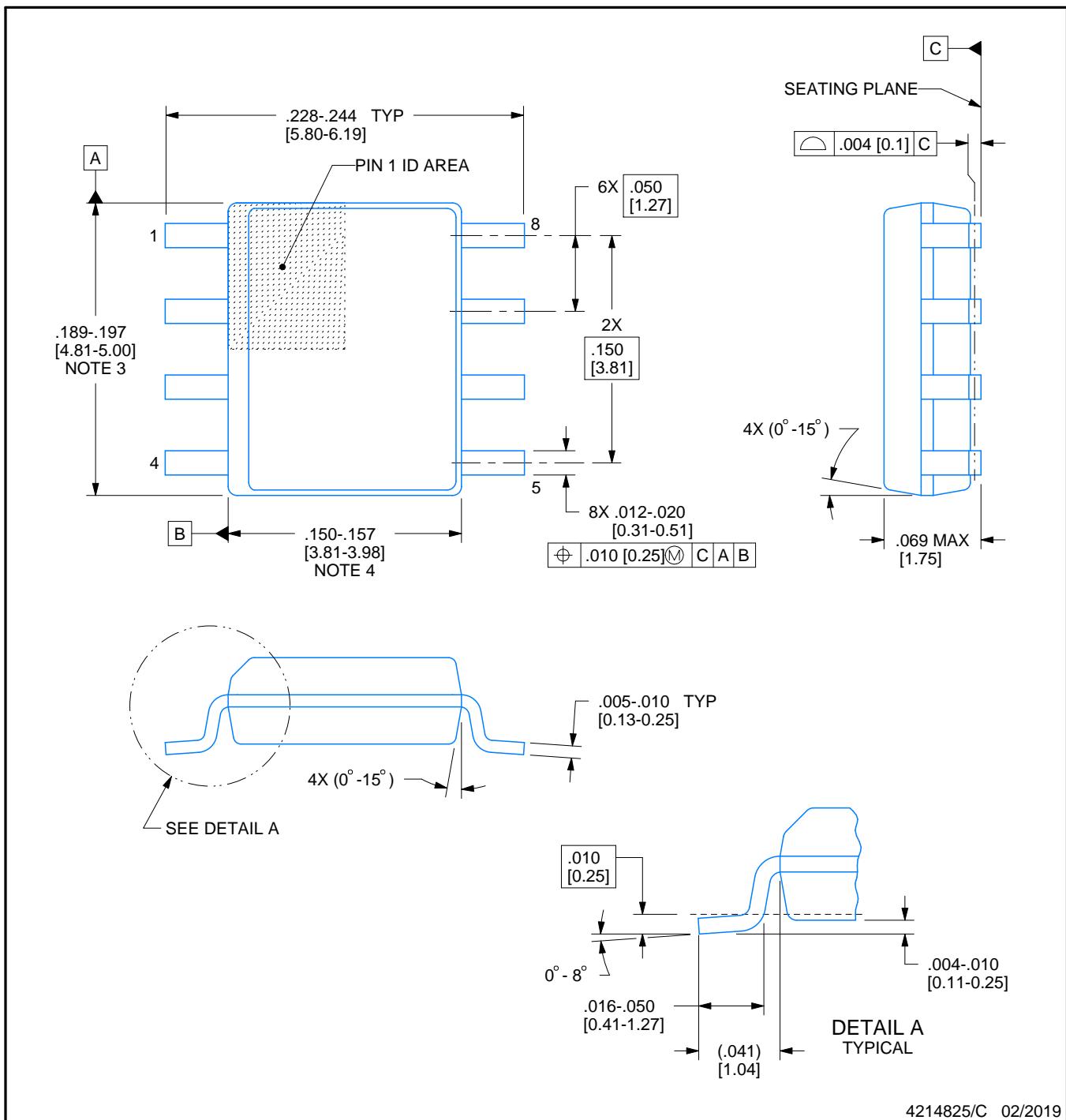
D0008A



PACKAGE OUTLINE

SOIC - 1.75 mm max height

SMALL OUTLINE INTEGRATED CIRCUIT



NOTES:

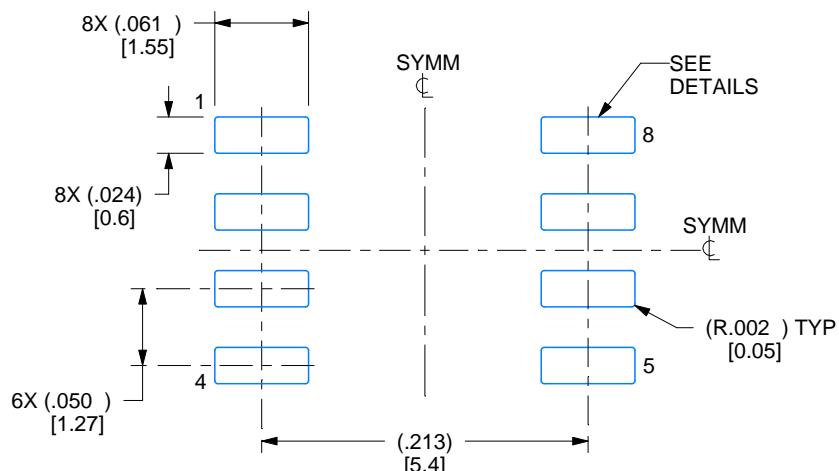
1. Linear dimensions are in inches [millimeters]. Dimensions in parenthesis are for reference only. Controlling dimensions are in inches. Dimensioning and tolerancing per ASME Y14.5M.
2. This drawing is subject to change without notice.
3. This dimension does not include mold flash, protrusions, or gate burrs. Mold flash, protrusions, or gate burrs shall not exceed .006 [0.15] per side.
4. This dimension does not include interlead flash.
5. Reference JEDEC registration MS-012, variation AA.

EXAMPLE BOARD LAYOUT

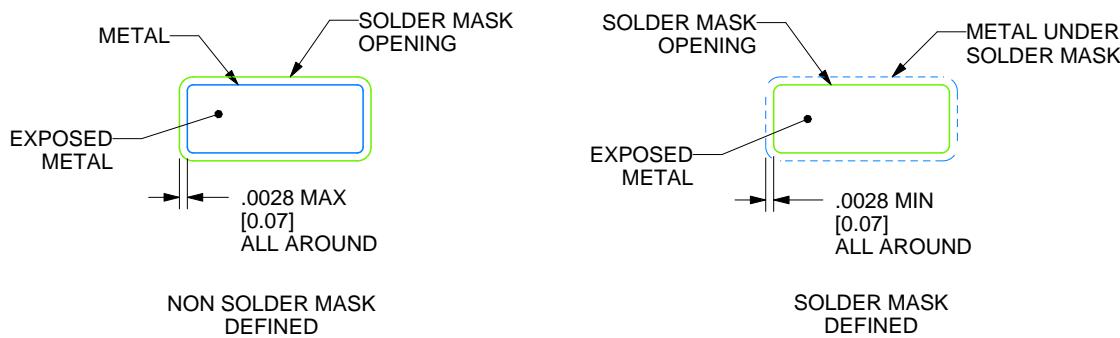
D0008A

SOIC - 1.75 mm max height

SMALL OUTLINE INTEGRATED CIRCUIT



LAND PATTERN EXAMPLE
EXPOSED METAL SHOWN
SCALE:8X



SOLDER MASK DETAILS

4214825/C 02/2019

NOTES: (continued)

6. Publication IPC-7351 may have alternate designs.

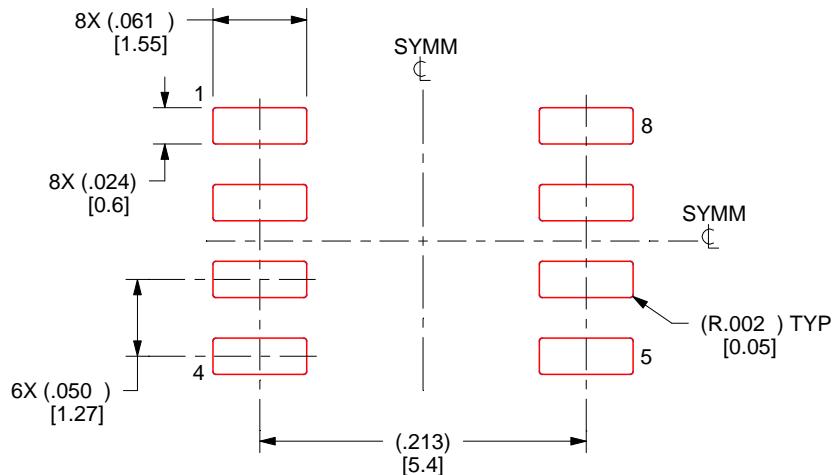
7. Solder mask tolerances between and around signal pads can vary based on board fabrication site.

EXAMPLE STENCIL DESIGN

D0008A

SOIC - 1.75 mm max height

SMALL OUTLINE INTEGRATED CIRCUIT



SOLDER PASTE EXAMPLE
BASED ON .005 INCH [0.125 MM] THICK STENCIL
SCALE:8X

4214825/C 02/2019

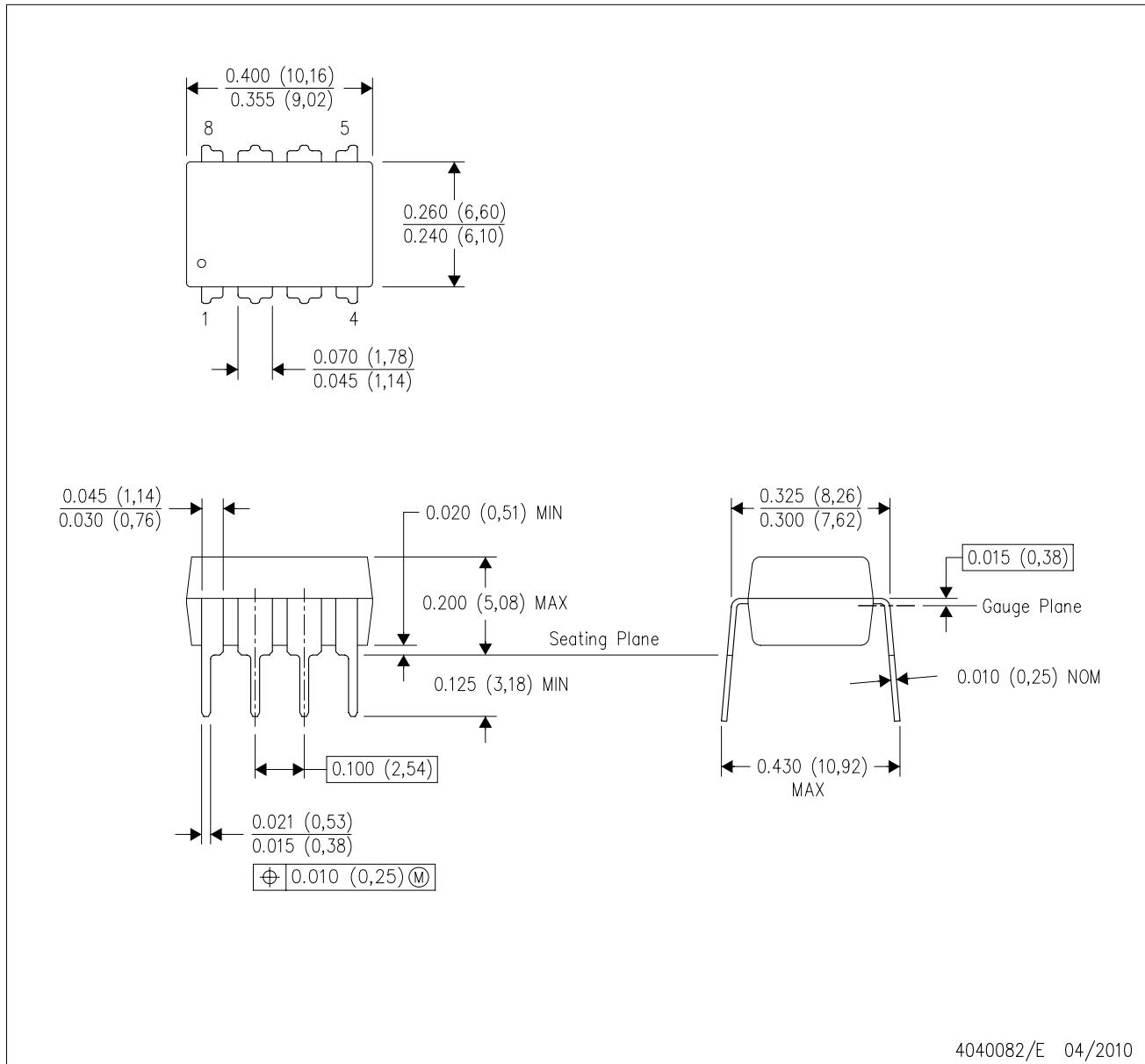
NOTES: (continued)

8. Laser cutting apertures with trapezoidal walls and rounded corners may offer better paste release. IPC-7525 may have alternate design recommendations.
9. Board assembly site may have different recommendations for stencil design.

MECHANICAL DATA

P (R-PDIP-T8)

PLASTIC DUAL-IN-LINE PACKAGE

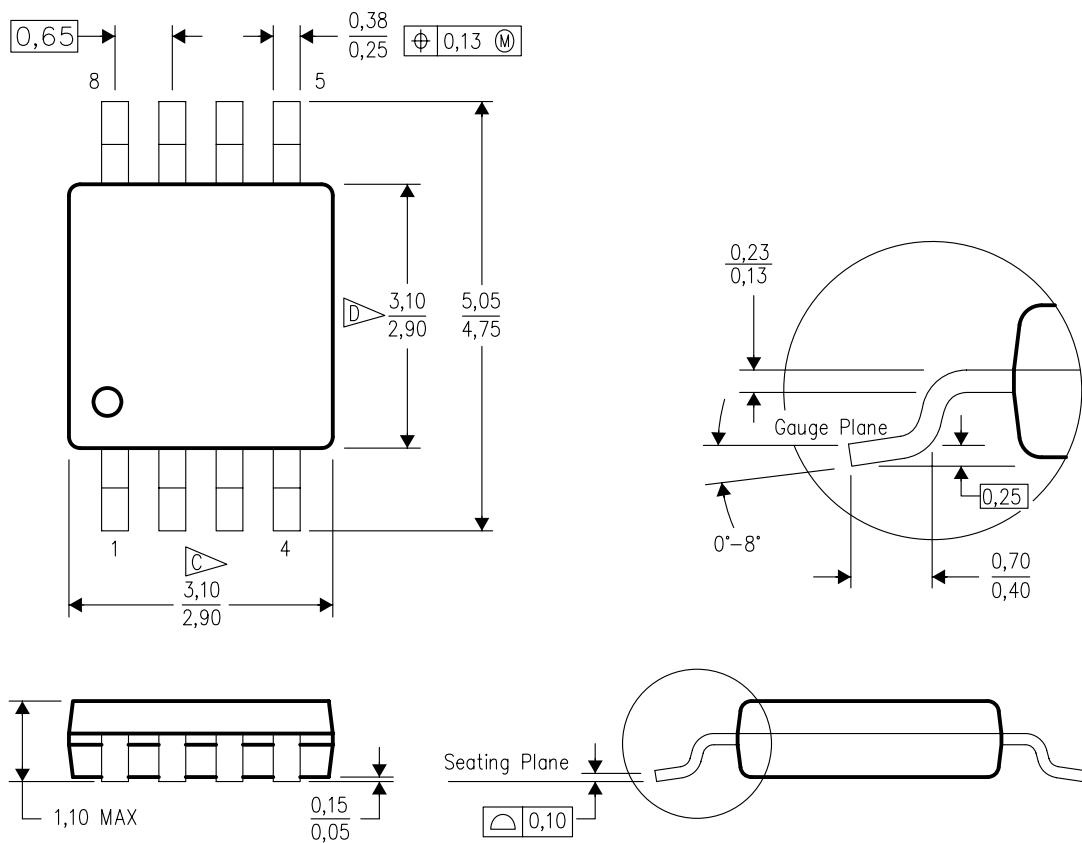


4040082/E 04/2010

- NOTES:
- All linear dimensions are in inches (millimeters).
 - This drawing is subject to change without notice.
 - Falls within JEDEC MS-001 variation BA.

DGK (S-PDSO-G8)

PLASTIC SMALL-OUTLINE PACKAGE



4073329/E 05/06

NOTES: A. All linear dimensions are in millimeters.

B. This drawing is subject to change without notice.

Body length does not include mold flash, protrusions, or gate burrs. Mold flash, protrusions, or gate burrs shall not exceed 0.15 per end.

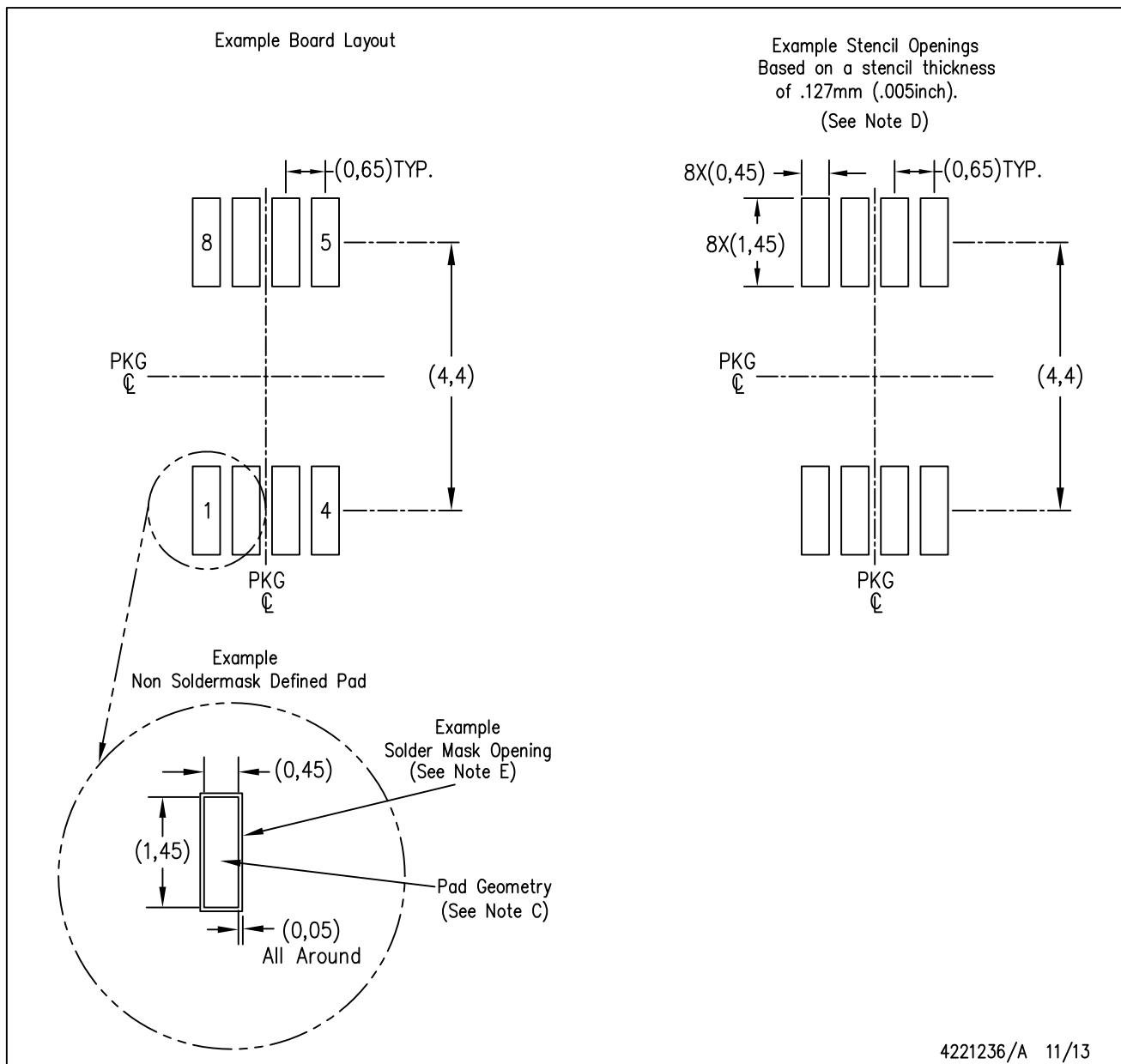
Body width does not include interlead flash. Interlead flash shall not exceed 0.50 per side.

E. Falls within JEDEC MO-187 variation AA, except interlead flash.

LAND PATTERN DATA

DGK (S-PDSO-G8)

PLASTIC SMALL OUTLINE PACKAGE



- NOTES:
- A. All linear dimensions are in millimeters.
 - B. This drawing is subject to change without notice.
 - C. Publication IPC-7351 is recommended for alternate designs.
 - D. Laser cutting apertures with trapezoidal walls and also rounding corners will offer better paste release. Customers should contact their board assembly site for stencil design recommendations. Refer to IPC-7525 for other stencil recommendations.
 - E. Customers should contact their board fabrication site for solder mask tolerances between and around signal pads.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

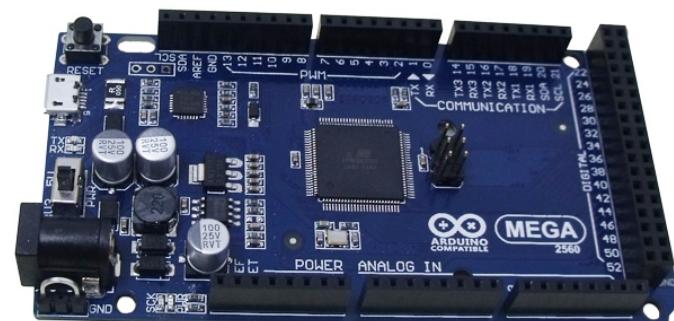
Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2020, Texas Instruments Incorporated

MEGA 2560 R3 Datasheet



■ INTRODUCTION

MEGA 2560 R3 is an Arduino compatible board, It's based on Arduino MEGA 2560 design, So you can use BUONO MEGA 2560 as Arduino MEGA 2560, All Code, shield and IDE for Arduino MEGA 2560 R3 are also valid on BUONO MEGA 2560, Some visible improvement on hardware make BUONO more flexible and easier use, For example: 3.3V or 5V IO selectable allow you connect some 3.3V modules (such as XBee) to BUONO directly.



■ THE DIFFERENCE BETWEEN Arduino MEGA 2560 AND BUONO MEGA 2560

	Arduino MEGA 2560	BUONO MEGA 2560
USB Connector	Type B Female	Micro USB connector
Operating Voltage	5V	3.3V or 5V selectable
3.3V Current	50mA	600mA
5V Current	500mA	1.8A
Input Range	7V to 12V	6V to 23V
MCU	ATMEGA2560	ATMEGA2560
Others		Ground terminal LPF for AVCC

■ FEATURES

- Inherits all of Arduino MEGA 2560's features
- Compatible to Arduino MEGA 2560's pin layout, screw hole and dimensions
- 3.3V or 5V operating voltage selectable
- More visible location of indication LEDs
- Evolved with SMD components
- Digital I/O pins 54(of which 15 provided PWM output)
- Analog Input pins 16
- Flash memory 256 KB of which 8KB used by bootloader
- SRAM 8 KB
- EEPROM 4KB
- Clock speed 16MHz
- Micro USB connection make the cable can be mixed use with most digital products such as cell phone
- 5V DC/DC provide 95% efficiency
- 6V to 23V wide range input
- Provide max 1.8A for peripheral circuits
- Ground terminal make measurement easier

■ PACKAGE LIST

- 1xBUONO MEGA 2560
- 1xMicro USB cable
- 4xScrew and nut

■ CONTACT US

540 Mill River LN San Jose,
CA95134

E-mail:support@inhaos.com

Web:www.inhaos.com

HD44780U (LCD-II)

(Dot Matrix Liquid Crystal Display Controller/Driver)

HITACHI

ADE-207-272(Z)

'99.9

Rev. 0.0

Description

The HD44780U dot-matrix liquid crystal display controller and driver LSI displays alphanumerics, Japanese kana characters, and symbols. It can be configured to drive a dot-matrix liquid crystal display under the control of a 4- or 8-bit microprocessor. Since all the functions such as display RAM, character generator, and liquid crystal driver, required for driving a dot-matrix liquid crystal display are internally provided on one chip, a minimal system can be interfaced with this controller/driver.

A single HD44780U can display up to one 8-character line or two 8-character lines.

The HD44780U has pin function compatibility with the HD44780S which allows the user to easily replace an LCD-II with an HD44780U. The HD44780U character generator ROM is extended to generate 208 5 × 8 dot character fonts and 32 5 × 10 dot character fonts for a total of 240 different character fonts.

The low power supply (2.7V to 5.5V) of the HD44780U is suitable for any portable battery-driven product requiring low power dissipation.

Features

- 5 × 8 and 5 × 10 dot matrix possible
- Low power operation support:
 - 2.7 to 5.5V
- Wide range of liquid crystal display driver power
 - 3.0 to 11V
- Liquid crystal drive waveform
 - A (One line frequency AC waveform)
- Correspond to high speed MPU bus interface
 - 2 MHz (when $V_{CC} = 5V$)
- 4-bit or 8-bit MPU interface enabled
- 80 × 8-bit display RAM (80 characters max.)
- 9,920-bit character generator ROM for a total of 240 character fonts
 - 208 character fonts (5 × 8 dot)
 - 32 character fonts (5 × 10 dot)

HITACHI

HD44780U

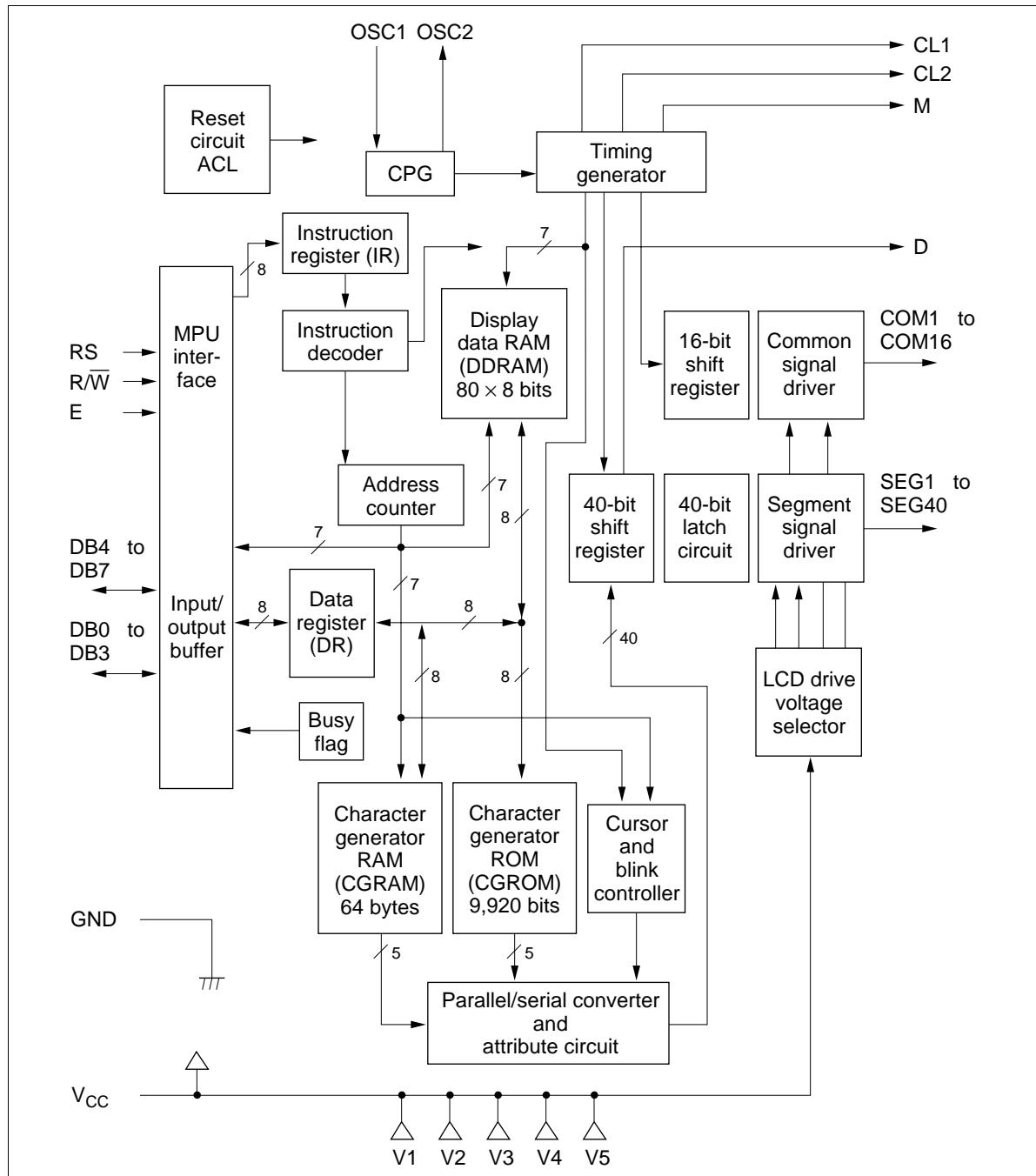
- 64 × 8-bit character generator RAM
 - 8 character fonts (5 × 8 dot)
 - 4 character fonts (5 × 10 dot)
- 16-common × 40-segment liquid crystal display driver
- Programmable duty cycles
 - 1/8 for one line of 5 × 8 dots with cursor
 - 1/11 for one line of 5 × 10 dots with cursor
 - 1/16 for two lines of 5 × 8 dots with cursor
- Wide range of instruction functions:
 - Display clear, cursor home, display on/off, cursor on/off, display character blink, cursor shift, display shift
- Pin function compatibility with HD44780S
- Automatic reset circuit that initializes the controller/driver after power on
- Internal oscillator with external resistors
- Low power consumption

Ordering Information

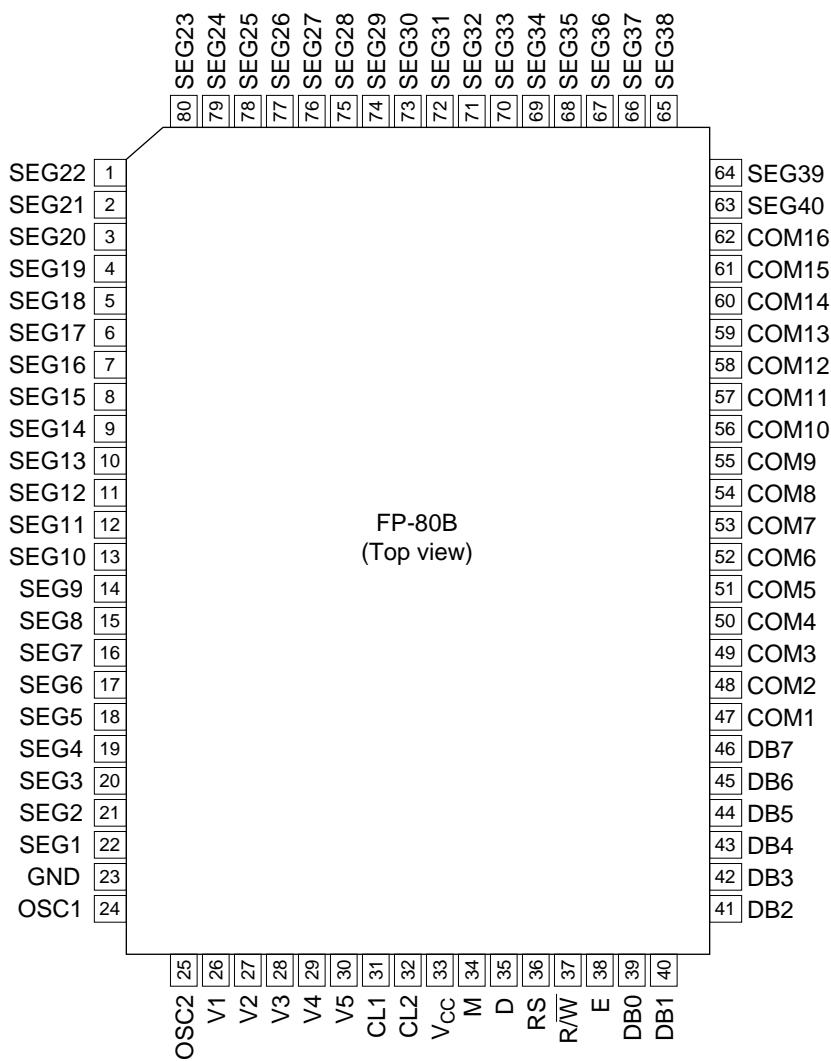
Type No.	Package	CGROM
HD44780UA00FS	FP-80B	Japanese standard font
HCD44780UA00	Chip	
HD44780UA00TF	TFP-80F	
HD44780UA02FS	FP-80B	European standard font
HCD44780UA02	Chip	
HD44780UA02TF	TFP-80F	
HD44780UBxxFS	FP-80B	Custom font
HCD44780UBxx	Chip	
HD44780UBxxTF	TFP-80F	

Note: xx: ROM code No.

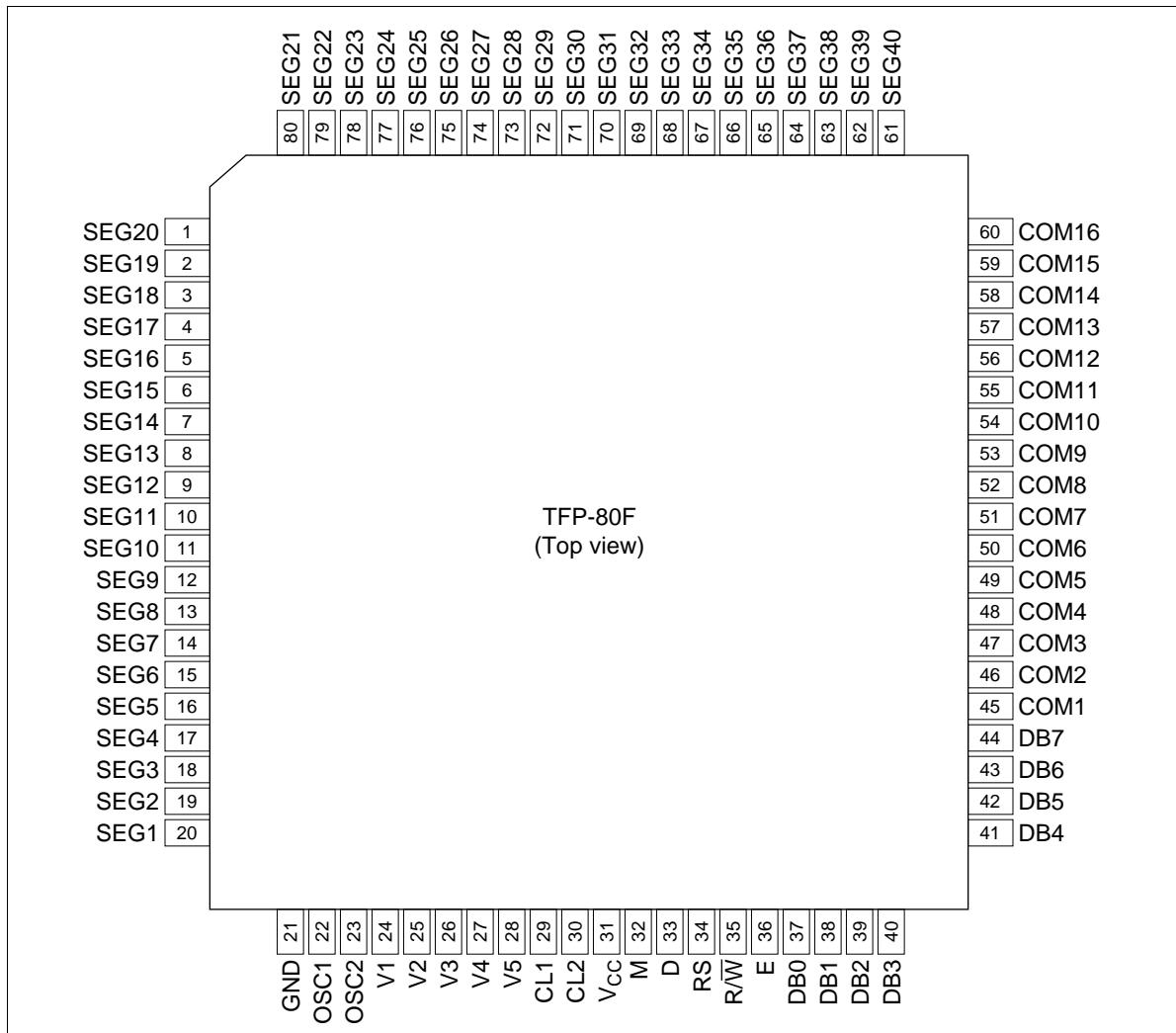
HD44780U Block Diagram



HD44780U Pin Arrangement (FP-80B)



HD44780U Pin Arrangement (TFP-80F)



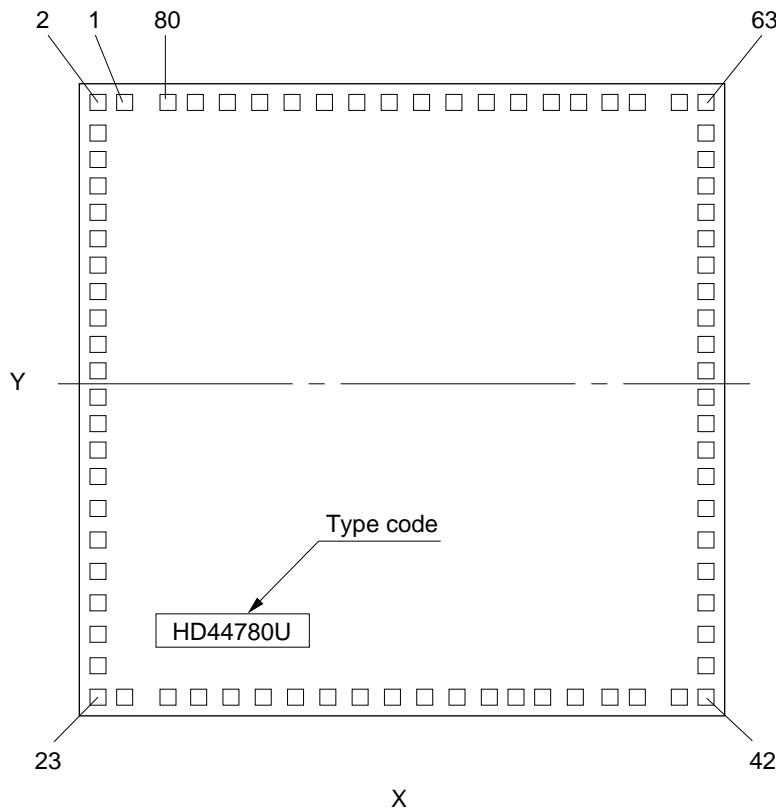
HD44780U Pad Arrangement

Chip size: $4.90 \times 4.90 \text{ mm}^2$

Coordinate: Pad center (μm)

Origin: Chip center

Pad size: $114 \times 114 \mu\text{m}^2$



HCD44780U Pad Location Coordinates

Pad No.	Function	Coordinate		Pad No.	Function	Coordinate	
		X (um)	Y (um)			X (um)	Y (um)
1	SEG22	-2100	2313	41	DB2	2070	-2290
2	SEG21	-2280	2313	42	DB3	2260	-2290
3	SEG20	-2313	2089	43	DB4	2290	-2099
4	SEG19	-2313	1833	44	DB5	2290	-1883
5	SEG18	-2313	1617	45	DB6	2290	-1667
6	SEG17	-2313	1401	46	DB7	2290	-1452
7	SEG16	-2313	1186	47	COM1	2313	-1186
8	SEG15	-2313	970	48	COM2	2313	-970
9	SEG14	-2313	755	49	COM3	2313	-755
10	SEG13	-2313	539	50	COM4	2313	-539
11	SEG12	-2313	323	51	COM5	2313	-323
12	SEG11	-2313	108	52	COM6	2313	-108
13	SEG10	-2313	-108	53	COM7	2313	108
14	SEG9	-2313	-323	54	COM8	2313	323
15	SEG8	-2313	-539	55	COM9	2313	539
16	SEG7	-2313	-755	56	COM10	2313	755
17	SEG6	-2313	-970	57	COM11	2313	970
18	SEG5	-2313	-1186	58	COM12	2313	1186
19	SEG4	-2313	-1401	59	COM13	2313	1401
20	SEG3	-2313	-1617	60	COM14	2313	1617
21	SEG2	-2313	-1833	61	COM15	2313	1833
22	SEG1	-2313	-2073	62	COM16	2313	2095
23	GND	-2280	-2290	63	SEG40	2296	2313
24	OSC1	-2080	-2290	64	SEG39	2100	2313
25	OSC2	-1749	-2290	65	SEG38	1617	2313
26	V1	-1550	-2290	66	SEG37	1401	2313
27	V2	-1268	-2290	67	SEG36	1186	2313
28	V3	-941	-2290	68	SEG35	970	2313
29	V4	-623	-2290	69	SEG34	755	2313
30	V5	-304	-2290	70	SEG33	539	2313
31	CL1	-48	-2290	71	SEG32	323	2313
32	CL2	142	-2290	72	SEG31	108	2313
33	V _{cc}	309	-2290	73	SEG30	-108	2313
34	M	475	-2290	74	SEG29	-323	2313
35	D	665	-2290	75	SEG28	-539	2313
36	RS	832	-2290	76	SEG27	-755	2313
37	R/W	1022	-2290	77	SEG26	-970	2313
38	E	1204	-2290	78	SEG25	-1186	2313
39	DB0	1454	-2290	79	SEG24	-1401	2313
40	DB1	1684	-2290	80	SEG23	-1617	2313

Pin Functions

Signal	No. of Lines	I/O	Device Interfaced with	Function
RS	1	I	MPU	Selects registers. 0: Instruction register (for write) Busy flag: address counter (for read) 1: Data register (for write and read)
R/W	1	I	MPU	Selects read or write. 0: Write 1: Read
E	1	I	MPU	Starts data read/write.
DB4 to DB7	4	I/O	MPU	Four high order bidirectional tristate data bus pins. Used for data transfer and receive between the MPU and the HD44780U. DB7 can be used as a busy flag.
DB0 to DB3	4	I/O	MPU	Four low order bidirectional tristate data bus pins. Used for data transfer and receive between the MPU and the HD44780U. These pins are not used during 4-bit operation.
CL1	1	O	Extension driver	Clock to latch serial data D sent to the extension driver
CL2	1	O	Extension driver	Clock to shift serial data D
M	1	O	Extension driver	Switch signal for converting the liquid crystal drive waveform to AC
D	1	O	Extension driver	Character pattern data corresponding to each segment signal
COM1 to COM16	16	O	LCD	Common signals that are not used are changed to non-selection waveforms. COM9 to COM16 are non-selection waveforms at 1/8 duty factor and COM12 to COM16 are non-selection waveforms at 1/11 duty factor.
SEG1 to SEG40	40	O	LCD	Segment signals
V1 to V5	5	—	Power supply	Power supply for LCD drive $V_{cc} - V5 = 11\text{ V (max)}$
V_{cc} , GND	2	—	Power supply	V_{cc} : 2.7V to 5.5V, GND: 0V
OSC1, OSC2	2	—	Oscillation resistor clock	When crystal oscillation is performed, a resistor must be connected externally. When the pin input is an external clock, it must be input to OSC1.

Function Description

Registers

The HD44780U 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 RAM (CGRAM). The IR can only be written from the MPU.

The DR temporarily stores data to be written into DDRAM or CGRAM and temporarily stores data to be read from DDRAM or CGRAM. Data written into the DR from the MPU is automatically written into DDRAM or CGRAM by an internal operation. The DR is also used for data storage when reading data from DDRAM or CGRAM. When address information is written into the IR, data is read and then stored into the DR from DDRAM or CGRAM by an internal operation. Data transfer between the MPU is then completed when the MPU reads the DR. After the read, data in DDRAM or CGRAM at the next address is sent to the DR for the next read from the MPU. By the register selector (RS) signal, these two registers can be selected (Table 1).

Busy Flag (BF)

When the busy flag is 1, the HD44780U is in the internal operation mode, and the next instruction will not be accepted. When RS = 0 and R/W = 1 (Table 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. When an address of an instruction is written into the IR, the address information is sent from the IR to the AC. Selection of either DDRAM or CGRAM is also determined concurrently by the instruction.

After writing into (reading from) DDRAM or CGRAM, the AC is automatically incremented by 1 (decremented by 1). The AC contents are then output to DB0 to DB6 when RS = 0 and R/W = 1 (Table 1).

Table 1 Register Selection

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 DB6)
1	0	DR write as an internal operation (DR to DDRAM or CGRAM)
1	1	DR read as an internal operation (DDRAM or CGRAM to DR)

Display Data RAM (DDRAM)

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80×8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. See Figure 1 for the relationships between DDRAM addresses and positions on the liquid crystal display.

The DDRAM address (A_{DD}) is set in the address counter (AC) as hexadecimal.

- 1-line display ($N = 0$) (Figure 2)
 - When there are fewer than 80 display characters, the display begins at the head position. For example, if using only the HD44780, 8 characters are displayed. See Figure 3.
When the display shift operation is performed, the DDRAM address shifts. See Figure 3.

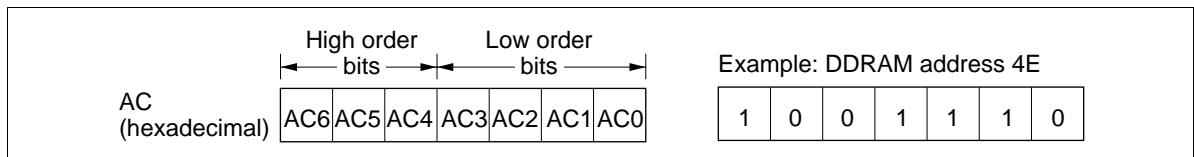


Figure 1 DDRAM Address

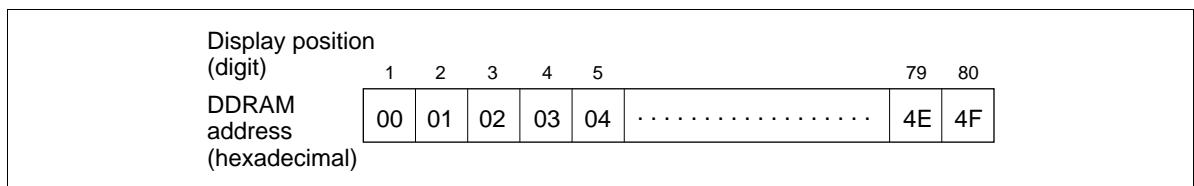


Figure 2 1-Line Display

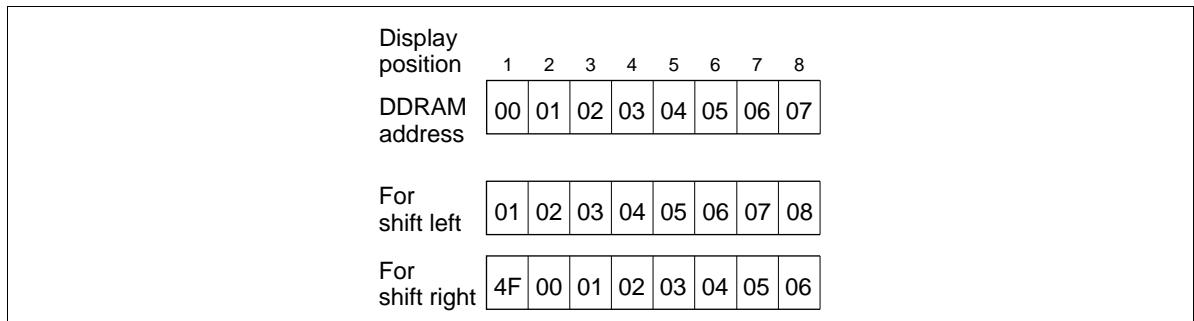


Figure 3 1-Line by 8-Character Display Example

- 2-line display ($N = 1$) (Figure 4)

— Case 1: When the number of display characters is less than 40×2 lines, the two lines are displayed from the head. Note that the first line end address and the second line start address are not consecutive. For example, when just the HD44780 is used, 8 characters \times 2 lines are displayed. See Figure 5.

When display shift operation is performed, the DDRAM address shifts. See Figure 5.

Display position	1	2	3	4	5	39	40	
DDRAM address (hexadecimal)	00	01	02	03	04	26	27
	40	41	42	43	44	66	67

Figure 4 2-Line Display

Display position	1	2	3	4	5	6	7	8
DDRAM address	00	01	02	03	04	05	06	07
	40	41	42	43	44	45	46	47

For shift left	01	02	03	04	05	06	07	08
	41	42	43	44	45	46	47	48

For shift right	27	00	01	02	03	04	05	06
	67	40	41	42	43	44	45	46

Figure 5 2-Line by 8-Character Display Example

- Case 2: For a 16-character × 2-line display, the HD44780 can be extended using one 40-output extension driver. See Figure 6.

When display shift operation is performed, the DDRAM address shifts. See Figure 6.

Display position	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
	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
HD44780U display										Extension driver display						
For shift left	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10
	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50
For shift right	27	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E
	67	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E

Figure 6 2-Line by 16-Character Display Example

Character Generator ROM (CGROM)

The character generator ROM generates 5×8 dot or 5×10 dot character patterns from 8-bit character codes (Table 4). It can generate 208 5×8 dot character patterns and 32 5×10 dot character patterns. User-defined character patterns are also available by mask-programmed ROM.

Character Generator RAM (CGRAM)

In the character generator RAM, the user can rewrite character patterns 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 codes at the addresses shown as the left column of Table 4 to show the character patterns stored in CGRAM.

See Table 5 for the relationship between CGRAM addresses and data and display patterns.

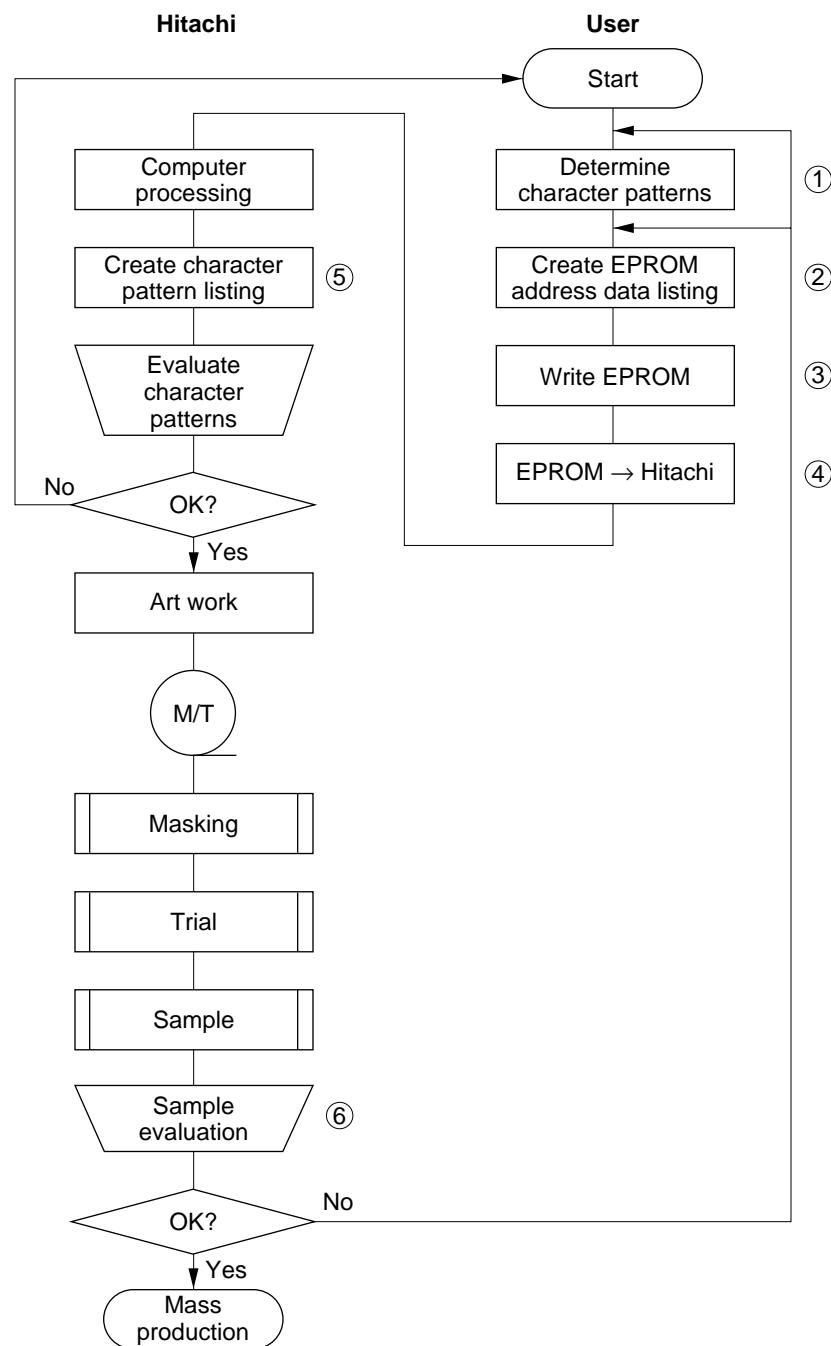
Areas that are not used for display can be used as general data RAM.

Modifying Character Patterns

- Character pattern development procedure

The following operations correspond to the numbers listed in Figure 7:

1. Determine the correspondence between character codes and character patterns.
2. Create a listing indicating the correspondence between EPROM addresses and data.
3. Program the character patterns into the EPROM.
4. Send the EPROM to Hitachi.
5. Computer processing on the EPROM is performed at Hitachi to create a character pattern listing, which is sent to the user.
6. If there are no problems within the character pattern listing, a trial LSI is created at Hitachi and samples are sent to the user for evaluation. When it is confirmed by the user that the character patterns are correctly written, mass production of the LSI proceeds at Hitachi.



Note: For a description of the numbers used in this figure, refer to the preceding page.

Figure 7 Character Pattern Development Procedure

- Programming character patterns

This section explains the correspondence between addresses and data used to program character patterns in EPROM. The HD44780U character generator ROM can generate 208 5×8 dot character patterns and 32 5×10 dot character patterns for a total of 240 different character patterns.

- Character patterns

EPROM address data and character pattern data correspond with each other to form a 5×8 or 5×10 dot character pattern (Tables 2 and 3).

Table 2 Example of Correspondence between EPROM Address Data and Character Pattern (5×8 Dots)

EPROM Address											Data					
A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	O4	O3	O2	O1	O0
												1	0	0	0	0
												1	0	0	0	0
												1	0	1	1	0
												1	1	0	0	1
												1	0	0	0	1
												1	0	0	0	1
												1	1	1	1	0
0	1	1	0	0	0	1	0	0	1	1	1	0	0	0	0	0
Character code											Line position					
											0	0	0	0	0	0
											0	0	0	0	0	0
											0	0	0	0	0	0
											0	0	0	0	0	0
											0	0	0	0	0	0
											0	0	0	0	0	0
											0	0	0	0	0	0
											0	0	0	0	0	0

← Cursor position

Notes:

1. EPROM addresses A11 to A4 correspond to a character code.

2. EPROM addresses A3 to A0 specify a line position of the character pattern.

3. EPROM data O4 to O0 correspond to character pattern data.

4. EPROM data O5 to O7 must be specified as 0.

5. A lit display position (black) corresponds to a 1.

6. Line 9 and the following lines must be blanked with 0s for a 5×8 dot character fonts.

— Handling unused character patterns

1. EPROM data outside the character pattern area: Always input 0s.
2. EPROM data in CGRAM area: Always input 0s. (Input 0s to EPROM addresses 00H to FFH.)
3. EPROM data used when the user does not use any HD44780U character pattern: According to the user application, handled in one of the two ways listed as follows.
 - a. When unused character patterns are not programmed: If an unused character code is written into DDRAM, all its dots are lit. By not programming a character pattern, all of its bits become lit. (This is due to the EPROM being filled with 1s after it is erased.)
 - b. When unused character patterns are programmed as 0s: Nothing is displayed even if unused character codes are written into DDRAM. (This is equivalent to a space.)

Table 3 Example of Correspondence between EPROM Address Data and Character Pattern (5 × 10 Dots)

EPROM Address										Data				
A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0										LSB O4 O3 O2 O1 O0				
										0	0	0	0	0
										0	0	0	0	0
										0	1	1	0	1
										1	0	0	1	1
										1	0	0	0	1
										1	0	0	0	1
										0	1	1	1	1
0	1	0	1	0	0	1	0	0	1	1	0	0	0	1
										1	0	0	0	1
										1	0	0	1	1
										1	0	1	0	0
										1	0	1	1	1
										1	1	0	0	0
										1	1	0	1	0
										1	1	1	0	0
										1	1	1	1	1
Character code										Line position				
Line position										Cursor position				

- Notes:
1. EPROM addresses A11 to A3 correspond to a character code.
 2. EPROM addresses A3 to A0 specify a line position of the character pattern.
 3. EPROM data O4 to O0 correspond to character pattern data.
 4. EPROM data O5 to O7 must be specified as 0.
 5. A lit display position (black) corresponds to a 1.
 6. Line 11 and the following lines must be blanked with 0s for a 5 × 10 dot character fonts.

Table 4 Correspondence between Character Codes and Character Patterns (ROM Code: A00)

Lower 4 Bits	Upper 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
	CG RAM (1)			Ø	ø	P	^	P				-	ø	ø	ø	ø	p
xxxx0000																	
xxxx0001	(2)		!	1	I	H	Q	a	q			ø	ø	ø	ø	ø	q
xxxx0010	(3)		"	2	B	R	b	r			ø	ø	ø	ø	ø	ø	ø
xxxx0011	(4)		#	3	C	S	c	s			ø	ø	ø	ø	ø	ø	ø
xxxx0100	(5)		\$	4	D	T	d	t			ø	ø	ø	ø	ø	ø	ø
xxxx0101	(6)		%	5	E	U	e	u			ø	ø	ø	ø	ø	ø	ø
xxxx0110	(7)		&	6	F	U	f	v			ø	ø	ø	ø	ø	ø	ø
xxxx0111	(8)		*	7	G	W	g	w			ø	ø	ø	ø	ø	ø	ø
xxxx1000	(1)		(8	H	X	h	x			ø	ø	ø	ø	ø	ø	ø
xxxx1001	(2))	9	I	Y	i	y			ø	ø	ø	ø	ø	ø	ø
xxxx1010	(3)		*	:	J	Z	j	z			ø	ø	ø	ø	ø	ø	ø
xxxx1011	(4)		+	:	K	C	k	{			ø	ø	ø	ø	ø	ø	ø
xxxx1100	(5)		,	<	L	¥	l	l			ø	ø	ø	ø	ø	ø	ø
xxxx1101	(6)		-	=	M	M	m)			ø	ø	ø	ø	ø	ø	ø
xxxx1110	(7)		,	>	N	^	n	*			ø	ø	ø	ø	ø	ø	ø
xxxx1111	(8)		/	?D	_	O	o	*			ø	ø	ø	ø	ø	ø	ø

Note: The user can specify any pattern for character-generator RAM.

Table 4 Correspondence between Character Codes and Character Patterns (ROM Code: A02)

Lower 4 Bits	Upper 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
	CG RAM (1)	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
xxxx0000	(2)	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
xxxx0001	(3)	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
xxxx0010	(4)	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
xxxx0011	(5)	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
xxxx0100	(6)	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
xxxx0101	(7)	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
xxxx0110	(8)	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
xxxx0111	(1)	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
xxxx1000	(2)	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
xxxx1001	(3)	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
xxxx1010	(4)	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
xxxx1011	(5)	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
xxxx1100	(6)	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
xxxx1101	(7)	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
xxxx1110	(8)	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
xxxx1111																	

Table 5 Relationship between CGRAM Addresses, Character Codes (DDRAM) and Character Patterns (CGRAM Data)

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 0	0 0 0	0 0 0 0 0 1 0 1 0 0 1 1 1 0 0 1 0 1 1 1 0 1 1 1	*	* * * 1 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 1 1 1 0 1 0 1 0 0 1 0 0 1 0 1 0 0 0 1 0 0 0 0 0
0 0 0 0 * 0 0 1	0 0 1	0 0 0 0 0 1 0 1 0 0 1 1 1 0 0 1 0 1 1 1 0 1 1 1	*	* * * 1 0 0 0 1 0 1 0 1 0 1 1 1 1 1 0 0 1 0 0 1 1 1 1 1 0 0 1 0 0 0 0 0 0 0
0 0 0 0 * 1 1 1	1 1 1	0 0 0 0 0 1 1 0 0 1 0 1 1 1 0 1 1 1	*	* * *

- Notes:
1. Character code bits 0 to 2 correspond to CGRAM address bits 3 to 5 (3 bits: 8 types).
 2. CGRAM address bits 0 to 2 designate the character pattern line position. The 8th line is the cursor position and its display is formed by a logical OR with the cursor. Maintain the 8th line data, corresponding to the cursor display position, at 0 as the cursor display. If the 8th line data is 1, 1 bits will light up the 8th line regardless of the cursor presence.
 3. Character pattern row positions correspond to CGRAM data bits 0 to 4 (bit 4 being at the left).
 4. As shown Table 5, CGRAM character patterns are selected when character code bits 4 to 7 are all 0. However, since character code bit 3 has no effect, the R display example above can be selected by either character code 00H or 08H.
 5. 1 for CGRAM data corresponds to display selection and 0 to non-selection.
- * Indicates no effect.

Table 5 Relationship between CGRAM Addresses, Character Codes (DDRAM) and Character Patterns (CGRAM Data) (cont)

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 0 0 0 0 0 0	0 0 0 1	* * *	0 0 0 0 0 0 0 0 0 0
		0 0 1 0	0 0 1 1	0 0 0 1	0 0 0 0 0 0 0 0 0 0
		0 1 0 0	0 1 0 1	1 0 1 1	1 0 1 1 0 0 0 1
		0 1 1 0	0 1 1 1	1 1 0 0	1 1 1 0 0 1
		0 1 1 1	1 0 0 0	1 0 0 0	1 0 0 0 0 0 1
		1 0 0 0	1 0 0 1	1 0 0 0	1 0 0 0 0 0 1
		1 0 0 1	1 0 1 0	1 0 0 0	1 0 0 0 0 0 0
		1 0 1 0	1 0 1 1	0 0 0 0	0 0 0 0 0 0 0 0
0 0 0 0 * 1 1 *	1 1	1 0 1 1	1 1 0 0	* * *	* * * * * * *
		1 1 0 0	1 1 0 1	* * *	* * * * * * *
		1 1 0 1	1 1 1 0	* * *	* * * * * * *
		1 1 1 0	1 1 1 1	* * *	* * * * * * *
		1 1 1 1	0 0 0 0	* * *	* * * * * * *
		0 0 0 0	0 0 0 1	0 0 0 1	* * * * * * *

The diagram illustrates the relationship between character codes (DDRAM), CGRAM addresses, and character patterns (CGRAM). It shows a 5x10 grid of character patterns. The top row of the grid is labeled 'Character pattern' and the bottom row is labeled 'Cursor position'. Arrows indicate the mapping from character codes to CGRAM addresses and then to character patterns. The cursor position is shown at the bottom of the grid.

- Notes:
1. Character code bits 1 and 2 correspond to CGRAM address bits 4 and 5 (2 bits: 4 types).
 2. CGRAM address bits 0 to 3 designate the character pattern line position. The 11th line is the cursor position and its display is formed by a logical OR with the cursor. Maintain the 11th line data corresponding to the cursor display position at 0 as the cursor display. If the 11th line data is "1", "1" bits will light up the 11th line regardless of the cursor presence. Since lines 12 to 16 are not used for display, they can be used for general data RAM.
 3. Character pattern row positions are the same as 5 × 8 dot character pattern positions.
 4. CGRAM character patterns are selected when character code bits 4 to 7 are all 0. However, since character code bits 0 and 3 have no effect, the P display example above can be selected by character codes 00H, 01H, 08H, and 09H.
 5. 1 for CGRAM data corresponds to display selection and 0 to non-selection.
- * Indicates no effect.

Timing Generation Circuit

The timing generation circuit generates timing signals for the operation of internal circuits such as DDRAM, CGROM and CGRAM. RAM read timing for display and internal operation timing by MPU access are generated separately to avoid interfering with each other. Therefore, when writing data to DDRAM, for example, there will be no undesirable interferences, such as flickering, in areas other than the display area.

Liquid Crystal Display Driver Circuit

The liquid crystal display driver circuit consists of 16 common signal drivers and 40 segment signal drivers. When the character font and number of lines are selected by a program, the required common signal drivers automatically output drive waveforms, while the other common signal drivers continue to output non-selection waveforms.

Sending serial data always starts at the display data character pattern corresponding to the last address of the display data RAM (DDRAM).

Since serial data is latched when the display data character pattern corresponding to the starting address enters the internal shift register, the HD44780U drives from the head display.

Cursor/Blink Control Circuit

The cursor/blink control circuit generates the cursor or character blinking. The cursor or the blinking will appear with the digit located at the display data RAM (DDRAM) address set in the address counter (AC).

For example (Figure 8), when the address counter is 08H, the cursor position is displayed at DDRAM address 08H.

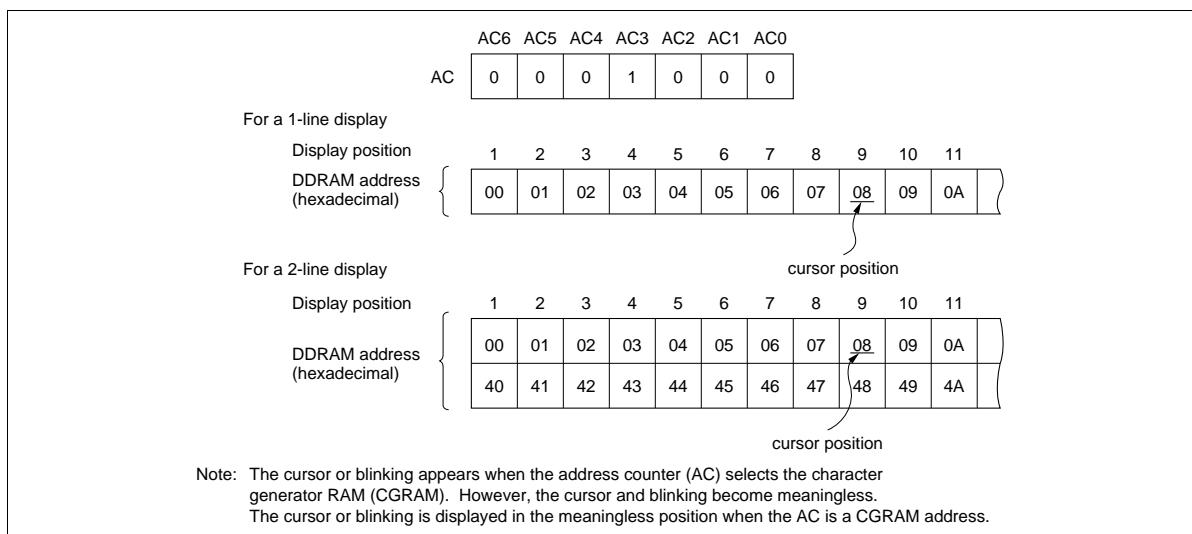


Figure 8 Cursor/Blink Display Example

Interfacing to the MPU

The HD44780U can send data in either two 4-bit operations or one 8-bit operation, thus allowing interfacing with 4- or 8-bit MPUs.

- For 4-bit interface data, only four bus lines (DB4 to DB7) are used for transfer. Bus lines DB0 to DB3 are disabled. The data transfer between the HD44780U and the MPU is completed after the 4-bit data has been transferred twice. As for the order of data transfer, the four high order bits (for 8-bit operation, DB4 to DB7) are transferred before the four low order bits (for 8-bit operation, DB0 to DB3).
The busy flag must be checked (one instruction) after the 4-bit data has been transferred twice. Two more 4-bit operations then transfer the busy flag and address counter data.
- For 8-bit interface data, all eight bus lines (DB0 to DB7) are used.

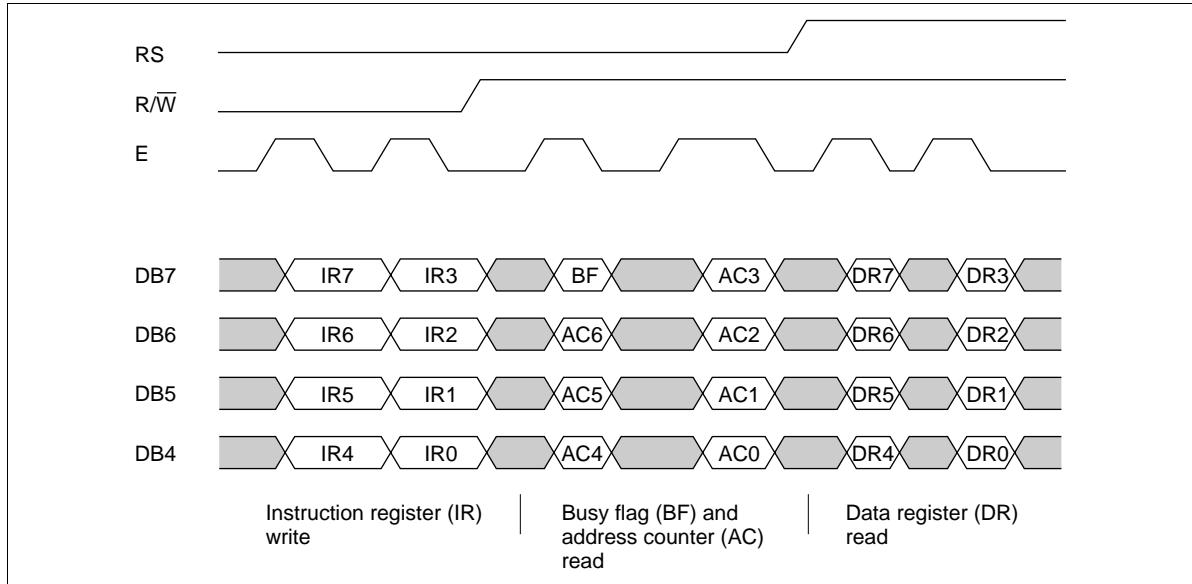


Figure 9 4-Bit Transfer Example

Reset Function

Initializing by Internal Reset Circuit

An internal reset circuit automatically initializes the HD44780U when the power is turned on. The following instructions are executed during the initialization. The busy flag (BF) is kept in the busy state until the initialization ends ($BF = 1$). The busy state lasts for 10 ms after V_{CC} rises to 4.5 V.

1. Display clear
2. Function set:
 $DL = 1$; 8-bit interface data
 $N = 0$; 1-line display
 $F = 0$; 5×8 dot character font
3. Display on/off control:
 $D = 0$; Display off
 $C = 0$; Cursor off
 $B = 0$; Blinking off
4. Entry mode set:
 $I/D = 1$; Increment by 1
 $S = 0$; No shift

Note: If the electrical characteristics conditions listed under the table Power Supply Conditions Using Internal Reset Circuit are not met, the internal reset circuit will not operate normally and will fail to initialize the HD44780U. For such a case, initialization must be performed by the MPU as explained in the section, Initializing by Instruction.

Instructions

Outline

Only the instruction register (IR) and the data register (DR) of the HD44780U can be controlled by the MPU. Before starting the internal operation of the HD44780U, control information is temporarily stored into these registers to allow interfacing with various MPUs, which operate at different speeds, or various peripheral control devices. The internal operation of the HD44780U is determined by signals sent from the MPU. These signals, which include register selection signal (RS), read/

write signal (R/\overline{W}), and the data bus (DB0 to DB7), make up the HD44780U instructions (Table 6). There are four categories of instructions that:

- Designate HD44780U functions, such as display format, data length, etc.
- Set internal RAM addresses
- Perform data transfer with internal RAM
- Perform miscellaneous functions

Normally, instructions that perform data transfer with internal RAM are used the most. However, auto-incrementation by 1 (or auto-decrementation by 1) of internal HD44780U RAM addresses after each data write can lighten the program load of the MPU. Since the display shift instruction (Table 11) can perform concurrently with display data write, the user can minimize system development time with maximum programming efficiency.

When an instruction is being executed for internal operation, no instruction other than the busy flag/address read instruction can be executed.

Because the busy flag is set to 1 while an instruction is being executed, check it to make sure it is 0 before sending another instruction from the MPU.

Note: Be sure the HD44780U is not in the busy state ($BF = 0$) before sending an instruction from the MPU to the HD44780U. If an instruction is sent without checking the busy flag, the time between the first instruction and next instruction will take much longer than the instruction time itself. Refer to Table 6 for the list of each instruction execution time.

Table 6 Instructions

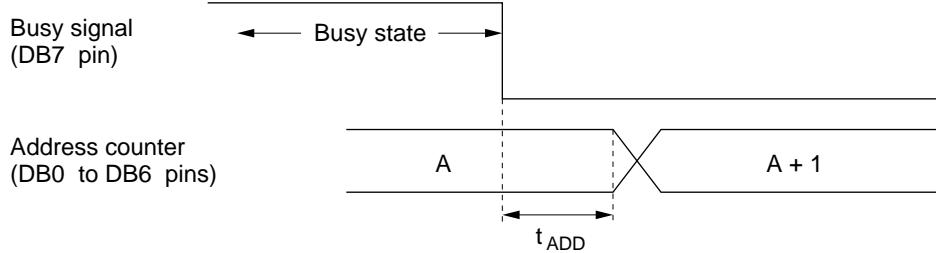
Instruction	Code										Description	Execution Time (max) (when f_{cp} or f_{osc} is 270 kHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear display	0	0	0	0	0	0	0	0	0	1	Clears entire display and sets DDRAM address 0 in address counter.	
Return home	0	0	0	0	0	0	0	0	1	—	Sets DDRAM address 0 in address counter. Also returns display from being shifted to original position. DDRAM contents remain unchanged.	1.52 ms
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction and specifies display shift. These operations are performed during data write and read.	37 μ s
Display on/off control	0	0	0	0	0	0	1	D	C	B	Sets entire display (D) on/off, cursor on/off (C), and blinking of cursor position character (B).	37 μ s
Cursor or display shift	0	0	0	0	0	1	S/C	R/L	—	—	Moves cursor and shifts display without changing DDRAM contents.	37 μ s
Function set	0	0	0	0	1	DL	N	F	—	—	Sets interface data length (DL), number of display lines (N), and character font (F).	37 μ s
Set CGRAM address	0	0	0	1	ACG	ACG	ACG	ACG	ACG	ACG	Sets CGRAM address. CGRAM data is sent and received after this setting.	37 μ s
Set DDRAM address	0	0	1	ADD	Sets DDRAM address. DDRAM data is sent and received after this setting.	37 μ s						
Read busy flag & address	0	1	BF	AC	Reads busy flag (BF) indicating internal operation is being performed and reads address counter contents.	0 μ s						

Table 6 Instructions (cont)

Instruction	RS	R/W	Code								Execution Time (max) (when f_{cp} or f_{osc} is 270 kHz)	
			DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Description	
Write data to CG or DDRAM	1	0	Write data								Writes data into DDRAM or CGRAM.	37 µs $t_{ADD} = 4 \mu s^*$
Read data from CG or DDRAM	1	1	Read data								Reads data from DDRAM or CGRAM.	37 µs $t_{ADD} = 4 \mu s^*$
			I/D = 1: Increment I/D = 0: Decrement S = 1: Accompanies display shift S/C = 1: Display shift S/C = 0: Cursor move R/L = 1: Shift to the right R/L = 0: Shift to the left DL = 1: 8 bits, DL = 0: 4 bits N = 1: 2 lines, N = 0: 1 line F = 1: 5 × 10 dots, F = 0: 5 × 8 dots BF = 1: Internally operating BF = 0: Instructions acceptable							DDRAM: Display data RAM CGRAM: Character generator RAM ACG: CGRAM address ADD: DDRAM address (corresponds to cursor address) AC: Address counter used for both DD and CGRAM addresses	Execution time changes when frequency changes Example: When f_{cp} or f_{osc} is 250 kHz, $37 \mu s \times \frac{270}{250} = 40 \mu s$	

Note: — indicates no effect.

- * After execution of the CGRAM/DDRAM data write or read instruction, the RAM address counter is incremented or decremented by 1. The RAM address counter is updated after the busy flag turns off. In Figure 10, t_{ADD} is the time elapsed after the busy flag turns off until the address counter is updated.



Note: t_{ADD} depends on the operation frequency
 $t_{ADD} = 1.5/(f_{cp} \text{ or } f_{osc})$ seconds

Figure 10 Address Counter Update

Instruction Description

Clear Display

Clear display writes space code 20H (character pattern for character code 20H must be a blank pattern) into all DDRAM addresses. It then sets DDRAM address 0 into the address counter, and returns the display to its original status if it was shifted. In other words, the display disappears and the cursor or blinking goes to the left edge of the display (in the first line if 2 lines are displayed). It also sets I/D to 1 (increment mode) in entry mode. S of entry mode does not change.

Return Home

Return home sets DDRAM address 0 into the address counter, and returns the display to its original status if it was shifted. The DDRAM contents do not change.

The cursor or blinking go to the left edge of the display (in the first line if 2 lines are displayed).

Entry Mode Set

I/D: Increments (I/D = 1) or decrements (I/D = 0) the DDRAM address by 1 when a character code is written into or read from DDRAM.

The cursor or blinking moves to the right when incremented by 1 and to the left when decremented by 1. The same applies to writing and reading of CGRAM.

S: Shifts the entire display either to the right (I/D = 0) or to the left (I/D = 1) when S is 1. The display does not shift if S is 0.

If S is 1, it will seem as if the cursor does not move but the display does. The display does not shift when reading from DDRAM. Also, writing into or reading out from CGRAM does not shift the display.

Display On/Off Control

D: The display is on when D is 1 and off when D is 0. When off, the display data remains in DDRAM, but can be displayed instantly by setting D to 1.

C: The cursor is displayed when C is 1 and not displayed when C is 0. Even if the cursor disappears, the function of I/D or other specifications will not change during display data write. The cursor is displayed using 5 dots in the 8th line for 5×8 dot character font selection and in the 11th line for the 5×10 dot character font selection (Figure 13).

B: The character indicated by the cursor blinks when B is 1 (Figure 13). The blinking is displayed as switching between all blank dots and displayed characters at a speed of 409.6-ms intervals when f_{cp} or f_{osc} is 250 kHz. The cursor and blinking can be set to display simultaneously. (The blinking frequency changes according to f_{osc} or the reciprocal of f_{cp} . For example, when f_{cp} is 270 kHz, $409.6 \times 250 / 270 = 379.2$ ms.)

Cursor or Display Shift

Cursor or display shift shifts the cursor position or display to the right or left without writing or reading display data (Table 7). This function is used to correct or search the display. In a 2-line display, the cursor moves to the second line when it passes the 40th digit of the first line. Note that the first and second line displays will shift at the same time.

When the displayed data is shifted repeatedly each line moves only horizontally. The second line display does not shift into the first line position.

The address counter (AC) contents will not change if the only action performed is a display shift.

Function Set

DL: Sets the interface data length. Data is sent or received in 8-bit lengths (DB7 to DB0) when DL is 1, and in 4-bit lengths (DB7 to DB4) when DL is 0. When 4-bit length is selected, data must be sent or received twice.

N: Sets the number of display lines.

F: Sets the character font.

Note: Perform the function at the head of the program before executing any instructions (except for the read busy flag and address instruction). From this point, the function set instruction cannot be executed unless the interface data length is changed.

Set CGRAM Address

Set CGRAM address sets the CGRAM address binary AAAAAA into the address counter.

Data is then written to or read from the MPU for CGRAM.

		RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Clear display	Code	0	0	0	0	0	0	0	0	1	
		RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Return home	Code	0	0	0	0	0	0	0	0	1	*
		RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Entry mode set	Code	0	0	0	0	0	0	0	1	I/D	S
		RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Display on/off control	Code	0	0	0	0	0	0	1	D	C	B
		RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Cursor or display shift	Code	0	0	0	0	0	1	S/C	R/L	*	*
		RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Function set	Code	0	0	0	0	1	DL	N	F	*	*
		RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Set CGRAM address	Code	0	0	0	1	A	A	A	A	A	A
		← Higher order bit				Lower order bit →					

Figure 11 Instruction (1)

Set DDRAM Address

Set DDRAM address sets the DDRAM address binary AAAAAAA into the address counter.

Data is then written to or read from the MPU for DDRAM.

However, when N is 0 (1-line display), AAAAAAA can be 00H to 4FH. When N is 1 (2-line display), AAAAAAA can be 00H to 27H for the first line, and 40H to 67H for the second line.

Read Busy Flag and Address

Read busy flag and address reads the busy flag (BF) indicating that the system is now internally operating on a previously received instruction. If BF is 1, the internal operation is in progress. The next instruction will not be accepted until BF is reset to 0. Check the BF status before the next write operation. At the same time, the value of the address counter in binary AAAAAAA is read out. This address counter is used by both CG and DDRAM addresses, and its value is determined by the previous instruction. The address contents are the same as for instructions set CGRAM address and set DDRAM address.

Table 7 Shift Function

S/C	R/L	
0	0	Shifts the cursor position to the left. (AC is decremented by one.)
0	1	Shifts the cursor position to the right. (AC is incremented by one.)
1	0	Shifts the entire display to the left. The cursor follows the display shift.
1	1	Shifts the entire display to the right. The cursor follows the display shift.

Table 8 Function Set

N	F	No. of Display Lines	Character Font	Duty Factor	Remarks
0	0	1	5 × 8 dots	1/8	
0	1	1	5 × 10 dots	1/11	
1	*	2	5 × 8 dots	1/16	Cannot display two lines for 5 × 10 dot character font

Note: * Indicates don't care.

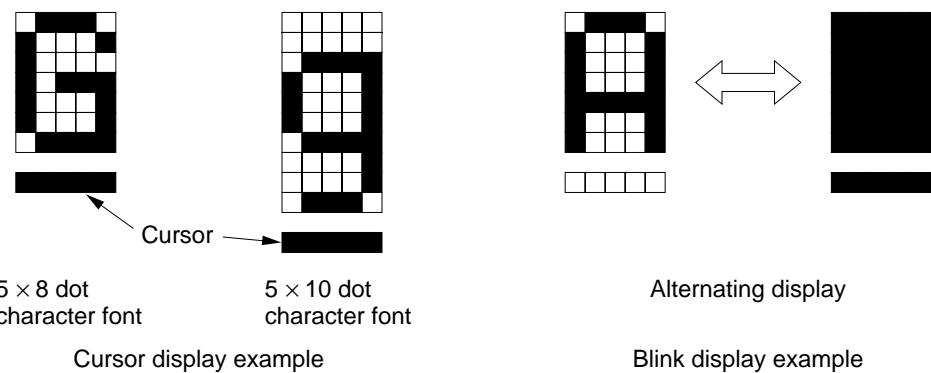


Figure 12 Cursor and Blinking

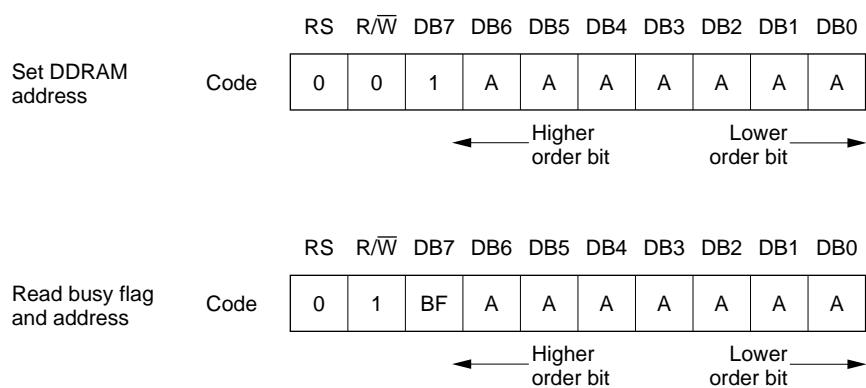


Figure 13 Instruction (2)

Write Data to CG or DDRAM

Write data to CG or DDRAM writes 8-bit binary data DDDDDDDDD to CG or DDRAM.

To write into CG or DDRAM is determined by the previous specification of the CGRAM or DDRAM address setting. After a write, the address is automatically incremented or decremented by 1 according to the entry mode. The entry mode also determines the display shift.

Read Data from CG or DDRAM

Read data from CG or DDRAM reads 8-bit binary data DDDDDDDDD from CG or DDRAM.

The previous designation determines whether CG or DDRAM is to be read. Before entering this read instruction, either CGRAM or DDRAM address set instruction must be executed. If not executed, the first read data will be invalid. When serially executing read instructions, the next address data is normally read from the second read. The address set instructions need not be executed just before this read instruction when shifting the cursor by the cursor shift instruction (when reading out DDRAM). The operation of the cursor shift instruction is the same as the set DDRAM address instruction.

After a read, the entry mode automatically increases or decreases the address by 1. However, display shift is not executed regardless of the entry mode.

Note: The address counter (AC) is automatically incremented or decremented by 1 after the write instructions to CGRAM or DDRAM are executed. The RAM data selected by the AC cannot be read out at this time even if read instructions are executed. Therefore, to correctly read data, execute either the address set instruction or cursor shift instruction (only with DDRAM), then just before reading the desired data, execute the read instruction from the second time the read instruction is sent.

		RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Write data to CG or DDRAM	Code	1	0	D	D	D	D	D	D	D	D
		← Higher order bits					Lower order bits →				
Read data from CG or DDRAM	Code	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
		1	1	D	D	D	D	D	D	D	D
		← Higher order bits					Lower order bits →				

Figure 14 Instruction (3)

Interfacing the HD44780U

Interface to MPUs

- Interfacing to an 8-bit MPU

See Figure 16 for an example of using a I/O port (for a single-chip microcomputer) as an interface device.

In this example, P30 to P37 are connected to the data bus DB0 to DB7, and P75 to P77 are connected to E, R/W, and RS, respectively.

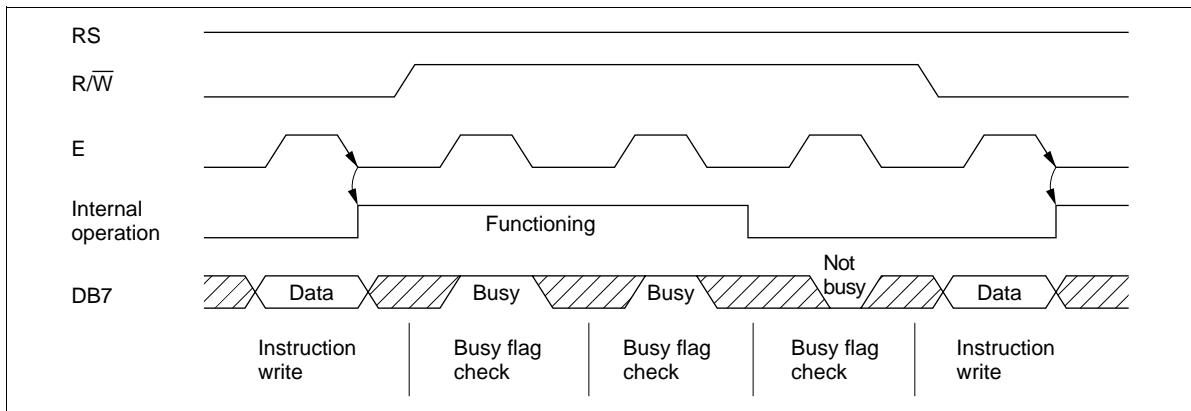


Figure 15 Example of Busy Flag Check Timing Sequence

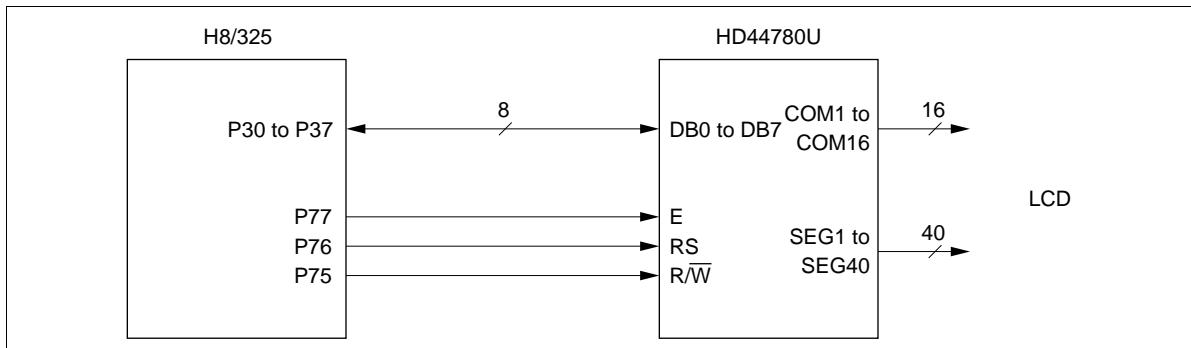


Figure 16 H8/325 Interface (Single-Chip Mode)

- Interfacing to a 4-bit MPU

The HD44780U can be connected to the I/O port of a 4-bit MPU. If the I/O port has enough bits, 8-bit data can be transferred. Otherwise, one data transfer must be made in two operations for 4-bit data. In this case, the timing sequence becomes somewhat complex. (See Figure 17.)

See Figure 18 for an interface example to the HMCS4019R.

Note that two cycles are needed for the busy flag check as well as for the data transfer. The 4-bit operation is selected by the program.

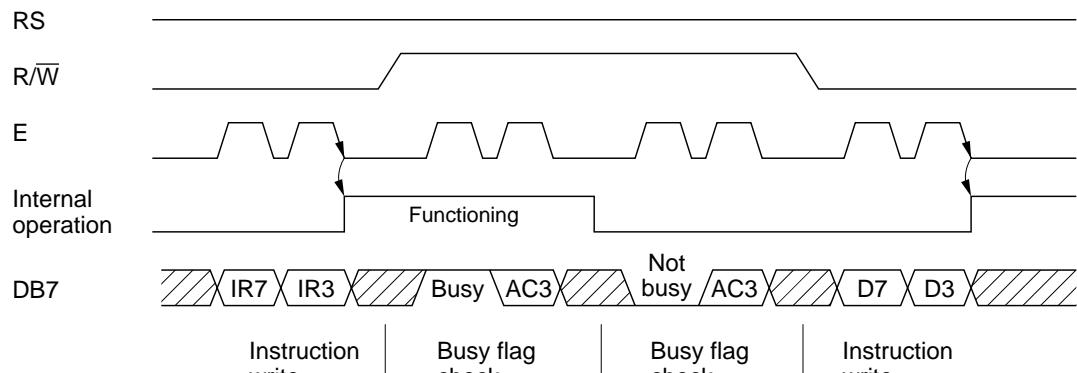


Figure 17 Example of 4-Bit Data Transfer Timing Sequence

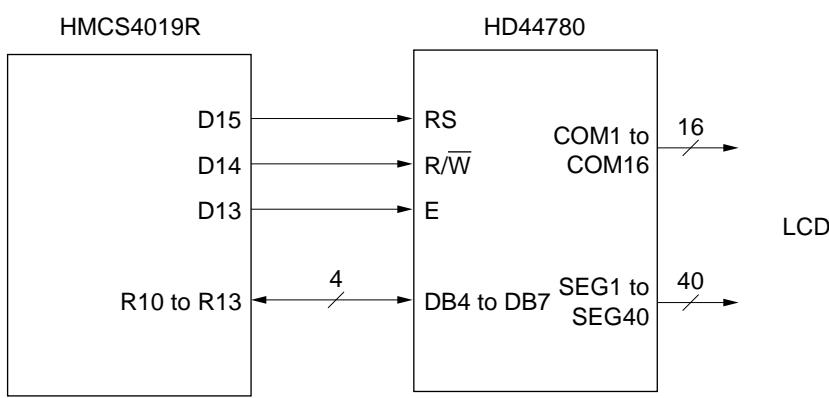


Figure 18 Example of Interface to HMCS4019R

Interface to Liquid Crystal Display

Character Font and Number of Lines: The HD44780U can perform two types of displays, 5×8 dot and 5×10 dot character fonts, each with a cursor.

Up to two lines are displayed for 5×8 dots and one line for 5×10 dots. Therefore, a total of three types of common signals are available (Table 9).

The number of lines and font types can be selected by the program. (See Table 6, Instructions.)

Connection to HD44780 and Liquid Crystal Display: See Figure 19 for the connection examples.

Table 9 Common Signals

Number of Lines	Character Font	Number of Common Signals	Duty Factor
1	5×8 dots + cursor	8	1/8
1	5×10 dots + cursor	11	1/11
2	5×8 dots + cursor	16	1/16

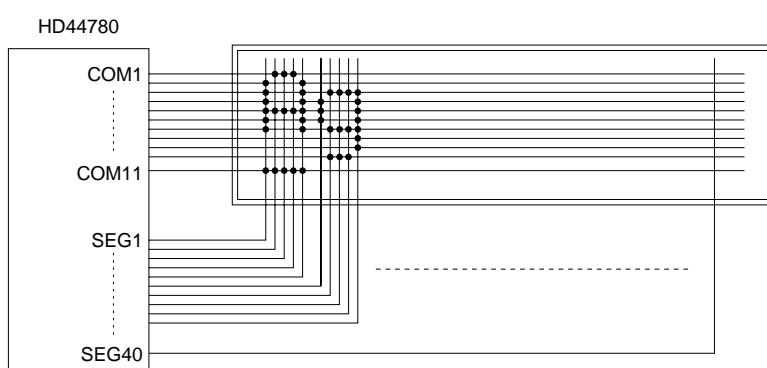
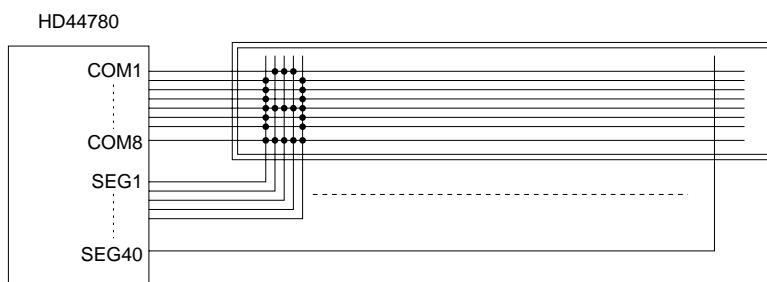


Figure 19 Liquid Crystal Display and HD44780 Connections

Since five segment signal lines can display one digit, one HD44780U can display up to 8 digits for a 1-line display and 16 digits for a 2-line display.

The examples in Figure 19 have unused common signal pins, which always output non-selection waveforms. When the liquid crystal display panel has unused extra scanning lines, connect the extra scanning lines to these common signal pins to avoid any undesirable effects due to crosstalk during the floating state.

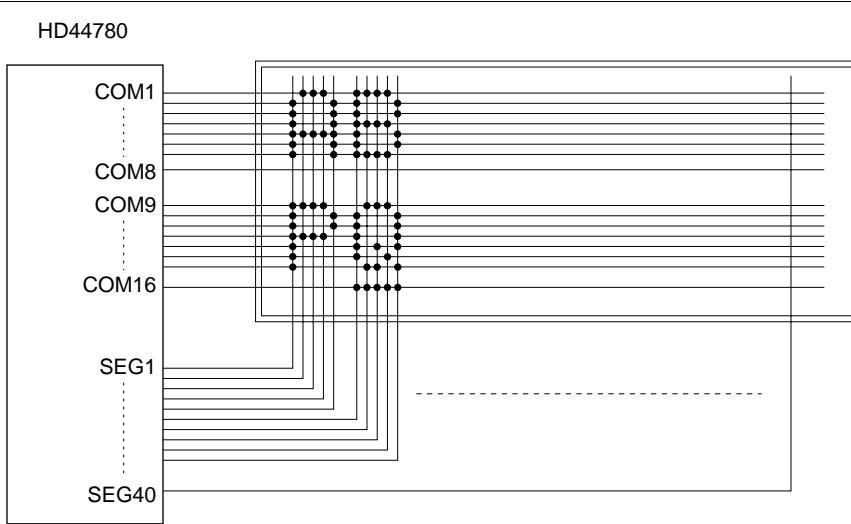


Figure 19 Liquid Crystal Display and HD44780 Connections (cont)

Connection of Changed Matrix Layout: In the preceding examples, the number of lines correspond to the scanning lines. However, the following display examples (Figure 20) are made possible by altering the matrix layout of the liquid crystal display panel. In either case, the only change is the layout. The display characteristics and the number of liquid crystal display characters depend on the number of common signals or on duty factor. Note that the display data RAM (DDRAM) addresses for 4 characters \times 2 lines and for 16 characters \times 1 line are the same as in Figure 19.

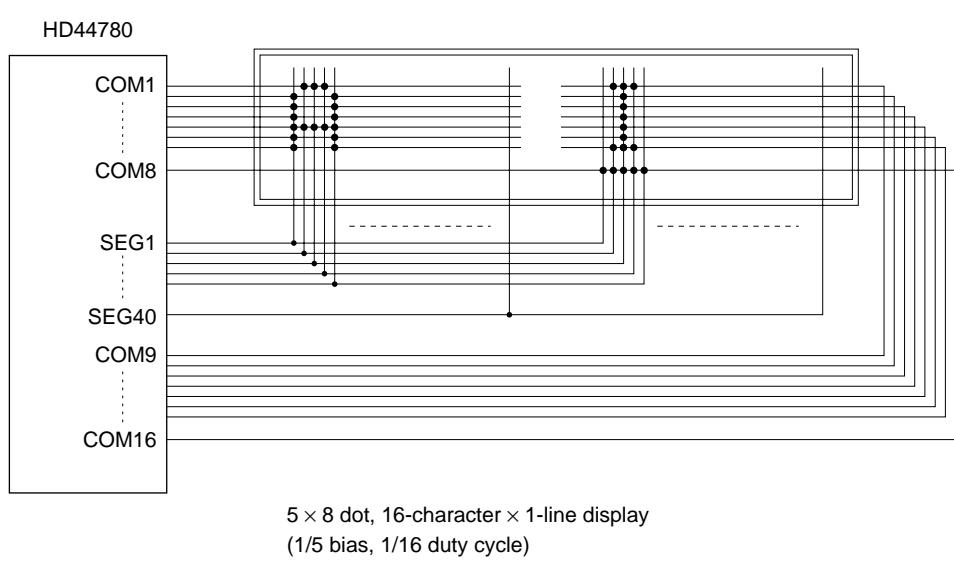


Figure 20 Changed Matrix Layout Displays

Power Supply for Liquid Crystal Display Drive

Various voltage levels must be applied to pins V1 to V5 of the HD44780U to obtain the liquid crystal display drive waveforms. The voltages must be changed according to the duty factor (Table 10).

VLCD is the peak value for the liquid crystal display drive waveforms, and resistance dividing provides voltages V1 to V5 (Figure 21).

Table 10 Duty Factor and Power Supply for Liquid Crystal Display Drive

Power Supply	Duty Factor	
	1/8, 1/11	1/16
	Bias	
V1	1/4	1/5
V2	V _{CC} –1/4 VLCD	V _{CC} –1/5 VLCD
V3	V _{CC} –1/2 VLCD	V _{CC} –2/5 VLCD
V4	V _{CC} –1/2 VLCD	V _{CC} –3/5 VLCD
V5	V _{CC} –3/4 VLCD	V _{CC} –4/5 VLCD
	V _{CC} –VLCD	V _{CC} –VLCD

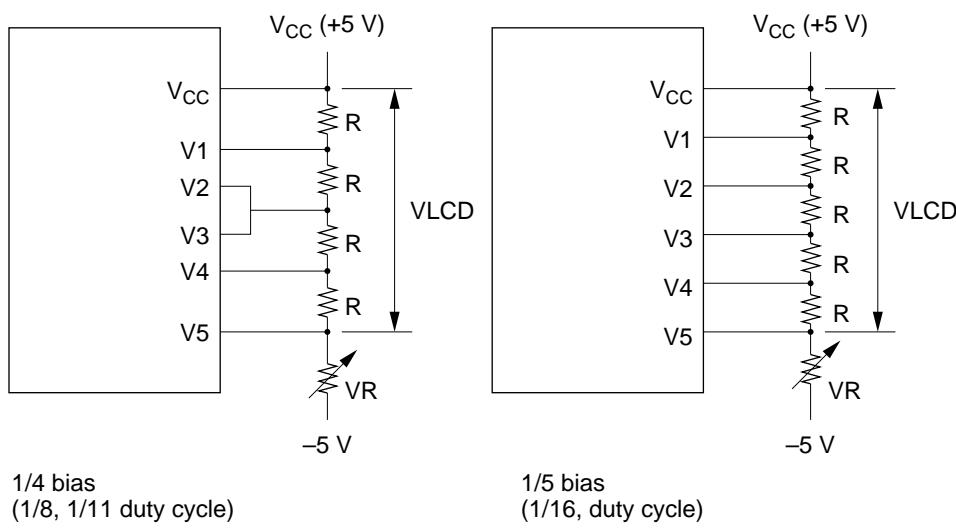
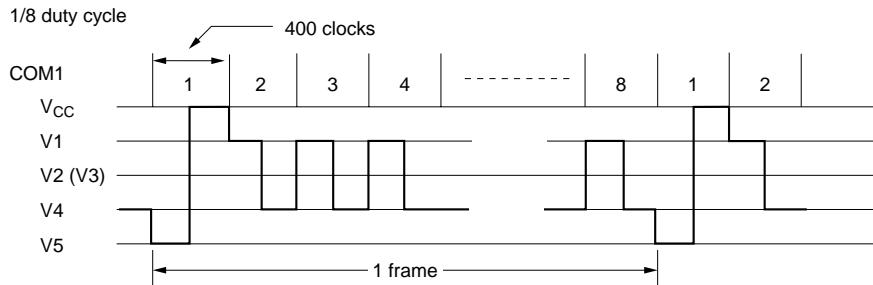


Figure 21 Drive Voltage Supply Example

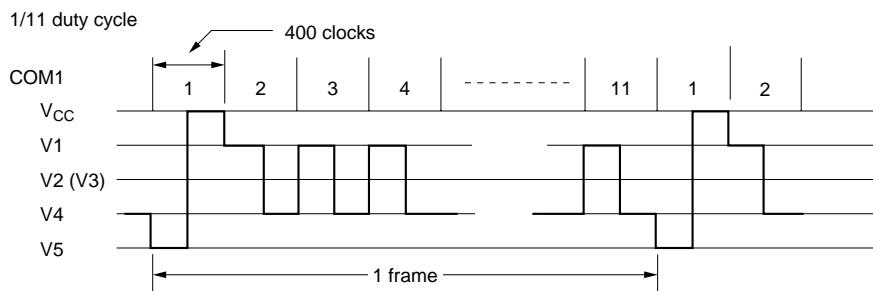
Relationship between Oscillation Frequency and Liquid Crystal Display Frame Frequency

The liquid crystal display frame frequencies of Figure 22 apply only when the oscillation frequency is 270 kHz (one clock pulse of 3.7 μ s).



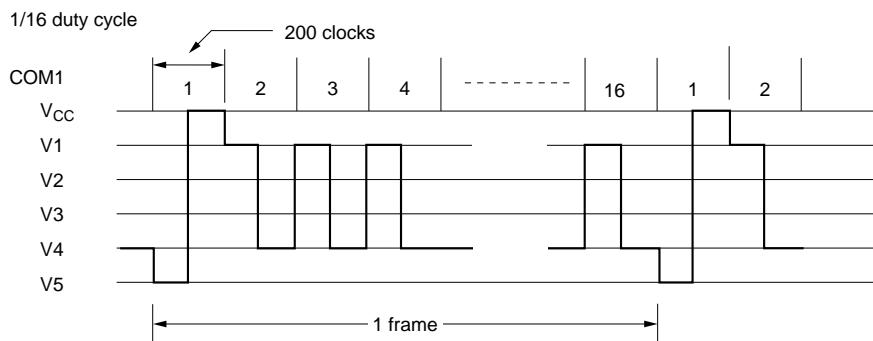
$$1 \text{ frame} = 3.7 \mu\text{s} \times 400 \times 8 = 11850 \mu\text{s} = 11.9 \text{ ms}$$

$$\text{Frame frequency} = \frac{1}{11.9 \text{ ms}} = 84.3 \text{ Hz}$$



$$1 \text{ frame} = 3.7 \mu\text{s} \times 400 \times 11 = 16300 \mu\text{s} = 16.3 \text{ ms}$$

$$\text{Frame frequency} = \frac{1}{16.3 \text{ ms}} = 61.4 \text{ Hz}$$



$$1 \text{ frame} = 3.7 \mu\text{s} \times 200 \times 16 = 11850 \mu\text{s} = 11.9 \text{ ms}$$

$$\text{Frame frequency} = \frac{1}{11.9 \text{ ms}} = 84.3 \text{ Hz}$$

Figure 22 Frame Frequency

Instruction and Display Correspondence

- 8-bit operation, 8-digit × 1-line display with internal reset

Refer to Table 11 for an example of an 8-digit × 1-line display in 8-bit operation. The HD4780U functions must be set by the function set instruction prior to the display. Since the display data RAM can store data for 80 characters, as explained before, the RAM can be used for displays such as for advertising when combined with the display shift operation.

Since the display shift operation changes only the display position with DDRAM contents unchanged, the first display data entered into DDRAM can be output when the return home operation is performed.

- 4-bit operation, 8-digit × 1-line display with internal reset

The program must set all functions prior to the 4-bit operation (Table 12). When the power is turned on, 8-bit operation is automatically selected and the first write is performed as an 8-bit operation. Since DB0 to DB3 are not connected, a rewrite is then required. However, since one operation is completed in two accesses for 4-bit operation, a rewrite is needed to set the functions (see Table 12). Thus, DB4 to DB7 of the function set instruction is written twice.

- 8-bit operation, 8-digit × 2-line display

For a 2-line display, the cursor automatically moves from the first to the second line after the 40th digit of the first line has been written. Thus, if there are only 8 characters in the first line, the DDRAM address must be again set after the 8th character is completed. (See Table 13.) Note that the display shift operation is performed for the first and second lines. In the example of Table 13, the display shift is performed when the cursor is on the second line. However, if the shift operation is performed when the cursor is on the first line, both the first and second lines move together. If the shift is repeated, the display of the second line will not move to the first line. The same display will only shift within its own line for the number of times the shift is repeated.

Note: When using the internal reset, the electrical characteristics in the Power Supply Conditions Using Internal Reset Circuit table must be satisfied. If not, the HD4780U must be initialized by instructions. See the section, Initializing by Instruction.

Table 11 8-Bit Operation, 8-Digit × 1-Line Display Example with Internal Reset

Step No.	Instruction											Operation
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Display	
1	Power supply on (the HD44780U is initialized by the internal reset circuit)											Initialized. No display.
2	Function set											Sets to 8-bit operation and selects 1-line display and 5×8 dot character font. (Number of display lines and character fonts cannot be changed after step #2.)
3	Display on/off control											Turns on display and cursor. Entire display is in space mode because of initialization.
4	Entry mode set											Sets mode to increment the address by one and to shift the cursor to the right at the time of write to the DD/CGRAM. Display is not shifted.
5	Write data to CGRAM/DDRAM											Writes H. DDRAM has already been selected by initialization when the power was turned on. The cursor is incremented by one and shifted to the right.
6	Write data to CGRAM/DDRAM											Writes I.
7	.											.
8	Write data to CGRAM/DDRAM											Writes I.
9	Entry mode set											Sets mode to shift display at the time of write.
10	Write data to CGRAM/DDRAM											Writes a space.

Table 11 8-Bit Operation, 8-Digit × 1-Line Display Example with Internal Reset (cont)

Step	Instruction											Operation	
	No.	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Display	
11	Write data to CGRAM/DDRAM											TACHI M_	Writes M.
	1	0	0	1	0	0	1	1	0	1			
12												.	.
					
13	Write data to CGRAM/DDRAM											MICROKO_	Writes O.
	1	0	0	1	0	0	1	1	1	1	1		
14	Cursor or display shift											MICROKO	Shifts only the cursor position to the left.
	0	0	0	0	0	1	0	0	*	*			
15	Cursor or display shift											MICROKO	Shifts only the cursor position to the left.
	0	0	0	0	0	1	0	0	*	*			
16	Write data to CGRAM/DDRAM											ICROCO	Writes C over K. The display moves to the left.
	1	0	0	1	0	0	0	0	1	1			
17	Cursor or display shift											MICROCO	Shifts the display and cursor position to the right.
	0	0	0	0	0	1	1	1	*	*			
18	Cursor or display shift											MICROCO_	Shifts the display and cursor position to the right.
	0	0	0	0	0	1	0	1	*	*			
19	Write data to CGRAM/DDRAM											ICROCOM_	Writes M.
	1	0	0	1	0	0	1	1	0	1			
20												.	.
					
21	Return home											HITACHI	Returns both display and cursor to the original position (address 0).
	0	0	0	0	0	0	0	0	1	0			

Table 12 4-Bit Operation, 8-Digit × 1-Line Display Example with Internal Reset

Step	Instruction						Display	Operation	
	No.	RS	R/W	DB7	DB6	DB5	DB4		
1	Power supply on (the HD44780U is initialized by the internal reset circuit)								Initialized. No display.
2	Function set	0	0	0	0	1	0		Sets to 4-bit operation. In this case, operation is handled as 8 bits by initialization, and only this instruction completes with one write.
3	Function set	0	0	0	0	1	0		Sets 4-bit operation and selects 1-line display and 5 × 8 dot character font. 4-bit operation starts from this step and resetting is necessary. (Number of display lines and character fonts cannot be changed after step #3.)
4	Display on/off control	0	0	0	0	0	0	—	Turns on display and cursor. Entire display is in space mode because of initialization.
5	Entry mode set	0	0	0	0	0	0	—	Sets mode to increment the address by one and to shift the cursor to the right at the time of write to the DD/CGRAM. Display is not shifted.
6	Write data to CGRAM/DDRAM	1	0	0	1	0	0	H_	Writes H. The cursor is incremented by one and shifts to the right.

Note: The control is the same as for 8-bit operation beyond step #6.

Table 13 8-Bit Operation, 8-Digit × 2-Line Display Example with Internal Reset

Step	Instruction											Operation	
	No.	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Display	
1	Power supply on (the HD44780U is initialized by the internal reset circuit)												Initialized. No display.
2	Function set												Sets to 8-bit operation and selects 2-line display and 5 × 8 dot character font.
3	Display on/off control												Turns on display and cursor. All display is in space mode because of initialization.
4	Entry mode set												Sets mode to increment the address by one and to shift the cursor to the right at the time of write to the DD/CGRAM. Display is not shifted.
5	Write data to CGRAM/DDRAM											H_	Writes H. DDRAM has already been selected by initialization when the power was turned on. The cursor is incremented by one and shifted to the right.
6	.											.	.
7	Write data to CGRAM/DDRAM											HITACHI_	Writes I.
8	Set DDRAM address											HITACHI	Sets DDRAM address so that the cursor is positioned at the head of the second line.
												_	

Table 13 8-Bit Operation, 8-Digit × 2-Line Display Example with Internal Reset (cont)

Step	Instruction											Operation	
	No.	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Display	
9	Write data to CGRAM/DDRAM											HITACHI M_	Writes M.
10					
11	Write data to CGRAM/DDRAM											HITACHI MICROCO_	Writes O.
12	Entry mode set											HITACHI MICROCO_	Sets mode to shift display at the time of write.
13	Write data to CGRAM/DDRAM											ITACHI ICROCOM_	Writes M. Display is shifted to the left. The first and second lines both shift at the same time.
14					
15	Return home											HITACHI MICROCOM	Returns both display and cursor to the original position (address 0).

Initializing by Instruction

If the power supply conditions for correctly operating the internal reset circuit are not met, initialization by instructions becomes necessary.

Refer to Figures 23 and 24 for the procedures on 8-bit and 4-bit initializations, respectively.

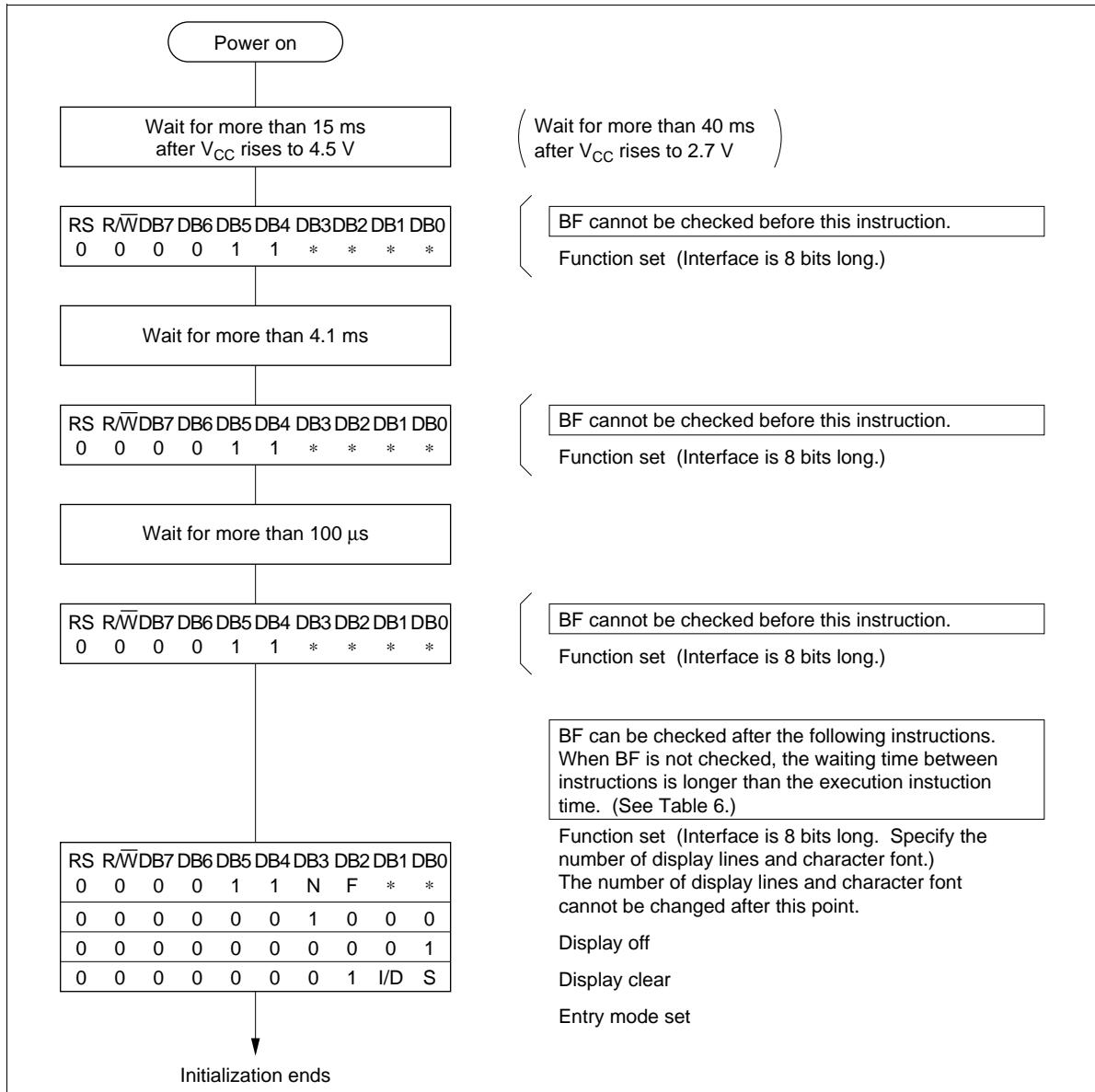


Figure 23 8-Bit Interface

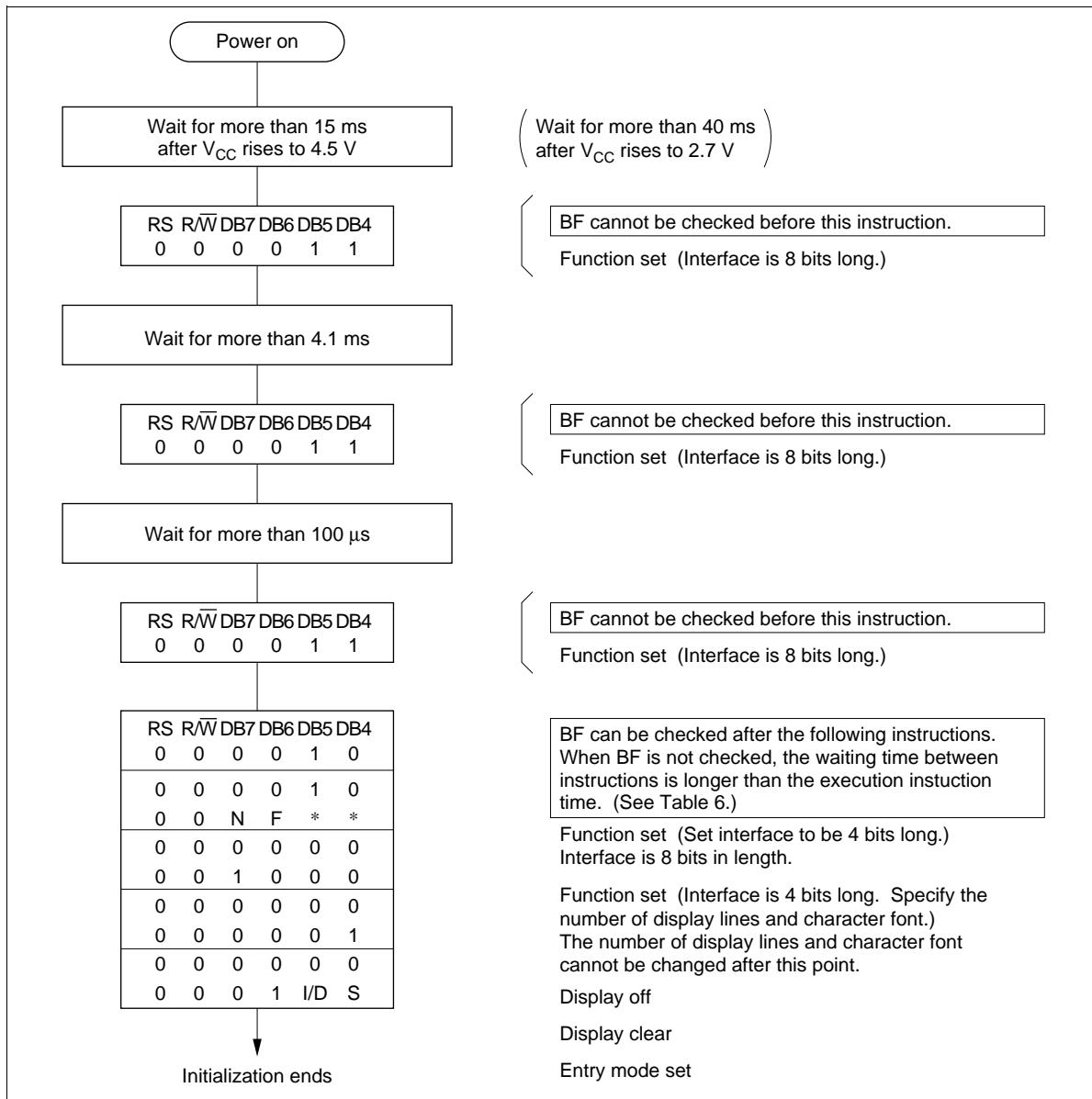


Figure 24 4-Bit Interface

Absolute Maximum Ratings*

Item	Symbol	Value	Unit	Notes
Power supply voltage (1)	V_{cc} -GND	-0.3 to +7.0	V	1
Power supply voltage (2)	V_{cc} -V5	-0.3 to +13.0	V	1, 2
Input voltage	V_t	-0.3 to V_{cc} +0.3	V	1
Operating temperature	T_{opr}	-30 to +75	°C	
Storage temperature	T_{stg}	-55 to +125	°C	4

Note: * If the LSI is used above these absolute maximum ratings, it may become permanently damaged. Using the LSI within the following electrical characteristic limits is strongly recommended for normal operation. If these electrical characteristic conditions are also exceeded, the LSI will malfunction and cause poor reliability.

DC Characteristics ($V_{CC} = 2.7$ to 4.5 V, $T_a = -30$ to $+75^\circ\text{C}$ *³)

Item	Symbol	Min	Typ	Max	Unit	Test Condition	Notes*
Input high voltage (1) (except OSC1)	VIH1	$0.7V_{CC}$	—	V_{CC}	V		6
Input low voltage (1) (except OSC1)	VIL1	-0.3	—	0.55	V		6
Input high voltage (2) (OSC1)	VIH2	$0.7V_{CC}$	—	V_{CC}	V		15
Input low voltage (2) (OSC1)	VIL2	—	—	$0.2V_{CC}$	V		15
Output high voltage (1) (DB0–DB7)	VOH1	$0.75V_{CC}$	—	—	V	$-I_{OH} = 0.1$ mA	7
Output low voltage (1) (DB0–DB7)	VOL1	—	—	$0.2V_{CC}$	V	$I_{OL} = 0.1$ mA	7
Output high voltage (2) (except DB0–DB7)	VOH2	$0.8V_{CC}$	—	—	V	$-I_{OH} = 0.04$ mA	8
Output low voltage (2) (except DB0–DB7)	VOL2	—	—	$0.2V_{CC}$	V	$I_{OL} = 0.04$ mA	8
Driver on resistance (COM)	R_{COM}	—	2	20	kΩ	$\pm I_d = 0.05$ mA, $VLCD = 4$ V	13
Driver on resistance (SEG)	R_{SEG}	—	2	30	kΩ	$\pm I_d = 0.05$ mA, $VLCD = 4$ V	13
Input leakage current	I_{LI}	-1	—	1	μA	$VIN = 0$ to V_{CC}	9
Pull-up MOS current (DB0–DB7, RS, R/W)	$-I_p$	10	50	120	μA	$V_{CC} = 3$ V	
Power supply current	I_{CC}	—	150	300	μA	R_f oscillation, external clock $V_{CC} = 3$ V, $f_{osc} = 270$ kHz	10, 14
LCD voltage	VLCD1	3.0	—	11.0	V	$V_{CC} - V_5$, 1/5 bias	16
	VLCD2	3.0	—	11.0	V	$V_{CC} - V_5$, 1/4 bias	16

Note: * Refer to the Electrical Characteristics Notes section following these tables.

AC Characteristics ($V_{CC} = 2.7$ to 4.5 V, $T_a = -30$ to $+75^\circ\text{C}$ ³⁾

Clock Characteristics

Item		Symbol	Min	Typ	Max	Unit	Test Condition	Note*
External clock operation	External clock frequency	f_{cp}	125	250	350	kHz		11
	External clock duty	Duty	45	50	55	%		
	External clock rise time	t_{rcp}	—	—	0.2	μs		
	External clock fall time	t_{fcp}	—	—	0.2	μs		
R_f oscillation	Clock oscillation frequency	f_{osc}	190	270	350	kHz	$R_f = 75 \text{ k}\Omega$, $V_{CC} = 3 \text{ V}$	12

Note: * Refer to the Electrical Characteristics Notes section following these tables.

Bus Timing Characteristics

Write Operation

Item		Symbol	Min	Typ	Max	Unit	Test Condition
Enable cycle time		t_{cycE}	1000	—	—	ns	Figure 25
Enable pulse width (high level)		PW_{EH}	450	—	—		
Enable rise/fall time		t_{Er}, t_{Ef}	—	—	25		
Address set-up time (RS, R/W to E)		t_{AS}	60	—	—		
Address hold time		t_{AH}	20	—	—		
Data set-up time		t_{DSW}	195	—	—		
Data hold time		t_H	10	—	—		

Read Operation

Item		Symbol	Min	Typ	Max	Unit	Test Condition
Enable cycle time		t_{cycE}	1000	—	—	ns	Figure 26
Enable pulse width (high level)		PW_{EH}	450	—	—		
Enable rise/fall time		t_{Er}, t_{Ef}	—	—	25		
Address set-up time (RS, R/W to E)		t_{AS}	60	—	—		
Address hold time		t_{AH}	20	—	—		
Data delay time		t_{DDR}	—	—	360		
Data hold time		t_{DHR}	5	—	—		

Interface Timing Characteristics with External Driver

Item		Symbol	Min	Typ	Max	Unit	Test Condition
Clock pulse width	High level	t_{CWH}	800	—	—	ns	Figure 27
	Low level	t_{CWL}	800	—	—	ns	
Clock set-up time		t_{CSU}	500	—	—	ns	
Data set-up time		t_{SU}	300	—	—	ns	
Data hold time		t_{DH}	300	—	—	ns	
M delay time		t_{DM}	—1000	—	1000	ns	
Clock rise/fall time		t_{ct}	—	—	200	ns	

Power Supply Conditions Using Internal Reset Circuit

Item		Symbol	Min	Typ	Max	Unit	Test Condition
Power supply rise time		t_{rCC}	0.1	—	10	ms	Figure 28
Power supply off time		t_{OFF}	1	—	—	ms	

DC Characteristics ($V_{CC} = 4.5$ to 5.5 V, $T_a = -30$ to $+75^\circ\text{C}$ *³)

Item	Symbol	Min	Typ	Max	Unit	Test Condition	Notes*
Input high voltage (1) (except OSC1)	VIH1	2.2	—	V_{CC}	V		6
Input low voltage (1) (except OSC1)	VIL1	-0.3	—	0.6	V		6
Input high voltage (2) (OSC1)	VIH2	$V_{CC}-1.0$	—	V_{CC}	V		15
Input low voltage (2) (OSC1)	VIL2	—	—	1.0	V		15
Output high voltage (1) (DB0–DB7)	VOH1	2.4	—	—	V	$-I_{OH} = 0.205$ mA	7
Output low voltage (1) (DB0–DB7)	VOL1	—	—	0.4	V	$I_{OL} = 1.2$ mA	7
Output high voltage (2) (except DB0–DB7)	VOH2	0.9 V_{CC}	—	—	V	$-I_{OH} = 0.04$ mA	8
Output low voltage (2) (except DB0–DB7)	VOL2	—	—	0.1 V_{CC}	V	$I_{OL} = 0.04$ mA	8
Driver on resistance (COM)	RCOM	—	2	20	kΩ	$\pm I_d = 0.05$ mA, $VLCD = 4$ V	13
Driver on resistance (SEG)	RSEG	—	2	30	kΩ	$\pm I_d = 0.05$ mA, $VLCD = 4$ V	13
Input leakage current	I_{LI}	-1	—	1	μA	$VIN = 0$ to V_{CC}	9
Pull-up MOS current (DB0–DB7, RS, R/W)	$-I_p$	50	125	250	μA	$V_{CC} = 5$ V	
Power supply current	I_{CC}	—	350	600	μA	R_t oscillation, external clock $V_{CC} = 5$ V, $f_{osc} = 270$ kHz	10, 14
LCD voltage	VLCD1	3.0	—	11.0	V	$V_{CC}-V5$, 1/5 bias	16
	VLCD2	3.0	—	11.0	V	$V_{CC}-V5$, 1/4 bias	16

Note: * Refer to the Electrical Characteristics Notes section following these tables.

AC Characteristics ($V_{CC} = 4.5$ to 5.5 V, $T_a = -30$ to $+75^\circ C$ ^{*3})**Clock Characteristics**

Item		Symbol	Min	Typ	Max	Unit	Test Condition	Notes*
External clock operation	External clock frequency	f_{cp}	125	250	350	kHz		11
	External clock duty	Duty	45	50	55	%		11
	External clock rise time	t_{rcp}	—	—	0.2	μs		11
	External clock fall time	t_{fcp}	—	—	0.2	μs		11
R_f oscillation	Clock oscillation frequency	f_{osc}	190	270	350	kHz	$R_f = 91\text{ k}\Omega$ $V_{CC} = 5.0\text{ V}$	12

Note: * Refer to the Electrical Characteristics Notes section following these tables.

Bus Timing Characteristics**Write Operation**

Item		Symbol	Min	Typ	Max	Unit	Test Condition
Enable cycle time		t_{cycE}	500	—	—	ns	Figure 25
Enable pulse width (high level)		PW_{EH}	230	—	—		
Enable rise/fall time		t_{Er}, t_{Ef}	—	—	20		
Address set-up time (RS, R/W to E)		t_{AS}	40	—	—		
Address hold time		t_{AH}	10	—	—		
Data set-up time		t_{DSW}	80	—	—		
Data hold time		t_H	10	—	—		

Read Operation

Item		Symbol	Min	Typ	Max	Unit	Test Condition
Enable cycle time		t_{cycE}	500	—	—	ns	Figure 26
Enable pulse width (high level)		PW_{EH}	230	—	—		
Enable rise/fall time		t_{Er}, t_{Ef}	—	—	20		
Address set-up time (RS, R/W to E)		t_{AS}	40	—	—		
Address hold time		t_{AH}	10	—	—		
Data delay time		t_{DDR}	—	—	160		
Data hold time		t_{DHR}	5	—	—		

Interface Timing Characteristics with External Driver

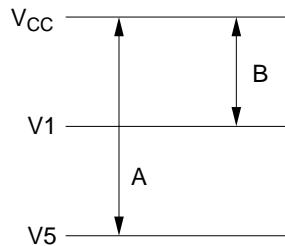
Item		Symbol	Min	Typ	Max	Unit	Test Condition
Clock pulse width	High level	t_{CWH}	800	—	—	ns	Figure 27
	Low level	t_{CWL}	800	—	—		
Clock set-up time		t_{CSU}	500	—	—		
Data set-up time		t_{SU}	300	—	—		
Data hold time		t_{DH}	300	—	—		
M delay time		t_{DM}	—1000	—	1000		
Clock rise/fall time		t_{ct}	—	—	100		

Power Supply Conditions Using Internal Reset Circuit

Item		Symbol	Min	Typ	Max	Unit	Test Condition
Power supply rise time		t_{rCC}	0.1	—	10	ms	Figure 28
Power supply off time		t_{OFF}	1	—	—		

Electrical Characteristics Notes

- All voltage values are referred to GND = 0 V.

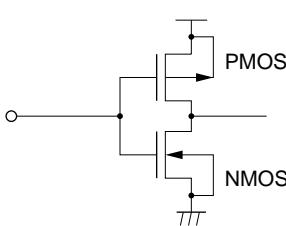


$$\begin{aligned}A &= V_{CC} - V_1 \\B &= V_{CC} - V_5 \\A &\geq 1.5 \text{ V} \\B &\leq 0.25 \times A\end{aligned}$$

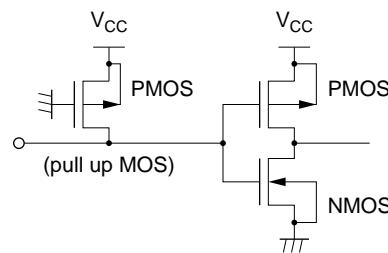
The conditions of V₁ and V₅ voltages are for proper operation of the LSI and not for the LCD output level. The LCD drive voltage condition for the LCD output level is specified as LCD voltage VLCD.

- $V_{CC} \geq V_1 \geq V_2 \geq V_3 \geq V_4 \geq V_5$ must be maintained.
- For die products, specified at 75°C.
- For die products, specified by the die shipment specification.
- The following four circuits are I/O pin configurations except for liquid crystal display output.

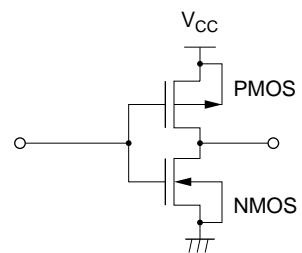
Input pin
Pin: E (MOS without pull-up)



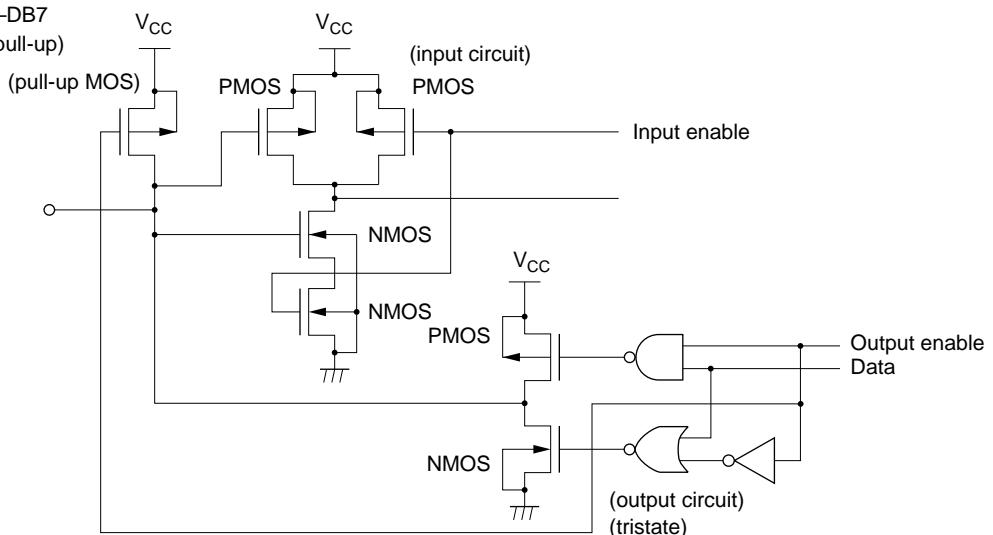
Pins: RS, R/W (MOS with pull-up)



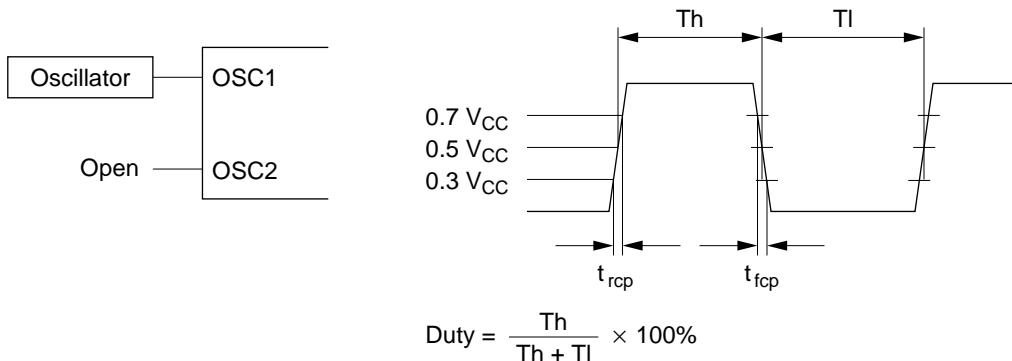
Output pin
Pins: CL1, CL2, M, D



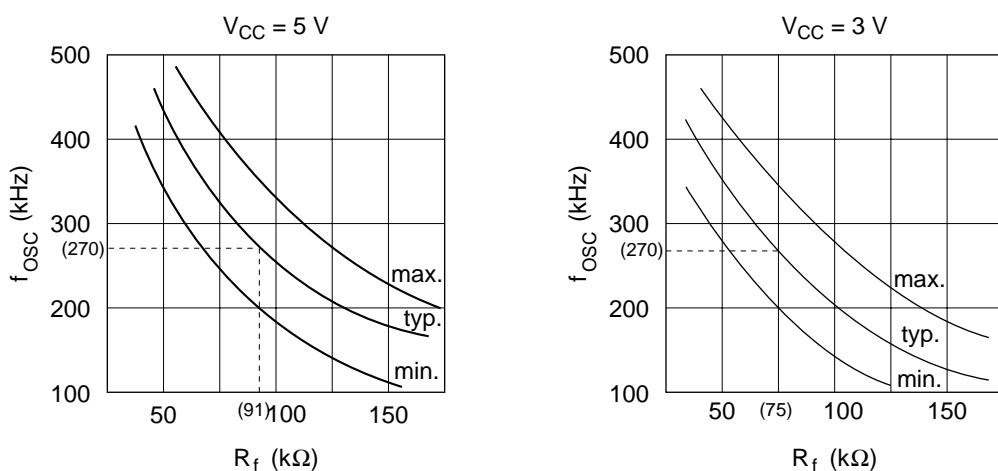
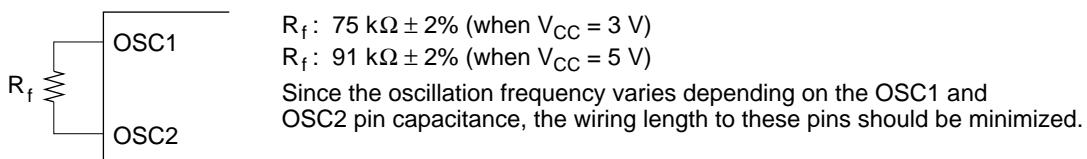
I/O Pin
Pins: DB0 –DB7
(MOS with pull-up)



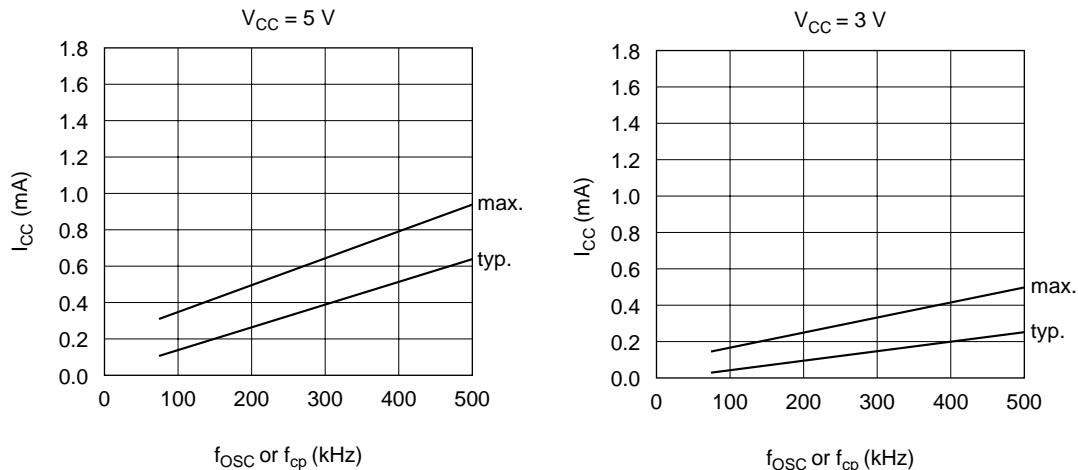
6. Applies to input pins and I/O pins, excluding the OSC1 pin.
7. Applies to I/O pins.
8. Applies to output pins.
9. Current flowing through pull-up MOSSs, excluding output drive MOSSs.
10. Input/output current is excluded. When input is at an intermediate level with CMOS, the excessive current flows through the input circuit to the power supply. To avoid this from happening, the input level must be fixed high or low.
11. Applies only to external clock operation.



12. Applies only to the internal oscillator operation using oscillation resistor R_f.



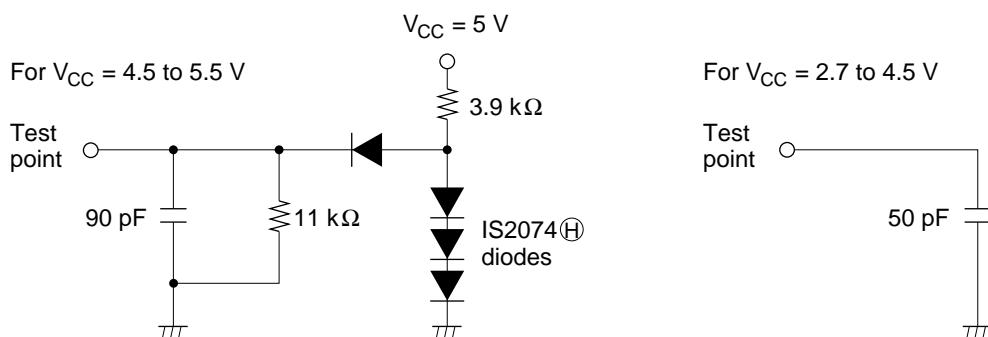
13. RCOM is the resistance between the power supply pins (V_{CC} , V1, V4, V5) and each common signal pin (COM1 to COM16).
RSEG is the resistance between the power supply pins (V_{CC} , V2, V3, V5) and each segment signal pin (SEG1 to SEG40).
14. The following graphs show the relationship between operation frequency and current consumption.



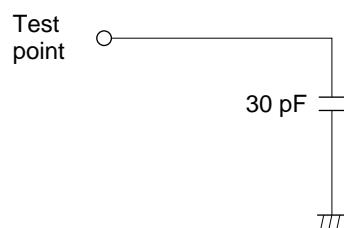
15. Applies to the OSC1 pin.
16. Each COM and SEG output voltage is within $\pm 0.15\text{ V}$ of the LCD voltage (V_{CC} , V1, V2, V3, V4, V5) when there is no load.

Load Circuits

Data Bus DB0 to DB7



External Driver Control Signals: CL1, CL2, D, M



Timing Characteristics

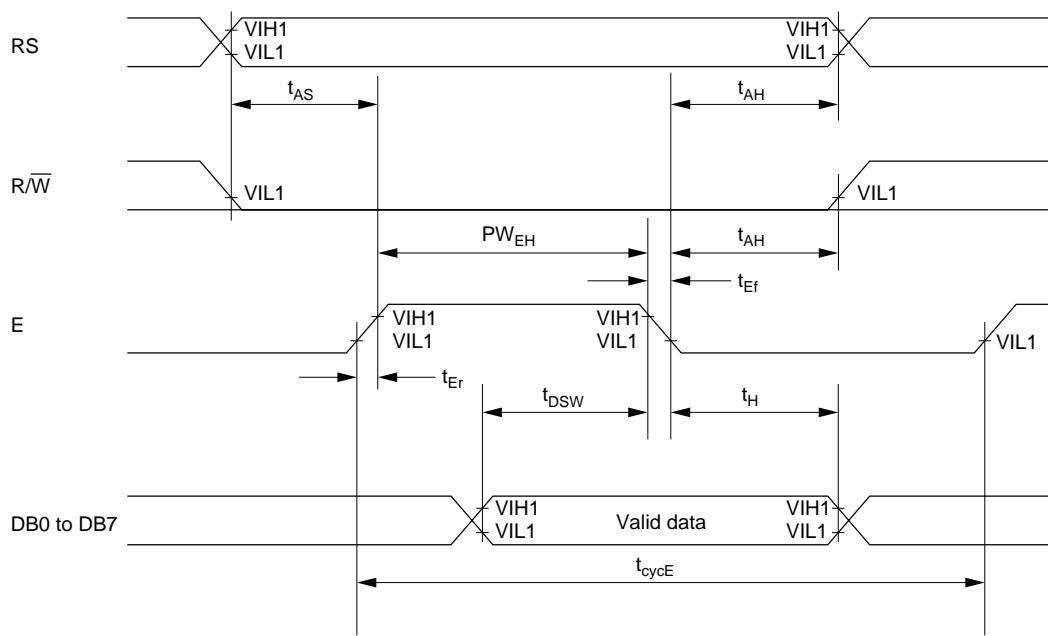
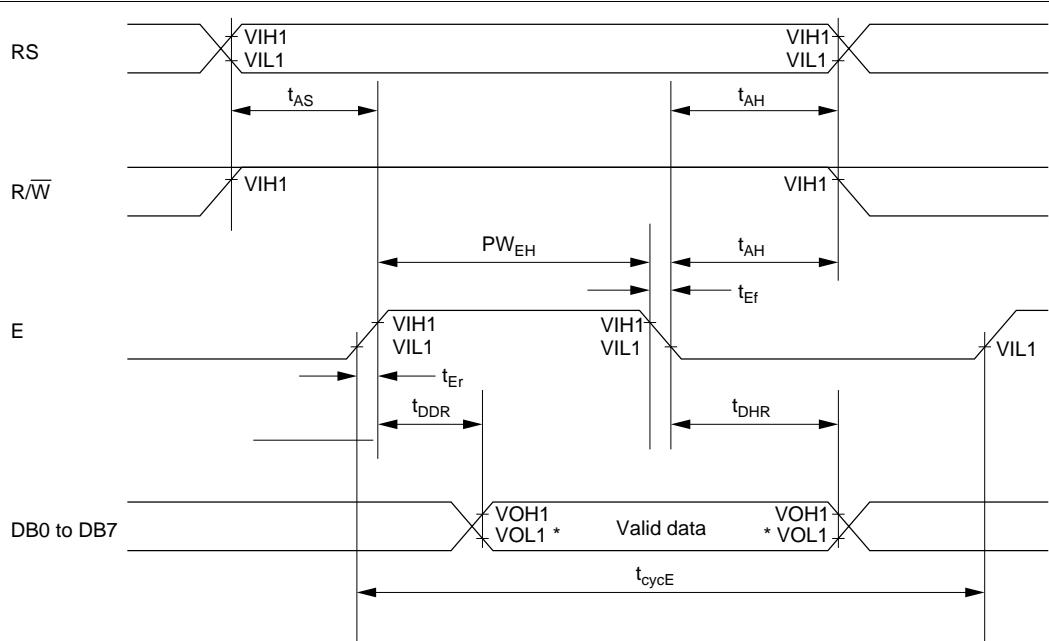


Figure 25 Write Operation



Note: * VOL1 is assumed to be 0.8 V at 2 MHz operation.

Figure 26 Read Operation

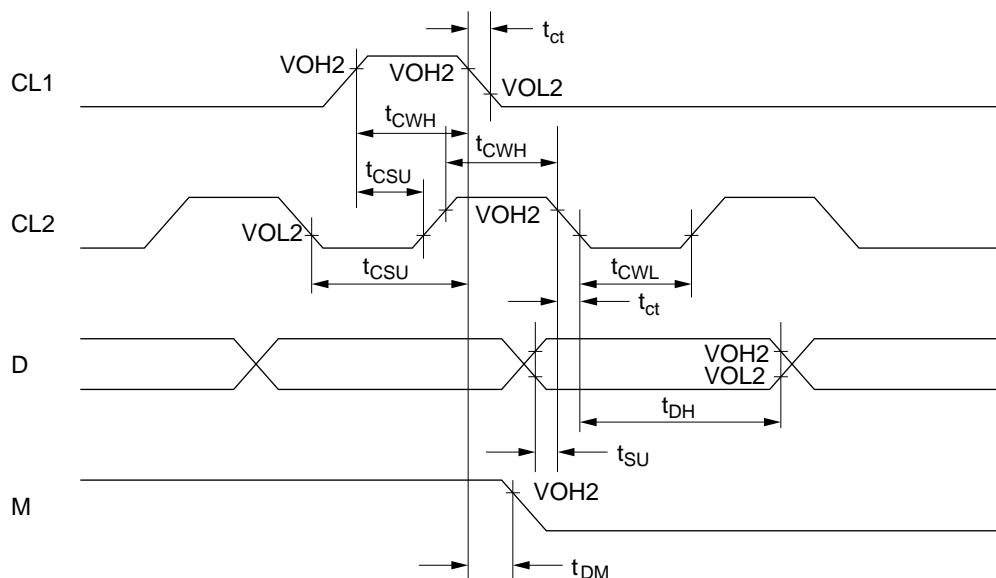
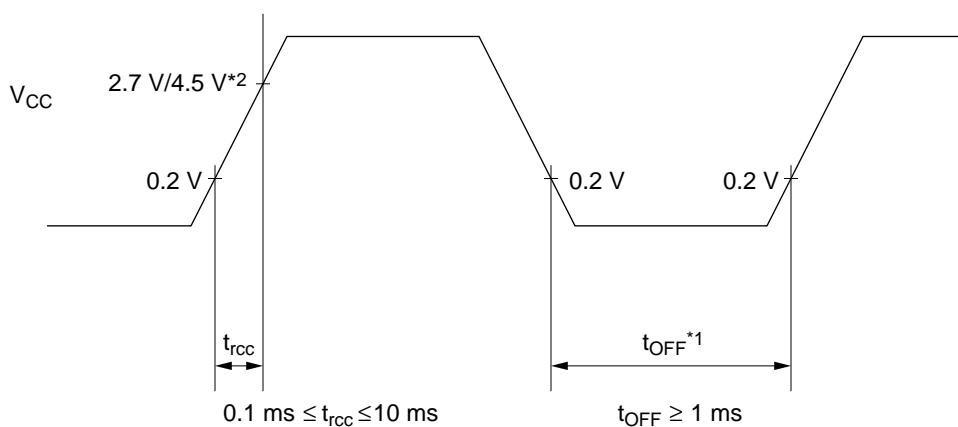


Figure 27 Interface Timing with External Driver



Notes:

1. t_{OFF} compensates for the power oscillation period caused by momentary power supply oscillations.
2. Specified at 4.5 V for 5-V operation, and at 2.7 V for 3-V operation.
3. For if 4.5 V is not reached during 5-V operation, the internal reset circuit will not operate normally.
In this case, the LSI must be initialized by software. (Refer to the Initializing by Instruction section.)

Figure 28 Internal Power Supply Reset

Cautions

1. Hitachi neither warrants nor grants licenses of any rights of Hitachi's or any third party's patent, copyright, trademark, or other intellectual property rights for information contained in this document. Hitachi bears no responsibility for problems that may arise with third party's rights, including intellectual property rights, in connection with use of the information contained in this document.
2. Products and product specifications may be subject to change without notice. Confirm that you have received the latest product standards or specifications before final design, purchase or use.
3. Hitachi makes every attempt to ensure that its products are of high quality and reliability. However, contact Hitachi's sales office before using the product in an application that demands especially high quality and reliability or where its failure or malfunction may directly threaten human life or cause risk of bodily injury, such as aerospace, aeronautics, nuclear power, combustion control, transportation, traffic, safety equipment or medical equipment for life support.
4. Design your application so that the product is used within the ranges guaranteed by Hitachi particularly for maximum rating, operating supply voltage range, heat radiation characteristics, installation conditions and other characteristics. Hitachi bears no responsibility for failure or damage when used beyond the guaranteed ranges. Even within the guaranteed ranges, consider normally foreseeable failure rates or failure modes in semiconductor devices and employ systemic measures such as fail-safes, so that the equipment incorporating Hitachi product does not cause bodily injury, fire or other consequential damage due to operation of the Hitachi product.
5. This product is not designed to be radiation resistant.
6. No one is permitted to reproduce or duplicate, in any form, the whole or part of this document without written approval from Hitachi.
7. Contact Hitachi's sales office for any questions regarding this document or Hitachi semiconductor products.

**Hitachi, Ltd.**

Semiconductor & Integrated Circuits.

Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

Tel: Tokyo (03) 3270-2111 Fax: (03) 3270-5109

URL	NorthAmerica	: http://semiconductor.hitachi.com/
	Europe	: http://www.hitachi-eu.com/hel/ecg
	Asia (Singapore)	: http://www.has.hitachi.com.sg/grp3/sicd/index.htm
	Asia (Taiwan)	: http://www.hitachi.com.tw/E/Product/SICD_Frame.htm
	Asia (HongKong)	: http://www.hitachi.com.hk/eng/bo/grp3/index.htm
	Japan	: http://www.hitachi.co.jp/Sicd/indx.htm

For further information write to:

Hitachi Semiconductor (America) Inc. 179 East Tasman Drive, San Jose, CA 95134 Tel: <1> (408) 433-1990 Fax: <1>(408) 433-0223	Hitachi Europe GmbH Electronic components Group Dornacher Straße 3 D-85622 Feldkirchen, Munich Germany Tel: <49> (89) 9 9180-0 Fax: <49> (89) 9 29 30 00 Hitachi Europe Ltd. Electronic Components Group. Whitebrook Park Lower Cookham Road Maidenhead Berkshire SL6 8YA, United Kingdom Tel: <44> (1628) 585000 Fax: <44> (1628) 778322
--	---

Hitachi Asia Pte. Ltd.	16 Collyer Quay #20-00
Hitachi Tower	Singapore 049318
Tel: 535-2100	Fax: 535-1533
Hitachi Asia Ltd.	Taipei Branch Office
	3F, Hung Kuo Building, No.167, Tun-Hwa North Road, Taipei (105)
	Tel: <886> (2) 2718-3666
	Fax: <886> (2) 2718-8180

Hitachi Asia (Hong Kong) Ltd.	Group III (Electronic Components) 7/F., North Tower, World Finance Centre, Harbour City, Canton Road, Tsim Sha Tsui, Kowloon, Hong Kong
	Tel: <852> (2) 735 9218 Fax: <852> (2) 730 0281 Telex: 40815 HITEC HX

Copyright © Hitachi, Ltd., 1998. All rights reserved. Printed in Japan.





PCA8574/74A

Remote 8-bit I/O expander for I²C-bus with interrupt

Rev. 02 — 14 May 2007

Product data sheet

1. General description

The PCA8574/74A provide general purpose remote I/O expansion for most microcontroller families via the two-line bidirectional I²C-bus (serial clock (SCL), serial data (SDA)).

The devices consist of an 8-bit quasi-bidirectional port and an I²C-bus interface. The PCA8574/74A have low current consumption and include latched outputs with 25 mA high current drive capability for directly driving LEDs.

The PCA8574/74A also possess an interrupt line ($\overline{\text{INT}}$) that can be connected to the interrupt logic of the microcontroller. By sending an interrupt signal on this line, the remote I/O can inform the microcontroller if there is incoming data on its ports without having to communicate via the I²C-bus.

The internal Power-On Reset (POR) initializes the I/Os as inputs.

2. Features

- 400 kHz I²C-bus interface
- 2.3 V to 5.5 V operation with 5.5 V tolerant I/Os
- 8-bit remote I/O pins that default to inputs at power-up
- Latched outputs with 25 mA sink capability for directly driving LEDs
- Total package sink capability of 200 mA
- Active LOW open-drain interrupt output
- 8 programmable slave addresses using 3 address pins
- Readable device ID (manufacturer, device type, and revision)
- Low standby current (10 μA max.)
- –40 °C to +85 °C operation
- ESD protection exceeds 2000 V HBM per JESD22-A114, 200 V MM per JESD22-A115, and 1000 V CDM per JESD22-C101
- Latch-up testing is done to JEDEC standard JESD78 which exceeds 100 mA
- Packages offered: DIP16, SO16, TSSOP16, SSOP20

3. Applications

- LED signs and displays
- Servers
- Industrial control
- Medical equipment
- PLCs

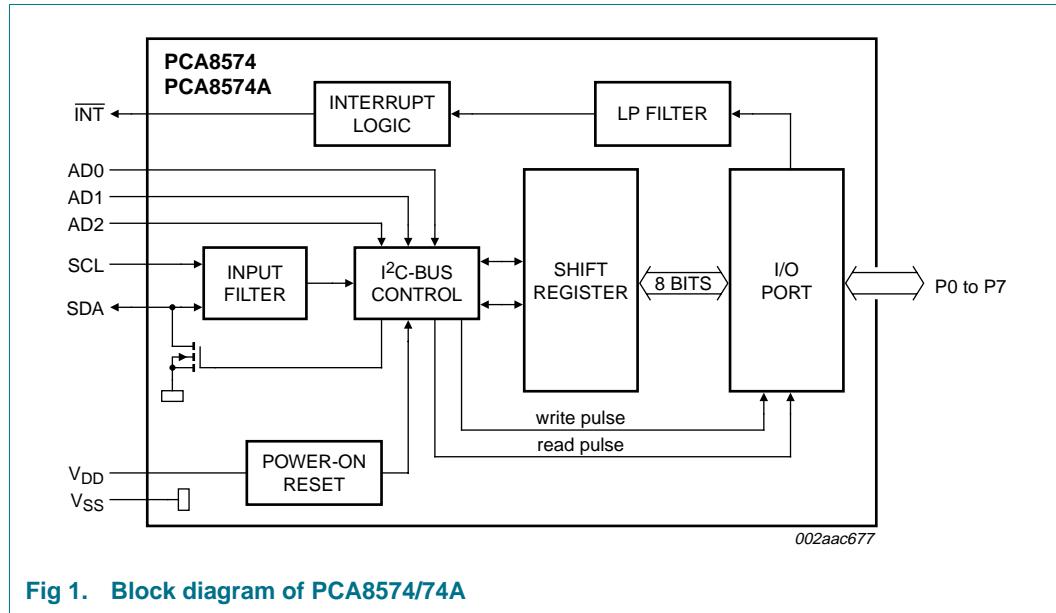
- Cellular telephones
- Gaming machines
- Instrumentation and test measurement

4. Ordering information

Table 1. Ordering information

Type number	Topside mark	Package		Version
		Name	Description	
PCA8574D	PCA8574D	SO16	plastic small outline package; 16 leads; body width 7.5 mm	SOT162-1
PCA8574AD	PCA8574AD			
PCA8574N	PCA8574N	DIP16	plastic dual in-line package; 16 leads (300 mil); long body	SOT38-1
PCA8574AN	PCA8574AN			
PCA8574PW	PCA8574	TSSOP16	plastic thin shrink small outline package; 16 leads; body width 4.4 mm	SOT403-1
PCA8574APW	PA8574A			
PCA8574TS	PCA8574	SSOP20	plastic shrink small outline package; 20 leads; body width 4.4 mm	SOT266-1
PCA8574ATS	PCA8574A			

5. Block diagram



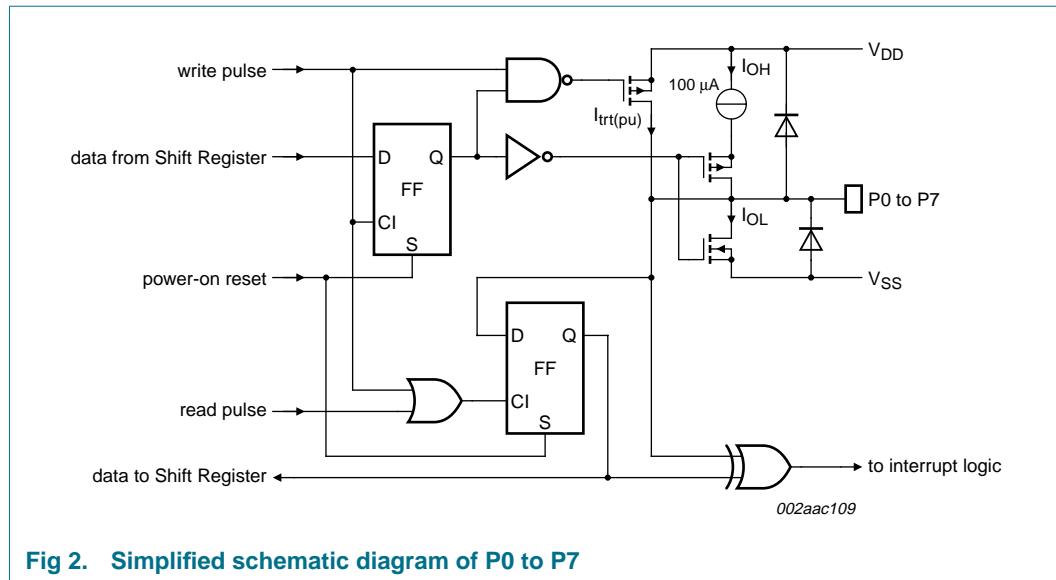


Fig 2. Simplified schematic diagram of P0 to P7

6. Pinning information

6.1 Pinning

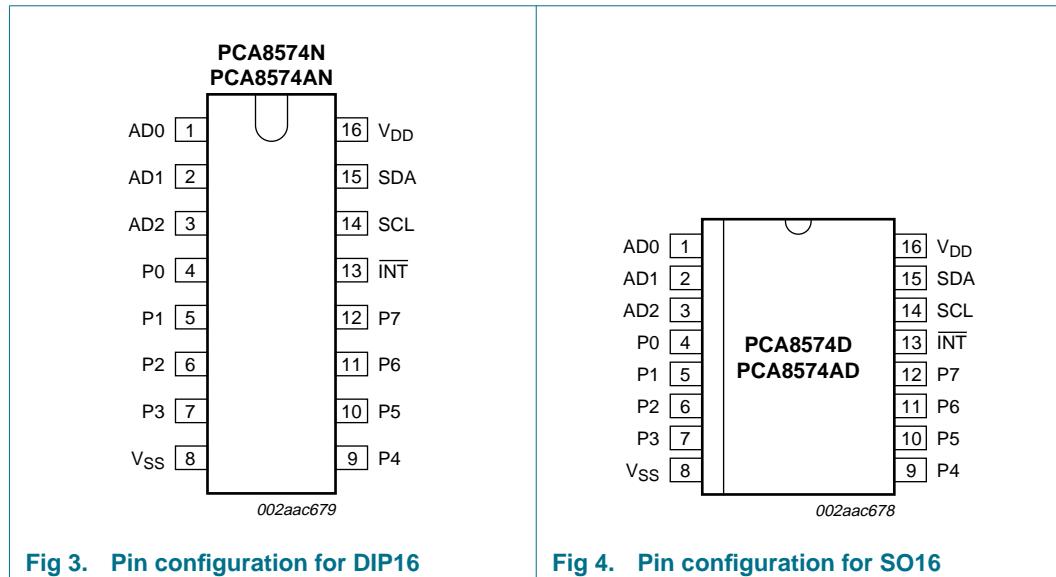


Fig 3. Pin configuration for DIP16

Fig 4. Pin configuration for SO16

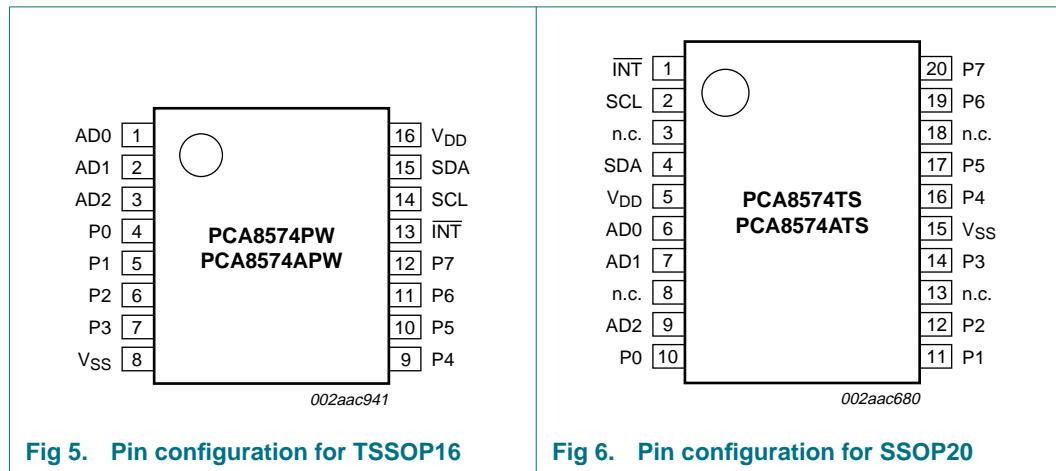


Fig 5. Pin configuration for TSSOP16

Fig 6. Pin configuration for SSOP20

6.2 Pin description

Table 2. Pin description for DIP16, SO16, TSSOP16

Symbol	Pin	Description
AD0	1	address input 0
AD1	2	address input 1
AD2	3	address input 2
P0	4	quasi-bidirectional I/O 0
P1	5	quasi-bidirectional I/O 1
P2	6	quasi-bidirectional I/O 2
P3	7	quasi-bidirectional I/O 3
V _{SS}	8	supply ground
P4	9	quasi-bidirectional I/O 4
P5	10	quasi-bidirectional I/O 5
P6	11	quasi-bidirectional I/O 6
P7	12	quasi-bidirectional I/O 7
INT	13	interrupt output (active LOW)
SCL	14	serial clock line
SDA	15	serial data line
V _{DD}	16	supply voltage

Table 3. Pin description for SSOP20

Symbol	Pin	Description
INT	1	interrupt output (active LOW)
SCL	2	serial clock line
n.c.	3	not connected
SDA	4	serial data line
V _{DD}	5	supply voltage
AD0	6	address input 0
AD1	7	address input 1
n.c.	8	not connected
AD2	9	address input 2
P0	10	quasi-bidirectional I/O 0
P1	11	quasi-bidirectional I/O 1
P2	12	quasi-bidirectional I/O 2
n.c.	13	not connected
P3	14	quasi-bidirectional I/O 3
V _{SS}	15	supply ground
P4	16	quasi-bidirectional I/O 4
P5	17	quasi-bidirectional I/O 5
n.c.	18	not connected
P6	19	quasi-bidirectional I/O 6
P7	20	quasi-bidirectional I/O 7

7. Functional description

Refer to [Figure 1 “Block diagram of PCA8574/74A”](#).

7.1 Device address

Following a START condition, the bus master must send the address of the slave it is accessing and the operation it wants to perform (read or write). The address of the PCA8574/74A is shown in [Figure 7](#). Slave address pins AD2, AD1, and AD0 choose 1 of 8 slave addresses. To conserve power, no internal pull-up resistors are incorporated on AD2, AD1, and AD0. Address values depending on AD2, AD1, and AD0 can be found in [Table 4 “PCA8574 address map”](#) and [Table 5 “PCA8574A address map”](#).

Remark: When using the PCA8574A, the General Call address (0000 0000b) and the Device ID address (1111 100Xb) are reserved and cannot be used as device address. Failure to follow this requirement will cause the PCA8574A not to acknowledge.

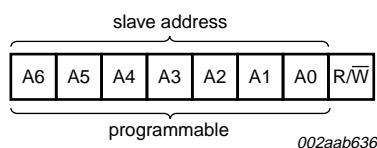


Fig 7. PCA8574/74A address

The last bit of the first byte defines the operation to be performed. When set to logic 1 a read is selected, while a logic 0 selects a write operation.

When AD2, AD1 and AD0 are held to V_{DD} or V_{SS}, the same address as the PCF8574 or PCF8574A is applied.

7.1.1 Address maps

Table 4. PCA8574 address map

A6	A5	A4	A3	A2	A1	A0	Address
0	1	0	0	0	0	0	20h
0	1	0	0	0	0	1	21h
0	1	0	0	0	1	0	22h
0	1	0	0	0	1	1	23h
0	1	0	0	1	0	0	24h
0	1	0	0	1	0	1	25h
0	1	0	0	1	1	0	26h
0	1	0	0	1	1	1	27h

Table 5. PCA8574A address map

A6	A5	A4	A3	A2	A1	A0	Address
0	1	1	1	0	0	0	38h
0	1	1	1	0	0	1	39h
0	1	1	1	0	1	0	3Ah
0	1	1	1	0	1	1	3Bh
0	1	1	1	1	0	0	3Ch
0	1	1	1	1	0	1	3Dh
0	1	1	1	1	1	0	3Eh
0	1	1	1	1	1	1	3Fh

8. I/O programming

8.1 Quasi-bidirectional I/O architecture

The PCA8574/74A's 8 ports (see [Figure 2](#)) are entirely independent and can be used either as input or output ports. Input data is transferred from the ports to the microcontroller in the Read mode (see [Figure 9](#)). Output data is transmitted to the ports in the Write mode (see [Figure 8](#)).

This quasi-bidirectional I/O can be used as an input or output without the use of a control signal for data directions. At power-on the I/Os are HIGH. In this mode only a current source (I_{OH}) to V_{DD} is active. An additional strong pull-up to V_{DD} ($I_{trt(pu)}$) allows fast rising edges into heavily loaded outputs. These devices turn on when an output is written HIGH, and are switched off by the negative edge of SCL. The I/Os should be HIGH before being used as inputs. After power-on, as all the I/Os are set HIGH, all of them can be used as inputs. Any change in setting of the I/Os as either inputs or outputs can be done with the write mode.

Remark: If a HIGH is applied to an I/O which has been written earlier to LOW, a large current (I_{OL}) will flow to V_{SS} .

8.2 Writing to the port (Output mode)

To write, the master (microcontroller) first addresses the slave device. By setting the last bit of the byte containing the slave address to logic 0 the write mode is entered. The PCA8574/74A acknowledges and the master sends the data byte for P7 to P0 and is acknowledged by the PCA8574/74A. The 8-bit data is presented on the port lines after it has been acknowledged by the PCA8574/74A.

The number of data bytes that can be sent successively is not limited. The previous data is overwritten every time a data byte has been sent.

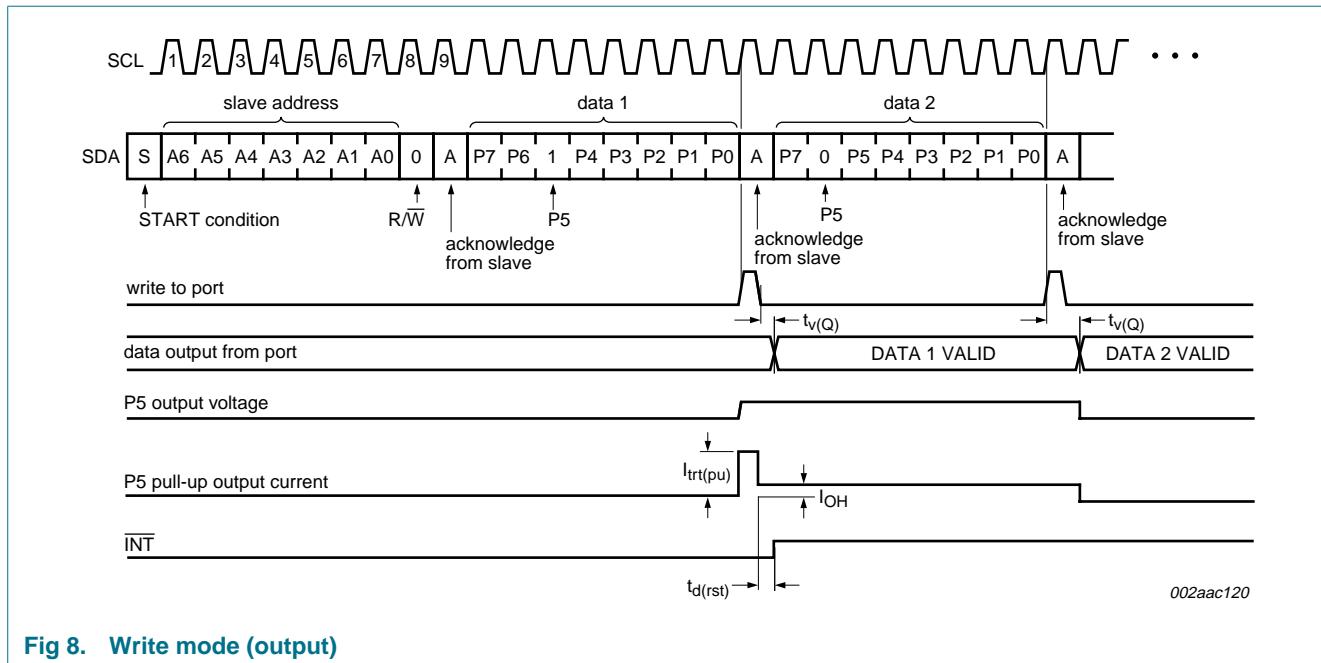
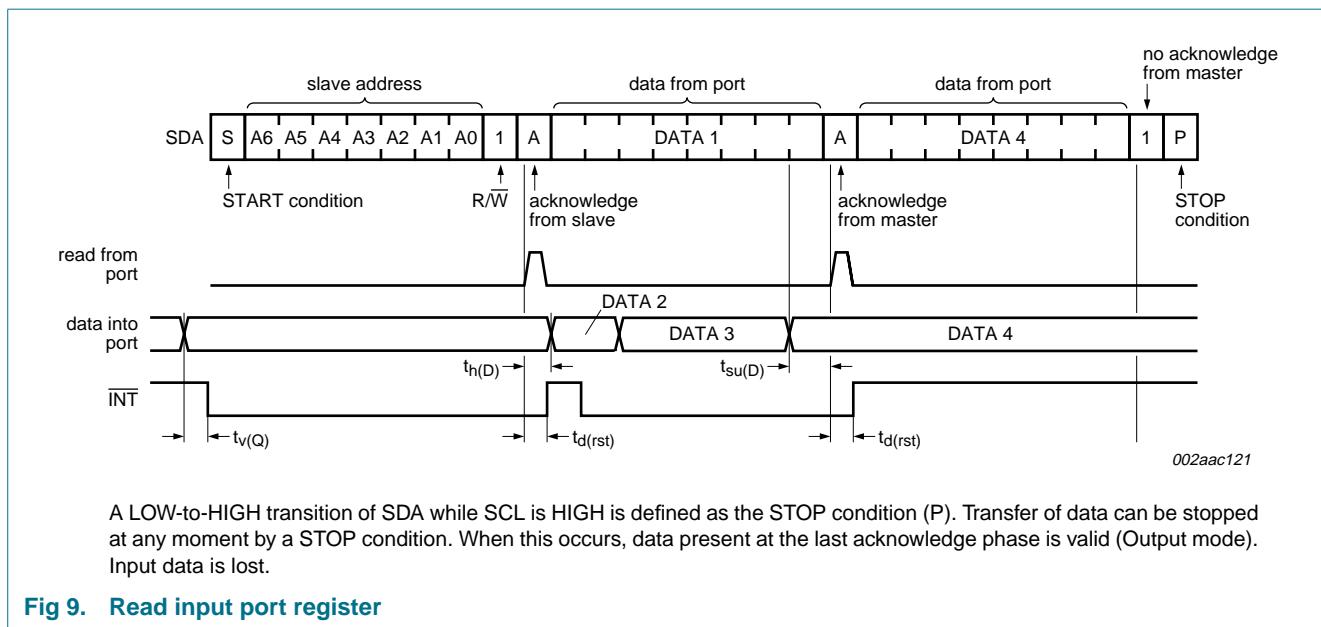


Fig 8. Write mode (output)

8.3 Reading from a port (Input mode)

All ports programmed as input should be set to logic 1. To read, the master (microcontroller) first addresses the slave device after it receives the interrupt. By setting the last bit of the byte containing the slave address to logic 1 the Read mode is entered. The data bytes that follow on the SDA are the values on the ports.

If the data on the input port changes faster than the master can read, this data may be lost.



A LOW-to-HIGH transition of SDA while SCL is HIGH is defined as the STOP condition (P). Transfer of data can be stopped at any moment by a STOP condition. When this occurs, data present at the last acknowledge phase is valid (Output mode). Input data is lost.

Fig 9. Read input port register

8.4 Power-on reset

When power is applied to V_{DD} , an internal Power-On Reset (POR) holds the PCA8574/74A in a reset condition until V_{DD} has reached V_{POR} . At that point, the reset condition is released and the PCA8574/74A registers and I²C-bus/SMBus state machine will initialize to their default states. Thereafter V_{DD} must be lowered below 0.2 V to reset the device.

8.5 Interrupt output (\overline{INT})

The PCA8574/74A provides an open-drain interrupt (\overline{INT}) which can be fed to a corresponding input of the microcontroller (see [Figure 8](#), [Figure 9](#), and [Figure 10](#)). This gives these chips a kind of master function which can initiate an action elsewhere in the system.

An interrupt is generated by any rising or falling edge of the port inputs. After time $t_{V(D)}$ the signal \overline{INT} is valid.

The interrupt disappears when data on the port is changed to the original setting or data is read from or written to the device which has generated the interrupt.

In the write mode, the interrupt may become deactivated (HIGH) on the rising edge of the write to port pulse. On the falling edge of the write to port pulse the interrupt is definitely deactivated (HIGH).

The interrupt is reset in the read mode on the rising edge of the read from port pulse.

During the resetting of the interrupt itself, any changes on the I/Os may not generate an interrupt. After the interrupt is reset any change in I/Os will be detected and transmitted as an \overline{INT} .

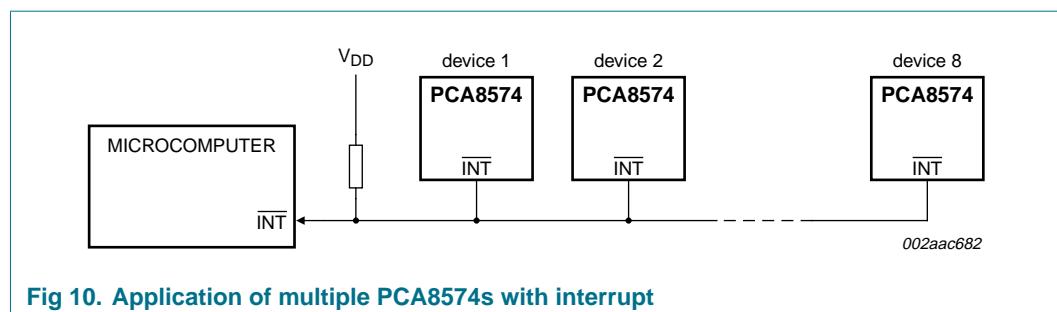


Fig 10. Application of multiple PCA8574s with interrupt

9. Characteristics of the I²C-bus

The I²C-bus is for 2-way, 2-line communication between different ICs or modules. The two lines are a serial data line (SDA) and a serial clock line (SCL). Both lines must be connected to a positive supply via a pull-up resistor when connected to the output stages of a device. Data transfer may be initiated only when the bus is not busy.

9.1 Bit transfer

One data bit is transferred during each clock pulse. The data on the SDA line must remain stable during the HIGH period of the clock pulse as changes in the data line at this time will be interpreted as control signals (see [Figure 11](#)).

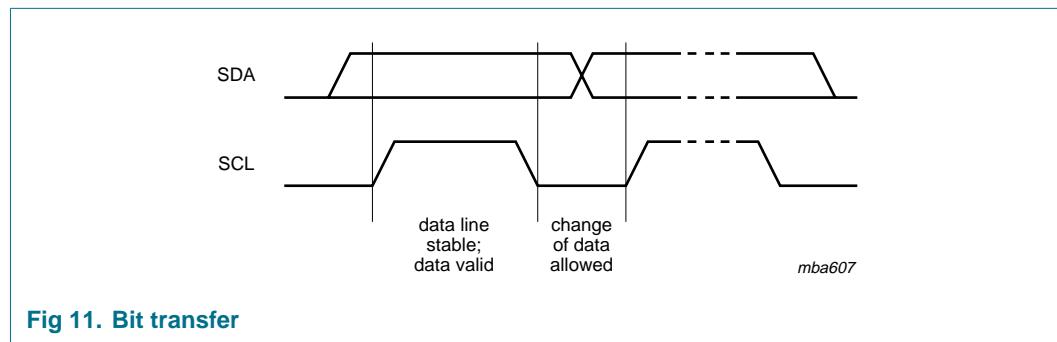


Fig 11. Bit transfer

9.1.1 START and STOP conditions

Both data and clock lines remain HIGH when the bus is not busy. A HIGH-to-LOW transition of the data line while the clock is HIGH is defined as the START condition (S). A LOW-to-HIGH transition of the data line while the clock is HIGH is defined as the STOP condition (P) (see [Figure 12](#).)

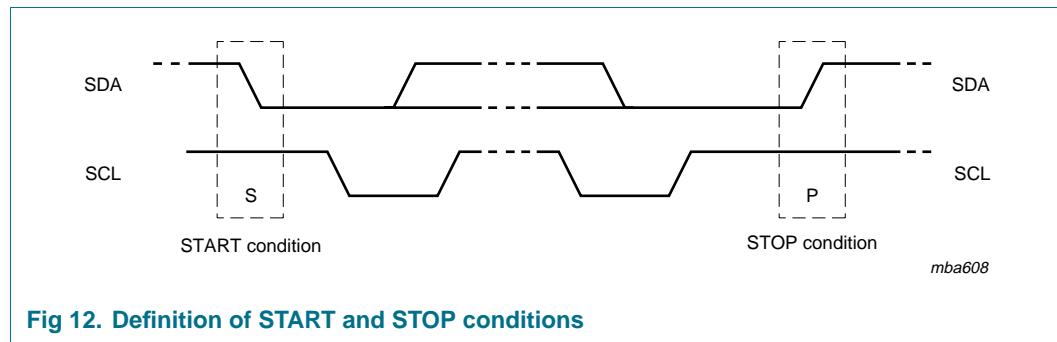


Fig 12. Definition of START and STOP conditions

9.2 System configuration

A device generating a message is a 'transmitter'; a device receiving is the 'receiver'. The device that controls the message is the 'master' and the devices which are controlled by the master are the 'slaves' (see [Figure 13](#)).

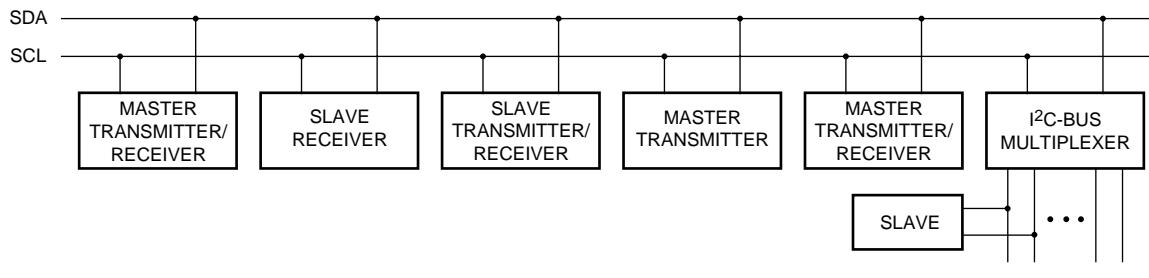


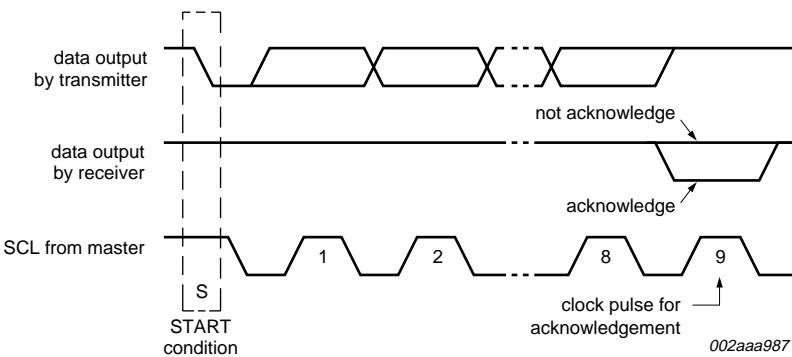
Fig 13. System configuration

9.3 Acknowledge

The number of data bytes transferred between the START and the STOP conditions from transmitter to receiver is not limited. Each byte of eight bits is followed by one acknowledge bit. The acknowledge bit is a HIGH level put on the bus by the transmitter, whereas the master generates an extra acknowledge related clock pulse.

A slave receiver which is addressed must generate an acknowledge after the reception of each byte. Also a master must generate an acknowledge after the reception of each byte that has been clocked out of the slave transmitter. The device that acknowledges has to pull down the SDA line during the acknowledge clock pulse, so that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse; set-up and hold times must be taken into account.

A master receiver must signal an end of data to the transmitter by not generating an acknowledge on the last byte that has been clocked out of the slave. In this event, the transmitter must leave the data line HIGH to enable the master to generate a STOP condition.

Fig 14. Acknowledgement on the I²C-bus

10. Application design-in information

10.1 Bidirectional I/O expander applications

In the 8-bit I/O expander application shown in [Figure 15](#), P0 and P1 are inputs, and P2 to P7 are outputs. When used in this configuration, during a write, the input (P0 and P1) must be written as HIGH so the external devices fully control the input ports. The desired HIGH or LOW logic levels may be written to the I/Os used as outputs (P2 to P7). During a read, the logic levels of the external devices driving the input ports (P0 and P1) and the previous written logic level to the output ports (P2 to P7) will be read.

The GPIO also has an interrupt line (\overline{INT}) that can be connected to the interrupt logic of the microprocessor. By sending an interrupt signal on this line, the remote I/O informs the microprocessor that there is incoming data or a change of data on its ports without having to communicate via the I²C-bus.

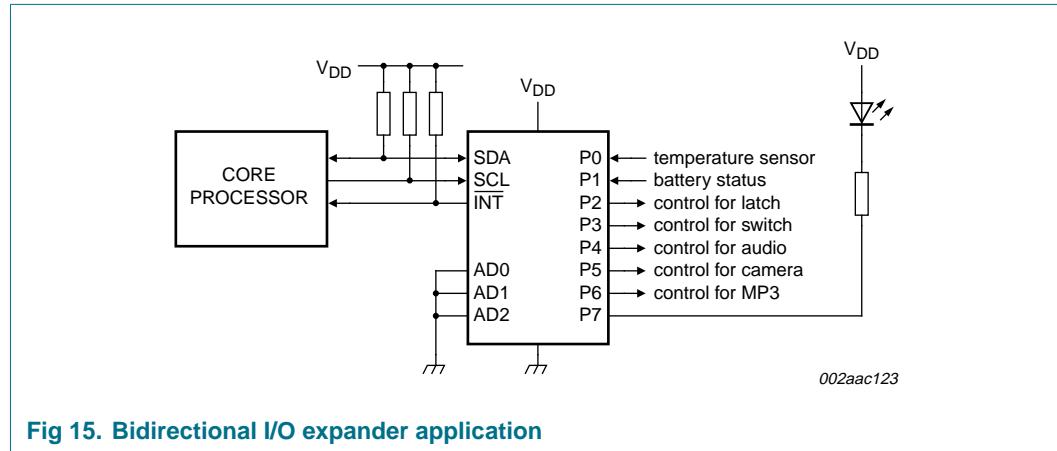


Fig 15. Bidirectional I/O expander application

10.2 High current-drive load applications

The GPIO has a maximum sinking current of 25 mA per bit. In applications requiring additional drive, two port pins in the same octal may be connected together to sink up to 50 mA current. Both bits must then always be turned on or off together. Up to 8 pins (one octal) can be connected together to drive 200 mA.

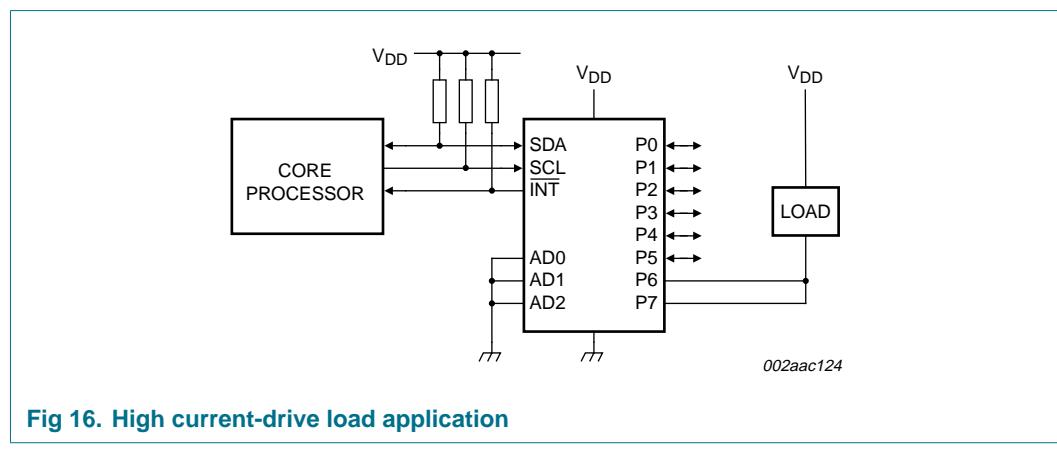


Fig 16. High current-drive load application

11. Limiting values

Table 6. Limiting values*In accordance with the Absolute Maximum Rating System (IEC 60134).*

Symbol	Parameter	Conditions	Min	Max	Unit
V _{DD}	supply voltage		-0.5	+6	V
I _{DD}	supply current		-	±100	mA
I _{SS}	ground supply current		-	±400	mA
V _I	input voltage		V _{SS} - 0.5	5.5	V
I _I	input current		-	±20	mA
I _O	output current	[1]	-	±50	mA
P _{tot}	total power dissipation		-	400	mW
P/out	power dissipation per output		-	100	mW
T _{stg}	storage temperature		-65	+150	°C
T _{amb}	ambient temperature	operating	-40	+85	°C

[1] Total package (maximum) output current is 400 mA.

12. Static characteristics

Table 7. Static characteristics

$V_{DD} = 2.3\text{ V}$ to 5.5 V ; $V_{SS} = 0\text{ V}$; $T_{amb} = -40^\circ\text{C}$ to $+85^\circ\text{C}$; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Supplies						
V_{DD}	supply voltage		2.3	-	5.5	V
I_{DD}	supply current	Operating mode; no load; $V_I = V_{DD}$ or V_{SS} ; $f_{SCL} = 400\text{ kHz}$; AD0, AD1, AD2 = static H or L	-	200	500	μA
I_{stb}	standby current	Standby mode; no load; $V_I = V_{DD}$ or V_{SS} ; $f_{SCL} = 0\text{ kHz}$	-	4.5	10	μA
V_{POR}	power-on reset voltage	[1]	-	1.8	2.0	V
Input SCL; input/output SDA						
V_{IL}	LOW-level input voltage		-0.5	-	$+0.3V_{DD}$	V
V_{IH}	HIGH-level input voltage		$0.7V_{DD}$	-	5.5	V
I_{OL}	LOW-level output current	$V_{OL} = 0.4\text{ V}$; $V_{DD} = 2.3\text{ V}$	20	35	-	mA
		$V_{OL} = 0.4\text{ V}$; $V_{DD} = 3.0\text{ V}$	25	44	-	mA
		$V_{OL} = 0.4\text{ V}$; $V_{DD} = 4.5\text{ V}$	30	57	-	mA
I_L	leakage current	$V_I = V_{DD}$ or V_{SS}	-1	-	+1	μA
C_i	input capacitance	$V_I = V_{SS}$	-	5	10	pF
I/Os; P0 to P7						
I_{OL}	LOW-level output current	$V_{OL} = 0.5\text{ V}$; $V_{DD} = 2.3\text{ V}$	[2] 12	26	-	mA
		$V_{OL} = 0.5\text{ V}$; $V_{DD} = 3.0\text{ V}$	[2] 17	33	-	mA
		$V_{OL} = 0.5\text{ V}$; $V_{DD} = 4.5\text{ V}$	[2] 25	40	-	mA
$I_{OL(tot)}$	total LOW-level output current	$V_{OL} = 0.5\text{ V}$; $V_{DD} = 4.5\text{ V}$	[2] -	-	200	mA
I_{OH}	HIGH-level output current	$V_{OH} = V_{SS}$	-30	-138	-300	μA
$I_{trt(pu)}$	transient boosted pull-up current	$V_{OH} = V_{SS}$; see Figure 8	-0.5	-1.0	-	mA
C_i	input capacitance	[3]	-	2.1	10	pF
C_o	output capacitance	[3]	-	2.1	10	pF
Interrupt INT (see Figure 8 and Figure 9)						
I_{OL}	LOW-level output current	$V_{OL} = 0.4\text{ V}$	3.0	-	-	mA
C_o	output capacitance		-	3	5	pF
Inputs AD0, AD1, AD2						
V_{IL}	LOW-level input voltage		-0.5	-	$+0.3V_{DD}$	V
V_{IH}	HIGH-level input voltage		$0.7V_{DD}$	-	5.5	V
I_{LI}	input leakage current		-1	-	+1	μA
C_i	input capacitance		-	3.5	5	pF

[1] The power-on reset circuit resets the I²C-bus logic with $V_{DD} < V_{POR}$ and sets all I/Os to logic 1 (with current source to V_{DD}).

[2] Each bit must be limited to a maximum of 25 mA and the total package limited to 200 mA due to internal busing limits.

[3] The value is not tested, but verified on sampling basis.

13. Dynamic characteristics

Table 8. Dynamic characteristics

$V_{DD} = 2.3\text{ V}$ to 5.5 V ; $V_{SS} = 0\text{ V}$; $T_{amb} = -40^\circ\text{C}$ to $+85^\circ\text{C}$; unless otherwise specified. Limits are for Fast-mode I²C-bus.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{SCL}	SCL clock frequency		0	-	400	kHz
t_{BUF}	bus free time between a STOP and START condition		1.3	-	-	μs
$t_{HD;STA}$	hold time (repeated) START condition		0.6	-	-	μs
$t_{SU;STA}$	set-up time for a repeated START condition		0.6	-	-	μs
$t_{SU;STO}$	set-up time for STOP condition		0.6	-	-	μs
$t_{HD;DAT}$	data hold time		0	-	-	ns
$t_{VD;ACK}$	data valid acknowledge time ^[1]		0.1	-	0.9	μs
$t_{VD;DAT}$	data valid time ^[2]		50	-	-	ns
$t_{SU;DAT}$	data set-up time		100	-	-	ns
t_{LOW}	LOW period of the SCL clock		1.3	-	-	μs
t_{HIGH}	HIGH period of the SCL clock		0.6	-	-	μs
t_f	fall time of both SDA and SCL signals	^{[3][4]}	$20 + 0.1C_b$ ^[5]	-	300	ns
t_r	rise time of both SDA and SCL signals		$20 + 0.1C_b$ ^[5]	-	300	ns
t_{SP}	pulse width of spikes that must be suppressed by the input filter ^[6]		-	-	50	ns

Port timing; $C_L \leq 100\text{ pF}$ (see Figure 8 and Figure 9)

$t_{V(Q)}$	data output valid time	-	-	4	μs
$t_{SU(D)}$	data input set-up time	0	-	-	μs
$t_{H(D)}$	data input hold time	4	-	-	μs
Interrupt timing; $C_L \leq 100\text{ pF}$ (see Figure 8 and Figure 9)					
$t_{V(D)}$	data input valid time	-	-	4	μs
$t_{D(rst)}$	reset delay time	-	-	4	μs

[1] $t_{VD;ACK}$ = time for Acknowledgement signal from SCL LOW to SDA (out) LOW.

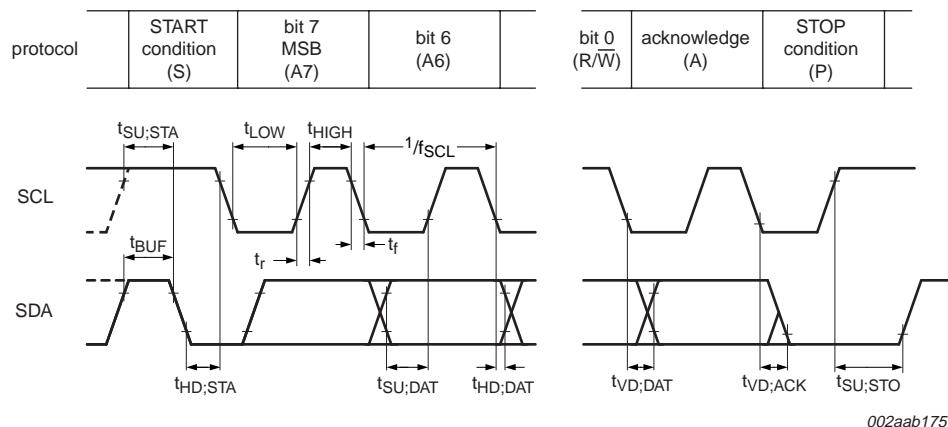
[2] $t_{VD;DAT}$ = minimum time for SDA data out to be valid following SCL LOW.

[3] A master device must internally provide a hold time of at least 300 ns for the SDA signal (refer to the V_{IL} of the SCL signal) in order to bridge the undefined region SCL's falling edge.

[4] The maximum t_f for the SDA and SCL bus lines is specified at 300 ns. The maximum fall time for the SDA output stage t_f is specified at 250 ns. This allows series protection resistors to be connected between the SDA and the SCL pins and the SDA/SCL bus lines without exceeding the maximum specified t_f .

[5] C_b = total capacitance of one bus line in pF.

[6] Input filters on the SDA and SCL inputs suppress noise spikes less than 50 ns.

**Fig 17. I²C-bus timing diagram**

14. Package outline

DIP16: plastic dual in-line package; 16 leads (300 mil); long body

SOT38-1

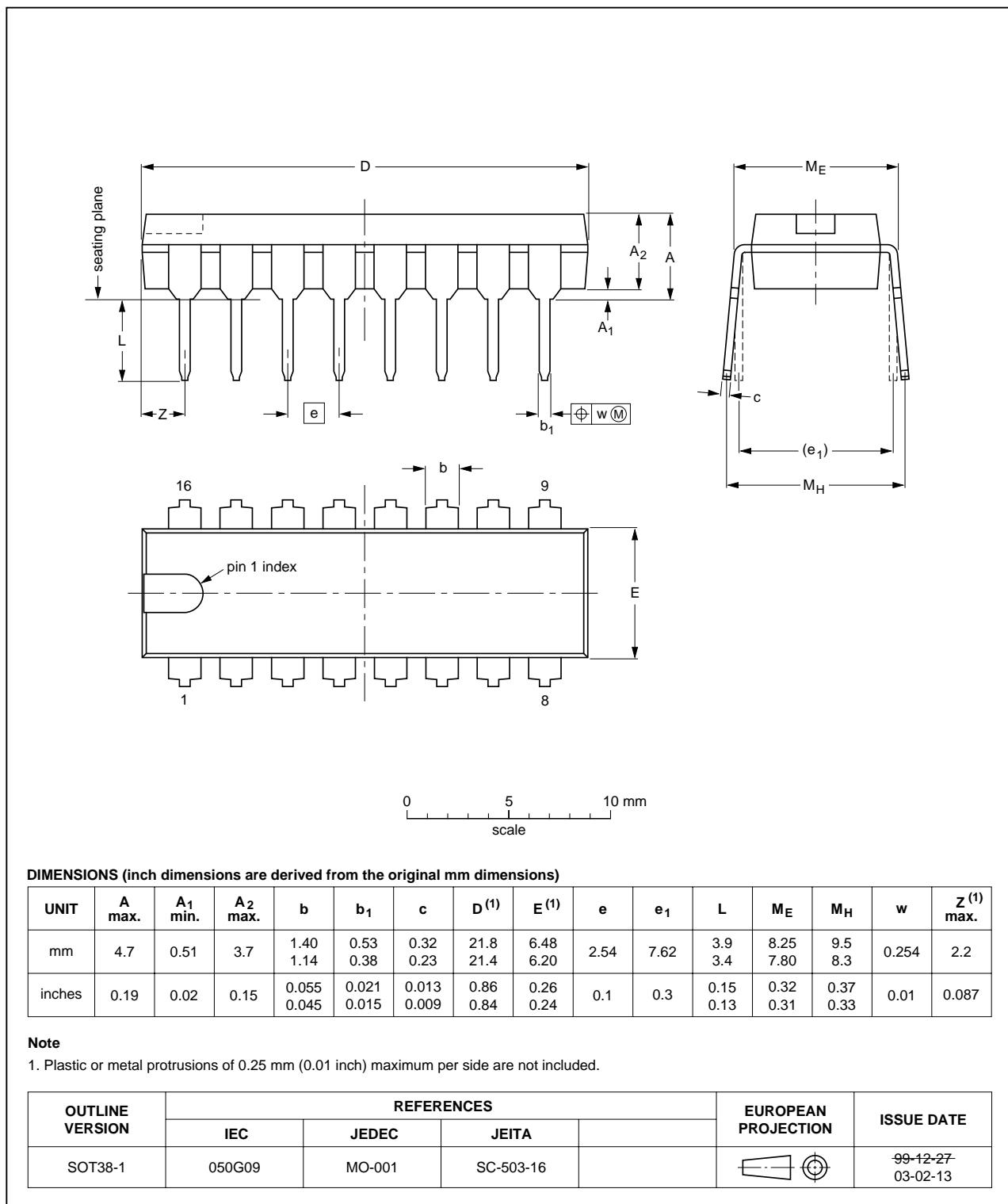


Fig 18. Package outline SOT38-1 (DIP16)

SO16: plastic small outline package; 16 leads; body width 7.5 mm

SOT162-1

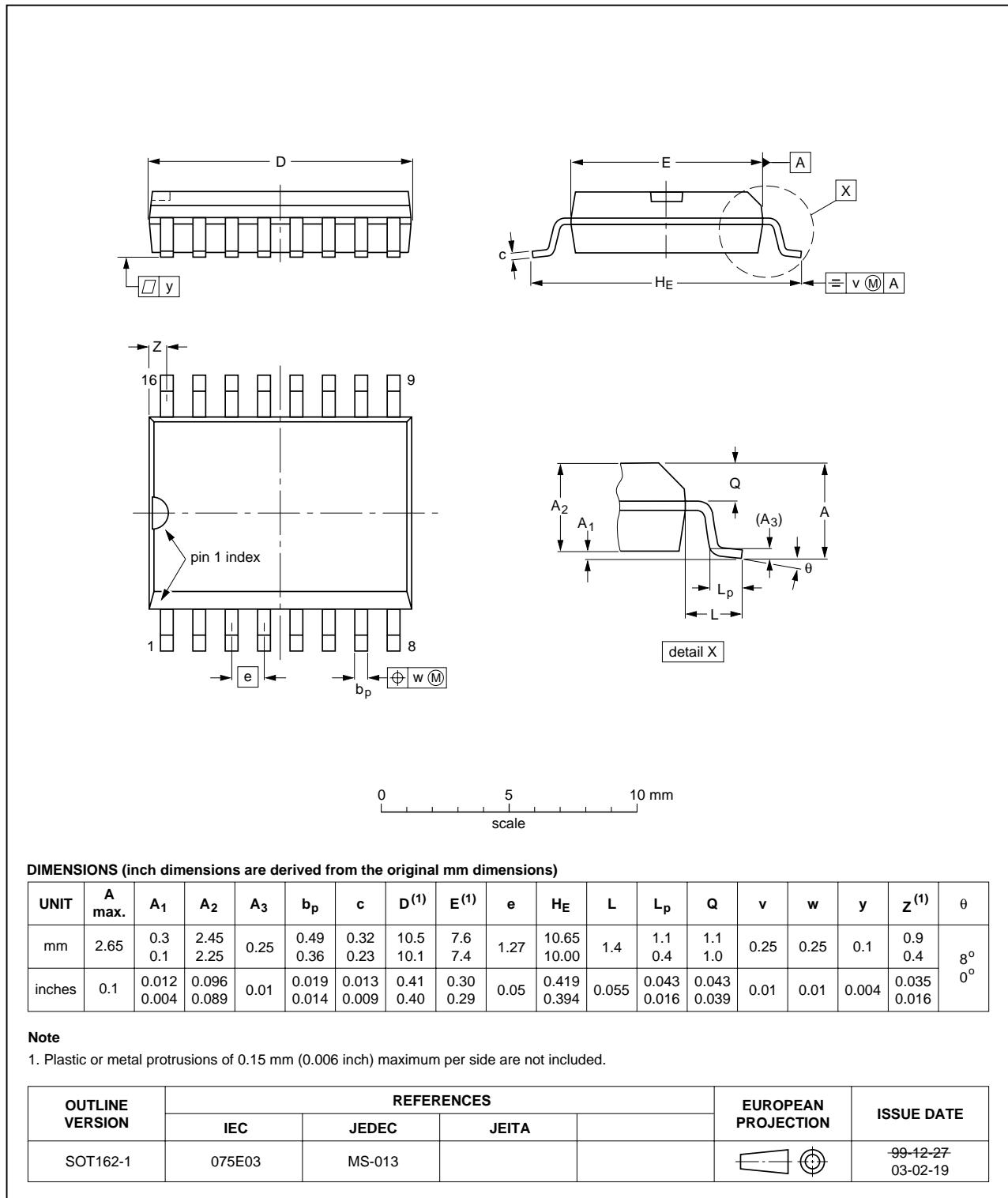


Fig 19. Package outline SOT162-1 (SO16)

TSSOP16: plastic thin shrink small outline package; 16 leads; body width 4.4 mm

SOT403-1

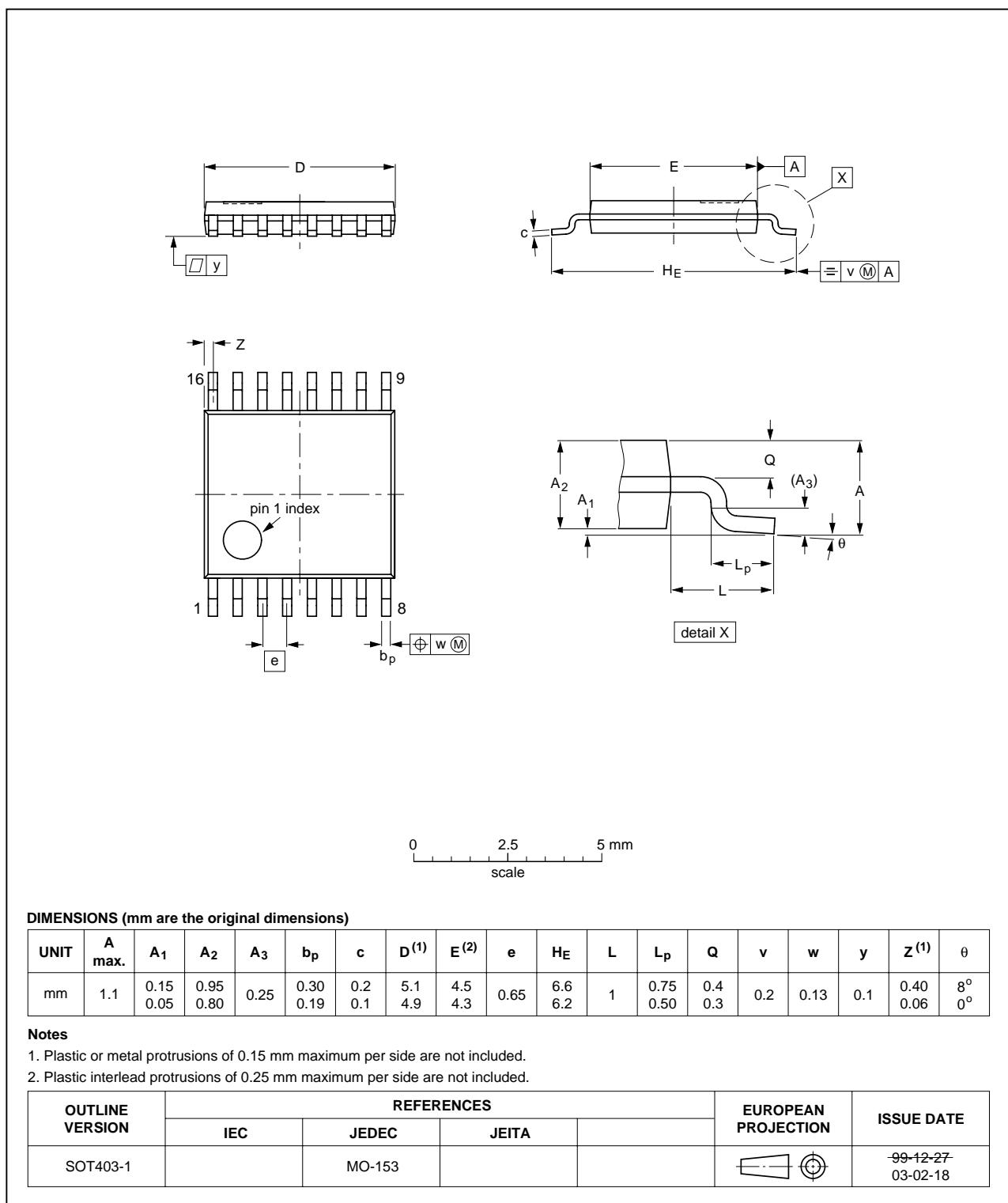


Fig 20. Package outline SOT403-1 (TSSOP16)

SSOP20: plastic shrink small outline package; 20 leads; body width 4.4 mm

SOT266-1

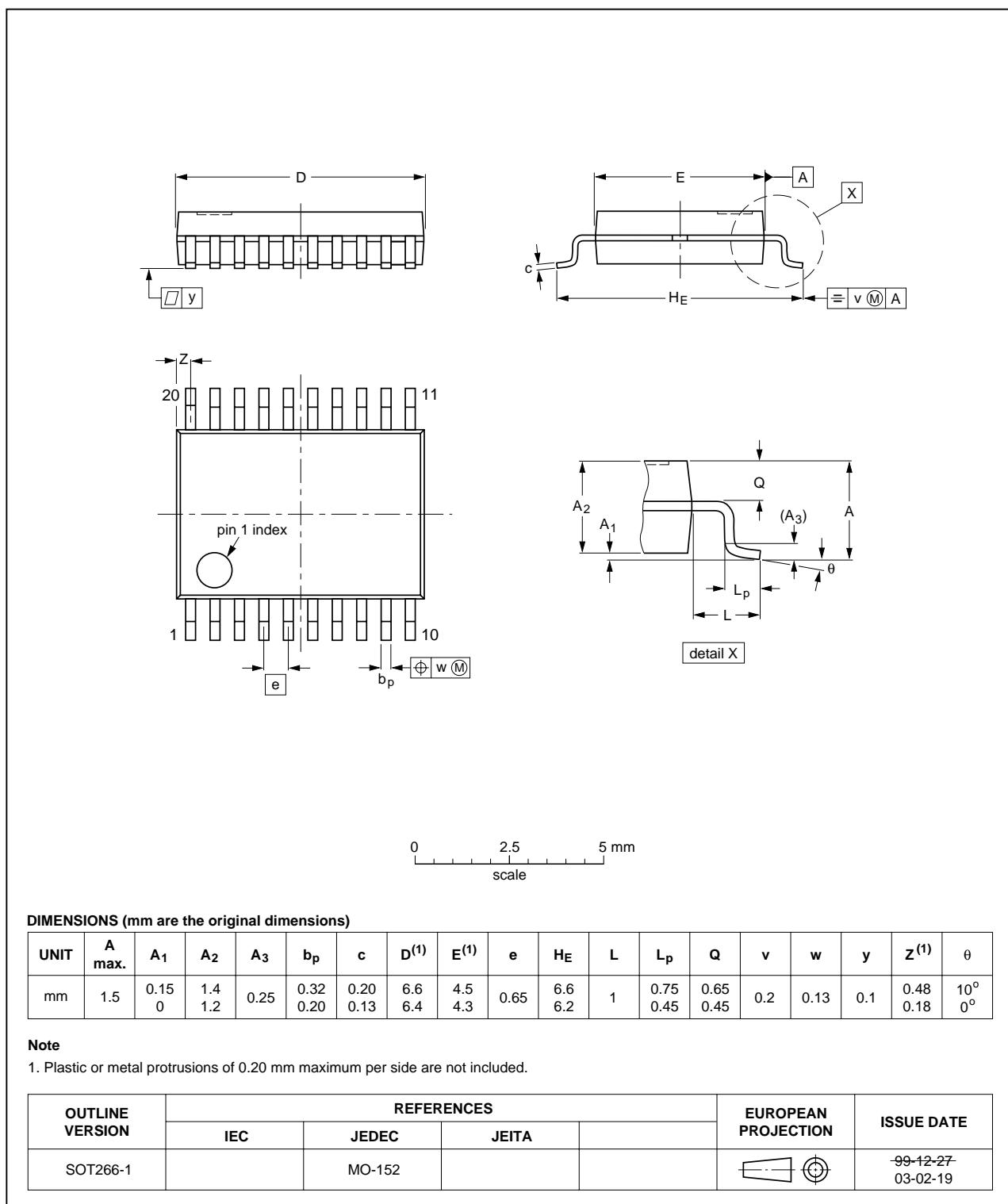


Fig 21. Package outline SOT266-1 (SSOP20)

15. Handling information

Inputs and outputs are protected against electrostatic discharge in normal handling. However, to be completely safe you must take normal precautions appropriate to handling integrated circuits.

16. Soldering

16.1 Introduction

There is no soldering method that is ideal for all surface mount IC packages. Wave soldering can still be used for certain surface mount ICs, but it is not suitable for fine pitch SMDs. In these situations reflow soldering is recommended.

16.2 Through-hole mount packages

16.2.1 Soldering by dipping or by solder wave

Typical dwell time of the leads in the wave ranges from 3 seconds to 4 seconds at 250 °C or 265 °C, depending on solder material applied, SnPb or Pb-free respectively.

The total contact time of successive solder waves must not exceed 5 seconds.

The device may be mounted up to the seating plane, but the temperature of the plastic body must not exceed the specified maximum storage temperature ($T_{stg(max)}$). If the printed-circuit board has been pre-heated, forced cooling may be necessary immediately after soldering to keep the temperature within the permissible limit.

16.2.2 Manual soldering

Apply the soldering iron (24 V or less) to the lead(s) of the package, either below the seating plane or not more than 2 mm above it. If the temperature of the soldering iron bit is less than 300 °C it may remain in contact for up to 10 seconds. If the bit temperature is between 300 °C and 400 °C, contact may be up to 5 seconds.

16.3 Surface mount packages

16.3.1 Reflow soldering

Key characteristics in reflow soldering are:

- Lead-free versus SnPb soldering; note that a lead-free reflow process usually leads to higher minimum peak temperatures (see [Figure 22](#)) than a PbSn process, thus reducing the process window
- Solder paste printing issues including smearing, release, and adjusting the process window for a mix of large and small components on one board
- Reflow temperature profile; this profile includes preheat, reflow (in which the board is heated to the peak temperature) and cooling down. It is imperative that the peak temperature is high enough for the solder to make reliable solder joints (a solder paste characteristic). In addition, the peak temperature must be low enough that the

packages and/or boards are not damaged. The peak temperature of the package depends on package thickness and volume and is classified in accordance with [Table 9](#) and [10](#)

Table 9. SnPb eutectic process (from J-STD-020C)

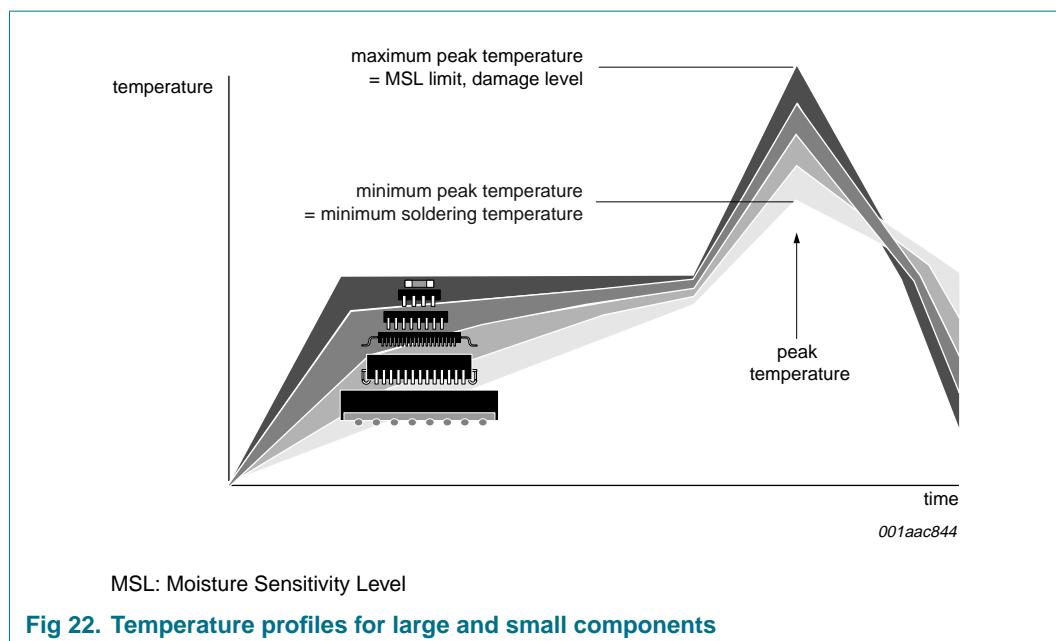
Package thickness (mm)	Package reflow temperature (°C)	
	Volume (mm ³)	
	< 350	≥ 350
< 2.5	235	220
≥ 2.5	220	220

Table 10. Lead-free process (from J-STD-020C)

Package thickness (mm)	Package reflow temperature (°C)		
	Volume (mm ³)		
	< 350	350 to 2000	> 2000
< 1.6	260	260	260
1.6 to 2.5	260	250	245
> 2.5	250	245	245

Moisture sensitivity precautions, as indicated on the packing, must be respected at all times.

Studies have shown that small packages reach higher temperatures during reflow soldering, see [Figure 22](#).



For further information on temperature profiles, refer to Application Note AN10365 "Surface mount reflow soldering description".

16.3.2 Wave soldering

Conventional single wave soldering is not recommended for surface mount devices (SMDs) or printed-circuit boards with a high component density, as solder bridging and non-wetting can present major problems.

To overcome these problems the double-wave soldering method was specifically developed.

If wave soldering is used the following conditions must be observed for optimal results:

- Use a double-wave soldering method comprising a turbulent wave with high upward pressure followed by a smooth laminar wave.
- For packages with leads on two sides and a pitch (e):
 - larger than or equal to 1.27 mm, the footprint longitudinal axis is **preferred** to be parallel to the transport direction of the printed-circuit board;
 - smaller than 1.27 mm, the footprint longitudinal axis **must** be parallel to the transport direction of the printed-circuit board.
- The footprint must incorporate solder thieves at the downstream end.
- For packages with leads on four sides, the footprint must be placed at a 45° angle to the transport direction of the printed-circuit board. The footprint must incorporate solder thieves downstream and at the side corners.

During placement and before soldering, the package must be fixed with a droplet of adhesive. The adhesive can be applied by screen printing, pin transfer or syringe dispensing. The package can be soldered after the adhesive is cured.

Typical dwell time of the leads in the wave ranges from 3 seconds to 4 seconds at 250 °C or 265 °C, depending on solder material applied, SnPb or Pb-free respectively.

A mildly-activated flux will eliminate the need for removal of corrosive residues in most applications.

16.3.3 Manual soldering

Fix the component by first soldering two diagonally-opposite end leads. Use a low voltage (24 V or less) soldering iron applied to the flat part of the lead. Contact time must be limited to 10 seconds at up to 300 °C.

When using a dedicated tool, all other leads can be soldered in one operation within 2 seconds to 5 seconds between 270 °C and 320 °C.

16.4 Package related soldering information

Table 11. Suitability of IC packages for wave, reflow and dipping soldering methods

Mounting	Package ^[1]	Soldering method		
		Wave	Reflow ^[2]	Dipping
Through-hole mount	CPGA, HCPGA	suitable	–	–
	DBS, DIP, HDIP, RDBS, SDIP, SIL	suitable ^[3]	–	suitable
Through-hole-surface mount	PMFP ^[4]	not suitable	not suitable	–

Table 11. Suitability of IC packages for wave, reflow and dipping soldering methods ...continued

Mounting	Package ^[1]	Soldering method		
		Wave	Reflow ^[2]	Dipping
Surface mount	BGA, HTSSON..T ^[5] , LBGA, LFBGA, SQFP, SSOP..T ^[5] , TFBGA, VFBGA, XSON	not suitable	suitable	–
	DHVQFN, HBCC, HBGA, HLQFP, HSO, HSOP, HSQFP, HSSON, HTQFP, HTSSOP, HVQFN, HVSON, SMS	not suitable ^[6]	suitable	–
	PLCC ^[7] , SO, SOJ	suitable	suitable	–
	LQFP, QFP, TQFP	not recommended ^{[7][8]}	suitable	–
	SSOP, TSSOP, VSO, VSSOP	not recommended ^[9]	suitable	–
	CWQCCN..L ^[10] , WQCCN..L ^[10]	not suitable	not suitable	–

- [1] For more detailed information on the BGA packages refer to the *(LF)BGA Application Note* (AN01026); order a copy from your NXP Semiconductors sales office.
- [2] All surface mount (SMD) packages are moisture sensitive. Depending upon the moisture content, the maximum temperature (with respect to time) and body size of the package, there is a risk that internal or external package cracks may occur due to vaporization of the moisture in them (the so called popcorn effect).
- [3] For SDIP packages, the longitudinal axis must be parallel to the transport direction of the printed-circuit board.
- [4] Hot bar soldering or manual soldering is suitable for PMFP packages.
- [5] These transparent plastic packages are extremely sensitive to reflow soldering conditions and must on no account be processed through more than one soldering cycle or subjected to infrared reflow soldering with peak temperature exceeding $217^{\circ}\text{C} \pm 10^{\circ}\text{C}$ measured in the atmosphere of the reflow oven. The package body peak temperature must be kept as low as possible.
- [6] These packages are not suitable for wave soldering. On versions with the heatsink on the bottom side, the solder cannot penetrate between the printed-circuit board and the heatsink. On versions with the heatsink on the top side, the solder might be deposited on the heatsink surface.
- [7] If wave soldering is considered, then the package must be placed at a 45° angle to the solder wave direction. The package footprint must incorporate solder thieves downstream and at the side corners.
- [8] Wave soldering is suitable for LQFP, QFP and TQFP packages with a pitch (e) larger than 0.8 mm; it is definitely not suitable for packages with a pitch (e) equal to or smaller than 0.65 mm.
- [9] Wave soldering is suitable for SSOP, TSSOP, VSO and VSSOP packages with a pitch (e) equal to or larger than 0.65 mm; it is definitely not suitable for packages with a pitch (e) equal to or smaller than 0.5 mm.
- [10] Image sensor packages in principle should not be soldered. They are mounted in sockets or delivered pre-mounted on flex foil. However, the image sensor package can be mounted by the client on a flex foil by using a hot bar soldering process. The appropriate soldering profile can be provided on request.

17. Abbreviations

Table 12. Abbreviations

Acronym	Description
CDM	Charged Device Model
CMOS	Complementary Metal Oxide Semiconductor
ESD	ElectroStatic Discharge
GPIO	General Purpose Input/Output
HBM	Human Body Model
LED	Light Emitting Diode
IC	Integrated Circuit
I ² C-bus	Inter-Integrated Circuit bus
ID	Identification
LSB	Least Significant Bit
MM	Machine Model
MSB	Most Significant Bit
PLC	Programmable Logic Controller
PWM	Pulse Width Modulation
RAID	Redundant Array of Independent Disks
SMBus	System Management Bus

18. Revision history

Table 13. Revision history

Document ID	Release date	Data sheet status	Change notice	Supersedes
PCA8574_PCA8574A_2	20070514	Product data sheet	-	PCA8574_PCA8574A_1
Modifications:			<ul style="list-style-type: none"> • Section 2 "Features", last bullet item: changed "TSSOP20" to "TSSOP16" • Table 1 "Ordering information": changed package from TSSOP20 (SOT360-1) to TSSOP16 (SOT403-1) • Section 6.1 "Pinning": deleted pin configuration for TSSOP20; added pin configuration for TSSOP16 • Table 2 title changed (added TSSOP16) • Table 3 title changed (deleted TSSOP20) • Section 14 "Package outline": changed package from TSSOP20 (SOT360-1) to TSSOP16 (SOT403-1) 	
PCA8574_PCA8574A_1	20070117	Product data sheet	-	-

19. Legal information

19.1 Data sheet status

Document status ^{[1][2]}	Product status ^[3]	Definition
Objective [short] data sheet	Development	This document contains data from the objective specification for product development.
Preliminary [short] data sheet	Qualification	This document contains data from the preliminary specification.
Product [short] data sheet	Production	This document contains the product specification.

[1] Please consult the most recently issued document before initiating or completing a design.

[2] The term 'short data sheet' is explained in section "Definitions".

[3] The product status of device(s) described in this document may have changed since this document was published and may differ in case of multiple devices. The latest product status information is available on the Internet at URL <http://www.nxp.com>.

19.2 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

Short data sheet — A short data sheet is an extract from a full data sheet with the same product type number(s) and title. A short data sheet is intended for quick reference only and should not be relied upon to contain detailed and full information. For detailed and full information see the relevant full data sheet, which is available on request via the local NXP Semiconductors sales office. In case of any inconsistency or conflict with the short data sheet, the full data sheet shall prevail.

19.3 Disclaimers

General — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in medical, military, aircraft, space or life support equipment, nor in applications where failure or malfunction of a NXP Semiconductors product can reasonably be expected to

result in personal injury, death or severe property or environmental damage. NXP Semiconductors accepts no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Limiting values — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) may cause permanent damage to the device. Limiting values are stress ratings only and operation of the device at these or any other conditions above those given in the Characteristics sections of this document is not implied. Exposure to limiting values for extended periods may affect device reliability.

Terms and conditions of sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at <http://www.nxp.com/profile/terms>, including those pertaining to warranty, intellectual property rights infringement and limitation of liability, unless explicitly otherwise agreed to in writing by NXP Semiconductors. In case of any inconsistency or conflict between information in this document and such terms and conditions, the latter will prevail.

No offer to sell or license — Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

19.4 Trademarks

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

I²C-bus — logo is a trademark of NXP B.V.

20. Contact information

For additional information, please visit: <http://www.nxp.com>

For sales office addresses, send an email to: salesaddresses@nxp.com

21. Contents

1	General description	1	19	Legal information	26
2	Features	1	19.1	Data sheet status	26
3	Applications	1	19.2	Definitions	26
4	Ordering information	2	19.3	Disclaimers	26
5	Block diagram	2	19.4	Trademarks	26
6	Pinning information	3	20	Contact information	26
6.1	Pinning	3	21	Contents	27
6.2	Pin description	4			
7	Functional description	6			
7.1	Device address	6			
7.1.1	Address maps	6			
8	I/O programming	7			
8.1	Quasi-bidirectional I/O architecture	7			
8.2	Writing to the port (Output mode)	7			
8.3	Reading from a port (Input mode)	8			
8.4	Power-on reset	9			
8.5	Interrupt output (\overline{INT})	9			
9	Characteristics of the I²C-bus	10			
9.1	Bit transfer	10			
9.1.1	START and STOP conditions	10			
9.2	System configuration	10			
9.3	Acknowledge	11			
10	Application design-in information	12			
10.1	Bidirectional I/O expander applications	12			
10.2	High current-drive load applications	12			
11	Limiting values	13			
12	Static characteristics	14			
13	Dynamic characteristics	15			
14	Package outline	17			
15	Handling information	21			
16	Soldering	21			
16.1	Introduction	21			
16.2	Through-hole mount packages	21			
16.2.1	Soldering by dipping or by solder wave	21			
16.2.2	Manual soldering	21			
16.3	Surface mount packages	21			
16.3.1	Reflow soldering	21			
16.3.2	Wave soldering	23			
16.3.3	Manual soldering	23			
16.4	Package related soldering information	23			
17	Abbreviations	25			
18	Revision history	25			

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

founded by

PHILIPS

© NXP B.V. 2007.

All rights reserved.

For more information, please visit: <http://www.nxp.com>
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 14 May 2007

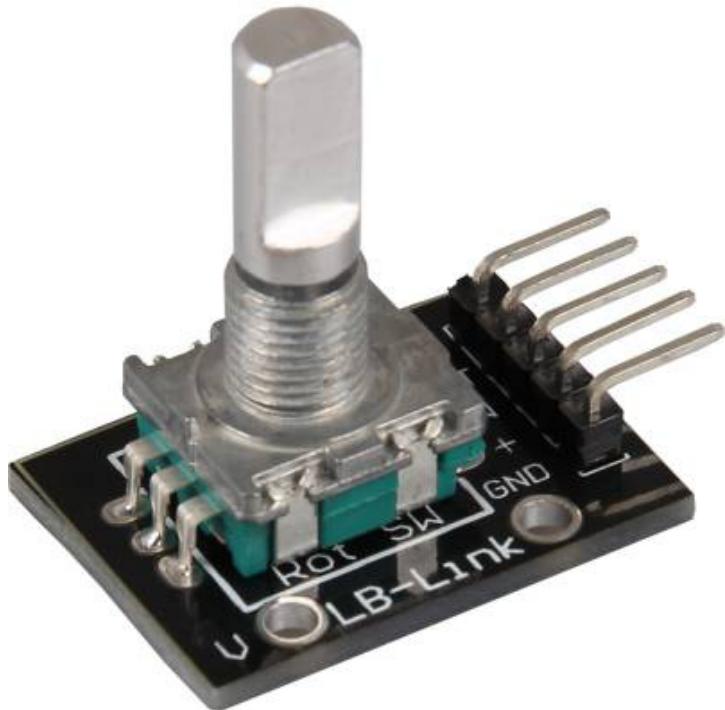
Document identifier: PCA8574_PCA8574A_2

KY-040 Rotary encoder

Contents

1 Picture	1
2 Technical data / Short description	1
3 Encoding	1
4 Pinout	2
5 Code example Arduino	3
6 Code example Raspberry Pi	4

Picture



Technical data / Short description

The current position of the rotary switch will be send encoded to the output.

Encoding

The idea of the rotary switch is that with every "step" only one of the conditions will change. In order of which status have changed first, you can see the rotational direction if you look at the following encoding.

Clockwise [A will change first] -> Pin_CLK

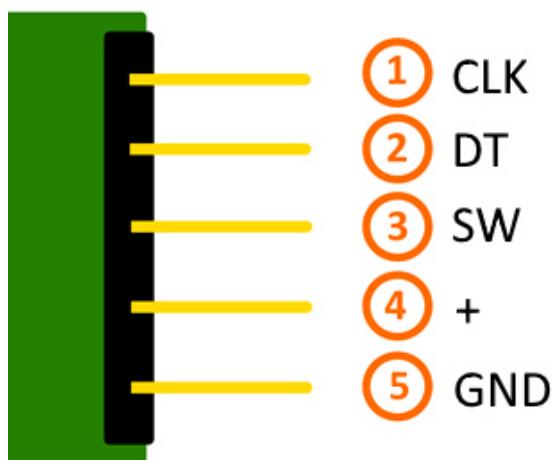
KY-040 Rotary encoder

A	B
0	0
1	0
1	1
0	1
0	0

Clockwise [B will change first] -> Pin_DT

A	B
0	0
0	1
1	1
1	0
0	0

Pinout



Code example Arduino

The program checks which pin has changed first, to get the rotational direction, after detecting a change at the pin status. You will get the rotational direction after you compare the current status of the pins with the last status. After detecting the rotational direction, the steps from the start will be counted and outputted. Pushing the button of the rotary encoder will reset the current position.

For serial output: Baudrate = 115200

```
// Initialization of the needed variables<br />
int Counter = 0;
boolean Richtung;
int Pin_clk_Letzter;
int Pin_clk_Aktuell;

// Definition of the input-pins
int pin_clk = 3;
int pin_dt = 4;
int button_pin = 5;

void setup()
{
    // Initialization of the input-pins...
    pinMode (pin_clk,INPUT);
    pinMode (pin_dt,INPUT);
    pinMode (button_pin,INPUT);

    // ...and activating of their pull up resistors
    digitalWrite(pin_clk, true);
    digitalWrite(pin_dt, true);
    digitalWrite(button_pin, true);

    // Initial reading of the Pin_CLK
    Pin_clk_Letzter = digitalRead(pin_clk);
    Serial.begin (115200);
}

// The program checks, which of the status pins have changed first
void loop()
{
    // Reading of the current status
    Pin_clk_Aktuell = digitalRead(pin_clk);

    // Check for a Change
    if (Pin_clk_Aktuell != Pin_clk_Letzter)
    {

        if (digitalRead(pin_dt) != Pin_clk_Aktuell)
        {
            // Pin_CLK has changed first
            Counter++;
            Richtung = true;
        }

        else
        {
            // Else Pin_DT changed first
            Richtung = false;
            Counter--;
        }

        Serial.println ("Rotation detected: ");
    }
}
```

KY-040 Rotary encoder

```
Serial.println ("Rotation detected: ");
Serial.print ("Rotational direction: ");

if (Richtung)
{
    Serial.println ("Clockwise");
}
else
{
    Serial.println("Counterclockwise");
}

Serial.print("Current position: ");
Serial.println(Counter);
Serial.println("-----");

}

// Preparation for the next run:
// The current value will be the last value for the next run.
Pin_clk_Letzter = Pin_clk_Aktuell;

// Reset function to save the current position
if (!digitalRead(button_pin) && Counter!=0)
{
    Counter = 0;
    Serial.println("Position resetted");
}

}
```

Connections Arduino:

CLK	= [Pin 3]
DT	= [Pin 4]
Button	= [Pin 5]
+	= [Pin 5V]
GND	= [Pin GND]

Code example download[KY-40_rotary-encoder_ARD](#)**Code example Raspberry Pi**

The program checks which pin has changed first, to get the rotational direction, after detecting a change at the pin status. You will get the rotational direction after you compare the current status of the pins with the last status. After detecting the rotational direction, the steps from the start will be counted and outputted. Pushing the button of the rotary encoder will reset the current position.

```
# coding=utf-8
# Needed modules will be imported and configured
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)

# Declaration and initialisation of the input pins which are connected with the sensor.
PIN_CLK = 16
```

KY-040 Rotary encoder

```
PIN_DT = 15
BUTTON_PIN = 14

GPIO.setup(PIN_CLK, GPIO.IN, pull_up_down = GPIO.PUD_UP)
GPIO.setup(PIN_DT, GPIO.IN, pull_up_down = GPIO.PUD_UP)
GPIO.setup(BUTTON_PIN, GPIO.IN, pull_up_down = GPIO.PUD_UP)

# Needed variables will be initialised
Counter = 0
Richtung = True
PIN_CLK_LETZTER = 0
PIN_CLK_AKTUELL = 0
delayTime = 0.01

# Initial reading of Pin_CLK
PIN_CLK_LETZTER = GPIO.input(PIN_CLK)

# This output function will start at signal detection
def ausgabeFunktion(null):
    global Counter

    PIN_CLK_AKTUELL = GPIO.input(PIN_CLK)

    if PIN_CLK_AKTUELL != PIN_CLK_LETZTER:

        if GPIO.input(PIN_DT) != PIN_CLK_AKTUELL:
            Counter += 1
            Richtung = True;
        else:
            Richtung = False
            Counter = Counter - 1

        print "Rotation detected: "

        if Richtung:
            print "Rotational direction: Clockwise"
        else:
            print "Rotational direction: Counterclockwise"

        print "Current position: ", Counter
        print "-----"

def CounterReset(null):
    global Counter

    print "Position reset!"
    print "-----"
    Counter = 0

# To include a debounce, the output function will be initialised from the GPIO Python Module
GPIO.add_event_detect(PIN_CLK, GPIO.BOTH, callback=ausgabeFunktion, bouncetime=50)
GPIO.add_event_detect(BUTTON_PIN, GPIO.FALLING, callback=CounterReset, bouncetime=50)

print "Sensor-Test [press ctrl-c to end]"

# Main program loop
try:
    while True:
        time.sleep(delayTime)

# Scavenging work after the end of the program
except KeyboardInterrupt:
    GPIO.cleanup()
```

KY-040 Rotary encoder

Connections Raspberry Pi:

CLK	=	GPIO16	[Pin 36]
DT	=	GPIO15	[Pin 10]
SW	=	GPIO14	[Pin 8]
+	=	3,3V	[Pin 1]
GND	=	GND	[Pin 6]

Example program download[KY-040_rotary-encoder_RPi](#)

To start, enter the command:

```
sudo python KY-040_rotary-encoder_RPi.py
```