

Universidad de Costa Rica
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica
IE-0624 Laboratorio de Microcontroladores

Laboratorio 3
Arduino: GPIO, ADC y comunicaciones

Daniel Chacon Mora - B72018.
Erick Sancho Alvarado - B87388.

Profesor: MSc. Marco Villalta Fallas.

29 de enero, 2024

Índice

1. Introducción.	1
2. Nota teórica.	2
2.1. Funciones de manejo de entradas y salidas.	4
2.1.1. Pines digitales	4
2.1.2. Pines analógicos	4
2.1.3. Comunicación entre Arduino y PC	5
2.2. Periféricos utilizados.	6
2.2.1. Protocolos de comunicación	6
2.3. Diseño del circuito.	8
2.3.1. Lista de componentes y precios.	12
2.3.2. Diagrama de flujo.	13
3. Análisis de resultados.	14
4. Conclusiones y recomendaciones.	17
5. Anexos.	19
5.1. Código implementado:	19

Índice de figuras

1.	Diagrama de bloques del Arduino ATmega328. Tomado de [1]	2
2.	Diagrama de pines del microcontrolador. Tomado de [2]	3
3.	Diagrama de pines del Arduino UNO. Tomado de [3]	3
4.	Diagrama del display PCD8544.[5]	6
5.	Diseño del circuito.	8
6.	Banco de divisores de tensión.	9
7.	Banco de divisores de tensión.	10
8.	LEDs de alerta de exceso de tensión.	10
9.	Arduino Uno.	11
10.	Pantalla LCD PCD8544.	11
11.	Diagrama de flujo del código.	13
12.	Banco de divisores de tensión en operación.	14
13.	Tensión medida en canales.	14
14.	Alerta de exceso de tensión en canal.	15
15.	Resultado mostrado en terminal.	15
16.	Resultado mostrado en CSV.	16

1. Introducción.

En el contexto del desarrollo tecnológico, la necesidad de sistemas de medición y monitoreo de tensiones eléctricas es fundamental para garantizar el correcto funcionamiento de diversos dispositivos. En este proyecto, se realiza la creación de un voltímetro de 4 canales basado en el microcontrolador Arduino UNO, como objetivo principal se debe medir simultáneamente voltajes en el rango de [-24, 24]V, tanto en corriente continua (DC) como alterna (AC), además de visualizar estos valores obtenidos en una pantalla LCD PCD8544. Dicho esto, se establece una comunicación efectiva con una computadora para el registro y almacenamiento de datos en un formato CSV.

Cabe mencionar que el sistema desarrollado basado en Arduino UNO es práctico y funcional, capaz de medir tensiones en un amplio rango sin comprometer la integridad del microcontrolador gracias al condicionamiento adecuado de las señales en el presente diseño. La implementación de un sistema de LEDs de alerta y el filtro RC demuestran ser eficaces en la estabilidad y confiabilidad de las mediciones. Por otro lado, la conexión exitosa entre el Arduino y la PC para el monitoreo de variables de medición permite simular la conexión al trabajar con el dispositivo real ya que se trabaja mediante simulaciones con el software Simulide en su versión 0.4, lo cual permite que la implementación de la comunicación serial con Python resulte efectiva.

Nota: el proceso de este proyecto se puede ver evidenciado en el siguiente repositorio de git: https://github.com/danielchacon9925/Lab_microcontroladores.

2. Nota teórica.

En el presente laboratorio se utilizará el microcontrolador ATMega328 (integrado en el Arduino UNO) que se caracteriza por ser un dispositivo que tiene 14 pines de entrada/salida digitales, 6 pines de entrada analógica, un puerto serie, un puerto I²C y un puerto SPI, donde los pines de entrada/salida digitales pueden utilizarse como entradas o salidas, cuando se utilizan como entradas pueden leer señales de 0 a 5 V y cuando se utilizan como salidas proporcionan una corriente de hasta 40 mA. Cuenta con las siguientes características generales según su hoja de datos:

- Usa la arquitectura AVR de 8 bits.
- Cuenta con una memoria flash 4 KB.
- Procesador de 8 bits con una frecuencia de reloj de hasta 20 MHz.
- 32 KB de memoria flash programable.
- 2 KB de memoria EEPROM.
- 1 KB de memoria RAM.
- 23 pines de entrada/salida
- Puerto serie, puerto I²C e interrupciones.
- Además cuenta con protección contra sobretensiones.

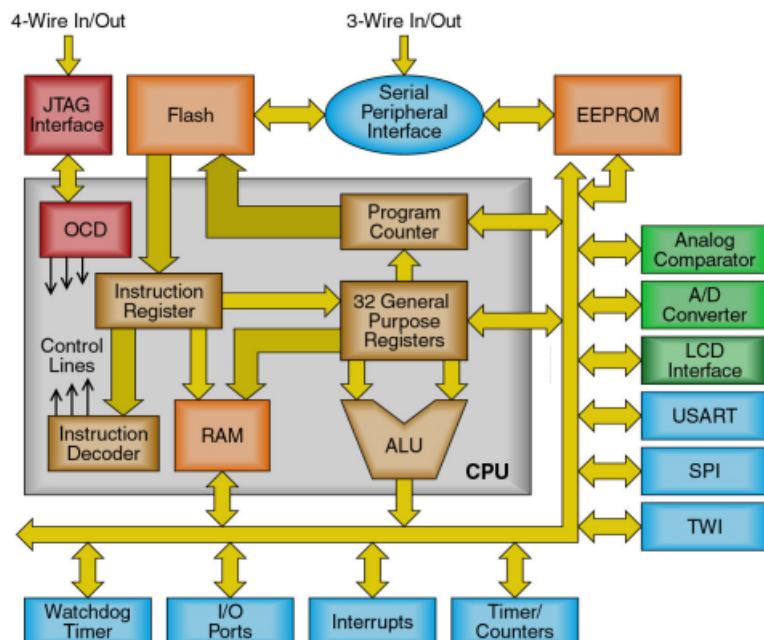


Figura 1: Diagrama de bloques del Arduino ATmega328. Tomado de [1]

Sobre sus características eléctricas según su hoja de datos:

- Voltaje de funcionamiento: 1,8V a 5,5V.
- Corriente de funcionamiento de 0,1mA a 10mA.

- Corriente de pico de 200mA.
- Resistencia de entrada y salida de $100\text{ k}\Omega$ y $25\text{ }\Omega$ respectivamente.
- Tensión y corriente de salida de 0 V a 5 V y 20 mA.
- Frecuencia de reloj de 1 MHz a 20 MHz.
- Su temperatura de funcionamiento va de -55 °C a +125 °C y la de almacenamiento va de -65 °C a +150 °C.

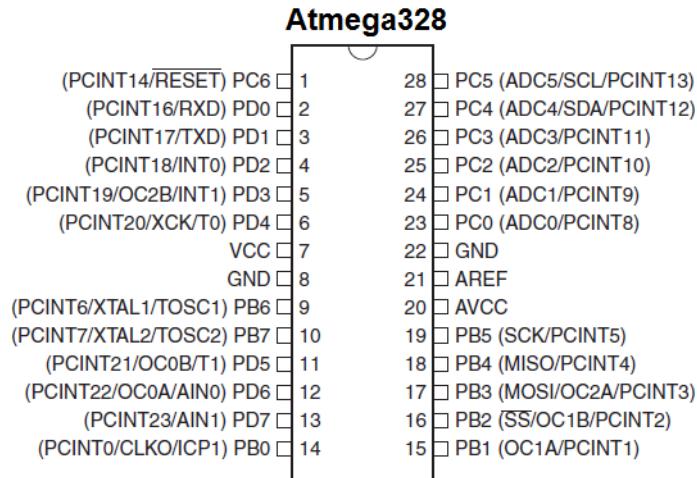


Figura 2: Diagrama de pines del microcontrolador. Tomado de [2]

En la siguiente figura se muestra el diagrama de pines Arduino UNO:

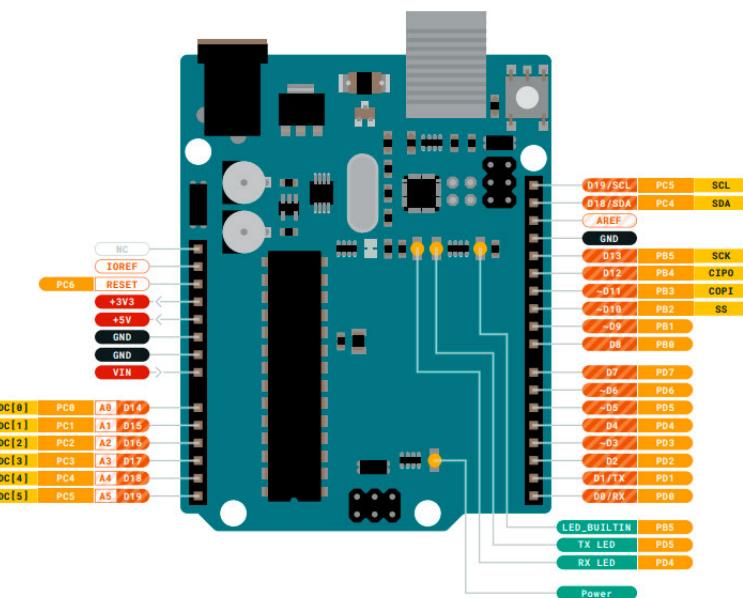


Figura 3: Diagrama de pines del Arduino UNO. Tomado de [3]

Cuenta con los siguientes pines principales:

- Cuenta con pines digitales del 0 a 13 que pueden utilizarse como pines de entrada o salida digitales.
- También cuenta con pines PWM, los cuales son D3, D5, D6, D9, D10 y D11. Estos pines permiten generar señales PWM.
- Además los pines el D2 y D3 se pueden configurar para activar interrupciones.

2.1. Funciones de manejo de entradas y salidas.

Según la guía de Arduino [4] se tienen las funciones:

2.1.1. Pines digitales

- **pinMode(pin, mode)**: Esta función se emplea para configurar el modo de un pin digital. El primer argumento especifica el número del pin, mientras que el segundo argumento indica el modo deseado (INPUT, OUTPUT o INPUT_PULLUP). Su propósito es establecer si un pin se utilizará como entrada o salida digital, así como configurar la resistencia pull-up interna.
- **digitalWrite(pin, state)**: Utilizada para establecer el estado de un pin digital. El primer argumento representa el número del pin, y el segundo argumento indica el estado deseado (HIGH o LOW). Esta función es esencial para controlar dispositivos como LEDs, relés y motores.
- **digitalRead(pin)**: Se utiliza para leer el estado de un pin digital. El único argumento es el número del pin que se desea leer. Esta función resulta útil para detectar la presencia o ausencia de señales en dispositivos como sensores, botones e interruptores.
- **pulseIn(pin, state, time)**: Utilizada para medir la duración de un pulso en un pin digital. El primer argumento especifica el número del pin, el segundo argumento indica el estado deseado (HIGH o LOW) del pulso a medir, y el tercer argumento representa el tiempo máximo en microsegundos que se esperará para que llegue el pulso antes de que la función devuelva un valor cero.

2.1.2. Pines analógicos

- **analogWrite(pin, value)**: Utilizada para generar una señal de salida analógica en un pin de salida analógica. El primer parámetro indica el número del pin que se desea configurar, mientras que el segundo parámetro especifica el valor deseado para la señal analógica. Este valor debe estar en una escala de 0 a 255.
- **analogRead(pin)**: Esta función se aprovecha para leer el valor de una señal analógica en un pin de entrada analógica. El único argumento es el número del pin que se desea examinar. La función devuelve un valor entero que representa la amplitud de la señal analógica, en una escala que va de 0 a 1023.
- **analogReference(type)**: Se utiliza para establecer el voltaje de referencia utilizado por el convertidor analógicodeigital (ADC) del Arduino UNO. El argumento especifica el tipo de referencia deseado, pudiendo ser DEFAULT, INTERNAL o EXTERNAL. La referencia predeterminada es el voltaje de alimentación del Arduino UNO, generalmente 5V. Las otras opciones permiten utilizar una referencia interna (1.1V) o una fuente de referencia externa conectada a través del pin AREF.

2.1.3. Comunicación entre Arduino y PC

- **Serial.begin(rate)**: Esta función se utiliza para iniciar la comunicación serial entre el Arduino UNO y un dispositivo externo, como una computadora o un módulo de comunicación. El único argumento es la velocidad de baudios deseada para la comunicación.
- **Serial.print(data)**: Permite enviar datos a través del puerto serie del Arduino UNO en formato ASCII legible. El argumento puede ser una cadena de texto o un valor numérico. Se emplea comúnmente para depuración y monitoreo de datos en tiempo real.
- **Serial.flush()**: Pone en espera la ejecución y espera a que se completen todas las transmisiones en serie pendientes antes de continuar el programa. Es útil para asegurarse de que no se acumulen datos en el buffer de salida.
- **Serial.println(data)**: Similar a Serial.print(), pero agrega un carácter de cambio de línea al final de los datos enviados.
- **Serial.write(data)**: Se utiliza para enviar datos a través del puerto serial en formato binario.
- **Serial.end()**: Detiene la comunicación en serie y libera recursos.
- **Serial.available()**: Determina si hay datos disponibles para ser leídos en el buffer de entrada del puerto serial y devuelve la cantidad de bytes disponibles.
- **Serial.read()**: Lee el siguiente byte disponible en el buffer de entrada del puerto serial y devuelve el valor leído o -1 si no hay bytes disponibles.
- **Serial.peek()**: Lee el siguiente byte disponible en el buffer de entrada del puerto serial sin eliminarlo, devolviendo su valor.
- **Serial.find()**: Busca un patrón específico de bytes en el buffer de entrada del puerto serial y devuelve true o false.
- **Serial.readBytes(buffer, length)**: Lee una cantidad específica de bytes del buffer de entrada del puerto serie y los almacena en un arreglo de bytes. El primer argumento es el arreglo de bytes donde se almacenarán los datos leídos, y el segundo argumento es el número de bytes que se desea leer.

2.2. Periféricos utilizados.

Descripción de los registros.

Pantalla LCD PC8544.

El display PCD8544 es un controlador LCD de bajo consumo CMOS diseñado para conducir una pantalla de matriz de puntos de 48 x 84 píxeles en el cual se puede mostrar texto, gráficos e imágenes. Es un controlador de pantalla simple y económico que se puede utilizar en una variedad de aplicaciones. A continuación se muestra su diagrama y pines:

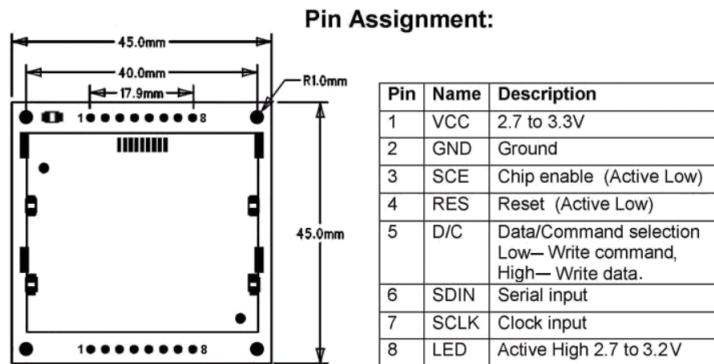


Figura 4: Diagrama del display PCD8544.[5]

2.2.1. Protocolos de comunicación

- **SPI:** El Protocolo de Interfaz Periférica Serial (SPI), cuyas siglas provienen del inglés "Serial Peripheral Interface", constituye un método sincrónico de comunicación empleado para interconectar dispositivos electrónicos digitales. Este protocolo desempeña un papel crucial en la transferencia de datos entre un microcontrolador y periféricos diversos, tales como sensores, pantallas, convertidores analógico-digital, entre otros [4].

Su estructura se basa en una conexión punto a punto entre el microcontrolador y los periféricos, haciendo uso de un bus compuesto por cuatro señales clave: SCLK (reloj), MOSI (datos de salida del maestro), MISO (datos de entrada del maestro) y SS (señal de selección de esclavo). Es el maestro quien tiene el control sobre la velocidad de transferencia de datos mediante la señal de reloj, y utiliza la señal de selección de esclavo para establecer comunicación con uno de los dispositivos periféricos conectados [6].

Este protocolo posibilita la transmisión bidireccional de datos, permitiendo que el dispositivo periférico también envíe información al microcontrolador. Esta bidireccionalidad contribuye a una comunicación eficiente y versátil en entornos donde la interacción fluida entre el microcontrolador y los periféricos es esencial.

- **USART:** El término USART, proveniente de sus siglas en inglés (Universal Synchronous/Asynchronous Receiver/Transmitter), representa un protocolo de comunicación ampliamente utilizado en sistemas de transmisión de datos, ya sea de manera síncrona o asíncrona. Este protocolo posibilita la transmisión bidireccional de datos a través de un solo canal de comunicación, característica que lo convierte en una opción valiosa en entornos de sistemas embebidos.

Usado con frecuencia en la comunicación entre microcontroladores, periféricos y dispositivos externos, el USART ofrece flexibilidad y adaptabilidad a diversos escenarios. Su capacidad para admitir velocidades de transmisión que oscilan desde pocos bits hasta megabits lo hace idóneo para aplicaciones con requerimientos variables de ancho de banda.

Además, el USART permite la configuración de parámetros clave, como la longitud de palabra, la paridad y el control de flujo. Esta capacidad de ajuste posibilita la adaptación del protocolo a las necesidades específicas de la aplicación en cuestión, otorgando un mayor control y eficiencia en la transmisión de datos en sistemas embebidos [7].

2.3. Diseño del circuito.

En la siguiente imagen se presenta el diseño del circuito realizado en el software SimulIDE para este laboratorio utilizando el ATmega328:

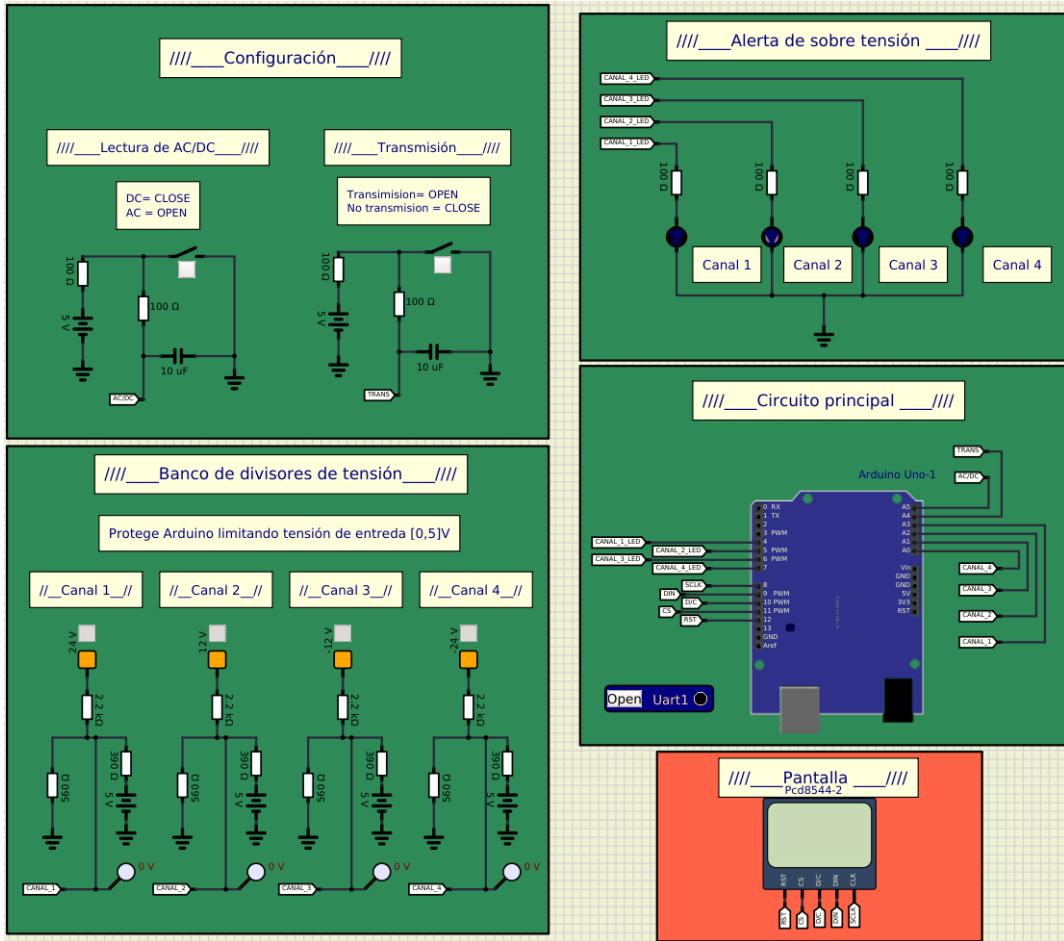


Figura 5: Diseño del circuito.

El Arduino debe de recibir tensiones en el rango de $[-24, 24]V$ debido a esto es necesario el diseño de una etapa que se encargue de reducir esta tensión al rango de $[0, 5]V$ para no comprometer la integridad del microcontrolador.

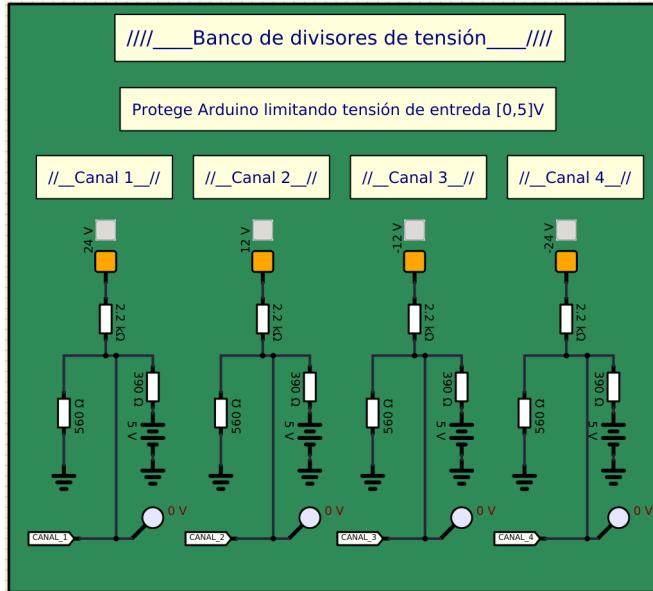


Figura 6: Banco de divisores de tensión.

La reducción de tensión es llevada a cabo por este conjunto de divisores de tensión. Los valores de resistencia fueron obtenidos utilizando el teorema de superposición.

Al apagar la fuente de tensión principal(24V) se obtiene la expresión:

$$V_{out} = \frac{R_1 \parallel R_3}{(R_1 \parallel R_3) + R_2} * V_{ref} \quad (1)$$

Al apagar la fuente de tensión de referencia(5V) principal se obtiene la expresión:

$$V_{out} = \frac{R_2 \parallel R_3}{(R_2 \parallel R_3) + R_1} * V_{in} \quad (2)$$

De esta forma, la tensión de salida está descrita por:

$$V_{out} = \frac{R_1 \parallel R_3}{(R_1 \parallel R_3) + R_2} * V_{ref} + \frac{R_2 \parallel R_3}{(R_2 \parallel R_3) + R_1} * V_{in} \quad (3)$$

Se decide tomar un valor arbitrario para R_1 de $2k\Omega$. Como requerimiento de diseño, el circuito al recibir $24V$ de V_{in} , debe de tener como tensión V_{out} un valor de $5V$; para V_{in} con valor de $-24V$, la tensión V_{out} debe de ser $0V$. Esto queda descrito en el siguiente sistema de ecuaciones:

$$5 = \frac{R_1 \parallel R_3}{(R_1 \parallel R_3) + R_2} * 5 + \frac{R_2 \parallel R_3}{(R_2 \parallel R_3) + R_1} * 24 \quad (4)$$

$$0 = \frac{R_1 \parallel R_3}{(R_1 \parallel R_3) + R_2} * 5 + \frac{R_2 \parallel R_3}{(R_2 \parallel R_3) + R_1} * -24 \quad (5)$$

Se obtienen los valores de $R_3 = 578,94\Omega$ y $R_2 = 458,33\Omega$. En el diseño se implementa $R_3 = 560\Omega$ debido a que es el valor comercial más cercano y $R_2 = 390\Omega$ debido a que al realizar pruebas con el valor más cercano ($R_2 = 470\Omega$) se observó que el valor de salida no era lo suficientemente cercano al valor deseado.

El circuito tiene la capacidad de cambiar su modo de lectura de tensión entre AC y DC, además brinda la opción de transmisión serial o no por medio de un switch.

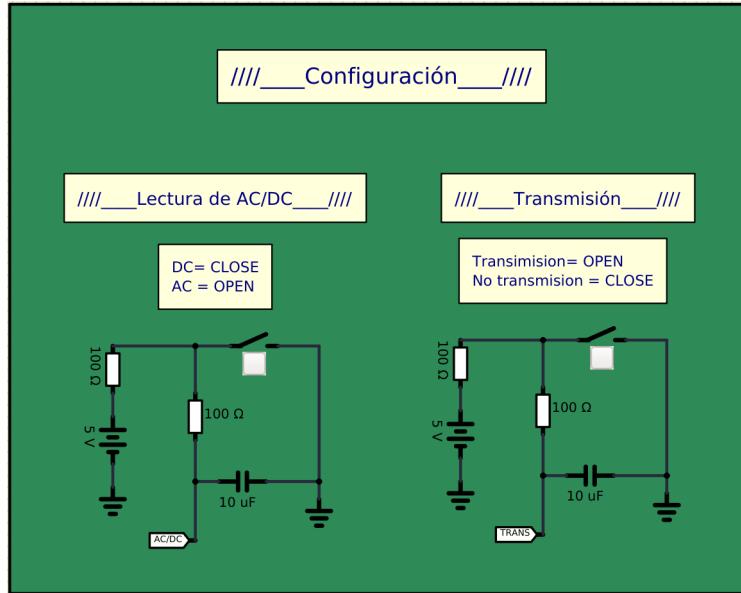


Figura 7: Banco de divisores de tensión.

Se implementa un filtro RC para evitar el rebote de la señal.

$$fc = \frac{1}{2\pi * 100\Omega * 10\mu F} = 159,2 Hz \quad (6)$$

El multímetro indica cuando se está operando en un rango más allá de $[-20, 20]V$ por medio de una etapa compuesta por LEDs amarillos y resistores de protección de 100Ω .

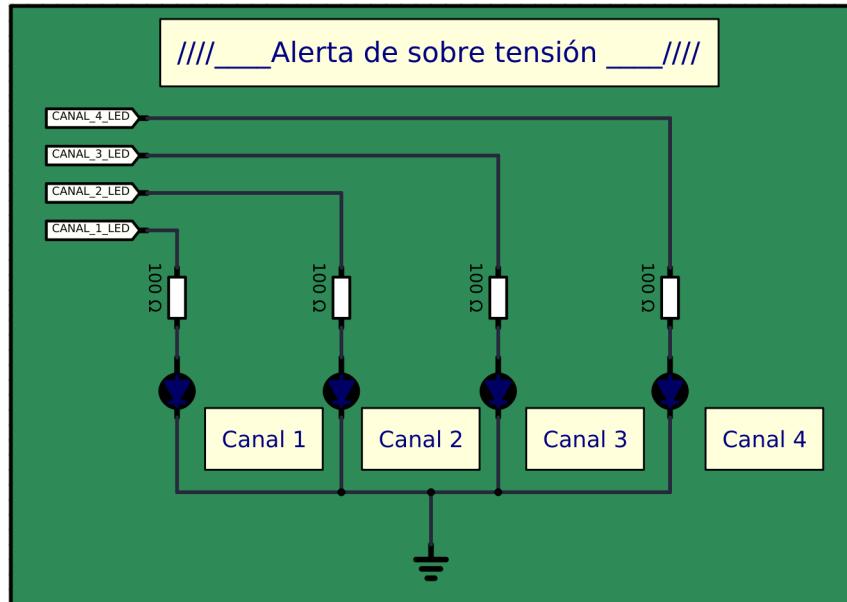


Figura 8: LEDs de alerta de exceso de tensión.

El componente principal de este circuito es el *Arduino Uno* y una pantalla LCD PCD8544 para observar en tiempo real los valores medidos.

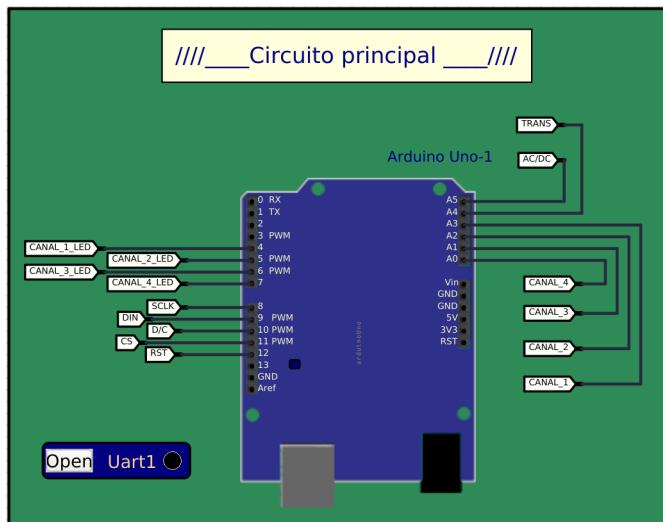


Figura 9: Arduino Uno.

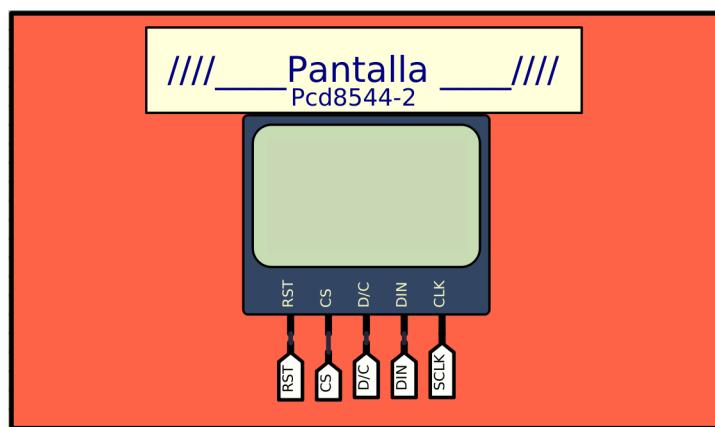


Figura 10: Pantalla LCD PCD8544.

Las conexiones fueron desarrolladas utilizando túneles para mejorar la comprensión de las diferentes etapas que constituyen el multímetro.

2.3.1. Lista de componentes y precios.

En la siguiente tabla se presenta la lista de componentes que fueron utilizados para el nuestro del circuito anterior, con información de cada uno de ellos y su precio respectivo:

Componente	Tipo	Cantidad	Precio C
Arduino UNO	Microcontrolador	1	10500 c/u
Display	PCD8544	2	5400 c/u
Resistencia	100 Ω de carbón	8	25 c/u
Resistencia	2.2 kΩ de carbón	4	25 c/u
Resistencia	560 Ω de carbón	4	25 c/u
Resistencia	390 Ω de carbón	4	25 c/u
Batería	5v	2	1200 c/u
Capacitor	Electrolítico radial de 10uF	3	130 c/u
LED Amarillo	5mm 2.1Vcc 15mA	4	75 c/u
Switch	Interruptor (ON/OFF)	2	100 c/u

Tabla 1: Lista de componentes utilizados.

2.3.2. Diagrama de flujo.

A continuación se presenta el diagrama de flujo del código realizado para cargar el firmware en el Arduino UNO:

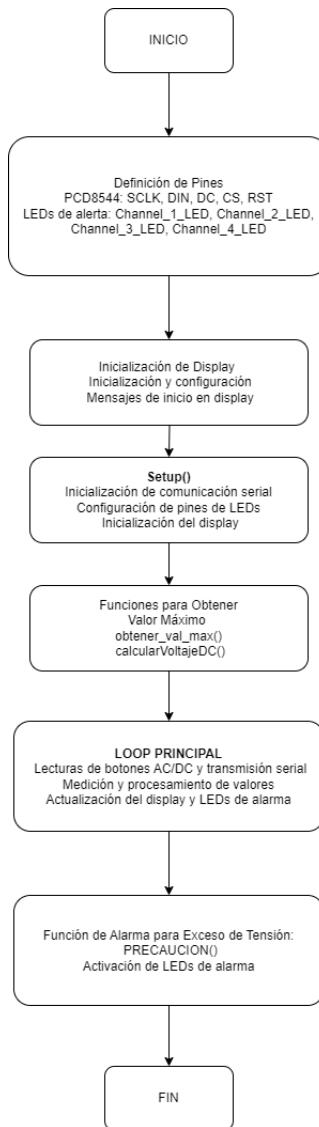


Figura 11: Diagrama de flujo del código.

3. Análisis de resultados.

El multímetro para operar correctamente debe de recibir la tensión adecuada, esto fue llevado a cabo exitosamente por el banco de divisores de tensión.

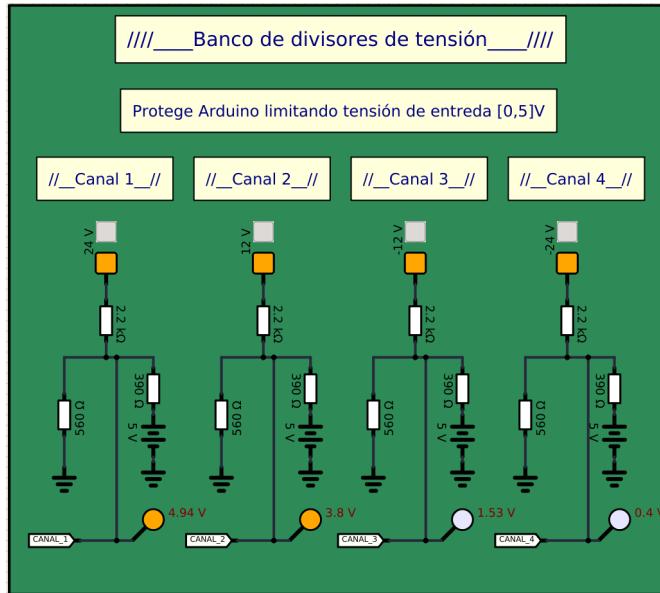


Figura 12: Banco de divisores de tensión en operación.

Se establecen 4 tensiones de entrada $[-24, -12, 12, 24]V$ y se obtiene en la salida los valores apropiados.

En la pantalla se muestran los valores correspondientes a cada canal.

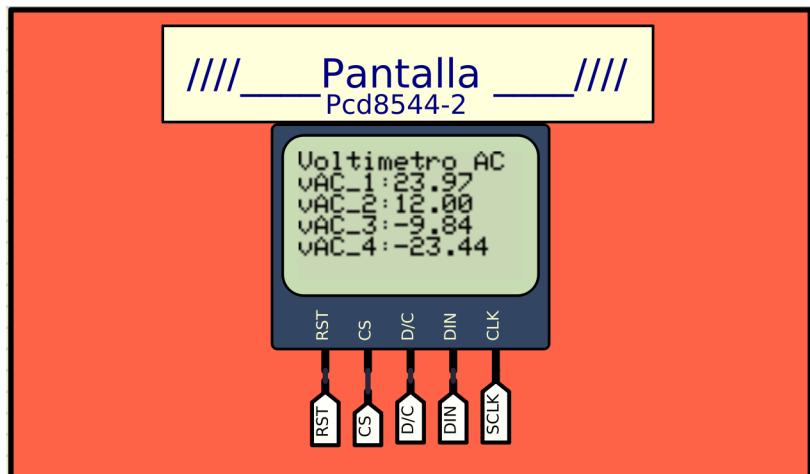


Figura 13: Tensión medida en canales.

El modo de operación del multímetro al tomar la captura de pantalla se encontraba en AC. Los canales que superan el valor máximo recomendado de operación son el canal 1 y el canal 4.

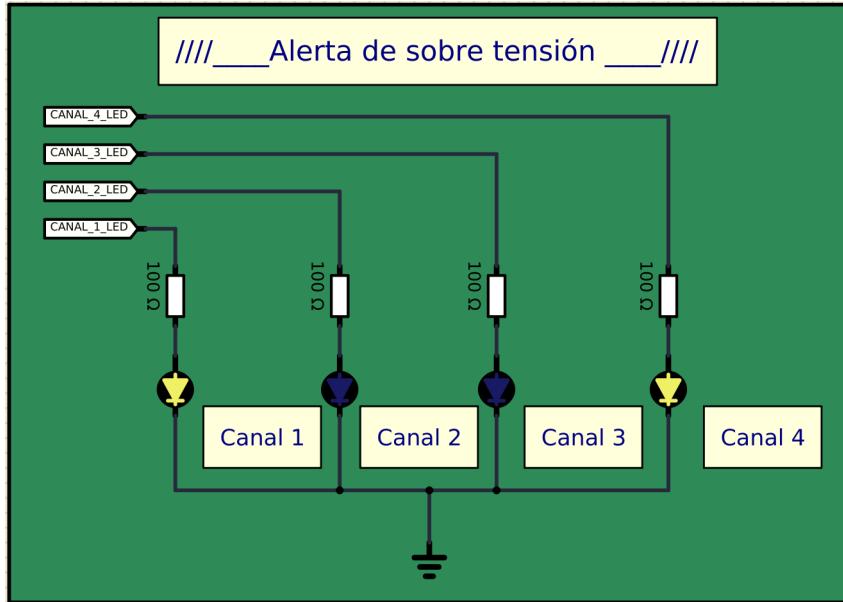


Figura 14: Alerta de exceso de tensión en canal.

La lectura de los puertos analógicos es llevada a cabo con una resolución de 10 bits, debido a esto el valor máximo que puede ser medido es $2^{10} = 1024$, el cuál representa los 5V de tensión máxima en entrada de pin. Para poder identificar el valor adecuadamente, es necesario realizar división del valor medido entre el valor máximo que puede ser medido y multiplicarlo por la tensión de referencia (5V), además es necesario sumarle la relación entre $\frac{R_2}{R_3}$.

$$Tensión_{medida} = \left(\frac{Tensión_{pinAnalog}}{1024} \right) * 5V * 0,83 + 0,165 \quad (7)$$

El valor de ajuste 0,165 fue encontrado experimentalmente y utilizado para representar la tensión medida correcta.

La comunicación serial con la computadora fue llevada a cabo exitosa. Los valores obtenidos en la pantalla son mostrados en la terminal:

```
-----MODO: AC -----
CANAL 1:...
23.97...
CANAL 2:...
12.00...
CANAL 3:...
-9.84...
CANAL 4:...
-23.44...
-----MODO: AC -----
CANAL 1:...
23.97...
CANAL 2:...
12.00...
CANAL 3:...
-9.84...
CANAL 4:...
-23.44...
```

Figura 15: Resultado mostrado en terminal.

Estos valores son obtenidos por medio del puerto *tmpettyS1* en el script de python:

:	23.97	CANAL 2:	12	CANAL 3:	
-9.84	CANAL 4:	-23.44	----	MODC	CANAL 1:
23.97	CANAL 2:	12	CANAL 3:	-9.84	
CANAL 4:	-23.44	----	MODC	CANAL 1:	23.97
CANAL 2:	12	CANAL 3:	-9.84	CANAL 4:	
-23.44	----	MODC	CANAL 1:	23.97	CANAL 2:
12	CANAL 3:	-9.84	CANAL 4:	-23.44	
----	MODC	CANAL 1:	23.97	CANAL 2:	12
CANAL 3:	-9.84	CANAL 4:	-23.44	----	MODO: AC
CANAL 1:	23.97	CANAL 2:	12	CANAL 3:	
-9.84	CANAL 4:	-23.44	----	MODC	CANAL 1:
23.97	CANAL 2:	12	CANAL 3:	-9.84	
CANAL 4:	-23.44	----	MODC	CANAL 1:	23.97
CANAL 2:	12	CANAL 3:	-9.84	CANAL 4:	
-23.44	----	MODC	CANAL 1:	23.97	CANAL 2:
12	CANAL 3:	-9.84	CANAL 4:	-23.44	
----	MODC	CANAL 1:	23.97	CANAL 2:	12
CANAL 3:	-9.84	CANAL 4:	-23.44	----	MODO: AC
CANAL 1:	23.97	CANAL 2:	12	CANAL 3:	
-9.84	CANAL 4:	-23.44	----	MODC	CANAL 1:
23.97	CANAL 2:	12	CANAL 3:	-9.84	
CANAL 4:	-23.44	----	MODC	CANAL 1:	23.97
CANAL 2:	12	CANAL 3:	-9.84	CANAL 4:	
-23.44	----	MODC	CANAL 1:	23.97	CANAL 2:
12	CANAL 3:	-9.84	CANAL 4:	-23.44	

Figura 16: Resultado mostrado en CSV.

4. Conclusiones y recomendaciones.

- El circuito diseñado funciona correctamente y cumple con los requisitos establecidos. La etapa de alerta de exceso de tensión funciona correctamente y advierte al usuario cuando la tensión de entrada está fuera del rango de operación seguro.
- Fue posible crear un sistema práctico y funcional empleando el Arduino UNO, capaz de medir tensiones entre $[-24, 24]V$ sin comprometer la integridad del microcontrolador, mantener un registro de los mismos y contar con un sistema de LEDs de alerta para excesos de tensión. El filtro RC es efectivo para evitar el rebote de la señal.
- Se establece conexión efectiva entre el Arduino y la PC para monitoreo de variables de medición, esto permite simular la conexión al trabajar con el dispositivo real.
- Por otro lado, Python es un lenguaje de programación de alto nivel y su uso es muy sencillo, a su vez fue posible realizar la comunicación serial con éxito para la comunicación con la PC de modo que se utiliza este método de comunicación entre dispositivos para extraer los datos obtenidos en el Arduino y poder documentarlos en un CSV.

Se pueden mencionar las siguientes recomendaciones:

- Se recomienda guardar constantemente el diseño en SimulIDE debido a que al incluir numerosos componentes, la aplicación tiende a congestionarse y cerrarse, lo cuál puede causar la pérdida de archivos.
- Además de utilizar la versión de Simulide 0.4 para establecer una comunicación serial correcta entre el Arduino y la PC ya que con la versión 1.0 no fue posible.

Referencias

- [1] electgpl. El atmega328p. <https://electgpl.blogspot.com/2016/06/el-atmega328p.html>.
- [2] learningaboutelectronics. Atmega328 pinout. <https://www.learningaboutelectronics.com/Articles/Atmega328-pinout.php>.
- [3] HobbyDu. Arduino uno rev. 3. <https://hobbydu.com/arduino-original/arduino-uno-rev-3-placa-desarrollo-34.html>, 2024.
- [4] Arduino. Arduino uno - technical specs, 2023. [Online].
- [5] EPA. Lcd's gráficos (pcd8544). <https://www.diarioelectronicohoy.com/blog/lcds-graficos-pcd8544>, 2013.
- [6] Texas Instruments. Understanding the spi bus, Sin fecha. Disponible en línea, accedido el 29/01/24.
- [7] Atmel Corporation. Usart universal synchronous/asynchronous receiver/transmitter, Sin fecha. Disponible en línea, accedido el 29/01/24.

5. Anexos.

5.1. Código implementado:

```
1 // Laboratorio 3: Daniel Chacon Mora (B72018), Erick Sancho Alvarado (B87388)
2
3 // Librerias utilizadas
4
5 // GFX: Funciones y operaciones basicas para el dibujo de formas geometricas, texto y manipulacion de pixeles en una pantalla.
6 #include <Adafruit_GFX.h>
7 // PCD8544: Funciones para la inicializacion de la pantalla, el dibujo de pixeles, la escritura de texto y otras operaciones graficas.
8 #include <Adafruit_PCD8544.h>
9
10
11 /////////////////
12 // Definicion de pines //
13 /////////////////
14
15 // Pantalla LCD PCD8544
16 #define SCLK 8
17 #define DIN 9
18 #define DC 10
19 #define CS 11
20 #define RST 12
21 // LEDs de alerta de exceso de tension por medir
22 #define Channel_1_LED 4
23 #define Channel_2_LED 5
24 #define Channel_3_LED 6
25 #define Channel_4_LED 7
26
27 // Inicializacion de display
28 Adafruit_PCD8544 display = Adafruit_PCD8544(SCLK, DIN, DC, CS, RST);
29
30 // Variables de medicion
31 // DC
32 float vDC_1, vDC_2, vDC_3, vDC_4 = {0.00};
33 // AC
34 float vAC_1, vAC_2, vAC_3, vAC_4 = {0.00};
35
36 // MAX values
37 // DC
38 #define MAX_DC 20
39 // AC: MAX_VOLTAGE=RMS_max=20/sqrt(20)
40 #define MAX_AC 14.14
41
42 // Setup
43 void setup() {
44
45     // Initialize serial communication
46     Serial.begin(9600);
```

```

47
48 // LEDs se configuran como salidas
49 pinMode(Channel_1_LED, OUTPUT);
50 pinMode(Channel_2_LED, OUTPUT);
51 pinMode(Channel_3_LED, OUTPUT);
52 pinMode(Channel_4_LED, OUTPUT);
53
54 // Llamado a función de display
55 // Inicialización
56 display.begin();
57 // Contraste
58 display.setContrast(50);
59 // Adafruit LOGO: Si se muestra florcita se inicia bien
60 display.display();
61 delay(1500);
62
63 // Configuración inicial de display
64 display.clearDisplay();
65 display.setTextSize(1); // Tamaño de texto
66 display.setTextColor(BLACK); // Color de texto
67 display.setCursor(5,0); // Establece cursor en coordenadas
68 display.println("Voltmetro AC/DC");
69 display.setCursor(5,10);
70 display.println("Midiendo: ");
71 display.display();
72 delay(2000);
73 display.clearDisplay(); // Clear display
74 }
75
76 // Función de las lecturas analógicas el valor máximo entre esas
    lecturas.
77 // Escalar y ajustar el valor máximo para el rango de [-24, 24]V y
    retornarlo.
78
79 float obtener_val_max(float PUERTO_ANALOGICO) {
80
81 float max = 0; // Para contar la iteración
82 for (int j = 0; j < count; j++) // Bucle de 100 iteraciones
83 {
84 // Lee el valor analógico del puerto
85 // Y se guarda en el flotante lectura
86 float lectura = analogRead(PUERTO_ANALOGICO);
87 // Compara la lectura con el valor actual max.
88 // Si la lectura es mayor que max, se actualiza val con el valor de
     lectura.
89 if (lectura > max){
90     max = lectura;
91 }
92 delayMicroseconds(300); // Pequeño delay entre las lecturas.
93 // Luego se las lecturas, se realiza una conversión para escalar
94 // El valor máximo que se leyó.
95 max = (((lectura * 5) / 1023) * 9.6) - 24;
96 }

```

```

97 // Se retorna el valor m ximo calculado.
98 return PUERTO_ANALOGICO;
99 }

100
101 // Funci n que toma como argumento el n mero de pin y
102 // retorna el voltaje DC calculado para ese canal.
103 float calcularVoltajeDC(int pin) {

104     // Lee el valor anal gico del pin especificado
105     float valorADC = analogRead(pin);

106
107     // Calcula y devuelve el voltaje DC para el canal
108     return (((valorADC * 5.0) / 1023.0) * 9.6) - 24;
109 }
110 }

111 /////////////////
112 ///LOOP Principal///
113 //////////////////

114

115 void loop(){
116     // Lee los valores anal gicos de dos pines (A1 y A0) para
117     // determinar
118     // el estado de los botones AC/DC y de transmisi n serial.
119     float read_ac_dc = analogRead(A1);
120     float read_transmicion = analogRead(A0);

121
122     if (read_ac_dc){ // AC/DC button pressed => AC MODE
123         """
124             Si el bot n AC/DC est presionado (cuando read_ac_dc es
125             diferente de cero)
126             realiza lecturas de valores m ximos para cada canal en modo AC,
127             calcula el valor RMS y muestra los resultados.
128             """
129             // Lectura
130             float vAC_1 = obtener_val_max(Channel_1_LED);
131             float vAC_2 = obtener_val_max(Channel_2_LED);
132             float vAC_3 = obtener_val_max(Channel_3_LED);
133             float vAC_4 = obtener_val_max(Channel_4_LED);

134             // RMS
135             vAC_1 = vAC_1/sqrt(2);
136             vAC_2 = vAC_2/sqrt(2);
137             vAC_3 = vAC_3/sqrt(2);
138             vAC_4 = vAC_4/sqrt(2);

139
140             if(read_transmicion){
141                 Serial.println("----- AC/DC: AC -----");
142                 Serial.println("CHANNEL 1:");
143                 Serial.println(vAC_1);
144                 Serial.println("CHANNEL 2:");
145                 Serial.println(vAC_2);
146                 Serial.println("CHANNEL 3:");
147                 Serial.println(vAC_3);

```

```

148     Serial.println("CHANNEL 4:");
149     Serial.println(vAC_4);
150 }
151
152 // Mostrar los valores medidos en el display (PCD8544-136).
153 display.print("Lectura del volt metro en AC.\n");
154
155 display.print("v1:", vAC_1, "V\n");
156 display.print("v2:", vAC_2, "V\n");
157 display.print("v3:", vAC_3, "V\n");
158 display.print("v4:", vAC_1, "V\n");
159
160 display.display();
161 display.clearDisplay();
162
163 // Call the function Led Alarm for AC
164 LED_Alarm(vAC_1, vAC_2, vAC_3, vAC_4, MAX_AC);
165 }
166 // Caso cuando se desea hacer la lectura en DC.
167 else{
168     vDC_1 = calcularVoltajeDC(Channel_1_LED);
169     vDC_1 = calcularVoltajeDC(Channel_2_LED);
170     vDC_1 = calcularVoltajeDC(Channel_3_LED);
171     vDC_1 = calcularVoltajeDC(Channel_4_LED);
172
173 // Se imprimen los valores medidos en el monitor serial.
174 if(read_transmicion){
175     Serial.println("----- AC/DC:  DC -----");
176     Serial.println("CHANNEL 1:", vDC_1);
177     Serial.println("CHANNEL 2:", vDC_2);
178     Serial.println("CHANNEL 3:", vDC_3);
179     Serial.println("CHANNEL 4:", vDC_4);
180 }
181
182 // Se imprimen los valores medidos en el display (PCD8544-136).
183 display.print("Lectura del volt metro en DC.");
184 display.print("\n");
185
186 display.print("v1: ", vDC_1, "V\n");
187 display.print("v2: ", vDC_2, "V\n");
188 display.print("v3: ", vDC_3, "V\n");
189 display.print("v4: ", vDC_4, "V\n");
190
191 display.display();
192 display.clearDisplay();
193
194 // Llama a una funcion Precaucion para verificar si se debe
195 // activar
196 // un LED de alarma basado en los valores medidos.
197 PRECAUCION(vDC_1, vDC_2, vDC_3, vDC_3, MAX_DC);
198 }
199
// Se aplica un retardo antes de reiniciar el bucle.

```

```

200 // Para estabilizar la lectura y controlar la frecuencia de
201 // actualizaci n .
202 delay(150);
203
204 // Alarma para cuando se sobrepasa valor de medici n m ximo
205 void PRECAUCION(float v1, float v2, float v3, float v4, float MODE){
206     /////////////////////
207     // Medici n en AC//  

208     /////////////////////
209     if (MODE == MAX_AC) {
210         //MAX AC VALUE: Channel 1
211         if(v1 > MAX_AC){
212             digitalWrite(Channel_1_LED, HIGH);
213         } else{
214             digitalWrite(Channel_1_LED, LOW);
215         }
216         //MAX AC VALUE: Channel 2
217         if(v2 > MAX_AC){
218             digitalWrite(Channel_2_LED, HIGH);
219         } else{
220             digitalWrite(Channel_2_LED, LOW);
221         }
222         //MAX AC VALUE: Channel 3
223         if(v3 > MAX_AC){
224             digitalWrite(Channel_3_LED, HIGH);
225         } else{
226             digitalWrite(Channel_3_LED, LOW);
227         }
228         //MAX AC VALUE: Channel 4
229         if(v4 > MAX_AC){
230             digitalWrite(Channel_4_LED, HIGH);
231         } else{
232             digitalWrite(Channel_4_LED, LOW);
233         }
234     /////////////////////
235     // Medici n en DC//  

236     /////////////////////
237     } else{
238         //MAX DC VALUE: Channel 1
239         if(vA > MAX_DC || vA < -MAX_DC){
240             digitalWrite(Channel_1_LED, HIGH);
241         } else{
242             digitalWrite(Channel_1_LED, LOW);
243         }
244         //MAX DC VALUE: Channel 2
245         if(vB > MAX_DC || vB < -MAX_DC){
246             digitalWrite(Channel_2_LED, HIGH);
247         } else{
248             digitalWrite(Channel_2_LED, LOW);
249         }
250         //MAX DC VALUE: Channel 3
251         if(vC > MAX_DC || vC < -MAX_DC){

```

```
252     digitalWrite(Channel_3_LED, HIGH);  
253 } else{  
254     digitalWrite(Channel_3_LED, LOW);  
255 }  
256 //MAX DC VALUE: Channel 4  
257 if(vD > MAX_DC || vD < -MAX_DC){  
258     digitalWrite(Channel_4_LED, HIGH);  
259 } else{  
260     digitalWrite(Channel_4_LED, LOW);  
261 }  
262 }  
263 }  
264 }
```

Features

- High Performance, Low Power AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
 - 131 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 20 MIPS Throughput at 20 MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
 - 4/8/16/32K Bytes of In-System Self-Programmable Flash program memory (ATmega48P/88P/168P/328P)
 - 256/512/1K Bytes EEPROM (ATmega48P/88P/168P/328P)
 - 512/1K/2K Bytes Internal SRAM (ATmega48P/88P/168P/328P)
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Six PWM Channels
 - 8-channel 10-bit ADC in TQFP and QFN/MLF package
 - Temperature Measurement
 - 6-channel 10-bit ADC in PDIP Package
 - Temperature Measurement
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Byte-oriented 2-wire Serial Interface (Philips I²C compatible)
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
 - 23 Programmable I/O Lines
 - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
 - 1.8 - 5.5V for ATmega48P/88P/168PV
 - 2.7 - 5.5V for ATmega48P/88P/168P
 - 1.8 - 5.5V for ATmega328P
- Temperature Range:
 - -40°C to 85°C
- Speed Grade:
 - ATmega48P/88P/168PV: 0 - 4 MHz @ 1.8 - 5.5V, 0 - 10 MHz @ 2.7 - 5.5V
 - ATmega48P/88P/168P: 0 - 10 MHz @ 2.7 - 5.5V, 0 - 20 MHz @ 4.5 - 5.5V
 - ATmega328P: 0 - 4 MHz @ 1.8 - 5.5V, 0 - 10 MHz @ 2.7 - 5.5V, 0 - 20 MHz @ 4.5 - 5.5V
- Low Power Consumption at 1 MHz, 1.8V, 25°C for ATmega48P/88P/168P:
 - Active Mode: 0.3 mA
 - Power-down Mode: 0.1 µA
 - Power-save Mode: 0.8 µA (Including 32 kHz RTC)



8-bit AVR® Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash

**ATmega48P/V
ATmega88P/V
ATmega168P/V
ATmega328P**

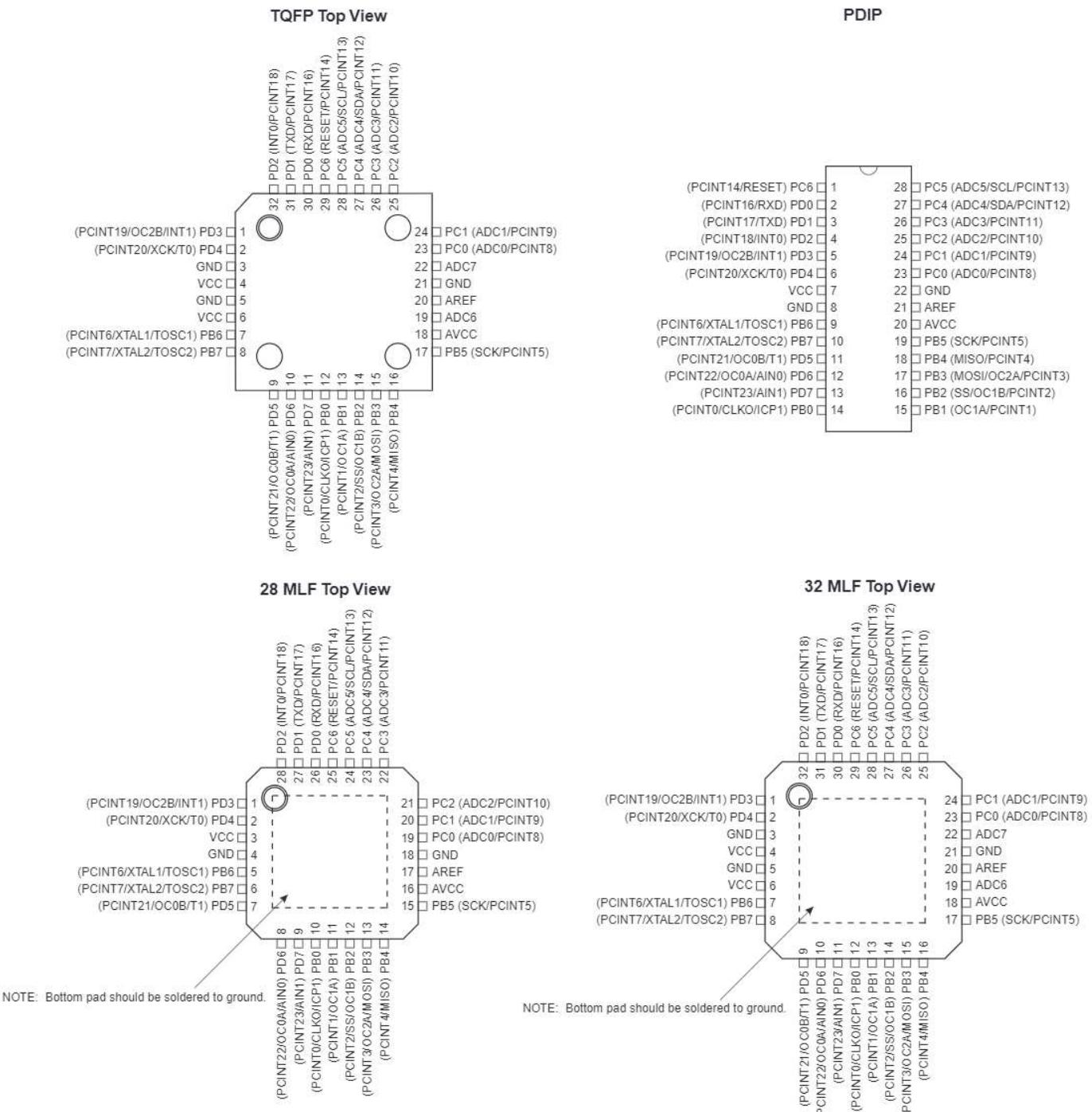
Preliminary

Summary



1. Pin Configurations

Figure 1-1. Pinout ATmega48P/88P/168P/328P



1.1 Pin Descriptions

1.1.1 VCC

Digital supply voltage.

1.1.2 GND

Ground.

1.1.3 Port B (PB7:0) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.

If the Internal Calibrated RC Oscillator is used as chip clock source, PB7..6 is used as TOSC2..1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

The various special features of Port B are elaborated in "Alternate Functions of Port B" on page 82 and "System Clock and Clock Options" on page 26.

1.1.4 Port C (PC5:0)

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5..0 output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

1.1.5 PC6/RESET

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. The minimum pulse length is given in [Table 28-3 on page 320](#). Shorter pulses are not guaranteed to generate a Reset.

The various special features of Port C are elaborated in "Alternate Functions of Port C" on page 85.

1.1.6 Port D (PD7:0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.





The various special features of Port D are elaborated in "[Alternate Functions of Port D](#)" on page 88.

1.1.7 AV_{CC}

AV_{CC} is the supply voltage pin for the A/D Converter, PC3:0, and ADC7:6. It should be externally connected to V_{CC}, even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter. Note that PC6..4 use digital supply voltage, V_{CC}.

1.1.8 AREF

AREF is the analog reference pin for the A/D Converter.

1.1.9 ADC7:6 (TQFP and QFN/MLF Package Only)

In the TQFP and QFN/MLF package, ADC7:6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.

1.2 Disclaimer

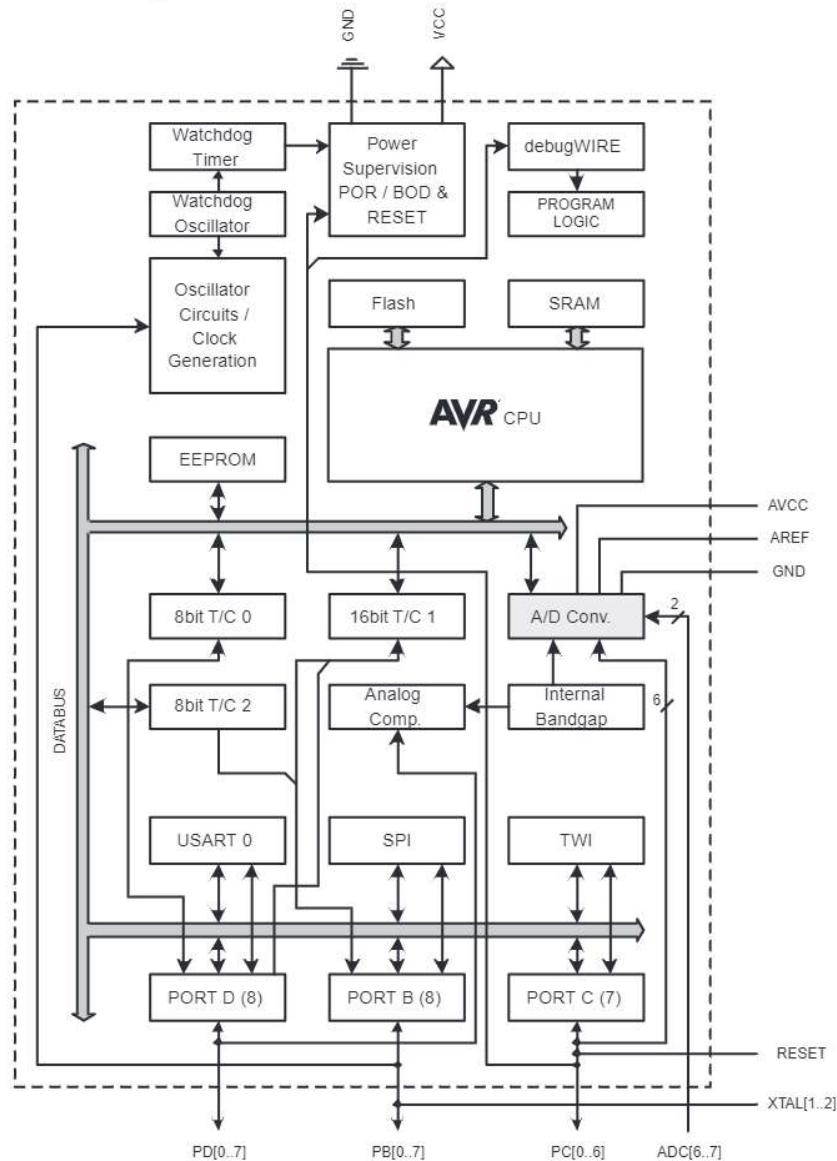
Typical values contained in this datasheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

2. Overview

The ATmega48P/88P/168P/328P is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega48P/88P/168P/328P achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

2.1 Block Diagram

Figure 2-1. Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting



architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega48P/88P/168P/328P provides the following features: 4K/8K/16K/32K bytes of In-System Programmable Flash with Read-While-Write capabilities, 256/512/512/1K bytes EEPROM, 512/1K/1K/2K bytes SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, a byte-oriented 2-wire Serial Interface, an SPI serial port, a 6-channel 10-bit ADC (8 channels in TQFP and QFN/MLF packages), a programmable Watchdog Timer with internal Oscillator, and five software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, USART, 2-wire Serial Interface, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or hardware reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption.

The device is manufactured using Atmel's high density non-volatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed In-System through an SPI serial interface, by a conventional non-volatile memory programmer, or by an On-chip Boot program running on the AVR core. The Boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega48P/88P/168P/328P is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega48P/88P/168P/328P AVR is supported with a full suite of program and system development tools including: C Compilers, Macro Assemblers, Program Debugger/Simulators, In-Circuit Emulators, and Evaluation kits.

2.2 Comparison Between ATmega48P, ATmega88P, ATmega168P, and ATmega328P

The ATmega48P, ATmega88P, ATmega168P, and ATmega328P differ only in memory sizes, boot loader support, and interrupt vector sizes. [Table 2-1](#) summarizes the different memory and interrupt vector sizes for the three devices.

Table 2-1. Memory Size Summary

Device	Flash	EEPROM	RAM	Interrupt Vector Size
ATmega48P	4K Bytes	256 Bytes	512 Bytes	1 instruction word/vector
ATmega88P	8K Bytes	512 Bytes	1K Bytes	1 instruction word/vector
ATmega168P	16K Bytes	512 Bytes	1K Bytes	2 instruction words/vector
ATmega328P	32K Bytes	1K Bytes	2K Bytes	2 instructions words/vector

ATmega88P, ATmega168P, and ATmega328P support a real Read-While-Write Self-Programming mechanism. There is a separate Boot Loader Section, and the SPM instruction can only execute from there. In ATmega48P, there is no Read-While-Write support and no separate Boot Loader Section. The SPM instruction can execute from the entire Flash.

3. Resources

A comprehensive set of development tools, application notes and datasheets are available for download on <http://www.atmel.com/avr>.

4. Data Retention

Reliability Qualification results show that the projected data retention failure rate is much less than 1 PPM over 20 years at 85°C or 100 years at 25°C.

5. Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page	
(0xFF)	Reserved	-	-	-	-	-	-	-	-	-	
(0xFE)	Reserved	-	-	-	-	-	-	-	-	-	
(0xFD)	Reserved	-	-	-	-	-	-	-	-	-	
(0xFC)	Reserved	-	-	-	-	-	-	-	-	-	
(0xFB)	Reserved	-	-	-	-	-	-	-	-	-	
(0xFA)	Reserved	-	-	-	-	-	-	-	-	-	
(0xF9)	Reserved	-	-	-	-	-	-	-	-	-	
(0xF8)	Reserved	-	-	-	-	-	-	-	-	-	
(0xF7)	Reserved	-	-	-	-	-	-	-	-	-	
(0xF6)	Reserved	-	-	-	-	-	-	-	-	-	
(0xF5)	Reserved	-	-	-	-	-	-	-	-	-	
(0xF4)	Reserved	-	-	-	-	-	-	-	-	-	
(0xF3)	Reserved	-	-	-	-	-	-	-	-	-	
(0xF2)	Reserved	-	-	-	-	-	-	-	-	-	
(0xF1)	Reserved	-	-	-	-	-	-	-	-	-	
(0xF0)	Reserved	-	-	-	-	-	-	-	-	-	
(0xEF)	Reserved	-	-	-	-	-	-	-	-	-	
(0xEE)	Reserved	-	-	-	-	-	-	-	-	-	
(0xED)	Reserved	-	-	-	-	-	-	-	-	-	
(0xEC)	Reserved	-	-	-	-	-	-	-	-	-	
(0xEB)	Reserved	-	-	-	-	-	-	-	-	-	
(0xEA)	Reserved	-	-	-	-	-	-	-	-	-	
(0xE9)	Reserved	-	-	-	-	-	-	-	-	-	
(0xE8)	Reserved	-	-	-	-	-	-	-	-	-	
(0xE7)	Reserved	-	-	-	-	-	-	-	-	-	
(0xE6)	Reserved	-	-	-	-	-	-	-	-	-	
(0xE5)	Reserved	-	-	-	-	-	-	-	-	-	
(0xE4)	Reserved	-	-	-	-	-	-	-	-	-	
(0xE3)	Reserved	-	-	-	-	-	-	-	-	-	
(0xE2)	Reserved	-	-	-	-	-	-	-	-	-	
(0xE1)	Reserved	-	-	-	-	-	-	-	-	-	
(0xE0)	Reserved	-	-	-	-	-	-	-	-	-	
(0xDF)	Reserved	-	-	-	-	-	-	-	-	-	
(0xDE)	Reserved	-	-	-	-	-	-	-	-	-	
(0xDD)	Reserved	-	-	-	-	-	-	-	-	-	
(0xDC)	Reserved	-	-	-	-	-	-	-	-	-	
(0xDB)	Reserved	-	-	-	-	-	-	-	-	-	
(0xDA)	Reserved	-	-	-	-	-	-	-	-	-	
(0xD9)	Reserved	-	-	-	-	-	-	-	-	-	
(0xD8)	Reserved	-	-	-	-	-	-	-	-	-	
(0xD7)	Reserved	-	-	-	-	-	-	-	-	-	
(0xD6)	Reserved	-	-	-	-	-	-	-	-	-	
(0xD5)	Reserved	-	-	-	-	-	-	-	-	-	
(0xD4)	Reserved	-	-	-	-	-	-	-	-	-	
(0xD3)	Reserved	-	-	-	-	-	-	-	-	-	
(0xD2)	Reserved	-	-	-	-	-	-	-	-	-	
(0xD1)	Reserved	-	-	-	-	-	-	-	-	-	
(0xD0)	Reserved	-	-	-	-	-	-	-	-	-	
(0xCF)	Reserved	-	-	-	-	-	-	-	-	-	
(0xCE)	Reserved	-	-	-	-	-	-	-	-	-	
(0xCD)	Reserved	-	-	-	-	-	-	-	-	-	
(0xCC)	Reserved	-	-	-	-	-	-	-	-	-	
(0xCB)	Reserved	-	-	-	-	-	-	-	-	-	
(0xCA)	Reserved	-	-	-	-	-	-	-	-	-	
(0xC9)	Reserved	-	-	-	-	-	-	-	-	-	
(0xC8)	Reserved	-	-	-	-	-	-	-	-	-	
(0xC7)	Reserved	-	-	-	-	-	-	-	-	-	
(0xC6)	UDR0	USART I/O Data Register									195
(0xC5)	UBRR0H	USART Baud Rate Register High									199
(0xC4)	UBRR0L	USART Baud Rate Register Low									199
(0xC3)	Reserved	-	-	-	-	-	-	-	-	-	-
(0xC2)	UCSR0C	UMSEL01	UMSEL00	UPM01	UPM00	USBS0	UCSZ01 / UDORD0	UCSZ00 / UCPHA0	UCPOL0	197/212	
(0xC1)	UCSR0B	RXCIE0	TXCIE0	UDRIE0	RXEN0	TXEN0	UCSZ02	RXB80	TXB80	196	
(0xC0)	UCSR0A	RXC0	TXC0	UDRE0	FE0	DOR0	UPE0	U2X0	MPCM0	195	

ATmega48P/88P/168P/328P

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
(0xBF)	Reserved	-	-	-	-	-	-	-	-	
(0xBE)	Reserved	-	-	-	-	-	-	-	-	
(0xBD)	TWAMR	TWAM6	TWAM5	TWAM4	TWAM3	TWAM2	TWAM1	TWAM0	-	245
(0xBC)	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE	242
(0xBB)	TWDR	2-wire Serial Interface Data Register								244
(0xBA)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	245
(0xB9)	TWSR	TWS7	TWS6	TWS5	TWS4	TWS3	-	TWPS1	TWPS0	244
(0xB8)	TWBR	2-wire Serial Interface Bit Rate Register								242
(0xB7)	Reserved	-	-	-	-	-	-	-	-	
(0xB6)	ASSR	-	EXCLK	AS2	TCN2UB	OCR2AUB	OCR2BUB	TCR2AUB	TCR2BUB	164
(0xB5)	Reserved	-	-	-	-	-	-	-	-	
(0xB4)	OCR2B	Timer/Counter2 Output Compare Register B								162
(0xB3)	OCR2A	Timer/Counter2 Output Compare Register A								162
(0xB2)	TCNT2	Timer/Counter2 (8-bit)								162
(0xB1)	TCCR2B	FOC2A	FOC2B	-	-	WGM22	CS22	CS21	CS20	161
(0xB0)	TCCR2A	COM2A1	COM2A0	COM2B1	COM2B0	-	-	WGM21	WGM20	158
(0xAF)	Reserved	-	-	-	-	-	-	-	-	
(0xAE)	Reserved	-	-	-	-	-	-	-	-	
(0xAD)	Reserved	-	-	-	-	-	-	-	-	
(0xAC)	Reserved	-	-	-	-	-	-	-	-	
(0xAB)	Reserved	-	-	-	-	-	-	-	-	
(0xAA)	Reserved	-	-	-	-	-	-	-	-	
(0xA9)	Reserved	-	-	-	-	-	-	-	-	
(0xA8)	Reserved	-	-	-	-	-	-	-	-	
(0xA7)	Reserved	-	-	-	-	-	-	-	-	
(0xA6)	Reserved	-	-	-	-	-	-	-	-	
(0xA5)	Reserved	-	-	-	-	-	-	-	-	
(0xA4)	Reserved	-	-	-	-	-	-	-	-	
(0xA3)	Reserved	-	-	-	-	-	-	-	-	
(0xA2)	Reserved	-	-	-	-	-	-	-	-	
(0xA1)	Reserved	-	-	-	-	-	-	-	-	
(0xA0)	Reserved	-	-	-	-	-	-	-	-	
(0x9F)	Reserved	-	-	-	-	-	-	-	-	
(0x9E)	Reserved	-	-	-	-	-	-	-	-	
(0x9D)	Reserved	-	-	-	-	-	-	-	-	
(0x9C)	Reserved	-	-	-	-	-	-	-	-	
(0x9B)	Reserved	-	-	-	-	-	-	-	-	
(0x9A)	Reserved	-	-	-	-	-	-	-	-	
(0x99)	Reserved	-	-	-	-	-	-	-	-	
(0x98)	Reserved	-	-	-	-	-	-	-	-	
(0x97)	Reserved	-	-	-	-	-	-	-	-	
(0x96)	Reserved	-	-	-	-	-	-	-	-	
(0x95)	Reserved	-	-	-	-	-	-	-	-	
(0x94)	Reserved	-	-	-	-	-	-	-	-	
(0x93)	Reserved	-	-	-	-	-	-	-	-	
(0x92)	Reserved	-	-	-	-	-	-	-	-	
(0x91)	Reserved	-	-	-	-	-	-	-	-	
(0x90)	Reserved	-	-	-	-	-	-	-	-	
(0x8F)	Reserved	-	-	-	-	-	-	-	-	
(0x8E)	Reserved	-	-	-	-	-	-	-	-	
(0x8D)	Reserved	-	-	-	-	-	-	-	-	
(0x8C)	Reserved	-	-	-	-	-	-	-	-	
(0x8B)	OCR1BH	Timer/Counter1 - Output Compare Register B High Byte								138
(0x8A)	OCR1BL	Timer/Counter1 - Output Compare Register B Low Byte								138
(0x89)	OCR1AH	Timer/Counter1 - Output Compare Register A High Byte								138
(0x88)	OCR1AL	Timer/Counter1 - Output Compare Register A Low Byte								138
(0x87)	ICR1H	Timer/Counter1 - Input Capture Register High Byte								139
(0x86)	ICR1L	Timer/Counter1 - Input Capture Register Low Byte								139
(0x85)	TCNT1H	Timer/Counter1 - Counter Register High Byte								138
(0x84)	TCNT1L	Timer/Counter1 - Counter Register Low Byte								138
(0x83)	Reserved	-	-	-	-	-	-	-	-	
(0x82)	TCCR1C	FOC1A	FOC1B	-	-	-	-	-	-	137
(0x81)	TCCR1B	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10	136
(0x80)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	-	-	WGM11	WGM10	134
(0x7F)	DIDR1	-	-	-	-	-	-	AIN1D	AIN0D	250
(0x7E)	DIDR0	-	-	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D	267

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
(0x7D)	Reserved	-	-	-	-	-	-	-	-	
(0x7C)	ADMUX	REFS1	REFS0	ADLAR	-	MUX3	MUX2	MUX1	MUX0	263
(0x7B)	ADCSRB	-	ACME	-	-	-	ADTS2	ADTS1	ADTS0	266
(0x7A)	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	264
(0x79)	ADCH	ADC Data Register High byte								266
(0x78)	ADCL	ADC Data Register Low byte								266
(0x77)	Reserved	-	-	-	-	-	-	-	-	
(0x76)	Reserved	-	-	-	-	-	-	-	-	
(0x75)	Reserved	-	-	-	-	-	-	-	-	
(0x74)	Reserved	-	-	-	-	-	-	-	-	
(0x73)	Reserved	-	-	-	-	-	-	-	-	
(0x72)	Reserved	-	-	-	-	-	-	-	-	
(0x71)	Reserved	-	-	-	-	-	-	-	-	
(0x70)	TIMSK2	-	-	-	-	-	OCIE2B	OCIE2A	TOIE2	163
(0x6F)	TIMSK1	-	-	ICIE1	-	-	OCIE1B	OCIE1A	TOIE1	139
(0x6E)	TIMSK0	-	-	-	-	-	OCIE0B	OCIE0A	TOIE0	111
(0x6D)	PCMSK2	PCINT23	PCINT22	PCINT21	PCINT20	PCINT19	PCINT18	PCINT17	PCINT16	74
(0x6C)	PCMSK1	-	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8	74
(0x6B)	PCMSK0	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	74
(0x6A)	Reserved	-	-	-	-	-	-	-	-	
(0x69)	EICRA	-	-	-	-	ISC11	ISC10	ISC01	ISC00	71
(0x68)	PCICR	-	-	-	-	-	PCIE2	PCIE1	PCIE0	
(0x67)	Reserved	-	-	-	-	-	-	-	-	
(0x66)	OSCCAL	Oscillator Calibration Register								37
(0x65)	Reserved	-	-	-	-	-	-	-	-	
(0x64)	PRR	PRTWI	PRTIM2	PRTIM0	-	PRTIM1	PRSPI	PRUSART0	PRADC	42
(0x63)	Reserved	-	-	-	-	-	-	-	-	
(0x62)	Reserved	-	-	-	-	-	-	-	-	
(0x61)	CLKPR	CLKPCE	-	-	-	CLKPS3	CLKPS2	CLKPS1	CLKPS0	37
(0x60)	WDTCR	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0	54
0x3F (0x5F)	SREG	I	T	H	S	V	N	Z	C	9
0x3E (0x5E)	SPH	-	-	-	-	-	(SP10) ⁵	SP9	SP8	12
0x3D (0x5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	12
0x3C (0x5C)	Reserved	-	-	-	-	-	-	-	-	
0x3B (0x5B)	Reserved	-	-	-	-	-	-	-	-	
0x3A (0x5A)	Reserved	-	-	-	-	-	-	-	-	
0x39 (0x59)	Reserved	-	-	-	-	-	-	-	-	
0x38 (0x58)	Reserved	-	-	-	-	-	-	-	-	
0x37 (0x57)	SPMCSR	SPMIE	(RWWSB) ⁵	-	(RWWSR) ⁵	BLBSET	PGWRT	PGERS	SELFPRGEN	293
0x36 (0x56)	Reserved	-	-	-	-	-	-	-	-	
0x35 (0x55)	MCUCR	-	BODS	BODSE	PUD	-	-	IVSEL	IVCE	44/68/92
0x34 (0x54)	MCUSR	-	-	-	-	WDRF	BORF	EXTRF	PORF	54
0x33 (0x53)	SMCR	-	-	-	-	SM2	SM1	SM0	SE	40
0x32 (0x52)	Reserved	-	-	-	-	-	-	-	-	
0x31 (0x51)	Reserved	-	-	-	-	-	-	-	-	
0x30 (0x50)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	248
0x2F (0x4F)	Reserved	-	-	-	-	-	-	-	-	
0x2E (0x4E)	SPDR	SPI Data Register								175
0x2D (0x4D)	SPSR	SPIF	WCOL	-	-	-	-	-	SPI2X	174
0x2C (0x4C)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	173
0x2B (0x4B)	GPIOR2	General Purpose I/O Register 2								25
0x2A (0x4A)	GPIOR1	General Purpose I/O Register 1								25
0x29 (0x49)	Reserved	-	-	-	-	-	-	-	-	
0x28 (0x48)	OCR0B	Timer/Counter0 Output Compare Register B								
0x27 (0x47)	OCR0A	Timer/Counter0 Output Compare Register A								
0x26 (0x46)	TCNT0	Timer/Counter0 (8-bit)								
0x25 (0x45)	TCCR0B	FOC0A	FOC0B	-	-	WGM02	CS02	CS01	CS00	
0x24 (0x44)	TCCR0A	COM0A1	COM0A0	COM0B1	COM0B0	-	-	WGM01	WGM00	
0x23 (0x43)	GTCCR	TSM	-	-	-	-	-	PSRASY	PSRSYNC	143/165
0x22 (0x42)	EEARH	(EEPROM Address Register High Byte) ⁵								21
0x21 (0x41)	EEARL	EEPROM Address Register Low Byte								21
0x20 (0x40)	EEDR	EEPROM Data Register								21
0x1F (0x3F)	EECR	-	-	EEP1	EEP0	EERIE	EEMPE	EEPE	EERE	21
0x1E (0x3E)	GPIOR0	General Purpose I/O Register 0								25
0x1D (0x3D)	EIMSK	-	-	-	-	-	-	INT1	INT0	72
0x1C (0x3C)	EIFR	-	-	-	-	-	-	INTF1	INTF0	72

ATmega48P/88P/168P/328P

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0x1B (0x3B)	PCIFR	-	-	-	-	-	PCIF2	PCIF1	PCIF0	
0x1A (0x3A)	Reserved	-	-	-	-	-	-	-	-	
0x19 (0x39)	Reserved	-	-	-	-	-	-	-	-	
0x18 (0x38)	Reserved	-	-	-	-	-	-	-	-	
0x17 (0x37)	TIFR2	-	-	-	-	-	OCF2B	OCF2A	TOV2	163
0x16 (0x36)	TIFR1	-	-	ICF1	-	-	OCF1B	OCF1A	TOV1	140
0x15 (0x35)	TIFR0	-	-	-	-	-	OCF0B	OCF0A	TOV0	
0x14 (0x34)	Reserved	-	-	-	-	-	-	-	-	
0x13 (0x33)	Reserved	-	-	-	-	-	-	-	-	
0x12 (0x32)	Reserved	-	-	-	-	-	-	-	-	
0x11 (0x31)	Reserved	-	-	-	-	-	-	-	-	
0x10 (0x30)	Reserved	-	-	-	-	-	-	-	-	
0x0F (0x2F)	Reserved	-	-	-	-	-	-	-	-	
0x0E (0x2E)	Reserved	-	-	-	-	-	-	-	-	
0x0D (0x2D)	Reserved	-	-	-	-	-	-	-	-	
0x0C (0x2C)	Reserved	-	-	-	-	-	-	-	-	
0x0B (0x2B)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	93
0x0A (0x2A)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	93
0x09 (0x29)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	93
0x08 (0x28)	PORTC	-	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	92
0x07 (0x27)	DDRC	-	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	92
0x06 (0x26)	PINC	-	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	92
0x05 (0x25)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	92
0x04 (0x24)	DDRB	DBB7	DBB6	DBB5	DBB4	DBB3	DBB2	DBB1	DBB0	92
0x03 (0x23)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	92
0x02 (0x22)	Reserved	-	-	-	-	-	-	-	-	
0x01 (0x21)	Reserved	-	-	-	-	-	-	-	-	
0x0 (0x20)	Reserved	-	-	-	-	-	-	-	-	

- Note:
- For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
 - I/O Registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions.
 - Some of the Status Flags are cleared by writing a logical one to them. Note that, unlike most other AVRs, the CBI and SBI instructions will only operate on the specified bit, and can therefore be used on registers containing such Status Flags. The CBI and SBI instructions work with registers 0x00 to 0x1F only.
 - When using the I/O specific commands IN and OUT, the I/O addresses 0x00 - 0x3F must be used. When addressing I/O Registers as data space using LD and ST instructions, 0x20 must be added to these addresses. The ATmega48P/88P/168P/328P is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in Opcode for the IN and OUT instructions. For the Extended I/O space from 0x60 - 0xFF in SRAM, only the ST/STS/STD and LD/LDS/DDD instructions can be used.
 - Only valid for ATmega88P/168P.



6. Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Clocks
ARITHMETIC AND LOGIC INSTRUCTIONS					
ADD	Rd, Rr	Add two Registers	$Rd \leftarrow Rd + Rr$	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	$Rd \leftarrow Rd + Rr + C$	Z,C,N,V,H	1
ADIW	Rdl,K	Add Immediate to Word	$Rdh:Rdl \leftarrow Rdh:Rdl + K$	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract two Registers	$Rd \leftarrow Rd - Rr$	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	$Rd \leftarrow Rd - K$	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry two Registers	$Rd \leftarrow Rd - Rr - C$	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	$Rd \leftarrow Rd - K - C$	Z,C,N,V,H	1
SBIW	Rdl,K	Subtract Immediate from Word	$Rdh:Rdl \leftarrow Rdh:Rdl - K$	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND Registers	$Rd \leftarrow Rd \bullet Rr$	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	$Rd \leftarrow Rd \bullet K$	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	$Rd \leftarrow Rd \vee Rr$	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	$Rd \leftarrow Rd \vee K$	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	$Rd \leftarrow Rd \oplus Rr$	Z,N,V	1
COM	Rd	One's Complement	$Rd \leftarrow 0xFF - Rd$	Z,C,N,V	1
NEG	Rd	Two's Complement	$Rd \leftarrow 0x00 - Rd$	Z,C,N,V,H	1
SBR	Rd,K	Set Bit(s) in Register	$Rd \leftarrow Rd \vee K$	Z,N,V	1
CBR	Rd,K	Clear Bit(s) in Register	$Rd \leftarrow Rd \bullet (0xFF - K)$	Z,N,V	1
INC	Rd	Increment	$Rd \leftarrow Rd + 1$	Z,N,V	1
DEC	Rd	Decrement	$Rd \leftarrow Rd - 1$	Z,N,V	1
TST	Rd	Test for Zero or Minus	$Rd \leftarrow Rd \bullet Rd$	Z,N,V	1
CLR	Rd	Clear Register	$Rd \leftarrow Rd \oplus Rd$	Z,N,V	1
SER	Rd	Set Register	$Rd \leftarrow 0xFF$	None	1
MUL	Rd, Rr	Multiply Unsigned	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
MULS	Rd, Rr	Multiply Signed	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
MULSU	Rd, Rr	Multiply Signed with Unsigned	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
FMUL	Rd, Rr	Fractional Multiply Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \lll 1$	Z,C	2
FMULS	Rd, Rr	Fractional Multiply Signed	$R1:R0 \leftarrow (Rd \times Rr) \lll 1$	Z,C	2
FMULSU	Rd, Rr	Fractional Multiply Signed with Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \lll 1$	Z,C	2
BRANCH INSTRUCTIONS					
RJMP	k	Relative Jump	$PC \leftarrow PC + k + 1$	None	2
JMP		Indirect Jump to (Z)	$PC \leftarrow Z$	None	2
JMP ⁽¹⁾	k	Direct Jump	$PC \leftarrow k$	None	3
RCALL	k	Relative Subroutine Call	$PC \leftarrow PC + k + 1$	None	3
ICALL		Indirect Call to (Z)	$PC \leftarrow Z$	None	3
CALL ⁽¹⁾	k	Direct Subroutine Call	$PC \leftarrow k$	None	4
RET		Subroutine Return	$PC \leftarrow STACK$	None	4
RETI		Interrupt Return	$PC \leftarrow STACK$	I	4
CPSE	Rd,Rr	Compare, Skip if Equal	if (Rd = Rr) $PC \leftarrow PC + 2$ or 3	None	1/2/3
CP	Rd,Rr	Compare	$Rd \leftarrow Rr$	Z,N,V,C,H	1
CPC	Rd,Rr	Compare with Carry	$Rd \leftarrow Rr - C$	Z,N,V,C,H	1
CPI	Rd,K	Compare Register with Immediate	$Rd \leftarrow K$	Z,N,V,C,H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	if (Rr(b)=0) $PC \leftarrow PC + 2$ or 3	None	1/2/3
SBRS	Rr, b	Skip if Bit in Register is Set	if (Rr(b)=1) $PC \leftarrow PC + 2$ or 3	None	1/2/3
SBIC	P, b	Skip if Bit in I/O Register Cleared	if (P(b)=0) $PC \leftarrow PC + 2$ or 3	None	1/2/3
SBIS	P, b	Skip if Bit in I/O Register is Set	if (P(b)=1) $PC \leftarrow PC + 2$ or 3	None	1/2/3
BRBS	s, k	Branch if Status Flag Set	if (SREG(s) = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRBC	s, k	Branch if Status Flag Cleared	if (SREG(s) = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BREQ	k	Branch if Equal	if (Z = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRNE	k	Branch if Not Equal	if (Z = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BRCS	k	Branch if Carry Set	if (C = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRCC	k	Branch if Carry Cleared	if (C = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BRSH	k	Branch if Same or Higher	if (C = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BRLO	k	Branch if Lower	if (C = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRMI	k	Branch if Minus	if (N = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRPL	k	Branch if Plus	if (N = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BRGE	k	Branch if Greater or Equal, Signed	if (N ⊕ V = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BRLT	k	Branch if Less Than Zero, Signed	if (N ⊕ V = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRHS	k	Branch if Half Carry Flag Set	if (H = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRHC	k	Branch if Half Carry Flag Cleared	if (H = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BRTS	k	Branch if T Flag Set	if (T = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRTC	k	Branch if T Flag Cleared	if (T = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BRVS	k	Branch if Overflow Flag is Set	if (V = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRVC	k	Branch if Overflow Flag is Cleared	if (V = 0) then $PC \leftarrow PC + k + 1$	None	1/2

ATmega48P/88P/168P/328P

Mnemonics	Operands	Description	Operation	Flags	#Clocks
BRIE	k	Branch if Interrupt Enabled	if (I = 1) then PC \leftarrow PC + k + 1	None	1/2
BRID	k	Branch if Interrupt Disabled	if (I = 0) then PC \leftarrow PC + k + 1	None	1/2
BIT AND BIT-TEST INSTRUCTIONS					
SBI	P,b	Set Bit in I/O Register	I/O(P,b) \leftarrow 1	None	2
CBI	P,b	Clear Bit in I/O Register	I/O(P,b) \leftarrow 0	None	2
LSL	Rd	Logical Shift Left	Rd(n+1) \leftarrow Rd(n), Rd(0) \leftarrow 0	Z,C,N,V	1
LSR	Rd	Logical Shift Right	Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7)	Z,C,N,V	1
ROR	Rd	Rotate Right Through Carry	Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	Rd(n) \leftarrow Rd(n+1), n=0..6	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	Rd(3..0) \leftarrow Rd(7..4), Rd(7..4) \leftarrow Rd(3..0)	None	1
BSET	s	Flag Set	SREG(s) \leftarrow 1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) \leftarrow 0	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	T \leftarrow Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) \leftarrow T	None	1
SEC		Set Carry	C \leftarrow 1	C	1
CLC		Clear Carry	C \leftarrow 0	C	1
SEN		Set Negative Flag	N \leftarrow 1	N	1
CLN		Clear Negative Flag	N \leftarrow 0	N	1
SEZ		Set Zero Flag	Z \leftarrow 1	Z	1
CLZ		Clear Zero Flag	Z \leftarrow 0	Z	1
SEI		Global Interrupt Enable	I \leftarrow 1	I	1
CLI		Global Interrupt Disable	I \leftarrow 0	I	1
SES		Set Signed Test Flag	S \leftarrow 1	S	1
CLS		Clear Signed Test Flag	S \leftarrow 0	S	1
SEV		Set Twos Complement Overflow.	V \leftarrow 1	V	1
CLV		Clear Twos Complement Overflow	V \leftarrow 0	V	1
SET		Set T in SREG	T \leftarrow 1	T	1
CLT		Clear T in SREG	T \leftarrow 0	T	1
SEH		Set Half Carry Flag in SREG	H \leftarrow 1	H	1
CLH		Clear Half Carry Flag in SREG	H \leftarrow 0	H	1
DATA TRANSFER INSTRUCTIONS					
MOV	Rd, Rr	Move Between Registers	Rd \leftarrow Rr	None	1
MOVW	Rd, Rr	Copy Register Word	Rd+1:Rd \leftarrow Rr+1:Rr	None	1
LDI	Rd, K	Load Immediate	Rd \leftarrow K	None	1
LD	Rd, X	Load Indirect	Rd \leftarrow (X)	None	2
LD	Rd, X+	Load Indirect and Post-Inc.	Rd \leftarrow (X), X \leftarrow X + 1	None	2
LD	Rd, -X	Load Indirect and Pre-Dec.	X \leftarrow X - 1, Rd \leftarrow (X)	None	2
LD	Rd, Y	Load Indirect	Rd \leftarrow (Y)	None	2
LD	Rd, Y+	Load Indirect and Post-Inc.	Rd \leftarrow (Y), Y \leftarrow Y + 1	None	2
LD	Rd, -Y	Load Indirect and Pre-Dec.	Y \leftarrow Y - 1, Rd \leftarrow (Y)	None	2
LDD	Rd,Y+q	Load Indirect with Displacement	Rd \leftarrow (Y + q)	None	2
LD	Rd, Z	Load Indirect	Rd \leftarrow (Z)	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	Rd \leftarrow (Z), Z \leftarrow Z + 1	None	2
LD	Rd, -Z	Load Indirect and Pre-Dec.	Z \leftarrow Z - 1, Rd \leftarrow (Z)	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	Rd \leftarrow (Z + q)	None	2
LDS	Rd, k	Load Direct from SRAM	Rd \leftarrow (k)	None	2
ST	X, Rr	Store Indirect	(X) \leftarrow Rr	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	(X) \leftarrow Rr, X \leftarrow X + 1	None	2
ST	-X, Rr	Store Indirect and Pre-Dec.	X \leftarrow X - 1, (X) \leftarrow Rr	None	2
ST	Y, Rr	Store Indirect	(Y) \leftarrow Rr	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	(Y) \leftarrow Rr, Y \leftarrow Y + 1	None	2
ST	-Y, Rr	Store Indirect and Pre-Dec.	Y \leftarrow Y - 1, (Y) \leftarrow Rr	None	2
STD	Y+q,Rr	Store Indirect with Displacement	(Y + q) \leftarrow Rr	None	2
ST	Z, Rr	Store Indirect	(Z) \leftarrow Rr	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	(Z) \leftarrow Rr, Z \leftarrow Z + 1	None	2
ST	-Z, Rr	Store Indirect and Pre-Dec.	Z \leftarrow Z - 1, (Z) \leftarrow Rr	None	2
STD	Z+q,Rr	Store Indirect with Displacement	(Z + q) \leftarrow Rr	None	2
STS	k, Rr	Store Direct to SRAM	(k) \leftarrow Rr	None	2
LPM		Load Program Memory	R0 \leftarrow (Z)	None	3
LPM	Rd, Z	Load Program Memory	Rd \leftarrow (Z)	None	3
LPM	Rd, Z+	Load Program Memory and Post-Inc	Rd \leftarrow (Z), Z \leftarrow Z + 1	None	3
SPM		Store Program Memory	(Z) \leftarrow R1:R0	None	-
IN	Rd, P	In Port	Rd \leftarrow P	None	1
OUT	P, Rr	Out Port	P \leftarrow Rr	None	1
PUSH	Rr	Push Register on Stack	STACK \leftarrow Rr	None	2





Mnemonics	Operands	Description	Operation	Flags	#Clocks
POP	Rd	Pop Register from Stack	Rd ← STACK	None	2
MCU CONTROL INSTRUCTIONS					
NOP		No Operation		None	1
SLEEP		Sleep	(see specific descr. for Sleep function)	None	1
WDR		Watchdog Reset	(see specific descr. for WDR/timer)	None	1
BREAK		Break	For On-chip Debug Only	None	N/A

Note: 1. These instructions are only available in ATmega168P and ATmega328P.

7. Ordering Information

7.1 ATmega48P

Speed (MHz)	Power Supply	Ordering Code ⁽²⁾	Package ⁽¹⁾	Operational Range
10 ⁽³⁾	1.8 - 5.5	ATmega48PV-10AU ATmega48PV-10MMU ATmega48PV-10MU ATmega48PV-10PU	32A 28M1 32M1-A 28P3	Industrial (-40°C to 85°C)
20 ⁽³⁾	2.7 - 5.5	ATmega48P-20AU ATmega48P-20MMU ATmega48P-20MU ATmega48P-20PU	32A 28M1 32M1-A 28P3	Industrial (-40°C to 85°C)

Note:

1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive).Also Halide free and fully Green.
3. See [Figure 28-1 on page 317](#) and [Figure 28-2 on page 318](#).

Package Type	
32A	32-lead, Thin (1.0 mm) Plastic Quad Flat Package (TQFP)
28M1	28-pad, 4 x 4 x 1.0 body, Lead Pitch 0.45 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
32M1-A	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
28P3	28-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)



7.2 ATmega88P

Speed (MHz)	Power Supply	Ordering Code ⁽²⁾	Package ⁽¹⁾	Operational Range
10 ⁽³⁾	1.8 - 5.5	ATmega88PV-10AU ATmega88PV-10MU ATmega88PV-10PU	32A 32M1-A 28P3	Industrial (-40°C to 85°C)
20 ⁽³⁾	2.7 - 5.5	ATmega88P-20AU ATmega88P-20MU ATmega88P-20PU	32A 32M1-A 28P3	Industrial (-40°C to 85°C)

Note:

1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive).Also Halide free and fully Green.
3. See [Figure 28-1 on page 317](#) and [Figure 28-2 on page 318](#).

Package Type	
32A	32-lead, Thin (1.0 mm) Plastic Quad Flat Package (TQFP)
28P3	28-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
32M1-A	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)

ATmega48P/88P/168P/328P

7.3 ATmega168P

Speed (MHz) ⁽³⁾	Power Supply	Ordering Code ⁽²⁾	Package ⁽¹⁾	Operational Range
10	1.8 - 5.5	ATmega168PV-10AU ATmega168PV-10MU ATmega168PV-10PU	32A 32M1-A 28P3	Industrial (-40°C to 85°C)
20	2.7 - 5.5	ATmega168P-20AU ATmega168P-20MU ATmega168P-20PU	32A 32M1-A 28P3	Industrial (-40°C to 85°C)

Note:

1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive).Also Halide free and fully Green.
3. See [Figure 28-1 on page 317](#) and [Figure 28-2 on page 318](#).

Package Type	
32A	32-lead, Thin (1.0 mm) Plastic Quad Flat Package (TQFP)
28P3	28-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
32M1-A	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)



7.4 ATmega328P

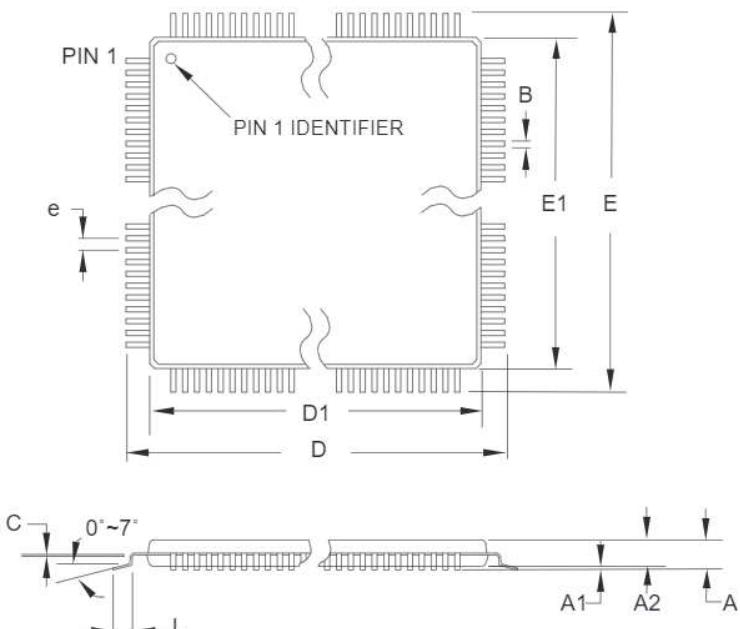
Speed (MHz) ⁽³⁾	Power Supply	Ordering Code ⁽²⁾	Package ⁽¹⁾	Operational Range
20	1.8 - 5.5	ATmega328P- AU ATmega328P- MU ATmega328P- PU	32A 32M1-A 28P3	Industrial (-40°C to 85°C)

- Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
 2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive).Also Halide free and fully Green.
 3. See [Figure 28-3 on page 318](#).

Package Type	
32A	32-lead, Thin (1.0 mm) Plastic Quad Flat Package (TQFP)
28P3	28-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
32M1-A	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)

8. Packaging Information

8.1 32A



COMMON DIMENSIONS
(Unit of Measure = mm)

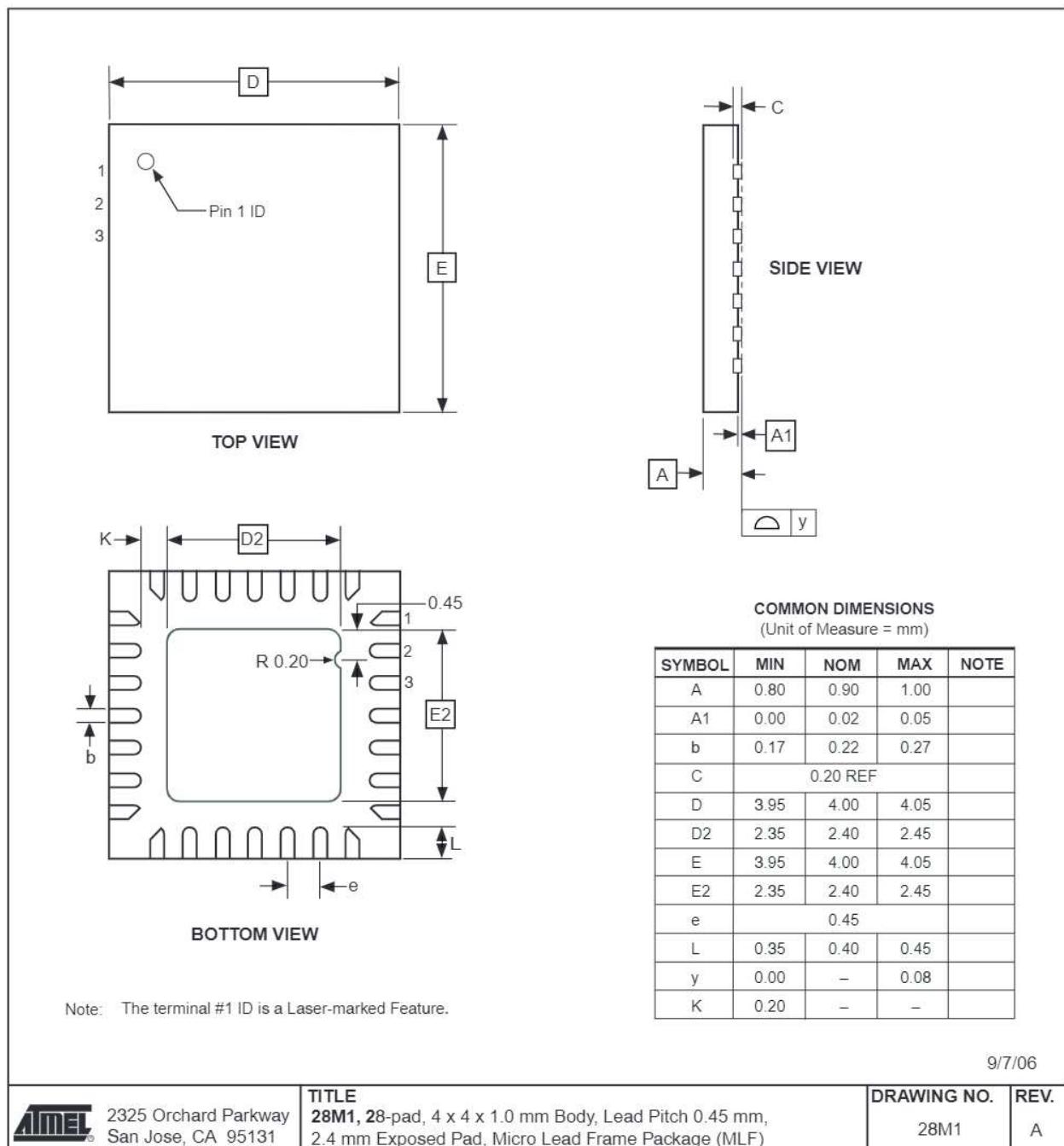
SYMBOL	MIN	NOM	MAX	NOTE
A	—	—	1.20	
A1	0.05	—	0.15	
A2	0.95	1.00	1.05	
D	8.75	9.00	9.25	
D1	6.90	7.00	7.10	Note 2
E	8.75	9.00	9.25	
E1	6.90	7.00	7.10	Note 2
B	0.30	—	0.45	
C	0.09	—	0.20	
L	0.45	—	0.75	
e	0.80 TYP			

- Notes:
1. This package conforms to JEDEC reference MS-026, Variation ABA.
 2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25 mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
 3. Lead coplanarity is 0.10 mm maximum.

10/5/2001

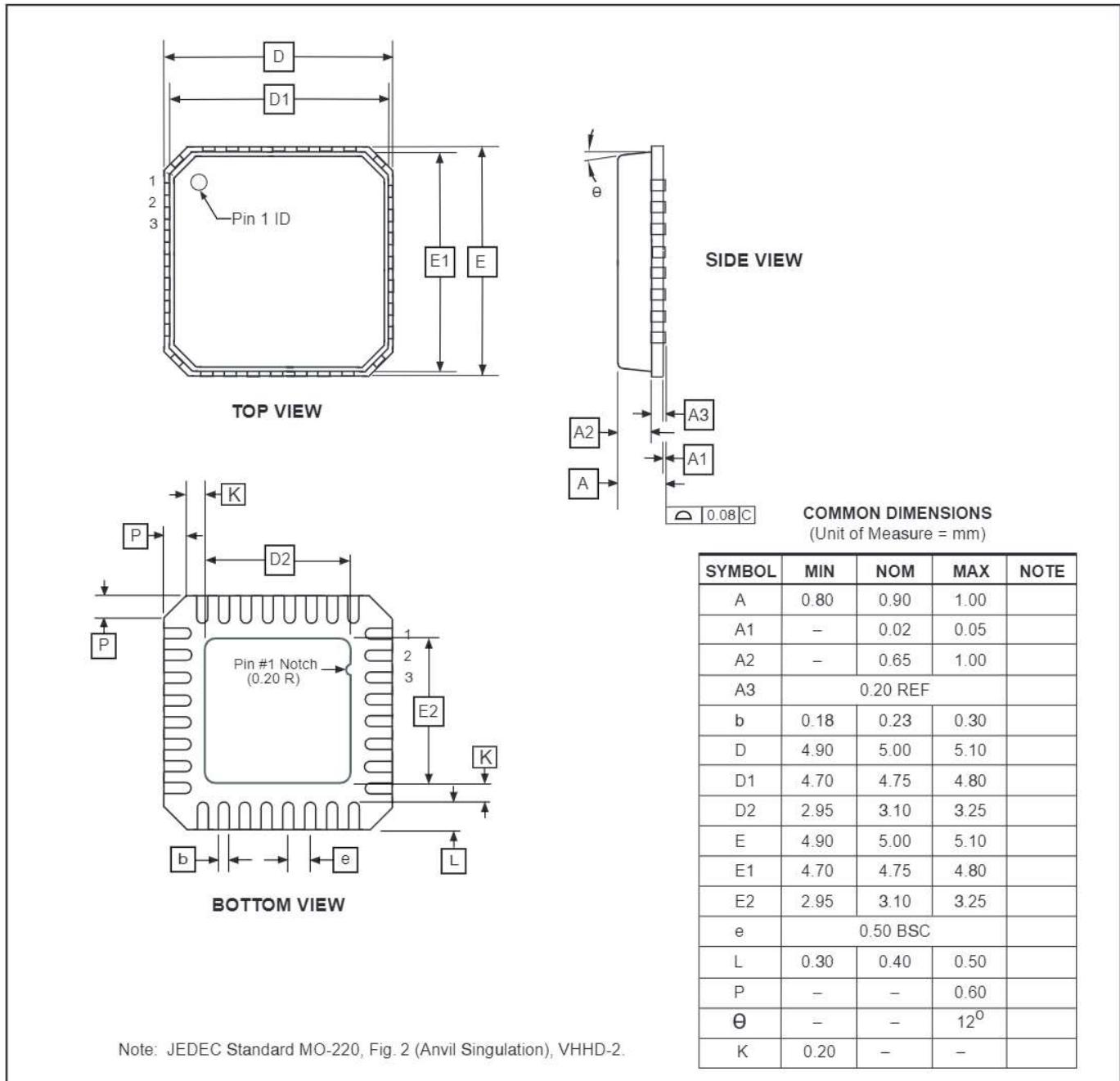
ATMEL 2325 Orchard Parkway San Jose, CA 95131	TITLE 32A, 32-lead, 7 x 7 mm Body Size, 1.0 mm Body Thickness, 0.8 mm Lead Pitch, Thin Profile Plastic Quad Flat Package (TQFP)	DRAWING NO. 32A	REV. B
--	--	---------------------------	------------------

8.2 28M1



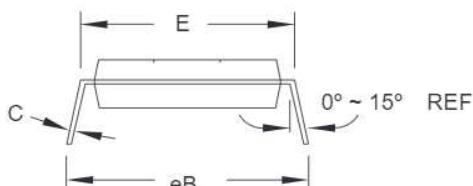
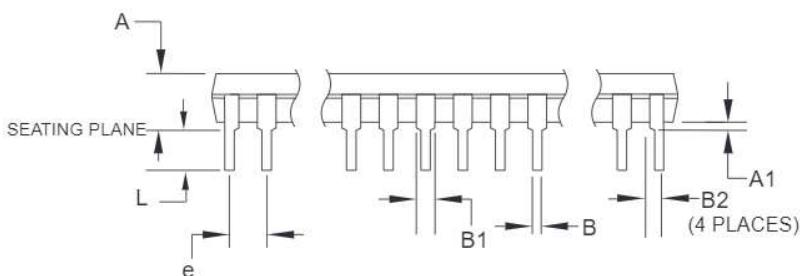
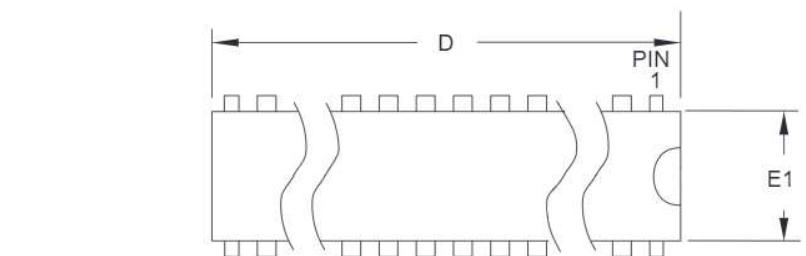
ATmega48P/88P/168P/328P

8.3 32M1-A



5/25/06

AMTEL	2325 Orchard Parkway San Jose, CA 95131	TITLE 32M1-A, 32-pad, 5 x 5 x 1.0 mm Body, Lead Pitch 0.50 mm, 3.10 mm Exposed Pad, Micro Lead Frame Package (MLF)	DRAWING NO. 32M1-A	REV. E
-------	--	--	-----------------------	-----------

8.4 28P3


COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	-	-	4.5724	
A1	0.508	-	-	
D	34.544	-	34.798	Note 1
E	7.620	-	8.255	
E1	7.112	-	7.493	Note 1
B	0.381	-	0.533	
B1	1.143	-	1.397	
B2	0.762	-	1.143	
L	3.175	-	3.429	
C	0.203	-	0.356	
eB	-	-	10.160	
e		2.540 TYP		

Note: 1. Dimensions D and E1 do not include mold Flash or Protrusion.
Mold Flash or Protrusion shall not exceed 0.25 mm (0.010").

09/28/01

ATMEL® 2325 Orchard Parkway San Jose, CA 95131	TITLE 28P3, 28-lead (0.300"/7.62 mm Wide) Plastic Dual Inline Package (PDIP)	DRAWING NO.	REV.
		28P3	B

9. Errata

9.1 Errata ATmega48P

The revision letter in this section refers to the revision of the ATmega48P device.

9.1.1 Rev. B

No known errata.

9.1.2 Rev. A

Not Sampled.

9.2 Errata ATmega88P

The revision letter in this section refers to the revision of the ATmega88P device.

9.2.1 Rev. A

No known errata.

9.3 Errata ATmega168P

The revision letter in this section refers to the revision of the ATmega168P device.

9.3.1 Rev A

No known errata.

9.4 Errata ATmega328P

The revision letter in this section refers to the revision of the ATmega328P device.

9.4.1 Rev B

- Unstable 32 kHz Oscillator

1. Unstable 32 kHz Oscillator

The 32 kHz oscillator does not work as system clock.

The 32 kHz oscillator used as asynchronous timer is inaccurate.

Problem Fix/ Workaround

None

9.4.2 Rev A

No known errata.



10. Datasheet Revision History

Please note that the referring page numbers in this section are referred to this document. The referring revision in this section are referring to the document revision.

10.1 Rev. 2545F-08/08

1. Updated "ATmega328P Typical Characteristics" on page 401 with Power-save numbers.
2. Added ATmega328P "Standby Supply Current" on page 408.

10.2 Rev. 2545E-08/08

1. Updated description of "Stack Pointer" on page 12.
2. Updated description of use of external capacitors in "Low Frequency Crystal Oscillator" on page 32.
3. Updated Table 8-9 in "Low Frequency Crystal Oscillator" on page 32.
4. Added note to "Address Match Unit" on page 222.
5. Added section "Reading the Signature Row from Software" on page 286.
6. Updated "Program And Data Memory Lock Bits" on page 295 to include ATmega328P in the description.
7. Added "ATmega328P DC Characteristics" on page 317.
8. Updated "Speed Grades" on page 317 for ATmega328P.
9. Removed note 6 and 7 from the table "2-wire Serial Interface Characteristics" on page 323.
10. Added figure "Minimum Reset Pulse width vs. V_{CC}." on page 352 for ATmega48P.
11. Added figure "Minimum Reset Pulse width vs. V_{CC}." on page 376 for ATmega88P.
12. Added figure "Minimum Reset Pulse width vs. V_{CC}." on page 400 for ATmega168P.
13. Added "ATmega328P Typical Characteristics" on page 401.
14. Updated Ordering Information for "ATmega328P" on page 18.

10.3 Rev. 2545D-03/08

1. Updated figures in "Speed Grades" on page 317.
2. Updated note in Table 28-4 in "System and Reset Characteristics" on page 320.
3. Ordering codes for "ATmega328P" on page 18 updated.
 - ATmega328P is offered in 20 MHz option only.
4. Added Errata for ATmega328P rev. B, "Errata ATmega328P" on page 23.

10.4 Rev. 2545C-01/08

1. Power-save Maximum values removed form "ATmega48P DC Characteristics" on page 315, "ATmega88P DC Characteristics" on page 316, and "ATmega168P DC Characteristics" on page 316.

10.5 Rev. 2545B-01/08

1. Updated "Features" on page 1.
2. Added "Data Retention" on page 7.
3. Updated [Table 8-2](#) on page 28.
4. Removed "Low-frequency Crystal Oscillator Internal Load Capacitance" table from "Low Frequency Crystal Oscillator" on page 32.
5. Removed JTD bit from "MCUCR – MCU Control Register" on page 44.
6. Updated typical and general program setup for Reset and Interrupt Vector Addresses in "Interrupt Vectors in ATmega168P" on page 62 and "Interrupt Vectors in ATmega328P" on page 65.
7. Updated Interrupt Vectors Start Address in [Table 11-5](#) on page 63 and [Table 11-7](#) on page 66.
8. Updated "Temperature Measurement" on page 262.
9. Updated ATmega328P "Fuse Bits" on page 296.
10. Removed V_{OL3}/V_{OH3} rows from "DC Characteristics" on page 314.
11. Updated condition for V_{OL} in "DC Characteristics" on page 314.
12. Updated max value for V_{IL2} in "DC Characteristics" on page 314.
13. Added "ATmega48P DC Characteristics" on page 315, "ATmega88P DC Characteristics" on page 316, and "ATmega168P DC Characteristics" on page 316.
14. Updated "System and Reset Characteristics" on page 320.
15. Added "ATmega48P Typical Characteristics" on page 329, "ATmega88P Typical Characteristics" on page 353, and "ATmega168P Typical Characteristics" on page 377.
16. Updated note in "Instruction Set Summary" on page 12.

10.6 Rev. 2545A-07/07

1. Initial revision.



Headquarters

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

International

Atmel Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Product Contact

Web Site
www.atmel.com

Technical Support
avr@atmel.com

Sales Contact
www.atmel.com/contacts

Literature Requests
www.atmel.com/literature

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. EXCEPT AS SET FORTH IN ATTEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATTEL'S WEB SITE, ATTEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATTEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATTEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

© 2008 Atmel Corporation. All rights reserved. Atmel®, logo and combinations thereof, AVR® and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.

DATA SHEET

PCD8544
48 · 84 pixels matrix LCD
controller/driver

Product specification
File under Integrated Circuits, IC17

1999 Apr 12

Philips
Semiconductors



PHILIPS

48 · 84 pixels matrix LCD controller/driver**PCD8544****CONTENTS**

1	FEATURES	8	INSTRUCTIONS
2	GENERAL DESCRIPTION	8.1	Initialization
3	APPLICATIONS	8.2	Reset function
4	ORDERING INFORMATION	8.3	Function set
5	BLOCK DIAGRAM	8.3.1	Bit PD
6	PINNING	8.3.2	Bit V
6.1	Pin functions	8.3.3	Bit H
6.1.1	R0 to R47 row driver outputs	8.4	Display control
6.1.2	C0 to C83 column driver outputs	8.4.1	Bits D and E
6.1.3	V_{SS1}, V_{SS2} : negative power supply rails	8.5	Set Y address of RAM
6.1.4	V_{DD1}, V_{DD2} : positive power supply rails	8.6	Set X address of RAM
6.1.5	V_{LCD1}, V_{LCD2} : LCD power supply	8.7	Temperature control
6.1.6	T1, T2, T3 and T4: test pads	8.8	Bias value
6.1.7	SDIN: serial data line	8.9	Set V_{OP} value
6.1.8	SCLK: serial clock line	9	LIMITING VALUES
6.1.9	D/C: mode select	10	HANDLING
6.1.10	SCE: chip enable	11	DC CHARACTERISTICS
6.1.11	OSC: oscillator	12	AC CHARACTERISTICS
6.1.12	RES: reset	12.1	Serial interface
7	FUNCTIONAL DESCRIPTION	12.2	Reset
7.1	Oscillator	13	APPLICATION INFORMATION
7.2	Address Counter (AC)	14	BONDING PAD LOCATIONS
7.3	Display Data RAM (DDRAM)	14.1	Bonding pad information
7.4	Timing generator	14.2	Bonding pad location
7.5	Display address counter	15	TRAY INFORMATION
7.6	LCD row and column drivers	16	DEFINITIONS
7.7	Addressing	17	LIFE SUPPORT APPLICATIONS
7.7.1	Data structure		
7.8	Temperature compensation		

48 · 84 pixels matrix LCD controller/driver

PCD8544

1 FEATURES

- Single chip LCD controller/driver
- 48 row, 84 column outputs
- Display data RAM 48 · 84 bits
- On-chip:
 - Generation of LCD supply voltage (external supply also possible)
 - Generation of intermediate LCD bias voltages
 - Oscillator requires no external components (external clock also possible).
- External $\overline{\text{RES}}$ (reset) input pin
- Serial interface maximum 4.0 Mbits/s
- CMOS compatible inputs
- Mux rate: 48
- Logic supply voltage range V_{DD} to V_{SS} : 2.7 to 3.3 V
- Display supply voltage range V_{LCD} to V_{SS}
 - 6.0 to 8.5 V with LCD voltage internally generated (voltage generator enabled)
 - 6.0 to 9.0 V with LCD voltage externally supplied (voltage generator switched-off).
- Low power consumption, suitable for battery operated systems
- Temperature compensation of V_{LCD}
- Temperature range: -25 to +70 °C.

2 GENERAL DESCRIPTION

The PCD8544 is a low power CMOS LCD controller/driver, designed to drive a graphic display of 48 rows and 84 columns. All necessary functions for the display are provided in a single chip, including on-chip generation of LCD supply and bias voltages, resulting in a minimum of external components and low power consumption.

The PCD8544 interfaces to microcontrollers through a serial bus interface.

The PCD8544 is manufactured in n-well CMOS technology.

3 APPLICATIONS

- Telecommunications equipment.

4 ORDERING INFORMATION

TYPE NUMBER	PACKAGE		
	NAME	DESCRIPTION	VERSION
PCD8544U	-	chip with bumps in tray; 168 bonding pads + 4 dummy pads	-

48 · 84 pixels matrix LCD controller/driver

PCD8544

5 BLOCK DIAGRAM

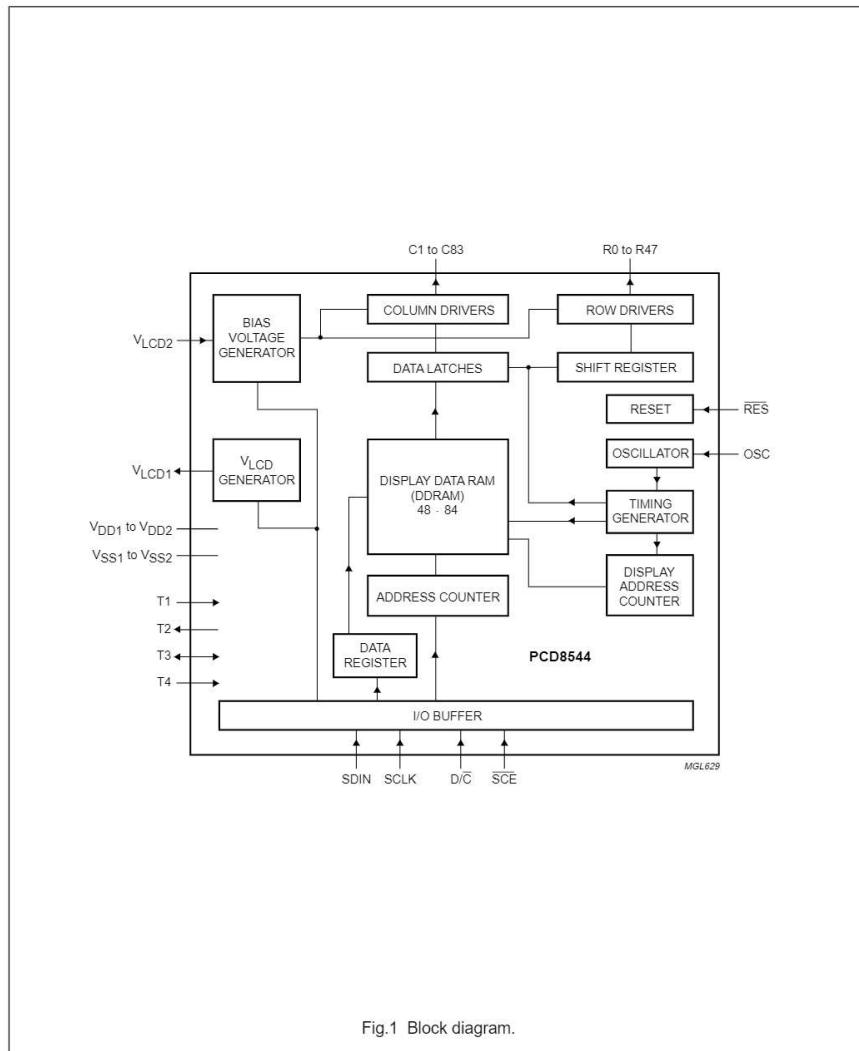


Fig.1 Block diagram.

48 · 84 pixels matrix LCD controller/driver

PCD8544

6 PINNING

SYMBOL	DESCRIPTION
R0 to R47	LCD row driver outputs
C0 to C83	LCD column driver outputs
V_{SS1}, V_{SS2}	ground
V_{DD1}, V_{DD2}	supply voltage
V_{LCD1}, V_{LCD2}	LCD supply voltage
T1	test 1 input
T2	test 2 output
T3	test 3 input/output
T4	test 4 input
SDIN	serial data input
SCLK	serial clock input
D/C	data/command
SCE	chip enable
OSC	oscillator
RES	external reset input
dummy1, 2, 3, 4	not connected

Note

1. For further details, see Fig.18 and Table 7.

6.1 Pin functions**6.1.1 R0 TO R47 ROW DRIVER OUTPUTS**

These pads output the row signals.

6.1.2 C0 TO C83 COLUMN DRIVER OUTPUTS

These pads output the column signals.

6.1.3 V_{SS1}, V_{SS2} : NEGATIVE POWER SUPPLY RAILS

Supply rails V_{SS1} and V_{SS2} must be connected together.

6.1.4 V_{DD1}, V_{DD2} : POSITIVE POWER SUPPLY RAILS

Supply rails V_{DD1} and V_{DD2} must be connected together.

6.1.5 V_{LCD1}, V_{LCD2} : LCD POWER SUPPLY

Positive power supply for the liquid crystal display. Supply rails V_{LCD1} and V_{LCD2} must be connected together.

6.1.6 T1, T2, T3 AND T4: TEST PADS

T1, T3 and T4 must be connected to V_{SS} , T2 is to be left open. Not accessible to user.

6.1.7 SDIN: SERIAL DATA LINE

Input for the data line.

6.1.8 SCLK: SERIAL CLOCK LINE

Input for the clock signal: 0.0 to 4.0 Mbits/s.

6.1.9 D/C: MODE SELECT

Input to select either command/address or data input.

6.1.10 SCE: CHIP ENABLE

The enable pin allows data to be clocked in. The signal is active LOW.

6.1.11 OSC: OSCILLATOR

When the on-chip oscillator is used, this input must be connected to V_{DD} . An external clock signal, if used, is connected to this input. If the oscillator and external clock are both inhibited by connecting the OSC pin to V_{SS} , the display is not clocked and may be left in a DC state. To avoid this, the chip should always be put into Power-down mode before stopping the clock.

6.1.12 RES: RESET

This signal will reset the device and must be applied to properly initialize the chip. The signal is active LOW.

48 . 84 pixels matrix LCD controller/driverPCD8544

7 FUNCTIONAL DESCRIPTION**7.1 Oscillator**

The on-chip oscillator provides the clock signal for the display system. No external components are required and the OSC input must be connected to V_{DD}. An external clock signal, if used, is connected to this input.

7.2 Address Counter (AC)

The address counter assigns addresses to the display data RAM for writing. The X-address X₆ to X₀ and the Y-address Y₂ to Y₀ are set separately. After a write operation, the address counter is automatically incremented by 1, according to the V flag.

7.3 Display Data RAM (DDRAM)

The DDRAM is a 48 . 84 bit static RAM which stores the display data. The RAM is divided into six banks of 84 bytes (6 . 8 . 84 bits). During RAM access, data is transferred to the RAM through the serial interface. There is a direct correspondence between the X-address and the column output number.

7.4 Timing generator

The timing generator produces the various signals required to drive the internal circuits. Internal chip operation is not affected by operations on the data buses.

7.5 Display address counter

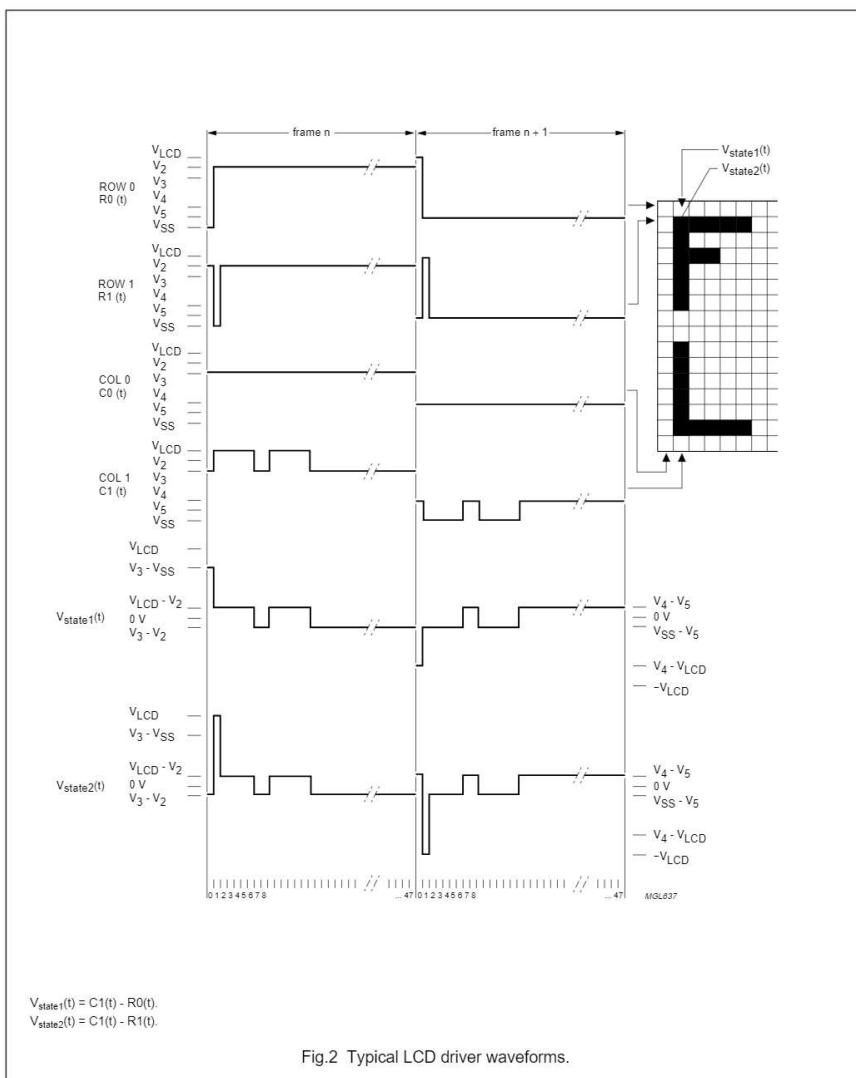
The display is generated by continuously shifting rows of RAM data to the dot matrix LCD through the column outputs. The display status (all dots on/off and normal/inverse video) is set by bits E and D in the 'display control' command.

7.6 LCD row and column drivers

The PCD8544 contains 48 row and 84 column drivers, which connect the appropriate LCD bias voltages in sequence to the display in accordance with the data to be displayed. Figure 2 shows typical waveforms. Unused outputs should be left unconnected.

48 · 84 pixels matrix LCD controller/driver

PCD8544



$V_{state1}(t) = C_1(t) - R_0(t)$.
 $V_{state2}(t) = C_1(t) - R_1(t)$.

Fig.2 Typical LCD driver waveforms.

48 · 84 pixels matrix LCD controller/driver

PCD8544

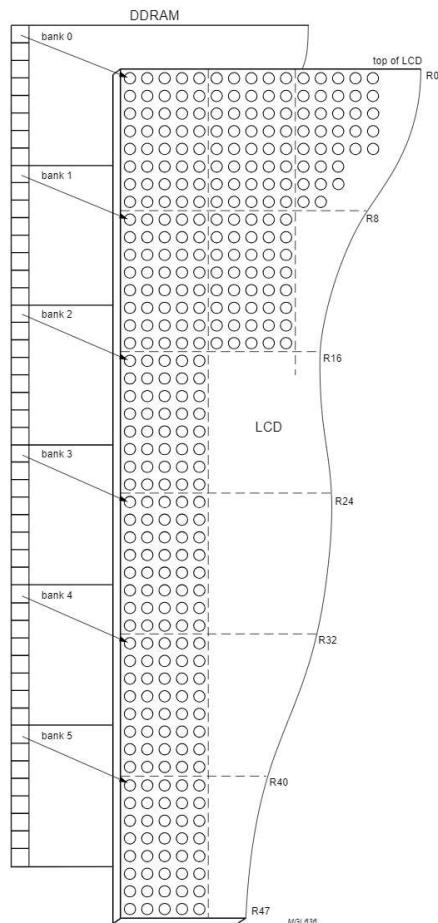


Fig.3 DDRAM to display mapping.

48 · 84 pixels matrix LCD controller/driver

PCD8544

7.7 Addressing

Data is downloaded in bytes into the 48 by 84 bits RAM data display matrix of PCD8544, as indicated in Figs. 3, 4, 5 and 6. The columns are addressed by the address pointer. The address ranges are: X 0 to 83 (1010011), Y 0 to 5 (101). Addresses outside these ranges are not allowed. In the vertical addressing mode ($V = 1$), the Y address increments after each byte (see

Fig.5). After the last Y address ($Y = 5$), Y wraps around to 0 and X increments to address the next column. In the horizontal addressing mode ($V = 0$), the X address increments after each byte (see Fig.6). After the last X address ($X = 83$), X wraps around to 0 and Y increments to address the next row. After the very last address ($X = 83$ and $Y = 5$), the address pointers wrap around to address ($X = 0$ and $Y = 0$).

7.7.1 DATA STRUCTURE

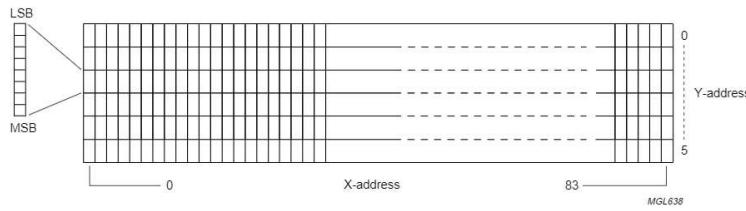
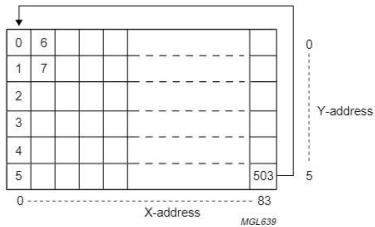
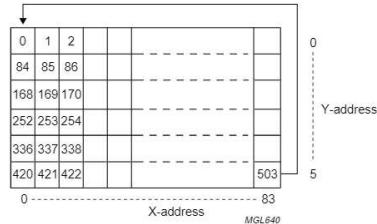


Fig.4 RAM format, addressing.

Fig.5 Sequence of writing data bytes into RAM with vertical addressing ($V = 1$).

48 × 84 pixels matrix LCD controller/driver

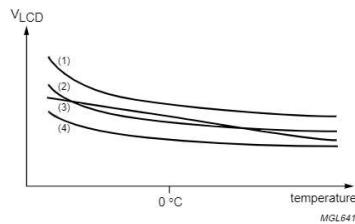
PCD8544

Fig.6 Sequence of writing data bytes into RAM with horizontal addressing ($V = 0$).

7.8 Temperature compensation

Due to the temperature dependency of the liquid crystals' viscosity, the LCD controlling voltage V_{LCD} must be increased at lower temperatures to maintain optimum

contrast. Figure 7 shows V_{LCD} for high multiplex rates. In the PCD8544, the temperature coefficient of V_{LCD} , can be selected from four values (see Table 2) by setting bits TC₁ and TC₀.



- (1) Upper limit.
- (2) Typical curve.
- (3) Temperature coefficient of IC.
- (4) Lower limit.

Fig.7 V_{LCD} as function of liquid crystal temperature (typical values).

48 · 84 pixels matrix LCD controller/driver

PCD8544

8 INSTRUCTIONS

The instruction format is divided into two modes: If D/C (mode select) is set LOW, the current byte is interpreted as command byte (see Table 1). Figure 8 shows an example of a serial data stream for initializing the chip. If D/C is set HIGH, the following bytes are stored in the display data RAM. After every data byte, the address counter is incremented automatically.

The level of the D/C signal is read during the last bit of data byte.

Each instruction can be sent in any order to the PCD8544. The MSB of a byte is transmitted first. Figure 9 shows one possible command stream, used to set up the LCD driver.

The serial interface is initialized when SCE is HIGH. In this state, SCLK clock pulses have no effect and no power is consumed by the serial interface. A negative edge on SCE enables the serial interface and indicates the start of a data transmission.

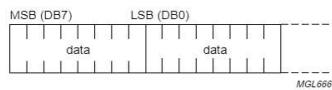


Fig.8 General format of data stream.

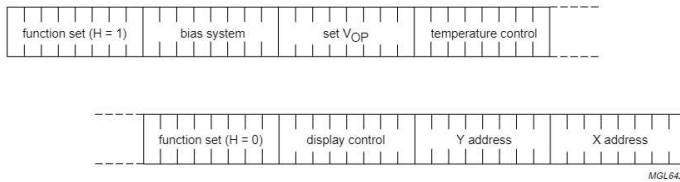


Fig.9 Serial data stream, example.

Figures 10 and 11 show the serial bus protocol.

- When SCE is HIGH, SCLK clock signals are ignored; during the HIGH time of SCE, the serial interface is initialized (see Fig.12)
- SDIN is sampled at the positive edge of SCLK
- D/C indicates whether the byte is a command (D/C = 0) or RAM data (D/C = 1); it is read with the eighth SCLK pulse
- If SCE stays LOW after the last bit of a command/data byte, the serial interface expects bit 7 of the next byte at the next positive edge of SCLK (see Fig.12)
- A reset pulse with RES interrupts the transmission. No data is written into the RAM. The registers are cleared. If SCE is LOW after the positive edge of RES, the serial interface is ready to receive bit 7 of a command/data byte (see Fig.13).

48 · 84 pixels matrix LCD controller/driver

PCD8544

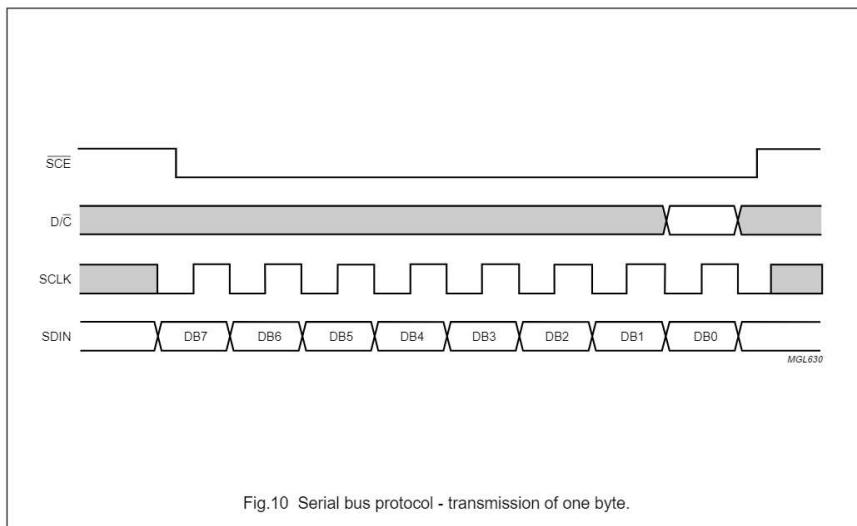


Fig.10 Serial bus protocol - transmission of one byte.

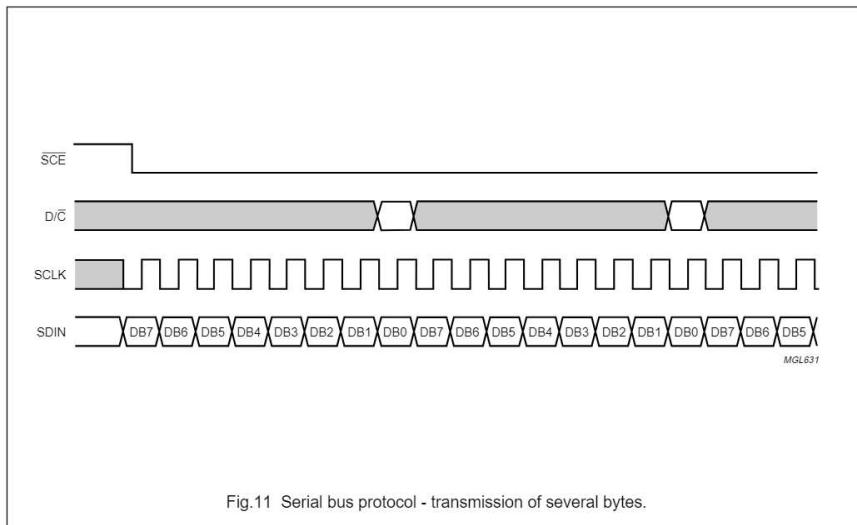
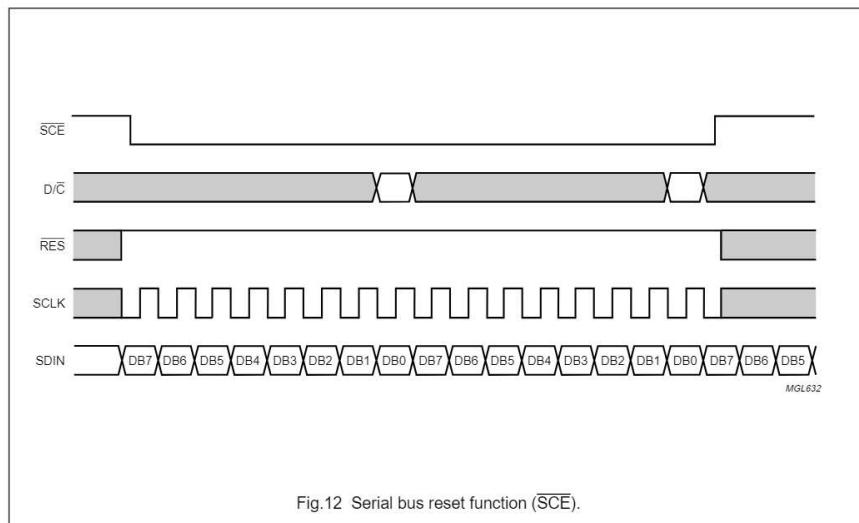
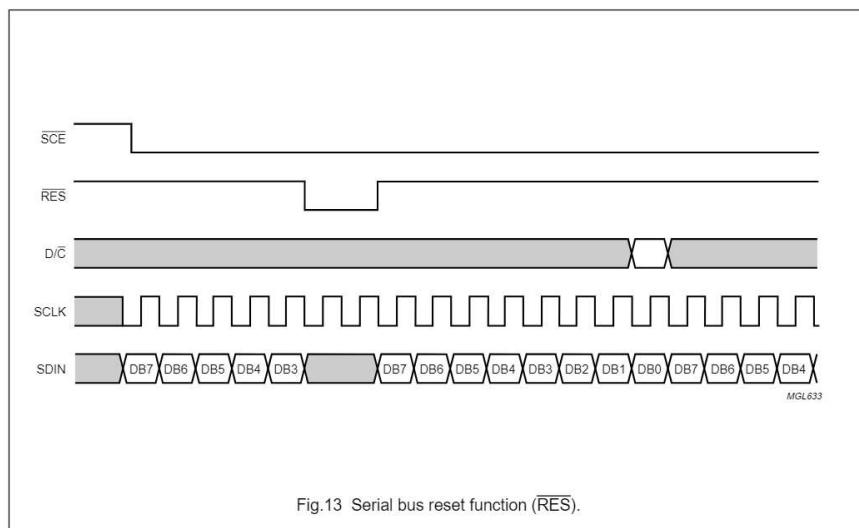


Fig.11 Serial bus protocol - transmission of several bytes.

48 · 84 pixels matrix LCD controller/driver

PCD8544

Fig.12 Serial bus reset function ($\overline{\text{SCE}}$).Fig.13 Serial bus reset function ($\overline{\text{RES}}$).

48 · 84 pixels matrix LCD controller/driver

PCD8544

Table 1 Instruction set

INSTRUCTION	D/C	COMMAND BYTE								DESCRIPTION
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
(H = 0 or 1)										
NOP	0	0	0	0	0	0	0	0	0	no operation
Function set	0	0	0	1	0	0	PD	V	H	power down control; entry mode; extended instruction set control (H)
Write data	1	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	writes data to display RAM
(H = 0)										
Reserved	0	0	0	0	0	0	1	X	X	do not use
Display control	0	0	0	0	0	1	D	0	E	sets display configuration
Reserved	0	0	0	0	1	X	X	X	X	do not use
Set Y address of RAM	0	0	1	0	0	0	Y ₂	Y ₁	Y ₀	sets Y-address of RAM; 0 ≤ Y ≤ 5
Set X address of RAM	0	1	X ₆	X ₅	X ₄	X ₃	X ₂	X ₁	X ₀	sets X-address part of RAM; 0 ≤ X ≤ 83
(H = 1)										
Reserved	0	0	0	0	0	0	0	0	1	do not use
	0	0	0	0	0	0	0	1	X	do not use
Temperature control	0	0	0	0	0	0	1	TC ₁	TC ₀	set Temperature Coefficient (TC _x)
Reserved	0	0	0	0	0	1	X	X	X	do not use
Bias system	0	0	0	0	1	0	BS ₂	BS ₁	BS ₀	set Bias System (BS _x)
Reserved	0	0	1	X	X	X	X	X	X	do not use
Set V _{OP}	0	1	V _{OP6}	V _{OP5}	V _{OP4}	V _{OP3}	V _{OP2}	V _{OP1}	V _{OP0}	write V _{OP} to register

Table 2 Explanations of symbols in Table 1

BIT	0	1
PD	chip is active	chip is in Power-down mode
V	horizontal addressing	vertical addressing
H	use basic instruction set	use extended instruction set
D and E		
00	display blank	
10	normal mode	
01	all display segments on	
11	inverse video mode	
TC ₁ and TC ₀		
00	V _{LCD} temperature coefficient 0	
01	V _{LCD} temperature coefficient 1	
10	V _{LCD} temperature coefficient 2	
11	V _{LCD} temperature coefficient 3	

48 · 84 pixels matrix LCD controller/driver

PCD8544

8.1 Initialization

Immediately following power-on, the contents of all internal registers and of the RAM are undefined. **A RES pulse must be applied.** Attention should be paid to the possibility that the **device may be damaged** if not properly reset.

All internal registers are reset by applying an external $\overline{\text{RES}}$ pulse (active LOW) at pad 31, within the specified time. However, the RAM contents are still undefined. The state after reset is described in Section 8.2.

The RES input must be $\leq 0.3V_{DD}$ when V_{DD} reaches $V_{DD\min}$ (or higher) within a maximum time of 100 ms after V_{DD} goes HIGH (see Fig.16).

8.2 Reset function

After reset, the LCD driver has the following state:

- Power-down mode (bit PD = 1)
- Horizontal addressing (bit V = 0) normal instruction set (bit H = 0)
- Display blank (bit E = D = 0)
- Address counter X_6 to X_0 = 0; Y_2 to Y_0 = 0
- Temperature control mode (TC_1 TC_0 = 0)
- Bias system (BS_2 to BS_0 = 0)
- V_{LCD} is equal to 0, the HV generator is switched off (V_{OP6} to V_{OP0} = 0)
- After power-on, the RAM contents are undefined.

8.3 Function set

8.3.1 Bit PD

- All LCD outputs at V_{SS} (display off)
- Bias generator and V_{LCD} generator off, V_{LCD} can be disconnected
- Oscillator off (external clock possible)
- Serial bus, command, etc. function
- Before entering Power-down mode, the RAM needs to be filled with '0's to ensure the specified current consumption.

8.3.2 Bit V

When V = 0, the horizontal addressing is selected. The data is written into the DDRAM as shown in Fig.6. When V = 1, the vertical addressing is selected. The data is written into the DDRAM, as shown in Fig.5.

8.3.3 Bit H

When H = 0 the commands 'display control', 'set Y address' and 'set X address' can be performed; when H = 1, the others can be executed. The 'write data' and 'function set' commands can be executed in both cases.

8.4 Display control

8.4.1 Bits D AND E

Bits D and E select the display mode (see Table 2).

8.5 Set Y address of RAM

Y_n defines the Y vector addressing of the display RAM.

Table 3 Y vector addressing

Y₂	Y₁	Y₀	BANK
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5

8.6 Set X address of RAM

The X address points to the columns. The range of X is 0 to 83 (53H).

8.7 Temperature control

The temperature coefficient of V_{LCD} is selected by bits TC_1 and TC_0 .

8.8 Bias value

The bias voltage levels are set in the ratio of R - R - nR - R - R, giving a 1/(n + 4) bias system. Different multiplex rates require different factors n (see Table 4). This is programmed by BS_2 to BS_0 . For Mux 1 : 48, the optimum bias value n, resulting in 1/8 bias, is given by:

$$n = \sqrt[4]{48} - 3 = 3.928 = 4 \quad (1)$$

48 × 84 pixels matrix LCD controller/driver

PCD8544

Table 4 Programming the required bias system

BS₂	BS₁	BS₀	n	RECOMMENDED MUX RATE
0	0	0	7	1 : 100
0	0	1	6	1 : 80
0	1	0	5	1 : 65/1 : 65
0	1	1	4	1 : 48
1	0	0	3	1 : 40/1 : 34
1	0	1	2	1 : 24
1	1	0	1	1 : 18/1 : 16
1	1	1	0	1 : 10/1 : 9/1 : 8

Table 5 LCD bias voltage

SYMBOL	BIAS VOLTAGES	BIAS VOLTAGE FOR $\frac{1}{8}$ BIAS
V1	V_{LCD}	V_{LCD}
V2	$(n + 3)/(n + 4)$	$\frac{7}{8} \cdot V_{LCD}$
V3	$(n + 2)/(n + 4)$	$\frac{6}{8} \cdot V_{LCD}$
V4	$2/(n + 4)$	$\frac{2}{8} \cdot V_{LCD}$
V5	$1/(n + 4)$	$\frac{1}{8} \cdot V_{LCD}$
V6	V_{SS}	V_{SS}

8.9 Set V_{OP} value

The operation voltage V_{LCD} can be set by software. The values are dependent on the liquid crystal selected. $V_{LCD} = a + (V_{OP6} - V_{OP0}) \cdot b$ [V]. In the PCD8544, $a = 3.06$ and $b = 0.06$ giving a program range of 3.00 to 10.68 at room temperature.

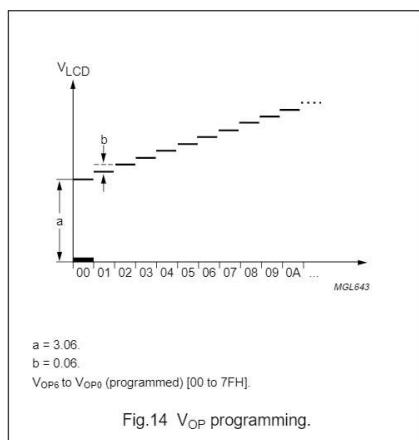
Note that the charge pump is turned off if V_{OP6} to V_{OP0} is set to zero.

For Mux 1 : 48, the optimum operation voltage of the liquid can be calculated as:

$$V_{LCD} = \frac{1 + \sqrt{48}}{2 \cdot \left(1 - \frac{1}{\sqrt{48}}\right)} \cdot V_{th} = 6.06 \cdot V_{th} \quad (2)$$

where V_{th} is the threshold voltage of the liquid crystal material used.

Caution, as V_{OP} increases with lower temperatures, care must be taken not to set a V_{OP} that will exceed the maximum of 8.5 V when operating at -25 °C.



48 · 84 pixels matrix LCD controller/driver

PCD8544

9 LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 134); see notes 1 and 2.

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
V_{DD}	supply voltage	note 3	-0.5	+7	V
V_{LCD}	supply voltage LCD	note 4	-0.5	+10	V
V_i	all input voltages		-0.5	$V_{DD} + 0.5$	V
I_{SS}	ground supply current		-50	+50	mA
I_I, I_O	DC input or output current		-10	+10	mA
P_{tot}	total power dissipation		-	300	mW
P_O	power dissipation per output		-	30	mW
T_{amb}	operating ambient temperature		-25	+70	°C
T_j	operating junction temperature		-65	+150	°C
T_{stg}	storage temperature		-65	+150	°C

Notes

1. Stresses above those listed under limiting values may cause permanent damage to the device.
2. Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to V_{SS} unless otherwise noted.
3. With external LCD supply voltage externally supplied (voltage generator disabled). $V_{DDmax} = 5$ V if LCD supply voltage is internally generated (voltage generator enabled).
4. When setting V_{LCD} by software, take care not to set a V_{OP} that will exceed the maximum of 8.5 V when operating at -25 °C, see Caution in Section 8.9.

10 HANDLING

Inputs and outputs are protected against electrostatic discharge in normal handling. However, to be totally safe, it is desirable to take normal precautions appropriate to handling MOS devices (see "Handling MOS devices").

11 DC CHARACTERISTICS $V_{DD} = 2.7$ to 3.3 V; $V_{SS} = 0$ V; $V_{LCD} = 6.0$ to 9.0 V; $T_{amb} = -25$ to $+70$ °C; unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
V_{DD1}	supply voltage 1	LCD voltage externally supplied (voltage generator disabled)	2.7	-	3.3	V
V_{DD2}	supply voltage 2	LCD voltage internally generated (voltage generator enabled)	2.7	-	3.3	V
V_{LCD1}	LCD supply voltage	LCD voltage externally supplied (voltage generator disabled)	6.0	-	9.0	V
V_{LCD2}	LCD supply voltage	LCD voltage internally generated (voltage generator enabled); note 1	6.0	-	8.5	V
I_{DD1}	supply current 1 (normal mode) for internal V_{LCD}	$V_{DD} = 2.85$ V; $V_{LCD} = 7.0$ V; $f_{SCLK} = 0$; $T_{amb} = 25$ °C; display load = $10 \text{ } \mu\text{A}$; note 2	-	240	300	mA
I_{DD2}	supply current 2 (normal mode) for internal V_{LCD}	$V_{DD} = 2.70$ V; $V_{LCD} = 7.0$ V; $f_{SCLK} = 0$; $T_{amb} = 25$ °C; display load = $10 \text{ } \mu\text{A}$; note 2	-	-	320	mA
I_{DD3}	supply current 3 (Power-down mode)	with internal or external LCD supply voltage; note 3	-	1.5	-	mA
I_{DD4}	supply current external V_{LCD}	$V_{DD} = 2.85$ V; $V_{LCD} = 9.0$ V; $f_{SCLK} = 0$; notes 2 and 4	-	25	-	mA
I_{LCD}	supply current external V_{LCD}	$V_{DD} = 2.7$ V; $V_{LCD} = 7.0$ V; $f_{SCLK} = 0$; $T = 25$ °C; display load = $10 \text{ } \mu\text{A}$; notes 2 and 4	-	42	-	mA
Logic						
V_{IL}	LOW level input voltage		V_{SS}	-	$0.3V_{DD}$	V
V_{IH}	HIGH level input voltage		$0.7V_{DD}$	-	V_{DD}	V
I_L	leakage current	$V_I = V_{DD}$ or V_{SS}	-1	-	+1	mA
Column and row outputs						
$R_{o(C)}$	column output resistance C0 to C83		-	12	20	kΩ
$R_{o(R)}$	row output resistance R0 to R47		-	12	20	kΩ
$V_{bias(tol)}$	bias voltage tolerance on C0 to C83 and R0 to R47		-100	0	+100	mV

48 · 84 pixels matrix LCD controller/driver

PCD8544

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
LCD supply voltage generator						
V _{LCD}	V _{LCD} tolerance internally generated	V _{DD} = 2.85 V; V _{LCD} = 7.0 V; f _{SCLK} = 0; display load = 10 μ A; note 5	-	0	300	mV
TC0	V _{LCD} temperature coefficient 0	V _{DD} = 2.85 V; V _{LCD} = 7.0 V; f _{SCLK} = 0; display load = 10 μ A	-	1	-	mV/K
TC1	V _{LCD} temperature coefficient 1	V _{DD} = 2.85 V; V _{LCD} = 7.0 V; f _{SCLK} = 0; display load = 10 μ A	-	9	-	mV/K
TC2	V _{LCD} temperature coefficient 2	V _{DD} = 2.85 V; V _{LCD} = 7.0 V; f _{SCLK} = 0; display load = 10 μ A	-	17	-	mV/K
TC3	V _{LCD} temperature coefficient 3	V _{DD} = 2.85 V; V _{LCD} = 7.0 V; f _{SCLK} = 0; display load = 10 μ A	-	24	-	mV/K

Notes

1. The maximum possible V_{LCD} voltage that may be generated is dependent on voltage, temperature and (display) load.
2. Internal clock.
3. RAM contents equal '0'. During power-down, all static currents are switched off.
4. If external V_{LCD}, the display load current is not transmitted to I_{DD}.
5. Tolerance depends on the temperature (typically zero at 27 °C, maximum tolerance values are measured at the temperate range limit).

12 AC CHARACTERISTICS

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
f_{osc}	oscillator frequency		20	34	65	kHz
$f_{clk(ext)}$	external clock frequency		10	32	100	kHz
t_{frame}	frame frequency	f_{osc} or $f_{clk(ext)} = 32$ kHz; note 1	-	67	-	Hz
t_{VHRL}	V_{DD} to \bar{RES} LOW	Fig.16	0(2)	-	30	ms
$t_{WL(RES)}$	\bar{RES} LOW pulse width	Fig.16	100	-	-	ns

Serial bus timing characteristics

t_{SCLK}	clock frequency	$V_{DD} = 3.0$ V $\pm 10\%$	0	-	4.00	MHz
T_{cy}	clock cycle SCLK	All signal timing is based on 20% to 80% of V_{DD} and maximum rise and fall times of 10 ns	250	-	-	ns
t_{WH1}	SCLK pulse width HIGH		100	-	-	ns
t_{WL1}	SCLK pulse width LOW		100	-	-	ns
t_{su2}	\bar{SCE} set-up time		60	-	-	ns
t_{h2}	\bar{SCE} hold time		100	-	-	ns
t_{WH2}	\bar{SCE} min. HIGH time		100	-	-	ns
t_{h5}	\bar{SCE} start hold time; note 3		100	-	-	ns
t_{su3}	D/\bar{C} set-up time		100	-	-	ns
t_{h3}	D/\bar{C} hold time		100	-	-	ns
t_{su4}	SDIN set-up time		100	-	-	ns
t_{h4}	SDIN hold time		100	-	-	ns

Notes

1. $T_{frame} = \frac{f_{clk(ext)}}{480}$
2. \bar{RES} may be LOW before V_{DD} goes HIGH.
3. t_{h5} is the time from the previous SCLK positive edge (irrespective of the state of \bar{SCE}) to the negative edge of \bar{SCE} (see Fig.15).

48 · 84 pixels matrix LCD controller/driver

PCD8544

12.1 Serial interface

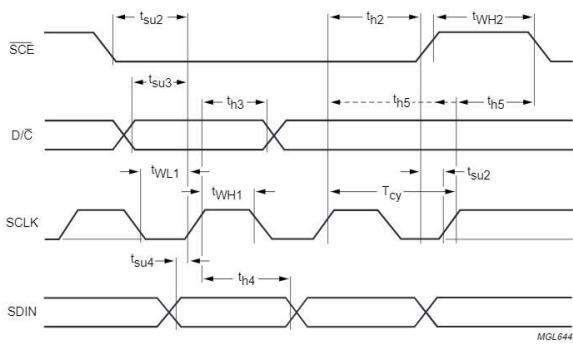


Fig.15 Serial interface timing.

12.2 Reset

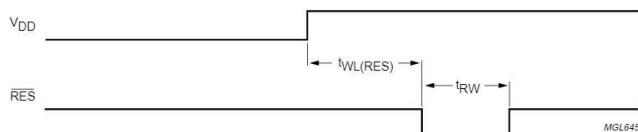
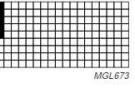
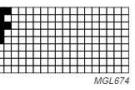
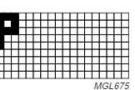
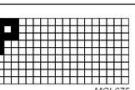
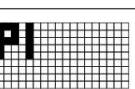


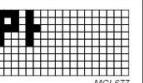
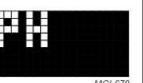
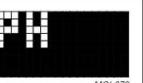
Fig.16 Reset timing.

13 APPLICATION INFORMATION**Table 6** Programming example

STEP	SERIAL BUS BYTE									DISPLAY	OPERATION
	D/C	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
1	start										SCE is going LOW
2	0	0	0	1	0	0	0	0	1		function set PD = 0 and V = 0, select extended instruction set (H = 1 mode)
3	0	1	0	0	1	0	0	0	0		set V _{OP} ; V _{OP} is set to a +16 · b [V]
4	0	0	0	1	0	0	0	0	0		function set PD = 0 and V = 0, select normal instruction set (H = 0 mode)
5	0	0	0	0	0	1	1	0	0		display control set normal mode (D = 1 and E = 0)
6	1	0	0	0	1	1	1	1	1	 MGL673	data write Y and X are initialized to 0 by default, so they are not set here
7	1	0	0	0	0	0	1	0	1	 MGL674	data write
8	1	0	0	0	0	0	1	1	1	 MGL675	data write
9	1	0	0	0	0	0	0	0	0	 MGL675	data write
10	1	0	0	0	1	1	1	1	1	 MGL676	data write

48 · 84 pixels matrix LCD controller/driver

PCD8544

STEP	SERIAL BUS BYTE								DISPLAY	OPERATION	
	D/C	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
11	1	0	0	0	0	0	1	0	0	 MGL677	data write
12	1	0	0	0	1	1	1	1	1	 MGL678	data write
13	0	0	0	0	0	1	1	0	1	 MGL679	display control; set inverse video mode (D = 1 and E = 1)
14	0	1	0	0	0	0	0	0	0	 MGL679	set X address of RAM; set address to '0000000'
15	1	0	0	0	0	0	0	0	0	 MGL680	data write

The pinning is optimized for single plane wiring e.g. for chip-on-glass display modules. Display size: 48 · 84 pixels.

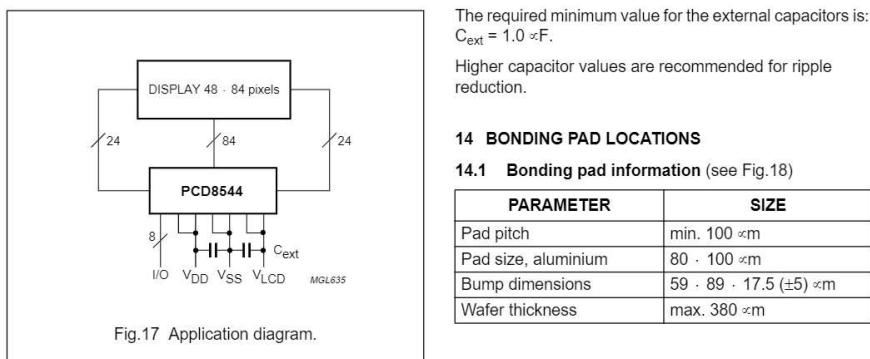


Fig.17 Application diagram.

48 · 84 pixels matrix LCD controller/driver

PCD8544

14.2 Bonding pad location

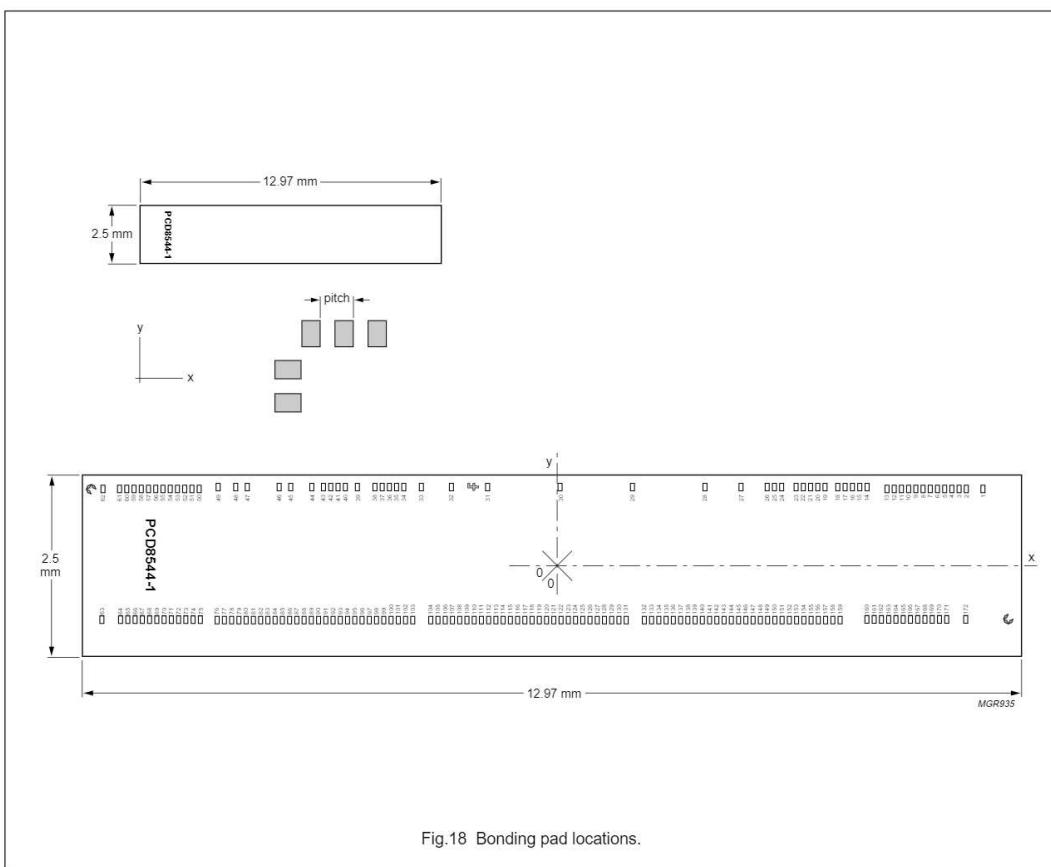


Fig.18 Bonding pad locations.

48 · 84 pixels matrix LCD controller/driver

PCD8544

Table 7 Bonding pad locations (dimensions in μm).
All X/Y coordinates are referenced to the centre
of chip (see Fig.18)

PAD	PAD NAME	x	y
1	dummy1	+5932	+1060
2	R36	+5704	+1060
3	R37	+5604	+1060
4	R38	+5504	+1060
5	R39	+5404	+1060
6	R40	+5304	+1060
7	R41	+5204	+1060
8	R42	+5104	+1060
9	R43	+5004	+1060
10	R44	+4904	+1060
11	R45	+4804	+1060
12	R46	+4704	+1060
13	R47	+4604	+1060
14	V _{DD1}	+4330	+1085
15	V _{DD1}	+4230	+1085
16	V _{DD1}	+4130	+1085
17	V _{DD1}	+4030	+1085
18	V _{DD1}	+3930	+1085
19	V _{DD2}	+3750	+1085
20	V _{DD2}	+3650	+1085
21	V _{DD2}	+3550	+1085
22	V _{DD2}	+3450	+1085
23	V _{DD2}	+3350	+1085
24	V _{DD2}	+3250	+1085
25	V _{DD2}	+3150	+1085
26	V _{DD2}	+3050	+1085
27	SCLK	+2590	+1085
28	SDIN	+2090	+1085
29	D/C	+1090	+1085
30	SCE	+90	+1085
31	RES	-910	+1085
32	OSC	-1410	+1085
33	T3	-1826	+1085
34	V _{SS2}	-2068	+1085
35	V _{SS2}	-2168	+1085
36	V _{SS2}	-2268	+1085
37	V _{SS2}	-2368	+1085
38	V _{SS2}	-2468	+1085

PAD	PAD NAME	x	y
39	T4	-2709	+1085
40	V _{SS1}	-2876	+1085
41	V _{SS1}	-2976	+1085
42	V _{SS1}	-3076	+1085
43	V _{SS1}	-3176	+1085
44	T1	-3337	+1085
45	V _{LCD2}	-3629	+1085
46	V _{LCD2}	-3789	+1085
47	V _{LCD1}	-4231	+1085
48	V _{LCD1}	-4391	+1085
49	T2	-4633	+1085
50	R23	-4894	+1060
51	R22	-4994	+1060
52	R21	-5094	+1060
53	R20	-5194	+1060
54	R19	-5294	+1060
55	R18	-5394	+1060
56	R17	-5494	+1060
57	R16	-5594	+1060
58	R15	-5694	+1060
59	R14	-5794	+1060
60	R13	-5894	+1060
61	R12	-5994	+1060
62	dummy2	-6222	+1060
63	dummy3	-6238	-738
64	R0	-5979	-738
65	R1	-5879	-738
66	R2	-5779	-738
67	R3	-5679	-738
68	R4	-5579	-738
69	R5	-5479	-738
70	R6	-5379	-738
71	R7	-5279	-738
72	R8	-5179	-738
73	R9	-5079	-738
74	R10	-4979	-738
75	R11	-4879	-738
76	C0	-4646	-746

48 · 84 pixels matrix LCD controller/driver

PCD8544

PAD	PAD NAME	x	y	PAD	PAD NAME	x	y
77	C1	-4546	-746	118	C42	-296	-746
78	C2	-4446	-746	119	C43	-196	-746
79	C3	-4346	-746	120	C44	-96	-746
80	C4	-4246	-746	121	C45	+4	-746
81	C5	-4146	-746	122	C46	+104	-746
82	C6	-4046	-746	123	C47	+204	-746
83	C7	-3946	-746	124	C48	+304	-746
84	C8	-3846	-746	125	C49	+404	-746
85	C9	-3746	-746	126	C50	+504	-746
86	C10	-3646	-746	127	C51	+604	-746
87	C11	-3546	-746	128	C52	+704	-746
88	C12	-3446	-746	139	C53	+804	-746
89	C13	-3346	-746	130	C54	+904	-746
90	C14	-3246	-746	131	C55	+1004	-746
91	C15	-3146	-746	132	C56	+1254	-746
92	C16	-3046	-746	133	C57	+1354	-746
93	C17	-2946	-746	134	C58	+1454	-746
94	C18	-2846	-746	135	C59	+1554	-746
95	C19	-2746	-746	136	C60	+1654	-746
96	C20	-2646	-746	137	C61	+1754	-746
97	C21	-2546	-746	138	C62	+1854	-746
98	C22	-2446	-746	139	C63	+1954	-746
99	C23	-2346	-746	140	C64	+2054	-746
100	C24	-2246	-746	141	C65	+2154	-746
101	C25	-2146	-746	142	C66	+2254	-746
102	C26	-2046	-746	143	C67	+2354	-746
103	C27	-1946	-746	144	C68	+2454	-746
104	C28	-1696	-746	145	C69	+2554	-746
105	C29	-1596	-746	146	C70	+2654	-746
106	C30	-1496	-746	147	C71	+2754	-746
107	C31	-1396	-746	148	C72	+2854	-746
108	C32	-1296	-746	149	C73	+2954	-746
109	C33	-1196	-746	150	C74	+3054	-746
110	C34	-1096	-746	151	C75	+3154	-746
111	C35	-996	-746	152	C76	+3254	-746
112	C36	-896	-746	153	C77	+3354	-746
113	C37	-796	-746	154	C78	+3454	-746
114	C38	-696	-746	155	C79	+3554	-746
115	C39	-596	-746	156	C80	+3654	-746
116	C40	-496	-746	157	C81	+3754	-746
117	C41	-396	-746	158	C82	+3854	-746

48 · 84 pixels matrix LCD controller/driver

PCD8544

PAD	PAD NAME	x	y
159	C83	+3954	-746
160	R35	+4328	-738
161	R34	+4428	-738
162	R33	+4528	-738
163	R32	+4628	-738
164	R31	+4728	-738
165	R30	+4828	-738
166	R29	+4928	-738
167	R28	+5028	-738
168	R27	+5128	-738
169	R26	+5228	-738
170	R25	+5328	-738
171	R24	+5428	-738
172	dummy4	+5694	-738

48 · 84 pixels matrix LCD controller/driver

PCD8544

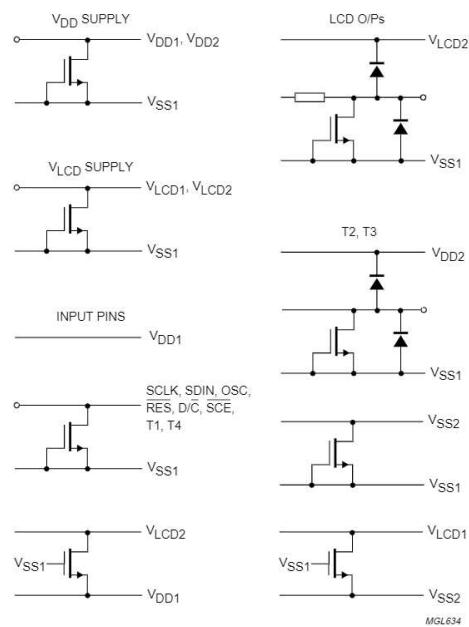
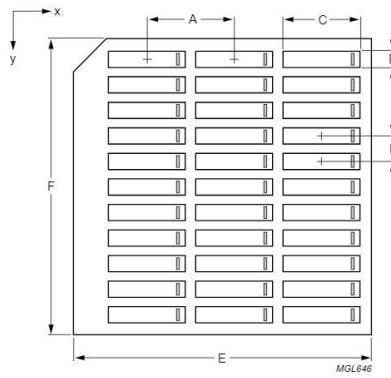


Fig.19 Device protection diagram.

15 TRAY INFORMATION

For the dimensions of x, y and A to F, see Table 8.

Fig.20 Tray details.

Table 8 Dimensions

DIM.	DESCRIPTION	VALUE
A	pocket pitch, in the x direction	14.82 mm
B	pocket pitch, in the y direction	4.39 mm
C	pocket width, in the x direction	13.27 mm
D	pocket width, in the y direction	2.8 mm
E	tray width, in the x direction	50.67 mm
F	tray width, in the y direction	50.67 mm
x	no. of pockets in the x direction	3
y	no. of pockets in the y direction	11

The orientation of the IC in a pocket is indicated by the position of the IC type name on the die surface with respect to the chamfer on the upper left corner of the tray. Refer to the bonding pad location diagram for the orientation and position of the type name on the die surface.

Fig.21 Tray alignment.

48 · 84 pixels matrix LCD controller/driver

PCD8544

16 DEFINITIONS

Data sheet status	
Objective specification	This data sheet contains target or goal specifications for product development.
Preliminary specification	This data sheet contains preliminary data; supplementary data may be published later.
Product specification	This data sheet contains final product specifications.
Limiting values	
Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.	
Application information	
Where application information is given, it is advisory and does not form part of the specification.	

17 LIFE SUPPORT APPLICATIONS

These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips for any damages resulting from such improper use or sale.

48 · 84 pixels matrix LCD controller/driver

PCD8544

NOTES

Philips Semiconductors – a worldwide company

Argentina: see South America

Australia: 34 Waterloo Road, NORTH RYDE, NSW 2113, Tel. +61 2 9805 4455, Fax. +61 2 9805 4466

Austria: Computerstr. 6, A-1101 WIEN, P.O. Box 213, Tel. +43 1 60 101 1248, Fax. +43 1 60 101 1210

Belarus: Hotel Minsk Business Center, Blvd. 3, r. 1211, Volodarski Str. 6, 220050 MINSK, Tel. +375 172 20 0733, Fax. +375 172 20 0773

Belgium: see The Netherlands

Brazil: see South America

Bulgaria: Philips Bulgaria Ltd., Energoproject, 15th floor, 51 James Bourchier Blvd., 1407 SOFIA, Tel. +359 2 68 9211, Fax. +359 2 68 9102

Canada: PHILIPS SEMICONDUCTORS/COMPONENTS, Tel. +1 800 234 7381, Fax. +1 800 943 0087

China/Hong Kong: 501 Hong Kong Industrial Technology Centre, 72 Tat Chee Avenue, Kowloon Tong, HONG KONG, Tel. +852 2319 7988, Fax. +852 2319 7700

Colombia: see South America

Czech Republic: see Austria

Denmark: Sydhavnsgade 23, 1780 COPENHAGEN V, Tel. +45 33 29 3333, Fax. +45 33 29 3905

Finland: Sinkkulantie 3, FIN-02630 ESPOO, Tel. +358 9 615 800, Fax. +358 9 6158 0920

France: 51 Rue Carnot, BP317, 92156 SURESNES Cedex, Tel. +33 1 4099 6161, Fax. +33 1 4099 6427

Germany: Hammerbrookstraße 69, D-20097 HAMBURG, Tel. +49 40 2353 60, Fax. +49 40 2353 6300

Hungary: see Austria

India: Philips INDIA Ltd., Band Box Building, 2nd floor, 254-D, Dr. Annie Besant Road, Worli, MUMBAI 400 025, Tel. +91 22 493 8541, Fax. +91 22 493 0966

Indonesia: PT Philips Development Corporation, Semiconductors Division, Gedung Philips, Jl. Buncit Raya Kav 99-100, JAKARTA 12810, Tel. +62 21 794 0040 ext. 2501, Fax. +62 21 794 0080

Ireland: Newstead, Clonskeagh, DUBLIN 14, Tel. +353 1 7640 000, Fax. +353 1 7640 200

Israel: RA'ACAH Electronics, 7 Kehilat Saloni St, PO Box 18053, TEL AVIV 61180, Tel. +972 3 645 0444, Fax. +972 3 649 1007

Italy: PHILIPS SEMICONDUCTORS, Piazza IV Novembre 3, 20124 MILANO, Tel. +39 2 6752 2531, Fax. +39 2 6752 2557

Japan: Philips Bld 13-37, Kohoku 2-chome, Minato-ku, TOKYO 108-8507, Tel. +81 3 3740 5130, Fax. +81 3 3740 5077

Korea: Philips House, 260-199 Itaewon-dong, Yongsan-ku, SEOUL, Tel. +82 2 709 1412, Fax. +82 2 709 1415

Malaysia: No. 76 Jalan Universiti, 46200 PETALING JAYA, SELANGOR, Tel. +60 3 750 5214, Fax. +60 3 757 4880

Mexico: 5900 Gateway East, Suite 200, EL PASO, TEXAS 79905, Tel. +9-5 800 234 7381, Fax +9-5 800 943 0087

Middle East: see Italy

Netherlands: Postbus 90050, 5600 PB EINDHOVEN, Bldg. VB, Tel. +31 40 27 92785, Fax. +31 40 27 98399

New Zealand: 2 Wagener Place, C.P.O. Box 1041, AUCKLAND, Tel. +64 9 849 4160, Fax. +64 9 849 7811

Norway: Box 1, Manglerud 0612, OSLO, Tel. +47 22 74 8000, Fax. +47 22 74 8341

Pakistan: see Singapore

Philippines: Philips Semiconductors Philippines Inc., 106 Valero St, Salcedo Village, P.O. Box 2108 MCC, MAKATI, Metro MANILA, Tel. +63 2 816 6380, Fax. +63 2 817 3474

Poland: Ul. Lukiska 10, PL 04-123 WARSZAWA, Tel. +48 22 612 2831, Fax. +48 22 612 2327

Portugal: see Spain

Romania: see Italy

Russia: Philips Russia, Ul. Usatcheva 35A, 119048 MOSCOW, Tel. +7 095 755 6919, Fax. +7 095 755 6919

Singapore: Lorong 1, Toa Payoh, SINGAPORE 319762, Tel. +65 350 2538, Fax. +65 251 6500

Slovakia: see Austria

Slovenia: see Italy

South Africa: S.A. PHILIPS Pty Ltd., 195-215 Main Road Martindale, 2092 JOHANNESBURG, P.O. Box 7430 Johannesburg 2000, Tel. +27 11 470 5911, Fax. +27 11 470 5494

South America: Al. Vicente Pinzon, 173, 8th floor, 04547-130 SAO PAULO, SP, Brazil

Tel. +55 11 821 2333, Fax. +55 11 821 2382

Spain: Balmes 22, 08007 BARCELONA, Tel. +34 93 301 6312, Fax. +34 93 301 4107

Sweden: Kottbygatan 7, Akalla, S-16485 STOCKHOLM, Tel. +46 8 5985 2000, Fax. +46 8 5985 2745

Switzerland: Allmendstrasse 140, CH-8027 ZÜRICH, Tel. +41 1 498 2741 Fax. +41 1 498 3263

Taiwan: Philips Semiconductors, 6F, No. 96, Chien Kuo N. Rd., Sec. 1, TAIPEI, Taiwan Tel. +886 2 2134 2866, Fax. +886 2 2134 2874

Thailand: PHILIPS ELECTRONICS (THAILAND) Ltd., 209/2 Sanpavut-Bangna Road Prakanong, BANGKOK 10260, Tel. +66 2 745 4090, Fax. +66 2 399 0793

Turkey: Talatpasa Cad. No. 5, 80640 GÜLTEPE/İSTANBUL, Tel. +90 212 279 2770, Fax. +90 212 282 6707

Ukraine: PHILIPS UKRAINE, 4 Patrice Lumumba str, Building B, Floor 7, 252042 KIEV, Tel. +380 44 264 2776, Fax. +380 44 268 0461

United Kingdom: Philips Semiconductors Ltd., 276 Bath Road, Hayes, MIDDLESEX UB3 5BX, Tel. +44 181 730 5000, Fax. +44 181 754 8421

United States: 811 East Arques Avenue, SUNNYVALE, CA 94088-3409, Tel. +1 800 234 7381, Fax. +1 800 943 0087

Uruguay: see South America

Vietnam: see Singapore

Yugoslavia: PHILIPS, Trg N. Pasica 5/v, 11000 BEOGRAD, Tel. +381 11 62 5344, Fax. +381 11 63 5777

Internet: <http://www.semiconductors.philips.com>

For all other countries apply to: Philips Semiconductors International Marketing & Sales Communications, Building BE-p, P.O. Box 218, 5600 MD EINDHOVEN, The Netherlands, Fax. +31 40 27 24825

Printed in The Netherlands

465008/750/01/PP32

Date of release: 1999 Apr 12

Document order number: 9397 750 05024

SCA63

All rights are reserved. Reproduction in whole or in part is prohibited without the prior written consent of the copyright owner.
The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license under patent- or other industrial or intellectual property rights.

Let's make things better.

Philips
Semiconductors



PHILIPS