

Universidad de Costa Rica
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica
IE0624: Laboratorio de Microcontroladores
I ciclo 2022

Laboratorio 3
PID,GPIO,ADC, comunicaciones USART y SPI

Matías Leandro Flores
B94199

Profesor: Marco Villalta Fallas

25 de mayo de 2022

Resumen

- El repositorio del presente laboratorio se puede acceder en el siguiente enlace: <https://github.com/matflow/Labo-3-Microcontroladores.git>.

1. Nota teórica

1.1. Microcontrolador ATmega328P

Para el presente laboratorio se utilizó la placa de desarrollo Arduino Uno, la cual contiene el ATmega328P. El ATmega328P es un microcontrolador de arquitectura RISC de 8 bits que contiene una memoria flash programable de 32kb, 2kB de SRAM y 1kb de EEPROM. Entre sus periféricos, contiene dos timers de 8 bits y un timer de 16 bits, además de seis canales PWM, un canal de comunicación USART programable y una interfaz serial SPI [1]. Su diagrama de pines se presenta en la siguiente figura.

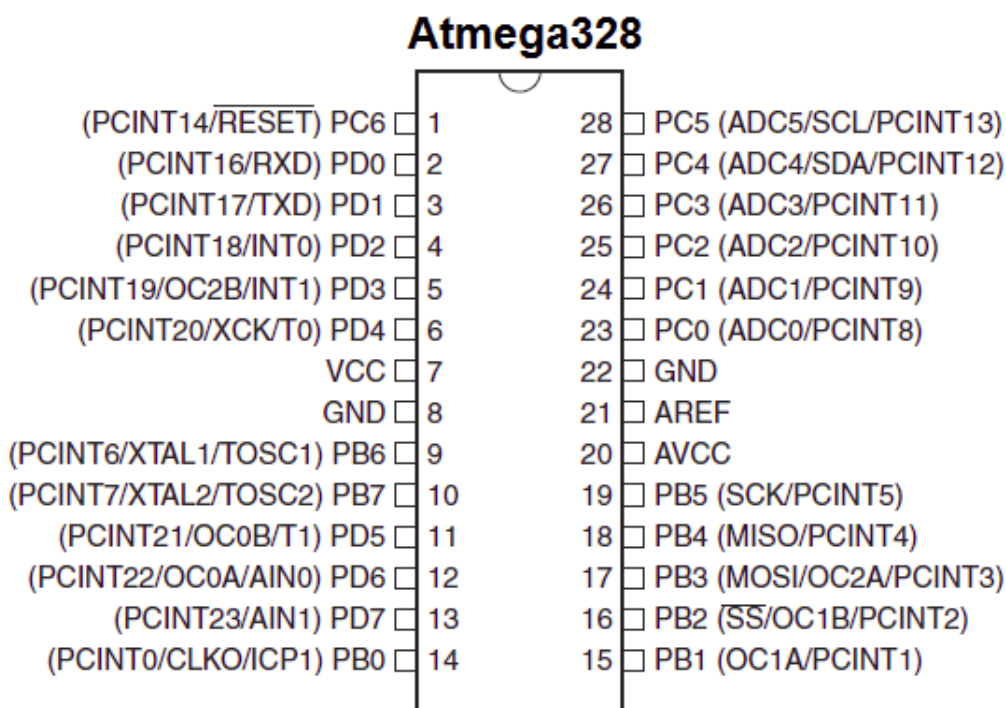


Figura 1: Diagrama de pines del ATmega328P [1]

El diagrama de bloques del núcleo del microcontrolador se muestra a continuación.

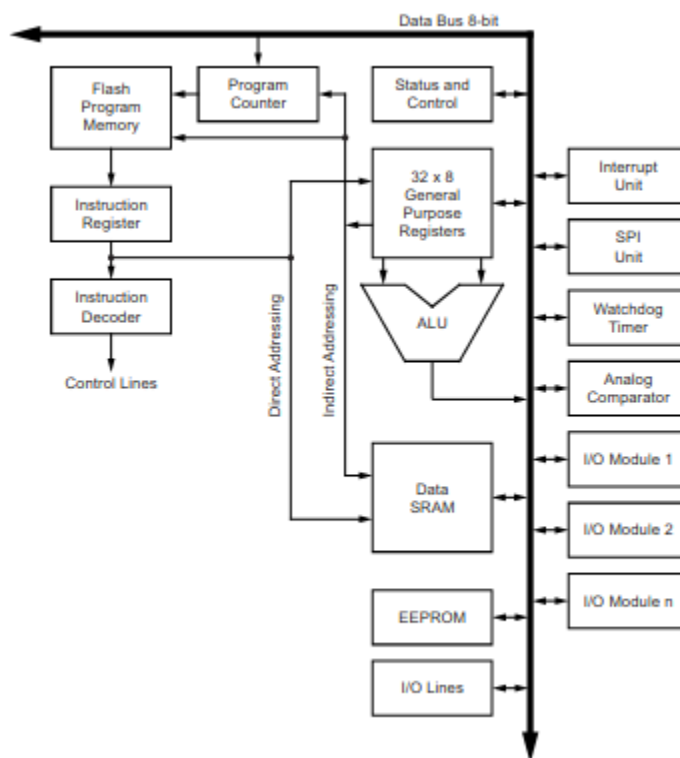


Figura 2: Diagrama de bloques de ATmega328P [1].

Entre las características eléctricas importantes de este microcontrolador destaca una corriente máxima por pin I/O de 40 mA, un voltaje de alimentación de 3.3 V o 5 V, y un voltaje de salida lógico alto de 4.1 V ($V_{cc} = 5$) o 2.3 V ($V_{cc} = 3.3$).

1.2. Componentes electrónicos adicionales

Como componentes electrónicos adicionales al microcontrolador se utilizó un **diodo LED rojo** y **diodo LED azul**. Se asumirá que se utilizó el LED rojo estándar de *Multicomp* [2] para estos dos LEDs. También se utilizó el **termistor NTC de 10kΩ NTCALUG03** de Vishay [3]. Este termistor utilizado tiene un valor de resistencia de 10kΩ a una temperatura de 25°C y un coeficiente β de 3740 a 3984. Para efectos de simulación se utilizó un termistor con el mismo valor de resistencia a temperatura ambiente y con un β de 3455, que es similar al del termistor escogido. Además, se escogió el **sensor de humedad DHT22**, el cual tiene un rango de operación de 0 - 100 % HR (humedad relativa), con una alimentación de 3.3 V a 6 V DC [4]. Para efectos de simulación se utilizó una fuente variable de voltaje de 5 V (dentro de su rango de alimentación), cuyos valores se mapearon a valores de 0 a 100 (representando el porcentaje de humedad relativa medido). Otro componente utilizado fue la pantalla **PCD8544**, la cual es una pantalla de tipo LCD que se comunica con el Arduino mediante el interfaz serial SPI [5].

Otro componente electrónico utilizado fue el **transistor BJT NTE123A**, el cual es capaz de generar una corriente de colector de hasta $I_C = 800$ mA [6]. Además de estos componentes mencionados se utilizaron seis resistores de distintos valores y un resistor de potencia que puede consumir hasta 5 W y un potenciómetro de hasta 10 kΩ.

1.3. Diseño del circuito

Para el diseño de los resistores de $120\ \Omega$ en serie con cada LED de alerta partió de desear una corriente en cada LED de entre 15mA a 20mA , lo cual aseguraría una buena iluminación. Para asegurar que no se sobrepase este valor de corriente se asumirá el nivel de voltaje en directo máximo del diodo rojo de Multicomp [2] de 2.5 V . De esta forma, realizando la ley de tensiones de Kirchhoff en el lazo de un LED y el resistor a diseñarse se tiene lo siguiente.

$$5V = V_R + 2,5\text{ V} \longrightarrow V_R = 2,5\text{ V} \quad (1)$$

O sea que, cuando el diodo está polarizado en directa la tensión del resistor a diseñarse será de 2.5 V y para asegurar que no sobrepase los 20 mA :

$$20\text{ mA} = \frac{2,5}{R} \longrightarrow R = 125\Omega \quad (2)$$

Este valor se aproxima a $R = 120\Omega$ para los seis resistores de los LEDs de semáforos.

Para ajustar el punto de operación, o *set point* por medio del potenciómetro con un valor máximo de $10\text{ k}\Omega$ se colocó en serie con un resistor, con el potenciómetro conectado a tierra y una alimentación de 5 V , de tal forma que el potenciómetro debe dar 5 V en su valor mayor de resistencia y 0 V cuando su valor es 0 . El valor del resistor en serie se determinó por medio del siguiente divisor de tensión.

$$\frac{5V \cdot 10k}{10k + R} = 5V \quad (3)$$

$$10k(5V - 5V) = 5V \cdot R \longleftrightarrow R \longrightarrow 0 \quad (4)$$

Por lo tanto el resistor en serie con el potenciómetro debe ser muy pequeño en comparación a $10\text{ k}\Omega$ y se eligió el valor comercial de $91\ \Omega$.

Para obtener la lectura de temperatura del termistor se colocó en serie con un resistor de $10\text{ k}\Omega$ y se determinó el valor de la resistencia del termistor despejando su resistencia de la ecuación para un divisor de tensión de dos resistores en serie, con lo que se obtuvo lo siguiente.

$$R_T = \frac{V_T \cdot 10k}{5V - V_T} \quad (5)$$

Donde R_T es la resistencia del termistor y V_T es el voltaje del termistor. Luego, para calcular el valor de temperatura en base a R_T y el *beta* (3455) se utilizó la ecuación del parámetro *beta* simplificado [?]:

$$\frac{1}{T} = \frac{1}{T_0} + \frac{1}{\beta} \ln\left(\frac{R_T}{R_o}\right) \quad (6)$$

Donde T_0 es la temperatura ambiente en Kelvin y R_o es el valor de resistencia del termistor a temperatura ambiente ($10\text{ k}\Omega$).

Para diseñar un circuito de calentador para prevenir que se demande mucha corriente del Arduino, se utilizó una alimentación externa de 12 V para el resistor de calentador, cuyo voltaje efectivo fue controlado por el transistor BJT NTE123A que funciona como un interruptor, variando entre las regiones de corte y saturación del transistor por medio de la señal de control PWM del Arduino. De esta forma fue posible modular la corriente efectiva que fluye sobre el resistor, y así variar el voltaje efectivo del resistor entre 0 V a 12 V , dependiendo de la señal PWM. Para esto fue necesario diseñar la resistencia de base del transistor, para no quemarlo y determinar la mayor corriente que puede dar

en el colector. Siendo la resistencia del calentador $R_c = 120\Omega$, por la Ley de Ohm, para un voltaje de 12 V:

$$12V/120\Omega = 100 \text{ mA} \quad (7)$$

Una corriente de $I_c = 100 \text{ mA}$ supera el valor máximo que se puede demandar del ATmega328P. Esta corriente de colector, debido a la ganancia de 100 del NTE123A significa una corriente de base de 1 mA. Por lo tanto se colocó el transistor BJT con una resistencia en la base tal que

$$R_B = \frac{V_{PWM} - V_{BE}}{I_B} = \frac{5V - 0,7V}{1mA} = 4,3 \text{ k}\Omega \quad (8)$$

La ecuación (8) se realizó asumiendo un voltaje de base-emisor de 0.7 V (el voltaje de umbral de una unión NP de un diodo de silicio) e indica que el resistor en la base del BJT debe ser de 4.3 k Ω . De esta forma, la corriente de base, que es la corriente que sale del pin de salida del Arduino es de tan solo 1mA, y gracias a la ganancia del transistor permite que la corriente a través del resistor de calentamiento sea de hasta 100 mA.

Cabe mencionar que también se agregó un switch que desactiva la comunicación serial si se cierra. El esquemático en SimulIDE del circuito diseñado se presenta en la Fig. 3.

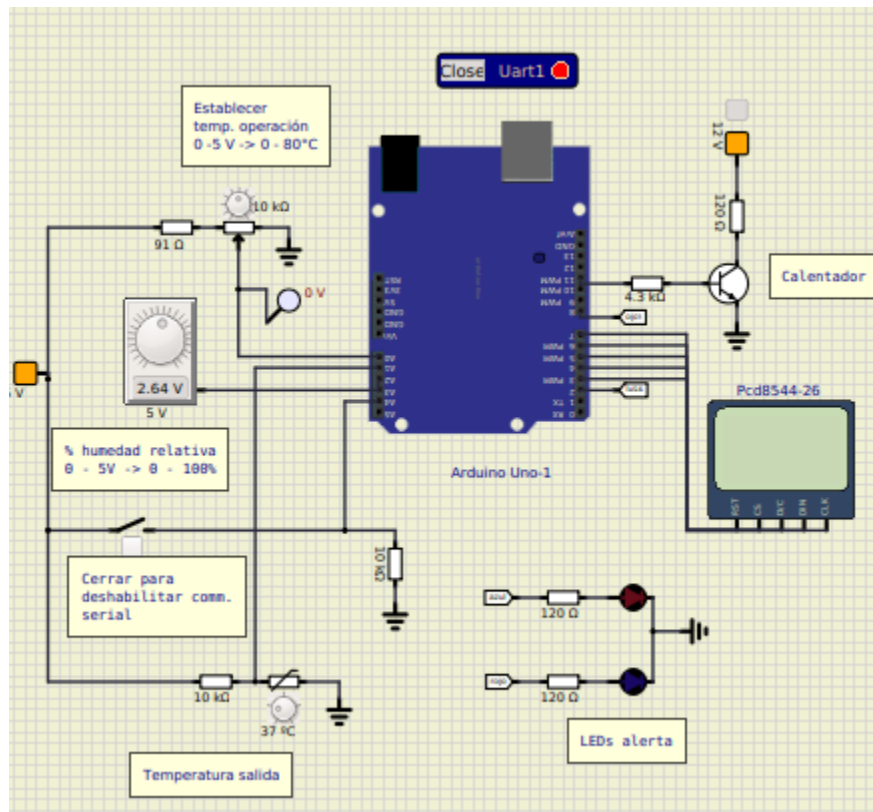


Figura 3: Esquemático en SimulIDE de circuito de semáforo diseñado.

En cuanto a los precios, de acuerdo con el portal Sparkfun, se tiene un precio de \$0.45 para un LED básico rojo de 5mm, se utilizará este precio para el LED azul también [7]. En cuanto a los resistores, un precio común para resistores de 1 a 4 W individuales es de \$0.09 y se tienen seis resistores normales. Mientras tanto, el precio de un resistor de 5 W de \$1.71. El precio actual en el

portal de Amazon para tres sensores de humedad DHT22 es de \$14.99 por lo que se puede estimar el precio de un solo sensor en \$5 dólares. El precio del termistor de Vishay utilizado [3] es de \$6 por 5 unidades, por lo que se estima un precio de \$0.83 dólares por uno solo. El precio actual de un potenciómetro de 10 k Ω en el portal de Amazon es de \$5.99. La tienda MicroJPM de Cartago vende un módulo PCD8544 Nokia 5510 en \$8.95. El precio del transistor NTE123A individual es de \$0.96. Por último, un Arduino Uno original tiene un precio de stock actual en su portal oficial de \$25.

Así, el precio total de la incubadora diseñada es de **\$49.89**.

2. Desarrollo/Análisis de resultados

2.1. Control PID

Para el control PID se utilizó la librería `¡PID_v1.h¿`, y se declaró un objeto PID cuya entrada es la misma temperatura sensada (debido al lazo realimentado), el set point se dio mediante el voltaje controlado por el potenciómetro mapeado a valores de 0 a 80 por medio de la función *map*, su salida de control es una señal PWM de 0 a 255 y como parámetros se eligió una constante proporcional al error de $Kp = 3$, una constante integral de $Ki = 5$ para reducir el error final sin aumentar el sobrepaso inicial y una constante derivativa de $Kd = 1$. Así, dentro de la función de `setup()` se inicializó el PID con los valores definidos de la siguiente forma.

```
PID incubadora(&temperatura, &salida, &setPoint, Kp, Ki, Kd, DIRECT);
```

Mientras tanto, para el display PCD8544 se mapearon sus pines de RESET, CS, D/S, Din y SCLK a los pines del 3 al 7, respectivamente y se creó un objeto de tipo PCD llamado *display* por medio de la API de la librería `¡Adafruit_PCD8544.h¿` la cual también debe utilizar la librería `¡SPI.h¿`. Para escribir al display se utilizaba el método `clear()` al inicio de cada ciclo y se fija el cursor en la posición (0,0) con el método `setCursor()` para luego escribir los datos indicados.

Del lado de Python se realizó un script que se comunica de forma serial con el puerto creado por medio del comando *socat* en Linux para poder comunicarse con el puerto serial de Arduino. Al inicio de cada ciclo del Arduino se envió imprimió en el puerto serial un string “loop” que se revisa dentro del script para así empezar a almacenar los cuatro valores (set point, humedad, temperatura y señal de control) en una lista en orden. Luego se abrió un archivo en modo escritura y se escribió esta lista a dicho archivo de forma cíclica, tal que cada elemento de la lista está separado por comas, como se muestra a continuación.

```
f = open('csvpy.csv', 'w', encoding='UTF8')
writer = csv.writer(f)
writer.writerow(header) # escribe encabezado
while True:
    # write a row to the csv file
    i = 0
    search = arduino.readline().decode('utf-8').replace('\r\n', '')
    if search == 'loop': # revisa si inició ciclo de datos
        for i in range(4):
            data[i] = arduino.readline().decode('utf-8')
            data[i] = data[i].replace('\r\n', '')
        writer.writerow(data)
```

La funcionalidad del programa se evidencia en la siguiente captura del archivo .csv generado por medio de Python con los datos capturados por el sistema del microcontrolador.

```

1 Set Point,Humedad,Temp sensada,Salida
2 39.00,51,39.00,0.00
3 39.00,51,39.00,0.00
4 39.00,51,39.00,0.00
5 39.00,51,39.00,0.00
6 39.00,51,39.00,0.00
7 39.00,51,39.00,0.00
8 39.00,51,39.00,0.00
9 39.00,51,39.00,0.00
10 39.00,51,39.00,0.00
11 39.00,51,39.00,0.00
12 39.00,65,39.00,0.00
13 39.00,65,39.00,0.00
14 39.00,65,39.00,0.00
15 39.00,65,39.00,0.00
16 39.00,65,39.00,0.00
17 39.00,52,39.00,0.00
18 39.00,52,39.00,0.00
19 39.00,52,39.00,0.00
20 39.00,52,39.00,0.00
21 39.00,52,78.09,0.00
22 39.00,52,64.97,79.31
23 39.00,52,64.97,0.00
24 39.00,52,64.97,0.00
25 39.00,52,51.02,115.42
26 39.00,52,48.03,11.82
27 39.00,52,40.07,77.43
28 39.00,52,40.07,0.00
29 39.00,52,28.98,135.93
30 39.00,52,28.98,30.05

```

Figura 4: Archivo de CSV generado en una corrida del script con el sistema simulado encendido.

De la Fig. 4 se aprecia que se dejó el set point fijo en 39°C, con una humedad de 51 %. La temperatura sensada es la misma que la del set point, por lo que la señal de salida de control es 0.00. Pero al variar la temperatura sensada de 39 °C a 78°C se observa que la salida de control comienza a cambiar de valores debido al PID controlando la salida para tratar de acercarla al set point.

2.2. Funcionalidad electrónica

La funcionalidad electrónica de los LEDs de alerta se aprecia en las figuras 5 y 6. En la Fig. 5 se observa que a una temperatura mayor de 42°C se activa el LED rojo mientras que en la Fig. 6 se muestra que se activa el LED azul a una temperatura menor a 30°C.

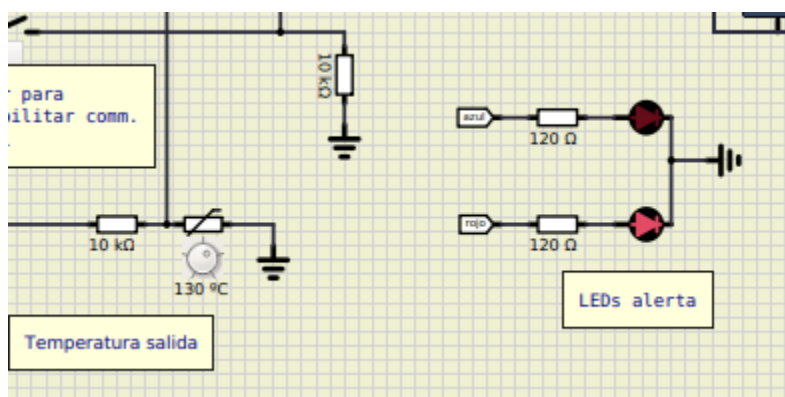


Figura 5: Activación de LED de alerta rojo.

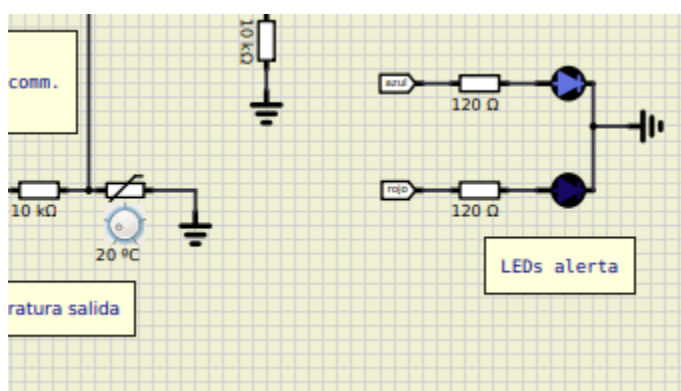


Figura 6: Activación de LED de alerta azul.

Además, la correcta implementación del LCD PCD8544 se muestra en la Fig. 7.

En la Fig. 7 se aprecia que se muestra el set point en 39°C, mientras que la temperatura sensada es de 25°C, lo cual es corroborado por el encendido del LED azul, además, se observa el valor de la humedad de 52% y el valor variable de la señal de control.

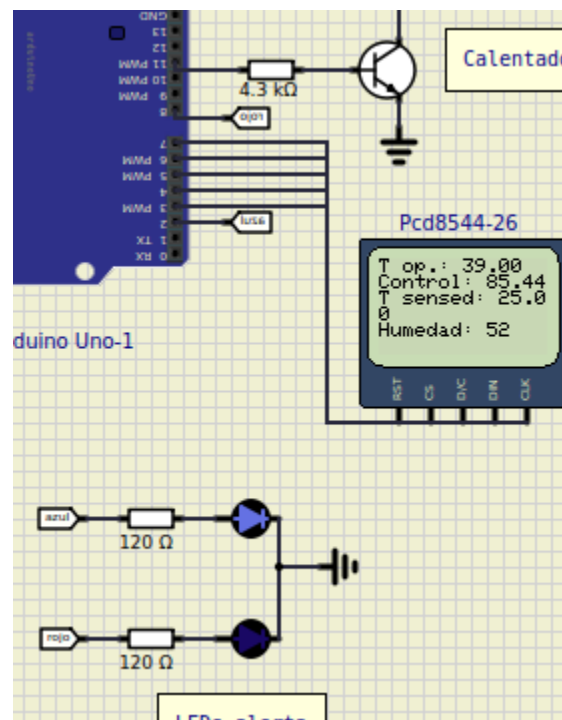


Figura 7: Captura de pantalla de PCD8544.

3. Conclusiones y recomendaciones

Con la elaboración del presente laboratorio se concluyó que la utilización de microcontroladores para poder generar datos a partir de sensores de la vida real es posible gracias a los protocolos de comunicación de estos. Además, es posible integrar distintos módulos que la dan mayor funcionalidad al microcontrolador para expandir el potencial del mismo.

Se tienen las siguientes recomendaciones luego de la realización del laboratorio.

- Investigar en detalle como usar los módulos externos al microcontrolador, e indagar sobre el uso correcto de su librería en caso de que tenga una.
- Conocer las características eléctricas del microcontrolador y saber utilizar fuentes de alimentación externas para que los pines de salida funcionen tan solo para entregar un valor lógico.
- Estudiar los pines de los módulos a utilizar, para poder conectarlos de la forma correcta y prevenir fallos del mismo

Referencias

- [1] Atmel, *ATmega328P*. Atmel Corporation, 1.0 ed., 2015. https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf.
- [2] Multicomp, *Standard Red-emitting LED*. Farnell, 1.1 ed., 2011.
- [3] Vishay, *NTCALUG03 Mini Lug Series*. Vishay Corporation, 1.0 ed., s.f. <https://www.datasheetarchive.com/pdf/download.php?id=88361afd824cda82d0015fb93c31884c30fc76&type=P&term=ntc%252010k%2520temperature%2520sensor>[online].
- [4] T. Liu, *DHT22 (DHT22 also named as AM2302)*. Aosong Electronics Co.,Ltd, 1.0 ed., s.f. <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>[online].
- [5] Phillips, *PCD8544 48 × 84 pixels matrix LCD controller/driver*. Phillips Semiconductors, 1999. <https://www.sparkfun.com/datasheets/LCD/Monochrome/Nokia5110.pdf>.
- [6] NTE Electronics, *NTE123A (nnp) NTE159M (pnp)*, 2003.
- [7] Sparkfun, *LED Basic Red 5mm*. Sparkfun Electronics, s.f. <https://www.sparkfun.com/products/9590>.

Apéndices

8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash

DATASHEET

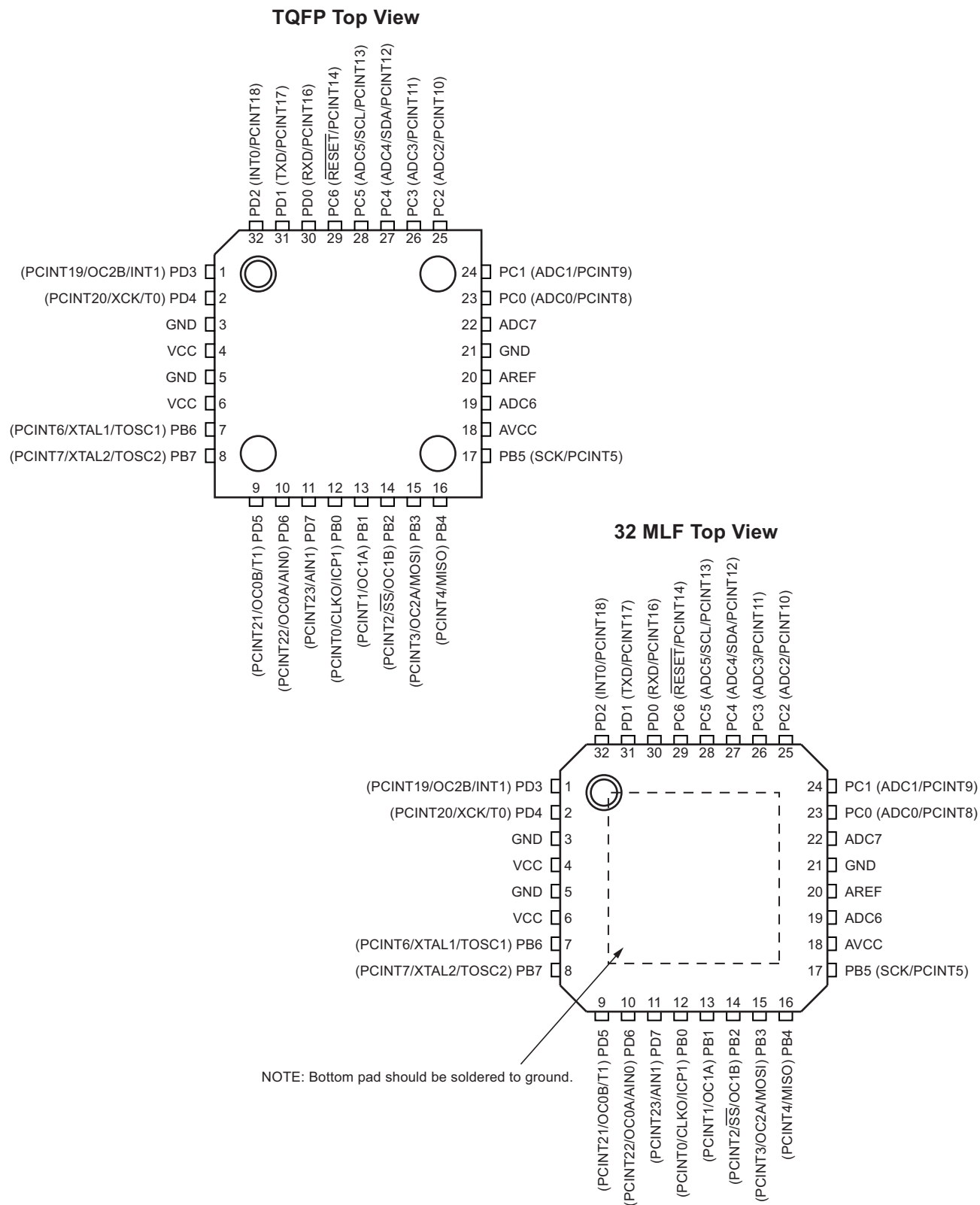
Features

- High performance, low power AVR® 8-bit microcontroller
- Advanced RISC architecture
 - 131 powerful instructions – most single clock cycle execution
 - 32 × 8 general purpose working registers
 - Fully static operation
 - Up to 16MIPS throughput at 16MHz
 - On-chip 2-cycle multiplier
- High endurance non-volatile memory segments
 - 32K bytes of in-system self-programmable flash program memory
 - 1Kbytes EEPROM
 - 2Kbytes internal SRAM
 - Write/erase cycles: 10,000 flash/100,000 EEPROM
 - 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
 - Programmable serial USART
 - Master/slave SPI serial interface
 - Byte-oriented 2-wire serial interface (Phillips 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
 - 32-lead TQFP, and 32-pad QFN/MLF
- Operating voltage:
 - 2.7V to 5.5V for ATmega328P
- Temperature range:
 - Automotive temperature range: -40°C to $+125^{\circ}\text{C}$
- Speed grade:
 - 0 to 8MHz at 2.7 to 5.5V (automotive temperature range: -40°C to $+125^{\circ}\text{C}$)
 - 0 to 16MHz at 4.5 to 5.5V (automotive temperature range: -40°C to $+125^{\circ}\text{C}$)
- Low power consumption
 - Active mode: 1.5mA at 3V - 4MHz
 - Power-down mode: 1 μA at 3V

1. Pin Configurations

Figure 1-1. Pinout

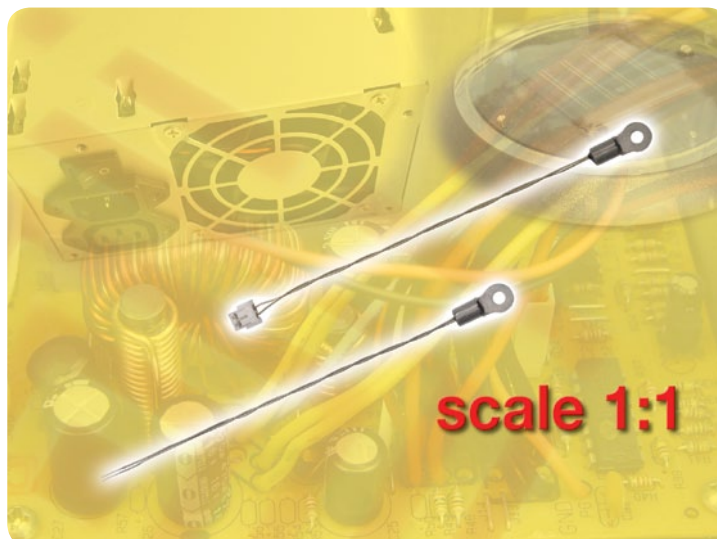


NTC THERMISTORS

NTCALUG03 Mini Lug Series



NTC Thermistors, Mini Lug Series



KEY BENEFITS

- Smallest mounting space of 65 mm² (0.1 in²) only with tiny 5 mm ring tongue
- Short response time, < 4 s
- Lower thermal gradient, < 5 % of ΔT
- Space savings for limited surface areas

APPLICATIONS

- Heatsink thermal management
- Power electronics thermal management
- Housing surface temperature measurement
- Pipe and conductor temperature measurement

RESOURCES

- Datasheet: NTCALUG03 Mini Lug Series - <http://www.vishay.com/doc?29114>
- For technical questions contact nlr@vishay.com



NTC THERMISTORS

NTCALUG03 Mini Lug Series



NTC Thermistors, Mini Lug Series



QUICK REFERENCE DATA		
PARAMETER	VALUE	UNIT
Resistance value at 25 °C	10K to 47K	Ω
Tolerance on R_{25} -value	± 2 to ± 3	%
$B_{25/85}$ -value	3740 to 3984	K
Tolerance on $B_{25/85}$ -value	± 0.5 to ± 1.5	%
Operating temperature range: At zero dissipation	- 40 to + 125	°C
Response time	3.5	s
Thermal time constant τ	≈ 5	s
Dissipation factor δ	10	mW/K
Maximum power dissipation at 25 °C	100	mW
Min. dielectric withstanding voltage between terminals and lug	1000	V _{AC}
Climatic category (LCT/UCT/days)	40/125/56	-
Weight		
without connector	0.5	g
with connector	0.6	g

Note

- Other R_{25} values and tolerances available upon request

FEATURES

- Fast time response for surface applications compared to industry standard NTC lug sensors
- Reduced thermal gradient, due to the use of small dimensions and nickel conductor, allowing for an accurate surface temperature measurement
- The sensor is not suitable for being permanently in contact with water or liquids
- Small size connector and small lug ring tongue terminal, allowing for temperature sensing at locations where only limited space is available
- Connector ZHR-2 (optional)
- Compliant to RoHS Directive 2011/65/EU



RoHS
COMPLIANT

APPLICATIONS

Thermistors used for surface temperature sensing and control in:

- Computer equipment
- MOSFETS, IC's, Power Electronics, heatsink temperature control, LED emitter heat-sink control
- Consumer appliances
- Industrial equipment
- Automotive equipment

DESCRIPTION

Miniature insulated chip thermistor with a negative temperature coefficient in accordance with IEC 60539. The device has no marking.

MOUNTING

- The sensor can be mounted by means of a screw. For stud size, metric 2 mm M2/American stud #1 or #2
- The end wire can be soldered, welded or crimped to a connector
- Optional connector for Wire-to-Wire or Wire-to-Board connections

ELECTRICAL DATA AND ORDERING INFORMATION					
R_{25} -VALUE (kΩ)	R_{25} -TOL. (%)	$B_{25/85}$ -VALUE (K)	$B_{25/85}$ -TOL. (± %)	SAP MATERIAL NUMBER	DESCRIPTION
10	± 3	3984	0.5	NTCALUG03A103H	NTC Mini Lug 10K 3 % 3984 K 0.5 %
10	± 3	3984	0.5	NTCALUG03A103HC	NTC Mini Lug 10K 3 % 3984 K 0.5 % with connector
10	± 2	3984	0.5	NTCALUG03A103G	NTC Mini Lug 10K 2 % 3984 K 0.5 %
10	± 2	3984	0.5	NTCALUG03A103GC	NTC Mini Lug 10K 2 % 3984 K 0.5 % with connector
12	± 3	3740	1.5	NTCALUG03A123H	NTC Mini Lug 12K 3 %
12	± 3	3740	1.5	NTCALUG03A123HC	NTC Mini Lug 12K 3 % with connector
47	± 3	3740	1.5	NTCALUG03A473H	NTC Mini Lug 47K 3 %
47	± 3	3740	1.5	NTCALUG03A473HC	NTC Mini Lug 47 kΩ 3 % with connector

Note

- Ordering information can be found on: www.vishay.com/doc?33036

DATA SHEET

PCD8544

**48 × 84 pixels matrix LCD
controller/driver**

Product specification
File under Integrated Circuits, IC17

1999 Apr 12

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

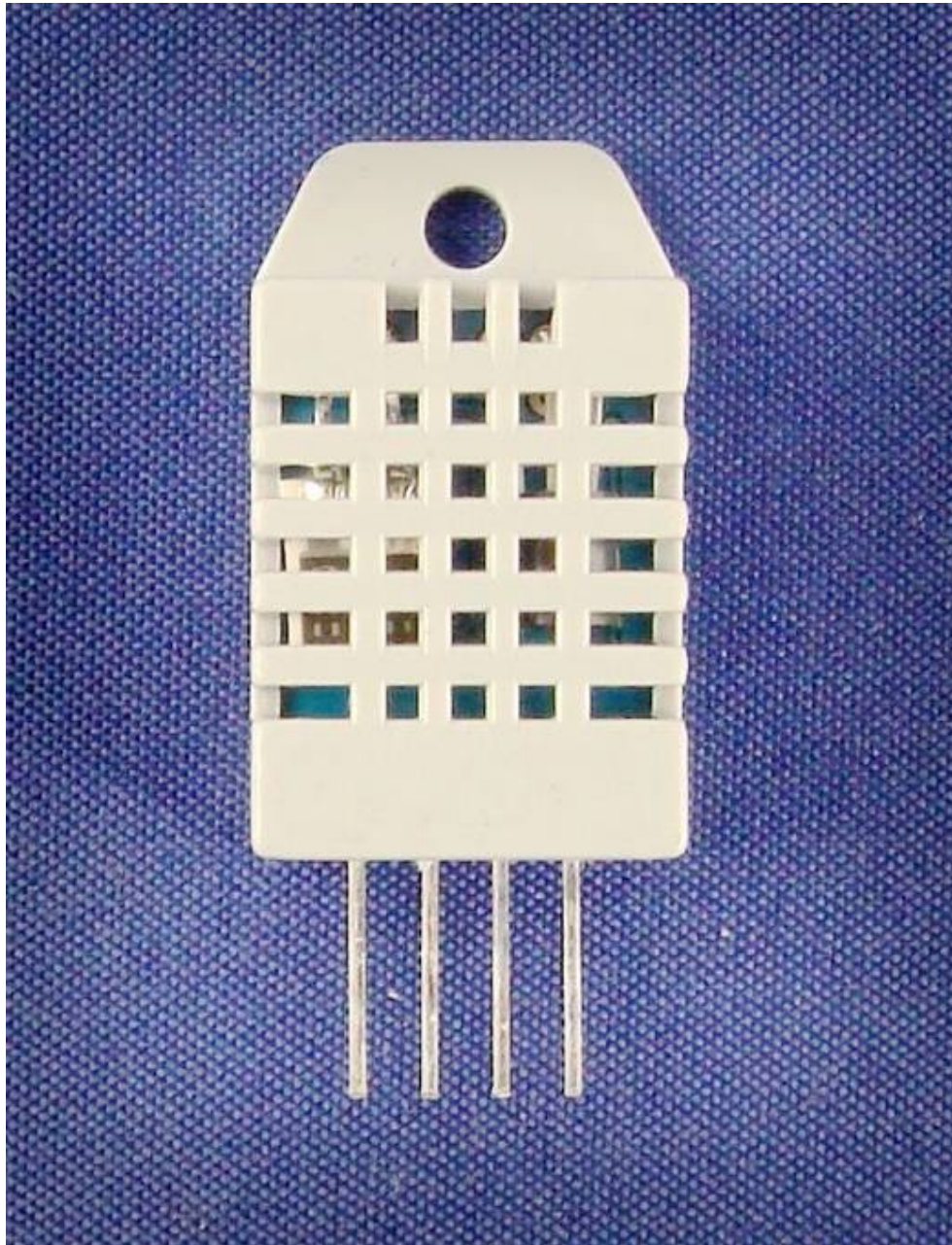
- Telecommunications equipment.

4 ORDERING INFORMATION

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

Digital-output relative humidity & temperature sensor/module

DHT22 (DHT22 also named as AM2302)



Capacitive-type humidity and temperature module/sensor

Aosong Electronics Co.,Ltd

Your specialist in innovating humidity & temperature sensors

1. Feature & Application:

- * Full range temperature compensated
- * Relative humidity and temperature measurement
- * Calibrated digital signal
- * Outstanding long-term stability
- * Extra components not needed
- * Long transmission distance
- * Low power consumption
- * 4 pins packaged and fully interchangeable

2. Description:

DHT22 output calibrated digital signal. It utilizes exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability. Its sensing elements are connected with 8-bit single-chip computer.

Every sensor of this model is temperature compensated and calibrated in accurate calibration chamber and the calibration-coefficient is saved in type of programme in OTP memory, when the sensor is detecting, it will cite coefficient from memory.

Small size & low consumption & long transmission distance(20m) enable DHT22 to be suited in all kinds of harsh application occasions.

Single-row packaged with four pins, making the connection very convenient.

3. Technical Specification:

Model	DHT22
Power supply	3.3-6V DC
Output signal	digital signal via single-bus
Sensing element	Polymer capacitor
Operating range	humidity 0-100%RH; temperature -40~80Celsius
Accuracy	humidity +2%RH(Max +5%RH); temperature <+-0.5Celsius
Resolution or sensitivity	humidity 0.1%RH; temperature 0.1Celsius
Repeatability	humidity +-1%RH; temperature +-0.2Celsius
Humidity hysteresis	+0.3%RH
Long-term Stability	+0.5%RH/year
Sensing period	Average: 2s
Interchangeability	fully interchangeable
Dimensions	small size 14*18*5.5mm; big size 22*28*5mm

4. Dimensions: (unit----mm)

1) Small size dimensions: (unit----mm)