

UNIVERSIDAD DE COSTA RICA

ESCUELA DE INGENIERÍA ELÉCTRICA

IE-0624: LABORATORIO DE MICROCONTROLADORES

Laboratorio 3

Arduino: PID, GPIO, ADC y comunicaciones

Prof. Marco Villalta

Estudiantes:
Sofía Fonseca Muñoz, B42634

11 de octubre de 2022

Índice

1. Introducción/Resumen	2
2. Nota teórica	3
2.1. Microcontrolador	3
2.2. Componentes electrónicos	4
2.3. Bibliotecas	5
2.4. Diseño	5
3. Desarrollo/Análisis de resultados	9
3.1. Activación de LEDs	9
3.2. Pantalla LCD muestra la temperatura de operación, la señal de control, la temperatura sensada y humedad de operación.	10
3.3. Funcionamiento de la resistencia de poder	11
3.4. Asignación de temperatura deseada a través del potenciómetro	12
3.5. Envío de valores sensados a través de USART según la activación del switch.	13
3.6. Lectura de datos a través de script de Python.	14
4. Conclusiones y recomendaciones	16
Referencias	17
5. Apéndice	18

1. Introducción/Resumen

Este proyecto consiste en la creación de una incubadora de huevos automática basada en Arduino UNO. La incubadora cuenta con una pantalla LCD PCD8544 donde se muestra la temperatura deseada, la señal de salida del PID que mantiene la temperatura adecuada, la temperatura y humedad que tiene la incubadora. Estas últimas dos señales se miden a través de un sensor llamado DHT22 el cual incluye un módulo para medir la temperatura y otro para medir la humedad. Por último incluye algunos componentes extra, como LEDs que le permiten indicar cuando está muy frío o muy caliente y una resistencia con la cual se calienta el circuito.

Además la incubadora se conecta con la computadora a través del protocolo USART y se crea un archivo .csv que contiene la información recolectada.

A través de este laboratorio, se pudo aprender sobre señales PWM que permiten enviar señales de tensión variable desde una fuente digital, la relación de entradas y salidas, la implementación de controladores PID y comunicaciones a través de USART con la computadora.

El repositorio correspondiente a este laboratorio se puede encontrar en el enlace <https://github.com/sofifon/IE0624-2022/tree/main/lab3>.

2. Nota teórica

2.1. Microcontrolador

Para este laboratorio se utilizó la placa Arduino UNO. Esta placa es muy popular debido a su bajo costo y su capacidad para resolver problemas en áreas educativas como industriales. Un ejemplo de ello es la capacidad de capturar información de sensores en tiempo real y enviar esta información a la computadora. [1]

La placa cuenta con un procesador ATmega328P, 14 pines de entrada y salida digital, 6 pines de entrada analógica y conexión USB. A nivel eléctrico, la placa cuenta con dos salidas fijas de 3V y 5V y con entrada de voltaje a través de un pin, el cual soporta de 6 a 20 V y a través del USB el cual por convención recibe 5.5V. Además el procesador puede correr a una frecuencia de hasta 20MHz.[1] En la siguiente imagen se puede observar el diagrama de bloques del procesador ATmega328P:

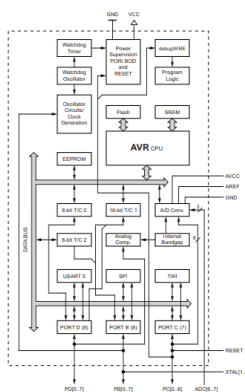


Figura 1: Diagrama de bloques del procesador ATmega328P. Tomada de https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf

Es importante destacar que este procesador se encuentra integrado en la placa Arduino UNO, por lo que no se realizan conexiones directas a los pines del microprocesador sino a los pines de la placa. La forma en que se integra el microprocesador en la placa, se puede observar en el siguiente diagrama:

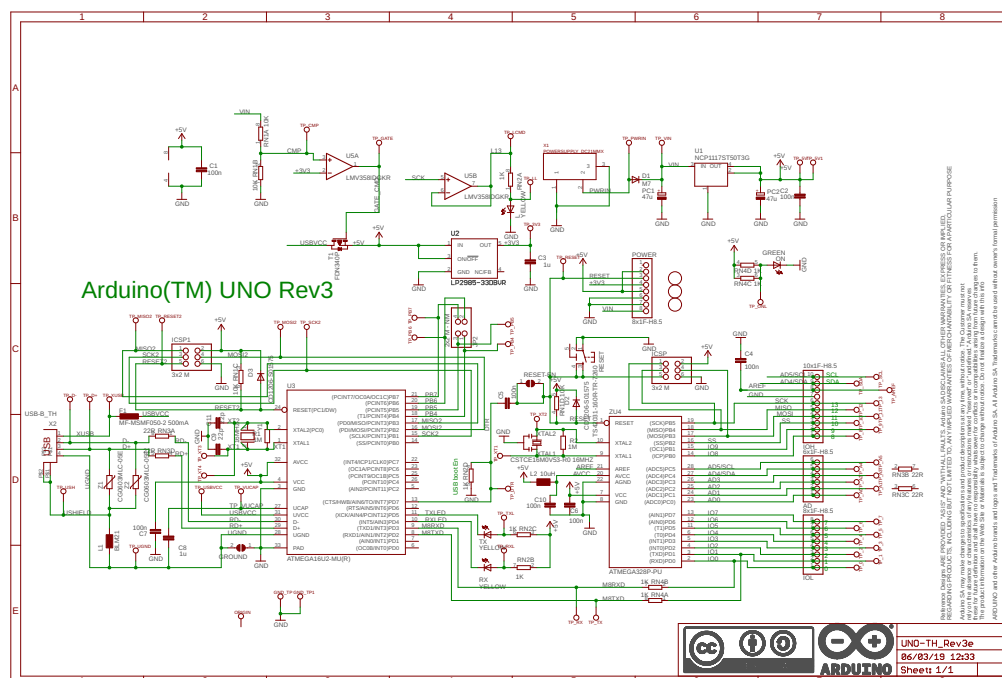


Figura 2: Diagrama de bloques de la placa Arduino UNO. Tomada de https://content.arduino.cc/assets/UNO-TH_Rev3e_sch.pdf

Por último, para simplificar el uso de la placa, la hoja del fabricante cuenta con un diagrama de pines de alto nivel que se utilizó como guía para realizar las conexiones en este laboratorio.

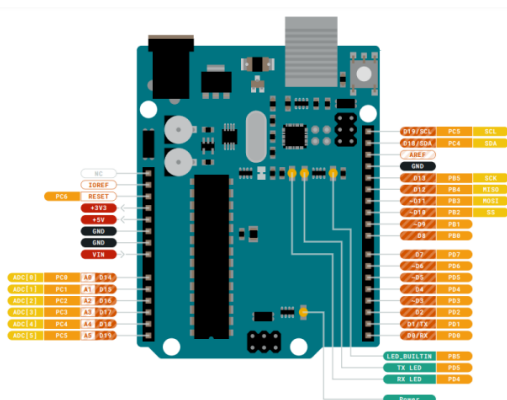


Figura 3: Conexiones de pines de la placa Arduino UNO. Tomada de <https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>

2.2. Componentes electrónicos

Para la creación de la incubadora, se utilizó la lista de componentes que se encuentra en la siguiente tabla. Los precios de cada componente fueron tomados de <https://www.crcibernetica.com/> tomando en cuenta un tipo de cambio de 650 colones equivalente a 1 USD. Además los precios

de las resistencias y los LEDs se calcularon a partir de paquetes grandes dividiendo su valor entre la cantidad, ya que por su valor no se consiguen individualmente.

Componente	Precio (colones)
PCD8544	1082
DHT22	5818
Switch	618
2 x Resistor 100 ohm	65
LED rojo	20
LED azul	20
Potenciometro 1k	618
fuelle 12v	9000
Resistor 29 ohm	33
Capacitor 10 uF	163
BJT	488
Precio total	17925

El precio total de todos los componentes necesarios para construir la incubadora sería 17925 colones, sin embargo se tendría que tomar en cuenta el precio de otras partes físicas como el contenedor que formaría la incubadora, y estuches para los componentes.

2.3. Bibliotecas

A nivel de software, se utilizó el lenguaje propio de Arduino UNO, donde se incluyeron cada una de las siguientes bibliotecas:

- **PCD8544.h**: para controlar la pantalla LCD.
- **DHT.h**: para controlar el sensor de humedad y temperatura DHT22.
- **ArduPID.h**: Para implementar un controlador PID que genere una temperatura adecuada.

2.4. Diseño

El diseño completo del circuito eléctrico que se utilizó se puede observar en la siguiente imagen:

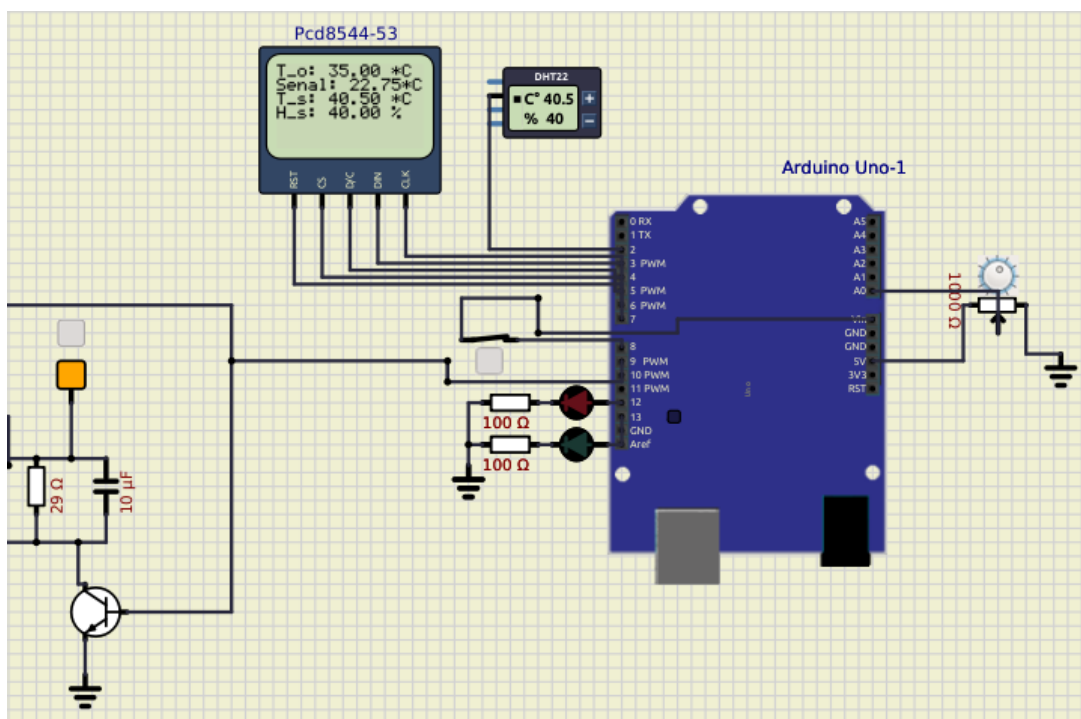


Figura 4: Diseño completo del circuito de la incubadora. Imagen propia.

A continuación se explicará cada una de las conexiones hechas y la justificación de su escogencia.

Para la lectura de la temperatura y humedad se escogió el sensor DHT22, ya que es un sensor que cumple con las dos medidas que se buscaban para la incubadora. Además el rango de funcionamiento deseado se encuentra dentro de las posibilidades que ofrece este sensor. En la hoja de datos que se encuentra anexa a este documento, se puede observar cómo el sensor sólo necesita un pin para enviar los datos al Arduino, así que la conexión se realizó al **pin 2**.

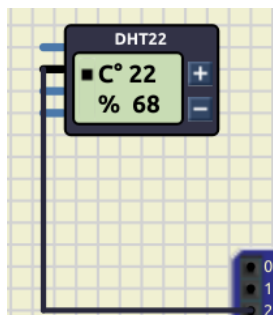


Figura 5: Conexión del sensor DHT22. Imagen propia.

Para la pantalla, se utilizó la biblioteca PCD8544.h que sugiere conectar SCLK al **pin3**, SDIN al **pin4**, DC al **pin5**, SCE al **pin7** y RST al **pin6**. También existen otras conexiones físicas que le dan poder a la pantalla, sin embargo a nivel de simulación estas no son necesarias. [2]

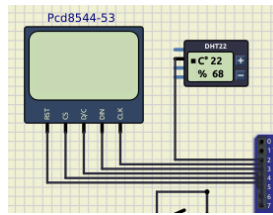


Figura 6: Conexión de la pantalla PCD8544. Imagen propia.

Para la conexión del switch se requiere conectar un lado al **pin 8** que va a servir como entrada, y del otro lado a V_{in} , de esta manera cuando se cierra el switch, le llega los 5V que entrega este pin al pin de entrada.

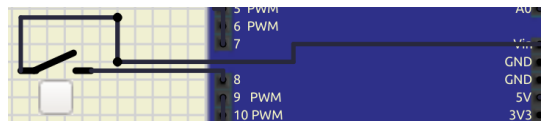


Figura 7: Conexión del switch. Imagen propia.

Para la conexiones de los LEDs rojo y azul se utilizan los **pines 13 y 12** respectivamente. A este circuito es necesario agregar una resistencia de 100 ohms para que disminuya el valor de la corriente y sea similar a la que el LED es capaz de soportar. En caso contrario los LEDs se quemarían.

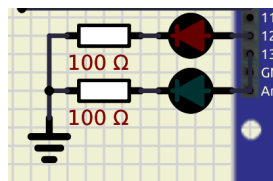


Figura 8: Conexión de los LEDs rojo y azul. Imagen propia.

El potenciómetro se conecta al **pin A0** ya que se quiere recibir una señal analógica del valor para que pueda ser variable. La salida del potenciómetro se conecta a este pin, su energía variable viene de V_{in} . El potenciómetro es de 1000 ohms ya que es un valor que permite variar su corriente sin tener valores muy limitados.

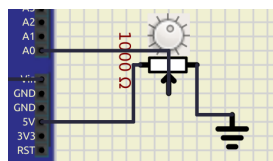


Figura 9: Conexión del potenciómetro. Imagen propia.

Para la resistencia de poder se crea un circuito más complejo. Como se mencionó anteriormente, el arduino sólo es capaz de entregar 5 V, y para lograr una potencia que logre calentar el circuito, se requiere una tensión mayor. En este caso se usa una fuente de poder externa de 12V, una resistencia de 28 ohms que nos va a dar una potencia de 5 W, un capacitor y un BJT.

Los arduinos son basados en circuitos digitales, es por eso que sólo pueden dar un valor de tensión fija de 5V. Por lo que para controlar una señal de salida de tensión variable, se utiliza una

técnica llamada Pulse Width Modulation que le permite a la señal enviar los 5V pero en una secuencia periódica suficientemente rápida que hace que parezca que se entrega una señal analógica. El arduino tiene una función llamada **AnalogWrite** que genera un pulso de tipo PWM. [3]

Ahora, esta señal igualmente solo tiene un máximo de 5V, por lo que el transistor BJT le permite generar un paso de corriente al circuito de 12 V en la misma frecuencia que lo hace la señal PWM. El capacitor presente en el circuito, le ayuda a la resistencia a tener un valor no tan variable de tensión sino que se vea más lineal aunque varíe la entrada.

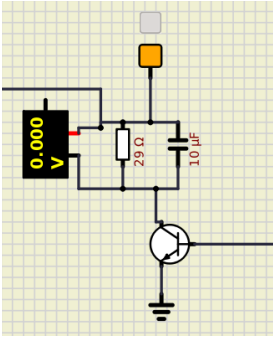


Figura 10: Conexión de la resistencia de calentamiento. Imagen propia.

En resumen, las conexiones presentes en el circuito se pueden ver de la siguiente manera:

Pin	Configuracion	Elemento	Funcion
PD2	Entrada	DHT22	Lectura de humedad y temperatura
PD3	Salida	PCD8544	Desplegar los datos de temperatura y humedad
PD4	Salida		
PD5	Salida		
PD6	Salida		
PD7	Salida		
PD8	Entrada	Switch	Controlar el envío de datos por medio de USARTS.
PD9	Salida	BJT	Controlar la señal de tensión para la resistencia.
PD12	Salida	LED azul	Indicar si la temperatura está muy baja.
PD13	Salida	LED rojo	Indicar si la temperatura está muy alta.
A0	Entrada	Potenciómetro	Dar un valor deseado para la incubadora.

3. Desarrollo/Análisis de resultados

Para analizar el funcionamiento de este circuito, a continuación se mostrará una serie de pasos de su secuencia que permita entender qué se quiere lograr antes de ver resultados:

1. A través del potenciómetro se escoge una temperatura deseada para la incubadora.
2. A través del sensor DHT22 se leen los valores de humedad y temperatura actuales de la incubadora.
3. Si la temperatura es menor de 30C se enciende el LED azul.
4. Si la temperatura es mayor de 42C se enciende el LED rojo.
5. Para conservar la temperatura deseada, el arduino procesa los valores de temperatura del sensor a través de un controlador de PID. Obteniendo una señal de salida para alcanzar la temperatura deseada.
6. La temperatura deseada, los valores leídos y la señal de salidad generada se despliegan en la pantalla.
7. Si el switch está cerrado, se envían los valores leídos a la computadora través de USARTs.
8. La señal de salida para regular el circuito se envía a la resistencia de poder.
9. La computadora crea un archivo .csv con los valores de temperatura y humedad leídos.

En el enlace <https://youtu.be/CpqqNG8jKV8> se puede ver una demostración de la simulación completa en SIMULIDE de la incubadora. Sin embargo, a continuación se mostrará un diagrama de flujo que explique como funciona y resultados de cada una de sus funciones individualmente.

3.1. Activación de LEDs

Los valores de temperatura y humedad se realizan a partir del sensor DHT22. Si el valor de temperatura se encuentra por debajo de los 30C, el LED azul se enciende. Si la temperatura es mayor a 42C, el led rojo se enciende, si se encuentra en el rango adecuado ningún LED se enciende. En la figura 11 se puede comprobar este funcionamiento.

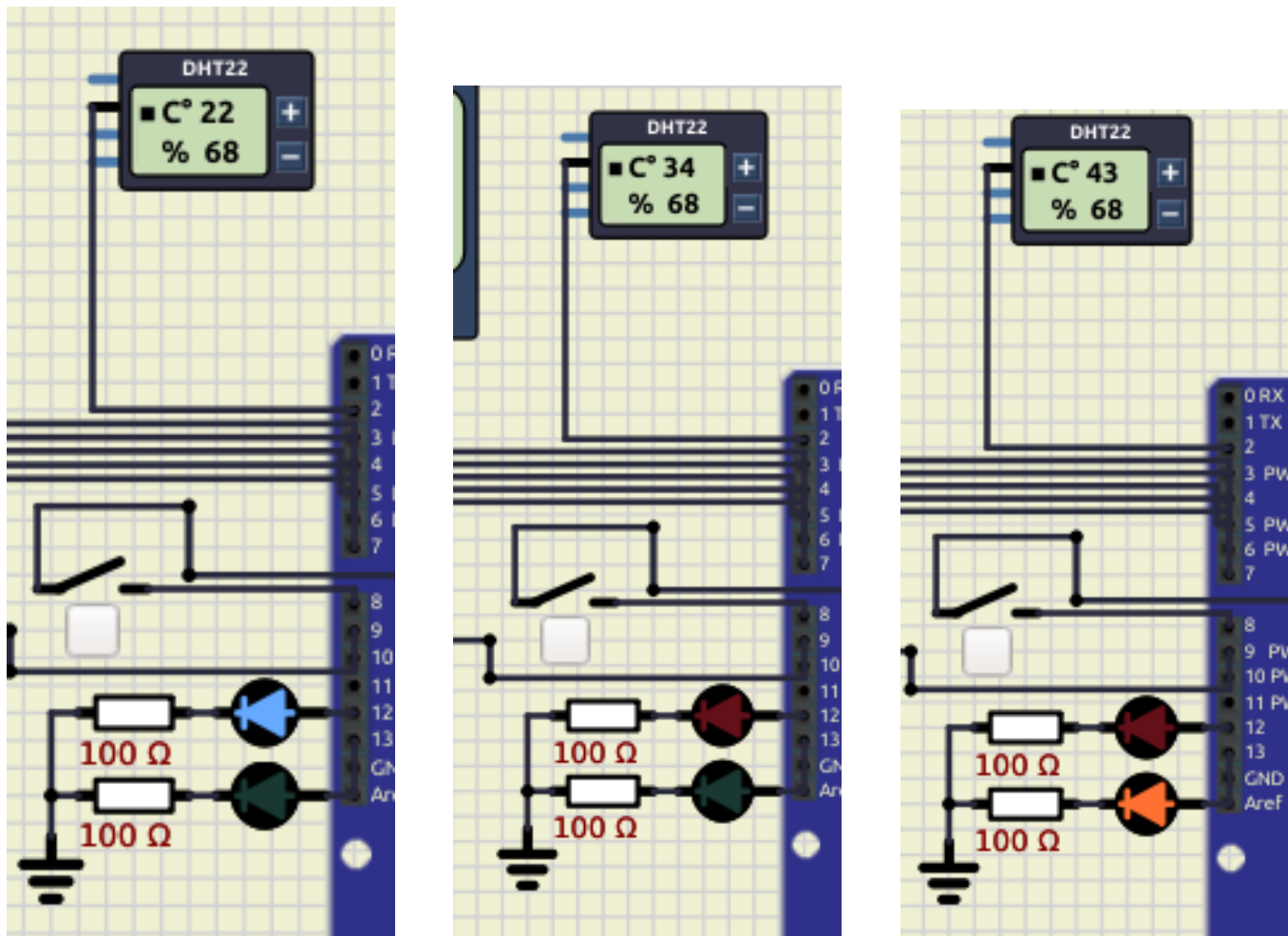


Figura 11: Funcionamiento de los LEDs azul y rojo.

3.2. Pantalla LCD muestra la temperatura de operación, la señal de control, la temperatura sensada y humedad de operación.

La pantalla led muestra la temperatura a la que se desea que opere, la señal de salida del controlador, la temperatura sensada y la humedad sensada. Es importante aclarar que la señal del controlador se muestra en grados Celcius a los cuáles la resistencia de poder deería estar. Este valor puede mostrar hasta 80C, porque requiere esta temperatura para llegar a la temperatura deseada de 42C. Además el valor de la resistencia no va a ser igual a la temperatura sensada en la incubadora, porque el espacio que calienta es muy grande en comparación con la resistencia.

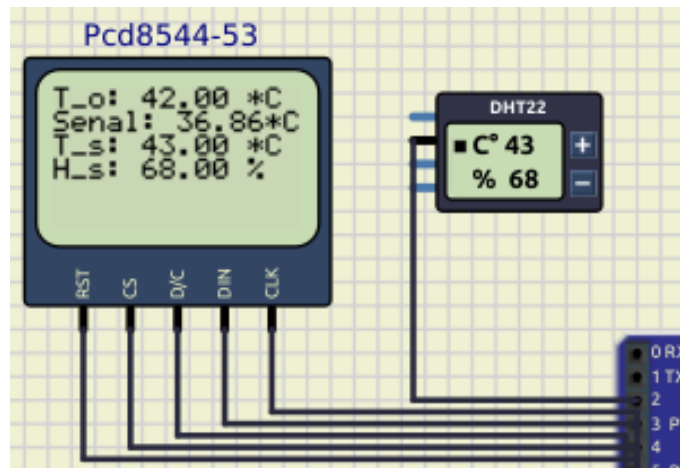


Figura 12: Funcionamiento de la pantalla LED. Imagen propia.

3.3. Funcionamiento de la resistencia de poder

La resistencia de poder obtiene sus valores a través de la señal PWM que está regida por la salida del controlador. En las siguientes imágenes se puede observar en la pantalla LED el valor de temperatura que debería tener la resistencia de salida y cómo la señal PWM (que se puede leer en el osciloscopio) se ajusta para llegar al valor deseado. En el caso de la temperatura alta, el osciloscopio muestra la señal PWM teniendo el valor en alto durante mucho más tiempo. En el caso de la temperatura baja se ve como la señal PWM tiene el valor en alto menos tiempo que el valor en alto. Además el voltímetro muestra el valor de la tensión que tiene la resistencia en cada uno de los casos.

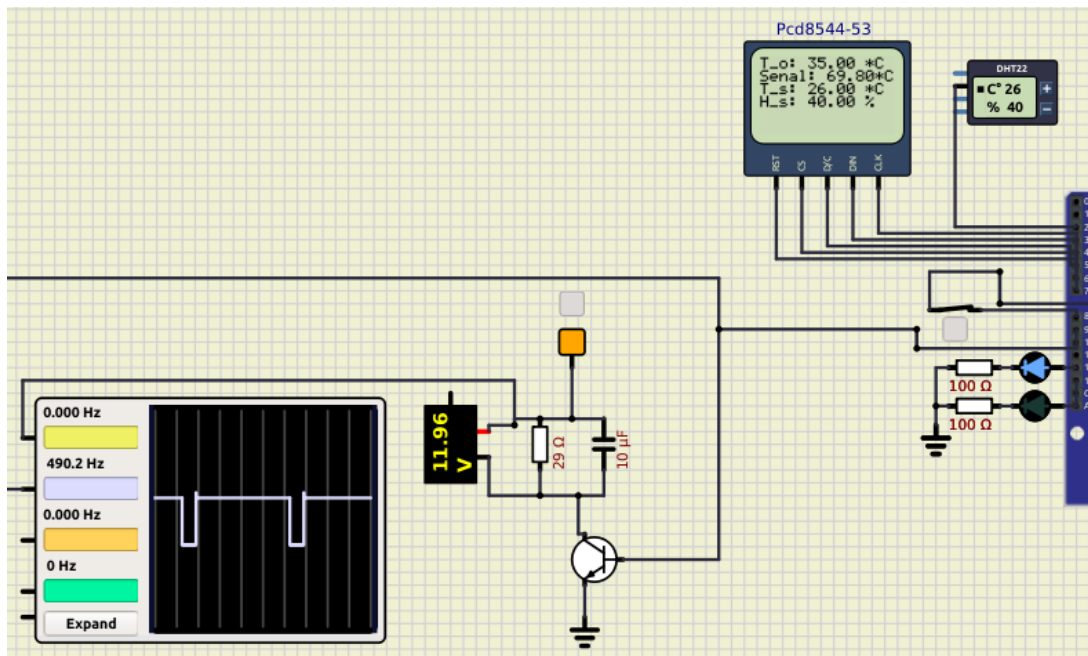


Figura 13: Temperatura alta.

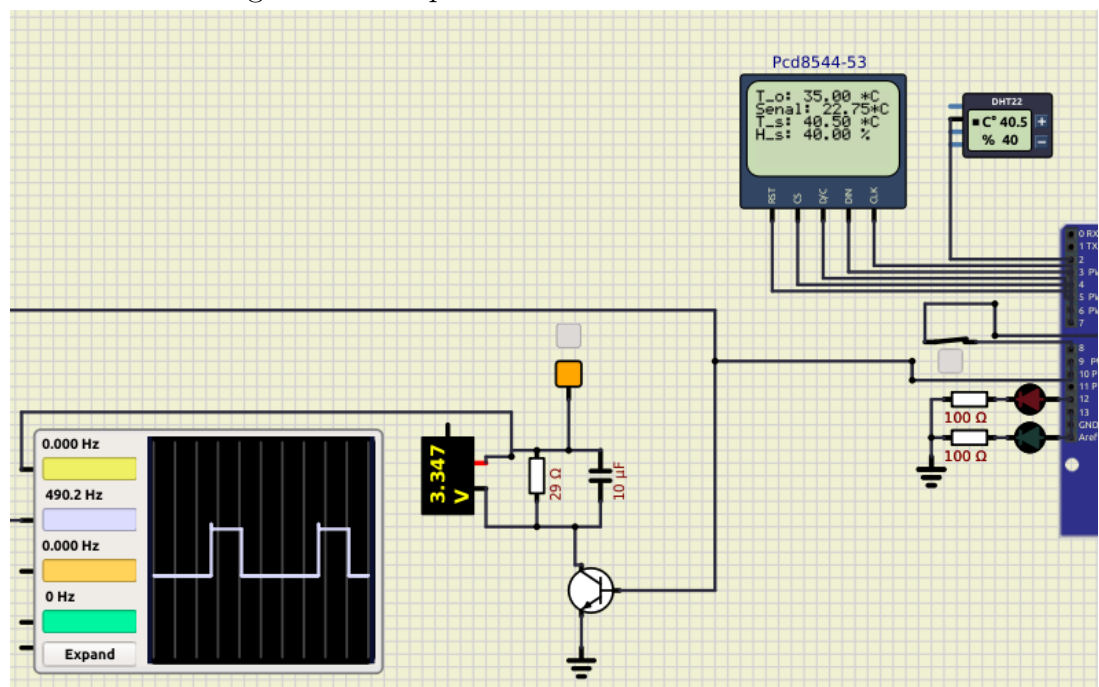


Figura 14: Temperatura baja.

Figura 15: Funcionamiento de la resistencia de poder. Figura propia.

3.4. Asignación de temperatura deseada a través del potenciómetro

En la figura 16 se muestra como cuando el potenciómetro está del lado derecho la temperatura deseada está en 30C y cuando se encuentra al lado izquierdo, es decir permitiendo mayor paso de corriente el valor se lee como 42C en la pantalla LED.

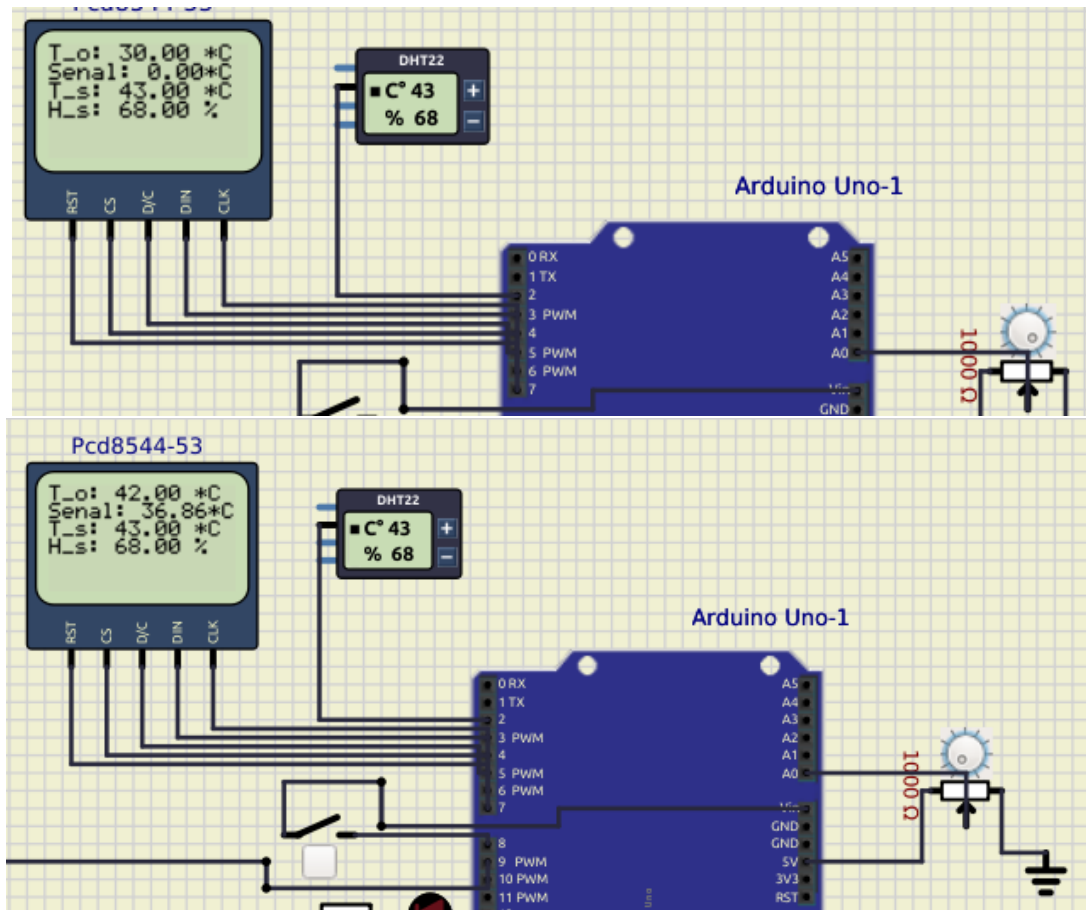


Figura 16: Funcionamiento de la asignación de temperatura deseada a través del potenciómetro. Figura propia.

3.5. Envío de valores sensados a través de USART según la activación del switch.

En la figura 17 se muestra cómo si el switch está desactivado no se envían datos a la computadora a pesar de que el sensor esté leyendo los datos. Cuando el sensor se activa se envía el valor de temperatura primero y luego la humedad. Estos valores se envían sin ninguna unidad o valor extra para disminuir la cantidad de datos que se envían pero los datos son trabajados al ser leídos por el script.

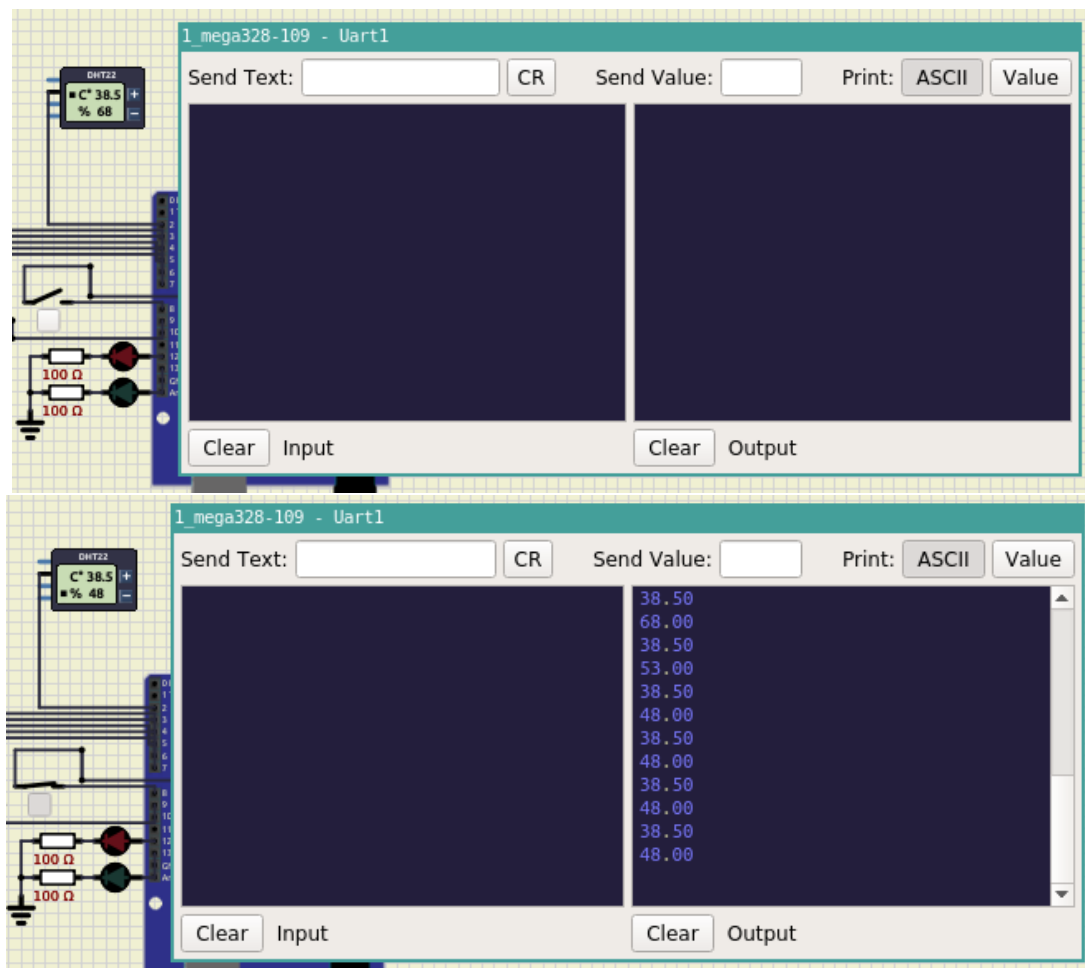


Figura 17: Funcionamiento del switch para la transmisión de datos.

3.6. Lectura de datos a través de script de Python.

Para la lectura de datos desde la computadora, no se pudo conectar el simulador al puerto de la computadora para hacer la transmisión de datos. Sin embargo, se programó un arduino UNO real, con un sumador que representara la temperatura y la letra **i** que representara la humedad. Los datos fueron recibidos por el script de python y generaron un archivo como el que se muestra en la figura 18.

```
1 Temperatura: 6, Humedad: , i
2 Temperatura: 7, Humedad: , i
3 Temperatura: 8, Humedad: , i
4 Temperatura: 9, Humedad: , i
5 Temperatura: 10, Humedad: , i
6 Temperatura: 11, Humedad: , i
7 Temperatura: 12, Humedad: , i
8 Temperatura: 13, Humedad: , i
9 Temperatura: 14, Humedad: , i
10 Temperatura: 15, Humedad: , i
11 Temperatura: 16, Humedad: , i
12 Temperatura: 17, Humedad: , i
13 Temperatura: 18, Humedad: , i
14 Temperatura: 19, Humedad: , i
15 Temperatura: 20, Humedad: , i
16 Temperatura: 21, Humedad: , i
17 Temperatura: 22, Humedad: , i
18 Temperatura: 23, Humedad: , i
19 Temperatura: 24, Humedad: , i
20 Temperatura: 25, Humedad: , i
```

Figura 18: Archivo .csv generado por el script de Python. Imagen propia.

4. Conclusiones y recomendaciones

- El arduino UNO facilita el uso del microcontrolador ATmega328P, ya que al estar integrado, el usuario no tiene que enfocarse en darle alimentación y los pines son mucho más fáciles de acceder.
- El protocolo USART permite que sólo un dispositivo esté escribiendo a un canal, sin embargo se puede realizar la lectura desde varios dispositivos. Si se quiere escribir desde varios, se recomienda utilizar un temporizador que permite acceder al canal en diferentes momentos.
- La técnica PWM permite crear senales digitales que simulen senales analógicas desde microcontroladores digitales que sólo pueden enviar un 1 o un 0. Para obtener una senal lineal deseada, se recomienda incluir un capacitor en el circuito fisico para suavizar la senal.
- Los pines digitales del Arduino se pueden utilizar como entradas y salidas pero los analógicos sólo como entradas.
- Realizar aplicaciones a través del Arduino puede traer soluciones de bajo costo a distintos problemas, ya que como se puede ver, con un valor precio muy bajo se puede obtener una incubadora automatizada.

Referencias

1. Arduino (2022) Arduino Uno R3. Recuperado de: <https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>
2. Rodrigues, C. (2020) PCD8544. Recuperado de: <https://github.com/carlosefr/pcd8544>
3. Aqib. (2018) Arduino PWM Tutorial. Recuperado de: <https://create.arduino.cc/projecthub/muhammad-aqib/arduino-pwm-tutorial-ae9d71>

5. Apéndice

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:

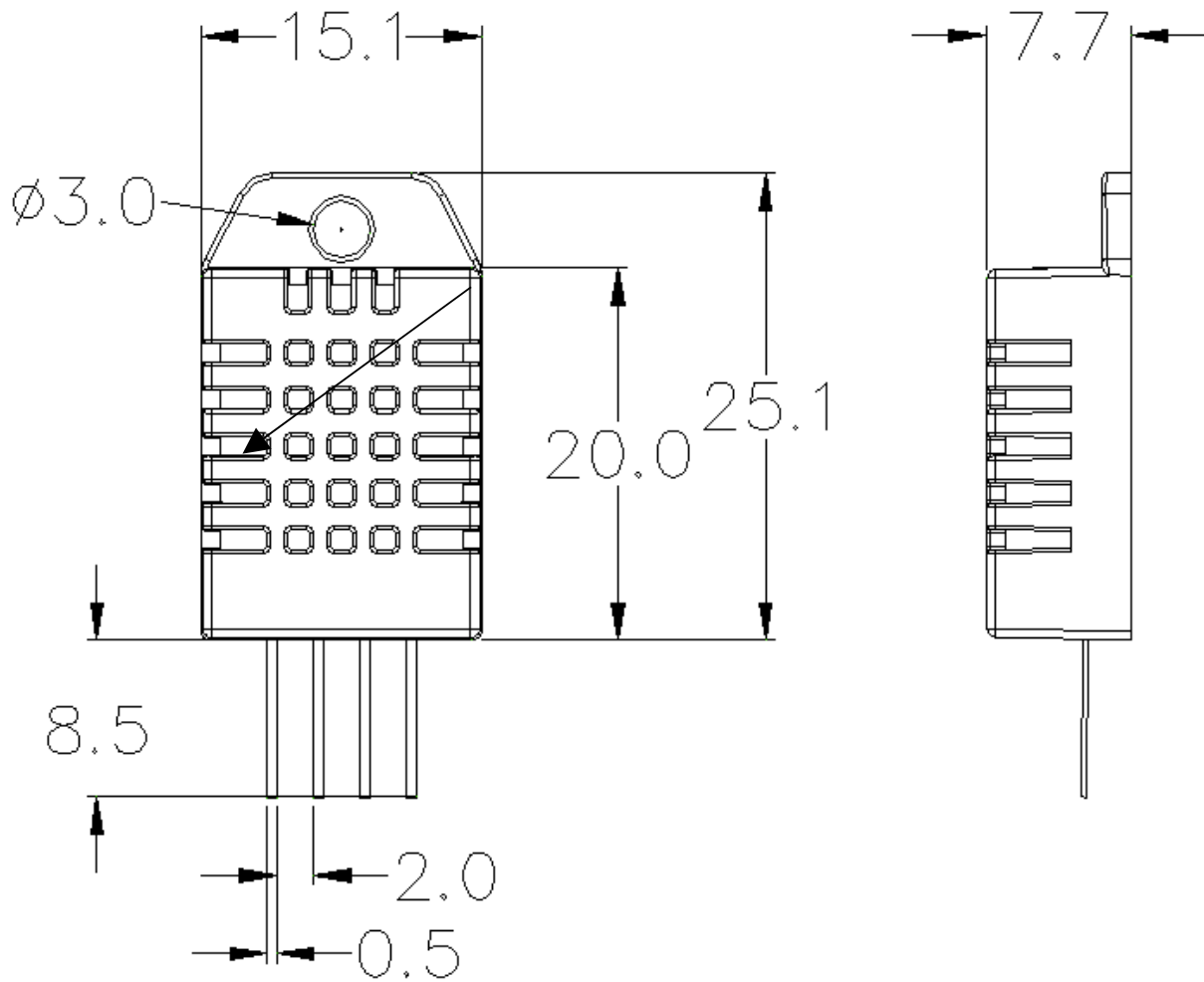
Model	DHT22
Power supply	3.3-6V DC
Output signal	digital signal via single-bus
Sensing element	Polymer capacitor
Operating range	humidity 0-100%RH; temperature -40~80Celsius
Accuracy	humidity +2%RH(Max +5%RH); temperature <+-0.5Celsius
Resolution or sensitivity	humidity 0.1%RH; temperature 0.1Celsius
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)

Aosong Electronics Co.,Ltd

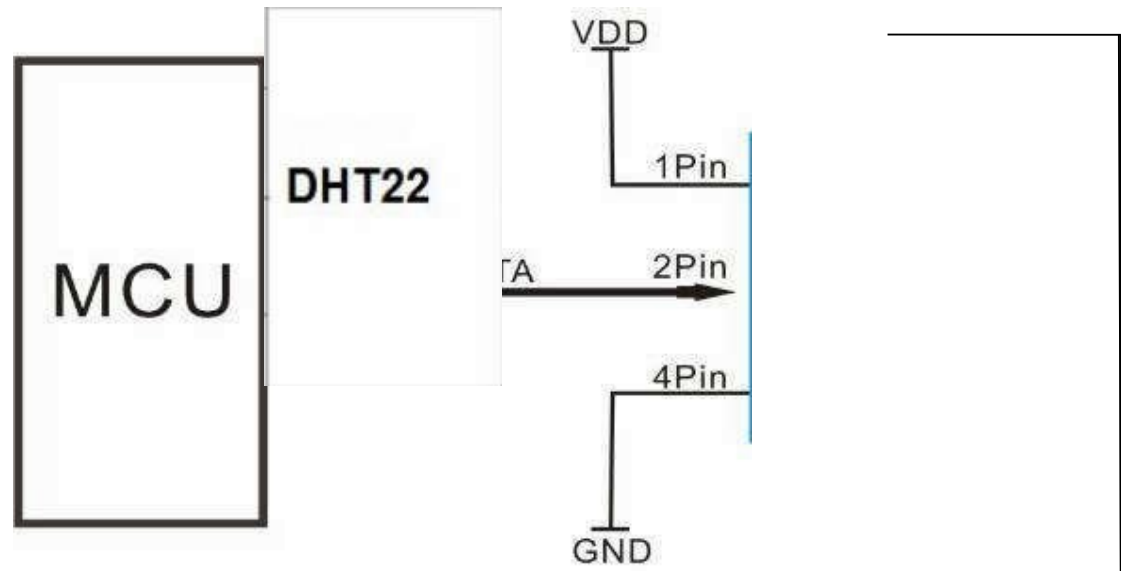
Your specialist in innovating humidity & temperature sensors



Pin sequence number: 1 2 3 4 (from left to right direction).

Pin	Function
1	VDD---power supply
2	DATA--signal
3	NULL
4	GND

5. Electrical connection diagram:



3Pin---NC, AM2302 is another name for DHT22

6. Operating specifications:

(1) Power and Pins

Power's voltage should be 3.3-6V DC. When power is supplied to sensor, don't send any instruction to the sensor within one second to pass unstable status. One capacitor valued 100nF can be added between VDD and GND for wave filtering.

(2) Communication and signal

Single-bus data is used for communication between MCU and DHT22, it costs 5mS for single time communication.

Data is comprised of integral and decimal part, the following is the formula for data.

DHT22 send out higher data bit firstly!

DATA=8 bit integral RH data+8 bit decimal RH data+8 bit integral T data+8 bit decimal T data+8 bit check-sum
If the data transmission is right, check-sum should be the last 8 bit of "8 bit integral RH data+8 bit decimal RH data+8 bit integral T data+8 bit decimal T data".

When MCU send start signal, DHT22 change from low-power-consumption-mode to running-mode. When MCU finishs sending the start signal, DHT22 will send response signal of 40-bit data that reflect the relative humidity

Aosong Electronics Co.,Ltd

Your specialist in innovating humidity & temperature sensors

8. Attentions of application:

(1) Operating and storage conditions

We don't recommend the applying RH-range beyond the range stated in this specification. The DHT22 sensor can recover after working in non-normal operating condition to calibrated status, but will accelerate sensors' aging.

(2) Attentions to chemical materials

Vapor from chemical materials may interfere DHT22's sensitive-elements and debase DHT22's sensitivity.

(3) Disposal when (1) & (2) happens

Step one: Keep the DHT22 sensor at condition of Temperature 50~60Celsius, humidity <10%RH for 2 hours;

Step two: After step one, keep the DHT22 sensor at condition of Temperature 20~30Celsius, humidity >70%RH for 5 hours.

(4) Attention to temperature's affection

Relative humidity strongly depend on temperature, that is why we use temperature compensation technology to ensure accurate measurement of RH. But it's still be much better to keep the sensor at same temperature when sensing.

DHT22 should be mounted at the place as far as possible from parts that may cause change to temperature.

(5) Attentions to light

Long time exposure to strong light and ultraviolet may debase DHT22's performance.

(6) Attentions to connection wires

The connection wires' quality will effect communication's quality and distance, high quality shielding-wire is recommended.

(7) Other attentions

- * Welding temperature should be bellow 260Celsius.

- * Avoid using the sensor under dew condition.

- * Don't use this product in safety or emergency stop devices or any other occasion that failure of DHT22 may cause personal injury.



Description

The Arduino UNO R3 is the perfect board to get familiar with electronics and coding. This versatile microcontroller is equipped with the well-known ATmega328P and the ATmega 16U2 Processor. This board will give you a great first experience within the world of Arduino.

Target areas:

Maker, introduction, industries



Features

- **ATMega328P Processor**
 - **Memory**
 - AVR CPU at up to 16 MHz
 - 32KB Flash
 - 2KB SRAM
 - 1KB EEPROM
 - **Security**
 - Power On Reset (POR)
 - Brown Out Detection (BOD)
 - **Peripherals**
 - 2x 8-bit Timer/Counter with a dedicated period register and compare channels
 - 1x 16-bit Timer/Counter with a dedicated period register, input capture and compare channels
 - 1x USART with fractional baud rate generator and start-of-frame detection
 - 1x controller/peripheral Serial Peripheral Interface (SPI)
 - 1x Dual mode controller/peripheral I2C
 - 1x Analog Comparator (AC) with a scalable reference input
 - Watchdog Timer with separate on-chip oscillator
 - Six PWM channels
 - Interrupt and wake-up on pin change
- **ATMega16U2 Processor**
 - 8-bit AVR® RISC-based microcontroller
- **Memory**
 - 16 KB ISP Flash
 - 512B EEPROM
 - 512B SRAM
 - debugWIRE interface for on-chip debugging and programming
- **Power**
 - 2.7-5.5 volts



1 The Board

1.1 Application Examples

The UNO board is the flagship product of Arduino. Regardless if you are new to the world of electronics or will use the UNO as a tool for education purposes or industry-related tasks.

First entry to electronics: If this is your first project within coding and electronics, get started with our most used and documented board; Arduino UNO. It is equipped with the well-known ATmega328P processor, 14 digital input/output pins, 6 analog inputs, USB connections, ICSP header and reset button. This board includes everything you will need for a great first experience with Arduino.

Industry-standard development board: Using the Arduino UNO board in industries, there are a range of companies using the UNO board as the brain for their PLC's.

Education purposes: Although the UNO board has been with us for about ten years, it is still widely used for various education purposes and scientific projects. The board's high standard and top quality performance makes it a great resource to capture real time from sensors and to trigger complex laboratory equipment to mention a few examples.

1.2 Related Products

- Starter Kit
- Tinkerkit Braccio Robot
- Example

2 Ratings

2.1 Recommended Operating Conditions

Symbol	Description	Min	Max
	Conservative thermal limits for the whole board:	-40 °C (-40°F)	85 °C (185°F)

NOTE: In extreme temperatures, EEPROM, voltage regulator, and the crystal oscillator, might not work as expected due to the extreme temperature conditions



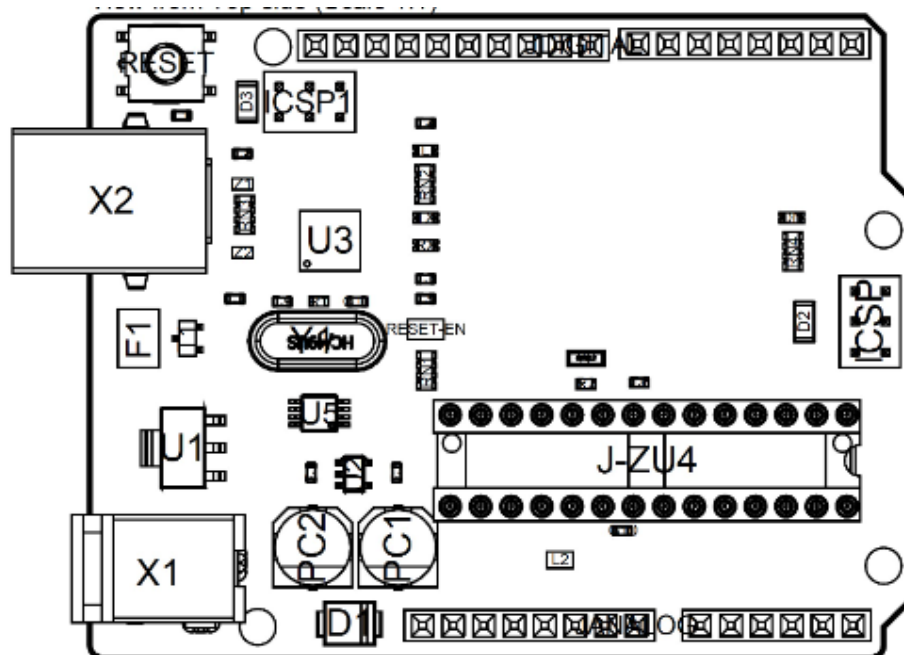
2.2 Power Consumption

Symbol	Description	Min	Typ	Max	Unit
VINMax	Maximum input voltage from VIN pad	6	-	20	V
VUSBMax	Maximum input voltage from USB connector		-	5.5	V
PMax	Maximum Power Consumption	-	-	xx	mA

3 Functional Overview

3.1 Board Topology

Top view



Board topology

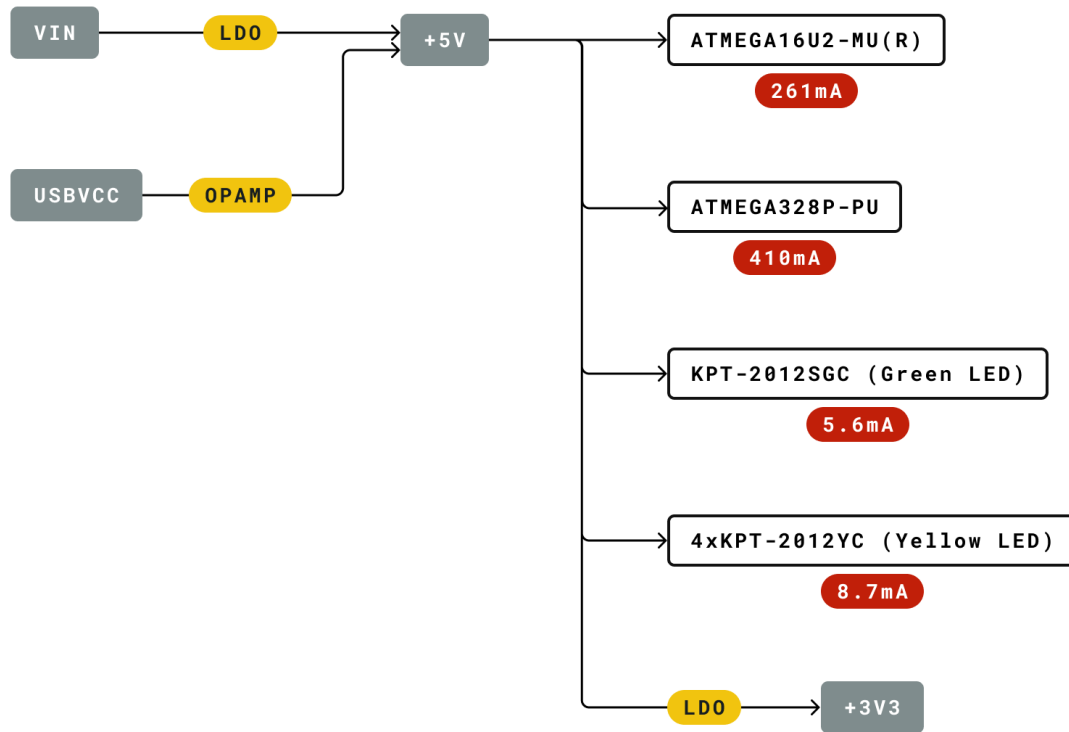
Ref.	Description	Ref.	Description
X1	Power jack 2.1x5.5mm	U1	SPX1117M3-L-5 Regulator
X2	USB B Connector	U3	ATMEGA16U2 Module
PC1	EEE-1EA470WP 25V SMD Capacitor	U5	LMV358LIST-A.9 IC
PC2	EEE-1EA470WP 25V SMD Capacitor	F1	Chip Capacitor, High Density
D1	CGRA4007-G Rectifier	ICSP	Pin header connector (through hole 6)
J-ZU4	ATMEGA328P Module	ICSP1	Pin header connector (through hole 6)
Y1	ECS-160-20-4X-DU Oscillator		



3.2 Processor

The Main Processor is a ATmega328P running at up to 20 MHz. Most of its pins are connected to the external headers, however some are reserved for internal communication with the USB Bridge coprocessor.

3.3 Power Tree



Legend:

- | | | |
|-----------|-------------|-----------------|
| Component | Power I/O | Conversion Type |
| | Max Current | Voltage Range |

Power tree



4 Board Operation

4.1 Getting Started - IDE

If you want to program your Arduino UNO while offline you need to install the Arduino Desktop IDE [1] To connect the Arduino UNO to your computer, you'll need a Micro-B USB cable. This also provides power to the board, as indicated by the LED.

4.2 Getting Started - Arduino Web Editor

All Arduino boards, including this one, work out-of-the-box on the Arduino Web Editor [2], by just installing a simple plugin.

The Arduino Web Editor is hosted online, therefore it will always be up-to-date with the latest features and support for all boards. Follow [3] to start coding on the browser and upload your sketches onto your board.

4.3 Getting Started - Arduino IoT Cloud

All Arduino IoT enabled products are supported on Arduino IoT Cloud which allows you to Log, graph and analyze sensor data, trigger events, and automate your home or business.

4.4 Sample Sketches

Sample sketches for the Arduino XXX can be found either in the "Examples" menu in the Arduino IDE or in the "Documentation" section of the Arduino Pro website [4]

4.5 Online Resources

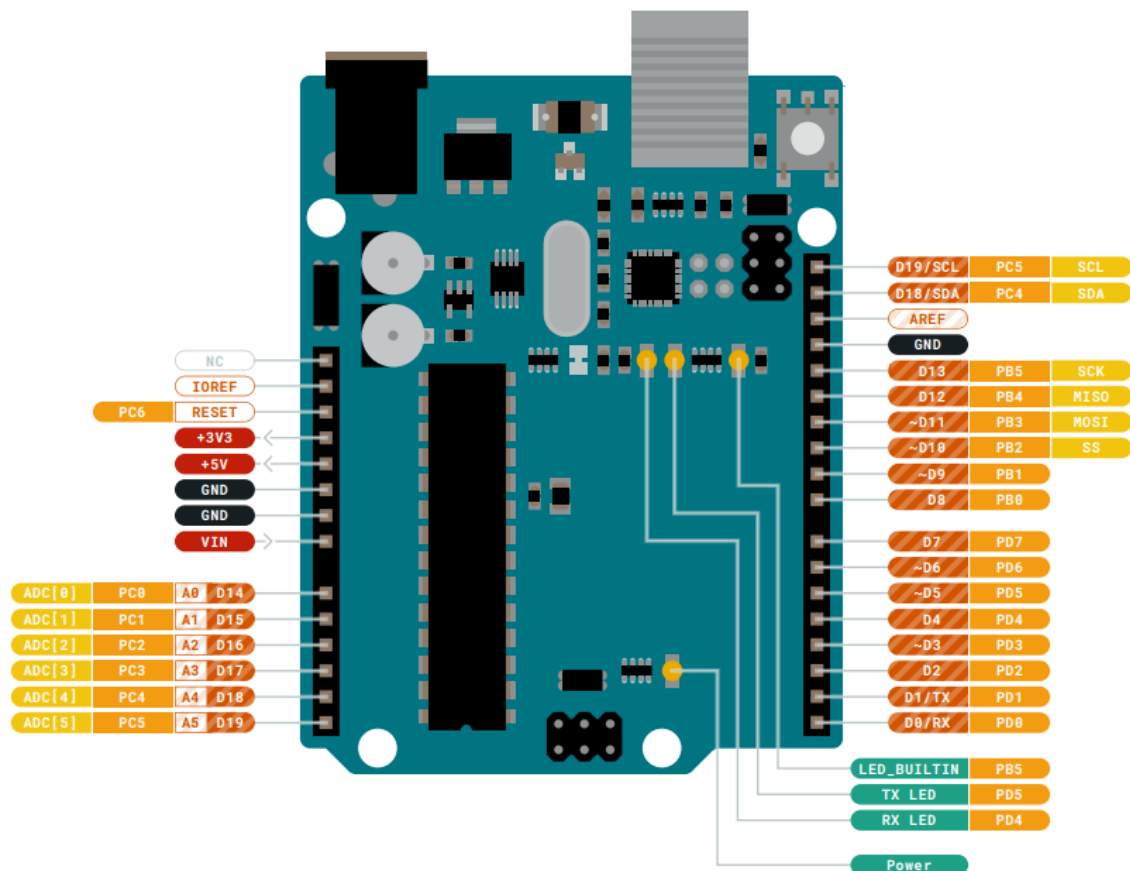
Now that you have gone through the basics of what you can do with the board you can explore the endless possibilities it provides by checking exciting projects on ProjectHub [5], the Arduino Library Reference [6] and the online store [7] where you will be able to complement your board with sensors, actuators and more



4.6 Board Recovery

All Arduino boards have a built-in bootloader which allows flashing the board via USB. In case a sketch locks up the processor and the board is not reachable anymore via USB it is possible to enter bootloader mode by double-tapping the reset button right after power up.

5 Connector Pinouts



Pinout



5.1 JANALOG

Pin	Function	Type	Description
1	NC	NC	Not connected
2	IOREF	IOREF	Reference for digital logic V - connected to 5V
3	Reset	Reset	Reset
4	+3V3	Power	+3V3 Power Rail
5	+5V	Power	+5V Power Rail
6	GND	Power	Ground
7	GND	Power	Ground
8	VIN	Power	Voltage Input
9	A0	Analog/GPIO	Analog input 0 /GPIO
10	A1	Analog/GPIO	Analog input 1 /GPIO
11	A2	Analog/GPIO	Analog input 2 /GPIO
12	A3	Analog/GPIO	Analog input 3 /GPIO
13	A4/SDA	Analog input/I2C	Analog input 4/I2C Data line
14	A5/SCL	Analog input/I2C	Analog input 5/I2C Clock line

5.2 JDIGITAL

Pin	Function	Type	Description
1	D0	Digital/GPIO	Digital pin 0/GPIO
2	D1	Digital/GPIO	Digital pin 1/GPIO
3	D2	Digital/GPIO	Digital pin 2/GPIO
4	D3	Digital/GPIO	Digital pin 3/GPIO
5	D4	Digital/GPIO	Digital pin 4/GPIO
6	D5	Digital/GPIO	Digital pin 5/GPIO
7	D6	Digital/GPIO	Digital pin 6/GPIO
8	D7	Digital/GPIO	Digital pin 7/GPIO
9	D8	Digital/GPIO	Digital pin 8/GPIO
10	D9	Digital/GPIO	Digital pin 9/GPIO
11	SS	Digital	SPI Chip Select
12	MOSI	Digital	SPI1 Main Out Secondary In
13	MISO	Digital	SPI Main In Secondary Out
14	SCK	Digital	SPI serial clock output
15	GND	Power	Ground
16	AREF	Digital	Analog reference voltage
17	A4/SD4	Digital	Analog input 4/I2C Data line (duplicated)
18	A5/SD5	Digital	Analog input 5/I2C Clock line (duplicated)