

Sistemas Embebidos en Tiempo Real

Maestría en Inteligencia Artificial
2024

1

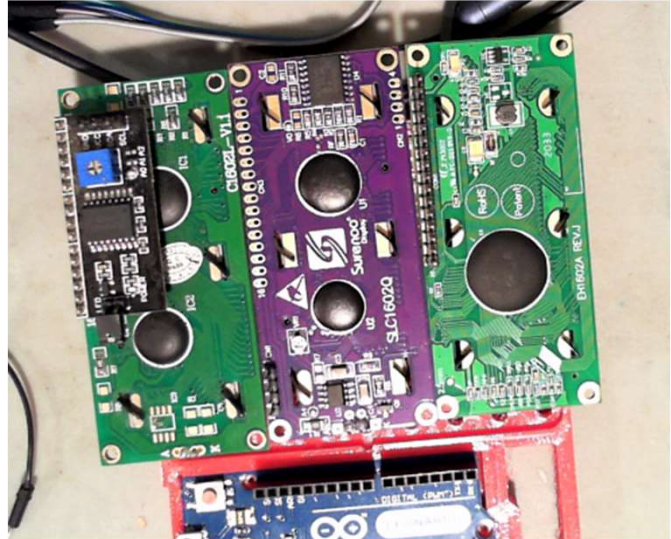
Sesión 19/11/2024

- Display alfanumérico 16x2 HD44780
- El módulo I2C LCD
- Interface del display 16x2 con el Arduino
- Ejemplo práctico

2

- Tres LCDs:

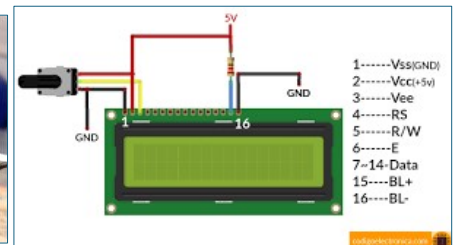
- El primero tiene montado una tarjeta para comunicación I2C
- El segundo tiene ambas interfaces (paralela e I2C)
- El tercero solo tiene interface paralela)



3

EL LCD alfanumérico HD44780

- Basado en el controlador Hitachi HD44780A
- Diferentes tamaños, desde 1x8 hasta 4x40
- Interface paralela de datos (4 ó 8 bits)
- Tiene control de contraste y luz de fondo
- Posee un ROM de caracteres predefinidos



4

El LCD alfanumérico HD44780

- ROM de caracteres:
 - Muy similar al código ASCII en 7 bits
 - El símbolo de grado (°) en ASCII es Alt+0167, en el ROM de caracteres del HD44780 es 0xDF
 - El símbolo “ñ” en ASCII es Alt+164, en el ROM de caracteres del HD44780 es 0xEE
 - Capacidad de ocho caracteres personalizados (CGRAM 0x00-0x07)
 - Para enviar un carácter directamente al display se emplea la función ENVIA_CHAR() (librería S_SAL)

Upper 8 bits		Lower 8 bits																
xxxx	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111		
xxxx0000					0	1	P	`	F				-	3	ε	α	ρ	
xxxx0001	(2)				!	1	A	Q	a	q			°	7	ç	ä	q	
xxxx0010	(3)				"	2	B	R	b	r			「	イ	ツ	×	β	θ
xxxx0011	(4)				#	3	C	S	c	s			」	ウ	て	ε	ω	
xxxx0100	(5)				\$	4	D	T	d	t			、	エ	ト	μ	ω	
xxxx0101	(6)				%	5	E	U	e	u			・	オ	ナ	1	ε	Ü
xxxx0110	(7)				&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)				'	7	G	W	g	w			フ	キ	ヲ	う	g	π
xxxx1000	(1)				(8	H	X	h	x			イ	ク	ネ	リ	フ	×
xxxx1001	(2))	9	I	Y	i	y			ゑ	ク	ノ	ル	"	γ
xxxx1010	(3)				*	:	J	Z	j	z			エ	コ	ハ	レ	j	≠
xxxx1011	(4)				+	;	K	L	k	l			オ	サ	ヒ	ロ	*	π
xxxx1100	(5)				,	<	L	¥	1	l			ハ	シ	フ	ワ	φ	π
xxxx1101	(6)				-	=	M	J	m	}			ユ	ズ	ン	も	÷	
xxxx1110	(7)				.	>	N	^	n	÷			ヨ	セ	ホ	"	ñ	
xxxx1111	(8)				/	?	O	_	o	€			ツ	リ	マ	"	ö	■

5

El LCD alfanumérico HD44780

- Tabla de caracteres ASCII de 7 bits
- Ref.
<http://www.asciitable.com>

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	#32;	Space	64	40	100	#64;	@	96	60	140	#96;	
1	1	001	SOH (start of heading)	33	21	041	#33;	!	65	41	101	#65;	A	97	61	141	#97;	a
2	2	002	STX (start of text)	34	22	042	#34;	"	66	42	102	#66;	B	98	62	142	#98;	b
3	3	003	ETX (end of text)	35	23	043	#35;	#	67	43	103	#67;	C	99	63	143	#99;	c
4	4	004	EOT (end of transmission)	36	24	044	#36;	\$	68	44	104	#68;	D	100	64	144	#100;	d
5	5	005	ENQ (enquiry)	37	25	045	#37;	%	69	45	105	#69;	E	101	65	145	#101;	e
6	6	006	ACK (acknowledge)	38	26	046	#38;	&	70	46	106	#70;	F	102	66	146	#102;	f
7	7	007	BEL (bell)	39	27	047	#39;	'	71	47	107	#71;	G	103	67	147	#103;	g
8	8	010	BS (backspace)	40	28	050	#40;	(72	48	110	#72;	H	104	68	150	#104;	h
9	9	011	TAB (horizontal tab)	41	29	051	#41;)	73	49	111	#73;	I	105	69	151	#105;	i
10	A	012	LF (NL line feed, new line)	42	2A	052	#42;	*	74	4A	112	#74;	J	106	6A	152	#106;	j
11	B	013	VT (vertical tab)	43	2B	053	#43;	+	75	4B	113	#75;	K	107	6B	153	#107;	k
12	C	014	FF (NP form feed, new page)	44	2C	054	#44;	,	76	4C	114	#76;	L	108	6C	154	#108;	l
13	D	015	CR (carriage return)	45	2D	055	#45;	-	77	4D	115	#77;	M	109	6D	155	#109;	m
14	E	016	SO (shift out)	46	2E	056	#46;	.	78	4E	116	#78;	N	110	6E	156	#110;	n
15	F	017	SI (shift in)	47	2F	057	#47;	/	79	4F	117	#79;	O	111	6F	157	#111;	o
16	10	020	DLE (data link escape)	48	30	060	#48;	0	80	50	120	#80;	P	112	70	160	#112;	p
17	11	021	DC1 (device control 1)	49	31	061	#49;	1	81	51	121	#81;	Q	113	71	161	#113;	q
18	12	022	DC2 (device control 2)	50	32	062	#50;	2	82	52	122	#82;	R	114	72	162	#114;	r
19	13	023	DC3 (device control 3)	51	33	063	#51;	3	83	53	123	#83;	S	115	73	163	#115;	s
20	14	024	DC4 (device control 4)	52	34	064	#52;	4	84	54	124	#84;	T	116	74	164	#116;	t
21	15	025	NAK (negative acknowledge)	53	35	065	#53;	5	85	55	125	#85;	U	117	75	165	#117;	u
22	16	026	SYN (synchronous idle)	54	36	066	#54;	6	86	56	126	#86;	V	118	76	166	#118;	v
23	17	027	ETB (end of trans. block)	55	37	067	#55;	7	87	57	127	#87;	W	119	77	167	#119;	w
24	18	030	CAN (cancel)	56	38	070	#56;	8	88	58	130	#88;	X	120	78	170	#120;	x
25	19	031	EM (end of medium)	57	39	071	#57;	9	89	59	131	#89;	Y	121	79	171	#121;	y
26	1A	032	SUB (substitute)	58	3A	072	#58;	:	90	5A	132	#90;	Z	122	7A	172	#122;	z
27	1B	033	ESC (escape)	59	3B	073	#59;	;	91	5B	133	#91;	[123	7B	173	#123;	{
28	1C	034	FS (file separator)	60	3C	074	#60;	<	92	5C	134	#92;	\	124	7C	174	#124;	
29	1D	035	GS (group separator)	61	3D	075	#61;	=	93	5D	135	#93;]	125	7D	175	#125;	}
30	1E	036	RS (record separator)	62	3E	076	#62;	>	94	5E	136	#94;	^	126	7E	176	#126;	~
31	1F	037	US (unit separator)	63	3F	077	#63;	?	95	5F	137	#95;	_	127	7F	177	#127;	DEL

Source: www.LookupTables.com

6

Caracteres personalizados en el LCD 16x2 HD44780

- Podemos incluir caracteres personalizados en nuestras visualizaciones.
- Cada caracter corresponde a una matriz de 8x5 (incluyendo el área del cursor)
- Disponible hasta **ocho** caracteres personalizados.
- Generador online de caracteres personalizados:
 - <https://maxpromer.github.io/LCD-Character-Creator/>

7

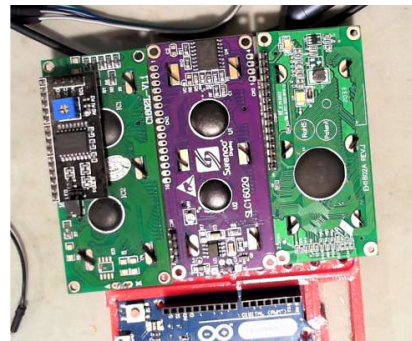
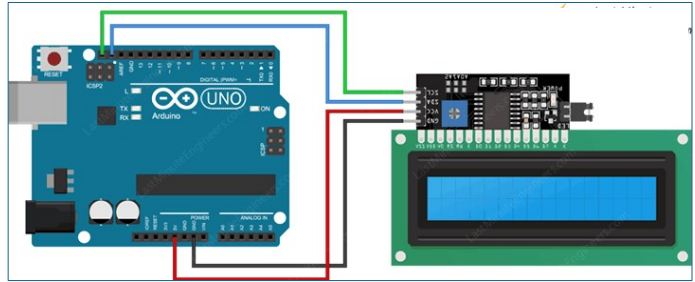
Información técnica para el LCD

- Referencia: Hoja técnica del HD44780
 - http://academy.cba.mit.edu/classes/output_devices/44780.pdf
- Video de manipulación de LCD sin microcontrolador:
 - https://www.youtube.com/watch?v=cXpeTxC3_A4
- Librería a emplear con Arduino:
 - Liquidcrystal_I2C de Frank de Brabander

8

Interface del LCD 2x16 HD44780 en I2C

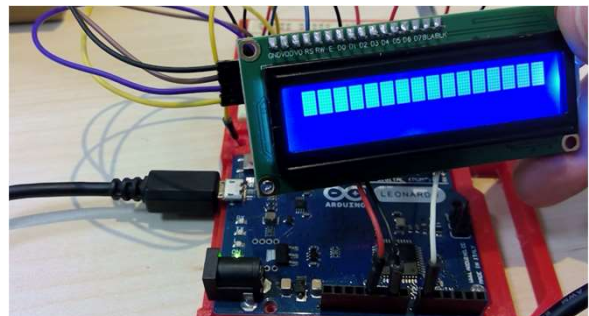
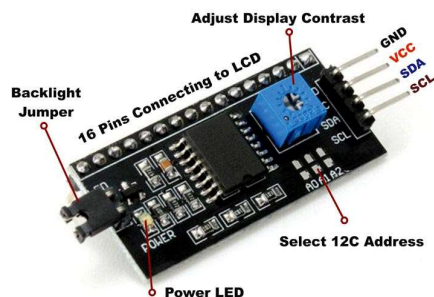
- Se le agrega un módulo adicional al LCD y permite la interface a dicho LCD usando I2C (comunicación serial), dicho módulo está basado en el circuito integrado PCF8574 (expansor de puertos vía I2C)
- Permite minimizar la cantidad de pines entre el LCD y el microcontrolador.
- Se requiere implementar una librería para poder lograr la interface
- Es mas fácil pero tiene un costo mayor



9

Ajuste inicial del contraste

- Ajustar el contraste con el potenciómetro que esta en el circuito I2C montado en el LCD hasta que se vea iluminado la línea superior:



10

Librería Liquidcrystal_I2C

- Para utilizar la librería previamente se debe de llamar a Wire.h necesario para que funcione el I2C del Arduino UNO
- Wire.h establece para el Arduino UNO el puerto A4 para SDA y A5 para SCL
- En el Arduino UNO hay pines SDA y SCL en la parte superior del módulo

The screenshot shows the LiquidCrystal_I2C library code in a code editor. The code includes the following lines:

```

1 //YUROBOT
2 #ifndef LiquidCrystal_I2C_h
3 #define LiquidCrystal_I2C_h
4
5 #include <inttypes.h>
6 #include "Print.h"
7 #include <Wire.h>
8
9 // commands
10 #define LCD_CLEARDISPLAY 0x01
11 #define LCD_RETURNHOME 0x02
12 #define LCD_ENTRYMODESET 0x04
13 #define LCD_DISPLAYCONTROL 0x08
14 #define LCD_CURSORSHIFT 0x10
15 #define LCD_FUNCTIONSET 0x20
16 #define LCD_SETCGRAMMOR 0x40
17 #define LCD_SETDDRAMADDR 0x80
  
```

Arrows point from the code to the Wire library reference page, which shows the default pins for the Arduino Uno R3: A4(SDA) and A5(SCL). An arrow also points to the physical pins on the Arduino Uno board.

11

Ejemplo de visualización de mensajes en el LCD I2C

The screenshot shows the Wokwi simulation environment. The sketch code is as follows:

```

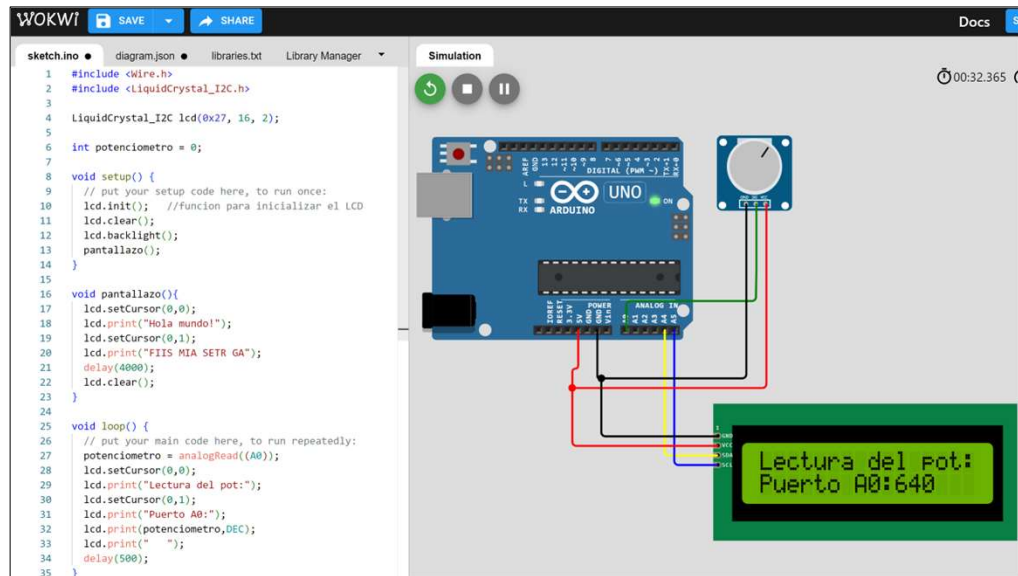
1 #include <Wire.h>
2 #include <LiquidCrystal_I2C.h>
3
4 LiquidCrystal_I2C lcd(0x27, 16, 2);
5
6 void setup() {
7   // put your setup code here, to run once:
8   lcd.init(); //funcion para inicializar el LCD
9   lcd.clear();
10  lcd.backlight();
11  pantalla();
12 }
13
14 void pantalla(){
15   lcd.setCursor(0,0);
16   lcd.print("Hola mundo!");
17   lcd.setCursor(0,1);
18   lcd.print("FIIS MIA SETR GA");
19   delay(4000);
20   lcd.clear();
21 }
22
23
24 void loop() {
25   // put your main code here, to run repeatedly:
26
27 }
  
```

The simulation shows an Arduino Uno connected to an LCD screen. The screen displays the text "Hola mundo!" on the first line and "FIIS MIA SETR GA" on the second line.

12

Agregando al ejemplo anterior un potenciómetro en A0

- Simulación:



15

Agregando al ejemplo anterior un potenciómetro en A0

- El parámetro de formato de visualización de variable DEC en la función lcd.print() solo muestra los dígitos válidos del valor de variable, si dicha variable tiene como valor 24, se mostrará 24 corriendo la visualización a la izquierda,
- Mejorando la visualización de la variable que aloja el resultado de la conversión A/D del canal A0 aplicando visualización individual de todos los dígitos.

```

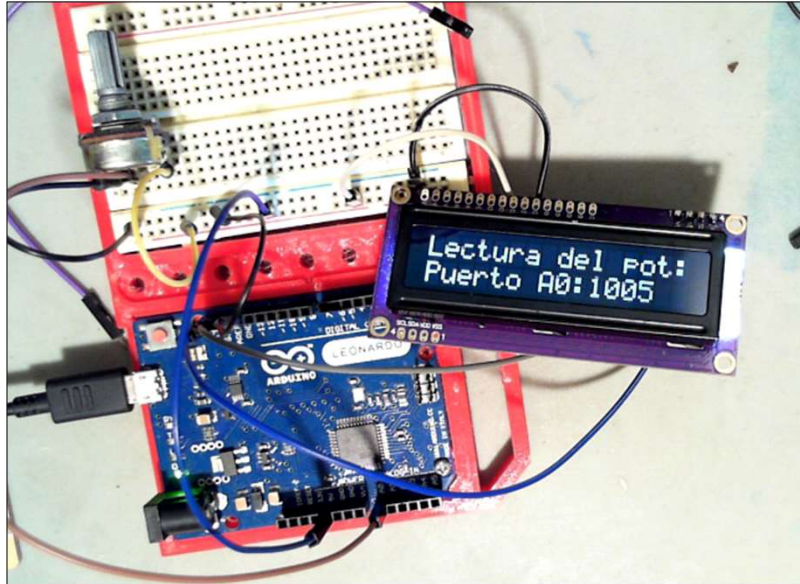
sketch_nov19a.ino
1 #include <Wire.h>
2 #include <LiquidCrystal_I2C.h>
3 LiquidCrystal_I2C lcd(0x27, 16, 2);
4
5 int potenciometro = 0;
6
7 void setup() {
8   // put your setup code here, to run once:
9   lcd.init(); //funcion para inicializar el LCD
10  lcd.clear();
11  lcd.backlight();
12  pantallazo();
13 }
14
15 > void pantallazo(){...
16 }
17
18 void loop() {
19   // put your main code here, to run repeatedly:
20   potenciometro = analogRead(A0);
21   lcd.setCursor(0,0);
22   lcd.print("Lectura del pot:");
23   lcd.setCursor(0,1);
24   lcd.print("Puerto A0:");
25   lcd.print(potenciometro/1000,DEC); //muestre el dígito millar
26   lcd.print((potenciometro%1000)/100,DEC); //muestre el dígito centena
27   lcd.print((potenciometro%100)/10,DEC); //muestre el dígito decena
28   lcd.print(potenciometro%10,DEC); //muestre el dígito unidad
29   //lcd.print(" ");
30   delay(50);
31 }

```

16

Agregando al ejemplo anterior un potenciómetro en A0

- Pruebas en físico:



17

Sesión 23/11/2024

- Medición de variables físicas: Medición de temperatura y humedad relativa del ambiente
- El sensor de humedad y temperatura DHT11
- Interface del sensor DHT11 con un sistema embebido
- Ejemplo de aplicación

18

Sensores de temperatura

- Dependiendo de la naturaleza de funcionamiento:
 - Dispositivos que aumentan o reducen su resistencia eléctrica según la temperatura: Termistor NTC ó PTC, PT100, PT1000.
 - Diferencia de tensión eléctrica basado en la unión de dos metales: termopar, temocupla.
 - Basado en semiconductor: LM35, diodos de temperatura.
 - Sensores digitales, donde la salida es un dato digital: DS18B20, DHT11, SHT11, DHT22, BMP280, BME280.
 - Sensores infrarrojos: MLX90614



19

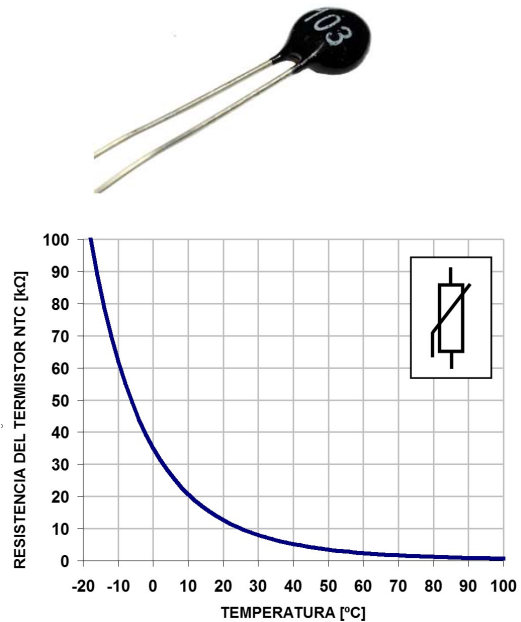
Sensores de temperatura

- Para seleccionar el sensor es indispensable tener en cuenta lo siguiente:
 - Rango de temperatura a medir
 - Precisión en la medición de la temperatura
 - Señal eléctrica de salida
 - Condiciones del entorno de trabajo
- Si el sensor arroja una señal eléctrica analógica, se deberá de emplear un conversor analógico/digital (A/D).
- La señal analógica debe de ser compatible con el rango de voltaje del canal analógico del conversor analógico a digital, de no serlo se deberá de acondicionar dicha señal mediante el uso de circuitos operacionales de amplificación y/o filtrado.

20

El termistor

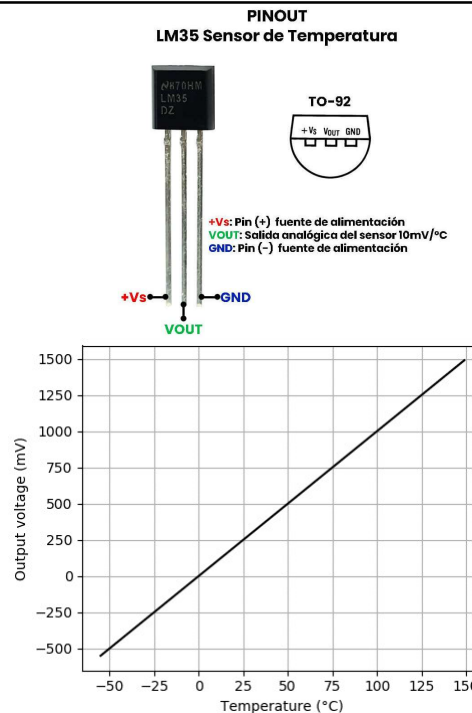
- Dependiendo del tipo:
 - NTC: Reduce su valor resistivo conforme aumenta la temperatura.
 - PTC: Aumenta su valor resistivo conforme aumenta la temperatura.
- Sencilla interface hacia un sistema embebido a través de un conversor analógico digital y un circuito divisor de tensión.
- Tener en cuenta que la respuesta no es lineal por lo que se deberá de compensar lo leído.



21

El LM35

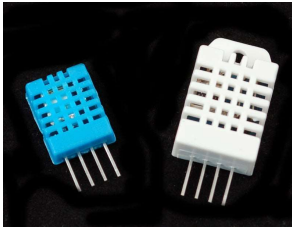
- Sensor basado en semiconductor.
- Respuesta lineal.
- Resolución 10mV/°C
- Sencilla interface hacia un sistema embebido a través de un conversor analógico digital y un circuito divisor de tensión.



22

DHT11/DHT22

- Sensores para medir la humedad y temperature del ambiente.
- Ampliamente usados en proyectos de sistemas embebidos con Arduino.



DHT11

- Ultra low cost
- 3 to 5V power and I/O
- 2.5mA max current use during conversion (while requesting data)
- Good for 20-80% humidity readings with 5% accuracy
- Good for 0-50°C temperature readings $\pm 2^\circ\text{C}$ accuracy
- No more than 1 Hz sampling rate (once every second)
- Body size 15.5mm x 12mm x 5.5mm
- 4 pins with 0.1" spacing

DHT22 / AM2302 (Wired version)

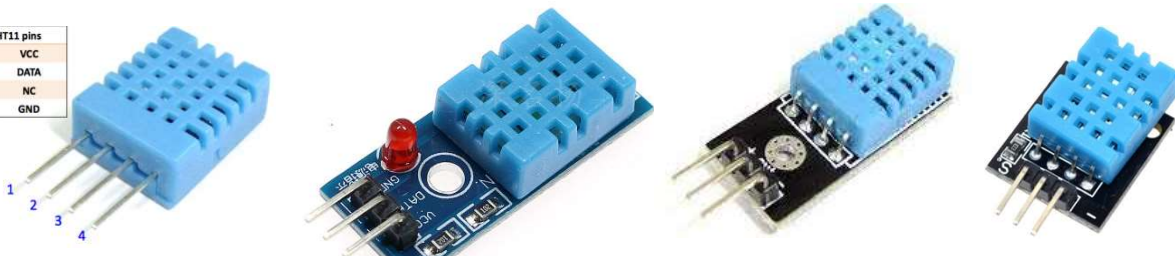
- Low cost
- 3 to 5V power and I/O
- 2.5mA max current use during conversion (while requesting data)
- Good for 0-100% humidity readings with 2-5% accuracy
- Good for -40 to 80°C temperature readings $\pm 0.5^\circ\text{C}$ accuracy
- No more than 0.5 Hz sampling rate (once every 2 seconds)
- Body size 15.1mm x 25mm x 7.7mm
- 4 pins with 0.1" spacing

23

El DHT11

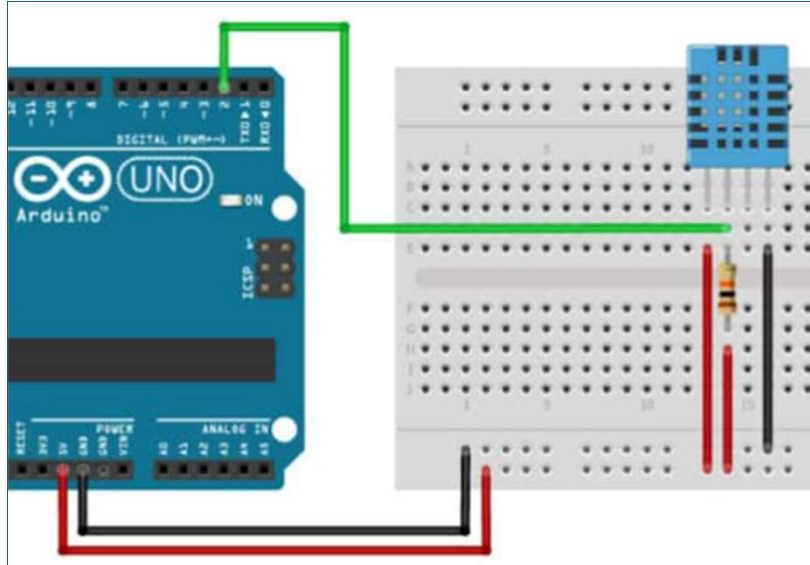
- Aspectos iniciales:
 - Al revisar la hoja técnica del DHT11 podemos ver que el DHT11 tiene un rango de voltaje de operación de 3V a 5.5V por lo que la conexión hacia el Curiosity Nano IC18F57Q43 será de manera directa.
 - Dependiendo del modelo de DHT11 puede que tenga integrado la resistencia de pull-up, sobre todo lo que tienen el sensor montado en una PCB:
 - Hoja técnica: <https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>

DHT11 pins	
1	VCC
2	DATA
3	NC
4	GND



24

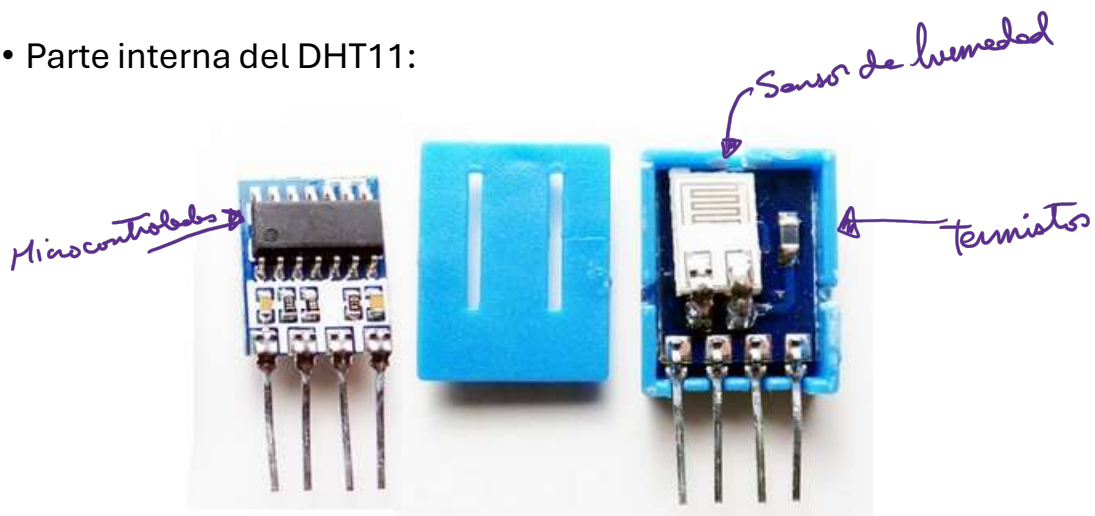
DHT11 solo el sensor:



25

EL DHT11

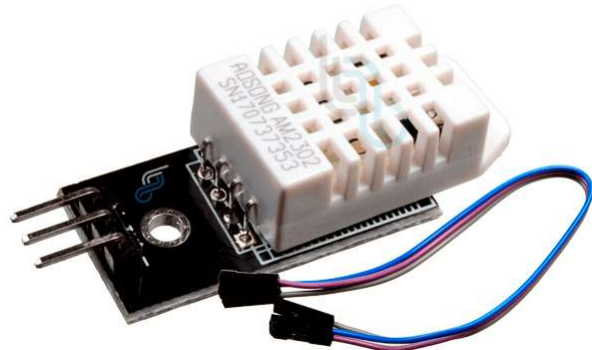
- Parte interna del DHT11:



26

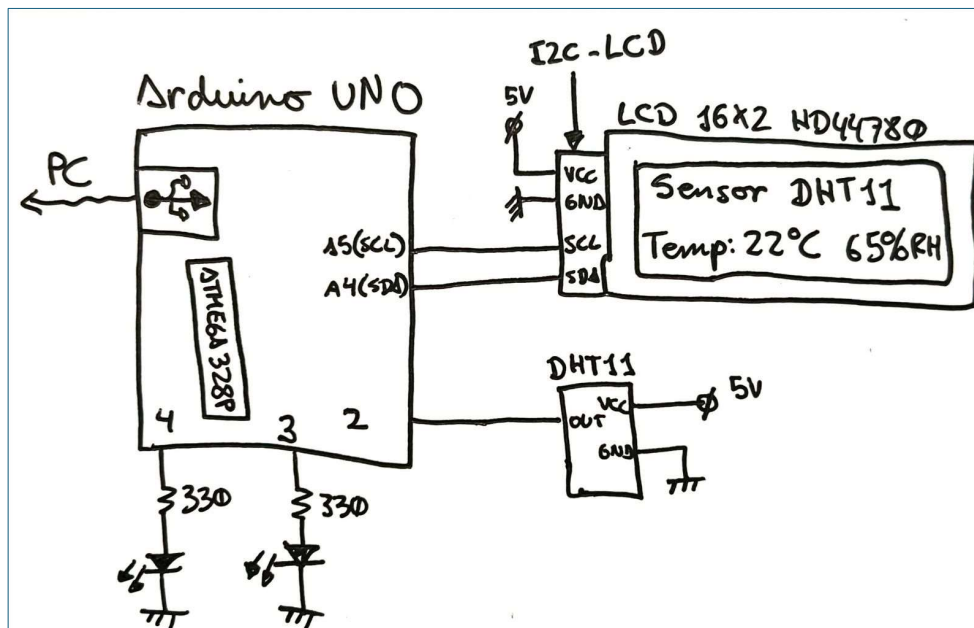
El DHT22

- También se le puede encontrar como AM2302 AOSONG
- De funcionamiento similar al DHT11, pero con mayor resolución en las medidas de temperatura y humedad



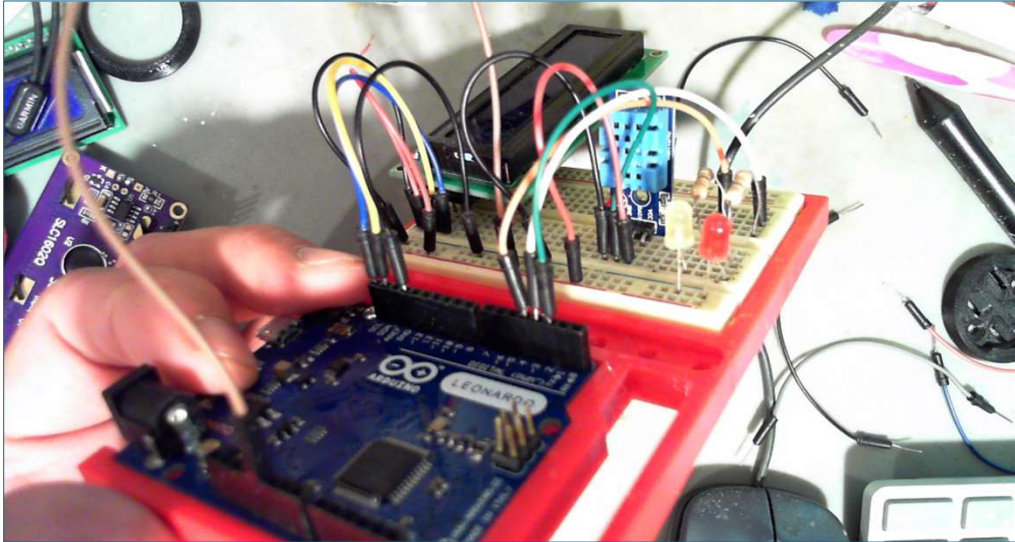
27

Interface del DHT11 con el Arduino UNO



28

Interface del DHT11 con el Arduino UNO



29

Código propuesto

```

1  #include <DHT.h>
2  #include <Wire.h>
3  #include <LiquidCrystal_I2C.h>
4
5  //declaracion de variables
6  float humedad = 0;
7  float temperatura = 0;
8
9  DHT dht(2, DHT11);    //configuracion del DHT(pin conectado y t
10 LiquidCrystal_I2C lcd(0x27,16,2); //conf del LCD I2C
11
12
13 void setup() {
14     // put your setup code here, to run once:
15     //inicializacion del LCD
16     lcd.init();
17     lcd.backlight();
18     //inicializacion del DHT11
19     dht.begin();
20     pantallazo();
21 }

```

```

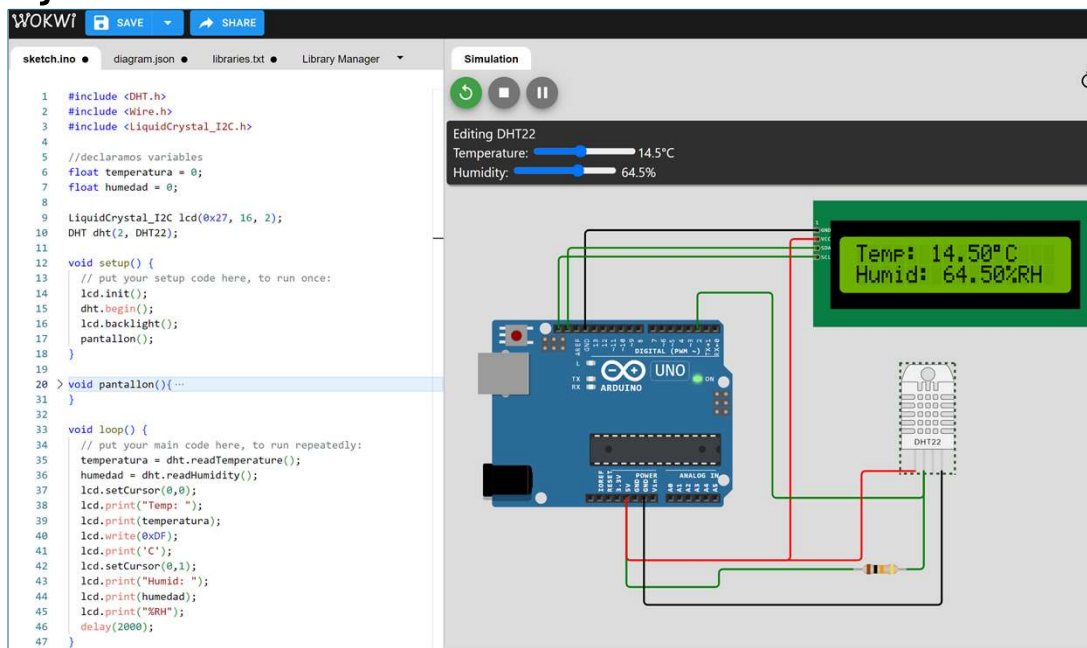
23 void pantallazo(){
24     lcd.setCursor(0,0);
25     lcd.print("Sensor DHT11");
26     lcd.setCursor(0,1);
27     lcd.print("Cargando...");
28     delay(3000);
29     lcd.clear();
30 }
31
32 void loop() {
33     // put your main code here, to run repeatedly
34     humedad = dht.readHumidity();
35     temperatura = dht.readTemperature();
36
37     lcd.setCursor(0,0);
38     lcd.print("Temp:");
39     lcd.print(temperatura);
40     lcd.write(0xDF);
41     lcd.print('C');
42     lcd.setCursor(0,1);
43     lcd.print("Humedad:");
44     lcd.print(humedad);
45     lcd.print("%RH");
46     delay(1000);
47 }

```

30

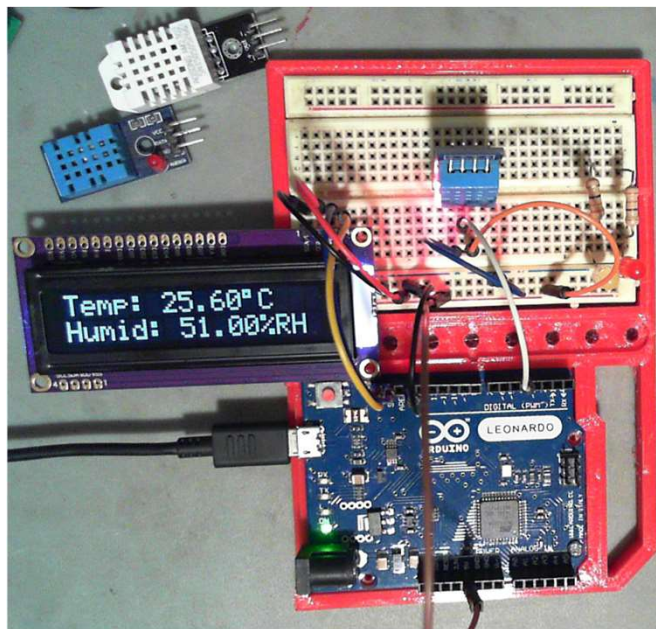
Ejercicio:

<https://wokwi.com/projects/415397096147178497>



31

Implementación:



32

Fin de la sesión