

MEDIDOR DE CALIDAD DEL AIRE PROYECTO

ALUMNOS: Asprilla Yacson, Cristian Moreno, Martin
Maciel, Nahuel Rodriguez.

PROFESOR: ARIEL TARAMAZZO

INSTITUTO SUPERIOR DE FORMACION TECNICA N°206





Por que surge
el proyecto



MEDIDOR DE CALIDAD DEL AIRE

Componentes

- Arduino UNO



Display 20x4
i2c

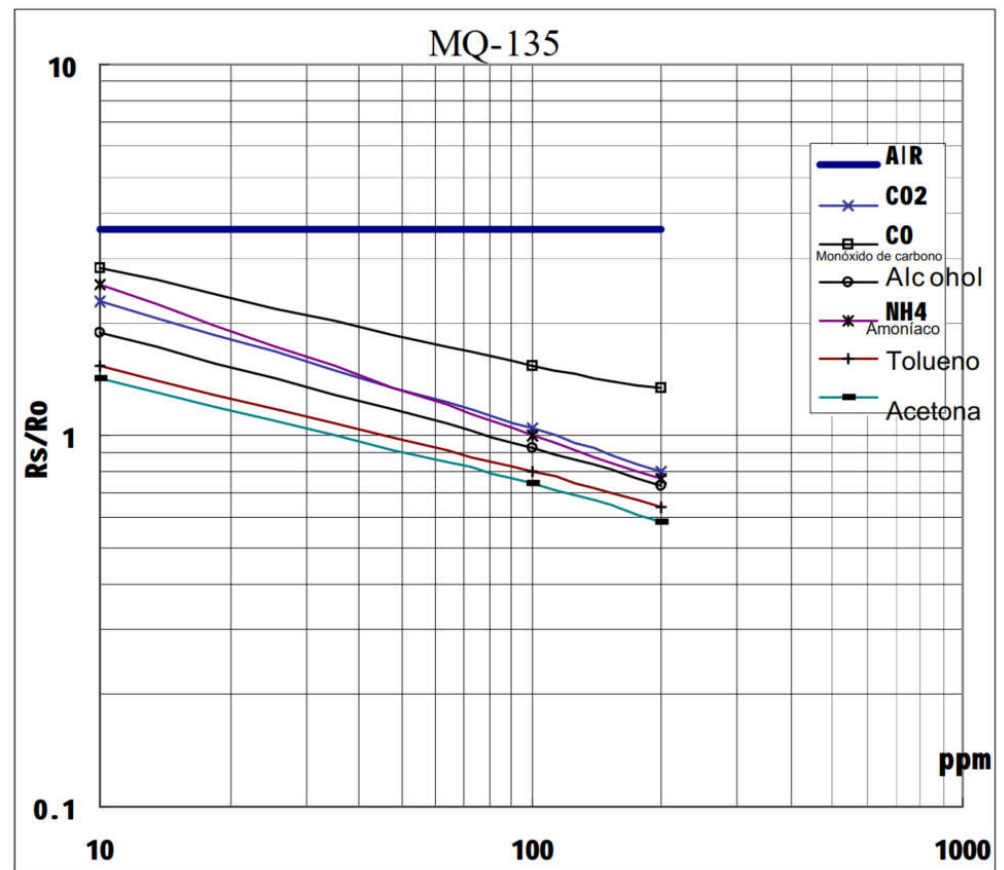
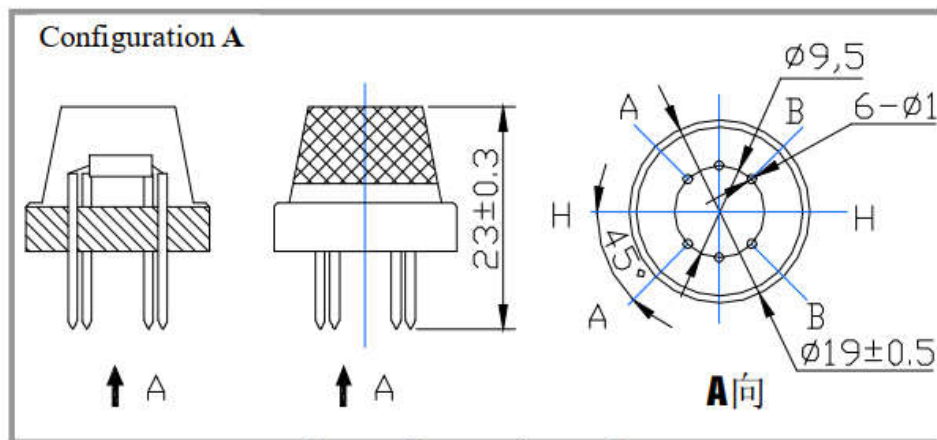


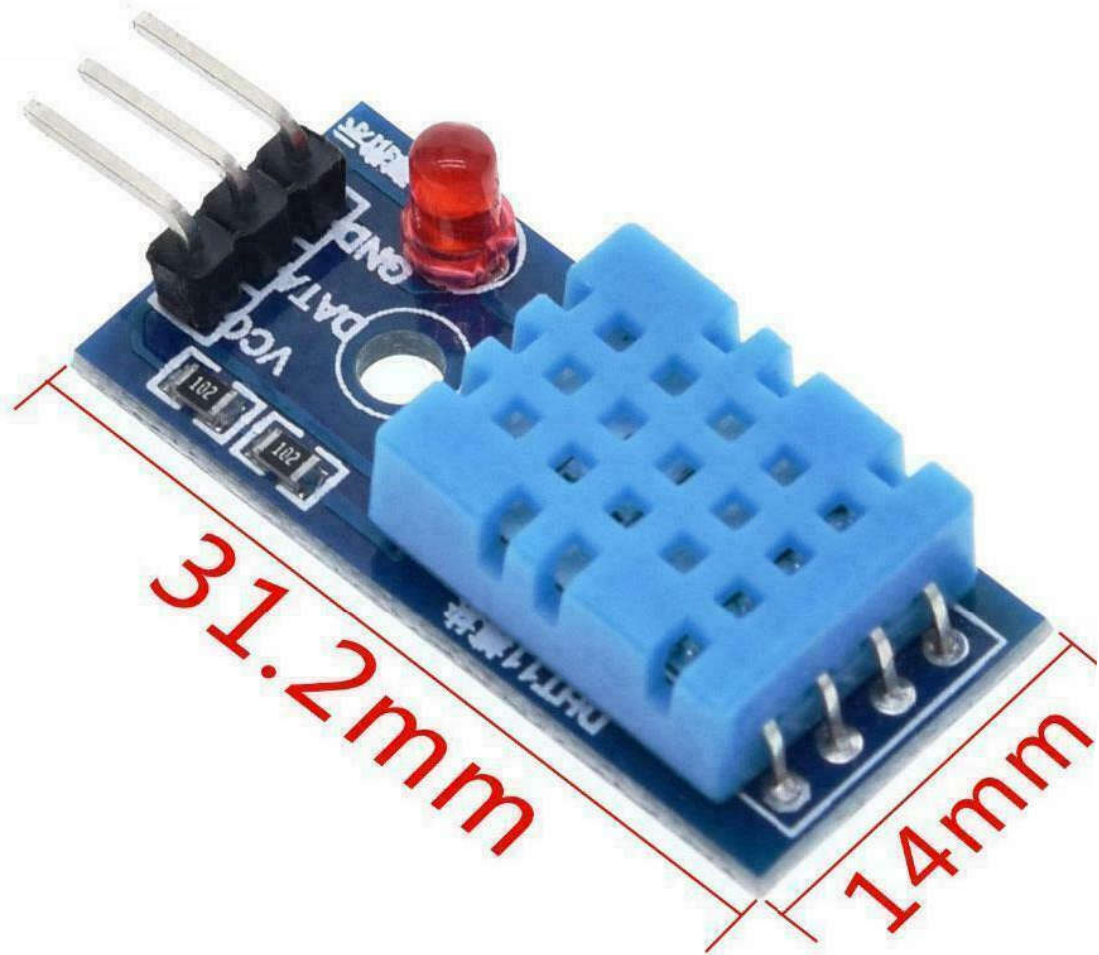
Familia sensores MQ

DETECCIÓN DE SUSTANCIAS

MODELO	SUSTANCIAS QUE DETECTA			
MQ-2	GLP	METANO	HUMO	PROPANO
MQ-4	METANO	GAS NATURAL COMPRIMIDO (GNP)		
MQ-5	GAS NATURAL	GAS LICUADO DE PETROLEO (GLP)		
MQ-7	MONOXIDO DE CARBONO			
MQ-9	MONOXIDO DE CARBONO		GASES INFLAMABLES	
MQ-135	DIOXIDO DE CARBONO		CALIDAD DEL AIRE	

MQ-135

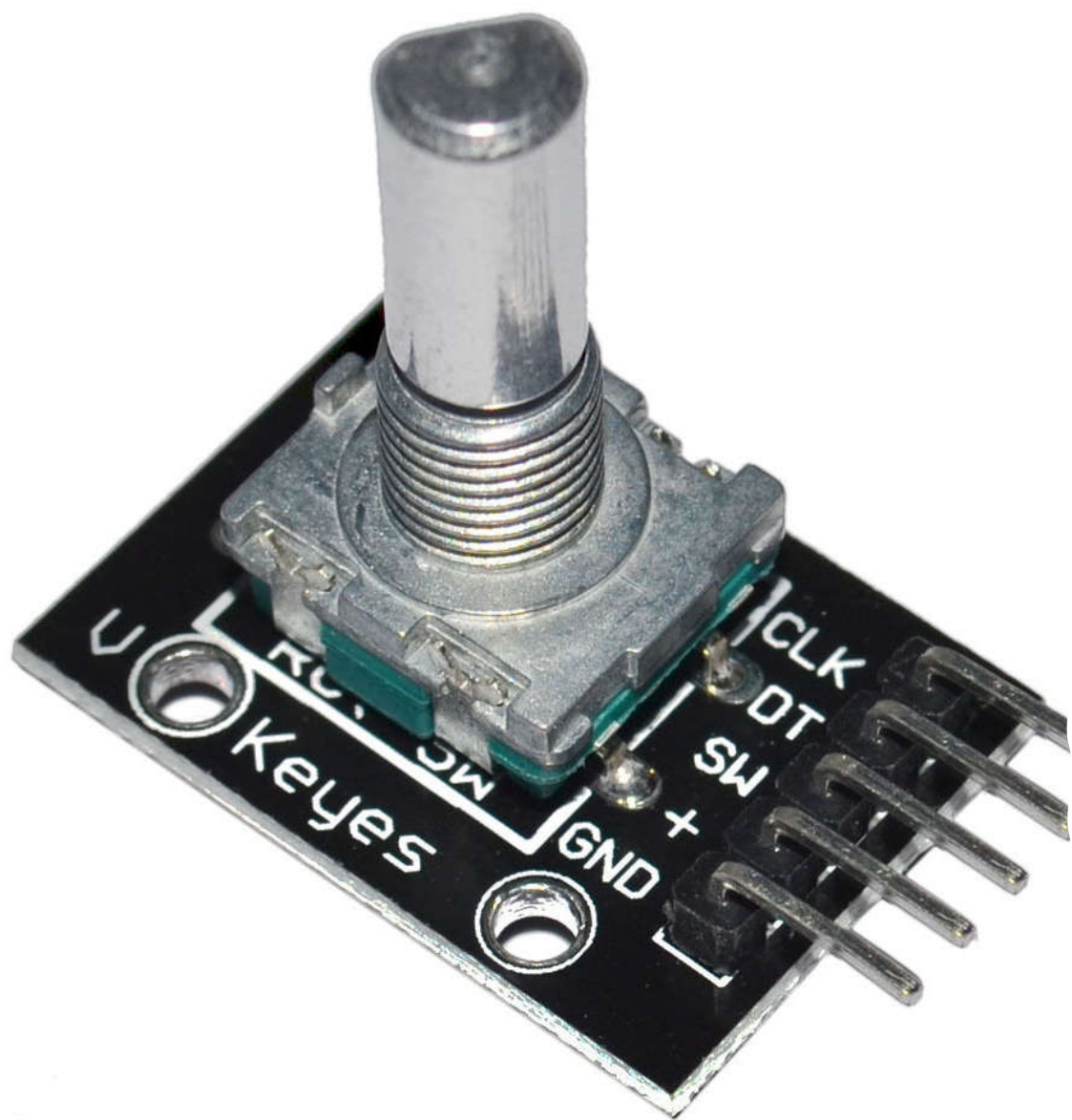




DTH-11 Módulo
sensor de
Humedad y
Temperatura

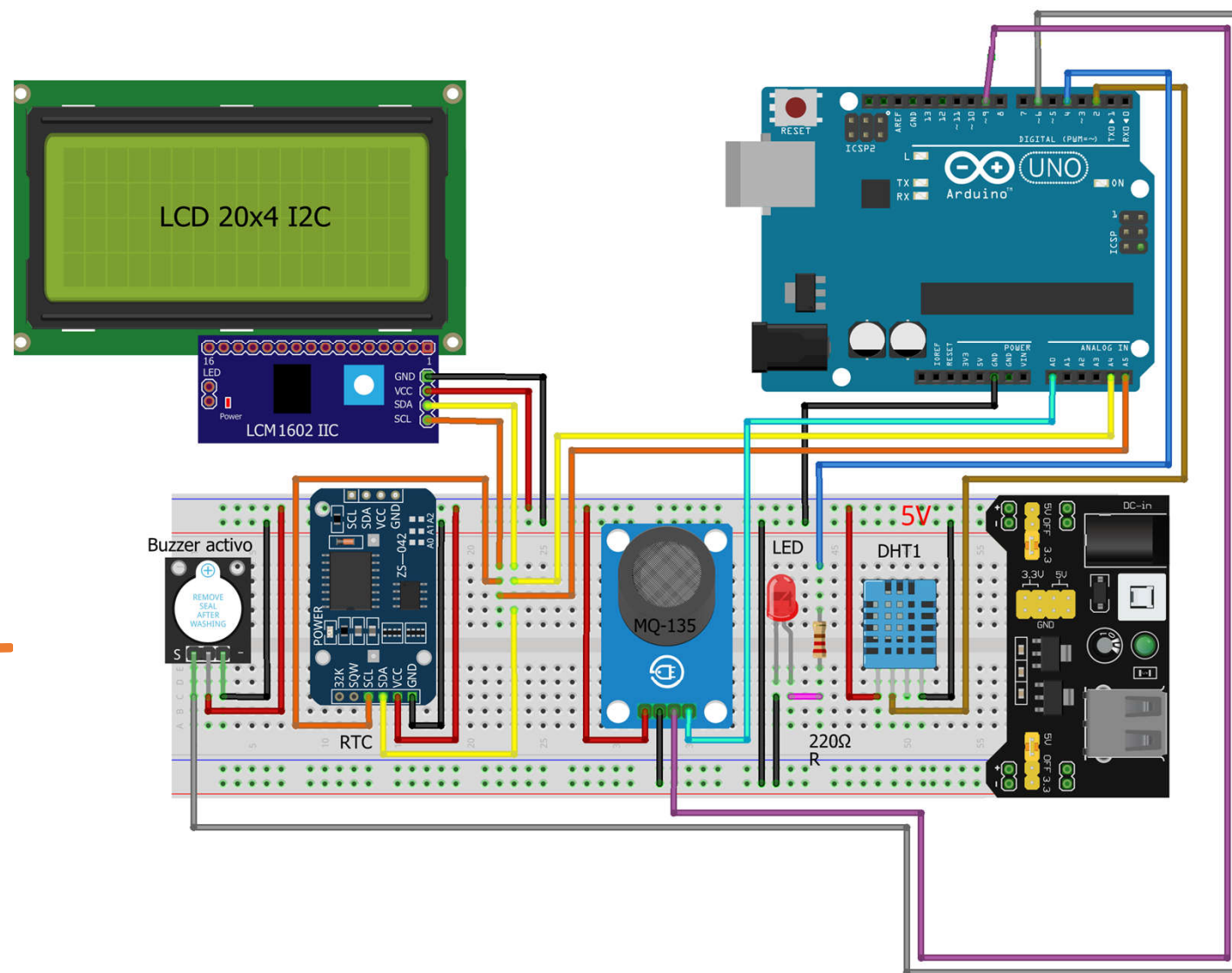
DS3231 – Reloj de Tiempo Real



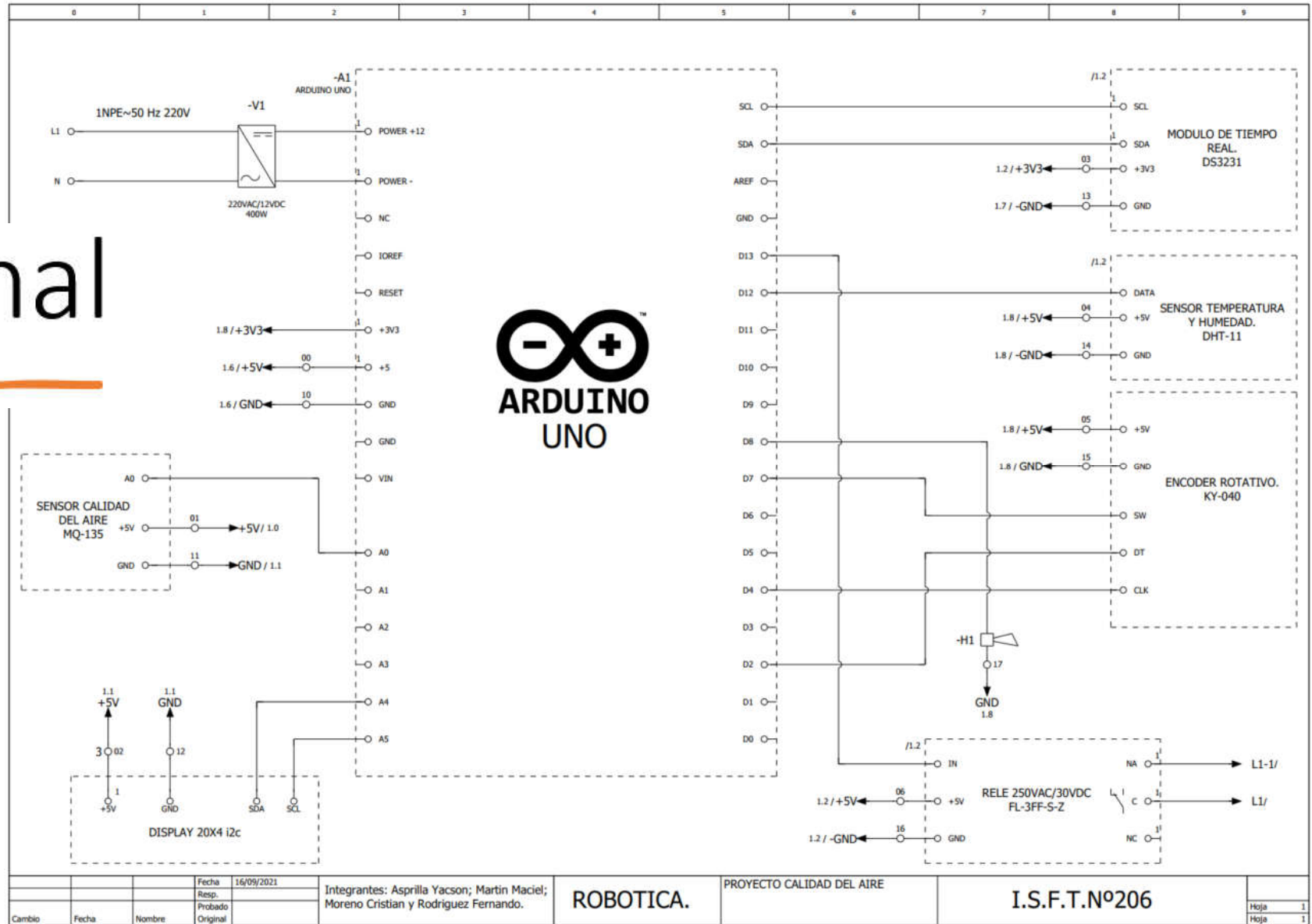


KY-040
Encoder
Rotativo

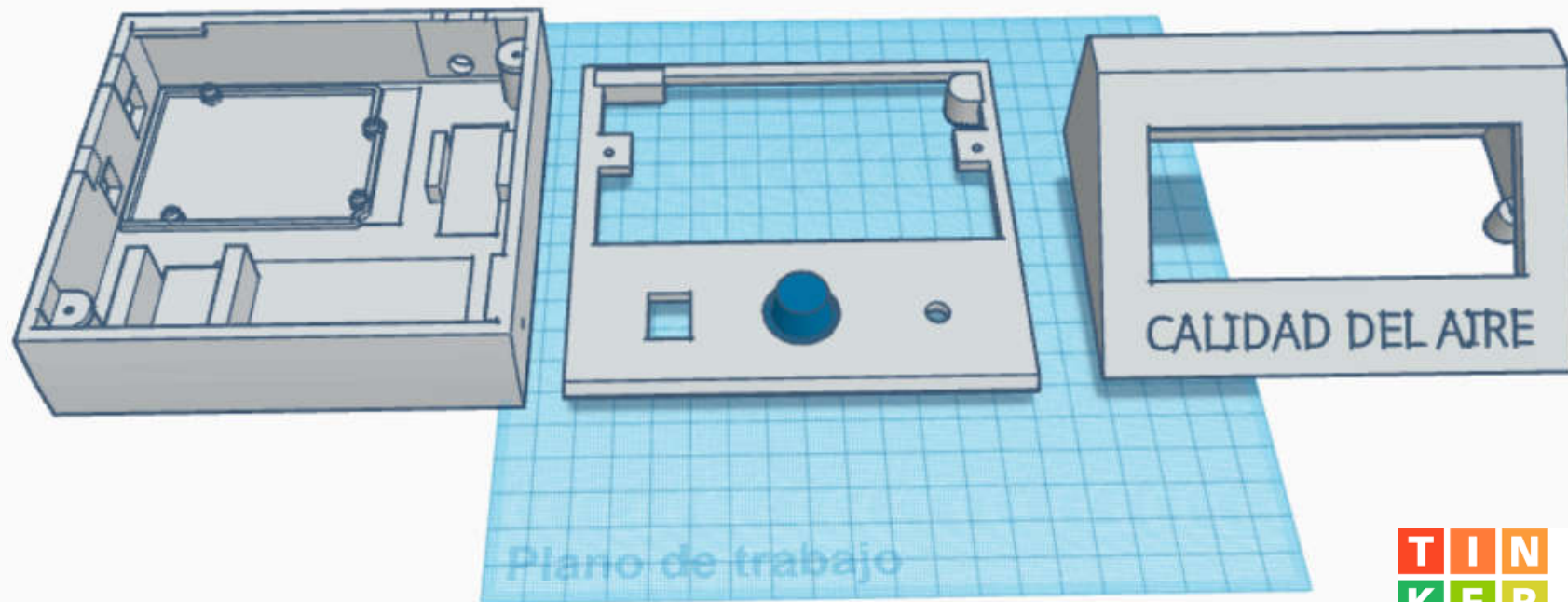
Esquema



Funcional



Modelado de Gabinete



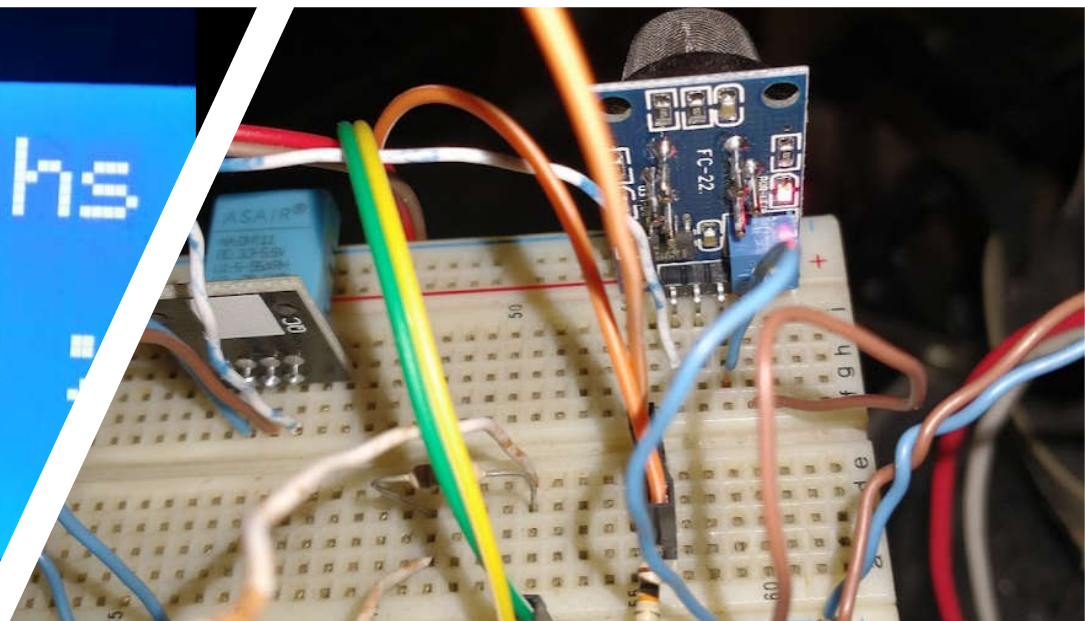
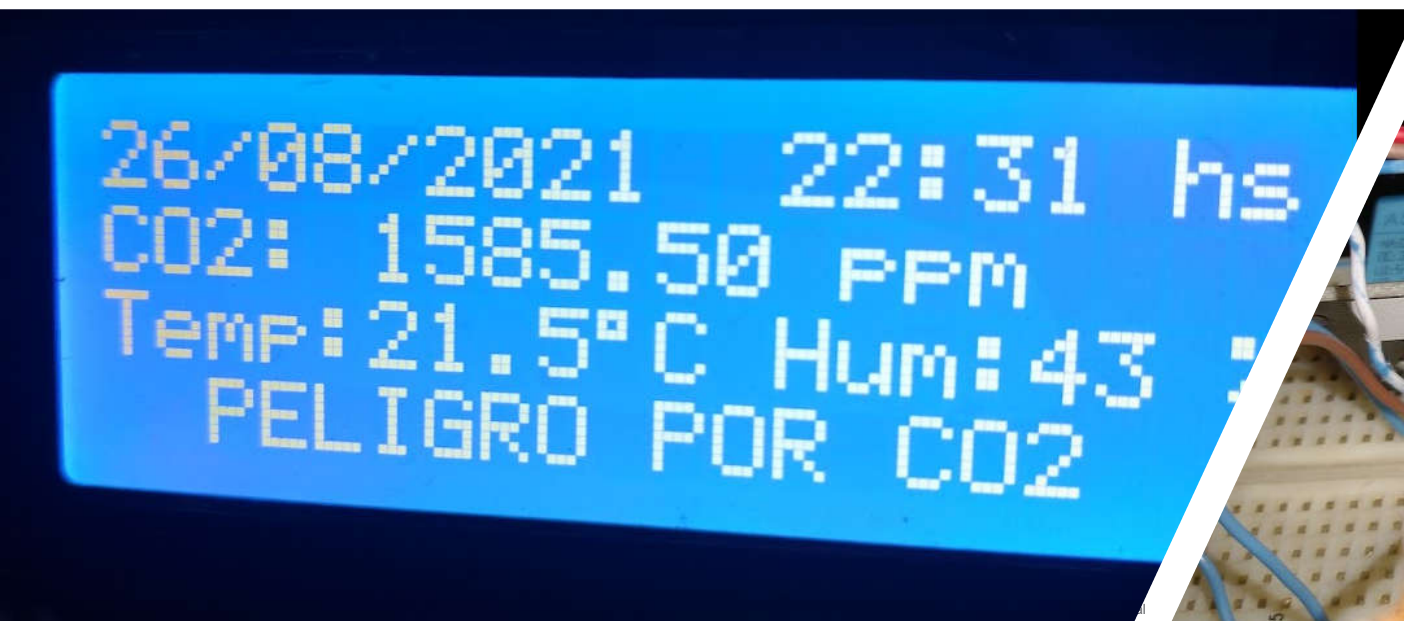
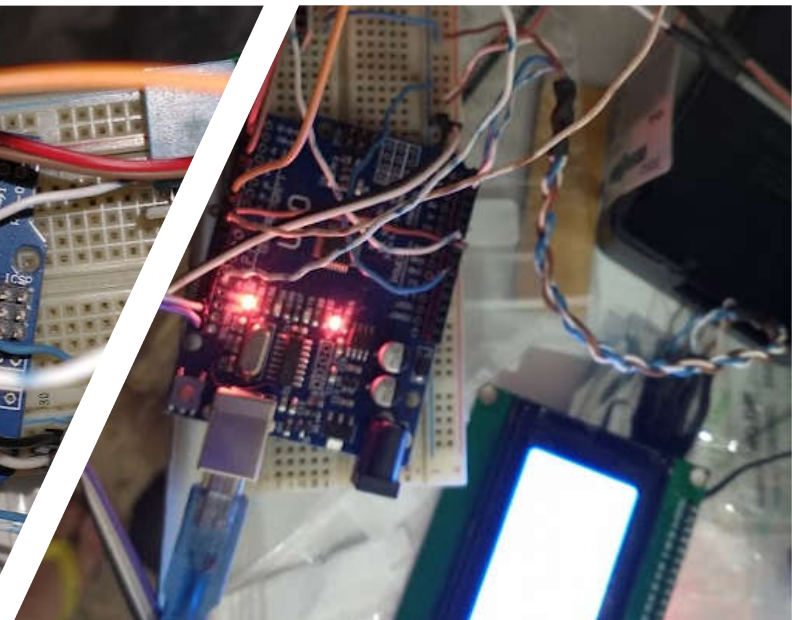
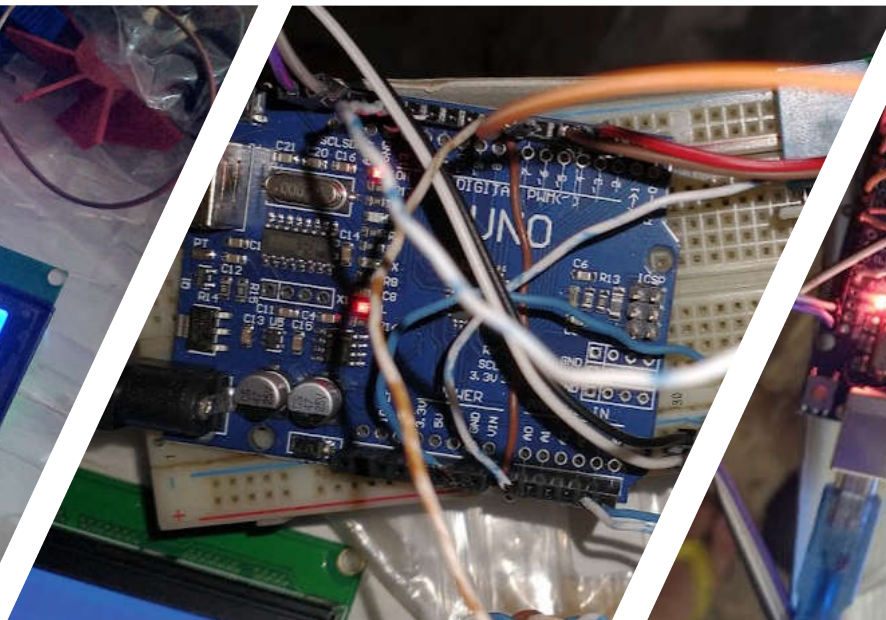
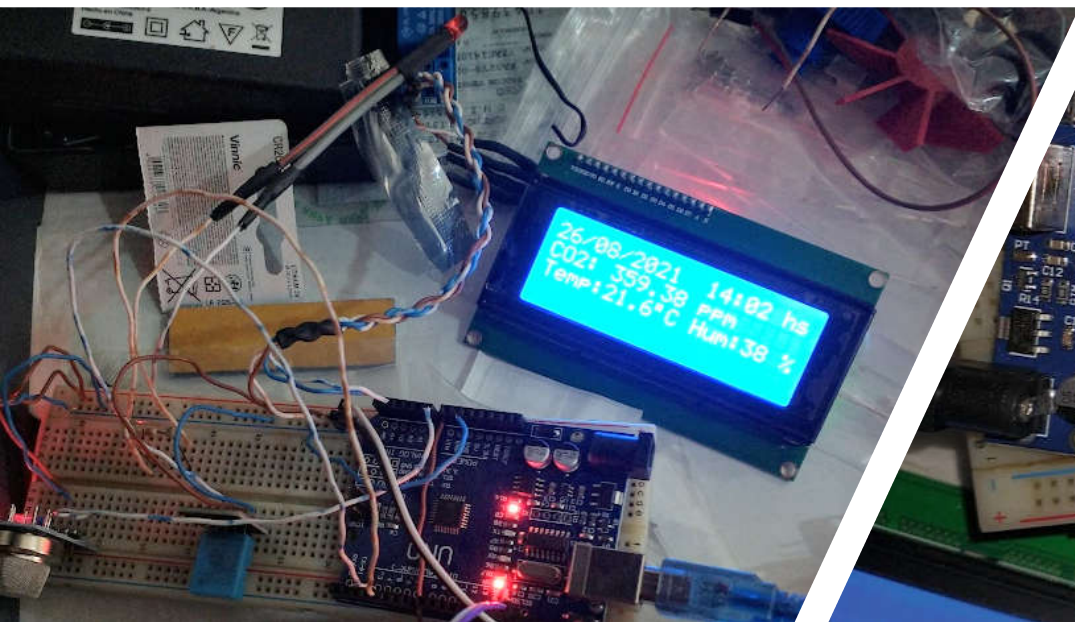
Presupuesto

Cant.	Unidad	Descripción	Monto
1	Un.	Sensor De Temperatura Y Humedad Relativa Dht11 Arduino	\$ 300,00
1	Un.	Display Lcd 2004 Backlight Azul 20x4 + Serie I2c Arduino	\$ 912,00
1	Un.	Modulo Rtc Ds3231 Alta Precision Eeprom 24c32 Arduino C/pila	\$ 475,00
1	Un.	Fuente Switching 12v 1,25a Tira Led Camara Arduino	\$ 550,00
1	Un.	Modulo Sensor De Gas Mq135 Mq-135 Arduino	\$ 490,00
1	Un.	Cables Macho Hembra 40 X 20cm Mh Dupont Arduino Protoboard	\$ 270,00
1	Un.	Modulo Rele Relay De 1 Canal 5v 10a Arduino Pic Ptec	\$ 250,00
1	Un.	Arduino Uno Ch340 C/cable Usb Compatible	\$ 1.000,00
1	Un.	Gabinete	\$ 250,00

TOTAL \$ 4.497,00

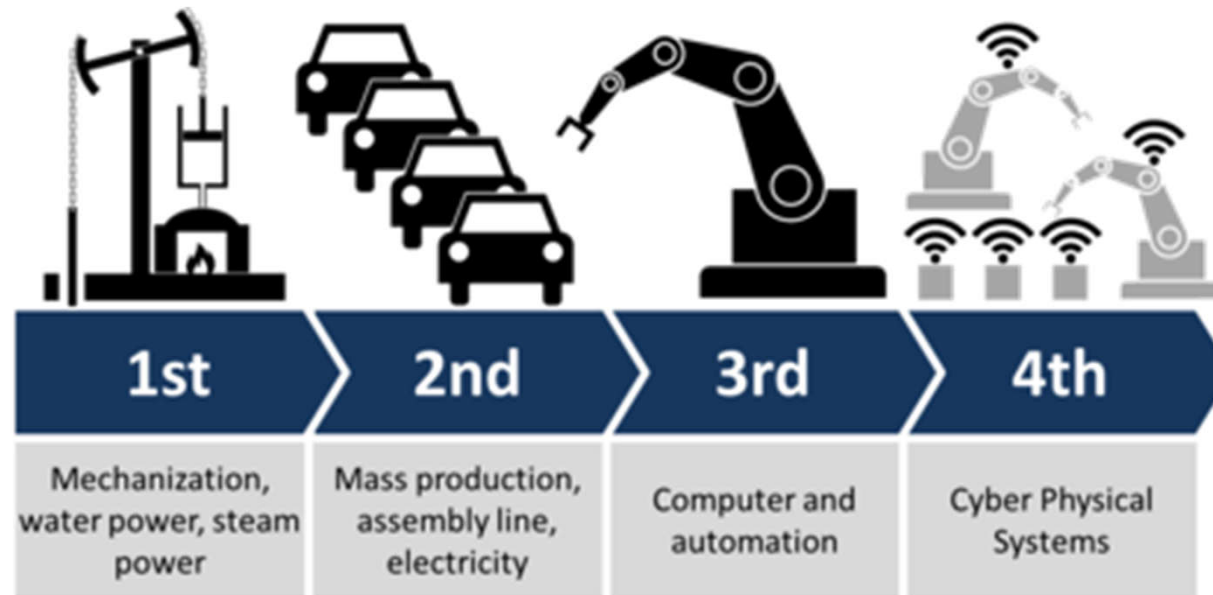
Fotos





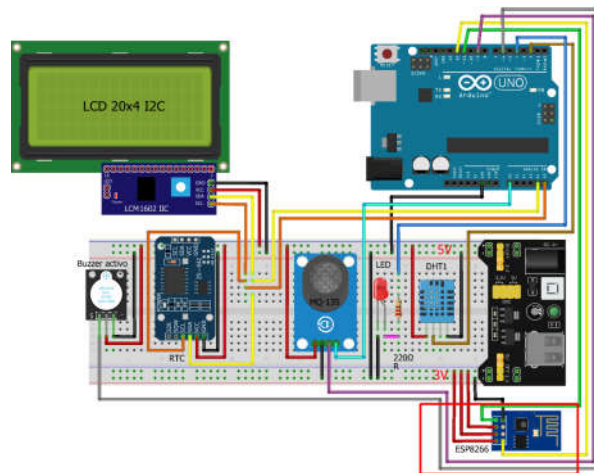
Futuro – Industria 4.0

- Mecanización, energía hidráulica, energía eléctrica
- Producción en masa, línea de montaje, electricidad
- Informática y automatización
- Sistema ciberfísico.



Este concepto de Industria 4.0 que aquí se presenta no es una realidad ya consolidada y experimentada, sino un nuevo hito en el desarrollo industrial que podría marcar importantes cambios sociales en los próximos años, haciendo un uso intensivo de Internet y de las tecnologías punta, con el fin primordial de desarrollar plantas industriales y generadores de energía más inteligentes y más respetuosos con el medio ambiente, y con cadenas de producción mucho mejor comunicadas entre sí y con los mercados de oferta y demanda.

Send Data from NodeMCU to Google Sheets



Modulo wi-fi



esp8266



3º año 2021

**MEDIDOR DE CALIDAD DEL AIRE
PROYECTO**

PROFESOR: ARIEL TARAMAZZO

**ALUMNOS: Asprilla Yacson, Cristian Moreno, Martin Maciel,
Nahuel Rodriguez.**

FECHA DE ENTREGA:

LUGAR Y FECHA: GREGORIO DE LAFERRERE, 26/08/2021

Contenido

Introducción:	3
Proyecto:	3
Arduino UNO.	3
Vista superior.....	4
Pinout	5
Sensor de calidad del aire MQ-135	5
Características del sensor de gases MQ-135	5
Sensor Temperatura y Humedad DHT11	6
Especificaciones técnicas	7
Display 20x4 i2c	7
Módulo de tiempo real 3231	8
Encoder Rotativo KY-040	9

Introducción:

El proyecto surge a través de la necesidad de conocer cual es la calidad de aire que respiramos ya que con la actual pandemia esto paso a ser un punto muy importante.

Por lo que conocemos la mayor tasa de contagios surgen en ambientes muy poco ventilados. El aire viciado representa la situación en la que el aire que respiramos en interiores ha sido contaminado y no se ha extraído ni renovado. Al pasar ese aire suficiente tiempo en el ambiente sin moverse hacia los exteriores e instalarse dentro de nuestros hogares, puede llegar a ser perjudicial para la salud.

Podemos conocer que tan contaminado está el aire a través de la cantidad de dióxido de carbono CO₂. El dióxido de carbono es uno de los contaminantes más habituales y que más afectan en edificios a la salud humana. Controlar su presencia en edificios es una importantísima ayuda en la salud y confort en los hogares.

Los diferentes estudios de referencia avisan que los niveles recomendados de dióxido de carbono (CO₂) máximos recomendados dentro de hogares, estancias y otros edificios están entre las 400 ppm y las 800 ppm.

Considerando los puntos anteriores entendemos que disponemos de todos los medios necesarios para elaborar un proyecto que resuelva la problemática planteada y a su vez nos aporte conocimientos para formarnos como técnicos en automatización y control.

Proyecto:

Para llevar a cabo esta proyecto utilizaremos a una placa Arduino modelo "UNO" ya que es el más utilizado a nivel mundial y el de mayor disponibilidad global. Este tiene un buen balance entre entradas y salidas y un precio módico que lo hace accesible a cualquiera. Arduino siendo cada vez mas utilizado en entidades educativas para introducir a sus estudiantes al mundo de la programación y la robótica.

Nuestro producto final será capaz de medir la cantidad de CO₂ en partes por millón, la temperatura y la humedad. Mostrará en tiempo real cada uno de los sensores a través de un display LCD de 20x4. Se podrán programar alarmas con valores editables por el usuario y en base a estas activar 4 salidas a relé opto acopladas para manejar cargas de gran potencia. A su vez contará con un reloj en tiempo real para guardar los eventos ocurridos.

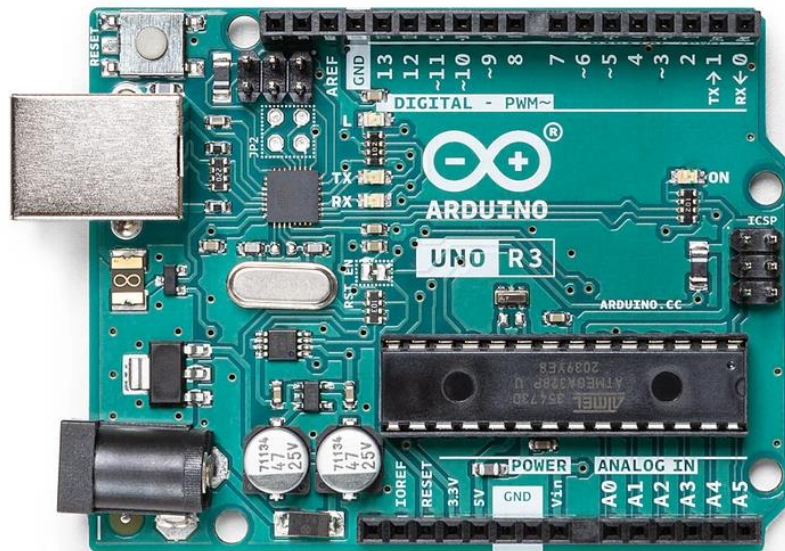
Arduino UNO.

El Arduino es una placa basada en un microcontrolador ATMEL. Los microcontroladores son circuitos integrados en los que se pueden grabar instrucciones, las cuales las escribes con el lenguaje de programación que puedes utilizar en el entorno Arduino IDE. Estas instrucciones permiten crear programas que interactúan con los circuitos de la placa.

El microcontrolador de Arduino posee lo que se llama una interfaz de entrada, que es una conexión en la que podemos conectar en la placa diferentes tipos de periféricos. La información de estos periféricos que conectes se trasladará al microcontrolador, el cual se encargará de procesar los datos que le lleguen a través de ellos.

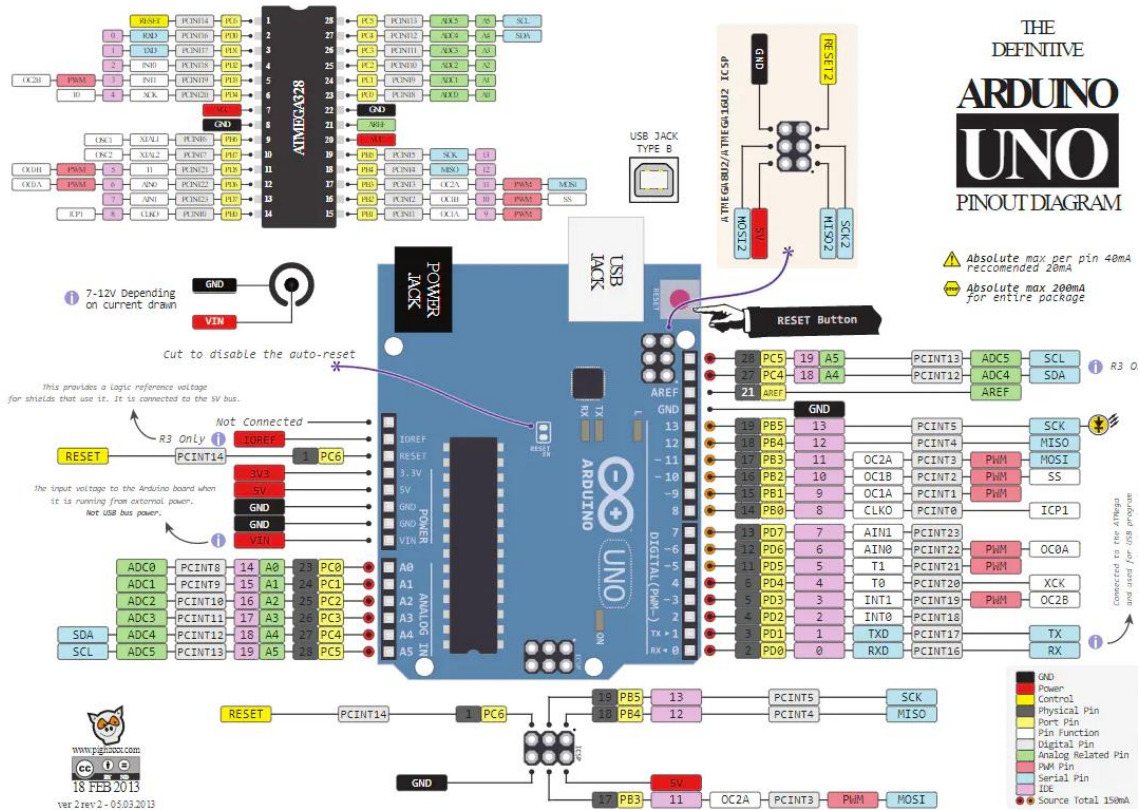
El tipo de periféricos que puedas utilizar para enviar datos al microcontrolador depende en gran medida de qué uso le estés pensando dar. Pueden ser cámaras para obtener imágenes, teclados para introducir datos, o diferentes tipos de sensores.

También cuenta con una interfaz de salida, que es la que se encarga de llevar la información que se ha procesado en el Arduino a otros periféricos. Estos periféricos pueden ser pantallas o altavoces en los que reproducir los datos procesados, pero también pueden ser otras placas o controladores.



Vista superior

La placa tiene 14 pines digitales, 6 pines analógicos programables con el Arduino IDE (Entorno de desarrollo integrado) a través de un cable USB. Puede ser alimentado por el cable USB o por una batería externa de 9 voltios, aunque acepta voltajes entre 7 y 20 voltios.

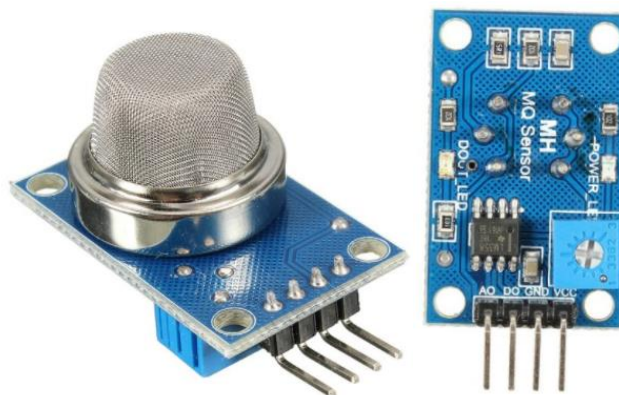


Pinout

Sensor de calidad del aire MQ-135

Utilizaremos el sensor de la familia MQ en este caso el MQ-135.

El Sensor de Gas MQ-135 es un módulo capaz de detectar varios tipos de gases tóxicos como amoníaco, dióxido de carbono, benceno, óxido nítrico, así como humo o alcohol.



Características del sensor de gases MQ-135

Se trata de un sensor electroquímico. Tiene la capacidad de variar su resistencia en función de las partes por millón de partículas de gas en el ambiente. Dispone de un calefactor interno que tiene la finalidad de mantener estable la temperatura y la humedad dentro del encapsulado, ya que estas afectan a la sensibilidad del sensor. Por este motivo, se recomienda usar el sensor en unas condiciones ambientales óptimas y tenerlo encendido unos minutos antes de usarlo. De esta forma, conseguiremos que el calefactor interno haga su trabajo.

También es recomendable dejar encendido el sensor unas 24 horas antes de su primer uso. A este proceso se le conoce como "burning" o "quemado". Sirve para que el calefactor interno queme cualquier residuo o partícula que se haya podido depositar dentro del sensor durante su proceso de fabricación.

Dispone de una micro malla exterior, del tamaño de unas pocas micras. Sirve para impedir que partículas externas entren dentro del encapsulado del sensor. Debemos tener cuidado cuando hagamos experimentos. No debemos echar directamente al sensor líquidos o gases de mecheros que puedan despedir "gotas" dentro del mismo. Corremos el riesgo de dañarlo para siempre.

Las principales características del sensor de gases MQ-135 según su datasheet son:

- Gran variedad de gases detectables
- Estable y larga vida útil
- Rápida respuesta y alta sensibilidad
- Sencillez de uso

Model			MQ135
Sensor Type			Semiconductor
Standard Encapsulation			Bakelite, Metal cap
Target Gas			ammonia gas, sulfide, benzene series steam
Detection range			10~1000ppm(ammonia gas, toluene, hydrogen, smoke)
Standard Circuit Conditions	Loop Voltage	V_c	$\leq 24V$ DC
	Heater Voltage	V_H	$5.0V \pm 0.1V$ AC or DC
	Load Resistance	R_L	Adjustable
Sensor character under standard test conditions	Heater Resistance	R_H	$29\Omega \pm 3\Omega$ (room tem.)
	Heater consumption	P_H	$\leq 950mW$
	Sensitivity	S	$R_s(\text{in air})/R_s(\text{in } 400ppm H_2) \geq 5$
	Output Voltage	V_s	$2.0V \sim 4.0V$ (in 400ppm H_2)
Standard test conditions	Concentration Slope	α	$\leq 0.6(R_{400ppm}/R_{100ppm} H_2)$
	Tem. Humidity		$20^\circ C \pm 2^\circ C$; $55\% \pm 5\% RH$
	Standard test circuit		$V_c: 5.0V \pm 0.1V$; $V_H: 5.0V \pm 0.1V$
	Preheat time		Over 48 hours

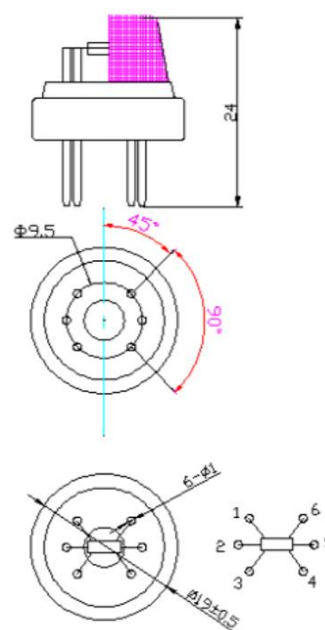


Fig1. Sensor Structure
Unit: mm

Sensor Temperatura y Humedad DHT11

El sensor de temperatura y humedad DHT11 presenta un complejo de sensores de temperatura y humedad con una salida de señal digital calibrada. Al utilizar la técnica exclusiva de adquisición de señal digital y la tecnología de detección de temperatura y humedad, garantiza una alta fiabilidad y una excelente estabilidad a largo plazo. Este sensor incluye un componente de medición de humedad de tipo resistivo y un componente de medición de

temperatura NTC, y se conecta a un microcontrolador de 8 bits de alto rendimiento, que ofrece excelente calidad, respuesta rápida, capacidad antiinterferente y rentabilidad.

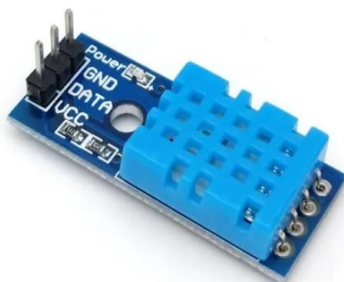
El voltaje de alimentación debe estar entre 3.3V y 5V DC. Cuando se suministra energía al sensor, no le envíe ninguna instrucción dentro de un segundo para pasar el estado inestable. Se puede agregar un condensador de 100nF entre VDD y GND para el filtrado de ondas.

Cada sensor de este modelo tiene compensación de temperatura y está calibrado en una cámara de calibración precisa y el coeficiente de calibración se guarda en el tipo de programa en la memoria OTP, cuando el sensor está detectando, citará el coeficiente de la memoria.

El tamaño pequeño, el bajo consumo y la larga distancia de transmisión (20 m) permiten que DHT11 sea adecuado en todo tipo de aplicaciones difíciles.

Especificaciones técnicas

Item	Descripción
Modelo	DHT11
Fuente de alimentación	3.3-5V DC
Señal de salida	Señal digital a través de bus único
Elemento sensor	Condensador de polímero
Rango de funcionamiento	humedad 20-90% HR; temperatura 0 ~ 50 Celsius
Precisión	humedad + -2% HR (Máx. +- 5% HR); temperatura < +- 2C
Resolución o sensibilidad	humedad 1% HR; temperatura 1 Celsius
Repetibilidad	humedad 1% HR; temperatura 1 Celsius
Período de detección Promedio	6s
Intercambiabilidad	totalmente intercambiable
Dimensiones	tamaño pequeño 12 * 15 * 5.5 mm



Display 20x4 i2c

Display LCD de 4 líneas de 20 caracteres con comunicaciones I2C y serie y retro iluminación controlable por software. Gracias a su iluminación posterior, los mensajes se ven claramente incluso en total oscuridad. Permite la conexión de un teclado matricial a través del módulo de comunicaciones.

Principales características:

- Pantalla – 20 caracteres por línea, 4 líneas
- Alimentación – 5v
- Consumo – 125mA con retroiluminación de pantalla encendida

- Comunicaciones – I2C o serie 9600bps
- Teclado – Escaneo automático de teclado matricial 3x4
- Caracteres personalizados – hasta 8 caracteres personalizados fácilmente definidos

Retroiluminación de pantalla apagada por defecto, puede ser encendida por software.

Disponible en dos colores de pantalla: Texto negro sobre fondo verde (clásico) y texto blanco sobre fondo azul. El consumo de la iluminación de pantalla es de unos 55mA para la Azul/Blanco y 125mA para la Verde/Negro, por otra parte, la Azul/Blanco requiere la luz encendida para poder leerla y la Verde/Negro se lee claramente sin la luz de pantalla.



Módulo de tiempo real 3231

El DS3231 es un reloj I2C en tiempo real (RTC) extremadamente preciso y de bajo costo con un oscilador de cristal (TCXO) y cristal integrados compensados por temperatura.

El dispositivo incorpora una entrada de batería y mantiene la hora exacta cuando se interrumpe la alimentación principal del dispositivo. La integración del resonador de cristal mejora la precisión a largo plazo del dispositivo y reduce el recuento de piezas en una línea de fabricación.

El DS3231 está disponible en rangos de temperatura comercial e industrial, y se ofrece en un paquete SO de 16 pines, 300 mil.

El RTC mantiene información sobre segundos, minutos, horas, día, fecha, mes y año. La fecha al final del mes se ajusta automáticamente a los meses con menos de 31 días, incluidas las correcciones del año bisiesto. El reloj funciona en formato de 24 horas o de 12 horas con un indicador AM / PM. Dos horas del día programables.

Se proporcionan alarmas y una salida de onda cuadrada programable. La dirección y los datos se transfieren en serie a través de un bus bidireccional I2C.

Una referencia de voltaje de precisión compensada y un circuito comparador monitorean el estado de VCC para detectar fallas de energía, para proporcionar una salida de reinicio y para cambiar automáticamente a la fuente de respaldo cuando sea necesario.

CARACTERISTICAS:

El RTC de alta precisión administra completamente todas las funciones de hora normal.

- El reloj de tiempo real cuenta los segundos, los minutos, las horas, la fecha del mes, el mes, el día de la semana y el año, con una compensación de año bisiesto válida hasta 2100
- Precisión ± 2 ppm de 0°C a $+40^{\circ}\text{C}$
- Precisión ± 3.5 ppm de -40°C a $+85^{\circ}\text{C}$
- Salida del sensor de temperatura digital: precisión de $\pm 3^{\circ}\text{C}$
- Registrarse para el ajuste de envejecimiento
- RST Output / Pushbutton Reset Debounce Input
- Dos alarmas de hora del día
- Señal de salida de onda cuadrada programable
- La interfaz en serie simple se conecta a la mayoría de los microcontroladores
- Interfaz I2C rápida (400kHz)
- Entrada de batería de respaldo para el cronometraje continuo
- La operación de baja potencia prolonga el tiempo de ejecución de la batería de respaldo
- Operación 3.3V
- Rangos de temperatura de funcionamiento: comercial (0°C a $+70^{\circ}\text{C}$) e industrial (-40°C a $+85^{\circ}\text{C}$)
- Underwriters Laboratories® (UL) Reconocido



Encoder Rotativo KY-040

El encoder rotatorio Keyes KY-040 es un dispositivo de entrada giratorio (como en la perilla) que proporciona una indicación de cuánto la perilla ha sido girada y en qué dirección está girando.

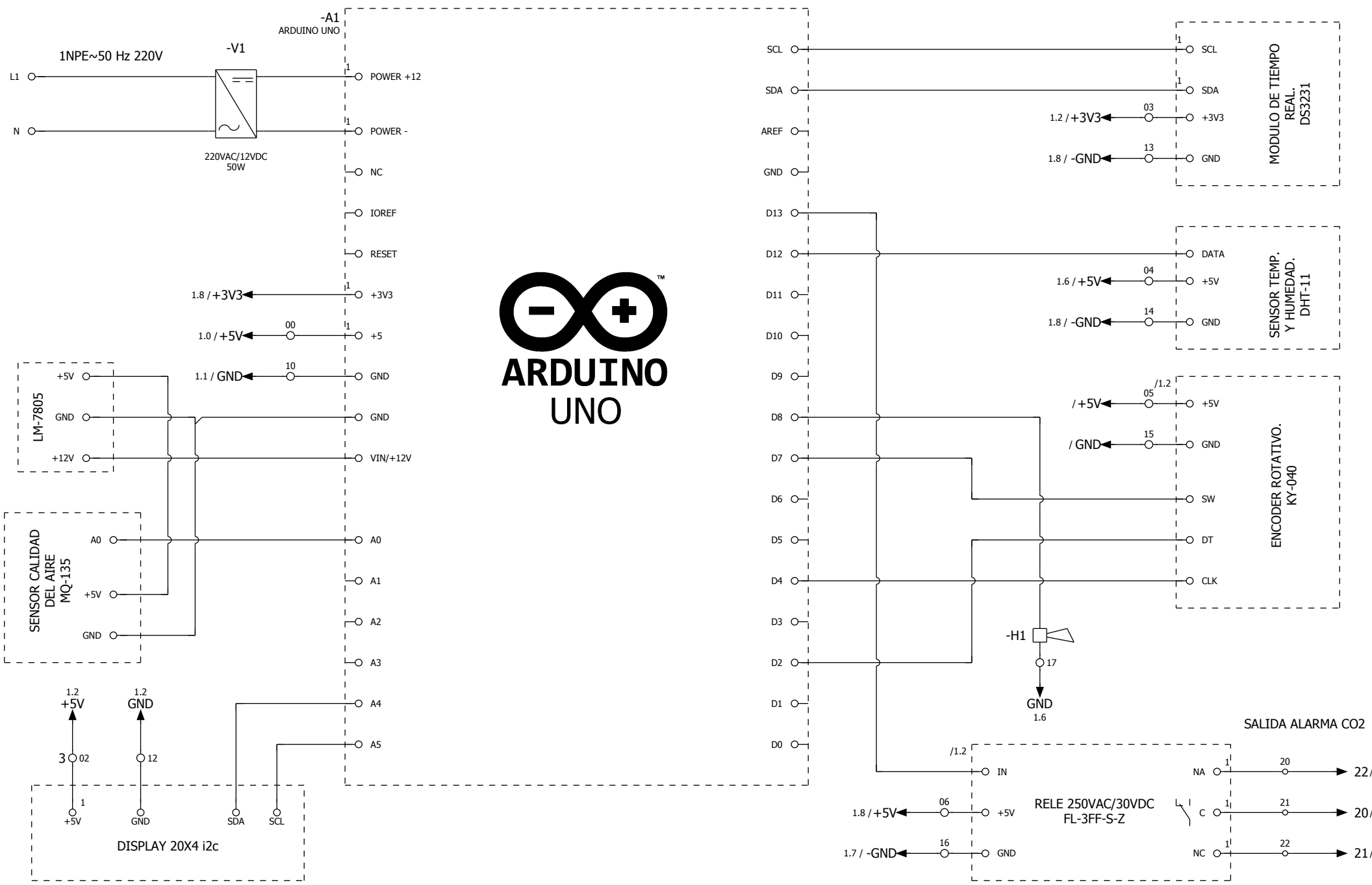
Este encoder incremental genera pulsos que codifican la rotación, ya sea en ciclo horario o anti-horario, con 20 pulsos por ciclo, a diferencia de un potenciómetro, este encoder no posee topes mecánicos, por lo que puede girar libremente, el encoder puede ser pulsado, de modo que se puede usar para seleccionar algo.

Muy útil para proyectos de audio, HMI, búsqueda y selección de archivos.

Especificaciones técnicas

- Voltaje de Operación: 5V
- Pulsos por revolución: 20
- Dimensiones: 2cm*3cm*3cm






```

1  //LIBRERIAS A INCLUIR
2
3  #include <LiquidCrystal_I2C.h> // para la pantalla LCD
4  #include <Wire.h> // incluye libreria para interfaz I2C
5  #include <RTCLib.h> // incluye libreria para el manejo del modulo RTC
6  #include "DHT.h" // para el sensor de temperatura y humedad
7
8  //DEFINICIONES
9
10 #define TIPO_DHT DHT11 // tipo de sensor de temperatura/humedad que vamos a usar
11 #define PIN_DHT11 12 // pin para el sensor de temperatura/humedad
12 #define PIN_ZUMBADOR 8 // pin para el zumbador de alarma
13 #define PIN_MQ_ANALOGICO PIN_A0 // pin analógico para el sensor MQ-135 que utilizaremos
14 para la medición
15 #define PIN_ENC_DT 2 //ENCODER pin digital 2 (DT en modulo)
16 #define PIN_ENC_CLK 4 //ENCODER pin digital 4 (CLK en modulo)
17 #define PULSADOR 7 //PULSADOR MODULO ENCODER
18 #define RELE 11 //RELE pin 11 Salida
19
20
21 //ENCODER
22 int ANTERIOR = 10; // almacena valor anterior de la variable POSICION
23 volatile int POSICION = 10; // variable GLOBAL POSICION con valor inicial de 10 y definida como
24 global al ser usada en loop e ISR (encoder)
25 int A = 2; //variable A a pin digital 2 (DT en modulo)
26 int B = 4; //variable B a pin digital 4 (CLK en modulo)
27
28 //CONSTANTES
29
30 const int RETARDO = 10; // Retardo bucle
31 const int co2_pre_alarma = 600; //NORMAL 500
32 const int co2_alarma = 800;

```

```

33  const int co2_evacuar = 1200;
34
35  /*
36   Normativa vigente sobre CO2
37   La actual normativa vigente y diversos documentos como la NTP 742 (Notas Técnicas de
38   Prevención) sobre "Ventilación General de los Edificios" recoge la categorización de la calidad
39   del aire interior en función del nivel de dióxido de carbono permitido en un recinto, método
40   que tradicionalmente se ha venido utilizando para el establecimiento de los estándares de
41   ventilación cuando la contaminación es debida casi exclusivamente a los ocupantes y su
42   actividad y en recintos donde no está permitido fumar.
43
44   Categorías del aire según los niveles de CO2
45   Está categorización identifica el aire de mala calidad a partir de 1200 ppm (partes por millón)
46   considerados como el nivel máximo recomendado de CO2 en ambientes interiores:
47
48   Hasta 350 ppm: calidad de aire interior alta
49   Entre 350 y 500 ppm: calidad de aire interior buena
50   Entre 500 y 800 ppm: calidad de aire interior moderada  // PRE ALARMA
51   Entre 800 y 1200 ppm: calidad de aire interior baja  // ALARMA
52   Nivel superior a 1200 ppm: calidad de aire interior mala // EVACUAR
53
54   Fuente:https://www.solerpalau.com/es-es/blog/efectos-co2/
55
56   */
57
58  //INICIALIZACIONES
59
60  //Modulo de tiempo real
61  RTC_DS3231 rtc;  // crea objeto del tipo RTC_DS3231
62
63  //Modulo LCD
64  LiquidCrystal_I2C lcd(0x26, 20, 4);
65

```

```

66 // Modulo Sensor temperatura - humedad
67 DHT dht(PIN_DHT11, TIPO_DHT); // creamos el objeto que modela el sensor de temperatura y
68 humedad, asociado
69         // al pin correspondiente y del tipo definido.
70
71
72
73 //SETUP >>> SOLO SE EJECUTA UNA VEZ
74
75 void setup () {
76
77     attachInterrupt(digitalPinToInterrupt(A), encoder, LOW); // interrupcion sobre pin A con
78         // funcion ISR encoder y modo LOW
79
80     Serial.begin(9600); // inicializa comunicacion serie a 9600 bps
81
82     if (! rtc.begin()) { // si falla la inicializacion del modulo
83         Serial.println("Modulo RTC no encontrado !"); // muestra mensaje de error
84         while (1); // bucle infinito que detiene ejecucion del programa
85     }
86
87
88 // rtc.adjust(DateTime(__DATE__, __TIME__)); // funcion que permite establecer fecha y
89 horario por primera vez.
90         // al momento de la compilacion. Comentar esta linea
91
92     lcd.init(); // Inicializamos el LCD
93     lcd.clear(); // limpia pantalla
94     lcd.backlight(); //Activamos la luz de fondo
95
96     dht.begin(); // inicializamos el sensor de temperatura y humedad
97

```

```

98  // SALIDAS
99  pinMode(PIN_ZUMBADOR, OUTPUT);
100  digitalWrite(PIN_ZUMBADOR, LOW);
101
102  pinMode(RELE, OUTPUT);
103  digitalWrite(RELE, HIGH);
104  // pinMode(PIN_LED, OUTPUT);
105
106  // ENTRADAS
107  pinMode(PIN_ENC_DT, INPUT); // ENCODER
108  pinMode(PIN_ENC_CLK, INPUT); // ENCODER
109  pinMode(PULSADOR, INPUT_PULLUP); //PULSADOR ENCODER
110
111
112
113  } //SETUP >>>FIN
114
115
116
117
118  //INICIO BUCLE INFINITO
119
120  void loop () {
121
122      DateTime fecha = rtc.now(); // funcion que devuelve fecha y horario en formato
123          // DateTime y asigna a variable fecha
124      int entrada=0; //variable de control
125
126
127
128

```

```

129
130
131  /////BUCLE DE INTERVALO PERIODICO//////////
132
133  static unsigned long ultimaLlamado = 0; // variable static con ultimo valor tiempo de
134  interrupcion
135
136  unsigned long tiempoTotal = millis(); // variable almacena valor de func. millis
137
138  double CO2ppm; //Variable donde Almacenamos el valor procesado del MQ-135
139
140  if (tiempoTotal - ultimaLlamado > 3000) //refresco cada 1 segundo = 1000
141  {
142      ultimaLlamado = tiempoTotal; //reset llamado
143
144      //MEDICION EN LOS SENSORES
145      int medidaAnalogica = analogRead(PIN_MQ_ANALOGICO); //sensor MQ-135
146      float temperatura = dht.readTemperature(); //sensor dht 11
147      float humedad = dht.readHumidity(); //sensor dht 11
148
149
150
151
152
153      //ESCALAMOS LA MEDICIÓN DE MQ-135
154      CO2ppm = medidaAnalogica * 10.57;
155
156      char form[20]; //ALMACENAMOS EN ESTA VARIABLE FECHA Y HORA COMO CADENA
157      sprintf(form, "%02d/%02d/%04d %02d:%02d hs", fecha.day(), fecha.month(),
158      fecha.year(), fecha.hour(), fecha.minute()); //CONVERTIMOS EL OBJETO FECHA EN CADENA
159
160

```

```

161      //IMPRESION EN DISPLAY
162
163
164      entrada =1;
165      lcd.setCursor(0, 0);
166      lcd.print(form);
167
168      lcd.setCursor(0, 1);
169      lcd.print("CO2: ");
170      for (int n = 5 ; n < 19; n++)
171      {
172          lcd.print(" ");
173      }
174
175      lcd.setCursor(5, 1);
176      lcd.print(CO2ppm);
177      lcd.print(" ppm");
178
179      lcd.setCursor(0, 2);
180      lcd.print("Temp:");
181      lcd.print(temperatura,1); // Con 1 decimal
182      lcd.print((char)223);
183      lcd.print("C");
184
185      lcd.setCursor(12, 2);
186      lcd.print("Hum:");
187      lcd.print(humedad,0); // Con 1 decimal
188      lcd.print(" %");
189
190
191

```

```

192
193
194
195
196
197     /////BORRAR
198     Serial.println(analogRead(PIN_MQ_ANALOGICO));
199     Serial.println(CO2ppm);
200
201     /////
202
203     //ALARMAS
204
205     if (CO2ppm >co2_evacuar){
206         lcd.setCursor(0, 3);
207         lcd.print("          ");
208         lcd.setCursor(2, 3);
209         lcd.print("PELIGRO POR CO2");
210         digitalWrite(RELE, LOW);
211
212     } else if (CO2ppm >co2_alarma){
213         lcd.setCursor(0, 3);
214         lcd.print("          ");
215         lcd.setCursor(5, 3);
216         lcd.print("ALARMA CO2");
217
218     } else if (CO2ppm >co2_pre_alarma){
219         lcd.setCursor(0, 3);
220         lcd.print("          ");
221         lcd.setCursor(3, 3);
222         lcd.print("PRE ALARMA CO2");

```



```

223
224     } else
225     {
226         lcd.setCursor(0,3 );
227         lcd.print("          ");
228         digitalWrite(RELE, HIGH);
229     }
230
231
232 }
233
234
235 //ALARMAS SONORA
236
237     if (CO2ppm >co2_evacuar){
238
239         digitalWrite(PIN_ZUMBADOR, HIGH);
240         delay(100);
241         digitalWrite(PIN_ZUMBADOR, LOW);
242         delay(100);
243
244
245     } else if (CO2ppm >co2_alarma){
246
247         digitalWrite(PIN_ZUMBADOR, HIGH);
248         delay(100);
249         digitalWrite(PIN_ZUMBADOR, LOW);
250         delay(500);
251
252     } else if (CO2ppm >co2_pre_alarma) {
253         digitalWrite(PIN_ZUMBADOR, HIGH);

```

```

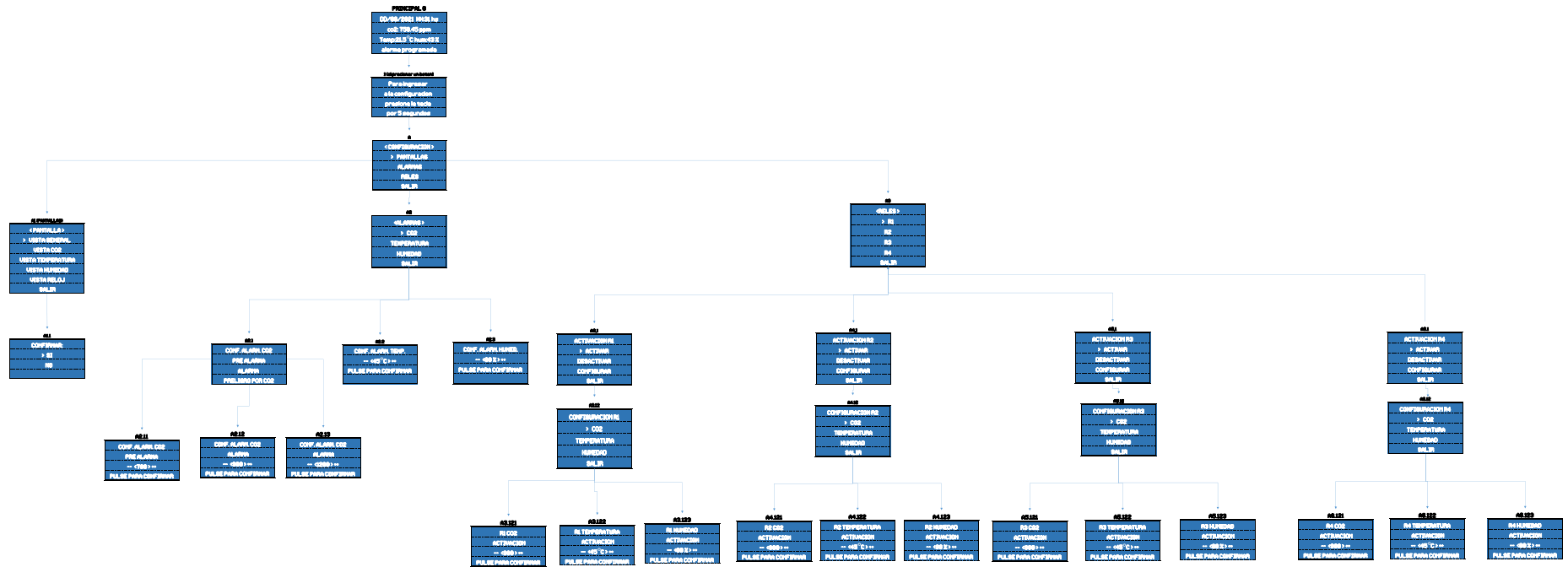
254         delay(100);
255         digitalWrite(PIN_ZUMBADOR, LOW);
256         delay(2000);
257     }
258
259     /*
260     //////////PRUEBA PULSADOR
261
262     if (POSICION != ANTERIOR) { // si el valor de POSICION es distinto de ANTERIOR
263         Serial.println(POSICION); // imprime valor de POSICION
264         lcd.setCursor(18, 1);
265         lcd.print(digitalRead(POSICION));
266         ANTERIOR = POSICION ; // asigna a ANTERIOR el valor actualizado de POSICION
267
268     }
269
270
271     if(digitalRead(PULSADOR) == HIGH)
272     {
273         lcd.setCursor(16, 1);
274         lcd.print("1");
275     }
276
277     if(digitalRead(PULSADOR) == LOW)
278     {
279         lcd.setCursor(16, 1);
280         lcd.print("0");
281
282
283     ////////////
284     */

```

```

285
286
287
288     delay(RETARDO);
289 }
290
291
292
293 //FUNCIONES
294
295 void encoder() {
296     static unsigned long ultimaInterrupcion = 0; // variable static con ultimo valor de
297                                     // tiempo de interrupcion
298     unsigned long tiempoInterrupcion = millis(); // variable almacena valor de func. millis
299
300     if (tiempoInterrupcion - ultimaInterrupcion > 5) { // rutina antirebote desestima
301                                     // pulsos menores a 5 mseg.
302
303         if (digitalRead(B) == HIGH) // si B es HIGH, sentido horario
304         {
305             POSICION++; // incrementa POSICION en 1
306         }
307         else { // si B es LOW, senti anti horario
308             POSICION--; // decrementa POSICION en 1
309         }
310
311         POSICION = min(10, max(0, POSICION)); // establece limite inferior de 0 y
312                                     // superior de 100 para POSICION
313         ultimaInterrupcion = tiempoInterrupcion; // guarda valor actualizado del tiempo de la
314         interrupcion en variable static
315     }

```

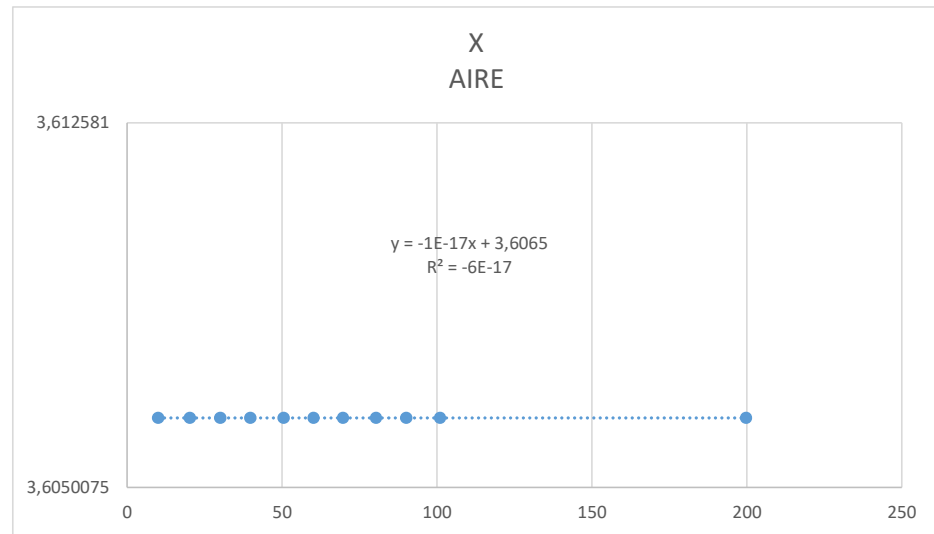
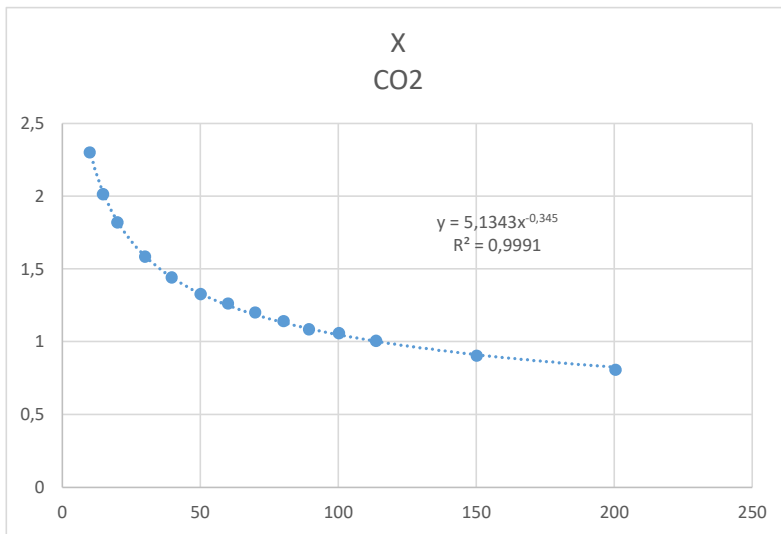
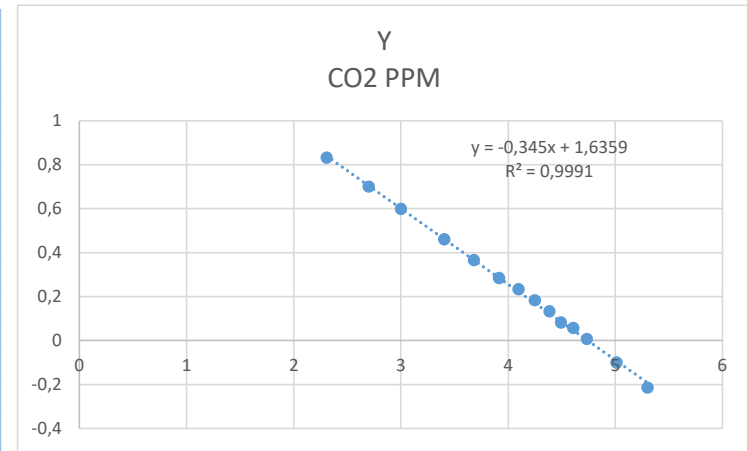



Linealización del MQ-135

CO2	
RS/RO	ppm
X	Y
10,03714513	2,300168333
14,86476834	2,014756414
20,06494433	1,821315748
30,04152573	1,585271503
39,67586194	1,442129785
50,16258368	1,328559583
60,05552953	1,263223835
69,9645014	1,201056089
80,18506213	1,141922127
89,42538859	1,085658882
100,2792119	1,058662904
113,6808454	1,006524557
150,1361104	0,904113589
200,4560234	0,807006628

AIRE	
RS/RO	ppm
X	Y
9,987848761	3,60645008
20,19108511	3,60645008
30,07085132	3,60645008
39,71876547	3,60645008
50,49650427	3,60645008
60,12994709	3,60645008
69,67471936	3,60645008
80,29406466	3,60645008
90,0429712	3,60645008
100,9747837	3,60645008
199,7201783	3,60645008

CO2 LOG NAT	
X	Y
2,306292724	0,832982308
2,698993872	0,700498302
2,998974229	0,599559179
3,402580616	0,460755688
3,680742991	0,366121038
3,915269404	0,284095335
4,095269626	0,233667053
4,247987991	0,183201244
4,38433724	0,132712919
4,493404631	0,082187067
4,607958414	0,0570067
4,73339492	0,006503365
5,011542285	-0,10080027
5,300594888	-0,2144234



Cant.	Unidad	Descripción	Monto
1 Un.		Sensor De Temperatura Y Humedad Relativa Dht11 Arduino	\$ 300,00
1 Un.		Display Lcd 2004 Backlight Azul 20x4 + Serie I2c Arduino	\$ 912,00
1 Un.		Modulo Rtc Ds3231 Alta Precision Eeprom 24c32 Arduino C/pila	\$ 475,00
1 Un.		Fuente Switching 12v 1,25a Tira Led Camara Arduino	\$ 550,00
1 Un.		Modulo Sensor De Gas Mq135 Mq-135 Arduino	\$ 490,00
1 Un.		Cables Macho Hembra 40 X 20cm Mh Dupont Arduino Protoboard	\$ 270,00
1 Un.		Modulo Rele Relay De 1 Canal 5v 10a Arduino Pic Ptec	\$ 250,00
1 Un.		Arduino Uno Ch340 C/cable Usb Compatible	\$ 1.000,00
1 Un.		Gabinete	\$ 250,00

TOTAL \$ 4.497,00

3° AÑO 2021

PROYECTO ROBOTICA

CARPETA DE CAMPO

PROFESOR: TARAMAZZO

ALUMNOS: Yacson Asprilla, Cristian Moreno, Martin Maciel, Nahuel Rodriguez.

FECHA DE ENTREGA:

GREGORIO DE LAFERRERE 17/09/2021

INTRODUCCION

Para la carpeta de campo se ha tomado parte de los diálogos entre los integrantes del grupo para la investigación, programación, desarrollo y ejecución del proyecto CALIDAD DEL AIRE.

Conversaciones en crudo:

[19:18, 20/8/2021] Cristian Curso: Yo para robótica voy a hacer lo de control de calidad deo aire

[19:18, 20/8/2021] Cristian Curso: Quiero ver si puedo hacer algo que se pueda salir a vender

[19:19, 20/8/2021] Nahuel: A ya tenes todo re bien

[19:21, 20/8/2021] Cristian Curso: Le quiero incorporar esta pantalla que es 64x128 en pixel y podes hacer mejores cosas

[19:21, 20/8/2021] Nahuel: Ese es solo un ejempli

[19:21, 20/8/2021] Cristian Curso: 🤔

[19:22, 20/8/2021] Nahuel: Tenes q conseguir los rele de 5v electronicos y le aislas la salidas al arduino y ya va como piña hajaja

[19:52, 26/8/2021] Martin: Con que te puedo ayudar?

[19:53, 26/8/2021] Nahuel: Anda a hacerle mates jajaa

[19:55, 26/8/2021] Martin: Mal jajajaajja

[19:59, 26/8/2021] Cristian Curso: mirate unos videos de este flaco explica re bien.

[20:03, 26/8/2021] Cristian Curso: El proyecto es utilizar un arduino uno con los siguientes componentes: Display 20x4 i2c ; Modulo de tiempo real DS3231; Modulo medidor de calidad de aire MQ-135; Modulo medición de temperatura y humedad DTH11; encoder rotativo con pulsador ky-040

[20:04, 26/8/2021] Cristian Curso: bajate la info de los datasheet

[20:06, 26/8/2021] Cristian Curso: y vamos a conectarles a la salida un modulo de 4 reles con salidas opto acopladas que se van a poder configurar en diferentes valores de humedad temp, y co2

[20:43, 26/8/2021] Nahuel: Faltaria el codigo de arduino

[20:44, 26/8/2021] Cristian Curso: xps?

[20:44, 26/8/2021] Nahuel: si es que le tuve que sacar unas hojas

[20:44, 26/8/2021] Cristian Curso: ahh

[20:44, 26/8/2021] Nahuel: pero los abre en el celu

[20:45, 26/8/2021] Cristian Curso: Esta es la intro. fijence si sumamos algo o algo esta mal

[20:45, 26/8/2021] Cristian Curso: Introducción:

El proyecto surge a través de la necesidad de conocer cual es la calidad de aire que respiramos ya que con la actual pandemia esto paso a ser un punto muy importante. Por lo que conocemos la mayor tasa de contagios surgen en ambientes muy poco ventilados. El aire viciado representa la situación en la que el aire que respiramos en

interiores ha sido contaminado y no se ha extraído ni renovado. Al pasar ese aire suficiente tiempo en el ambiente sin moverse hacia los exteriores e instalarse dentro de nuestros hogares, puede llegar a ser perjudicial para la salud.

Podemos conocer que tan contaminado está el aire a través de la cantidad de dióxido de carbono CO₂. El dióxido de carbono es uno de los contaminantes más habituales y que más afectan en e...

[20:53, 26/8/2021] Martin: ahi estan los datasheet abreviados

[21:12, 26/8/2021] Cristian Curso: Dale

[22:45, 26/8/2021] Cristian Curso: El profesor quiere que dividamos las tareas y después cada uno tiene que explicar su parte

[22:48, 26/8/2021] Cristian Curso: Hasta ahora solo esta esta esta pantalla ponele que es la principal. Hay que diseñar todas las demás y los menús para que el usuario pueda cambiar la configuraciones

[23:06, 26/8/2021] Cristian Curso: Es medio un quilombo.

[11:08, 16/9/2021] Nahuel: Si tendrias q volver a pasarme lo q le mandaron a taramazo para ver q era lo q teniamos.

[11:09, 16/9/2021] Cristian Curso: Esto es lo que tengo así lo armé

[11:10, 16/9/2021] Cristian Curso: Claramente hay que mejorarlo

[11:11, 16/9/2021] Nahuel: Si eso lo paso

[11:12, 16/9/2021] Cristian Curso: todos los + y - toman de gnd y +5 de la pinera de arduino

[11:13, 16/9/2021] Cristian Curso: menos el + del MQ-135 QUE TOMA DE VIN Y ENTRA A UN 7805 DE AHI VA AL MQ-135

[11:14, 16/9/2021] Cristian Curso: porque si lo alimento del +5 comun me tira el voltaje de todo abajo ahora estoy cambiando eso en el circuito

[11:16, 16/9/2021] Nahuel: Dale no hay drama despues te preparo algo en el eplan igual si erro algo me decis y lo cambiamos.

[11:17, 16/9/2021] Cristian Curso: Dale joya!!

[11:18, 16/9/2021] Cristian Curso: trata de conseguir el pin out completo de arduino algo asi

[11:23, 16/9/2021] Nahuel: Ese no es el completo?

[11:24, 16/9/2021] Nahuel: Si vas con taramazo le preguntas si ya va a ser siempre presencial?

[11:29, 16/9/2021] Cristian Curso: yo fui el martes y ayer también. Practicas y robotica van a ser presenciales!

[11:30, 16/9/2021] Cristian Curso: Si pero por ej el pinout del integrado es al pedo que este

[11:30, 16/9/2021] Cristian Curso: Lo importante es lo que esta marcado en la placa de cada pin

[11:31, 16/9/2021] Cristian Curso: el resto es como podes configurar cada pin

[11:31, 16/9/2021] Cristian Curso: y varia de la programación

[11:31, 16/9/2021] Cristian Curso: en arduino un pin puede ser entrada o salida depende de como lo programes

[12:41, 16/9/2021] Nahuel: algo asi?

[12:44, 16/9/2021] Cristian Curso: Así esta Joya

[12:45, 16/9/2021] Nahuel: Lo rojo lo saco?

[12:45, 16/9/2021] Nahuel: Los nombres de los pines estan bien?

[12:46, 16/9/2021] Cristian Curso: Si podes meter un espacio entre la descripción y el pin mejor

[12:46, 16/9/2021] Cristian Curso: del lado derecho

[13:43, 16/9/2021] Nahuel: Lo q no se q es r1 y a8

[14:27, 16/9/2021] Cristian Curso: A8 es un buzzer conexión directa

[14:28, 16/9/2021] Cristian Curso: Y R1 es un relé que se acciona la bobina con un transistor

[15:03, 16/9/2021] Nahuel: ahi esta creo

[15:33, 16/9/2021] Cristian Curso: Aca podemos poner 3 bornes

[15:33, 16/9/2021] Cristian Curso: y poner el texto Salida alarma CO2

[15:34, 16/9/2021] Cristian Curso: y las 3 borneras

[15:34, 16/9/2021] Nahuel: Si le pondo

[17:48, 16/9/2021] Nahuel: Ese le tuviste q poner por q te tiraba abajo lo fuente de 5v?

[17:49, 16/9/2021] Cristian Curso: Si

[17:49, 16/9/2021] Cristian Curso: y ahora le tuve que poner un disipador porque calienta un montón

[17:50, 16/9/2021] Cristian Curso: hoy le voy a preguntar al profe como solucionarlo

[17:52, 16/9/2021] Cristian Curso: $I = v/r \implies 5v/33ohms \implies 150mA$

[17:52, 16/9/2021] Cristian Curso: No tendría que tener problemas con 150 mA el 7812 es 1.5 Amp

[17:52, 16/9/2021] Cristian Curso: Algo mas pasa

[17:53, 16/9/2021] Cristian Curso: Capaz esta mal el mq135 habria que medir bien el consumo

[17:54, 16/9/2021] Nahuel: El arduino

[17:54, 16/9/2021] Nahuel: Lo q tengo entendido muchos eligen alimentarlo con 9 v

[17:54, 16/9/2021] Nahuel: 12 dicen q ya esta en el limite

[17:55, 16/9/2021] Cristian Curso: Si, pero no tengo una fuente de 9 con ese conector agarre la fuente del deco de telecentro 😊

[17:55, 16/9/2021] Cristian Curso: En este datasheet dice otra cosa

[17:56, 16/9/2021] Cristian Curso: Si son 950mw es casi 1w en 5V $\implies 1w / 5v \implies 200mA$

[17:58, 16/9/2021] Nahuel: De ultima no agarres los 12 de vin

[17:58, 16/9/2021] Nahuel: Agarralo del cable del jack

[18:01, 16/9/2021] Cristian Curso: lo simule

[18:01, 16/9/2021] Cristian Curso: Los calculos estan bien

[18:02, 16/9/2021] Cristian Curso: Si tal cual

[18:03, 16/9/2021] Cristian Curso: Pero para mi en la realidad esta consumiendo mas de 192 mA

[18:03, 16/9/2021] Cristian Curso: Probé con el tester pero no me midio nada.

[18:03, 16/9/2021] Nahuel: Por lo q dice

[18:03, 16/9/2021] Nahuel: A 12 v

[18:03, 16/9/2021] Nahuel: Lo maximo q podes obtber del regulador de arduino sin q caliente es 285

[18:04, 16/9/2021] Cristian Curso: Si por eso en principio no tendría que tener problemas