

GusControl

Dispositivo para la gestión remota de
un observatorio astronómico

Introducción

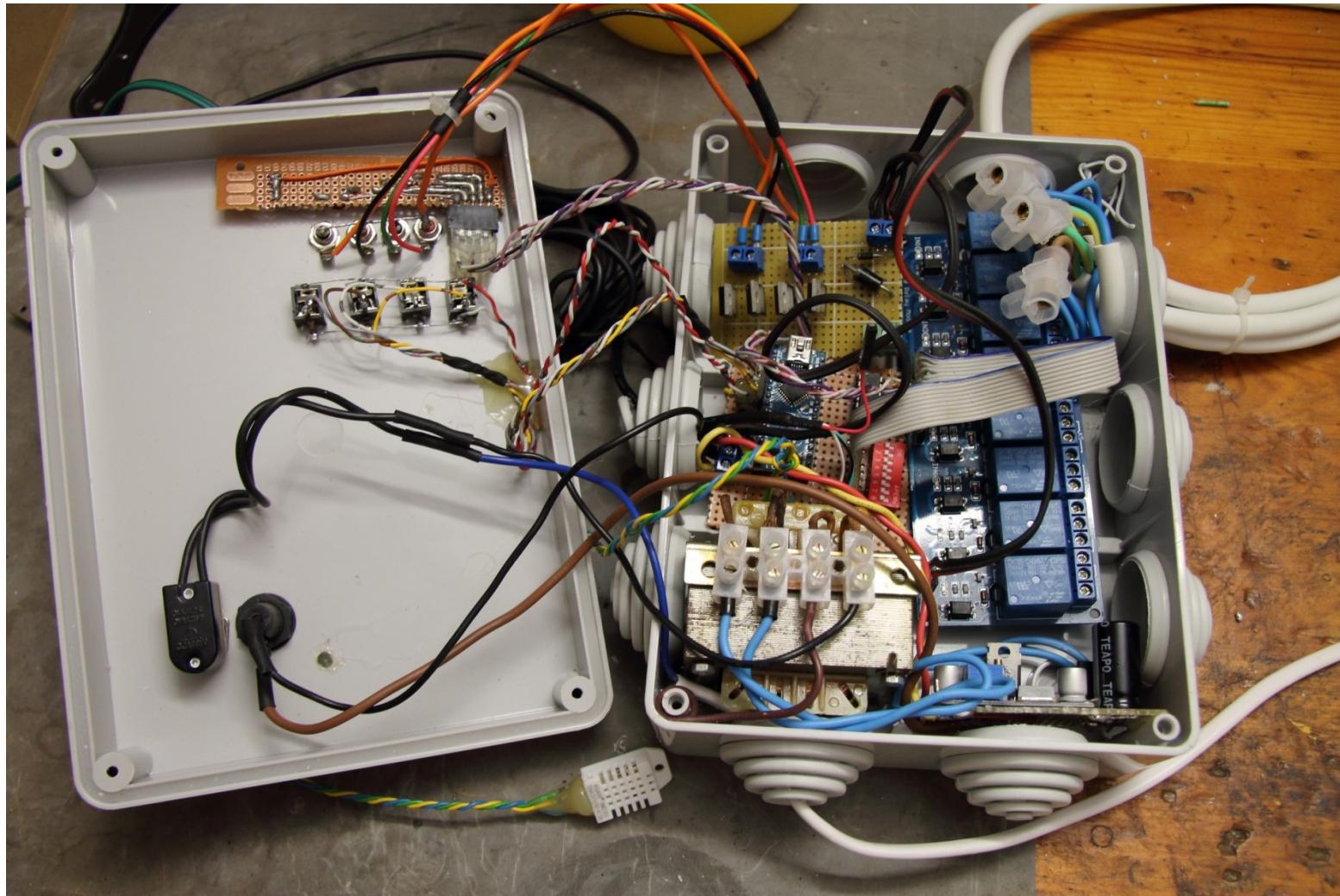
- Este circuito se diseña para poder operar de manera básica otros mecanismos y dispositivos presentes en el observatorio. Encender y apagar por un conjunto de relés, modular la potencia en PWM por un conjunto de transistores FET y tomar medidas por distintas sondas analógicas.
- Debe ser suficientemente autónomo y poderse manejar a través de un puerto USB de un ordenador o tarjeta-PC.
- Se realizará utilizando materiales y elementos de los que ya dispongamos, es un dispositivo sostenible. Entre ellos el 4011, los IRF1404, el Arduino NANO, los LM35, el cable de comunicaciones USB-TTL, el transformador, la placa de relés, el conversor DC-DC... de entre los cajones, donantes e inventitos reciclados

Decisiones de diseño

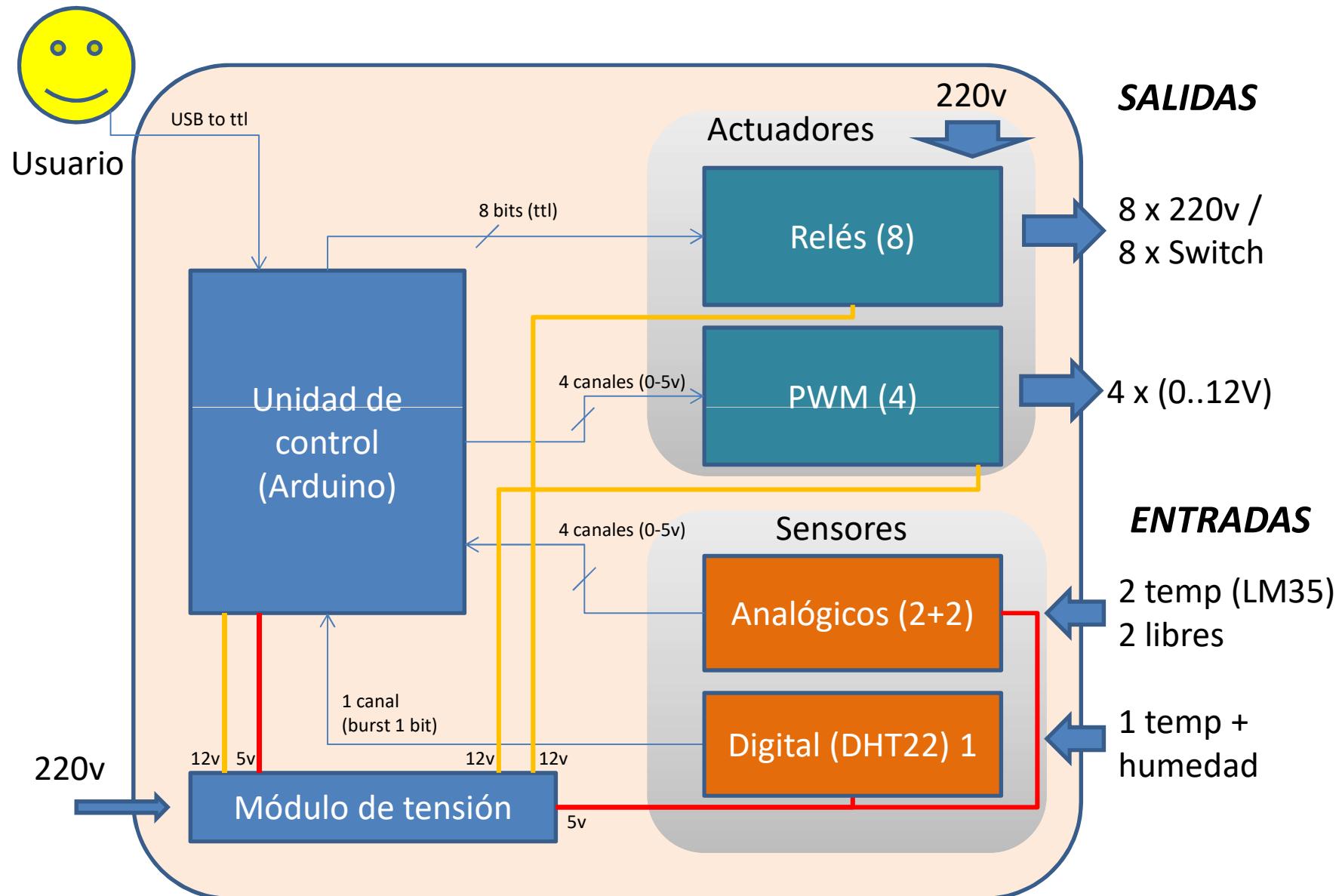
- La inteligencia será a cargo de un Arduino.
- La conectividad es a través de un puerto USB-TTL.
- Implementa un lenguaje regular que permite este control.
- Se realiza a través de un emulador de terminal y comandos de texto.

DISPOSITIVO FÍSICO

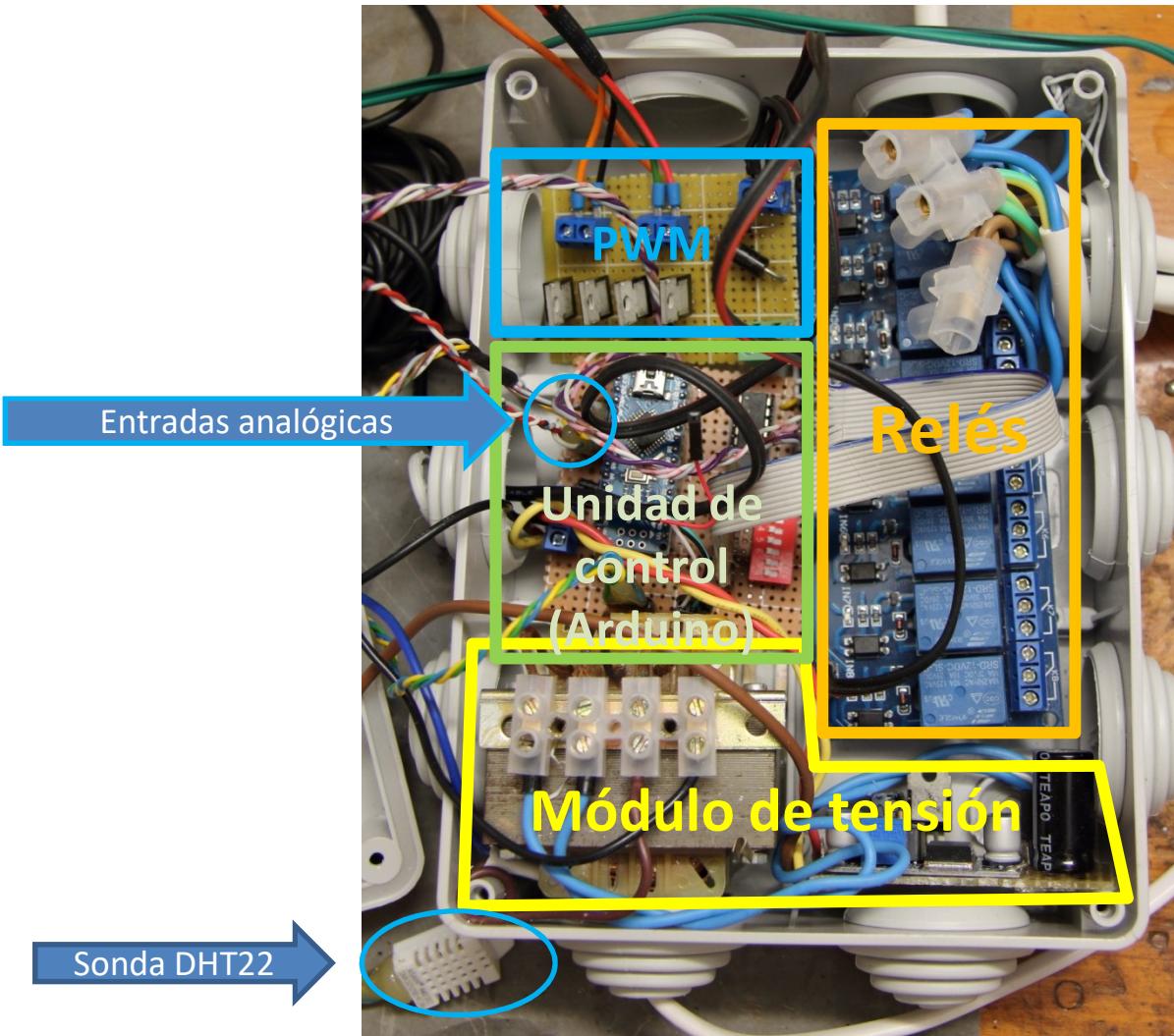
El GusControl



Bloques físicos

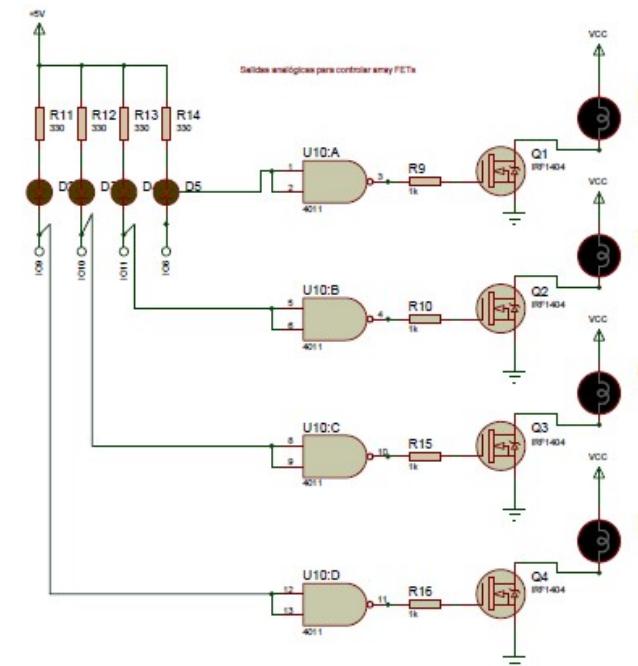
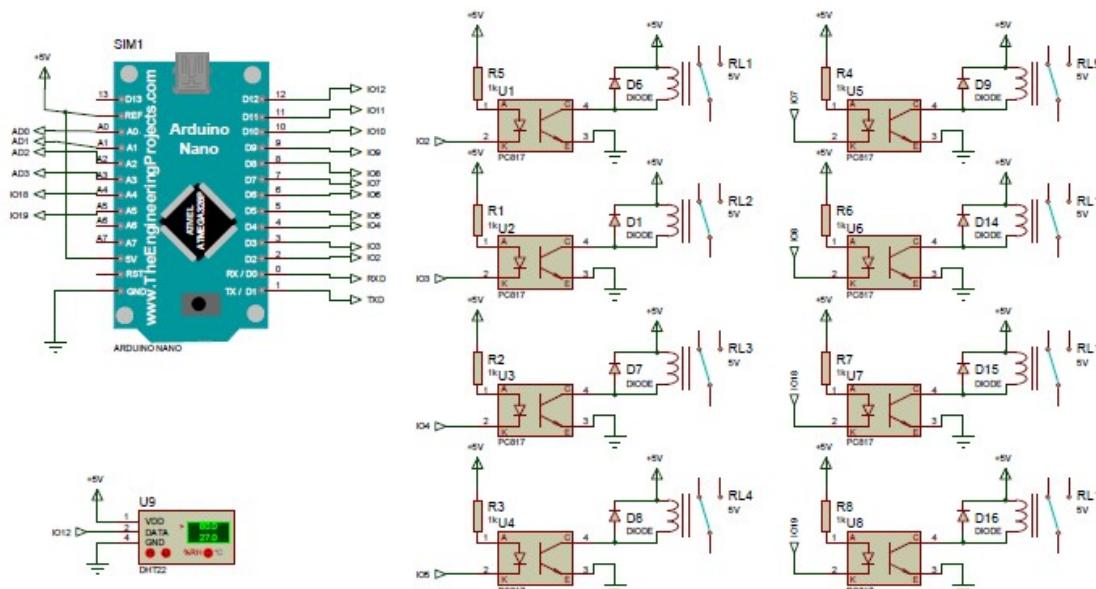
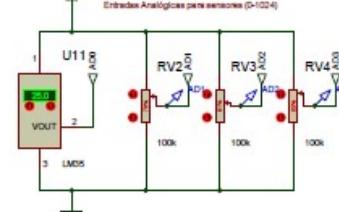
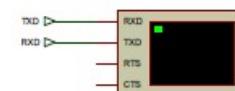
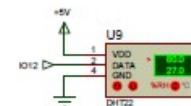
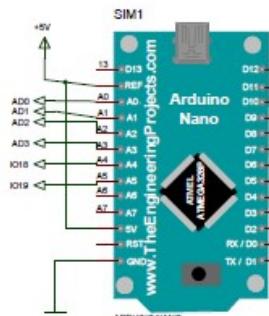


Bloques físicos



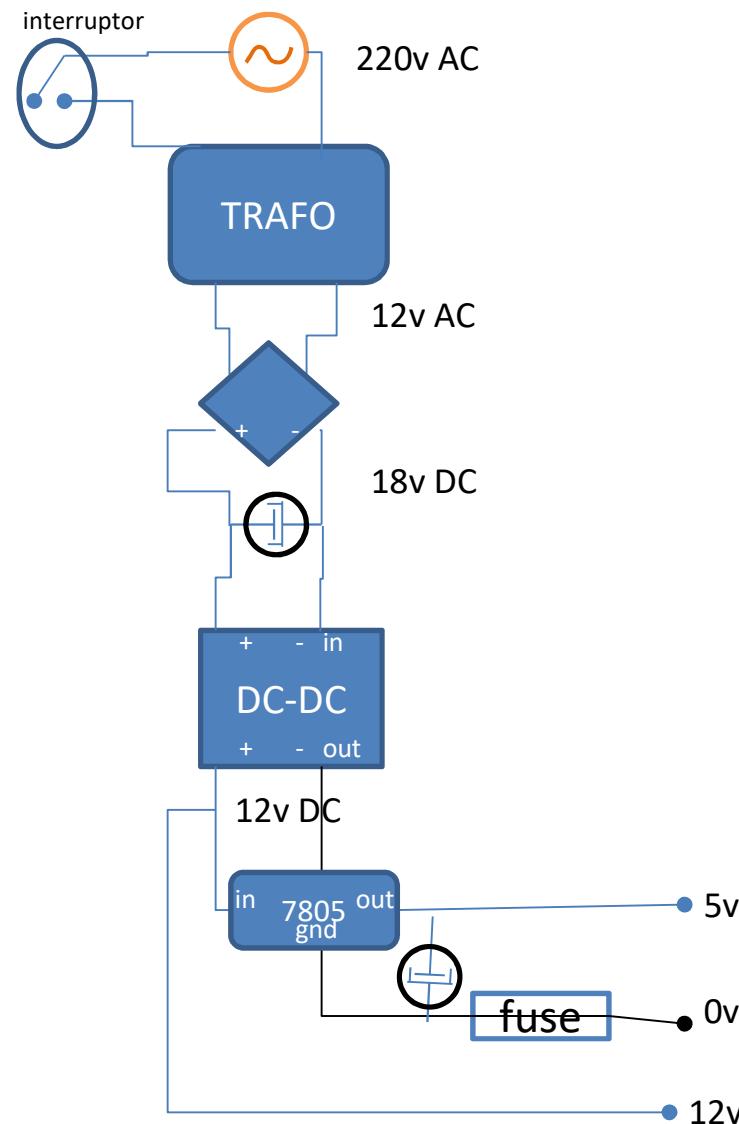
Esquema electrónico

- El circuito se ha emulado con “Proetus” y en él se ha dibujado el esquema.
- Existen librerías que emulan el Arduino, se ha incorporado y aplicado el Sketch compilado para su ejecución en el emulador.
- También existe el módulo simulador de la sonda de temperatura y humedad, aunque no he podido obtener resultados en la emulación, en el mundo real funciona perfectamente.



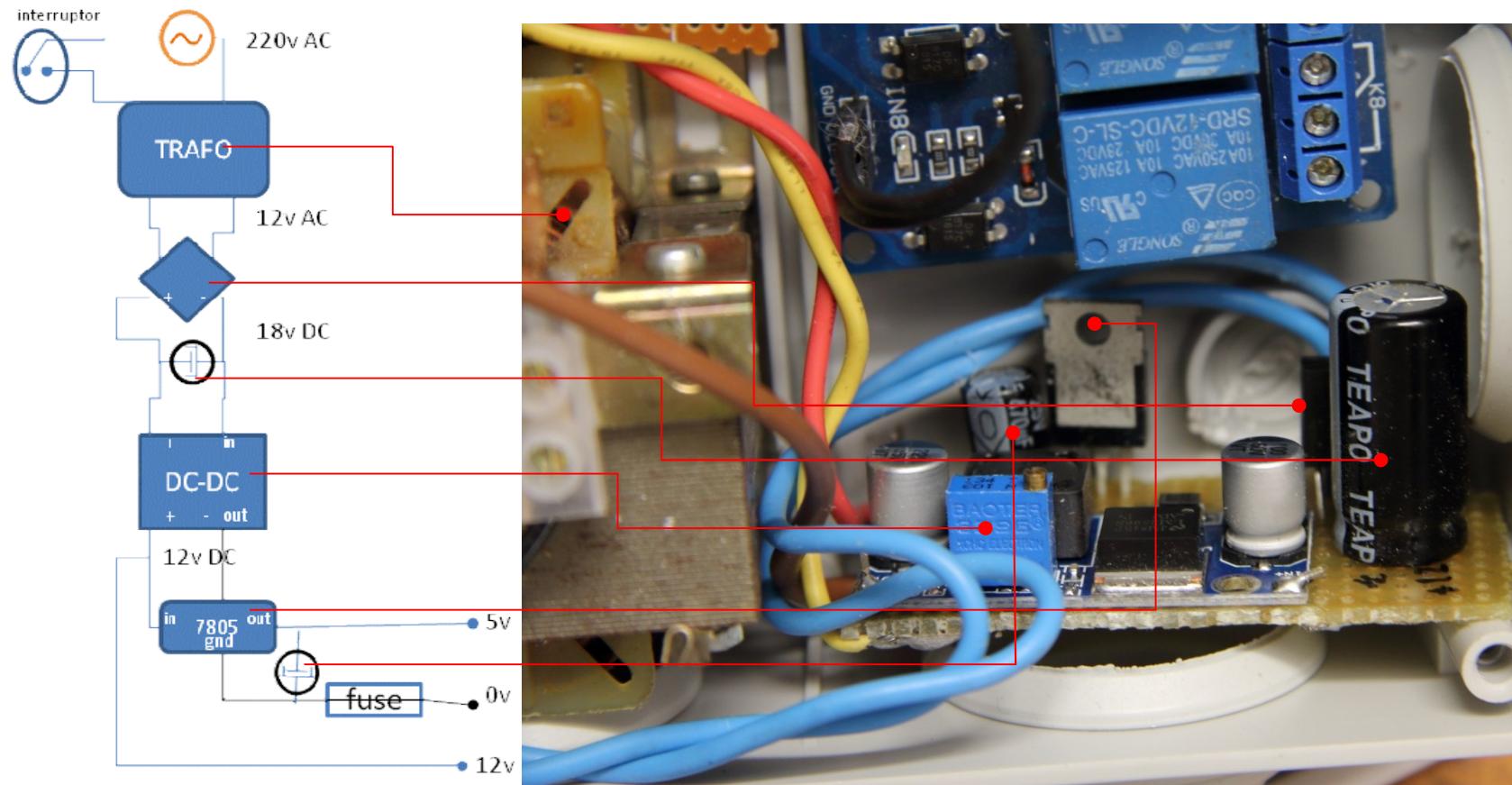
Módulo de tensión

- El módulo de tensión recibe 220v de la red y proporciona al dispositivo doble alimentación a 12v y 5v.
- Dispone de un interruptor que corta la tensión de entrada de 220v.
- Se protege de la red con transformador de 220v a 12v reciclado de unas bombillas dicroicas.
- Un puente de diodos y un condensador electrolítico rectifica y filtra la tensión en una primera etapa. Aproximadamente 18v con rizado.
- Se entrega a un módulo DC-DC en primera instancia del que se obtiene la tensión de 12v. Ajustado y Regulado.
- Se incorpora un 7805 y un condensador electrolítico para obtener en cascada los 5v.
- La corriente se recoge por el negativo con un fusible que protege los dispositivos internos. Es un fusible de cristal de 2,5 A



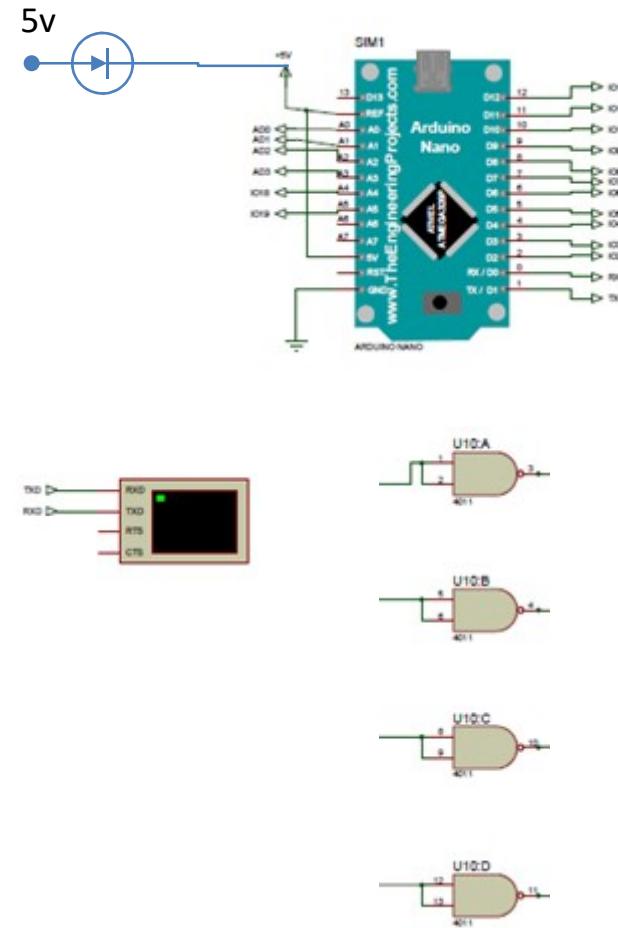
Módulo de tensión

- El cable amarillo entrega 12v
- El cable rojo entrega 5 v
- El cable marrón entrega 0v al fusible que se monta en el panel frontal
- Los cables azules conectan el secundario del transformador
- El interruptor actua sobre la tensión que llega al primario



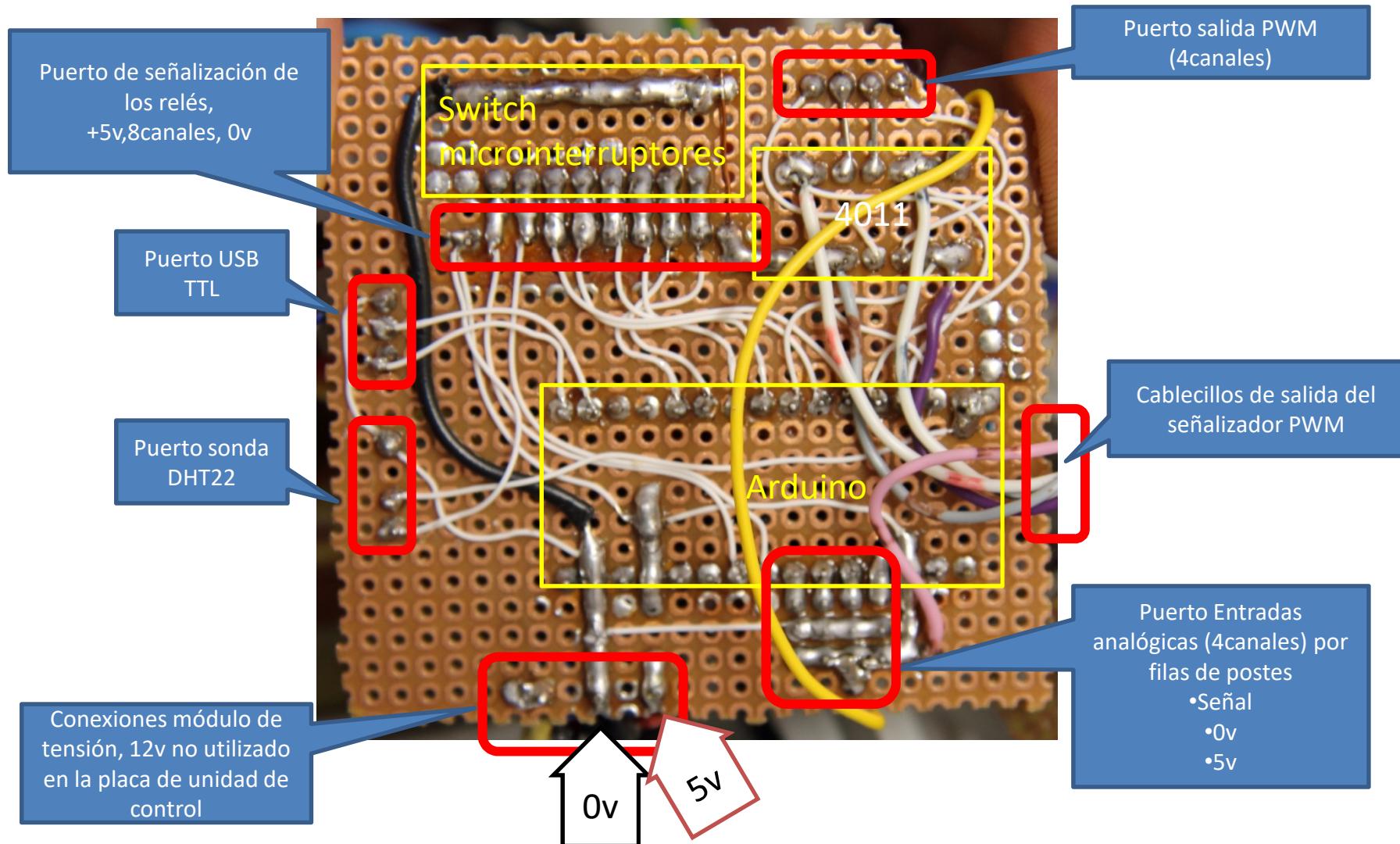
Unidad de Control, Arduino

- A la placa llega alimentación a 5 y 12 voltios.
- La entrada de 5v tiene un diodo, desde donde se alimenta el arduino directamente a la pata 5v, evitando usar el circuito regulador del propio arduino. Es decir, VIN no se utiliza.
- Los 12v no se usan, se pensó para alimentar el circuito CMOS 4011 que actuara como adaptador de niveles para las etapas de potencia de PWM. Este chip se alimenta con 5v y se excitan los MosFET con esta tensión.
- La comunicación con el usuario se realiza por los puertos RxD y TxD del arduino. Para ello se utiliza un cable USB-ttl por puerto serie. En el esquema se representa como el emulador de terminal. De los cuatro cables que salen del cable, se conectan TxD, RxD y GND al arduino, el cable de 5v queda al aire. No se vincula la corriente del ordenador al arduino, este debería funcionar con el módulo de tensión del Gus-control. Tampoco se utiliza el mini-USB del arduino, pues no se tendría control de la alimentación. El objetivo es tener las alimentaciones separadas. (1)
- En la placa se encuentra un 4011 CMOS es un chip con cuatro puertos NAND. Se puentean las dos entradas para hacerlo trabajar como inversor y driver para gobernar la “puerta” del MosFET de la etapa de potencia PWM. (lo uso así, porque es el que tenía por casa)
- No he puesto resistencias entre entradas y salidas... de los chips. El arduino y su procesador tienen las entradas salidas adaptadas, con los pull-ups o protecciones pertinentes.
- Se ha puesto unos microinterruptores cortocircuitando a tierra la salida de las señales de relé, con ello se enciende el dispositivo asociado al relé, sí o sí. Siempre que se le dé al interruptor del panel y se alimente esa placa.



Unidad de Control, Arduino

- Se construye artesanalmente sobre una placa de prototipo topeada.
- El cable amarillo conecta la alimentación del chip CMOS 4011, se ha conectado a 5v, se pensó para 12v pero la señalización lógica del arduino no era bien interpretada, a 5v funciona correctamente.

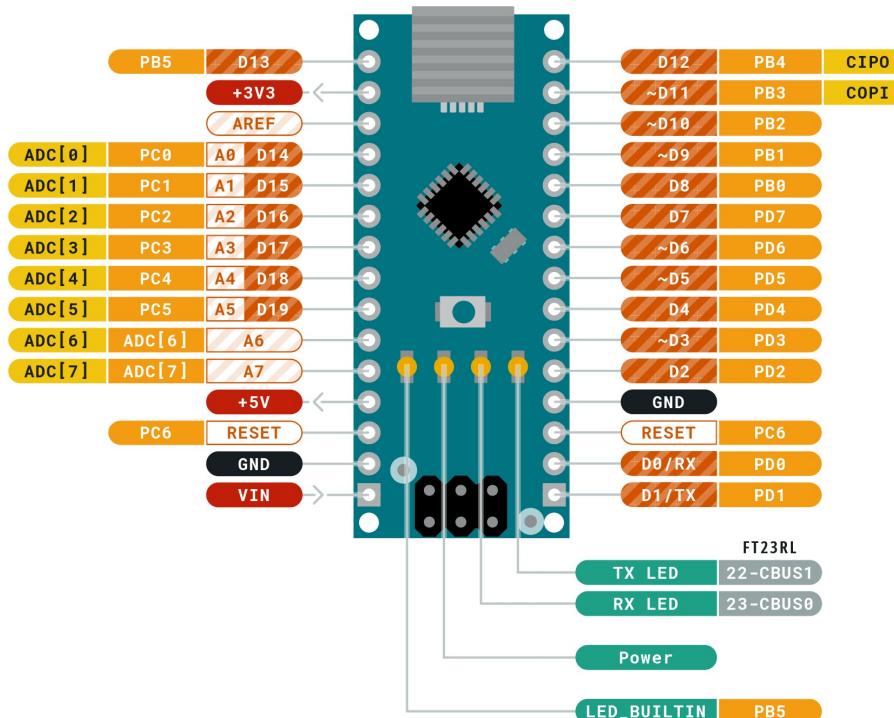


Unidad de Control, Arduino

En este montaje se utiliza un Arduino NANO.
Es fácil de programar a través del IDE (entorno de desarrollo integrado), y dispone de librerías y funciones adecuadas para el GusControl.
Es económico y robusto.

Practicamente todas sus patas están en uso :D :D

El utilizado es un clon chino, pero funciona.



■	Ground	■	Internal Pin	■	Digital Pin	■	Microcontroller's Port
■	Power	■	SWD Pin	■	Analog Pin	■	
■	LED	□	Other Pin	■	Default	■	

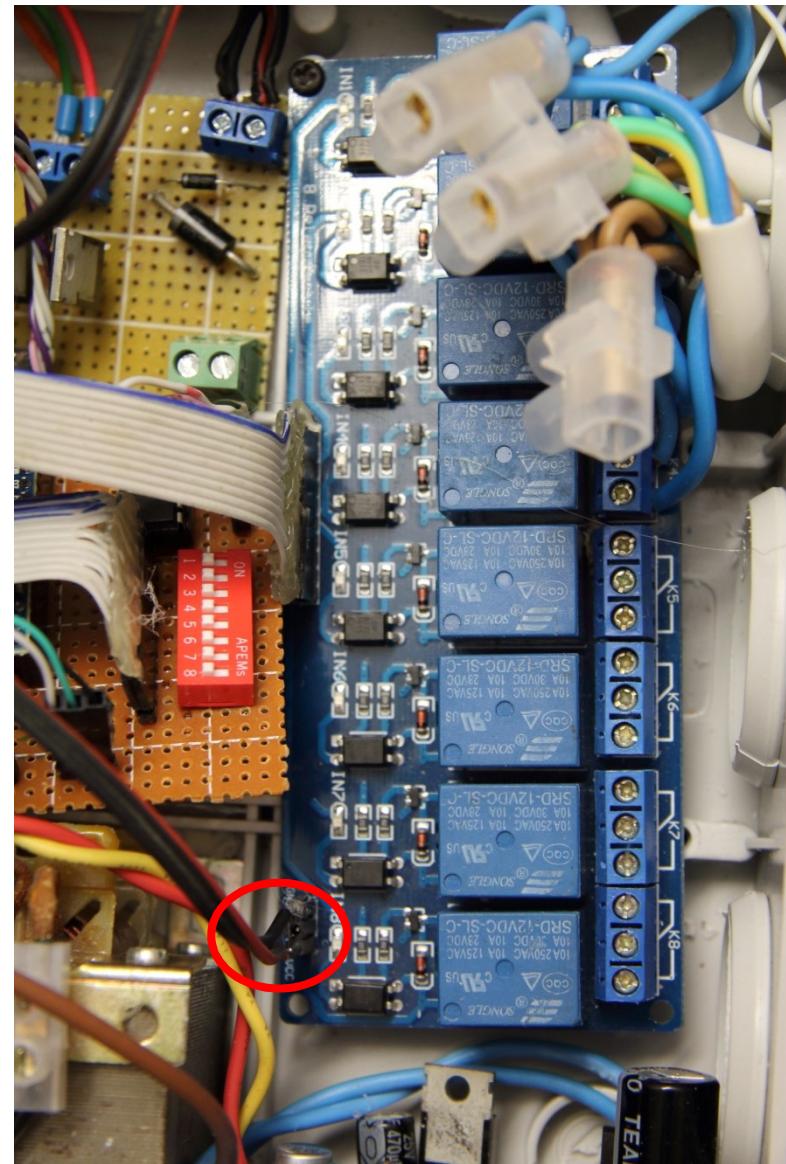
ARDUINO .CC



This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International Licence. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

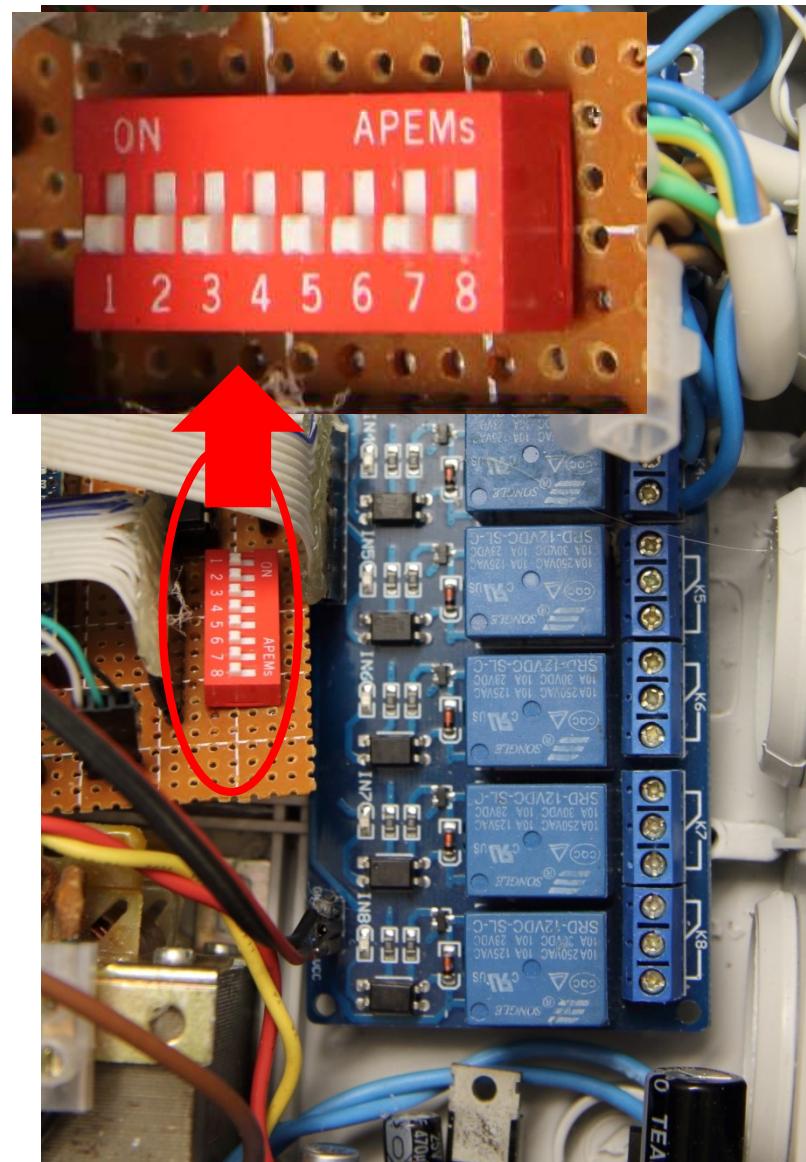
Los Relés (I)

- La placa de relés es un dispositivo comúnmente utilizado para bricolajes electrónicos. En este caso particular es una placa de 8 relés que se activa con lógica negativa y trabaja a 12V.
- Las señales de control que provienen del módulo de control, lo hacen por un cable plano de 10 contactos donde además de la señalización de los 8 relés, se añade +5v y Gnd.
- Está optoacoplado aislando los circuitos de tensión y señales de entrada y salida.
- En concreto, esta placa debe alimentarse con 12 v en el conector que se ve en la parte inferior porque sus relés necesitan esa tensión para activarse. (hay versiones de esta placa que funcionan a 5V)



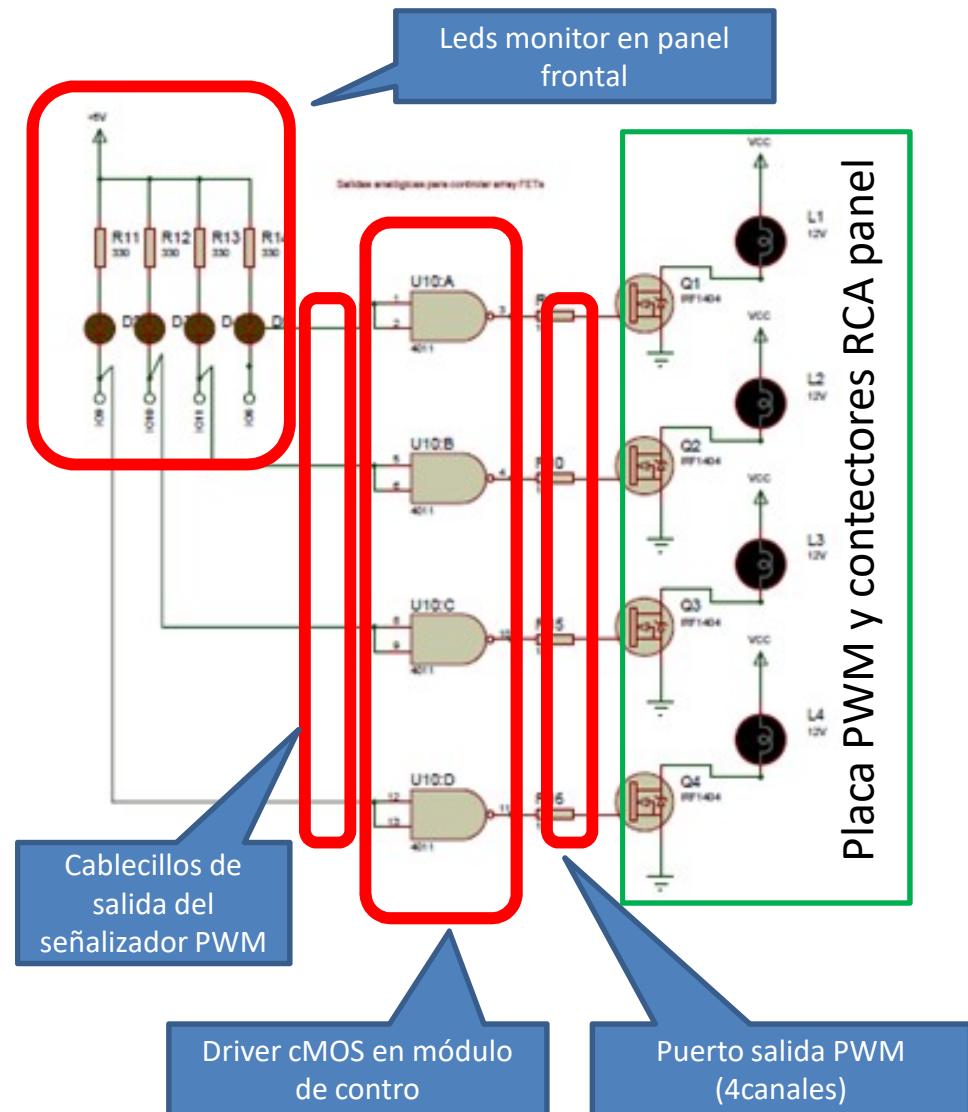
Los Relés (II)

- Como se ha indicado, los relés sólo se activarán cuando se alimente la tarjeta con 12v.
 - Existe una tira de 8 microinterruptores que activan estos canales y en el caso de ponerlo a “on” siempre, independientemente de lo que diga la unidad de control, al encender GusControl activará ese relé. Puede servir para Test de un relé o del dispositivo que tenga asociado.
 - Los 4 primeros Relés tienen un conector schuko que se alimentan con el cable de tensión externo. Con ello se pueden controlar hasta 4 dispositivos a 220v.
 - Los 4 últimos Relés están libres, para el uso que se precise.
-
- La corriente que circule por los bobinados de los relés es proporcionada por el módulo de tensión. Por eso se recomienda que en la parte de PWM se añada una fuente externa, para evitar que un sobreconsumo produzca desconexiones.

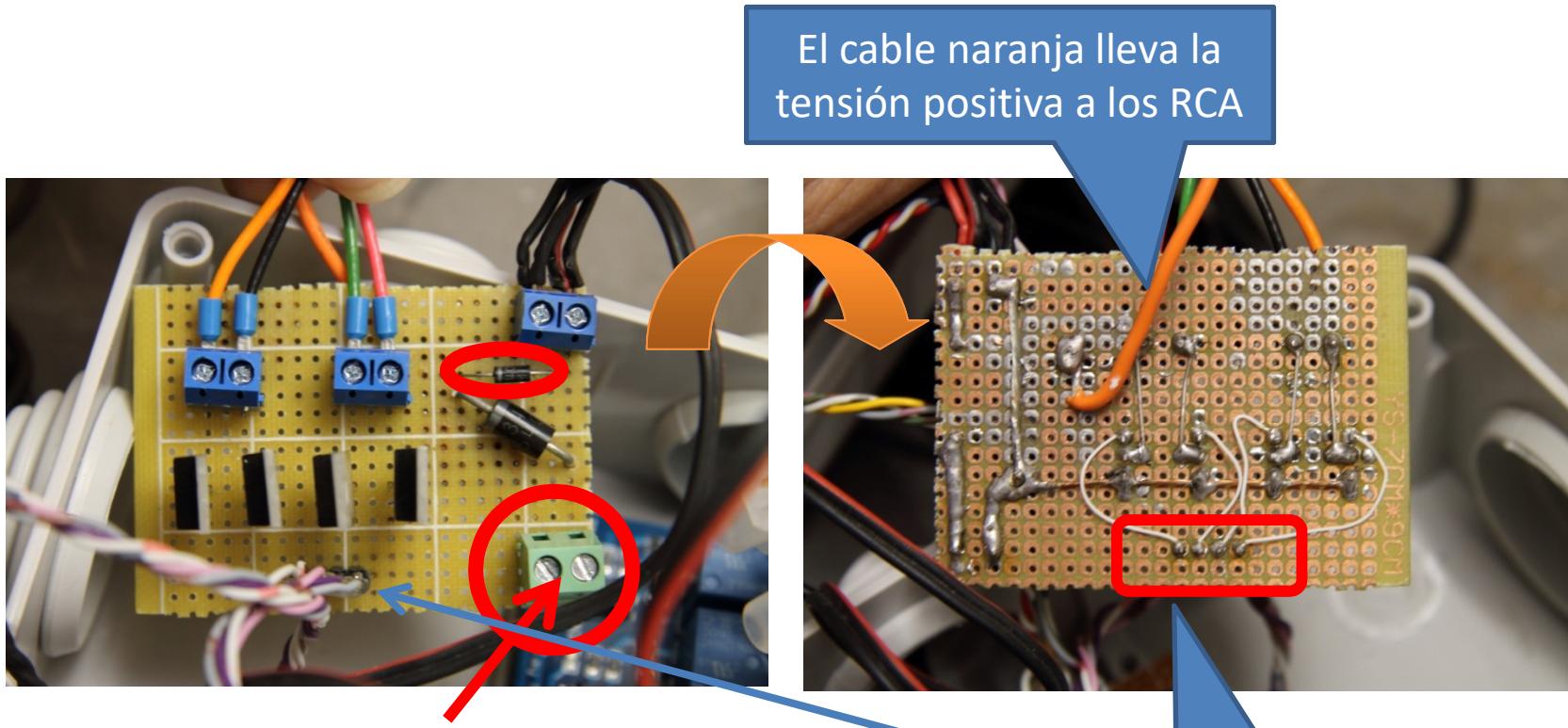


PWM, modulación de potencia por amplitud de impulso

- El circuito CMOS 4011 que actuará como adaptador de niveles para las etapas de potencia de PWM. Este chip se alimenta con 5v y se excitan los MosFET con esta tensión.
- La monitorización de la potencia se hace con una plaquita específica donde se encuentran las resistencias y leds que son gobernadas por las salidas del arduino.
- La lógica,
 - cuando el arduino se enciende, las patas de salida están en alta impedancia, estas resistencias y leds, actúan como pull-up señalizando la puerta del 4011 con un “1” lógico y su salida es un “0” 0 voltios que se dirigen a la “puerta” (gate del MOSFET) está en “corte”. El led no luce.
 - Cuando el arduino está a “1” ocurre lo mismo que el caso anterior, el MOSFET en “corte” y el Led no luce
 - Sólo cuando el arduino señala un “0”, el led luce y su corriente viene directamente del módulo de tensión y el 4011 señalará un “1” 5v a la puerta del MOSFET, entrará en estado de conducción.
- Estos componentes están en diversas placas
- Las resistencias y Leds adjuntos al panel frontal
- El 4011 en el módulo de control
- Y Los MosFET en la placa de modulación de potencia PWM
- Las resistencias sólo están en el simulador, no he puesto ninguna en el circuito real
- Las lámparas son los conectores RCA que hay en el frontal donde conectar las cargas, la parte externa es el negativo y la parte interna es el positivo (para aquellos dispositivos que tengan polaridad)



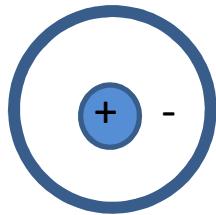
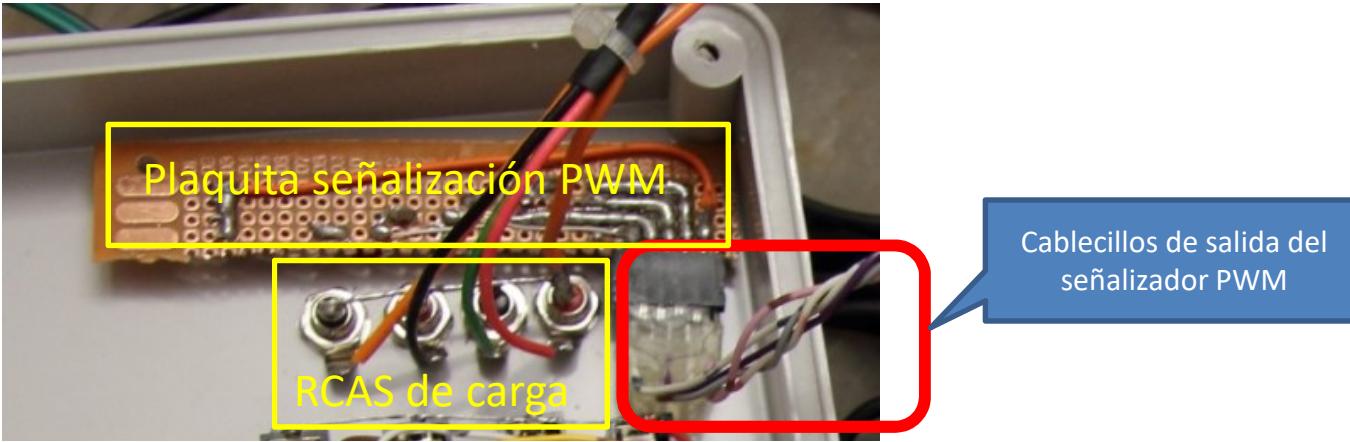
PWM, modulación de potencia por amplitud de impulso



Para pruebas se alimenta la placa con 12v del módulo de tensión. Deberá ser sustituido cuando se conecten dispositivos con alto consumo de corriente por una conexión directa a los bornes dispuestos para ello. Para desconectar la alimentación interna, se deberá cortar la pata del diodo chiquitín, el señalado por el circulito. **Se recomienda añadir una fuente externa, que se podrá controlar con los relés**

La entrada actúa directamente sobre las patas “gate” de los MOSFET

PWM, modulación de potencia por amplitud de impulso

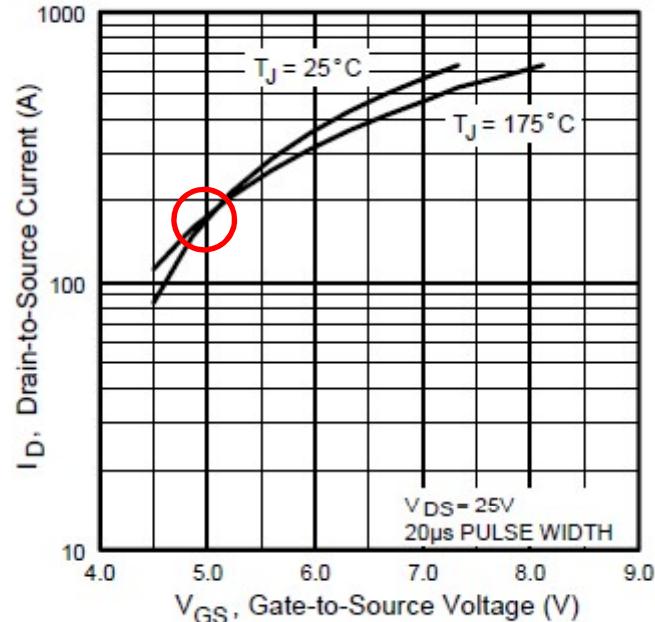
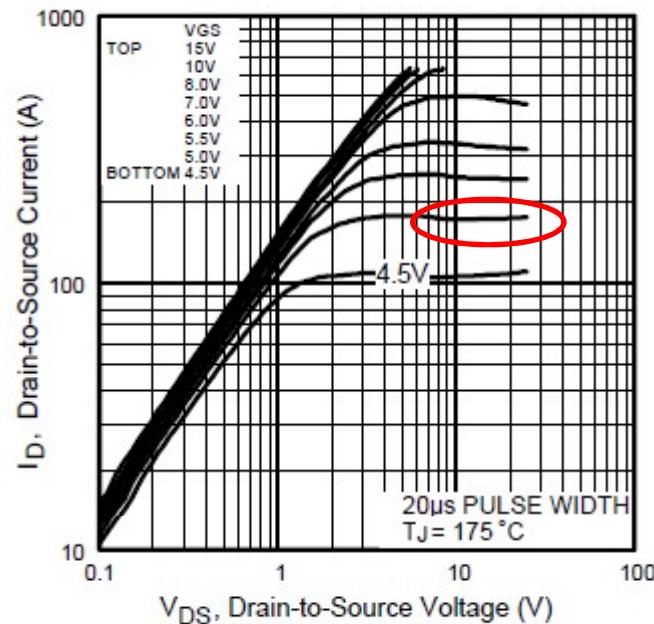


Conejito RCA

- Pin central Positivo (cable naranja)
- Carcasa Negativo (otros cables de los borneros conectados a los “drenajes” de los MOSFET)

Los MOSFET tienen protección para evitar que las cargas inductivas destruyan el transistor. Podría ocurrir que el diodo “gordo” se quemara... en ese caso, puentearlo... sólo está por si convive doble alimentación, y la fuente externa es mayor de 12v... en cualquier otro caso, sería adecuado puentearlo (cortocircuitar sus extremos). Si la tensión del dispositivo es menor de 12v habría que cortar el diodo pequeño sí o si.

PWM, el irf1404

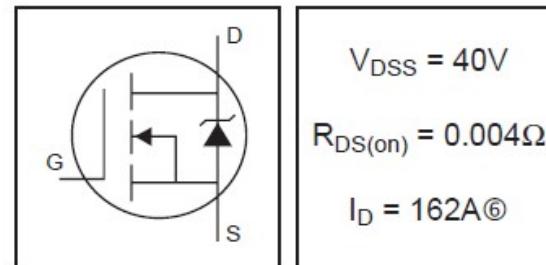


Este MosFET tiene una resistencia directa muy pequeña del orden de 4 milésimas de ohm a 10 v. Esto le permite manejar grandes intensidades, como se ve en los gráficos con una tensión de 5V en la puerta puede manejar del orden de **120 Amperios**. Significa que antes que se queme el transistor, se quemará algo... la fuente, la carga, algún cable... Es una bestia!!!

Además la caída de tensión será mínima y la potencia a disipar casi nula. No debería calentarse lo más mínimo para las cargas habituales, por eso no he montado disipadores, me los ahorro.

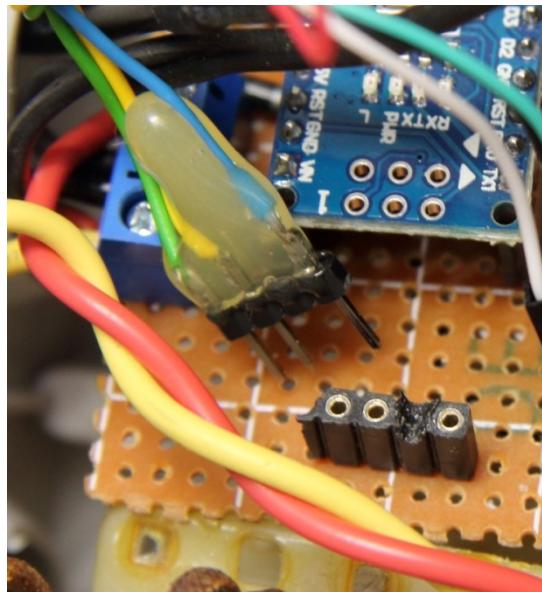
IRF1404

HEXFET® Power MOSFET

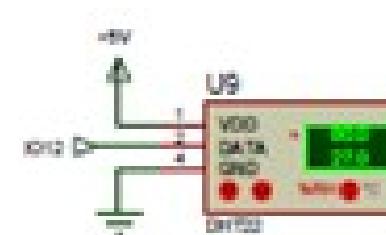


Sonda de humedad y temperatura (DHT22)

- La sonda se conecta al arduino por el puerto IO12.
 - Se lee con una ráfaga de bits que se ha de interpretar con una librería específica para Arduino.
 - Está calibrada de fábrica y es suficientemente precisa como para tomarse de medida de referencia.
-
- El zócalo de conexión tiene el espaciado de 4 postes, pero sólo se utilizan el 1, 2 y 4. Para evitar conectarlo incorrectamente se ha tapado el 3.

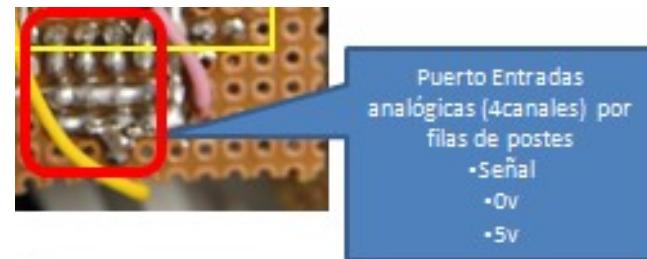
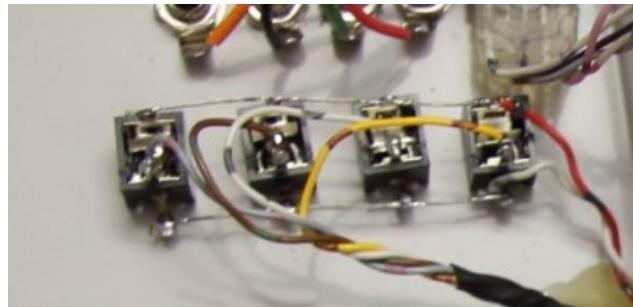


Model	AM2302
Power supply	3.3-5.5V DC
Output signal	digital signal via 1-wire bus
Sensing element	Polymer humidity capacitor
Operating range	humidity 0-100%RH; temperature -40~80Celsius
Accuracy	humidity +2%RH (Max +5%RH); temperature +-0.5Celsius
Resolution or sensitivity	humidity 0.1%RH; temperature 0.1Celsius
Repeatability	humidity +-1%RH; temperature +-0.2Celsius
Humidity hysteresis	+-0.3%RH
Long-term Stability	+-0.5%RH/year
Interchangeability	fully interchangeable



Entradas analógicas

- Las entradas analógicas se realizan a través de unos conectores Jack estéreo de 3,5 mm.
- El vástago tiene la referencia de tierra 0v
- El anillo central la señal
- La punta la señal de referencia de 5V
- El canal 1 y 2 interpretan las sondas tipo LM35 donde señalan la temperatura en milivoltios. No están calibradas y se especifica la lectura con esa interpretación. De manera que puede ser más o menos fiable su medida, pero, sí que es relevante la variación de este valor. Se señalizan como T1 y T2 en el panel frontal.
- El canal 3 y 4 se emite el valor leído de la señal de entrada que debe estar entre 0-5v y se digitaliza con 10 bits, de manera que se obtienen 1024 valores distintos desde 0 a 1023. La interpretación de estos valores vendrá determinado por el diseño de la sonda y su circuito de adaptación de señal. Se señalizan como A3 y A4 en el panel frontal
- Se entregan las dos sondas de temperatura LM35 con un cable coaxial, para poderlas instalar en las superficies o lugares donde tomar la temperatura, por ejemplo, en una célula peltier, o en una cinta calentadora antirocio.

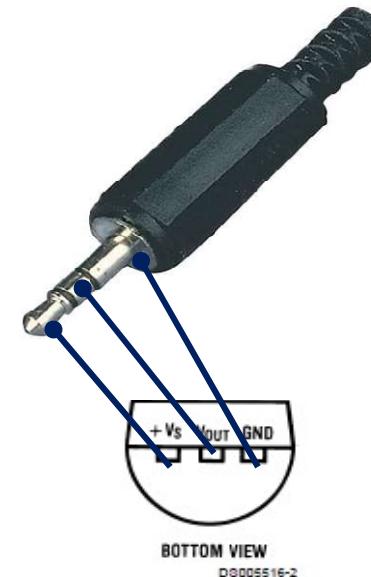


Entradas analógicas, LM35

- A la práctica es sensible a la resistencia de los contactos ohmnicos de los conectores. Y a la precisión del conversor A/D del Arduino, sus lecturas no deben ser tomadas como temperaturas correctas, si no como referencia para procesos térmicos de calentamiento enfriamiento.
- La temperatura ambiente deberá ser determinada por la DHT22.
- Si se requiere precisión en las LM35 habría que hacer un proceso de calibración y modificar el firmware (añadiendo o substrayendo una constante calculada para la calibración).
- Para el propósito buscado son suficiente.

Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guaranteeable (at +25°C)
- Rated for full -55° to +150°C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 µA current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only $\pm\frac{1}{4}^{\circ}\text{C}$ typical
- Low impedance output, 0.1 Ω for 1 mA load



Order Number LM35CZ,
LM35CAZ or LM35DZ
See NS Package Number Z03A

Typical Applications

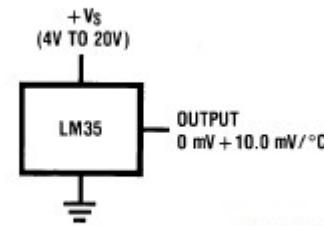


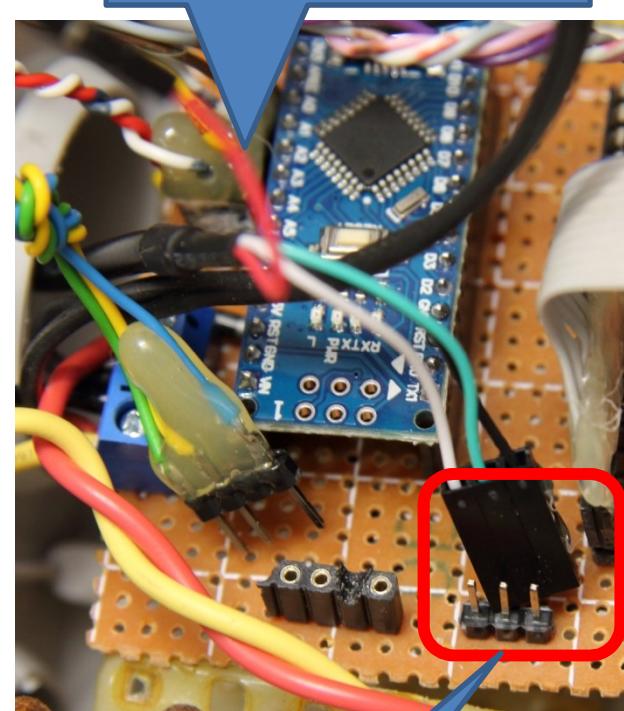
FIGURE 1. Basic Centigrade Temperature Sensor
(+2°C to +150°C)

Conexión USB-TTL

- El cablecillo se conecta al puerto del arduino:
 - Blanco -> TX1
 - Verde -> RX0
 - Negro-> GND
 - Rojo (no conectado)

(NOTA) El puerto nativo de programación del Arduino se preserva para esos menesteres, por si hay que actualizar el firmware. Por eso se ha optado por montar un cable USB-TTL complementandolo.

El cable +5v que suministra el puerto USB se deja al aire para aislar la alimentación del ordenador conectado del dispositivo.



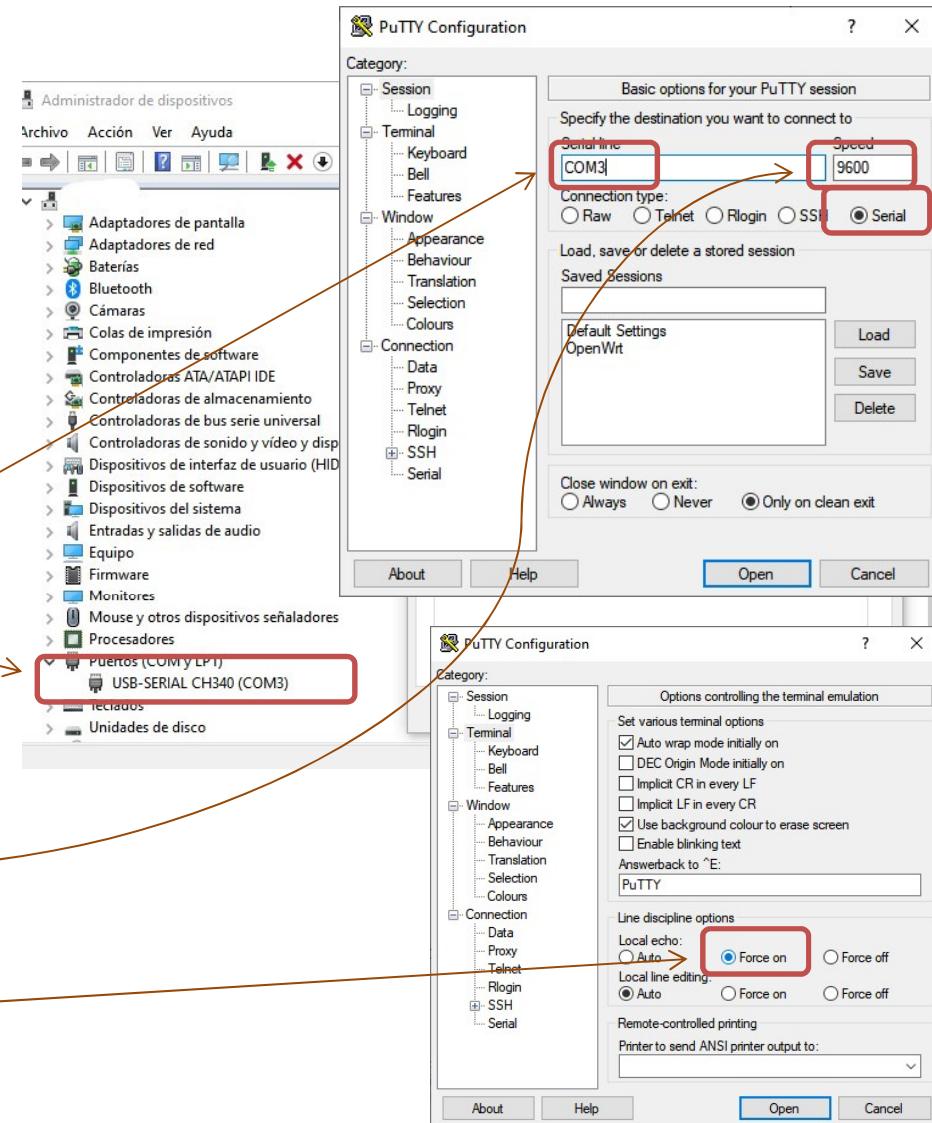
Puerto USB
TTL

CONEXIÓN Y COMANDOS

Conexión USB-TTL

Acceso al GusControl

- Al conectar el cable USB-TTL al ordenador se activará un puerto serie. Para reconocerlo habrá que acceder al “administrador de dispositivos” y consultar el puerto asignado. En este caso es el “COM3”.
- Con un emulador de terminal, por ejemplo el “Putty” se puede acceder a 9600 baudios 8 bits sin paridad y un bit de stop. Se recomienda activar el echo local.



Instrucciones, Lenguaje de control

- El arduino interpreta un leguaje regular implementado con un autómata de estados finitos que lo reconoce. A continuación se detallan las particularidades.
- Las instrucciones se van componiendo desde el estado inicial, al final que realiza la ejecución del comando.
- Al iniciarse el dispositivo responderá con “GusCtrl” y la versión de firmware, se situará en el estado inicial del autómata de reconocimiento
- El estado inicial espera a recibir un carácter ":" que es el iniciador de cualquier comando. O un "?" al cual responderá, sólo si está en este estado inicial con una pequeña ayuda mnemotécnica para recordar este lenguaje.
- En caso de duda, lo mejor es emitir un par de "?" porque el primero abortará el comando en curso y el segundo hará que aparezca la ayuda. De esta manera sabemos que se encuentra en el estado inicial.
- Se recomienda utilizar un emulador de terminal con el “echo” activado para poder ver el comando que se le emite al arduino.

Instrucciones, Inicio y Pantalla de Ayuda

- Esta pantalla es una captura del terminal virtual del simulador de circuitos. Un emulador de terminal obtiene los mismos resultados.
- A excepción de la ayuda todos los comandos tienen la siguiente estructura

:C<n>a

":" Inicio de comando

"C" comando puede ser A, T, R o P

"n" número de canal, cuando se precise

"a" acción a realizar.

Todos los comandos al terminarlos correctamente

Se responden con una cadena de interpretación

“CMD: C, Acc: a, C: n, ratio: r”

- Los canales son los disponibles para cada tipo que pueden ser nominados individualmente, o el **canal “0” referirse a todos los canales** de ese tipo y ejecutar la acción siguiente de manera global.
- Las acciones son + para activar, - para desactivar, ? Para consultar el estado y en la parte de modulación de potencia “r<nn>” indica el ratio a aplicar, donde nn es un número de 0 al 99

Ayuda Mnemotécnicos
de los comandos

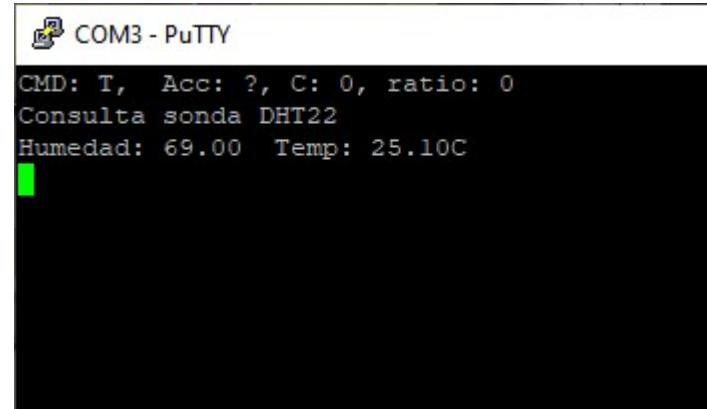
```
GusCtrl_v2.0
?
Comandos implementados:
:R[0/<n>]? :T/T? :R[<n>]<+/-?> :P[<n>]<+/-?/r0..r99>
-----
:R      -Analogico solo consulta
[0/<n>]?, 1..4 numero de entrada
:T/T?    -Temperatura y humedad solo consulta
           sin parametros
:R parte para Relés
[0/<n>] 1..8 numero de Rele
<+/-?> comando, + enciende, - apaga, ? consulta
:P parte de modulacion de potencia
[0/<n>] 1..4 numero de canal
<+/-?/r..r99> comando, + maximo, - minimo, ? consulta, r ratio
```

Comandos “T”, temperatura y humedad

:T

:T?

- Permite consultar el estado de la sonda DHT22



COM3 - PuTTY

```
CMD: T, Acc: ?, C: 0, ratio: 0
Consulta sonda DHT22
Humedad: 69.00 Temp: 25.10C
```

Es un terminal PuTTY en un windows 10, consultando el dispositivo real.
En el simulador el DHT22 no funciona, pero en real sí. ¿¿¿???

Comandos “A”, Sondas analógicas

:A?

:A<0/n>?

- Comando sólo de consulta
- **:A?** y **:AO?** Producen el mismo resultado, la consulta de todos los canales
- **:A1?** Consulta la sonda LM35 del canal 1
- **:A2?** Consulta la sonda LM35 del canal 1
- **:A3?** Consulta el valor de la entrada analógica del canal 3, codificando 0-5v a 0-1023 valores
- **:A4?** Consulta el valor de la entrada analógica del canal 4, codificando 0-5v a 0-1023 valores
- Cualquier otro valor dará un error de canal
- La ausencia de sondas dará un valor arbitrario
- En este ejemplo sólo la sonda T1 tiene un dispositivo conectado

Virtual Terminal

```
:A
-- Error en canal --
:A?
CMD: A, Acc: ?, C: 0, ratio: 0
Sonda_T(1) Temperatura :25.42
Sonda_T(2) Temperatura :395.41
Anl : 3 Valor: 624
Anl : 4 Valor: 870
:AO?
CMD: A, Acc: ?, C: 0, ratio: 0
Sonda_T(1) Temperatura :25.42
Sonda_T(2) Temperatura :395.41
Anl : 3 Valor: 624
Anl : 4 Valor: 870
```

Virtual Terminal

```
:A1?
CMD: A, Acc: ?, C: 1, ratio: 0
Sonda_T(1) Temperatura :25.42
:A2?
CMD: A, Acc: ?, C: 2, ratio: 0
Sonda_T(2) Temperatura :395.41
:A3?
CMD: A, Acc: ?, C: 3, ratio: 0
Anl : 3 Valor: 624
:A4?
CMD: A, Acc: ?, C: 4, ratio: 0
Anl : 4 Valor: 870
:A5-- Error en canal --
```

Comandos “R”, Relés

:R[0/<n>]<+/-/?>

- Comando referente a los relés,
- Se ha de indicar el número de canal o “0” para referirse a todos.
- Las acciones son:
- + activar el relé
- - desctivar el relé
- ? Consultar el estado del relé.
- Por ejemplo:
- **:R0?** Consulta el estado de todos los relés.
- **:R1+** Activa el relé 1
- **:R3-** Desactiva el relé 3
- **:R0+** Activa todos los relés
- **:R7?** Consulta el estado del relé 7
- **:R0-** Desactiva todos los relés

Virtual Terminal

```
:R0?
CMD: R, Acc: ?, C: 0, ratio: 0
Relé : 1 Estado: 0
Relé : 2 Estado: 0
Relé : 3 Estado: 0
Relé : 4 Estado: 0
Relé : 5 Estado: 0
Relé : 6 Estado: 0
Relé : 7 Estado: 0
Relé : 8 Estado: 0
:R1+
CMD: R, Acc: +, C: 1, ratio: 0
:R3-
CMD: R, Acc: -, C: 3, ratio: 0
:R0+
CMD: R, Acc: +, C: 0, ratio: 0
:R7?
CMD: R, Acc: ?, C: 7, ratio: 0
Relé : 7 Estado: 1
:R0-
CMD: R, Acc: -, C: 0, ratio: 0
```

Comandos “P”, Modulación de potencia

:P[0/<n>]<+/-/?/r0..r99>

- Comando referente a los relés,
- Se ha de indicar el número de canal o “0” para referirse a todos.
- Las acciones son:
 - + activar el canal pwm al 100% de potencia
 - - desactivar el canal pwm
 - ? Consultar el estado del canal pwm.
- r0-r99 activar el canal pwm el ratio % de potencia indicado (99 equivale al 100%)

- Por ejemplo:
 - :P1+ Activa el PWM 1 al 100% (valor 99)
 - :P3- Desactiva el PWM 3
 - :P0r35 Activa todos los PWM al 35%
 - :P0? Consulta el estado de todos los PWM
 - :P0- Desactiva todos los PWM
 - :P0? Consulta el estado de todos los PWM

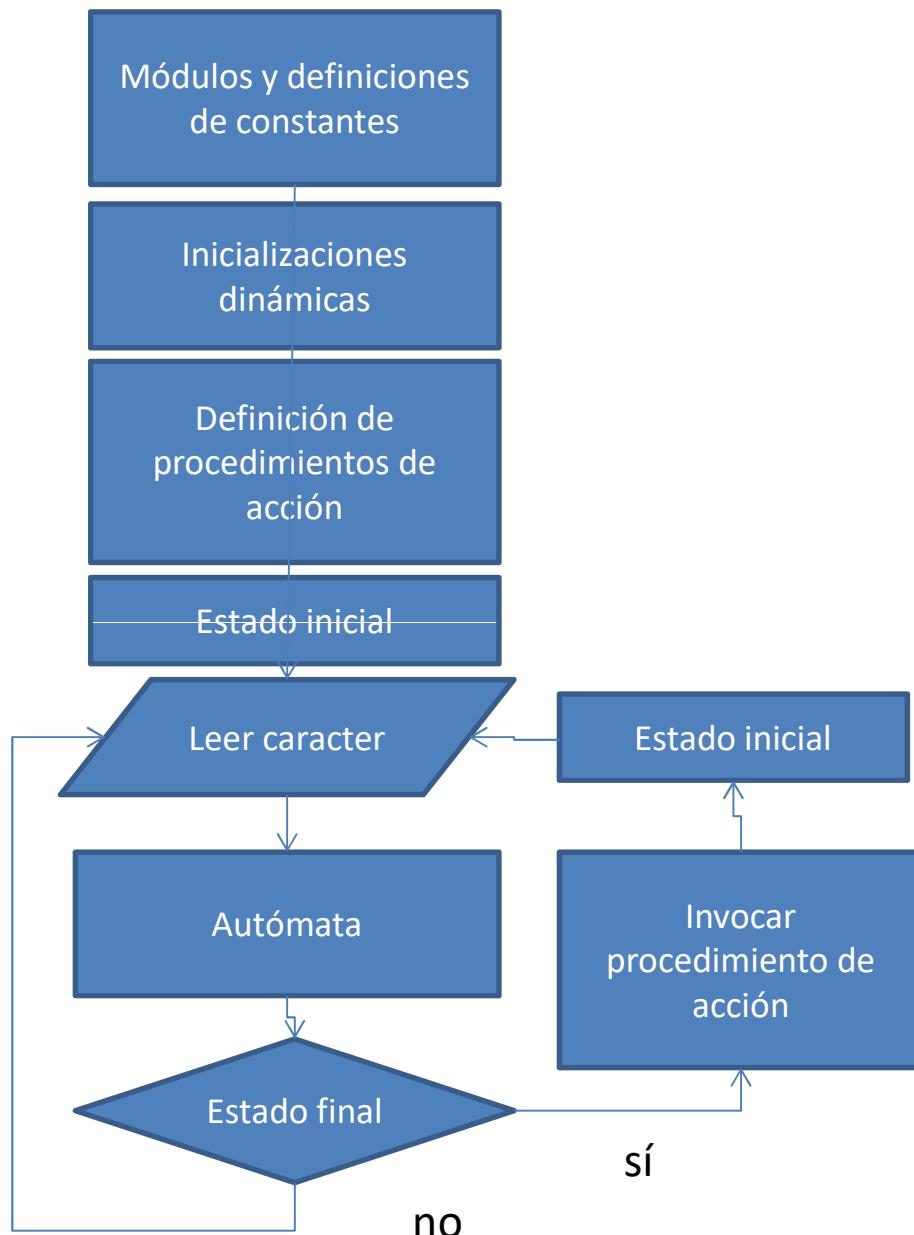
Virtual Terminal

```
:P1+
CMD: P, Acc: +, C: 1, ratio: 0
:P3-
CMD: P, Acc: -, C: 3, ratio: 0
:P0r35
CMD: P, Acc: R, C: 0, ratio: 35
:P0?
CMD: P, Acc: ?, C: 0, ratio: 0
/PWM : 1 nivel: 35
/PWM : 2 nivel: 35
/PWM : 3 nivel: 35
/PWM : 4 nivel: 35
:P0-
CMD: P, Acc: -, C: 0, ratio: 0
:P0?
CMD: P, Acc: ?, C: 0, ratio: 0
/PWM : 1 nivel: 0
/PWM : 2 nivel: 0
/PWM : 3 nivel: 0
/PWM : 4 nivel: 0
```

PROGRAMACIÓN Y CÓDIGO

El programita...

- Se desarrolla en el entorno IDE de Arduino en lenguaje “C”
- La estructura del programa es muy sencilla, es un pooling continuo del canal serie donde los caracteres que se reciben se interpretan con una rutina que implementa el autómata de estados.
- Cuando se llega al estado final, se ejecuta el comando decodificado. Para ello se pasa como parámetros a la rutina de acción esos parámetros, Comando, Acción, Canal y Ratio



El programita... (bucle principal)

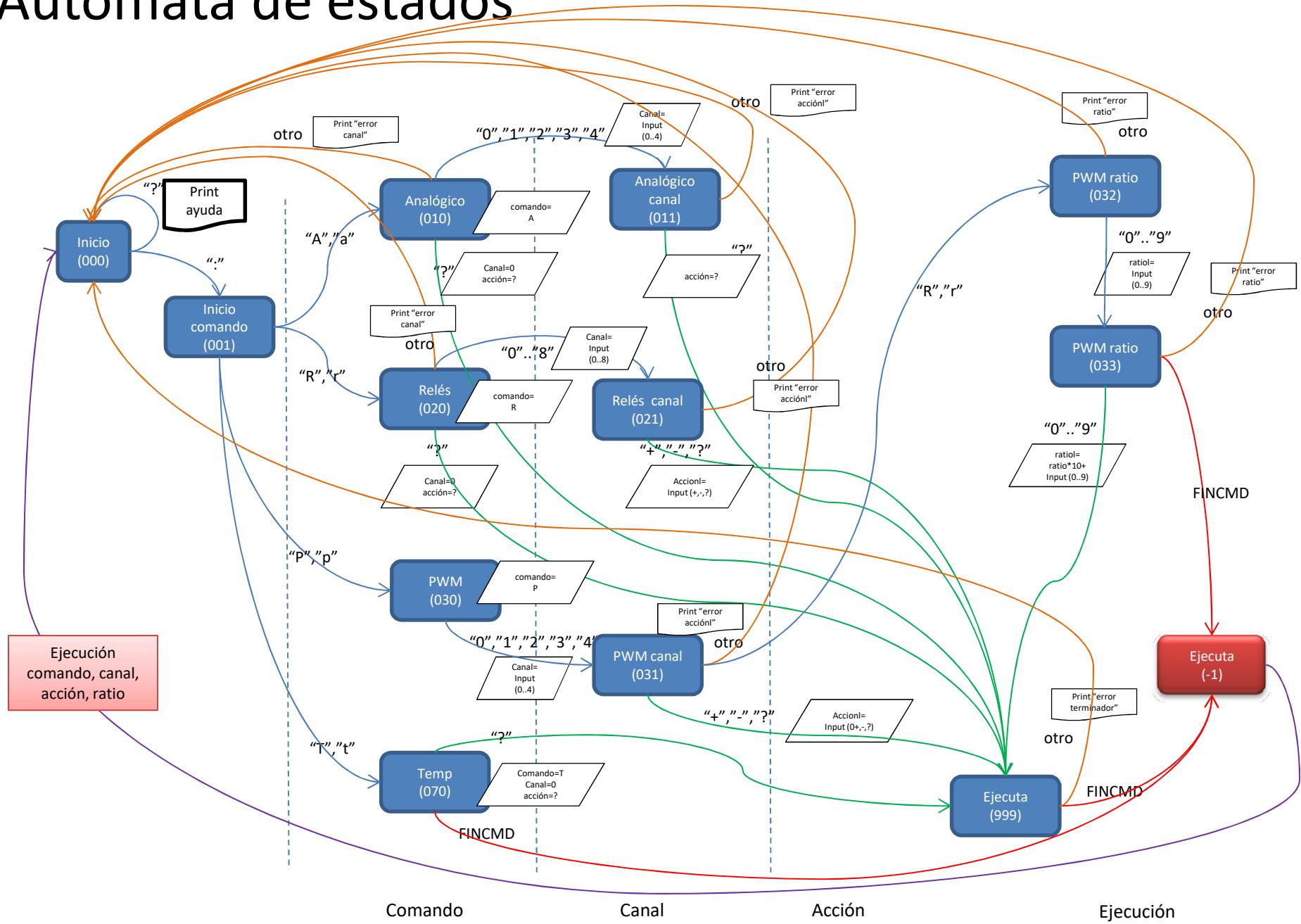
- Ah! Tiene una rutina que hace parpadear el led... de manera que sabemos si el código está en ejecución

Función que implementa el autómata

```
void loop()
{
    // put your main code here, to run repeatedly:
    delay(10);

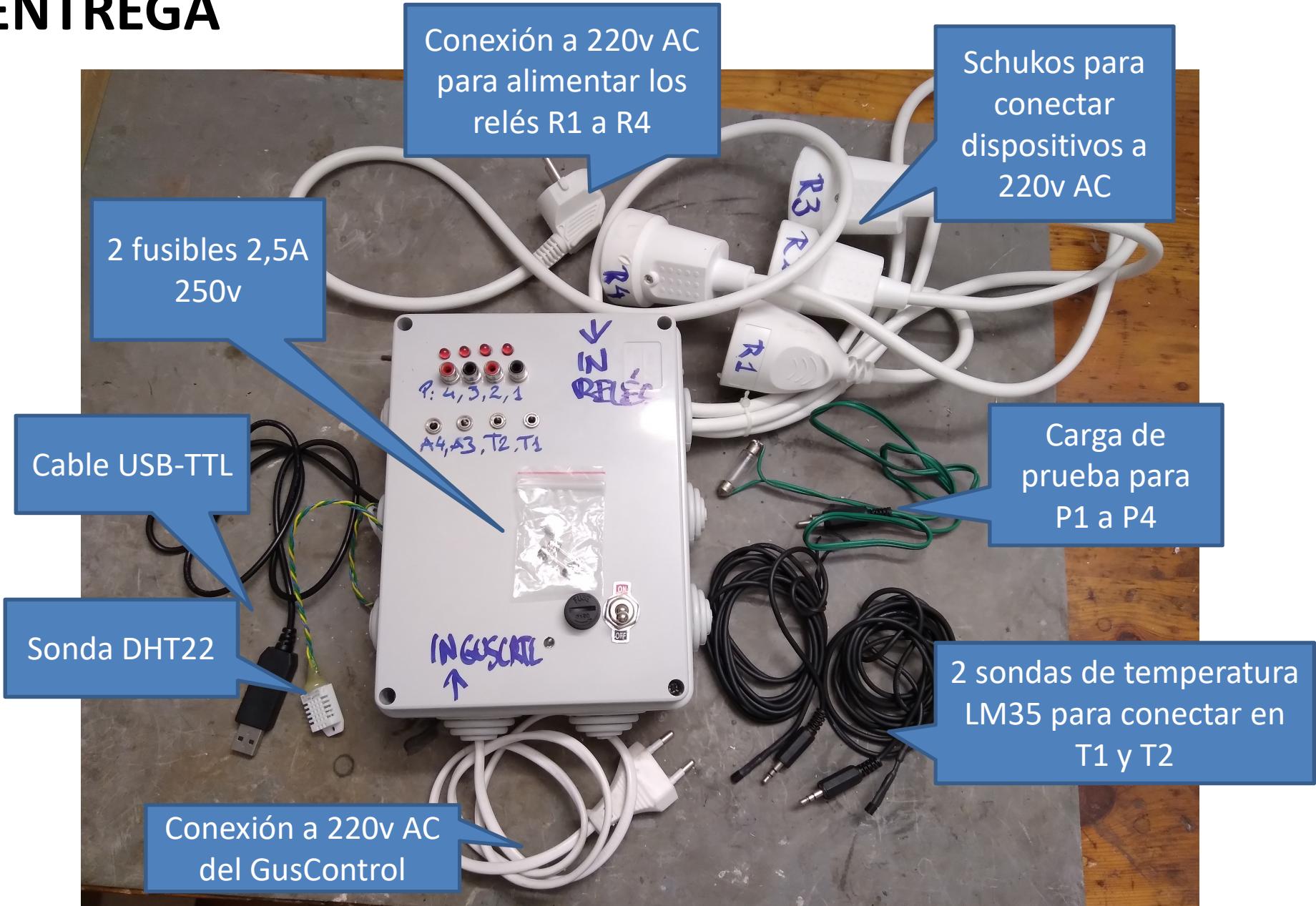
    if ( Serial.available() > 0 )
    {
        in_c= Serial.read();
        // Serial.write(in_c); //echo para poder ver lo que escribimos (hay que configurar el terminal con eco local)
        protocolo( in_c );
        if ( estado == -1 ) // el estado -1 indica ejecución del comando decodificado
        {
            actua();
            estado = 000;
        }
    }
    parpadea();
}
```

Autómata de estados



DOTACIÓN FINAL

ENTREGA



PROYECTO EN LA RED

COMPARTIDO

*Bajo licencia **GNU GPL3**, el código se puede usar, estudiar, compartir (copiar) y modificar. Se debe mencionar su origen y no puede restringirse ninguna variante posterior al uso de cualquier otra persona, deberá estar bajo la misma versión de licencia GNP GPL3.*

El recurso de internet donde encontrar el código, este manual e información adicional al proyecto es:

<https://github.com/Nandorroloco/GusControl>