

INTERFACE FRECUENCIMETRO GENERAL RADIO 1192-B BASADA EN ARDUINO

Se acopla un Arduino Mega (clon de origen chino) para generar una interface digital. El chipset USB del Arduino no es standard, por lo que hay que instalar en la PC el driver para el chipset CH341 y que aparezca como puerto Serie COM:

El programa que se carga en el Arduino espera a recibir un carácter “?” por la interface (serie en modo 9600 baud, no parity, 8 bits, 1 stop bit) y responde con:

<Frecuencia>, <Overload>, <Válido><EOL>

donde

Frecuencia es el valor medido en Hz

Overload = 0 si no sucedió un SPILL
1 si sucedió un SPILL

Válido = 0 si el Gate sigue abierto
1 si el Gate está cerrado y el display es válido

En caso que la llave selectora esté en la posición Time Interval la respuesta literal es:

TimeInterval<EOL>

El frecuencímetro General Radio 1192-B ofrece los 5 dígitos del display en formato BCD en un conector Amphenol de 50 pines en la parte trasera. También ofrece un pin (PGT) que indica si el gate está abierto (el aparato sigue contando) y, por lo tanto, el valor no es válido.

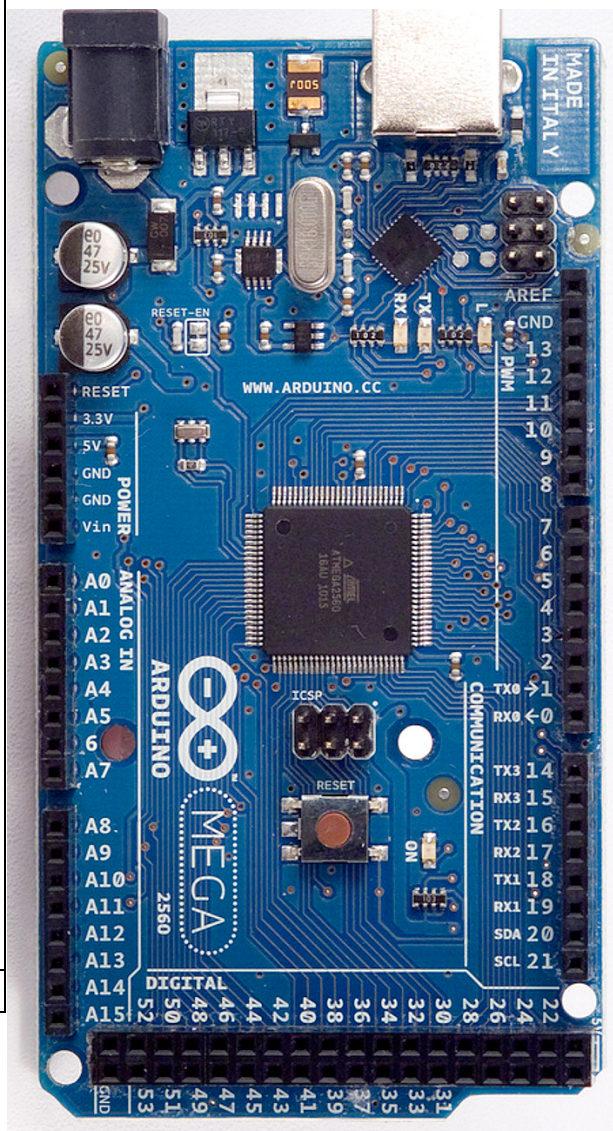
Originalmente, no ofrece una señal de Overflow (Spill). Tampoco ofrece el tiempo de conteo (Gate Interval) por lo que no se sabe la potencia de 10 por la que multiplicar el número leído.

Se adosa a la llave selectora de rango (Gate Interval) un potenciómetro alimentado con 5 V (alimentación interna del aparato) y su punto medio se saca por pin 23 del conector Amphenol. La señal analógica indicará la posición de la llave selectora.

Se saca la señal de comando de la lámpara SPILL por el pin 22 del conector Amphenol.

Se conecta a un Arduino Mega 2560 con el siguiente mapa

ARDUINO		AMPHENOL	
22	DG1-1	1	Este grupo en enchufe macho 18x2 extremo de placa Mega
23	DG1-2	2	
24	DG1-4	26	
25	DG1-8	27	
26	DG2-1	3	
27	DG2-2	4	
28	DG2-4	28	
29	DG2-8	29	
30	DG3-1	5	
31	DG3-2	6	
32	DG3-4	30	
33	DG3-8	31	
34	DG4-1	7	
35	DG4-2	8	
36	DG4-4	32	
37	DG4-8	33	
38	DG5-1	9	
39	DG5-2	10	
40	DG5-4	34	
41	DG5-8	35	
42	PGT	24	
43	SPILL	22	
GND	Ground	50	
A0	Pote Llave	23	Cable separado



PROGRAMA SUBIDO AL ARDUINO

```
const int PinLLAVE = 0; //Entrada analogica para leer posicion llave selectora
const int BASE = 22;    // Base del block de pines digitales
const int PGT = 42;     // Pin de status
const int SPILL = 43;   // Pin de Overflow (Spill)

//-----

//Se ejecuta una vez al prender o resetear la Arduino
void setup() {

    // Puerto serie a 9600 baud
    Serial.begin(9600);

    //define pines digitales como entradas
    for (int i=BASE; i < (BASE+22); i++)
        {pinMode(i, INPUT);}
}

//-----

// the loop routine runs over and over again forever:
void loop() {

    int Puerto = BASE;
    unsigned long Peso = 1;
    unsigned long Frecuencia = 0;
    unsigned long Digito = 0;

    // Lee los 5 digitos desde los pines y suma con peso decimal adecuado
    for (int i = 1; i<=5; i++)
    {
        Digito = digitalRead(Puerto);
        Digito = Digito + 2 * digitalRead(Puerto+1);
        Digito = Digito + 4 * digitalRead(Puerto+2);
        Digito = Digito + 8 * digitalRead(Puerto+3);
        Frecuencia = Frecuencia + Peso * Digito;
        Puerto = Puerto+4;
        Peso=Peso*10;
    }

    //Lee si todavia esta contando
    boolean ValidData = digitalRead(PGT);

    // Lee si sucedio overflow (spill en el frecuencimetro)
    boolean Overflow = digitalRead(SPILL);

    // Lee estado de Gate Time (llave selectora) y multiplica por el valor adecuado
    // 100 microsec = 5.015 V (1017 = lectura del A0 del Arduino)
    // 1  milisec = 4.815 V (982)
    // 10 milisec = 4.169 V (848)
```

```
// 100 milisec = 3.506 V (715)
// 1 sec = 2.909 V (596)
// 10 sec = 2.36 V (486)
// Time Interval = 1.81 V (371)
```

```
int Llave = analogRead(PinLLAVE);
```

```
// Define escala en base a posicion de llave
```

```
float Multiplicador = 10000;
```

```
if (Llave < 1000)
```

```
    {Multiplicador = 1000;}
if (Llave < 920)
```

```
    {Multiplicador = 100;}
if (Llave < 785)
```

```
    {Multiplicador = 10;}
if (Llave < 655)
```

```
    {Multiplicador = 1;}
if (Llave < 540)
```

```
    {Multiplicador = 0.1;}
if (Llave < 425)
```

```
    {Multiplicador = -1;}

// Calcula la frecuencia en Hz
float Medicion = Multiplicador * Frecuencia;
```

```
// Se fija si hay comando en interface serie y si hay lo lee
```

```
if(Serial.available() > 0) {
    char inByte = Serial.read();

    //Si pregunta por la frecuencia con un "?", envia los datos
    if (inByte=='?')
    {
        if (Multiplicador > 0)
        {
            // Manda con precision de 1 Hz excepto para 10 seg
            // de conteo, que manda con precision de 0.1Hz
            {Serial.print(Medicion, Multiplicador<1);}
            //Manda status de Overflow (spill)
            Serial.print(",");
            Serial.print(Overflow);
            //Manda status de Dato valido o todavia contando
            Serial.print(",");
            Serial.println(ValidData);
        }
        else
        // Si esta en modo de Time Interval avisa mandando el cartel
        {Serial.println("TimeInterval");}
    }
}
}
```