

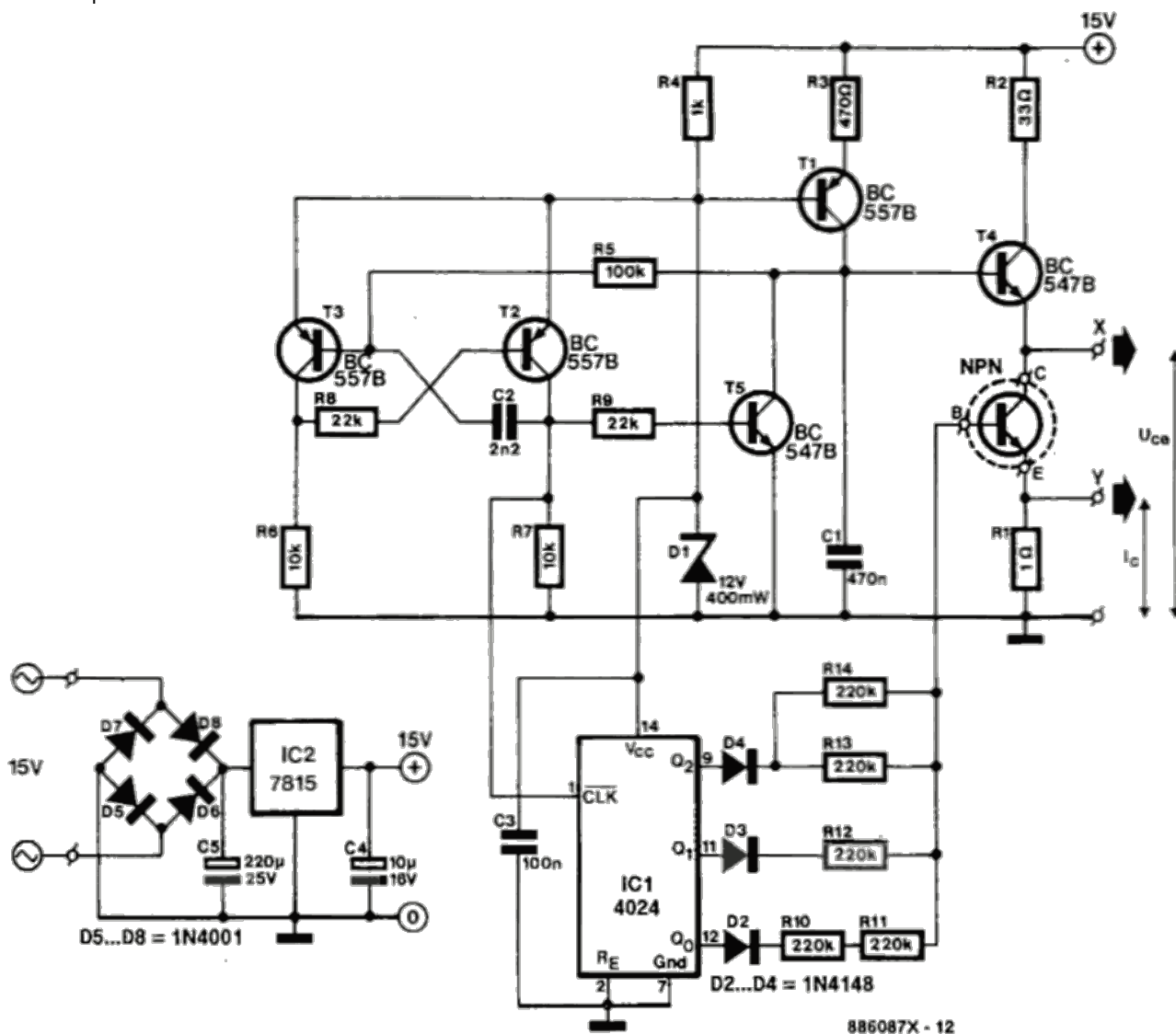
[LAB] Visualizador de las curvas del TRT



- Información obtenida de:
 - [ARDUINO -Trazador curvas de transistores \(Juan Galaz, SETA43\)](#)
 - [Trazador de curvas de Transistores. Interface gráfica con Gambas \(VB Linux\) \(Juan Galaz, SETA43\)](#)
- Otros enlaces interesantes:
 - [Gráficas de datos meteorológicos - Python - Generar página web.](#) En él puede verse como hacer gráficas con Python, en un programa de temperaturas usando la librería matplotlib.
 - [Sensores - 18B20 LDR BMP180 DHT22 - Arduino - Gambas - Python.](#) En este proyecto se indica como comunicarse el Arduino y el PC con Python.
 - [Traçador de Curvas \(INS573\)](#)

Circuito de la revista Elektor

- Publicado en la edición española de enero de 1990, número 116, páginas 1-54 y siguientes. Permite visualizar en un osciloscopio las curvas de un transistor.



Lista de componentes

Resistencias ($\pm 5\%$):

$$R_1 = 1\text{R0}$$

$$R_2 = 33\Omega$$

$$R_3 = 470\Omega$$

$$R_4 = 1\text{K0}$$

$$R_5 = 100\text{K}$$

$$R_6; R_7 = 10\text{K}$$

$$R_8; R_9 = 22\text{K}$$

$$R_{10} \dots R_{14} \text{ incl.} = 220\text{K}$$

Condensadores:

$$C_1 = 470\text{n}$$

$$C_2 = 2\text{n}2$$

$$C_3 = 100\text{n}$$

$$C_4 = 10\mu; 16\text{ V}$$

$$C_5 = 220\mu; 25\text{ V}$$

Semiconductores:

$$D_1 = \text{diodo zener } 12\text{ V}; \\ 400\text{ mW}$$

$$D_2; D_3; D_4 = 1\text{N}4148$$

$$D_5 \dots D_8 \text{ incl.} = 1\text{N}4001$$

$$IC_1 = 4024$$

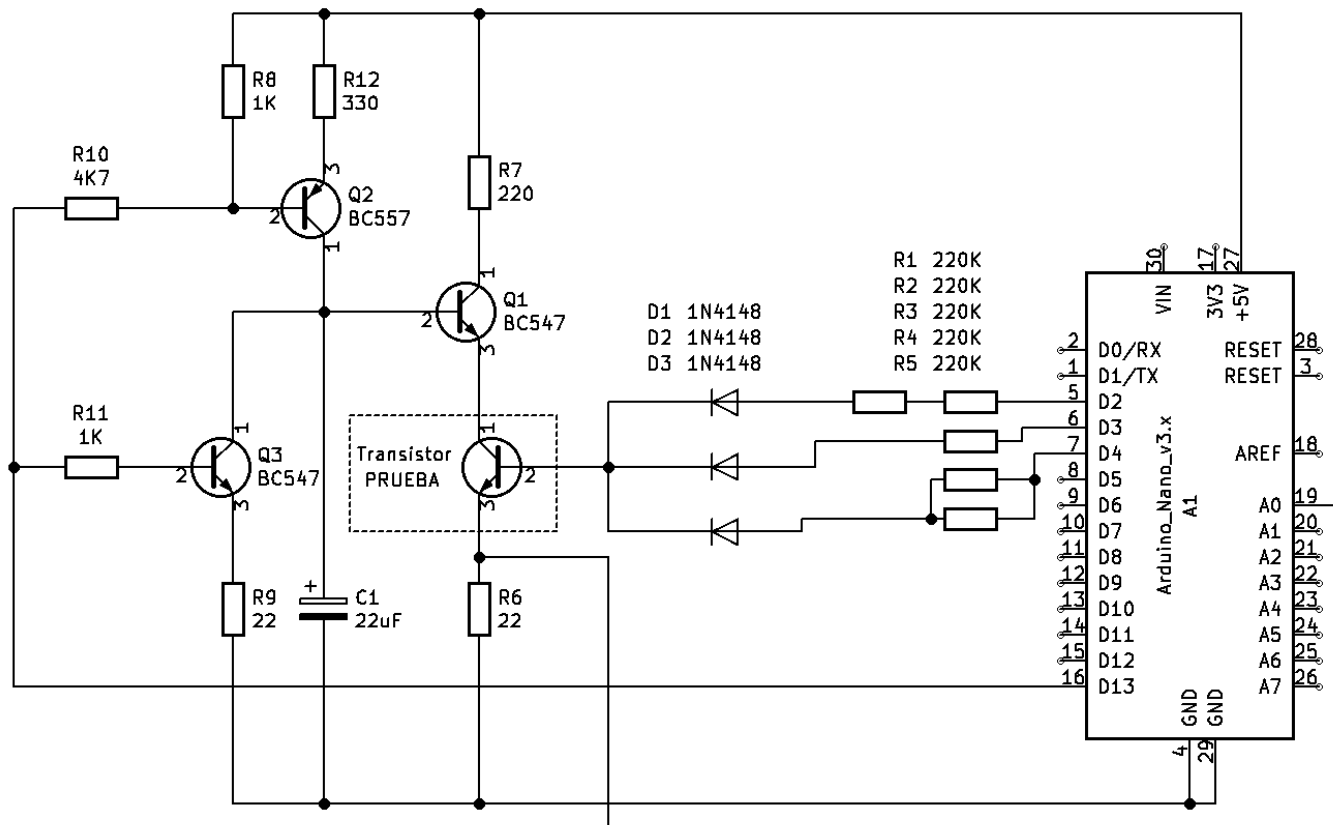
$$IC_2 = 7815$$

$$T_1; T_2; T_3 = \text{BC}557\text{B}$$

$$T_4; T_5 = \text{BC}547\text{B}$$

- Otro mucho más complejo es el aparecido en la edición inglesa de Elektor, año 2009, número 386 (febrero), páginas 24 y siguientes.

Circuito de Juan Galaz



Funcionamiento

- Se ha adaptado el circuito para poder capturar los datos con un Arduino Nano.
- El transistor Q2 produce una corriente constante que hace cargar linealmente el condensador C1, una señal de diente de sierra lineal.
- Esta señal que toma valores desde 0 a 5 V, se aplica al colector del transistor a probar mediante el transistor Q1.
- Para descargar la carga del condensador se utiliza el transistor Q3.
- La carga y descarga del condensador se realiza con el Pin D13 del arduino.
- Para generar las diferentes intensidades que se le aplica a la base del transistor de prueba, utilizamos los Pin D2-D3-D4 del arduino.
- Con las diferentes combinaciones de los Pin D2-D3-D4 podemos mediante resistencias y diodos aplicar una intensidad al transistor de prueba. Al ser 3 bits las combinaciones el número de curvas son 8.
- La intensidad se mide mediante la resistencia R6, y es capturado con el Pin A0.

Programa del Arduino

[transistor3.ino](#)

```

//Curvas de transistores
//xsetaseta@gmail.com
// 26/08/2020

void setup()
{
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(13, OUTPUT);

  Serial.begin(57600);

```

```
Serial.println("CURVAS");

}

void loop()
{
  int z,z1;
  int v0;
  int mA;
  long matrix[8][10];

  for(z=0;z<8;z++)
  {
    if(z&1)
      digitalWrite(2, 1);
    else
      digitalWrite(2, 0);

    if(z&2)
      digitalWrite(3, 1);
    else
      digitalWrite(3, 0);

    if(z&4)
      digitalWrite(4, 1);
    else
      digitalWrite(4, 0);

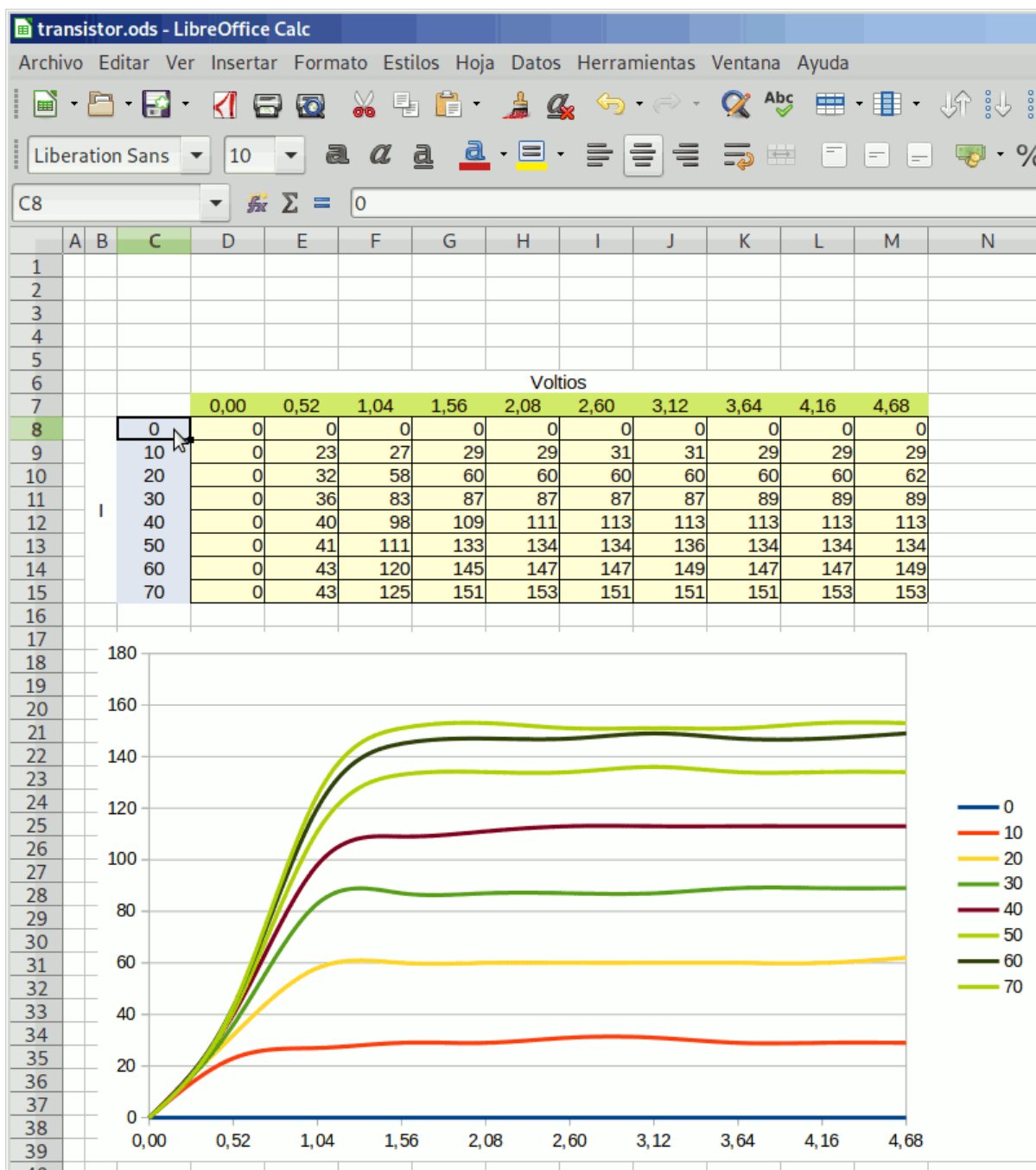
    digitalWrite(13, 1);
    delay(100);
    digitalWrite(13, 0);
    for(z1=0;z1<10;z1++)
    {
      delay(15);
      matrix[z][z1]=analogRead(A0);
    }
  }

  mA=0;

  for(z=0;z<8;z++)
  {
    Serial.print(mA);Serial.print(",");
    for(z1=0;z1<10;z1++)
    {
      Serial.print( (186*matrix[z][z1])/102); Serial.print(",");
    }
    mA+=10;
    Serial.println("");
  }
}
```

En la parte de programación en el Arduino el funcionamiento es el siguiente:

- El Pin 13 se pone a 1, con lo que conseguimos dos cosas, descargar el condensador C1 y anular el generador de intensidad constante.
- El pin 13 se pone a 0, anulamos la puesta a cero del condensador, y activamos la carga linealmente el condensador C1.

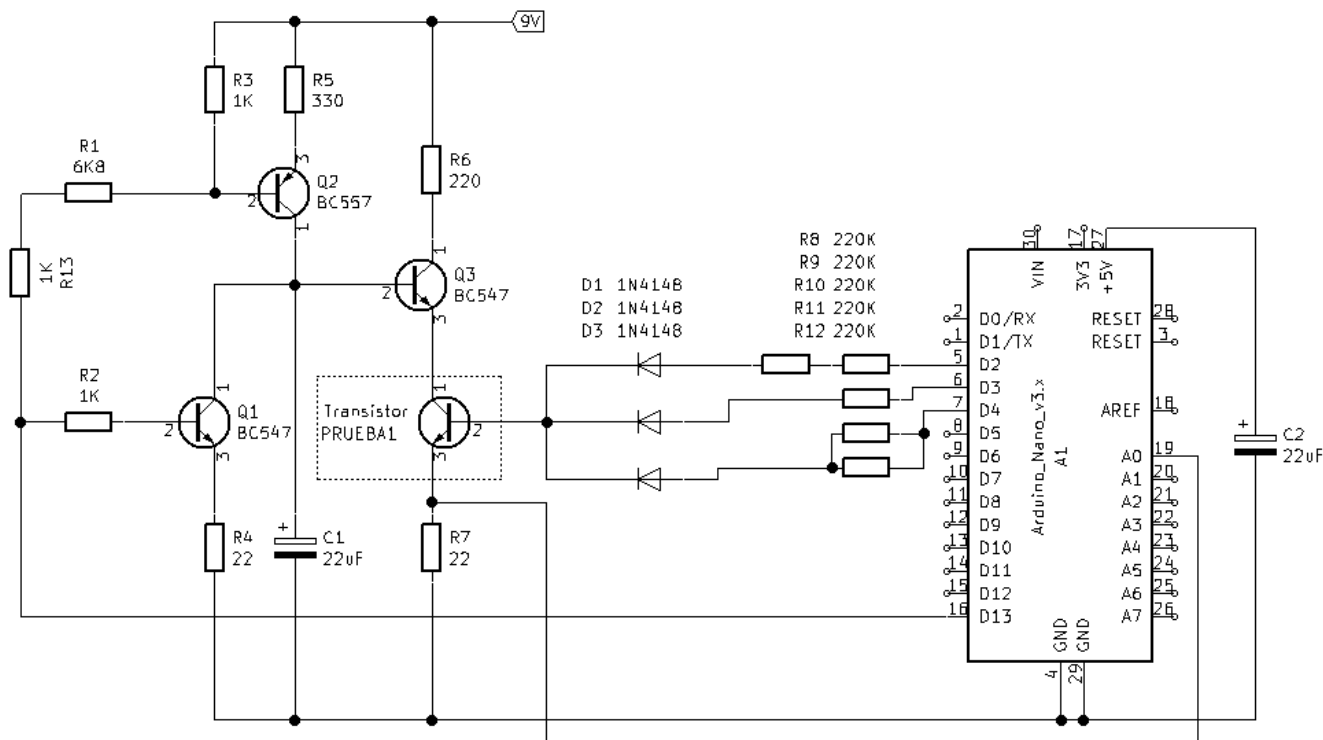


Conclusiones del autor del artículo

- Después de muchas medidas con diferentes transistores y con iguales modelos pero de diferentes fabricantes he observado que:
 - Aunque sea el mismo tipo (BC547), las curvas son muy diferentes entre los diferentes fabricantes.
 - Hay transistores bastante especiales, 2SD2375 con una ganancia enorme.
 - Los Darlington tienen otro tipo de curvas, y su tensión B-E es de 1.4V.
- Mi conclusión es que se fabrican lotes de transistores comunes, y dependiendo de los resultados en los test, los numeran con un número que se adapte a las tablas standard. Eso creo que pasaba en el pasado, ahora creo que son más precisos a la hora de su fabricación.
- Los transistores no funcionan igual con tensiones más altas de 5V, pero no he querido complicar el circuito para poder alimentarlo a partir del Arduino.
- También se podría hacer que la entrada A0 digitalizase más rápido además de con más ganancia, pero he querido usar técnicas normales del Arduino.
- Otra forma de medir la señal sería medir en la resistencia R7 e invertir la señal, pero he creído no necesario

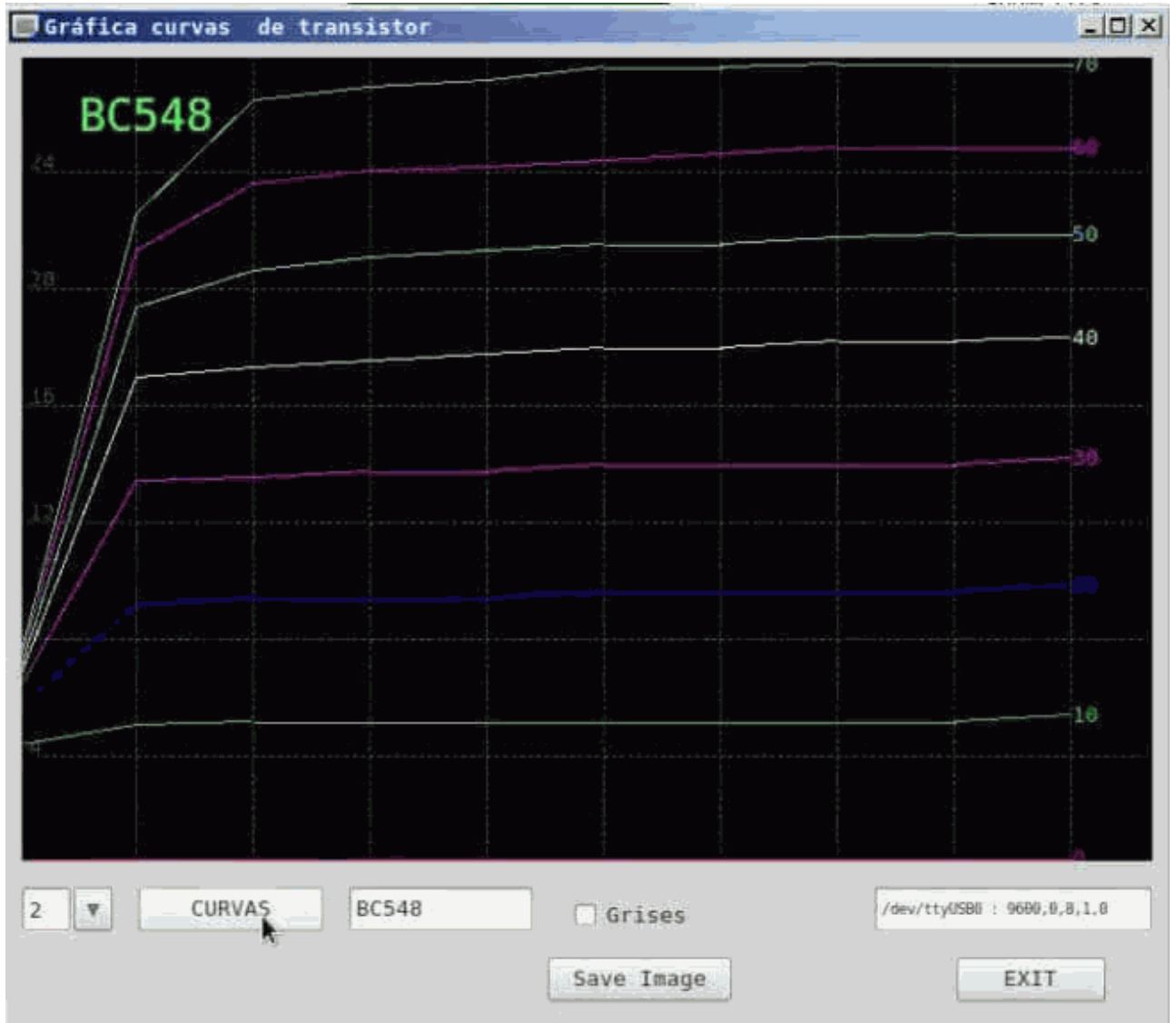
complicar las cosas. Es un sencillo montaje que permite observar el funcionamiento de los diferentes transistores, no se pretende mucho más, no se quiere hacer un instrumento de medida exacto.

Circuito de Juan Galaz usando interfaz gráfica con Gambas (VB Linux)

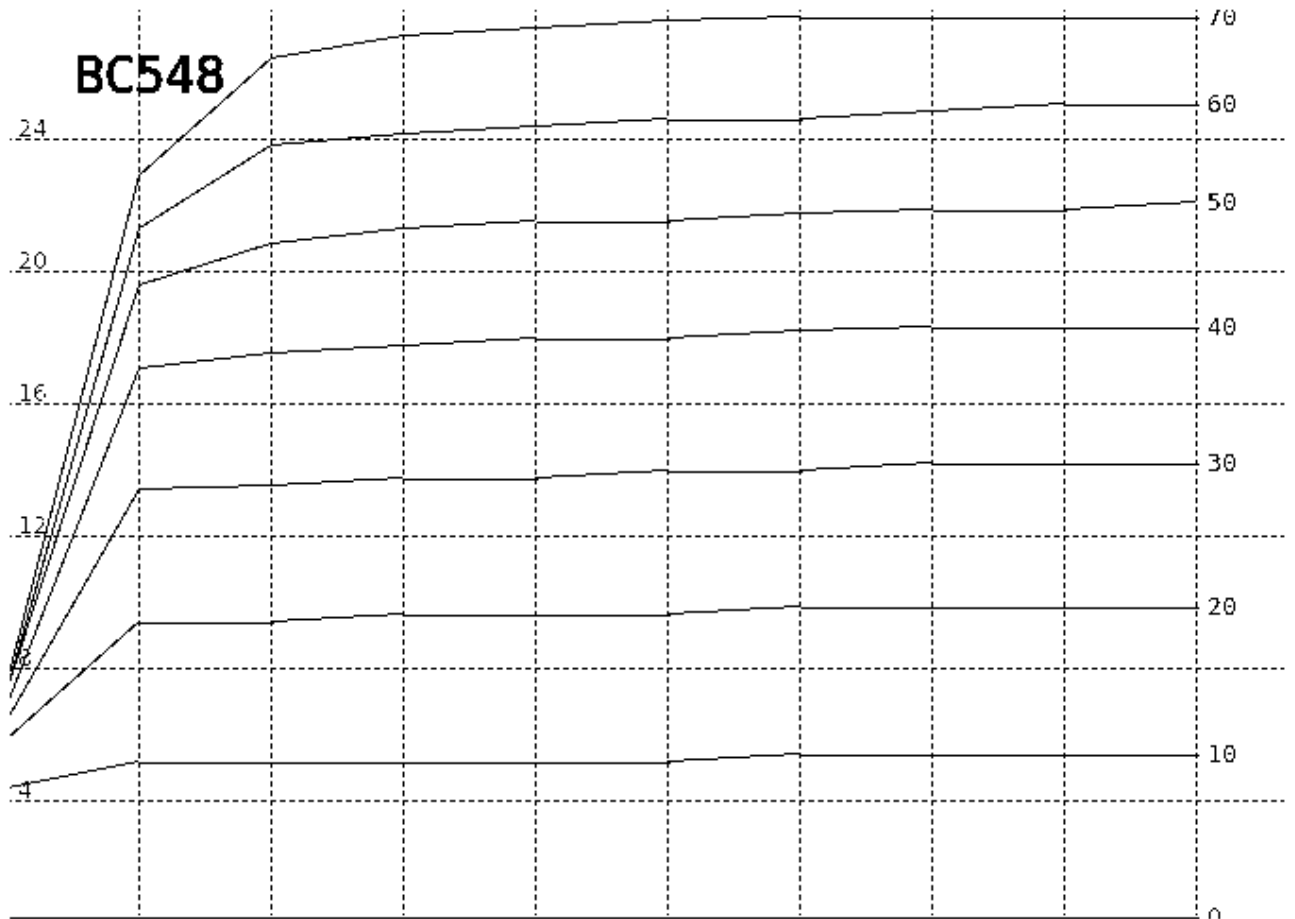


Funcionamiento

- El circuito es similar al anterior realizado pero en este caso la alimentación es de 9V, que se ha obtenido de un módulo convertidor.
- En esta ocasión los datos se importan automáticamente mediante un programa realizado en Gambas (VB para Linux). Si se quiere la gráfica de curvas se puede exportar a un fichero de formato png.
- Programa realizado en Gambas (VB para Linux):



- Imagen exportada en formato png , blanco y negro:



Programas

- Todo comprimido: [tragra.7z](#)
- Programa del Arduino:

[transistor4.ino](#)

```
//Curvas de transistores
//xsetaseta@gmail.com
//17/10/2020
//http: / / seta43.duckdns.org
//http: / / seta43.blogspot.com /

char clave[] = "TRANSIS";
#define MAXENTRADA 50
char buffer[MAXENTRADA];
int contador=0;
char stringComplete = false;

void setup()
{
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(13, OUTPUT);

  Serial.begin(9600);
  for (char zx = 0; zx < 3; zx++) { Serial.println(clave); delay(200); }
```

```
}

void loop()
{
  while (Serial.available())
  {
    char inChar = (char)Serial.read();
    buffer[contador]=inChar;
    if (inChar == '\n')
    {
      stringComplete = true;
      buffer[contador]=0;
    }
    if(contador<MAXENTRADA)contador++;
  }
  if (stringComplete)
  {
    Orden(buffer);
    contador=0;
    stringComplete = false;
  }
}

void runFunciones(void)
{
  if(strstr(buffer,clave) )
  {
    Serial.println("OK");
    goto salir0;
  }

  if(strstr(buffer,"CURVAS"))
  {
    curvas();
    goto salir0;
  }
  Serial.println("aERROR ");
salir0:
;
}

void Orden(char *bufferS)
{
  runFunciones();
}

void curvas(void)
{
  int z,z1;
  int v0;
  int mA;
  long matrix[8][10];

  for(z=0;z<8;z++)
  {
    if(z&1)
```

```
        digitalWrite(2, 1);
        else
        digitalWrite(2, 0);

    if(z&2)
        digitalWrite(3, 1);
        else
        digitalWrite(3, 0);

    if(z&4)
        digitalWrite(4, 1);
        else
        digitalWrite(4, 0);

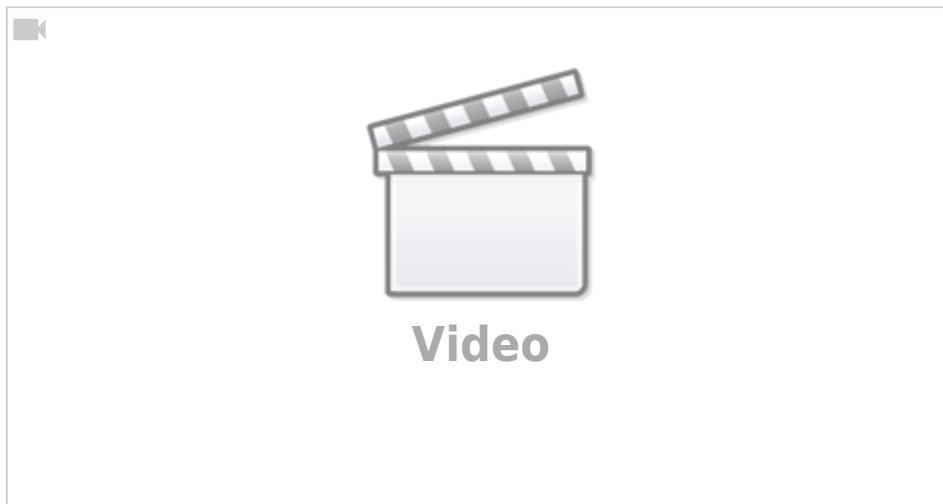
    digitalWrite(13, 1);
    delay(100);
    digitalWrite(13, 0);
    for(z1=0;z1<10;z1++)
    {
        delay(7);
        matrix[z][z1]=analogRead(A0);
    }
}

mA=0;

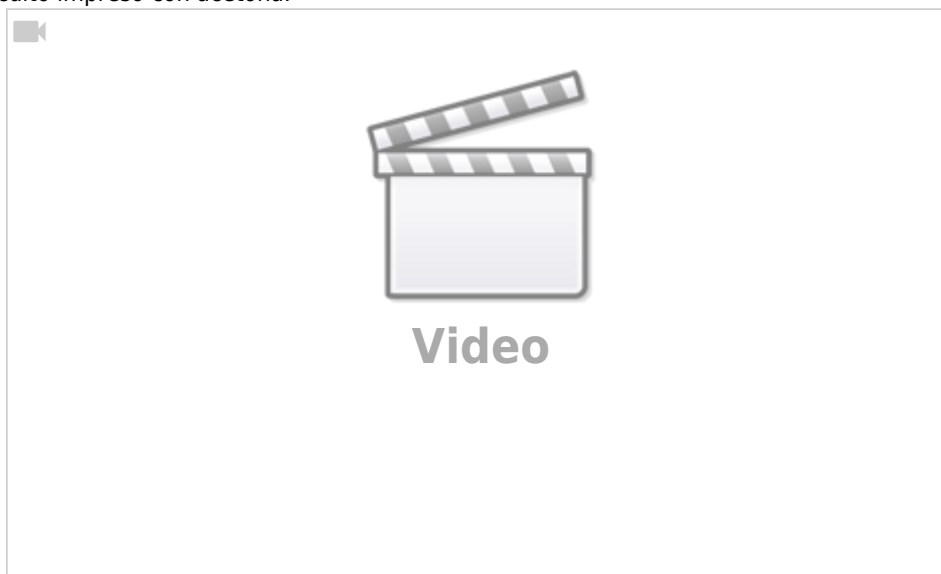
for(z=0;z<8;z++)
{
    Serial.print(mA);Serial.print(",");
    for(z1=0;z1<10;z1++)
    {
        Serial.print( (186*matrix[z][z1])/102); Serial.print(",");
    }
    mA+=10;
    Serial.println("");
}
}
```

Vídeos

- Trazador de curvas:



- Realización circuito impreso con acetona:



From:
<https://ww.euloxio.myds.me/dokuwiki/> - **Euloxio wiki**

Permanent link:
https://ww.euloxio.myds.me/dokuwiki/doku.php/doc:tec:lab:curvas_trt:inicio?rev=1779877943

Last update: **2026/05/27 12:32**

