

Diseño de un entrenador de cálculo de circuitos electrónicos para la Tecnología de la ESO con Labview.

Cálculo de circuitos con transistores en emisor común

Autor: Buisán Carrasco, Ismael (Ingeniero agrónomo, Profesor de Tecnología y Director de IES).

Público: Profesorado de Tecnología de la ESO y el Bachillerato. **Materia:** Tecnología. **Idioma:** Español.

Título: Diseño de un entrenador de cálculo de circuitos electrónicos para la Tecnología de la ESO con Labview. Cálculo de circuitos con transistores en emisor común.

Resumen

Explicación detallada de la construcción de circuitos electrónicos con transistores realizado con la el programa Labview que permite: 1.- La realización de ejercicios de cálculo de circuitos multiejecución, modificando los valores de las entradas. 2.- Posibilidad de autocorrección por parte del alumnado, ya que está programado el circuito para que deje pasar al cálculo siguiente, sólo si el anterior es correcto. 3.- Mediante un sistema de puertas lógicas, el alumno sabrá si los cálculos hechos son los correctos. Aviso mediante LEDs 4.- Aumentar el grado de confianza del alumnado en el cálculo de estos circuitos.

Palabras clave: Transistores, Cálculo, Multiejecución, Autocorrección, Electrónica.

Title: Design of a calculation electronic circuit trainer for Technology in Secondary Schools with Labview. Calculation of common emitter transistor circuits.

Abstract

Detailed explanation of the construction of electronic circuits with transistors made with the Labview program that allows: 1.- The realization of calculation exercises of multi-execution circuits, modifying the values of the entries. 2.- Possibility of self-correction by the students, since the circuit is programmed to allow the next calculation to pass, only if the previous one is correct. 3.- Through a system of logical gates, the student will know if the calculations made are correct by means of LED warnings. 4.- Increase the degree of confidence of the students in the calculation of these circuits.

Keywords: Transistors, Calculation, Multiexecution, Self-correction, Electronics.

Recibido 2018-11-14; Aceptado 2018-11-20; Publicado 2018-12-25; Código PD: 102066

1. RESUMEN

El área de Tecnología en los centros educativos debe tener una concepción eminentemente interactiva y manipulativa; La demanda de materiales didácticos que conecten con los intereses del alumnado es prácticamente una imposición, y ello conlleva el desarrollo de experiencias de cálculo a través de interfaces atractivas y de carácter recursivo.

Lo manifestado anteriormente, junto con la necesidad de desarrollo del currículo de Tecnología en lo referente a automatización de procesos, control por ordenador de los proyectos de Tecnología de de la ESO etc, es lo que me ha llevado a la elaboración de estas aplicaciones de cálculo de circuitos eléctricos recursivos, de gran aceptación por parte de mi alumnado, y que contribuyen a la consolidación de los contenidos del área.

Entre los objetivos que se persiguen son de destacar los siguientes:

1. Elaboración de actividades relacionadas con la "Electrónica". Garantizar la recursividad a través de la modificación de los datos de entrada. El programa dejará progresar al paso siguiente en función de que el anterior sea correcto, a través de funciones lógicas.
2. Trabajar aspectos de programación de procesos matemáticos mediante la recursividad ya mencionada.
3. Elaboración de interfaces de programación atractivas y no exentas de creatividad.

La didáctica de las diferentes áreas de la ESO debe ser revisada de forma continua, y ello es una exigencia en la Tecnología, ya que propicia un acercamiento a la realidad tecnológica circundante al alumnado desde una perspectiva práctica.

2. EL PROGRAMA

2.1 Consideraciones Generales:

Labview es un software diseñado por National Instruments que está especialmente dirigido a procesos de adquisición y automatización de procesos, y que resulta muy útil en el campo de la elaboración de materiales informáticos atrayentes para el alumnado, en este caso para el cálculo matemático de circuitos con transistores. Ésta última es la vertiente que pretendo desarrollar como forma de que sirva de instrumento de trabajo tanto a alumnos/as como al profesorado en su tarea docente.

La versión utilizada para la elaboración de esta actividad ha sido “Labview 7 Express”, (seguro que existen ya versiones más actuales).

2.2 Breves consideraciones sobre el programa:

La forma en que el programa trabaja es mediante la utilización de dos pantallas que puedes tener siempre a la vista, a izquierda y a derecha:

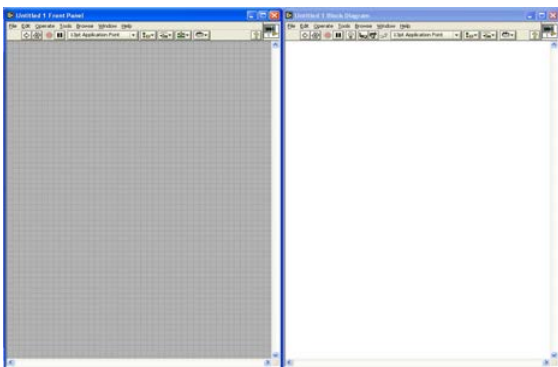


Fig.1 Pantallas de trabajo en Labview (Fuente Labview)

La pantalla gris se denomina “Front Panel”, y es donde desarrollaremos la interfaz.

La pantalla blanca se denomina “Block Diagram”, y es donde desarrollaremos la programación.

2.3 Barras de trabajo:

En “Front panel”, “Controls”, entradas(controles) y salidas(indicadores) de nuestro sistema. Sirve para dar la apariencia a la interfaz.

En “Block Diagram”, “Functions”, herramientas de unión de variables de entrada y salida, funciones lógicas etc.

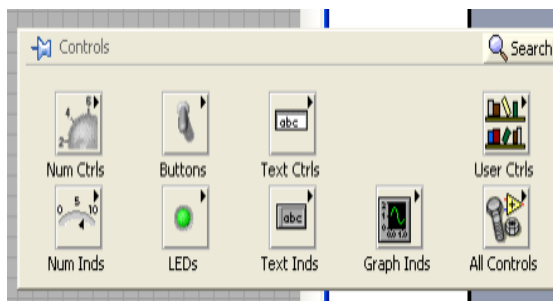


Fig.2 Controls (Barra de tarea en Front Panel)(Fuente Labview)

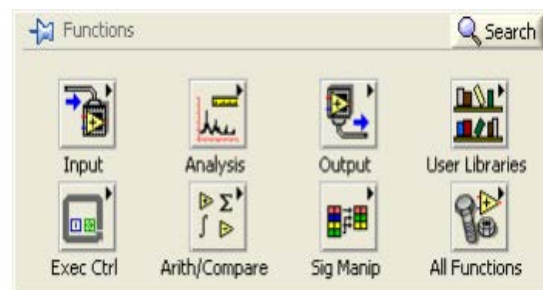


Fig.3 Functions.(Barra de tarea en Block Diagram).(Fuente Labview)

3. LA APLICACIÓN.

3.1 Objetivo:

Construcción de un circuito electrónico con un transistor con polarización en Emisor Común, en el que el alumnado situará los resultados, previamente calculados en la libreta, en los respectivos “controles (entradas)” y el programa especificará si ese resultado es correcto. Una vez se constate que cada paso dado es correcto, se procederá de la manera explicada con el siguiente paso.

Objetivo didáctico: Que el alumnado sea capaz de auto proponerse ejercicios distintos, modificando las entradas del sistema, y corregirlos de manera autónoma.

En este caso se trata de la resolución de un **circuito electrónico** de un transistor polarizado en **emisor común**, orientado a 4º de la ESO o al Bachiller, y en el que lo que se pide es lo siguiente:

EL CIRCUITO REPRESENTA UN TRANSISTOR EN EMISOR COMÚN, LA RESISTENCIA DE LA BASE DEL TRANSISTOR NPN Y LA RESISTENCIA DEL COLECTOR, ADEMÁS DE LAS FUENTES DE ALIMENTACIÓN DE BASE Y COLECTOR. DADOS LOS DATOS QUE SE MUESTRAN EN LA FIGURA, CALCULA LO QUE SE PIDE A CONTINUACIÓN:

1. LA INTENSIDAD DE BASE. (I_b)
2. LA INTENSIDAD DE COLECTOR. (I_c)
3. LA TENSIÓN COLECTOR-EMISOR (V_{CE}).
4. POTENCIA DISIPADA EN EL COLECTOR. (P_c)

3.2 Sobre cómo hacer la interfaz.

PROPUESTA DE INTERFAZ

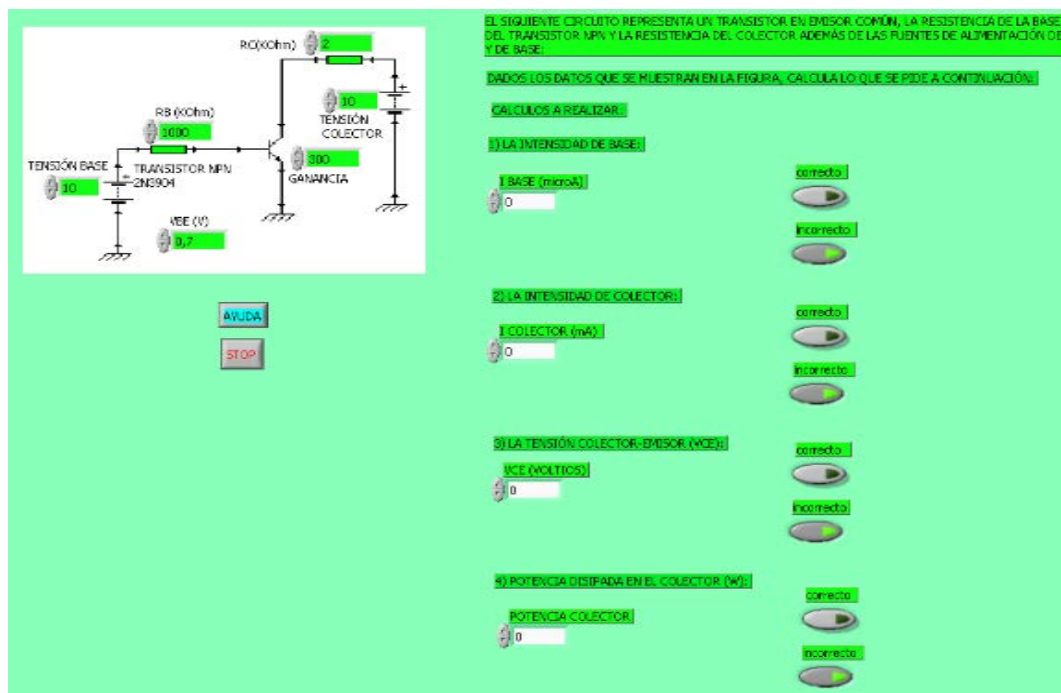


Fig.4 . Interfaz del programa que se explicará a continuación(fuente Labview).

3.2.1 Diseño del circuito:

Nos apoyaremos en cualquier programa de electricidad para estudiantes de los que corrientemente utilizamos en Tecnología. En este caso, el utilizado es el “Crocodile Clips”. El circuito a utilizar es el siguiente:

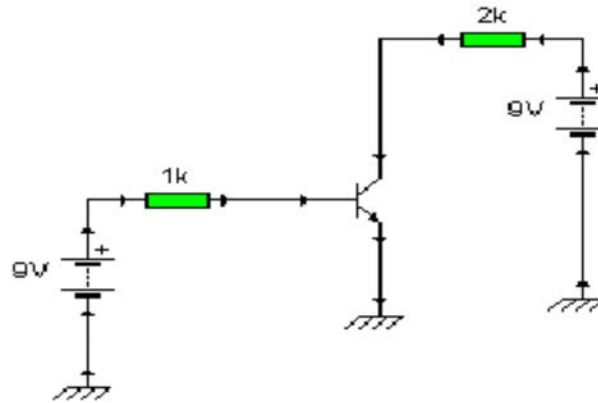


Fig. 5 Diseño del circuito base.(Crocodile Clips)

Los valores que muestra el programa serán ocultos por los controles e indicadores de Labview. Con cualquier gestor de imágenes, recortaremos la imagen del circuito y la pegaremos en Front Panel.

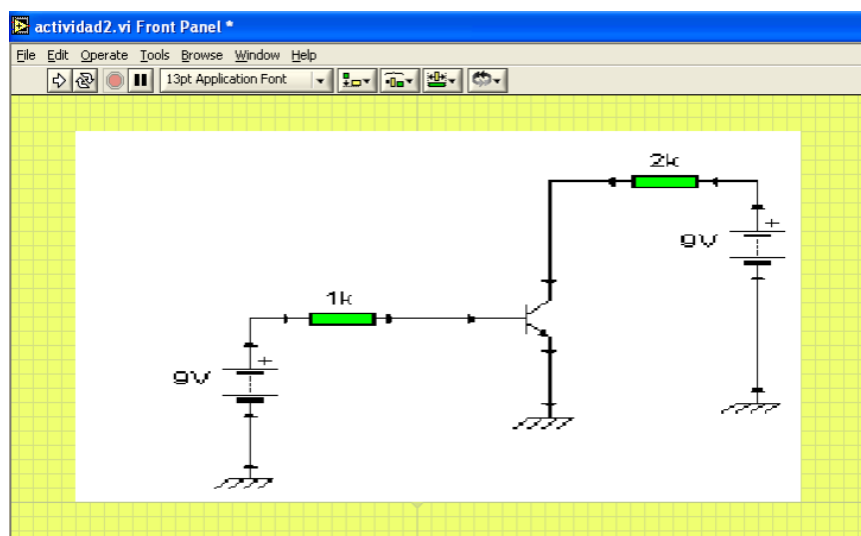


Fig. 6 Inserción del circuito en Front Panel.(fuente Labview) (color de la pantalla modificable)

3.2.2 Sobre cómo introducir controles (datos - entradas) en el circuito:

La aplicación que pretendemos diseñar está basada en la existencia de variables de **entrada** o **datos**, llamada por el programa “**Controls**”, así como en variables de **salida**, que son los resultados que pretendemos ofrecer, y que se denominan “**Indicators**”. Lógicamente, la programación está basada en la unión de unas y de otras mediante las correspondientes **operaciones matemáticas** que lleva aparejado el cálculo, en este caso, la resolución de un circuito con un transistor en **Emisor Común**.

Si pinchamos con el botón derecho del ratón sobre la pantalla de **Front Panel**, veremos que se despliega el cuadro de **“Controls”**. Del cuadro de opciones buscaremos **“Num Ctrls”** y dentro de ésta, arrastraremos la opción **“Num Ctrl.”** a **Front Panel**.

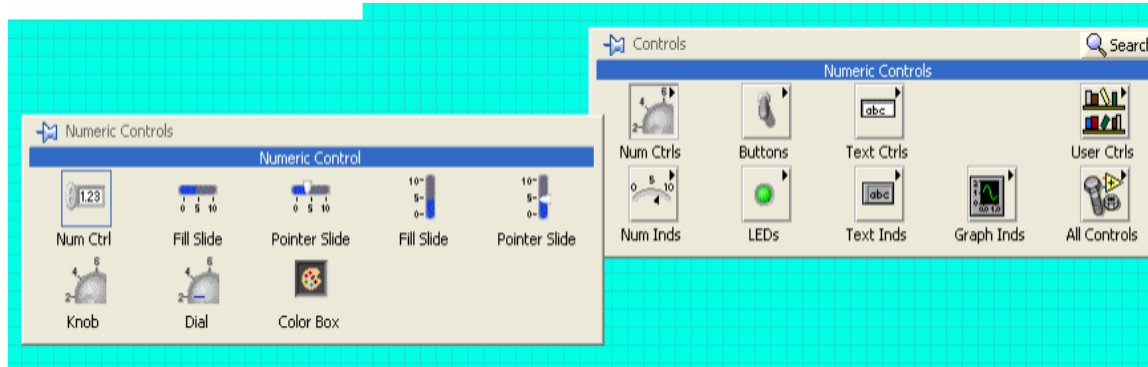


Fig.7 Apariencia de la barra de Controls en Front Panel (Fuente Labview)

Cuando arrastremos aparecerá lo siguiente tanto en **Front Panel** como en **Block Diagram**:

Front Panel	Block Diagram

Ambos símbolos corresponden al mismo elemento; En este caso a un **control** en la **interfaz** y su equivalente en la pantalla de **Programación**. Se recuerda que para pasar de una pantalla a otra hay que pulsar la combinación (**Ctrl. + E**).

Para cambiar la denominación del control lo que debemos hacer es ir a la pestaña **“Window”**, y pulsar **“Tools Palette”**. Pincharemos en la letra **A** mayúscula que está en el menú y podemos ir a donde pone **Numeric**, subrayarlo y cambiarlo por el control oportuno.

En este caso le cambiamos el nombre por el de **Resistencia de Base**, quedando de la siguiente manera:

--	--

Situamos el **Control (Rb)** en su posición definitiva en **Front Panel**, sin más que arrastrar éste al lugar correspondiente sobre el dibujo:

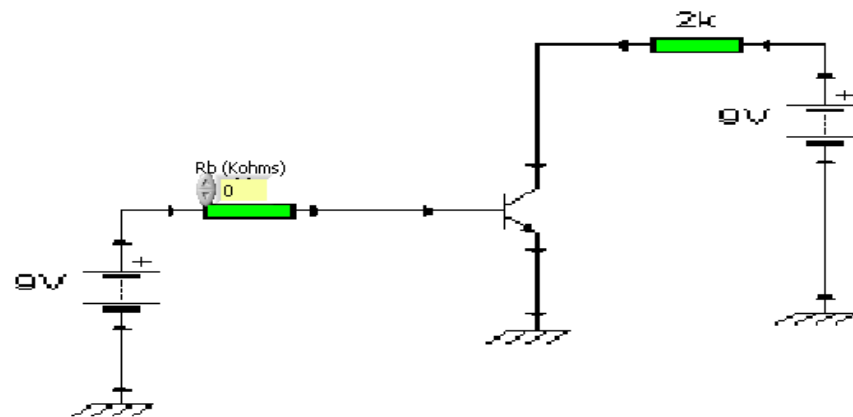


Fig.8. Situación de datos (controles) sobre el dibujo, en Front Panel (Fuente Labview)

Procederemos de igual manera con los demás controles (entradas) correspondientes a las cuatro resistencias que componen el ejercicio:

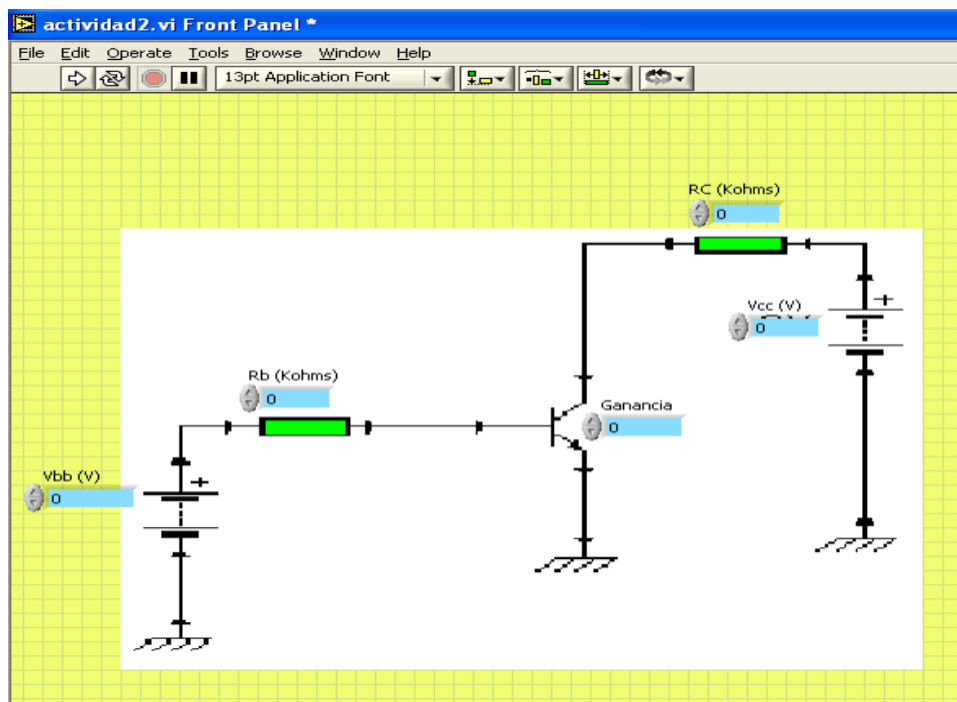


Fig.9 Apariencia final de los controles en Front Panel (Datos del Problema modificables) Labview

Se puede observar cómo a la izquierda de cada control aparece la correspondiente barra para modificar el valor del mismo.

3.3 Sobre el diseño de la actividad:

Del análisis del enunciado, que habremos escrito con Tools Palette + A, vamos a calcular **I_b** (Intensidad de Base) en μA , **I_c** (Intensidad de Colector) en mA, **VCE** (tensión colector – emisor) en V y la **P_c** (Potencia de colector) en W.

La novedad del ejercicio radica en que en este caso se introducirán los valores previamente calculados por el alumnado y el programa nos debe devolver si el valor introducido es correcto.

Para obtener nuestro objetivo situaremos un control y un indicador por cada apartado a calcular (Por ejemplo la **I_b**):



En el Control se introducirá el valor calculado por los alumnos/as mientras que en el **Indicador** aparecerá el valor obtenido por el programa. Cuando los valores del **Control** y del **Indicador** coincidan, con un pequeño margen de tolerancia, se encenderá un **LED** que nos dirá que el resultado introducido es el correcto. En caso contrario, inferiremos que el resultado no es correcto.

Un aspecto importante del programa es el de especificar el número de unidades con que se debe ofrecer el resultado en los indicadores. Para ello nos situamos encima de los Indicadores y con el botón derecho del ratón pinchamos. Nos desplazamos al último apartado, a “Properties”, y dentro de ésta a “Format and Precision” siendo recomendable los valores que se ofrecen a continuación, aunque otras notaciones también son posibles (fig.10). La apariencia en Front Panel se ofrece en la fig.11.

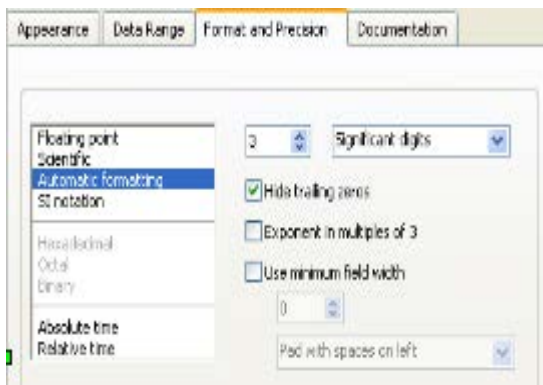


Fig.10 Format and precision (F.Labview)



Fig.11 Apariencia Control e Indicador de Ib

Lógicamente, al final del diseño de la actividad, el **indicador** tiene que ser ocultado, puesto que, de lo contrario, estaríamos dando el resultado al alumnado.

La forma de ocultar el **indicador** es pulsando con el botón derecho del ratón y desplegándose el siguiente menú:

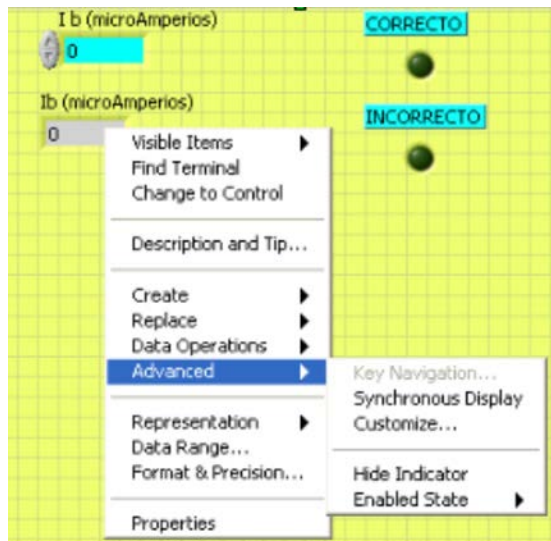


Fig.12 Ocultación del indicador al alumnado (Fuente Labview)

Pinchando en “**Hide Indicator**” conseguiremos ocultar el Indicador.

Se deberá proceder de la misma manera que la explicada para todos los controles e indicadores que sean objeto del cálculo del ejercicio (Ic, VCE y Pc), siendo válida la explicación de la ocultación de los indicadores en el resto de los casos.

La apariencia final de Front Panel es la mostrada en la figura 4 y la de Block Diagram se verá más adelante.

3.4 Sobre la programación:

Lo primero que deberemos hacer es introducir una estructura de programación conocida por todos, y que en los lenguajes de programación más antiguos se expresaba con el comando “**While**” (mientras). En esta programación es un símbolo que se encuentra apretando con el botón derecho del ratón la pantalla de **Block Diagram**, desplegándose el siguiente menú:



Fig13. While Loop (Fuente Labview)

El While Loop (en Block Diagram) lo haremos grande estirándolo con el ratón y situaremos todos los controles e indicadores dentro de éste.



Fig.14 While Loop(en posición “para si verdadero”).

Deberemos transformarlo a “Continua si verdadero” para que la ejecución del programa sea permanente.

El paso de “Stop if true” a “Continue if true” se muestra a continuación:



En Block Diagram y con el botón derecho del ratón, en All Functions, Boolean, encontraremos la puerta lógica Not de la figura. Las uniones se realizan con la bobina que podemos localizar en : Window >>> Show Tools Palette.

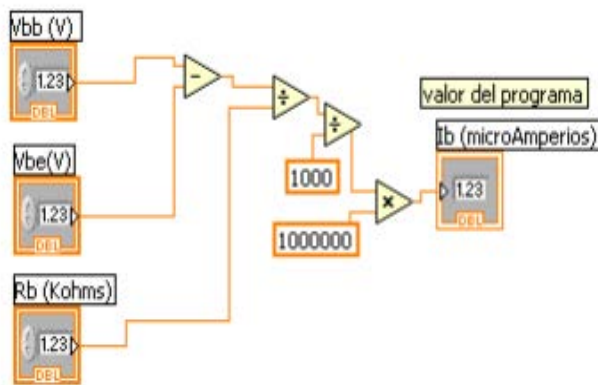
3.5 Sobre la unión de las variables de entrada (controles) y los indicadores LED's:

El cálculo de **I_b**, **I_c**, **V_{ce}** y **P_c**, se debe efectuar ligando las variables de entrada y de salida en función de las distintas relaciones matemáticas que se establezcan entre ellas.

Es fácilmente demostrable que el valor de la Intensidad de Base en transistores en emisor común se calcula como sigue: **I_b** (Intensidad de Base);

$$I_b = \frac{V_{bb} - V_{be}}{R_b}$$

Su traslación a Block Diagram es la siguiente:



Puede observarse como en el último paso, antes de repercutir los cálculos sobre el **Indicador**, dividimos por **1000**, porque el valor del **control R_b** está en **KΩ**, y por tanto conseguir **Amperios**. Como este es un valor muy pequeño en **Amperios**, **multiplicamos por 1.000.000** para pasar a una unidad más manejable en este caso, el **µA** (microAmperio).

El **indicador** mostrado en la expresión anterior ofrecerá los valores de **cálculo del programa**, que deberán ser comparados con los introducidos por los alumnos.

Fig.15 Cálculo de I_b en µA (microAmperios) Labview

Se deberá proceder de la misma manera con todos los indicadores que queremos calcular, y el programa ya sería ejecutable.

3.6 Comparación del valor “introducido” y del “calculado” por el programa:

Debemos tener claro que el objetivo del ejercicio es que los alumnos introduzcan un valor calculado en los correspondientes controles (variables de entrada), compare con un indicador (variable de salida), que ocultaremos, y que compare los valores y nos diga si el resultado es correcto o incorrecto mediante el encendido de un LED.

Estableciendo un 1% de tolerancia entre el valor introducido por los alumnos y el calculado por el programa, las condiciones exigibles para que se encienda el LED de correcto son:

$$0,99 \bullet ValorIndicador \leq ValorControl \leq 1,01 \bullet ValorIndicador$$

Además de que cuando el valor del control sea \neq de cero, el LED deberá encenderse.

Las condiciones anteriores las vamos a poder establecer mediante el siguiente programa, utilizando la **Lógica digital**, tal y como se muestra a continuación:

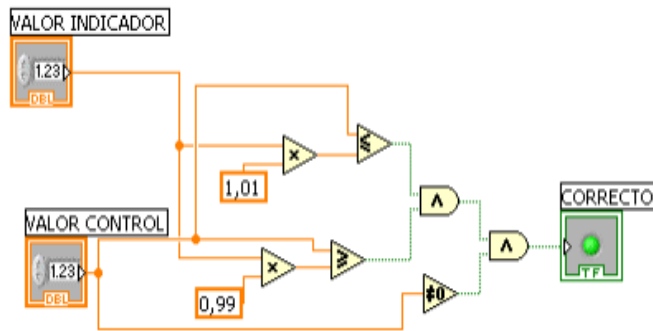
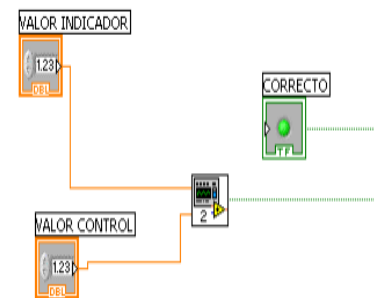


Fig.16 LED de correcto. Labview

Esta estructura de Correcto es aplicable al resto de cálculos, y Labview permite condensar la estructura que se repite en subprogramas llamados subvis. Quedaría como sigue:



Las condiciones de programación para cuando el cálculo de los alumnos es incorrecto y su subvi son las siguientes:

$$Valor.Control \leq 0,99 \bullet Valor.Indicador \quad ValorControl \geq 1,01 \bullet ValorIndicador$$

además de ser cierto lo anterior, cuando el valor del control sea = a cero, el LED incorrecto deberá encenderse.

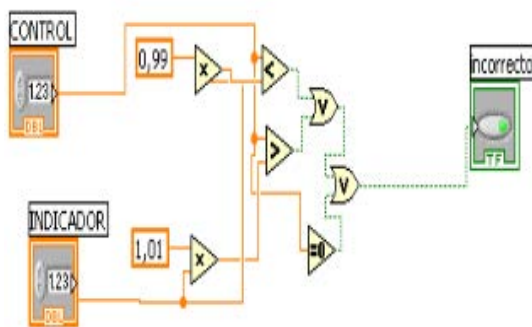


Fig.17 LED de incorrecto. Labview

Al igual que en el caso anterior, procederemos a realizar el subvi de incorrecto, para utilizarlo en todos los casos de cálculo.

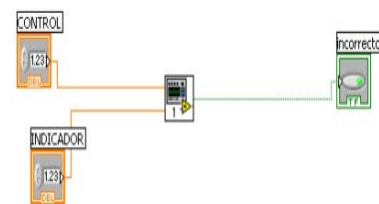


Fig.18 Subvi de incorrecto. Labview

Ya por último, sólo queda mostrar cual sería la apariencia de toda la programación del ejercicio, teniendo en cuenta que la interfaz de Front Panel deberá parecerse a la ofrecida en la figura 4.

Así pues, la programación final, teniendo en cuenta los subvis pertinentes, quedaría como sigue:

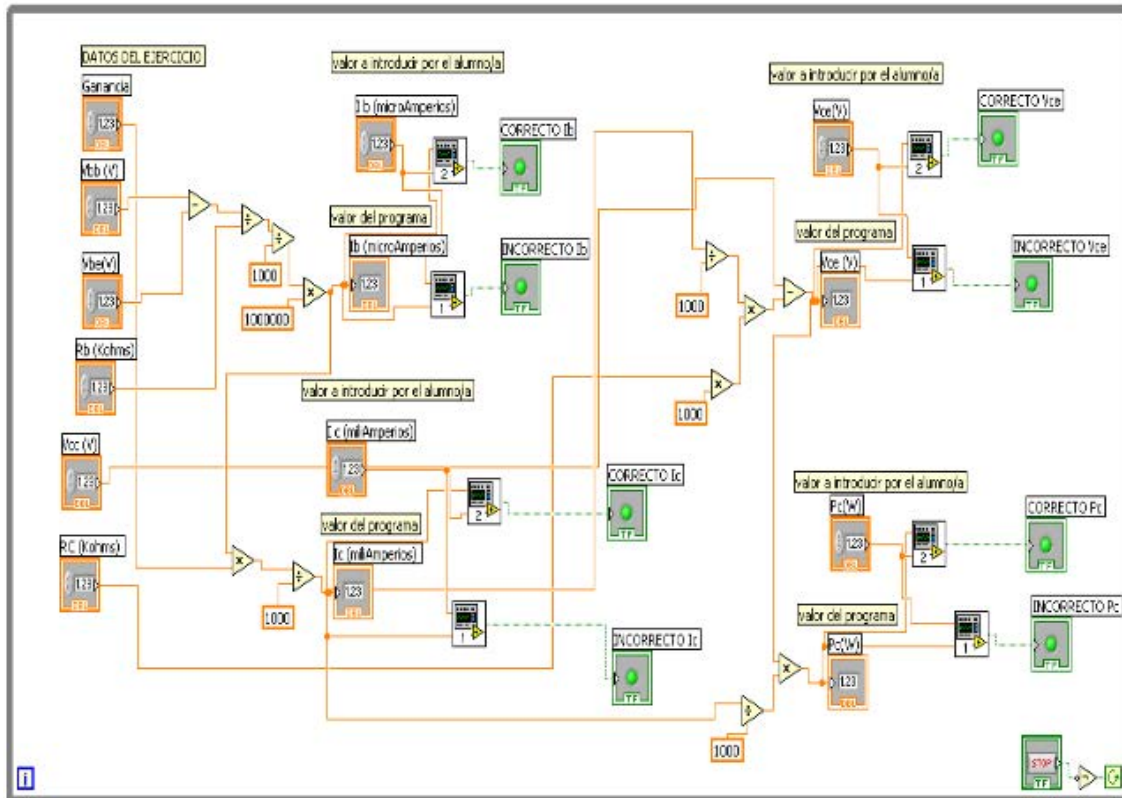


Fig.19 Programación en Block Diagram de todo el ejercicio. Labview

Bibliografía

- [1] Moreno Velasco I., Sánchez Ortega P (Introducción a la Instrumentación Virtual. Programación en Labview).
- Manual de aprendizaje Labview:
- <http://www.ni.com/getting-started/labview-basics/esa/>