

Proteus (Isis, Ares)

Muitos conhecem este poderoso software, ou já escutaram falar, ou já viram exemplos de circuitos simulados nele. Realmente no mundo da simulação de circuitos eletrônicos é de tirar o chapéu para software como este.

Aos amigos do forum: <http://www.forumnow.com.br/vip/foruns.asp?forum=41935> , que aconteceu uma oportunidade de alguns que não conhecem este poderoso software. A eles que gostam de programar microcontroladores e tirar suas duvidas é que foi dedicado esta página ...

Textos de **Arturo Sandoval Bermúdez**

Créditos para a amiga **Prof. Joana** pela tradução e correções, e sua dedicação de elevar nossos conhecimentos nesta área da tecnologia

Todos os exemplos foram tirados da versão **6.2**, que não difere muitos dos mais novos, ótima oportunidade de conhecer um pouco mais sobre este software . .. Boa leitura !

Introdução.

O software de desenho e simulação Proteus VSM é uma ferramenta útil para estudantes e profissionais que desejam acelerar e melhorar suas habilidades para do desenvolvimento de aplicações analógicas e digitais.

Ele permite o desenho de circuitos empregando um entorno gráfico no qual é possível colocar os símbolos representativos dos componentes e realizar a simulação de seu funcionamento sem o risco de ocasionar danos aos circuitos.

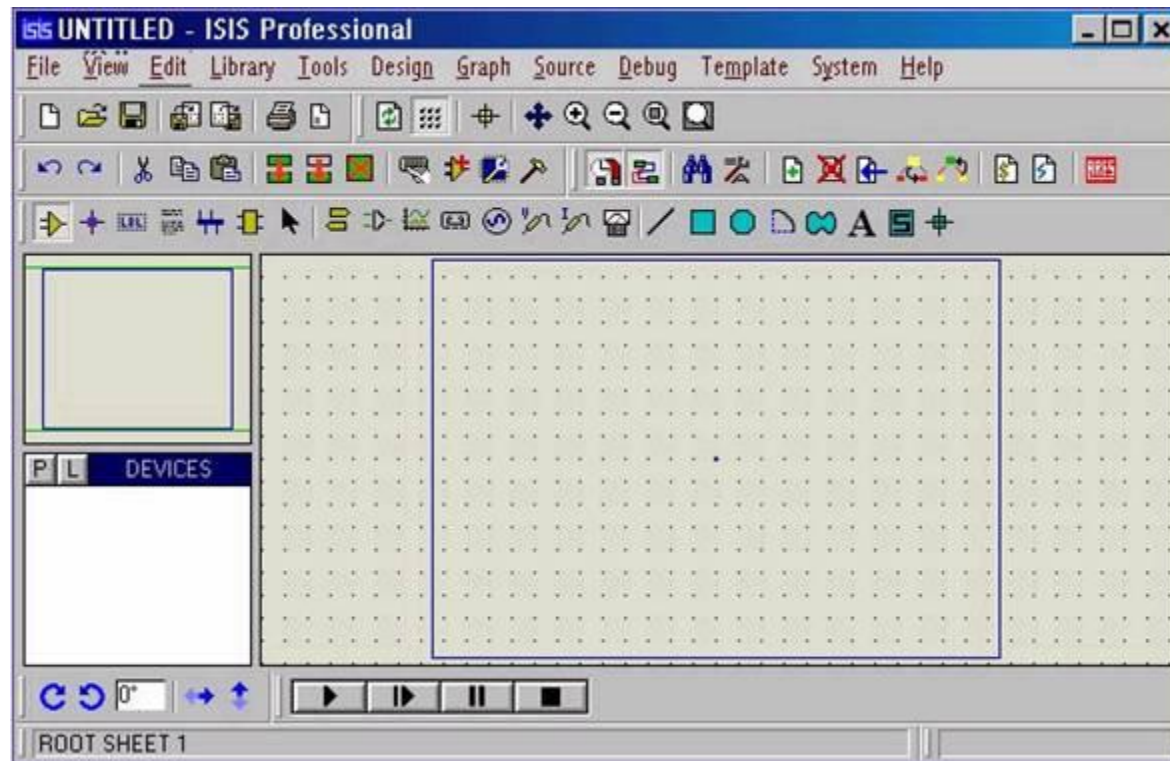
A simulação pode incluir instrumentos de medição e a inclusão de gráficas que representam os sinais obtidos na simulação.

O que mais interesse despertou é a capacidade de simular adequadamente o funcionamento dos microcontroladores mais populares (PICS, ATMEL-AVR, Motorola, 8051, etc.)

Também tem a capacidade de passar o desenho a um programa integrado chamado ARES no qual se pode levar a cabo o desenvolvimento de placas de circuitos impressos.

Procedimento de Arranque do programa:

1.- Início -> Programas -> Proteus 6 Professional -> ISIS 6 Professional.



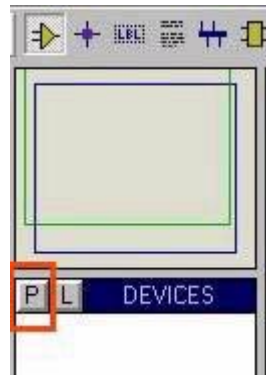
2.- A forma curta é dar um duplo click no ícone do programa que se localizado no desktop.

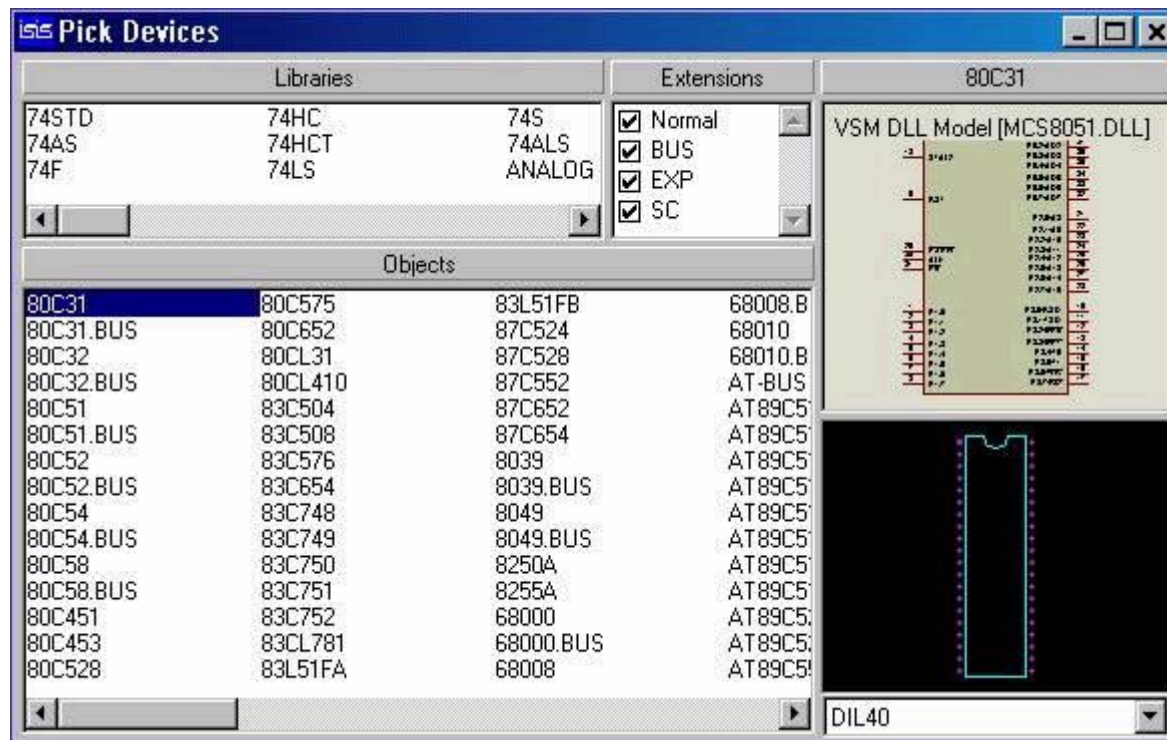


CIRCUITOS BÁSICOS

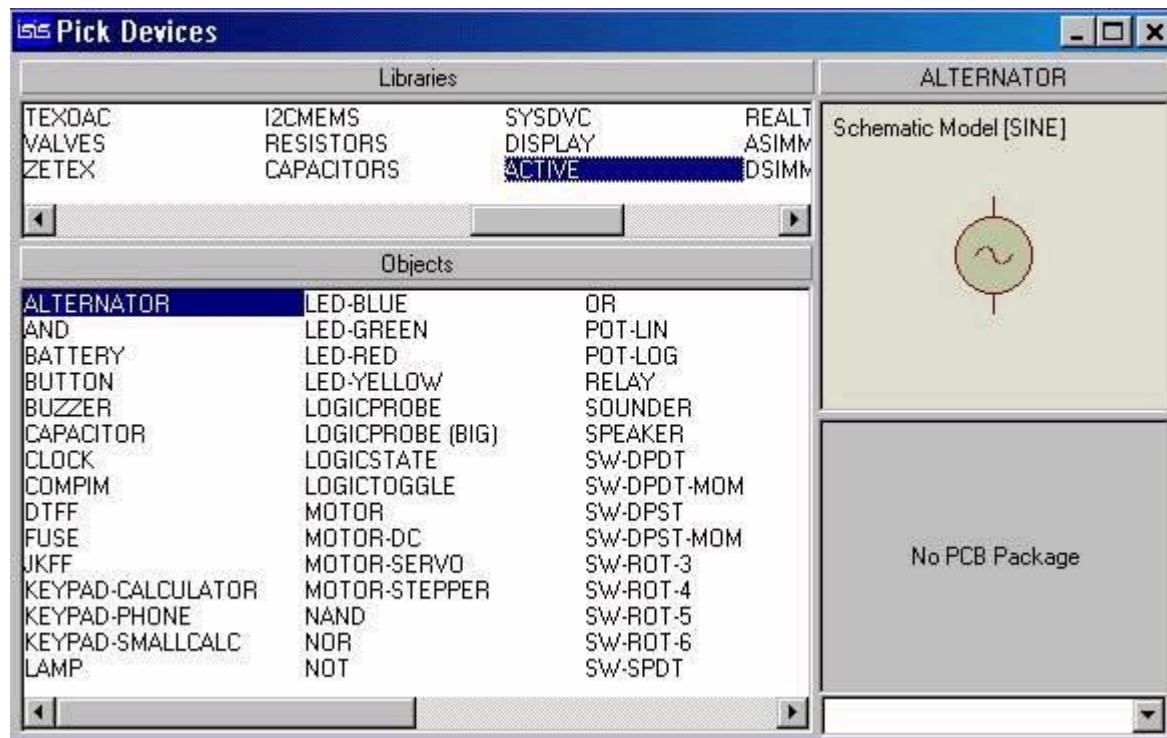
B.- Circuito Básico # 1 (Desenvolvimento) - Alimentação de um Lâmpada de corrente alterna.

1.- Dar um click no botão **Pick Devices** localizado na parte esquerda da tela debaixo da tela de exploração do diagrama para abrir a forma do mesmo nome.

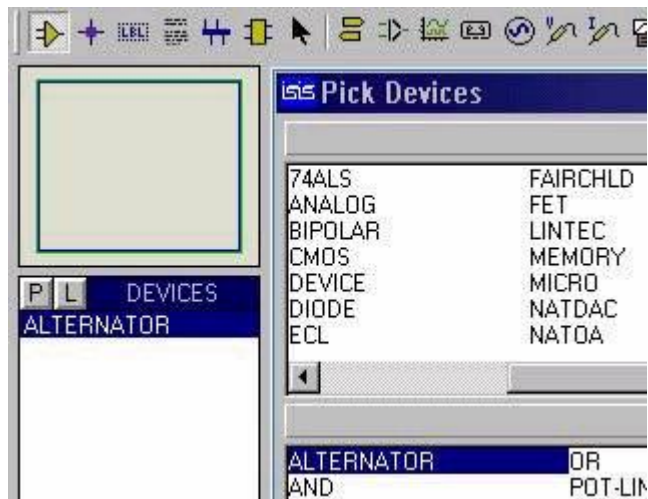




2.- Na janela **Libraries** (Parte superior esquerda) procurar a biblioteca **ACTIVE**, e dar um click sobre ela.



3.- Na janela **Objects** escolher o componente **ALTERNATOR** dando duplo click sobre o nome.



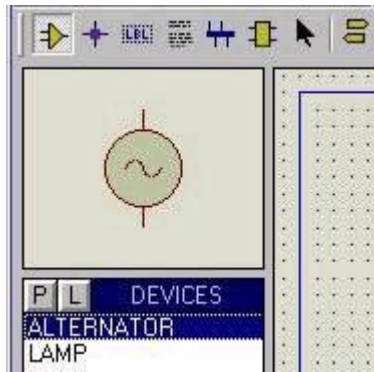
pode-se observar que na janela **DEVICES** aparece o nome do componente eleito. Se for o único componente que vai se escolher se pode fechar a forma Pick Devices, mas se for necessário mais de um, pode-se continuar escolhendo os componentes necessários para nosso desenho

4.- Na mesma biblioteca **ACTIVE** dar duplo click sobre o componente **LAMP**.



5.- Fechar a Forma **Pick Devices** no botão padrão. (A cruz na esquina superior direita)

6.- Dar um click na palavra **ALTERNATOR** da janela **DEVICES** e observar que aparece o componente na tela de exploração do circuito.



7.- Explorar as funções de orientação do componente, parte inferior esquerda da tela.

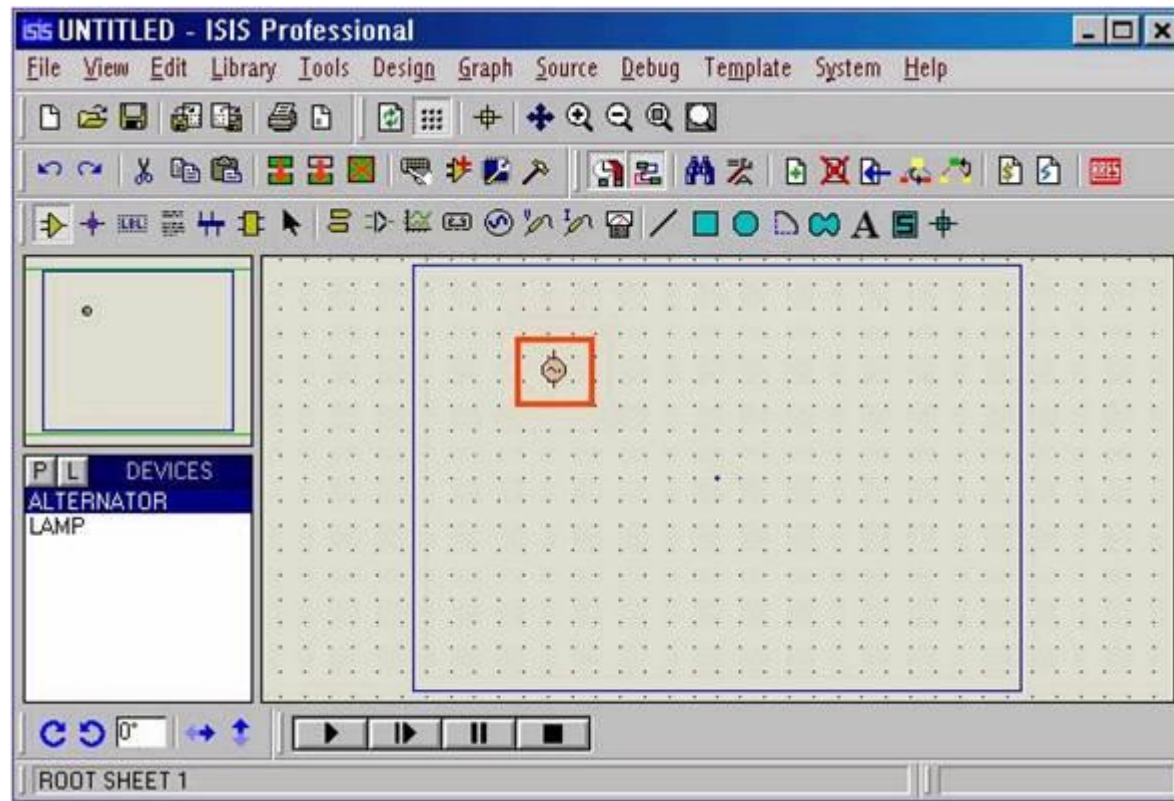


8.- Começando pela esquerda pressionar cada um dos botões de orientação.

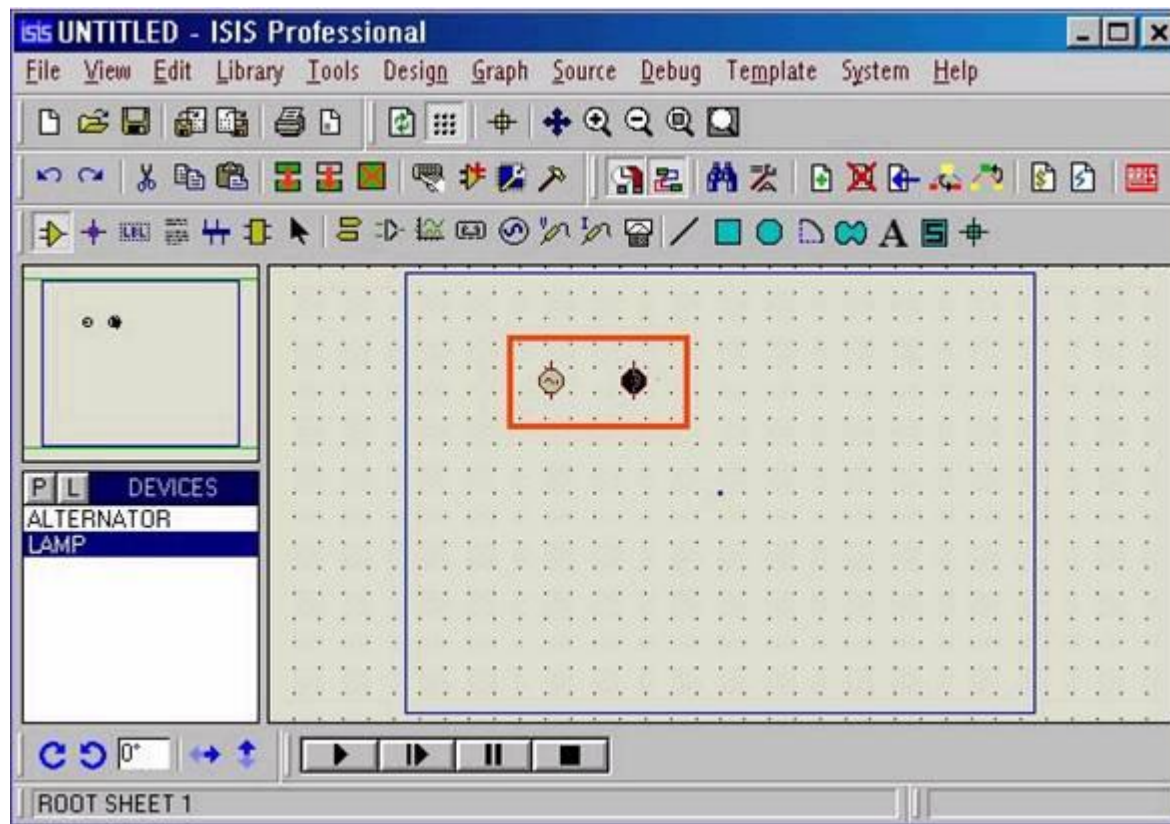
9.- No quadro de texto se pode introduzir um ângulo mas só aceita valores de (0° , $\pm 90^\circ$, $\pm 180^\circ$, $\pm 270^\circ$), por isso é melhor dirigir a orientação por meio dos botões. Este mesmo quadro de texto mostra o ângulo atual obtido ao pressionar os botões.

10.- Deixar o componente na posição inicial.

11.- Com o componente selecionado dar um click na área de trabalho, com o que se consegue colocar o componente na área de trabalho.



12.- Repetir o procedimento anterior com o componente LAMP.



13.- Configurar os componentes da seguinte maneira.

A.- Dar um click com o botão direito sobre o componente **ALTERNATOR**. Notar que seu contorno troca para vermelho.

B.- Dar um click agora com o botão esquerdo para abrir a forma **Edit Component**.

C.- Dar um nome ao componente no campo Component Reference (AC1), Pôr o valor do componente no Component Value (12V), Modificar o valor da amplitude para (12V) e a frequência para 0.5Hz.

d.- Pressionar o botão OK.

E.- Verificar os valores do componente **LAMP** e se o valor da voltagem corresponder com o do **ALTERNATOR**, não é necessário realizar nenhuma modificação. Pressionar OK.

Edit Component

Component Reference: Hidden: ☐

Nominal Voltage: Hidden: ☐

Resistance: Hide All

Other Properties:

Attach hierarchy module: ☐

Edit all properties as text: ☐

14.- Realizar a conexão dos componentes da seguinte forma:

A.- Colocar o ponteiro do mouse no extremo superior do ALTERNATOR. Aparece uma cruz no extremo da flecha.

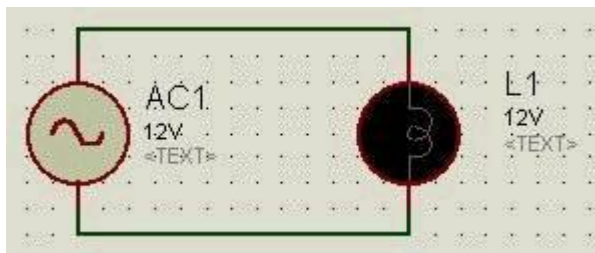
B.- Dar um click para habilitar a conexão por meio de cabo.

C.- Deslocar o mouse (desaparece a cruz) até o extremo superior do componente LAMP e obter que volte a aparecer a cruz no extremo da flecha.

d.- Dar outro click para realizar a conexão.

E.- Repetir os passos anteriores para a pare inferior dos componentes.

Resultado:



Este é o procedimento padrão para conectar qualquer componente com o que se trabalhe no programa.

15.- Provar o funcionamento do circuito pressionando o botão play que se encontra na parte inferior da tela.

16.- Para aproximar o circuito e poder observar melhor a simulação se pode recorrer aos controles de zoom.



Começando da esquerda para a direita temos:

A.- Re-centrar a tela.

B.- Incrementar a aproximação.

C.- Decrementar a aproximação.

d.- Ver a folha completa.

E.- Ver uma área selecionada

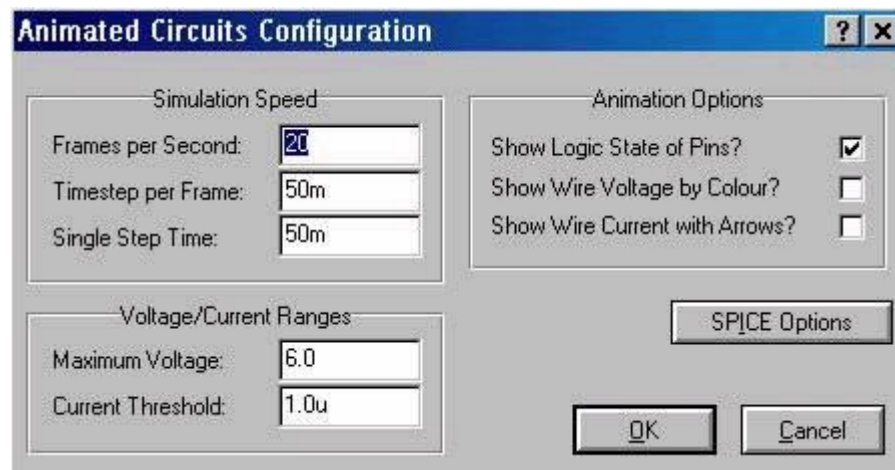
Usar a ferramenta para ver uma área selecionada dando um click

Usando o cursor modificado pressionar o botão esquerdo no extremo superior esquerdo do circuito armado e sem soltar o botão formar um retângulo que contenha todo o circuito, por último soltar o botão.

Este procedimento se pode usar para aproximar partes de um circuito de maior tamanho.

17.- Habilitar as cores de voltagem e as flechas de corrente do circuito para completar as simulação.

A.- Ingressar no menu System e selecionar **Set Animation Options...** para abrir a forma **Animated Circuits Configuration**.



B.- Habilitar as casinhas Show Wire Voltagem by Colour? e Show Wire Current with Arrows?.

C.- Pressionar OK.

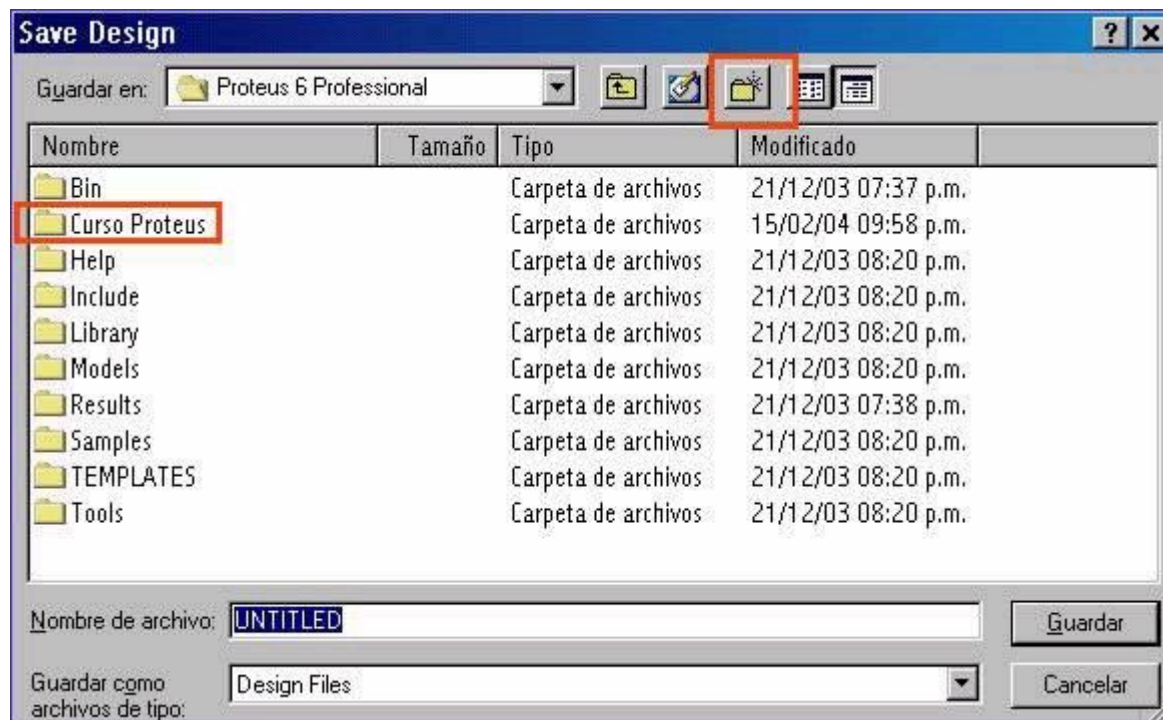
18.- Voltar a simular o circuito e observar o que ocorre.

19.- Salvar o circuito.

A.- Selecionar a ferramenta Save current design.



B.- Dar un click no botão criar uma nova pasta e lhe dar o nome de Curso Proteus.



C.- Ingressar na pasta proteus e criar dentro desta, uma pasta com o nome Exercício 1.



d.- No campo **Nome de arquivo** nomear ao arquivo como Exercício 1.

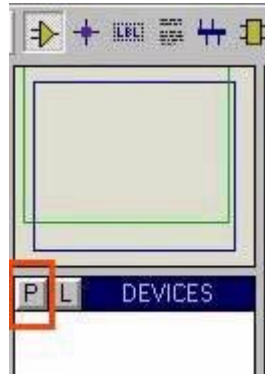
dar um click no botão salvar ou pressionar a tecla **ENTER**.

C.- Circuito Básico #2 (Desenvolvimento) - Bateria - Interruptor - Lampada.

1.- Dar um click em **Create a New Design**.



2. Presionar el botón **Pick Devices**.

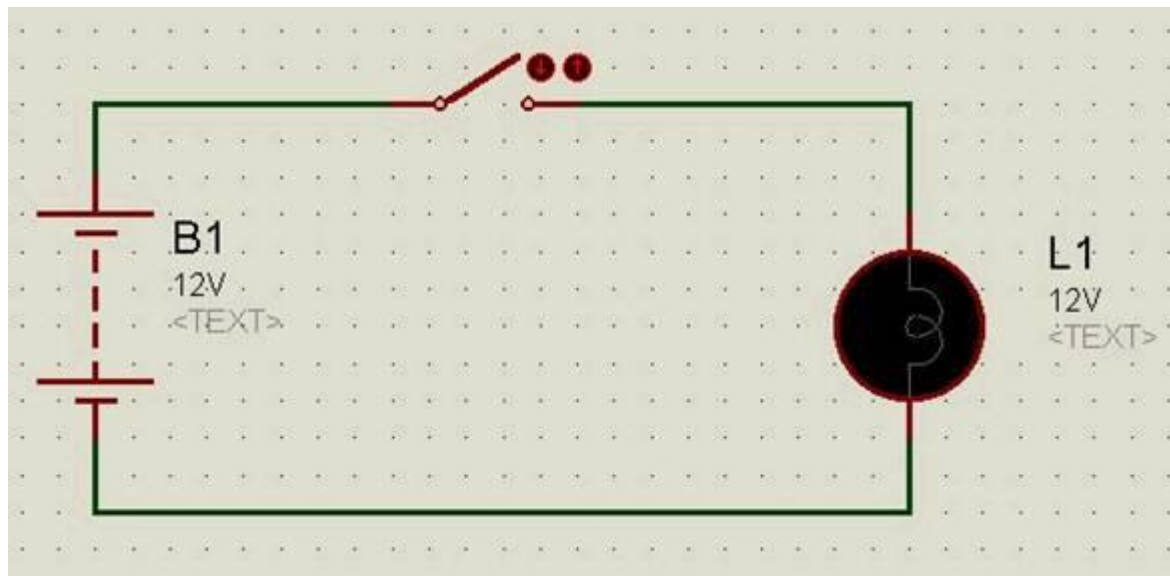


3.- Da janela **Libraries** selecionar ATIVE e na janela Objects escolher os componentes:

- BATTERY
- LAMP
- SWITCH



4.- Armar o siguiente circuito:



Modificar os valores dos componentes se for necessário.

- 5.- Executar a simulação do circuito e testar o funcionamento do interruptor. Dando clicks com o botão esquerdo nas flechas acima - abaixo do interruptor.
- 6.- Salvar o arquivo.
 - A.- Pressionar Save current design.
 - B.- Sair da pasta do Exercício 1. (Subir um nível)
 - C.- dentro da pasta de Curso Proteus, criar uma nova pasta com o nome Exercício 2.
 - d.- Ingressar na pasta Exercício 2 e dentro dela salvar o arquivo com o nome Exercício 2.

D.- Circuito Básico #3 (Desenvolvimento) - Bateria - Potenciometro - Lampada.

1.- Dar um click em **Create a New Design.**

2. Pressionar o botão **Pick Devices.**

3.- Da janela **Libraries** selecionar ACTIVE e na janela Objects escolher os componentes:

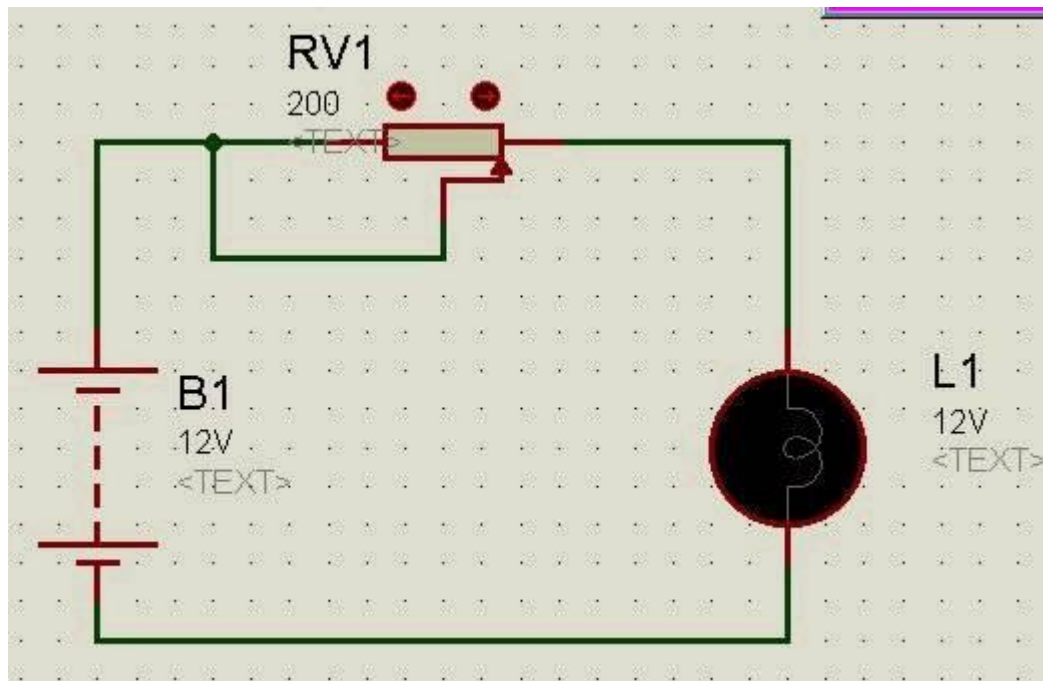
- BATTERY

- LAMP

- POT-LIN



4.- Armar o seguinte circuito:

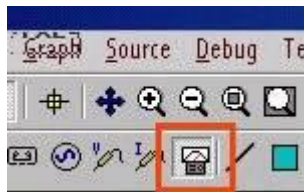


- 5.- Executar a simulação do circuito e provar o funcionamento do potenciometro. Com o ponteiro do mouse dar click nas flechas para aumentar ou diminuir a resistência.
- 6.- Salvar o arquivo.
 - A.- Pressionar Save current design.
 - B.- Sair da pasta do Exercício 2. (Subir um nível)
 - C.- dentro da pasta de Curso Proteus, criar uma nova pasta com o nome Exercício 3.
 - d.- Ingressar na pasta Exercício 3 e dentro dela salvar o arquivo com o nome Exercício 3.

E.- Adicionar instrumentos de medida a um circuito.

Usando o circuito anterior fazer o seguinte:

A.- Da barra superior de ferramentas selecionar Virtual Instruments.

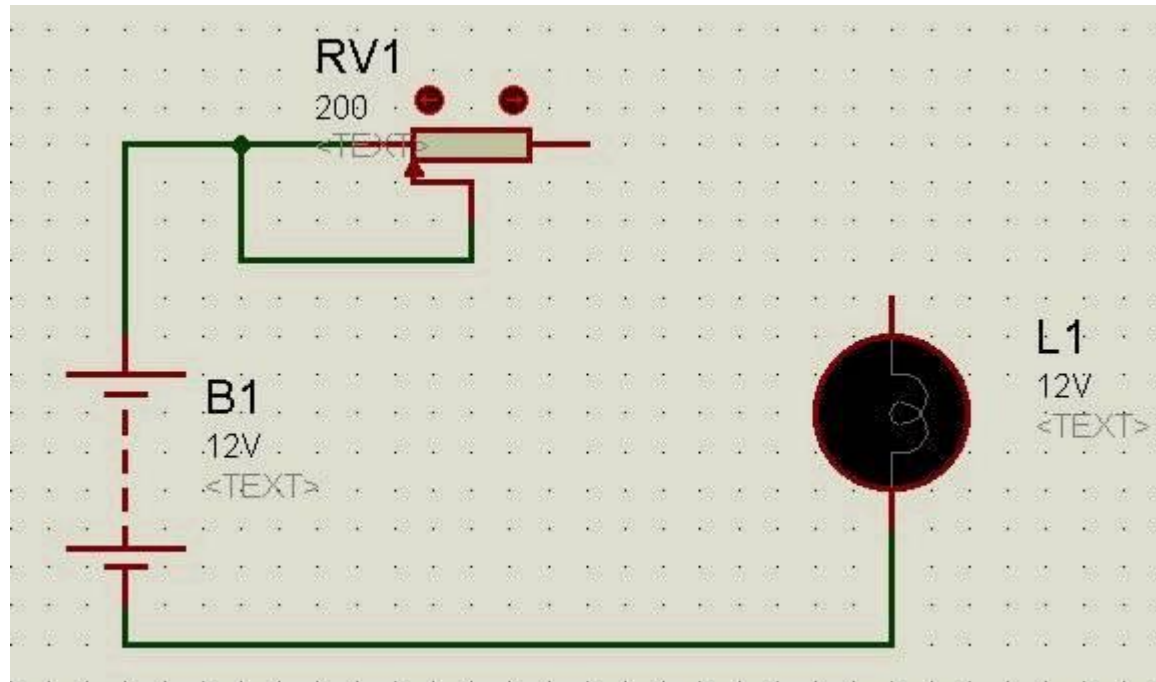


B.- Dar duplo click com o botão no cabo que une o potenciometro e a lampada para deixar espaço a um amperímetro. Se for necessário mover um pouco a lampada para a direita.

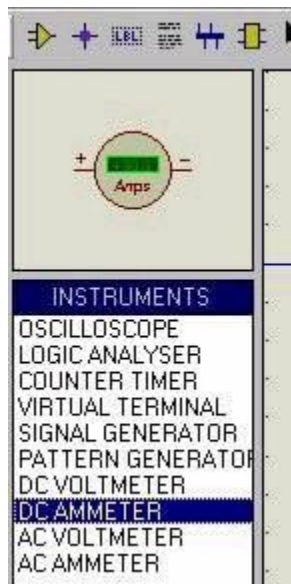
NOTA: Para mover um componente na área de trabalho se realizam os seguintes passos:

I.- Dar um click com o botão direito sobre o componente que se deseja mover.

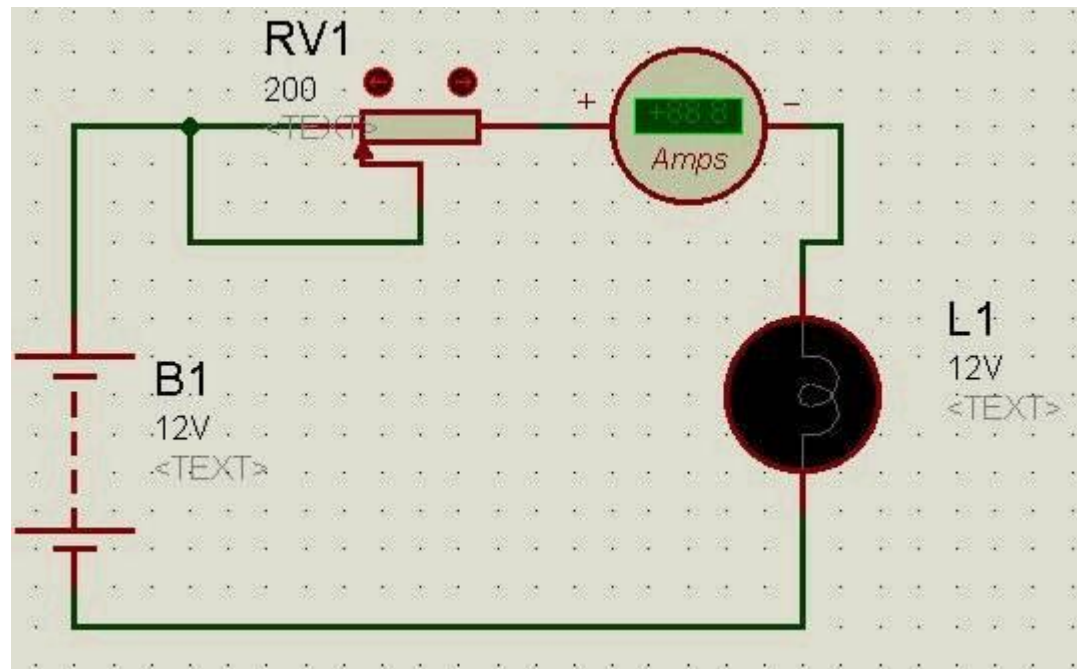
ii.- Pressionar o botão esquerdo sobre o componente e sem soltar arrastar o componente à posição desejada.

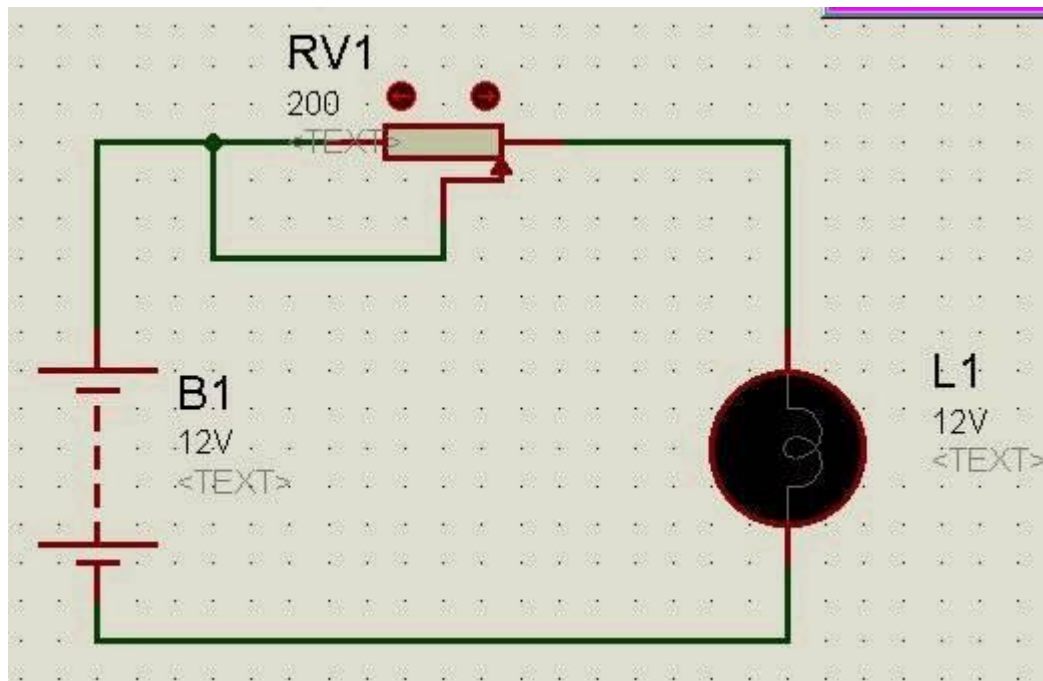


C.- Da janela **INSTRUMENTS** ao lado esquerdo da tela selecionar com um click esquerdo o instrumento **DC AMMETER**.



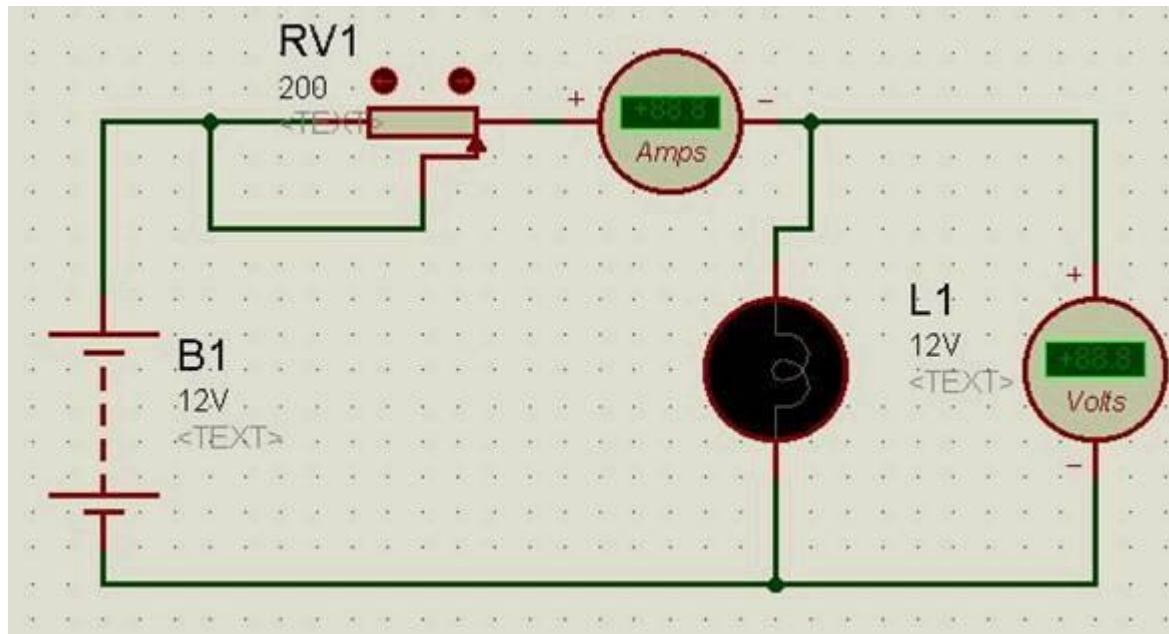
d.- Dar um click na área de trabalho entre o potenciômetro e a lâmpada, e conectar os componentes para obter o seguinte circuito.





E.- Selecionar o instrumento **DC VOLTMETER** da janela **INSTRUMENTS**.

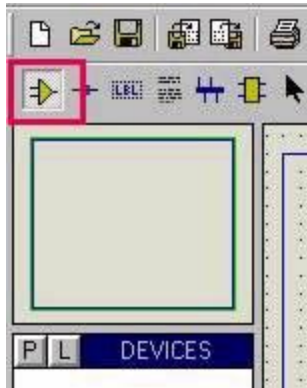
F.- Colocar na área de trabalho da mesma forma que o instrumento anterior para obter o seguinte circuito.



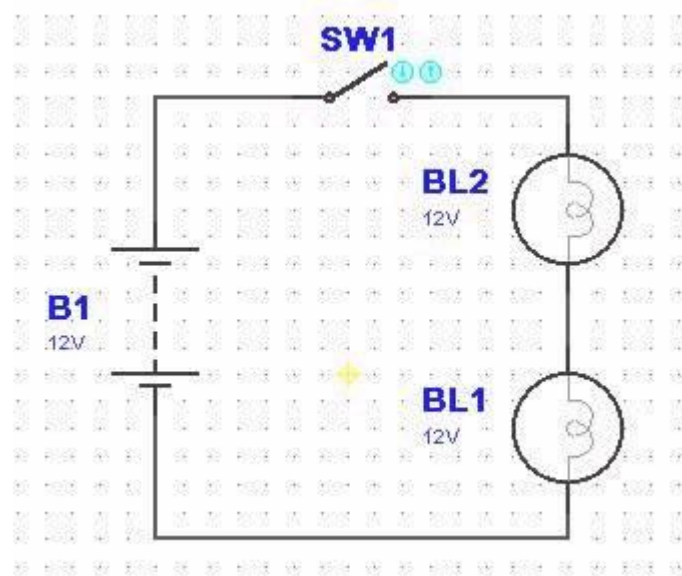
G.- Executar a simulação do circuito.

H.- Salvar o circuito. Como já tinha sido salvando o arquivo já não é necessário trocar de diretório ou nomear o arquivo.

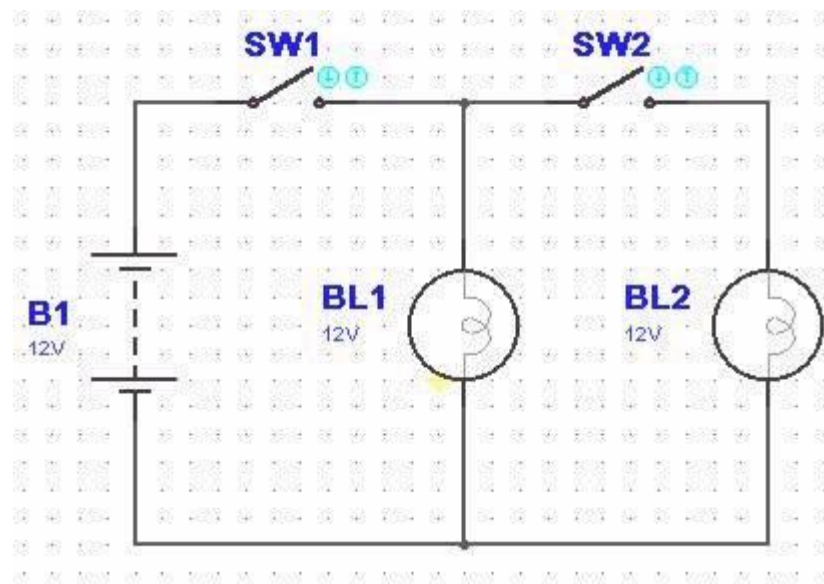
NOTA: para voltar há habilitar a janela **DEVICES** pressionar, na barra de ferramentas, **Component**.



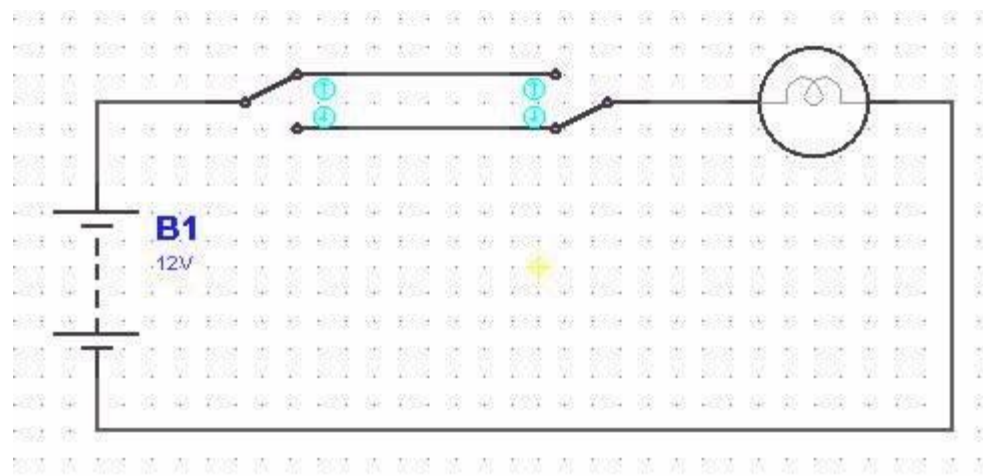
F.- Circuito Básico #4 - Circuito Serie.



G.- Circuito Básico #5 - Circuito Paralelo.



H.- Circuito Básico #6 - Circuito com dois interruptores para controle em duas direções.

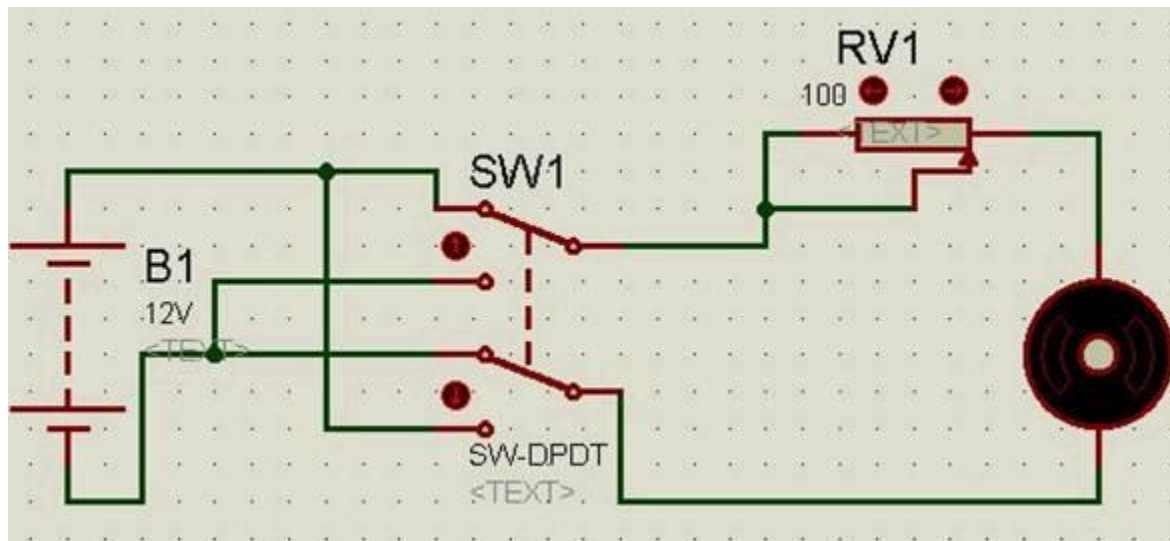


NOTA: Usar o componente **SW-SPDT** da biblioteca **ACTIVE**.



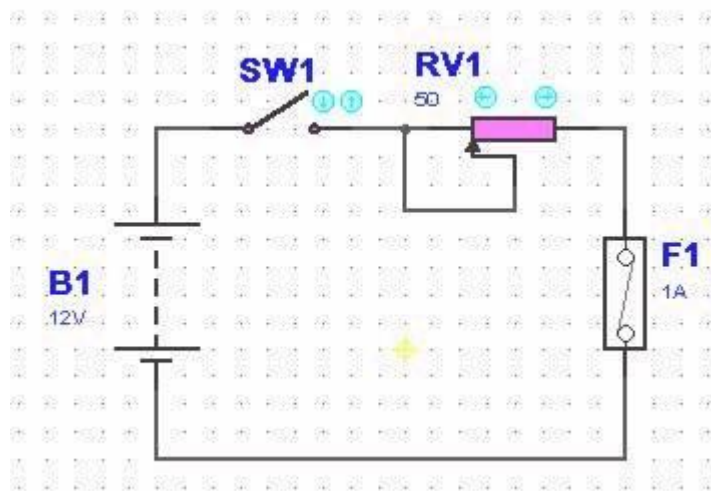
PROTEUS **VSM 2**

I.- Circuito Básico #7 - Controle de direção de um motor CD.



O motor se encontra na biblioteca **ACTIVE** e terá que selecionar o que só tem o nome **MOTOR**.
Os interruptores também se encontram na biblioteca **ACTIVE** e seu nome é **SW-DPDT**.

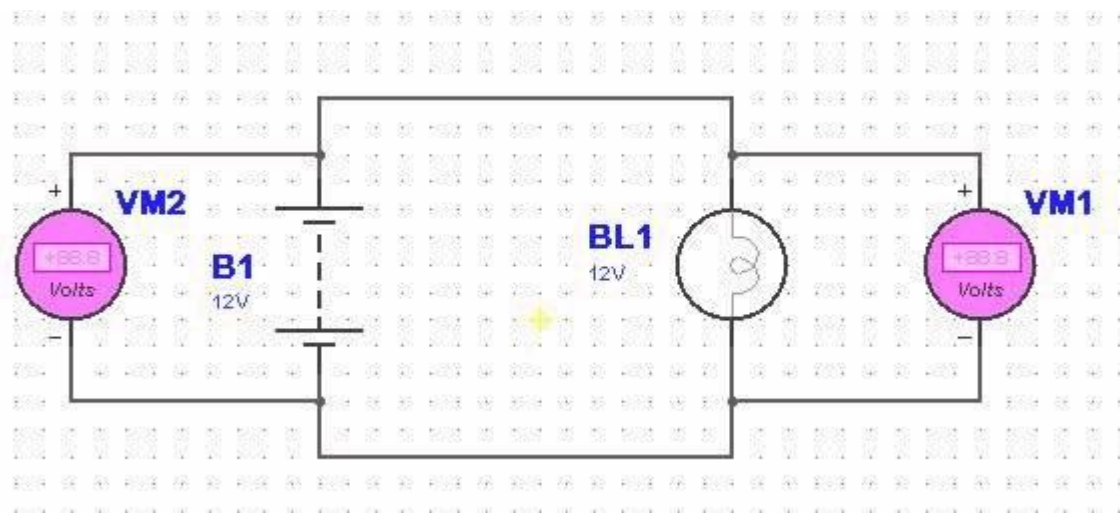
J.- Circuito Básico #8 - Uso de Fusíveis.

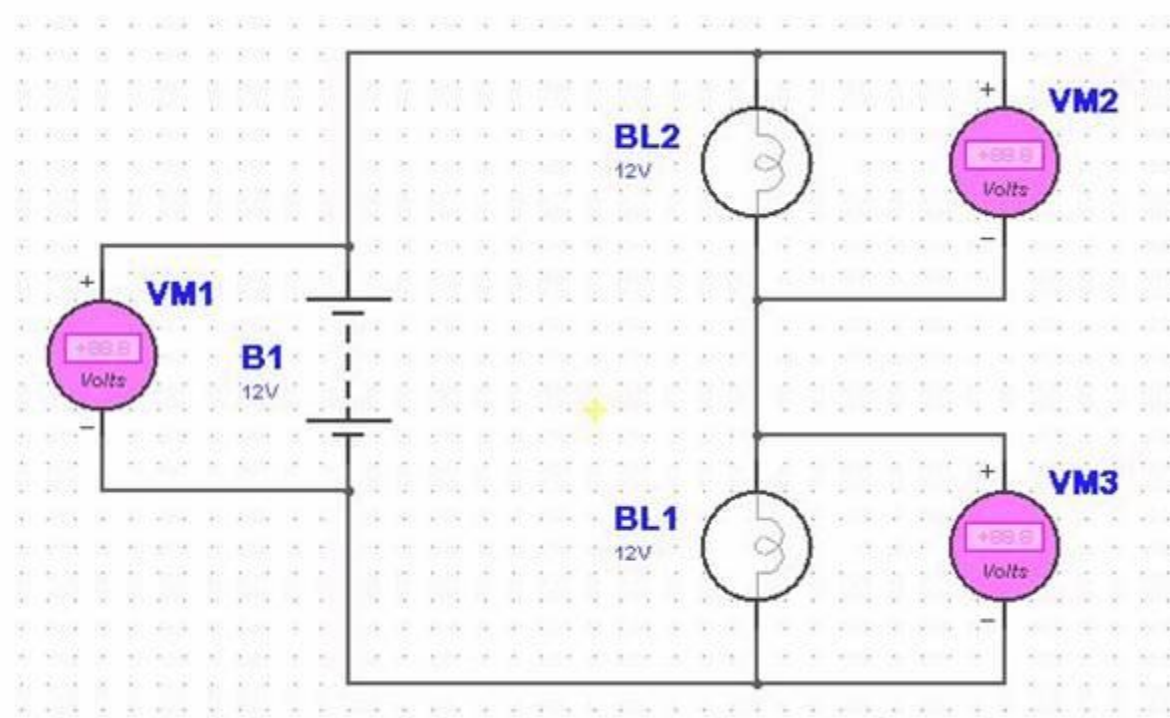


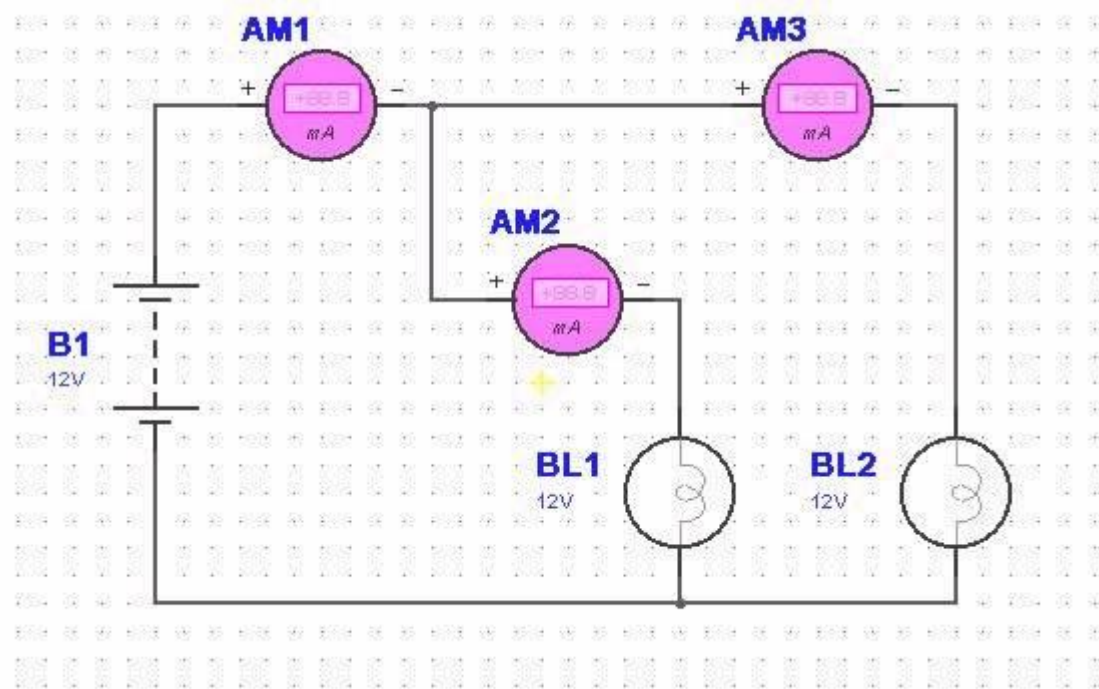
O fusível se encontra dentro da biblioteca **ACTIVE** baixo o nome do FUSE.

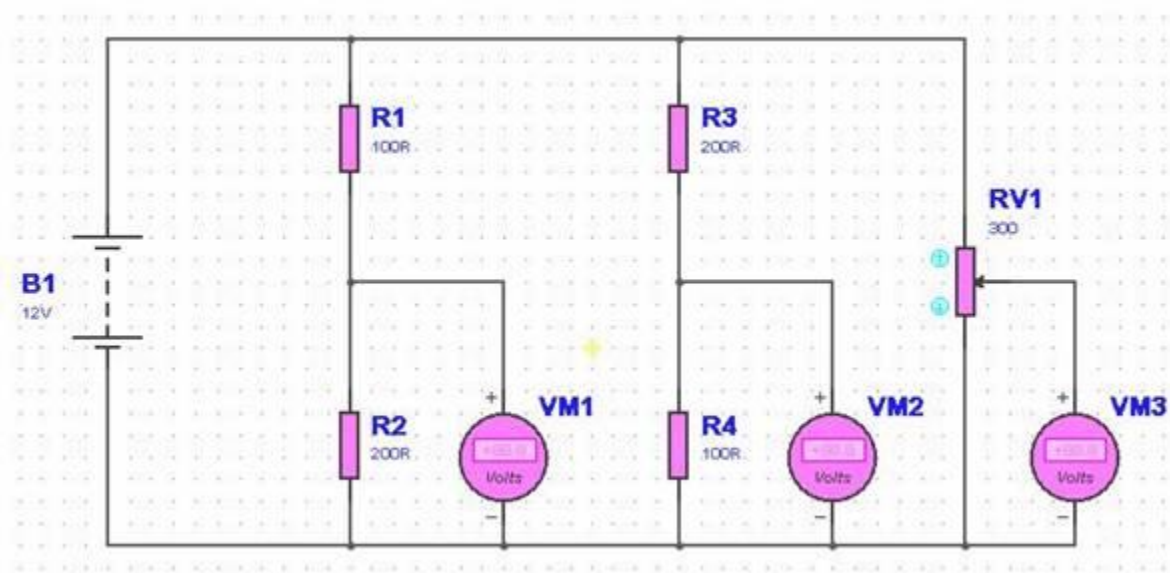
- Modificar este circuito lhe adicionando um amperímetro.
- Modificar o circuito do motor adicionando um amperímetro e um vóltmetro.

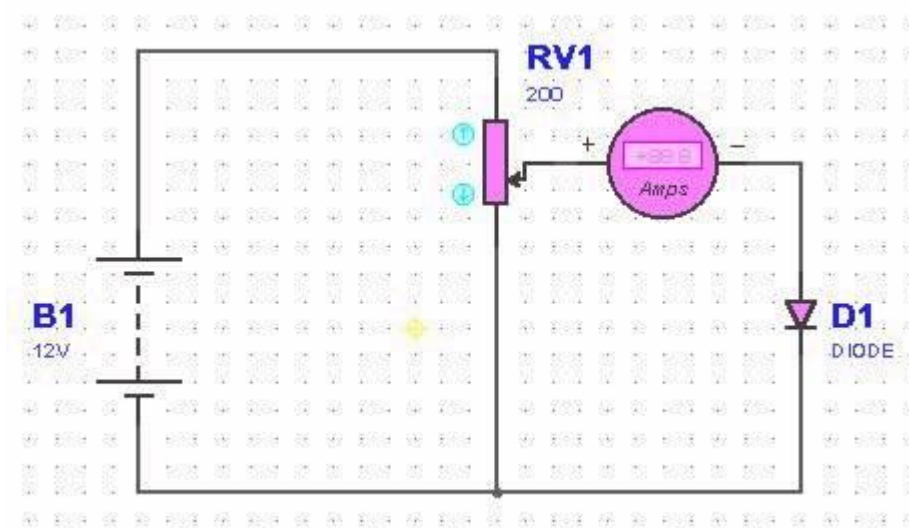
K.- Exercícios:



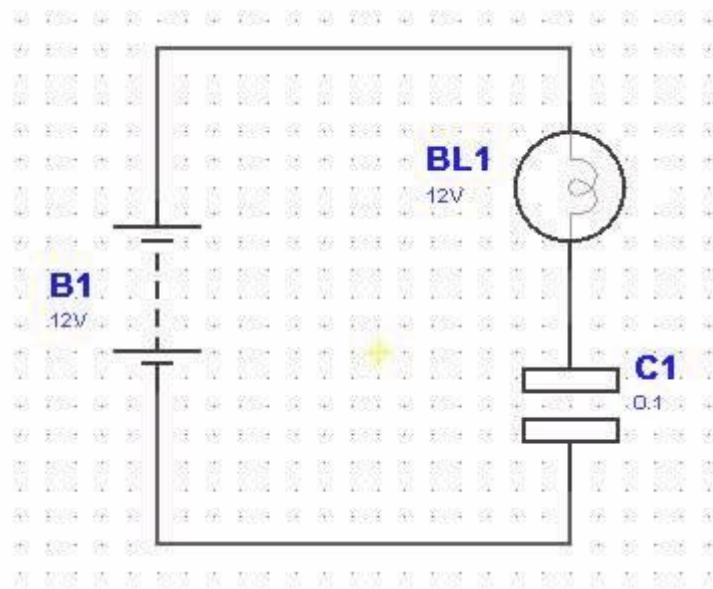




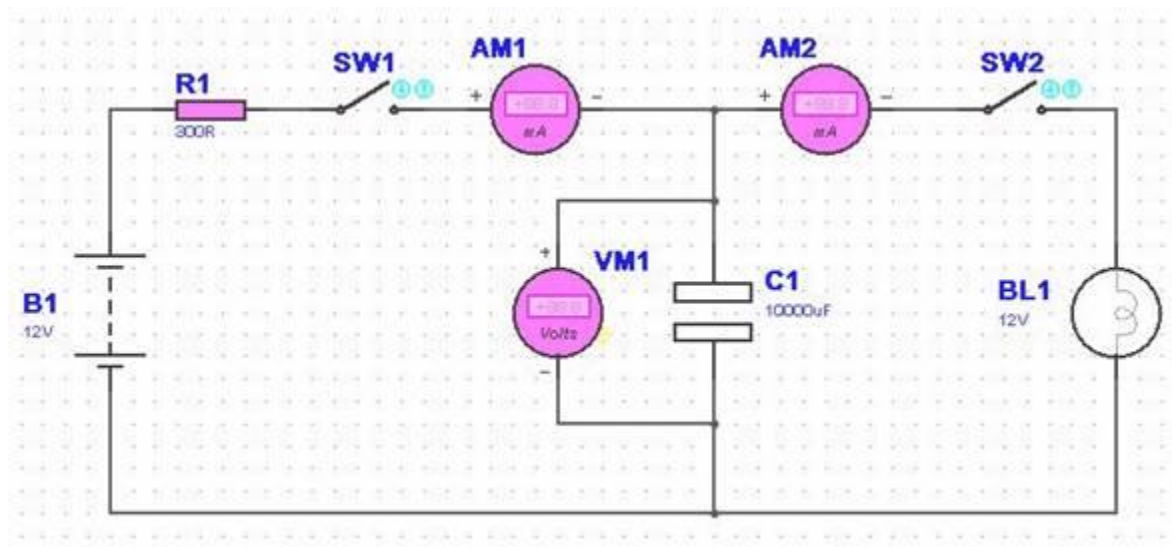




O Diodo se encontra na biblioteca DEVICE baixo o nome DIODE.



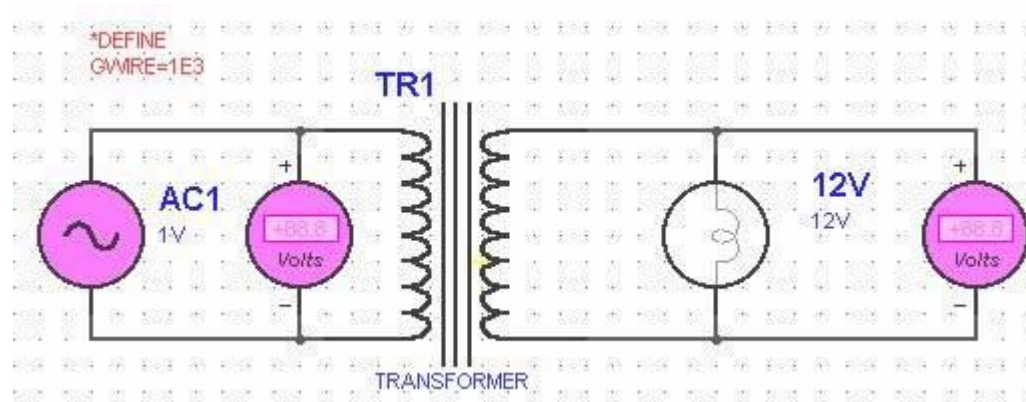
O capacitor se encontra na livreria ATIVE abaixo do nome CAPACITOR.



PROTEUS VSM 3

L.- Exercícios 2:

A.- Uso de um transformador.



TRANSFORMER = LIBRERÍA (DEVICE) - TRAN-2P2S

*DEFINE

GWIRE=1E3

Configuração do ALTERNATOR

Edit Component

Component Reference: AC1 Hidden: ☐

Component Value: 1V Hidden: ☐

Amplitude: 1V Hide All ▼

Frequency: 0.2Hz Hide All ▼

Other Properties:

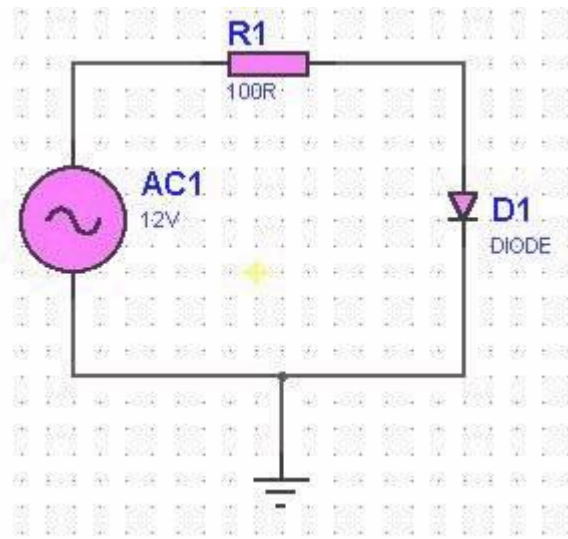
Attach hierarchy module: ☐

Edit all properties as text: ☐

OK Cancel

Configuração do TRAN-2P2S

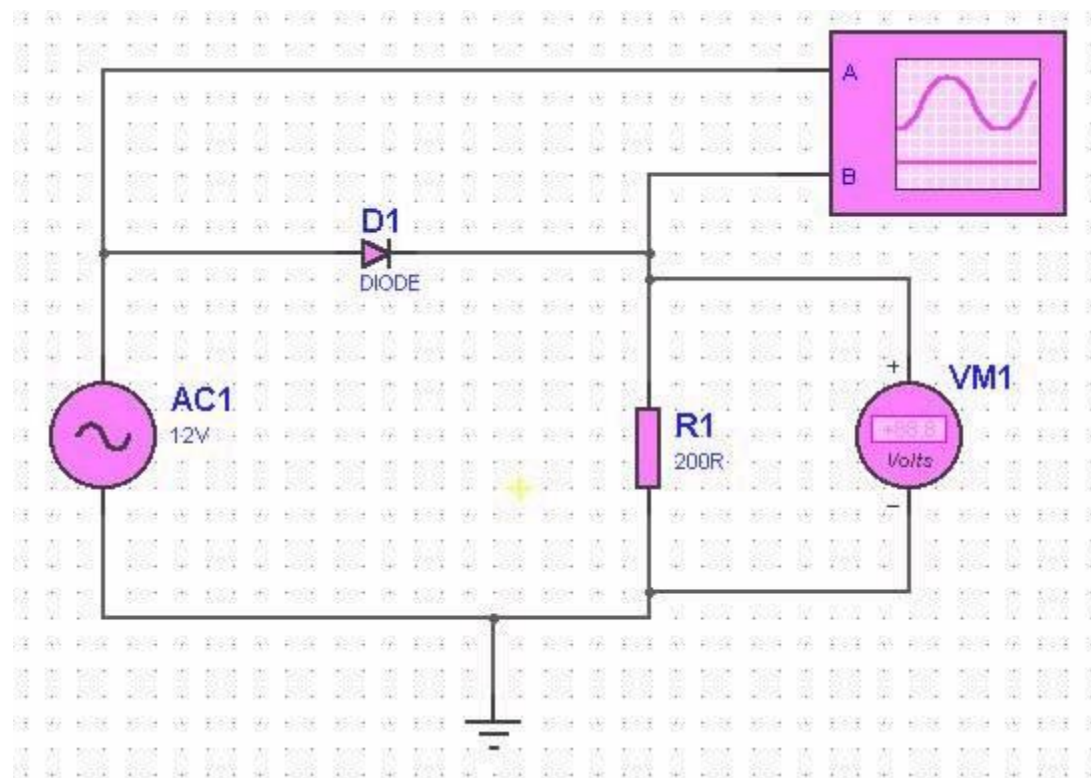
B.- Diodo com uma fonte alternada.



Resistência = DEVICE (RES)

Diodo = DEVICE (DIODE)

B.- Retificador de Meia Onda.



Usar o ALTERNATOR com a seguinte configuração inicial.

Edit Component

Component Reference: Hidden: ☐

Component Value: Hidden: ☐

Amplitude: Hide All ▼

Frequency: Hide All ▼

Other Properties:

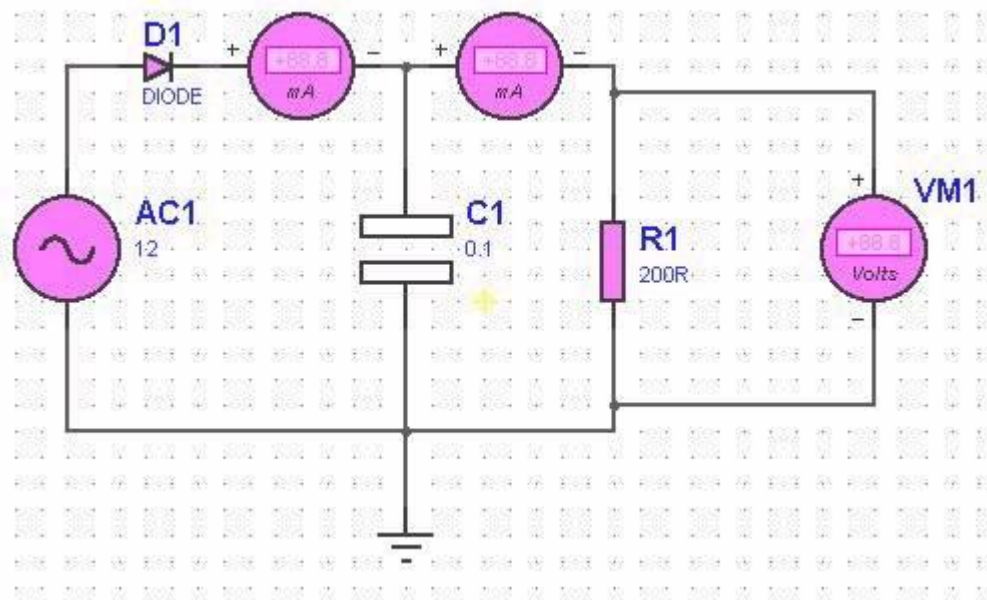
Attach hierarchy module: ☐

Edit all properties as text: ☐

OK Cancel

Para poder observar a saída no osciloscópio trocar a Frequência a 60Hz.

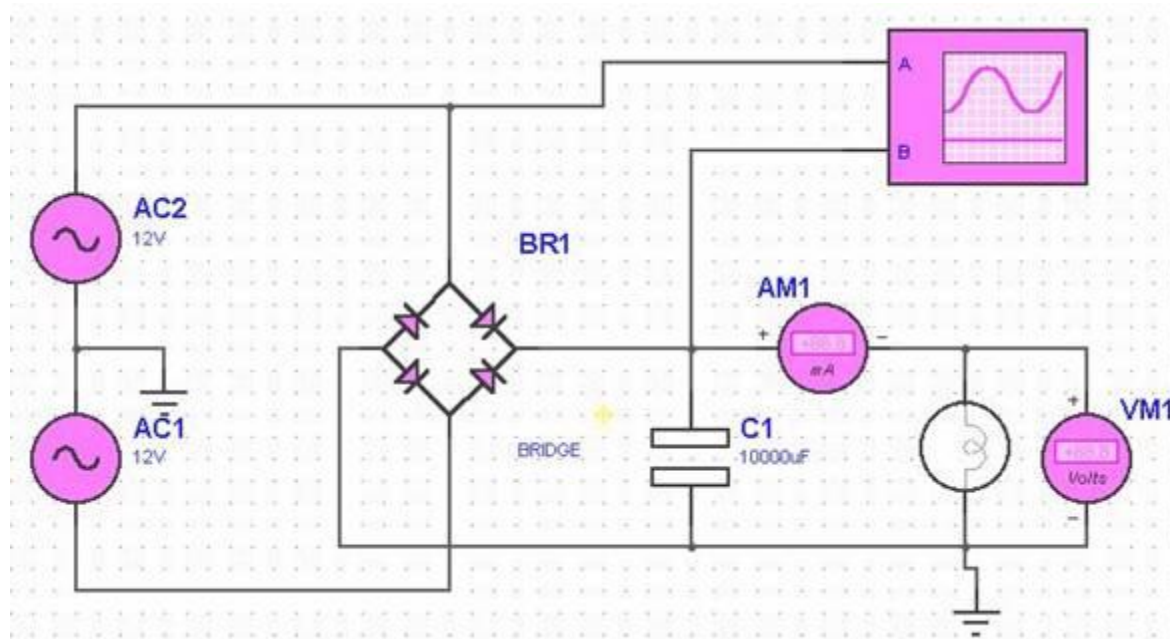
***** Adicionar amperímetros e capacitor.**



CAPACITOR = ACTIVE (CAPACITOR)

Trocar as características dos amperímetros ao MA. E provar a resolução com as outras filas.

C.- Retificador de onda completa usando uma Ponte Retificadora.



PONTE RETIFICADORA = DEVICE (BRIDGE)

*****Configuração ALTERNATOR.**

Edit Component

Component Reference: Hidden: ☐

Component Value: Hidden: ☐

Amplitude: Hide All ▼

Frequency: Hide All ▼

Other Properties:

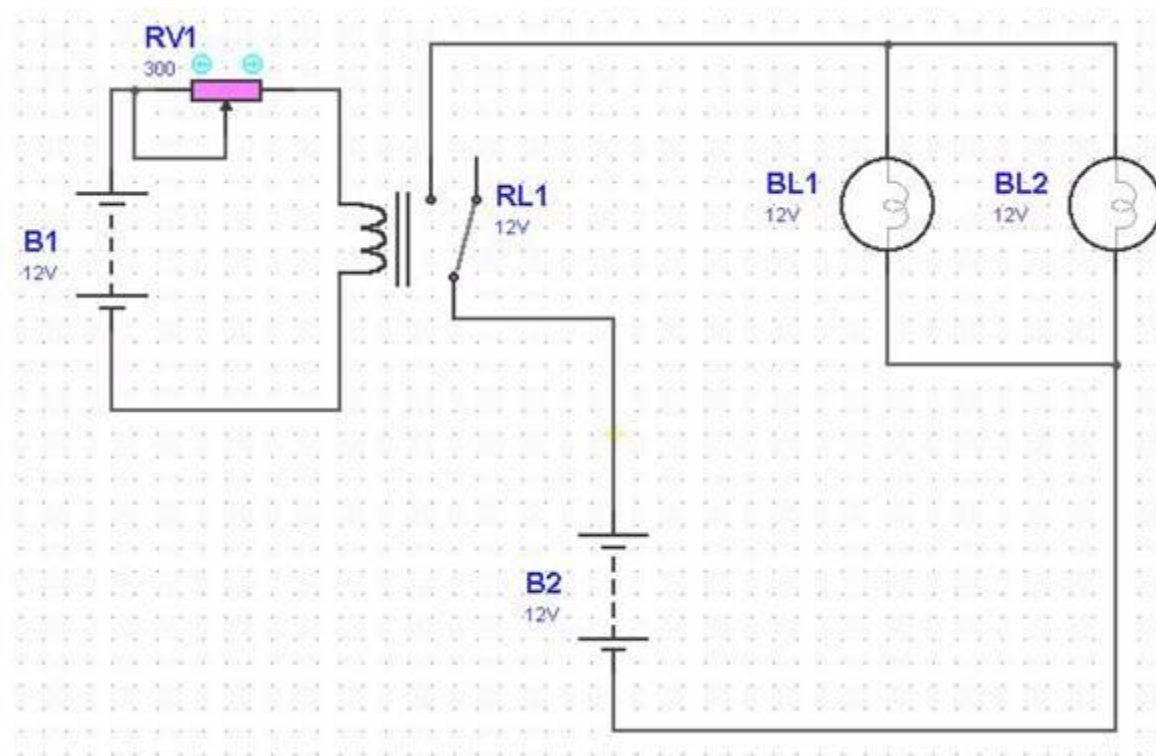
Attach hierarchy module: ☐

Edit all properties as text: ☐

OK Cancel

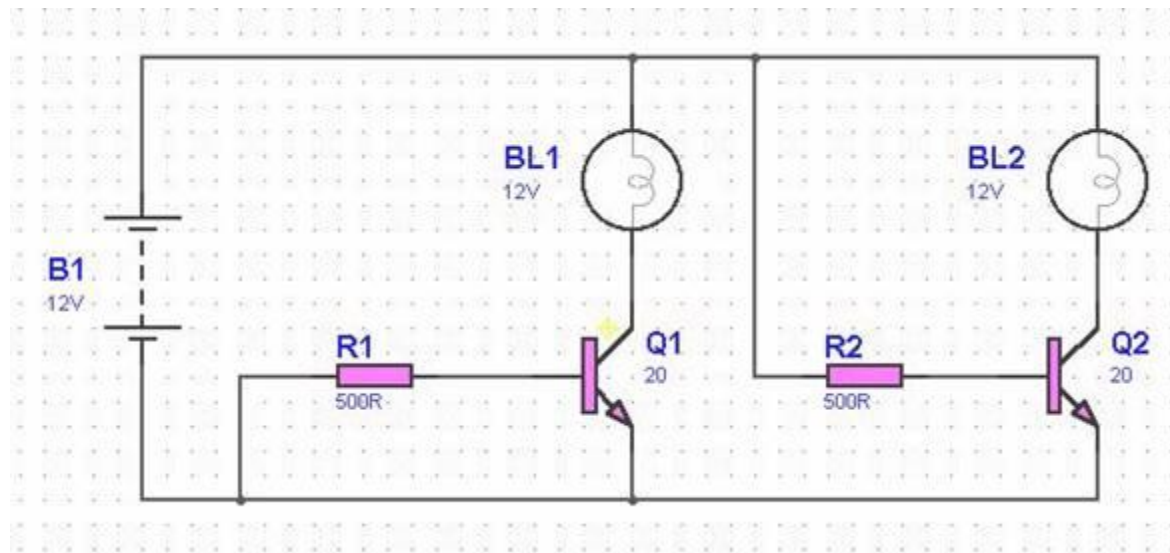
d.- Uso de reles.

Rele = ACTIVE (RELAY)

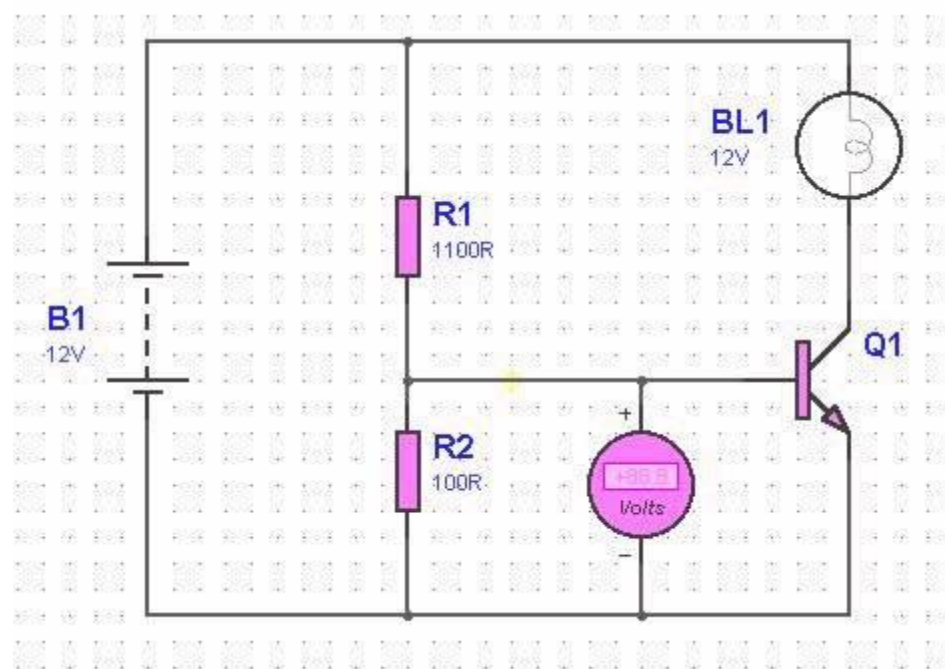


E.- Uso de transístores.

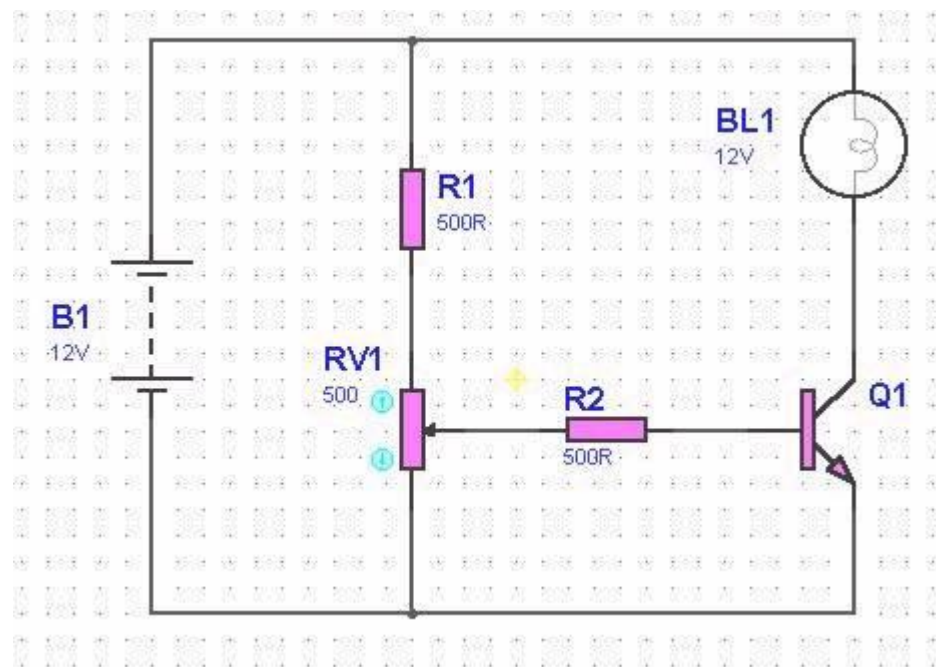
***Ativação Direta.



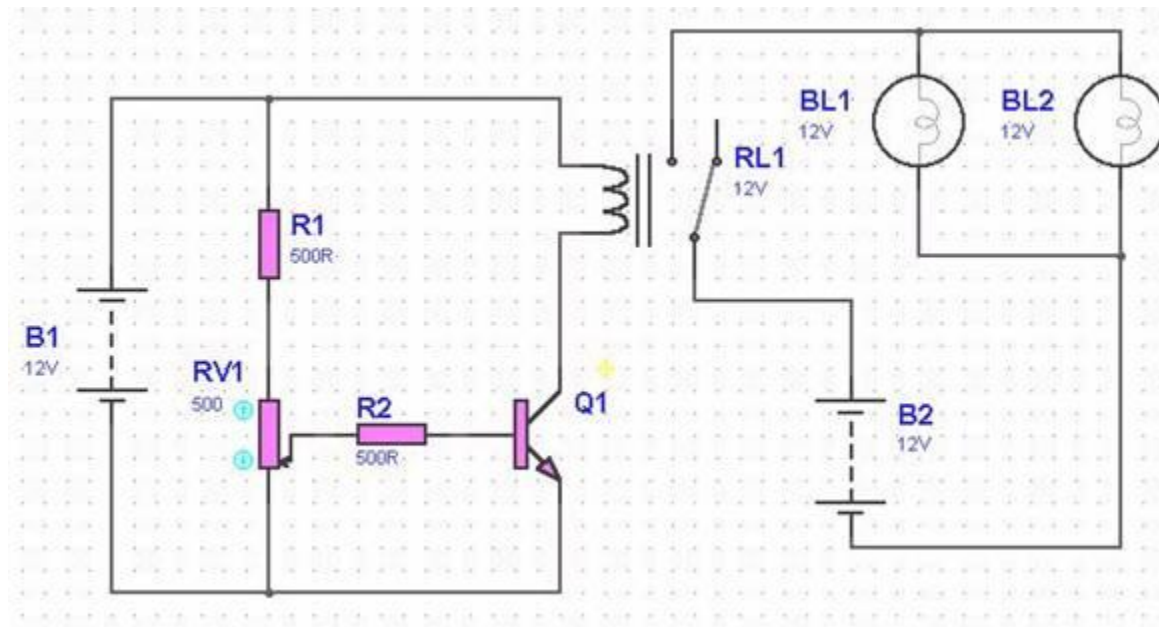
*** Usando um divisor de tensão.



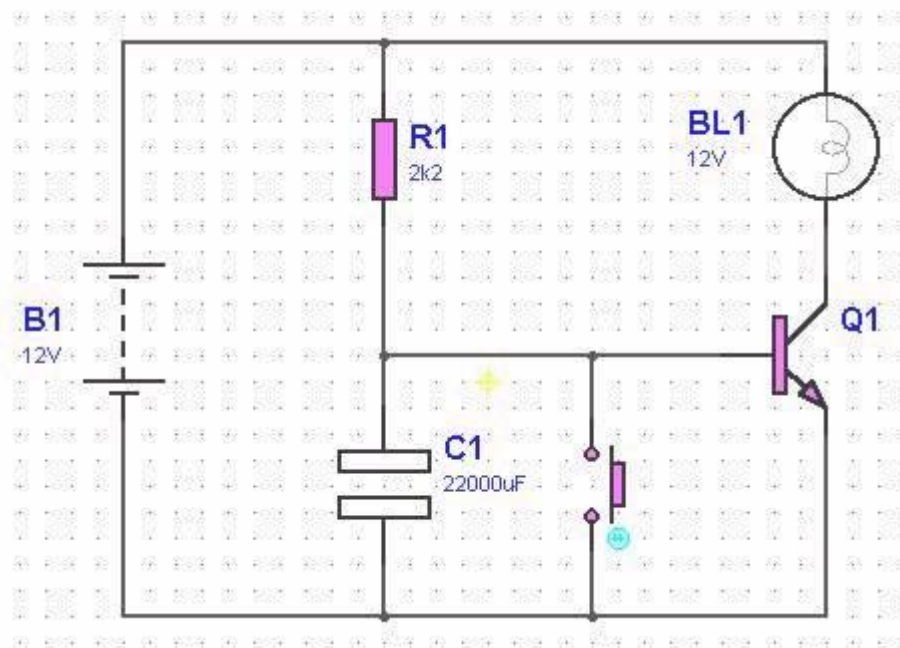
*** Com um Potenciometro.



***** Ativando um Relé.**



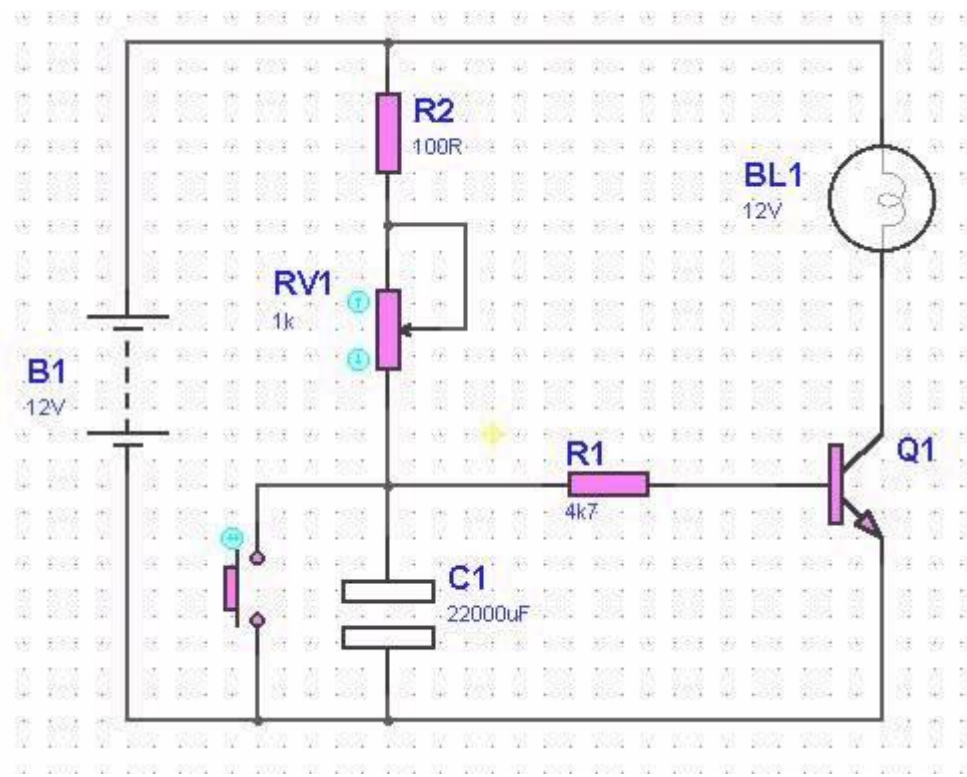
***** Usando um capacitor e resistor para criar um retardo de tempo, assim como um botão como reset do circuito.**



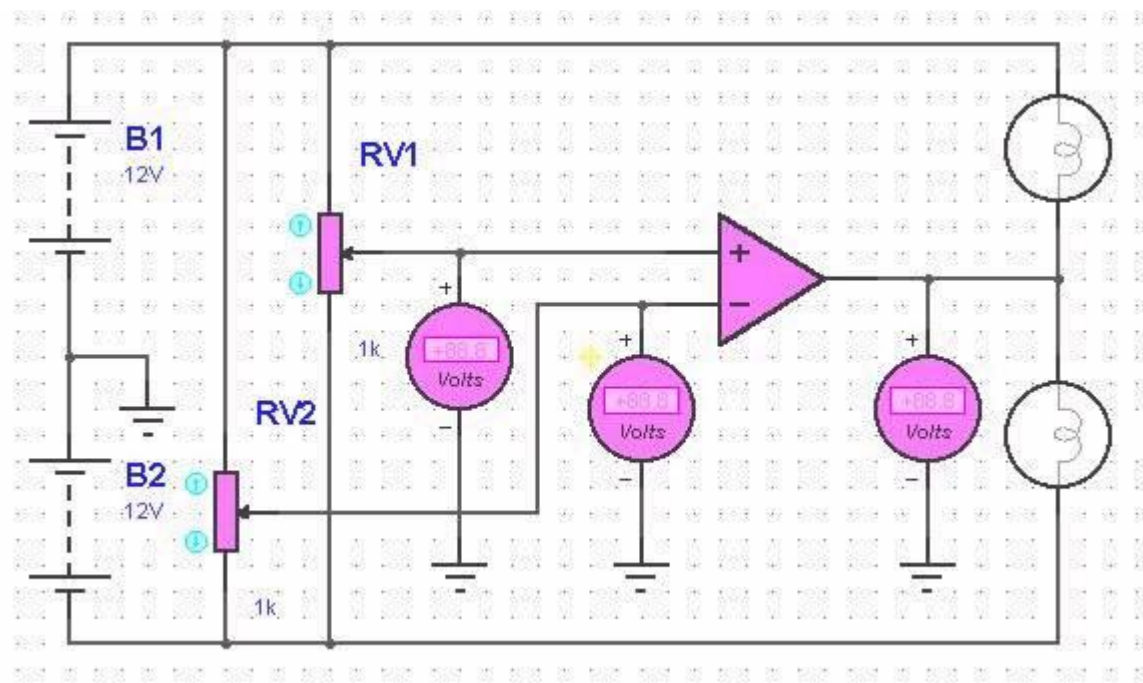
Botão = ACTIVE (BUTTON)

***** Provar o seguinte circuito e observar a mudança.**

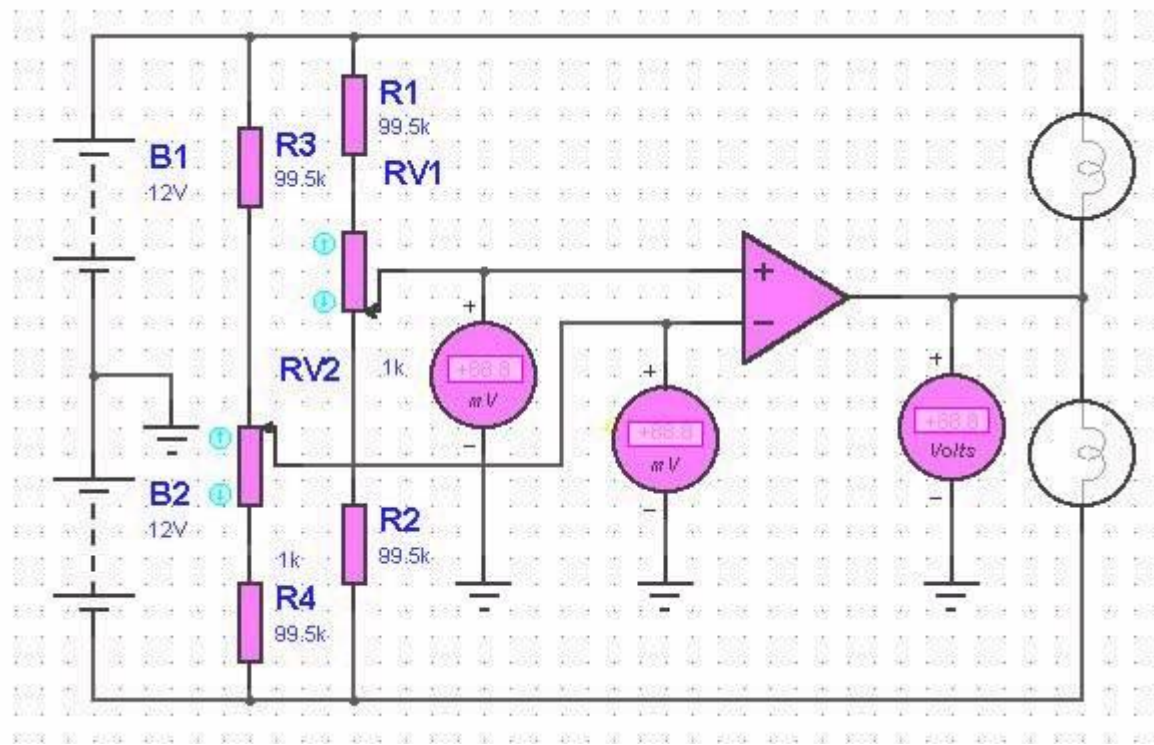
***** Circuito com retardo de aceso variável.**



F.- Uso de operacionais.

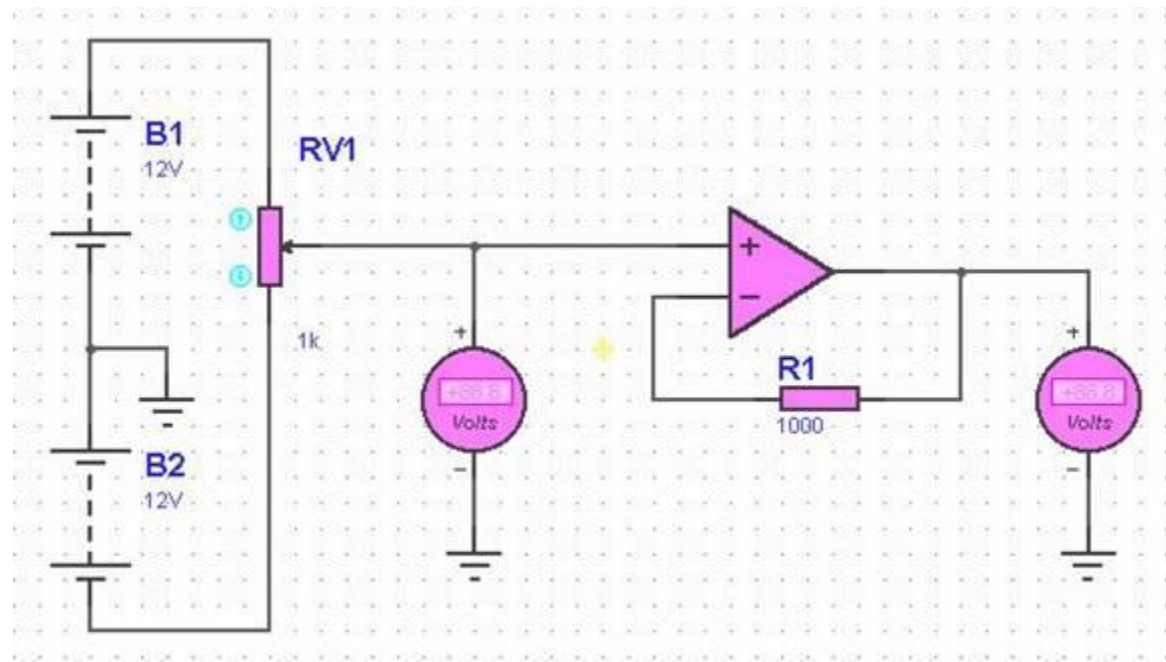


***** Circuito de controle de intensidade de cada lâmpada.**

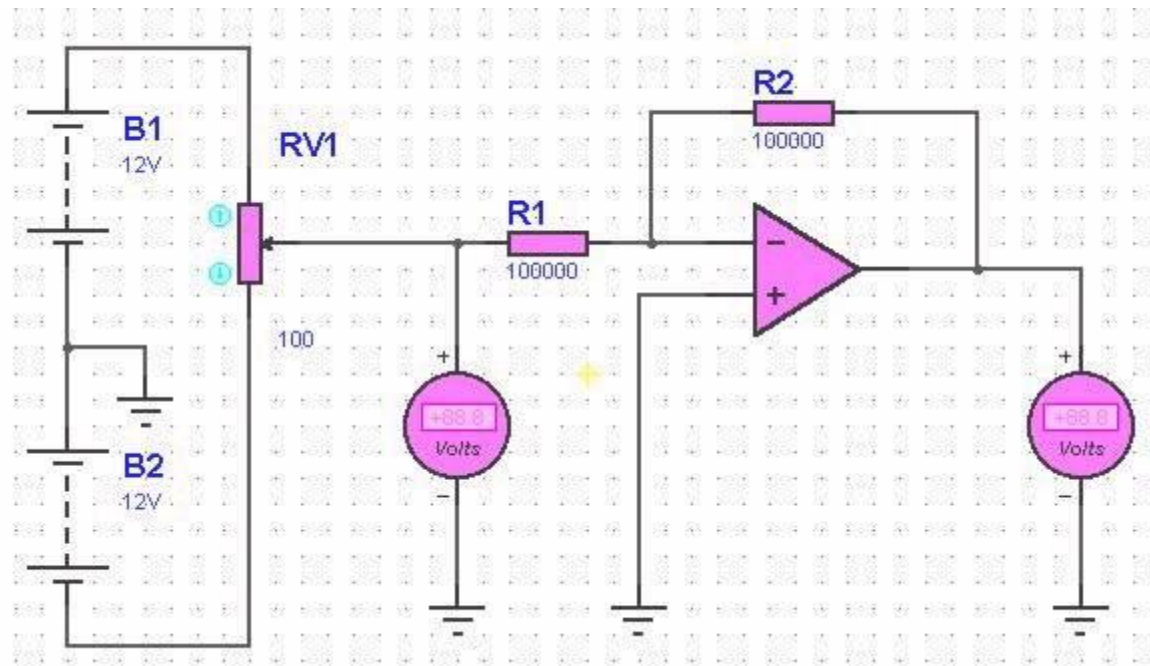


*** Exercícios com amplificadores operacionais.:

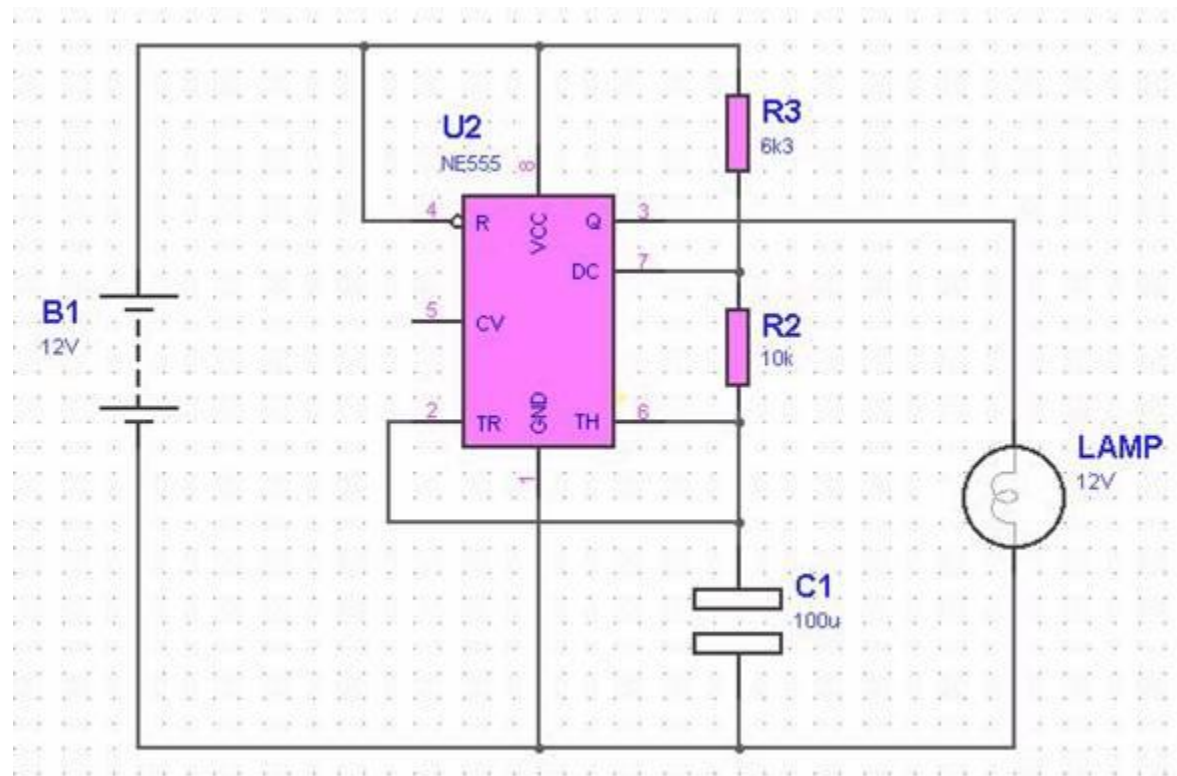
Circuito NÃO Inversor



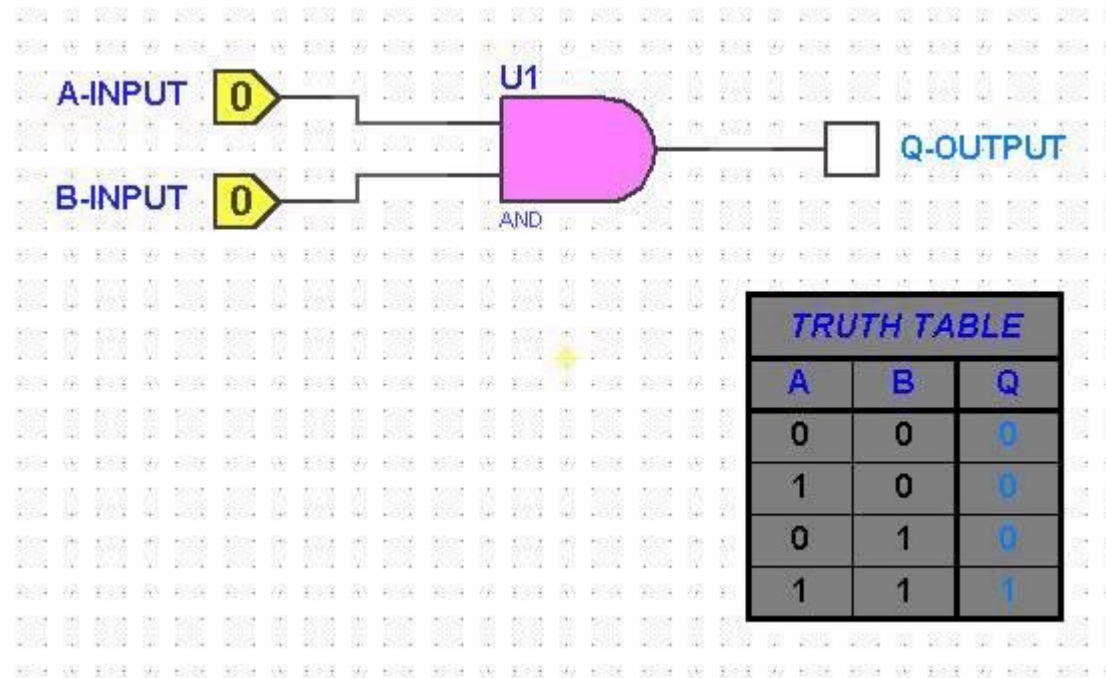
Circuito Inversor



g.- Uso do Circuito Integrado 555



***** Porta AND.**



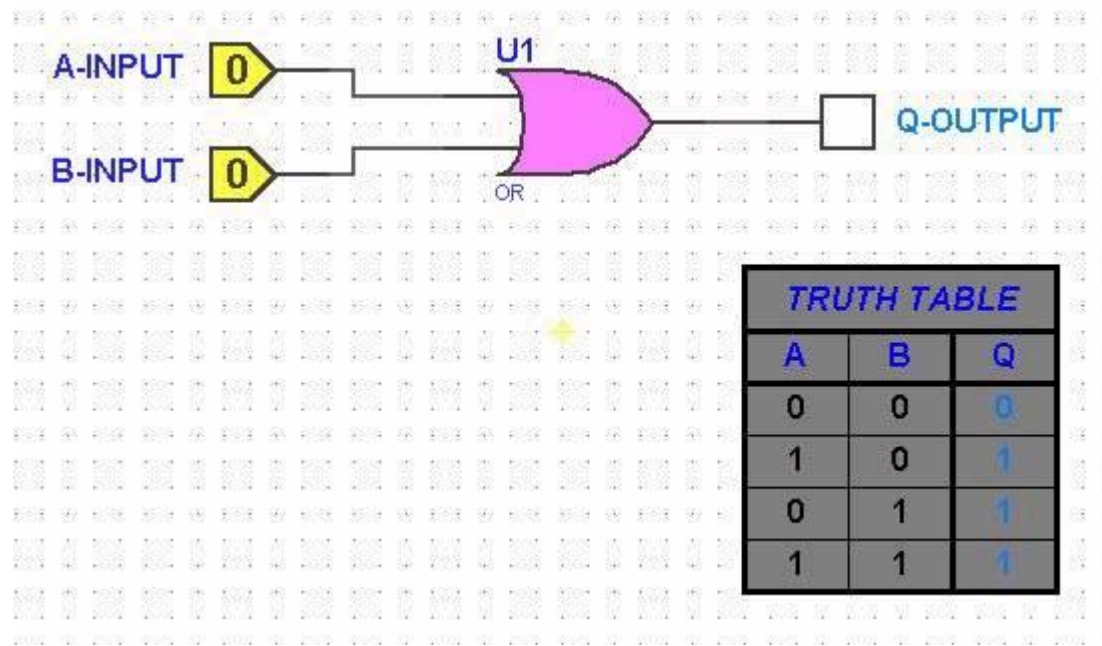
A-INPUT = ACTIVE (LOGICSTATE)

B-INPUT = ACTIVE (LOGICSTATE)

Q-OUTPUT = ACTIVE (LOGICPROBE(BIG))

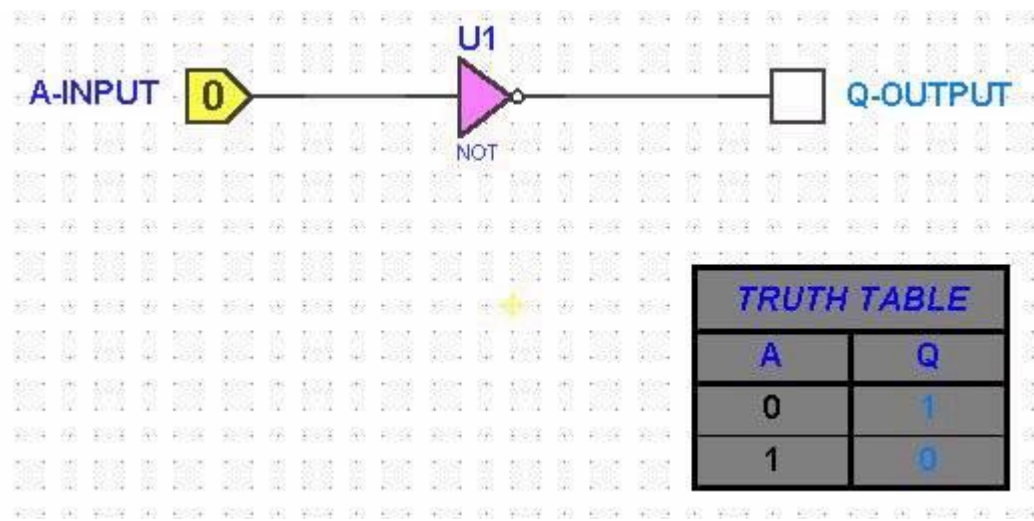
AND = ACTIVE (AND)

*** Porta OR. (Armar no mesmo circuito)



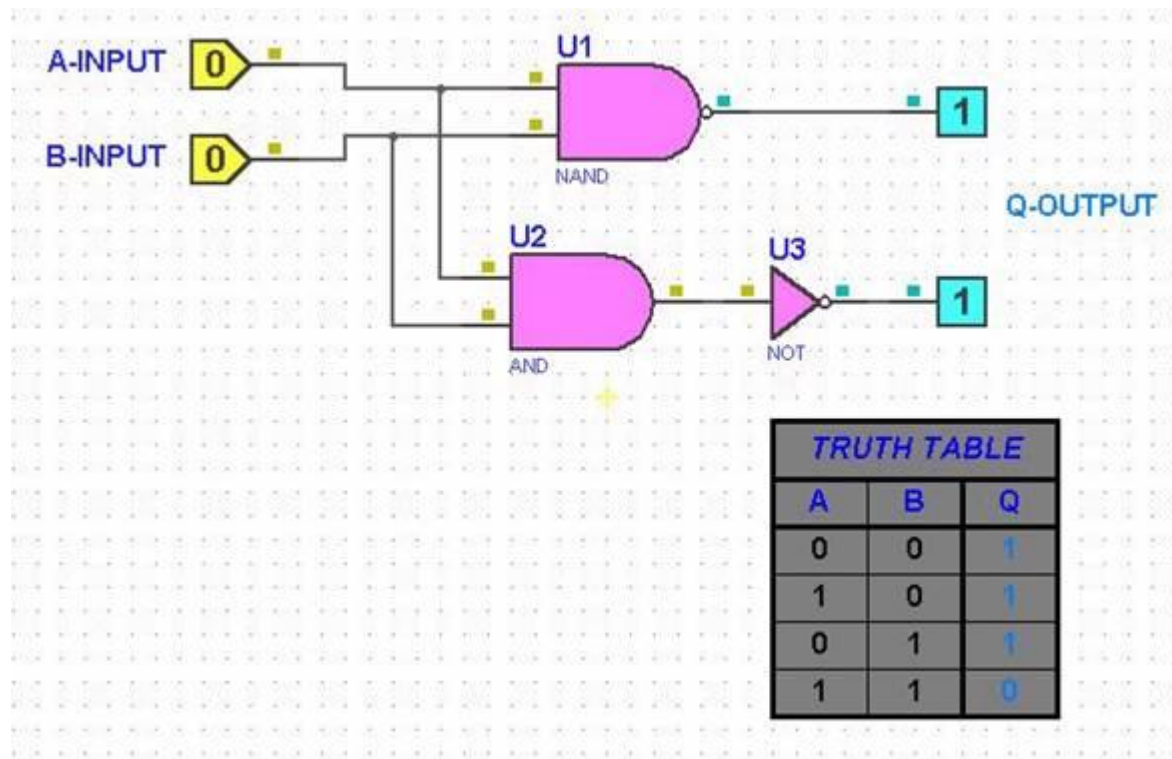
OR = ACTIVE (OR)

*** Porta NOT. (Armar no mesmo circuito)



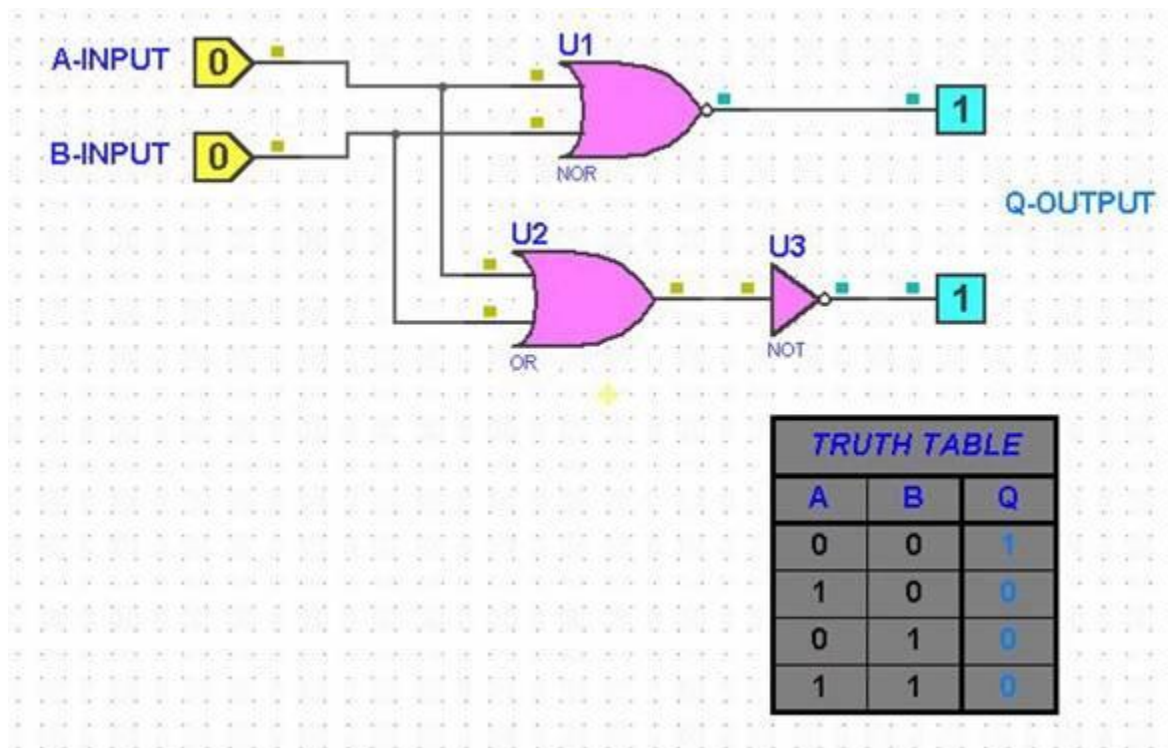
NOT = ACTIVE (NOT)

***** Pota NAND. (Armar no mesmo circuito)**



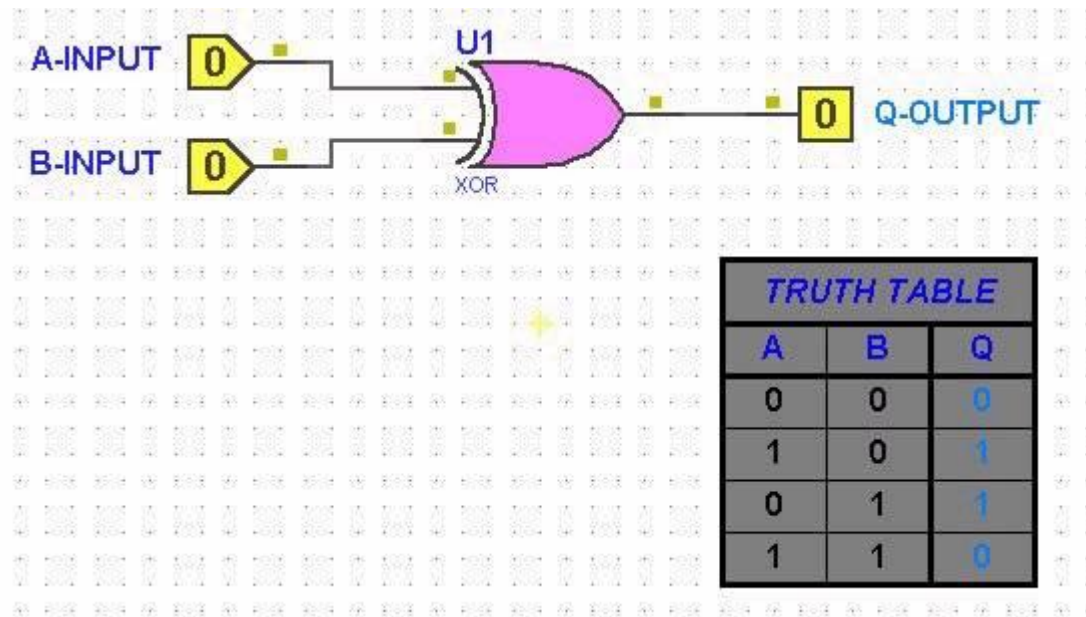
NAND = ACTIVE (NAND)

***** Porta NOR (Armar no mesmo circuito)**



NOR = ACTIVE (NOR)

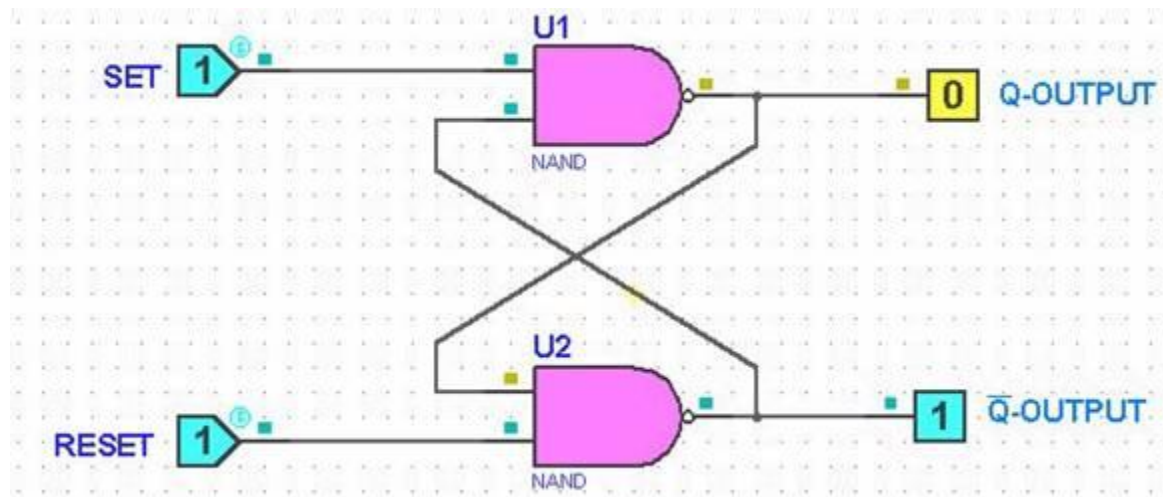
***** Porta XOR. (Armar no mesmo circuito)**



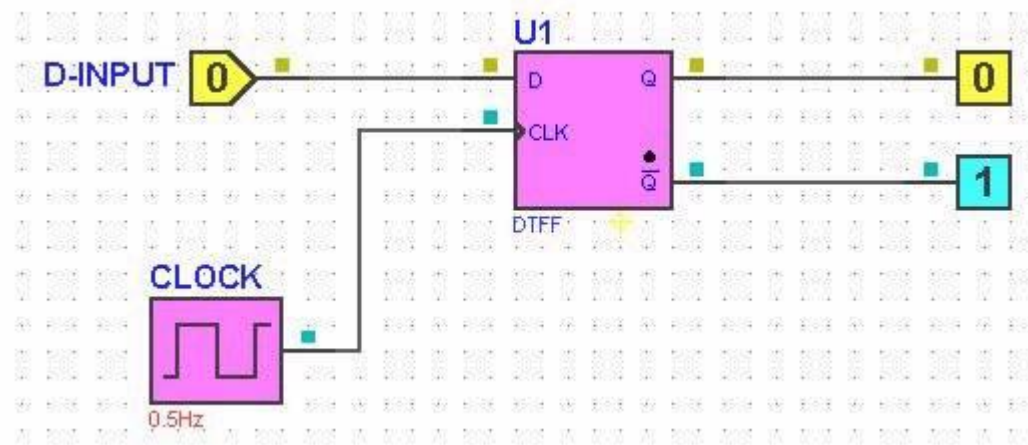
XOR = ACTIVE (XOR)

I.- Lógica Seqüencial.

***** FLIP-FLOP (RS)**



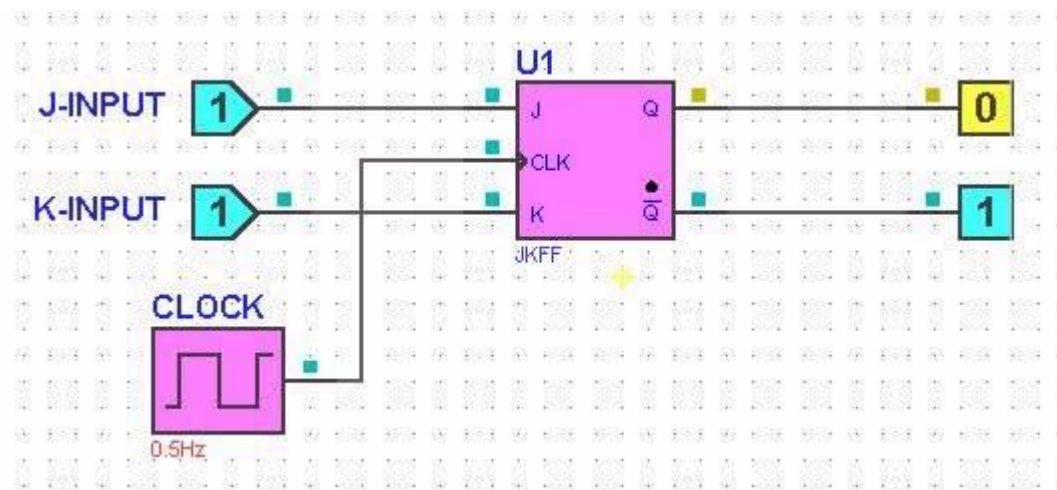
*** FLIP-FLOP (D)



DTFF = ACTIVE (DTFF)

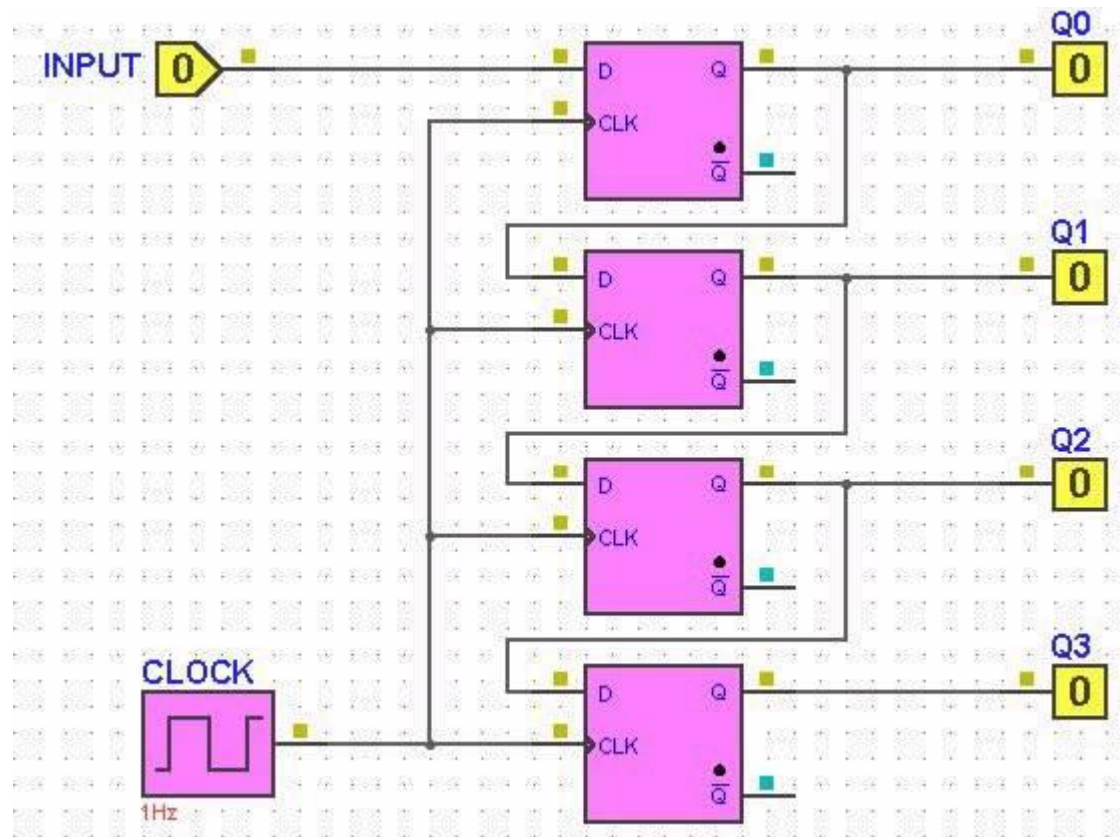
CLOCK = ACTIVE (CLOCK)

***** FLIP-FLOP (JK)**

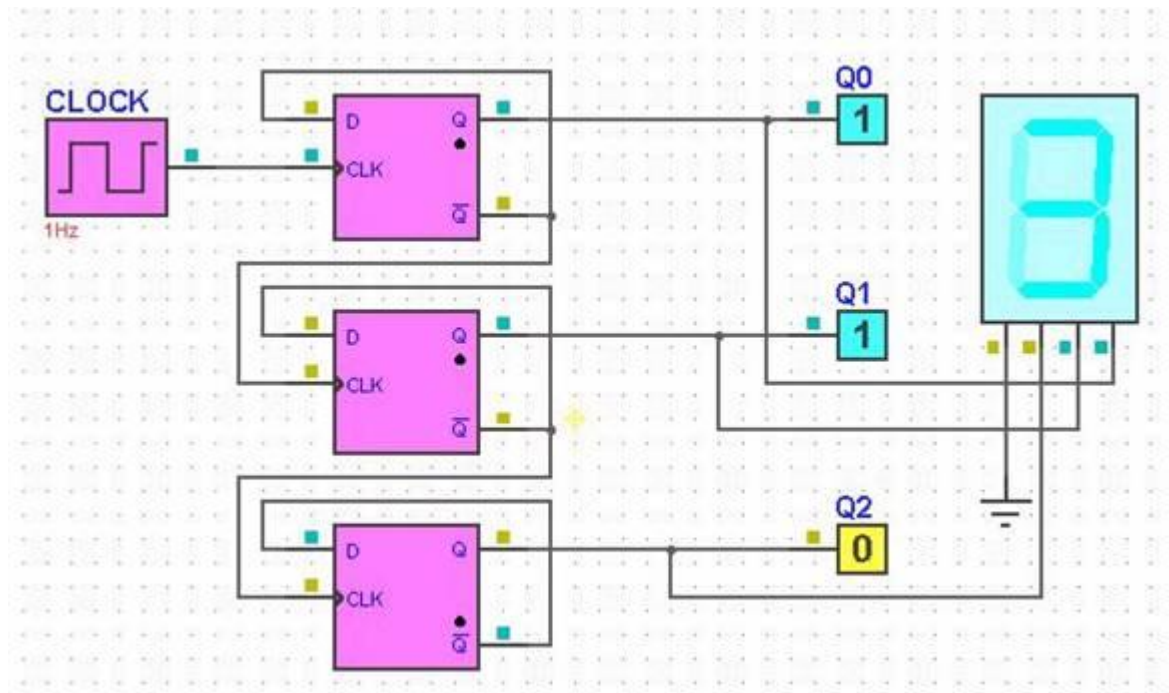


JKFF = ACTIVE (JKFF)

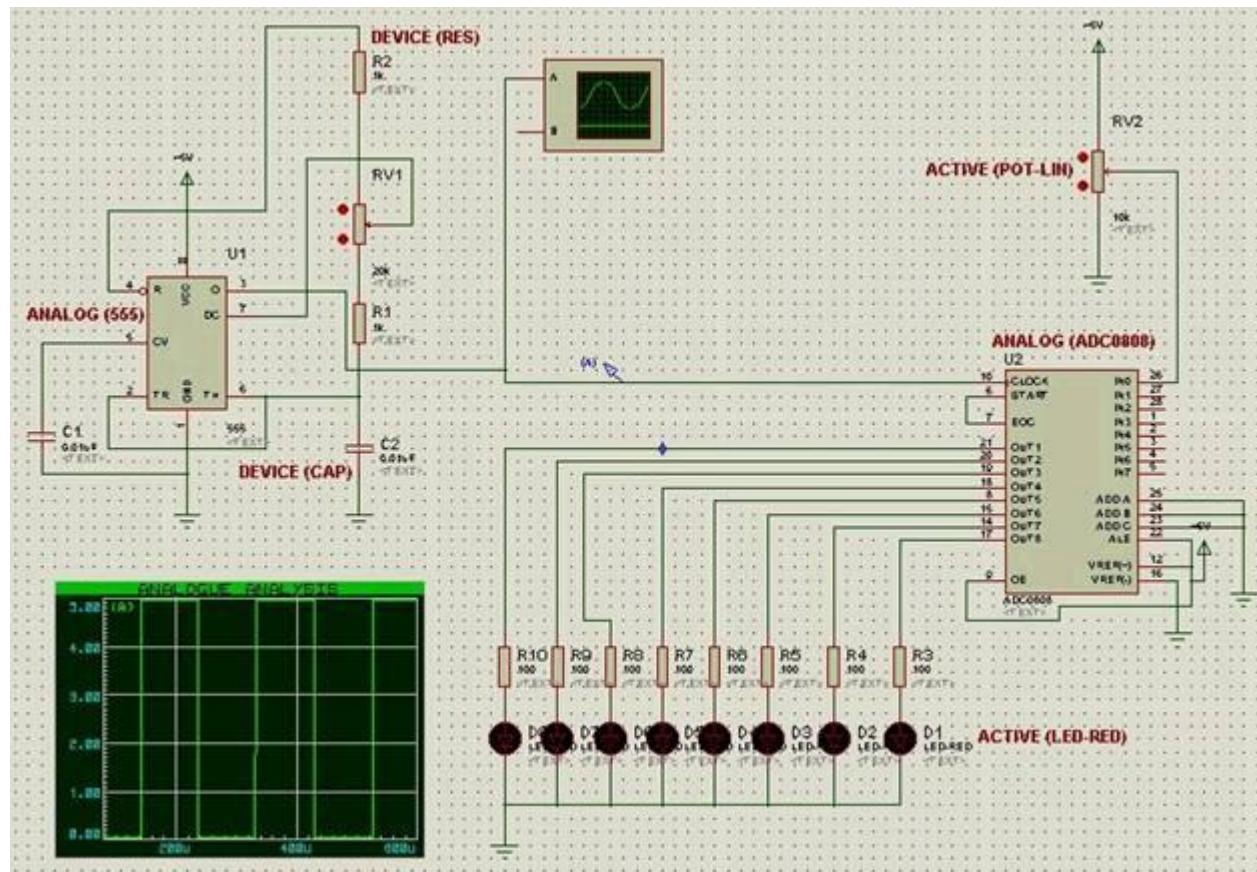
***** REGISTRO DE DESLOCAMENTO**



*** CONTADOR BINÁRIO c/ DISPLAY 7 SEGMENTOS = DISPLAY (7SEG-BCD)



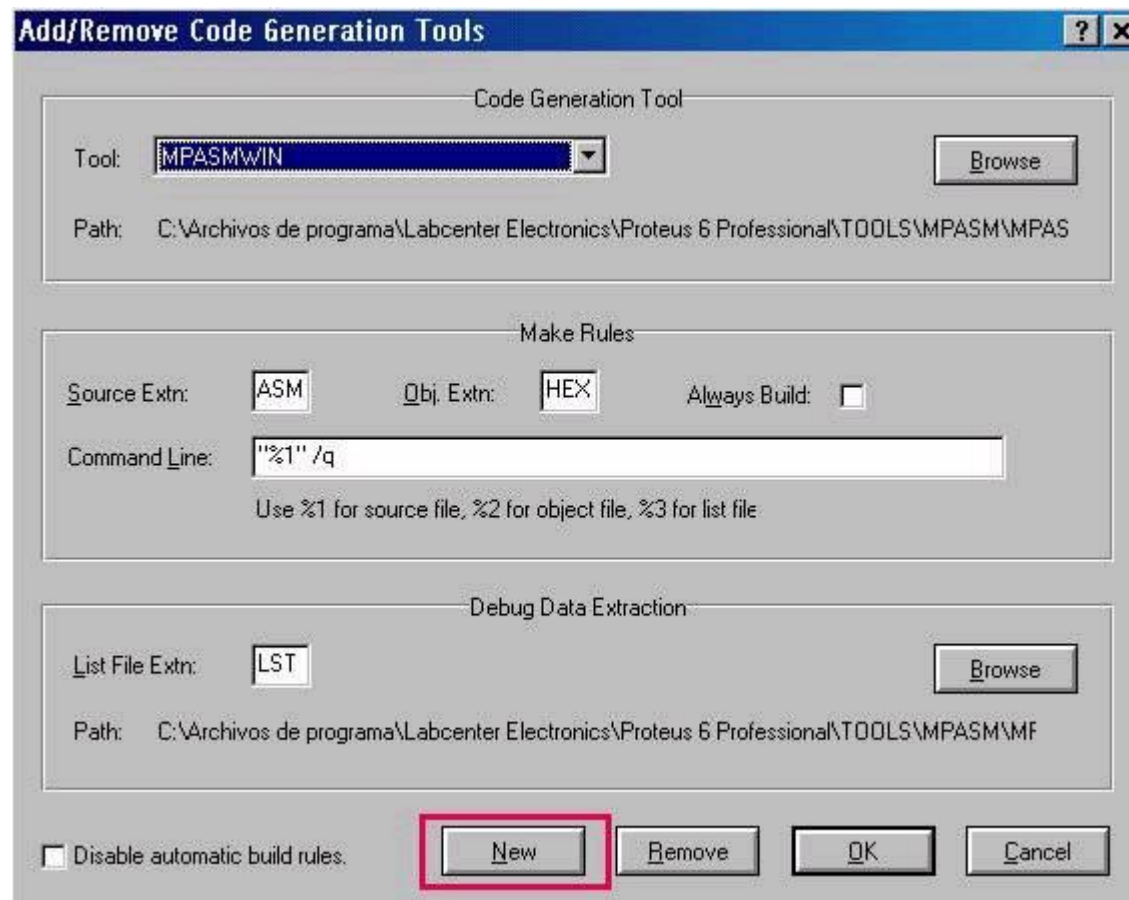
.- Uso do Conversor Analógico Digital.



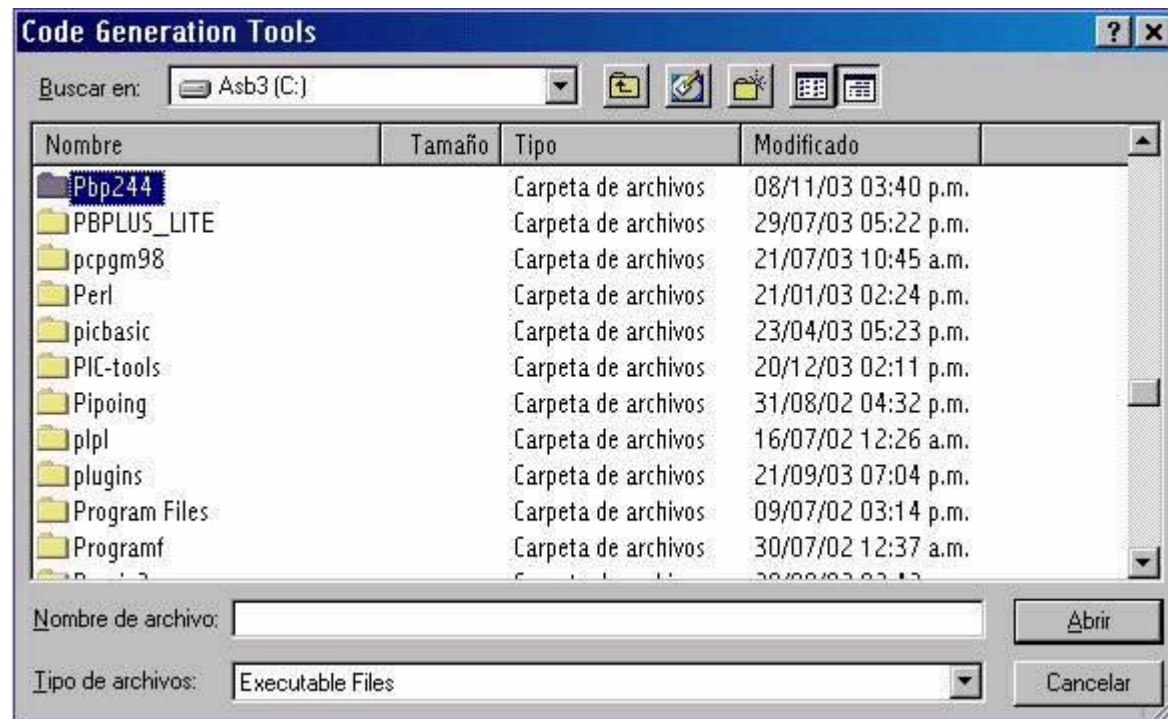
K.- Habilitação do compilador PIC BASIC PRO.

I.- Ingressar no menu Source.

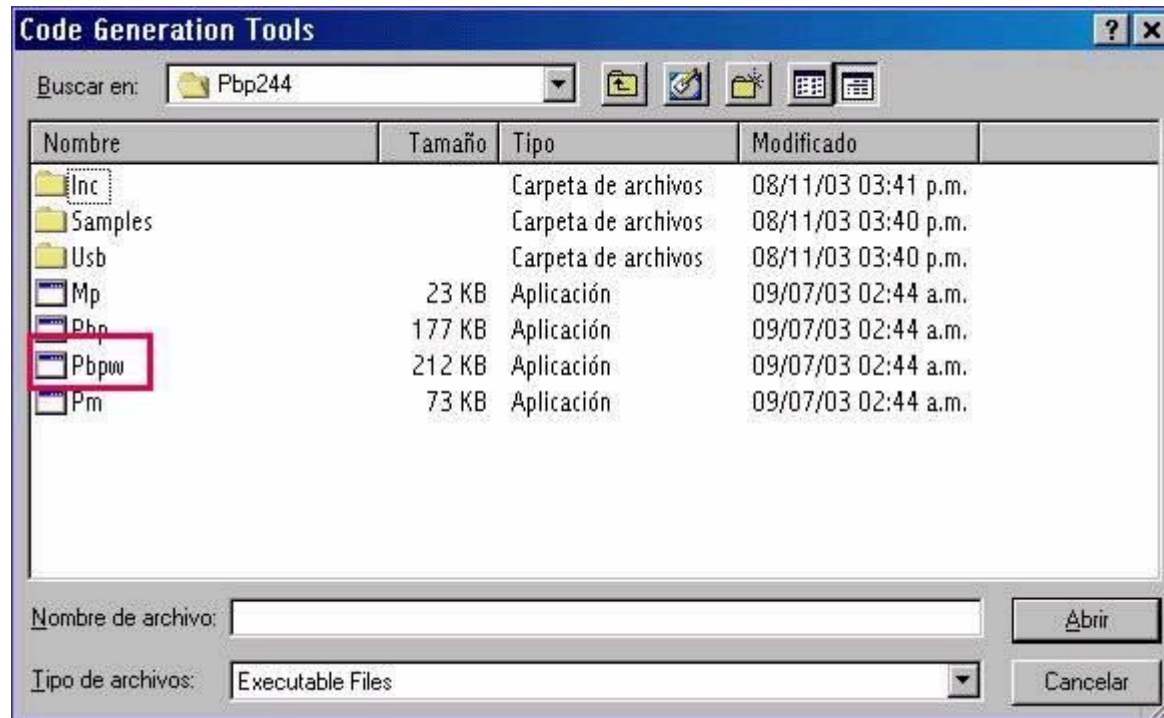
II.- Selecionar Define Code Generation Tools... e pressionar o botão NEW.



III.- Procurar a pasta na raiz de "C" com o nome Pbp244.



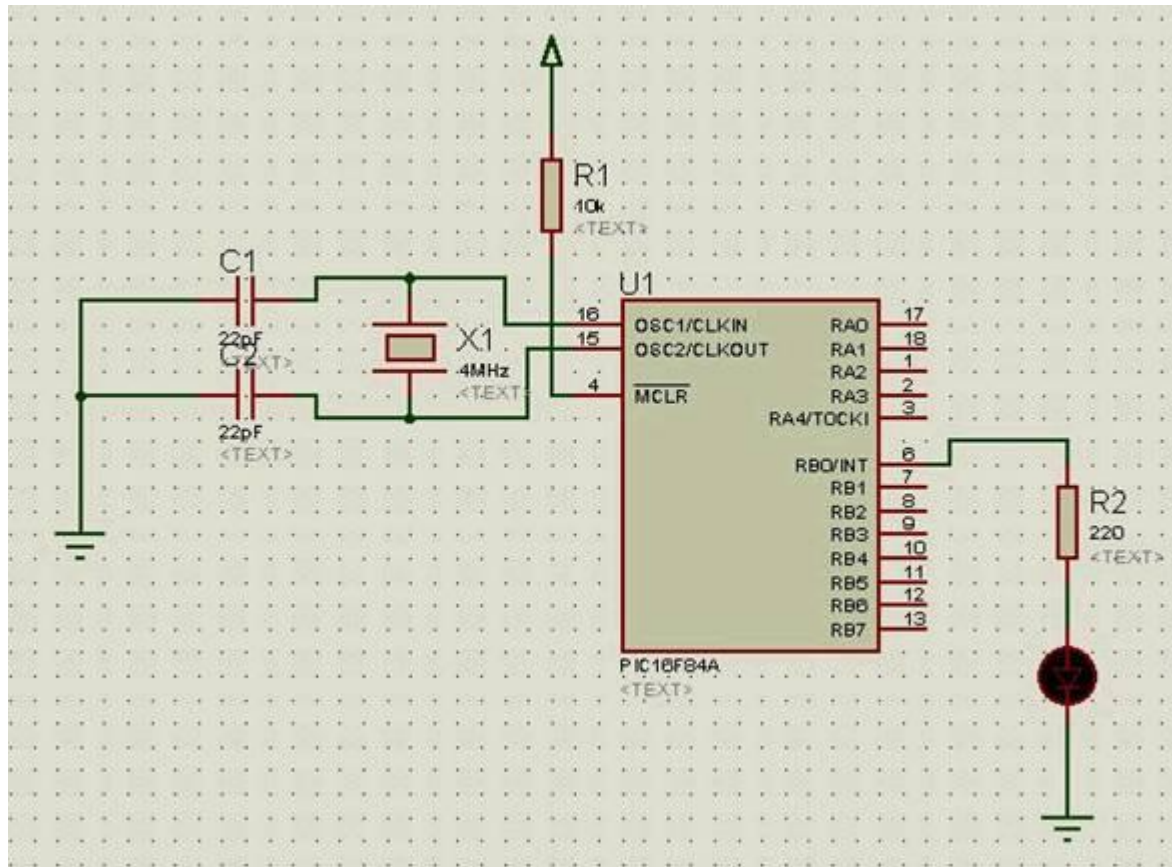
IV.- Ingressar na pasta Pbp244 e selecionar o arquivo Pbpw.



V.- Completar a forma como se mostra na seguinte figura e pressionar OK.

L.- Prova de funcionamento do programa BLINK. faz-se piscar um led conectado a PORTB.0

I.- Criar o seguinte circuito.



DEVICE (CAP)

DEVICE (CRYSTAL)

DEVICE (RES)

MICRO (PIC16F84A)

ACTIVE (LED-RED)

II.- Salvar o circuito em uma pasta com o nome Blink e nomear ao arquivo Blink.

III.- Copiar o seguinte texto ao NOTEPAD do Windows e salva-lo em uma arquivo de texto com o nome Blink.bas dentro da pasta Blink.

'Programa de exemplo que faz piscar a um LED conectado a PORTB.0
'cada segundo.

loop: High PORTB.0 'Acende o LED
Pause 500 'Retardo de .5 segundos

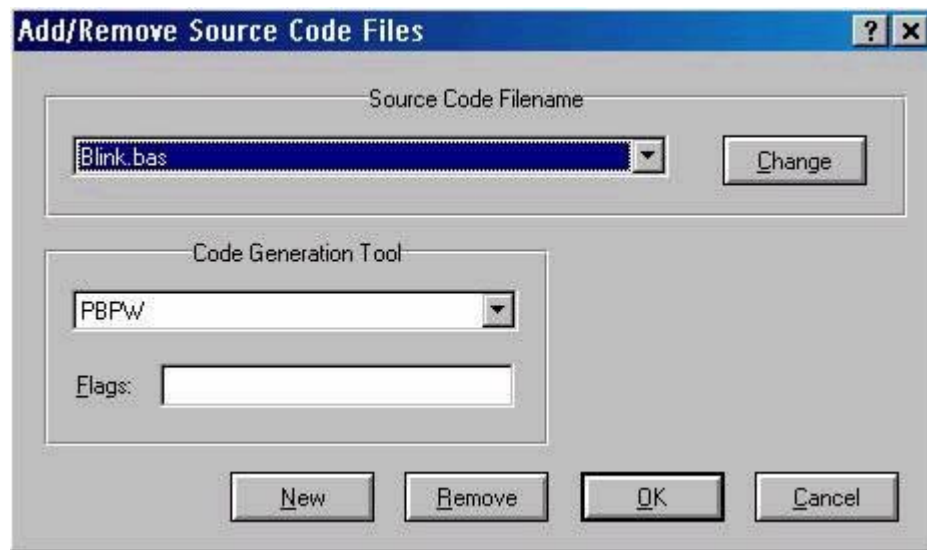
Low PORTB.0 'Apaga o LED
Pause 500 'Retardo de .5 segundos

Goto loop 'Retorna ao loop e o LED fica
'piscando.

End

IV.- Ingressar no menu Source.

V.- Selecionar Add/Remove Source Code Files...



VI.- No Code Generation Tools escolher PBPW.

VII.- Pressionar o botão NEW.

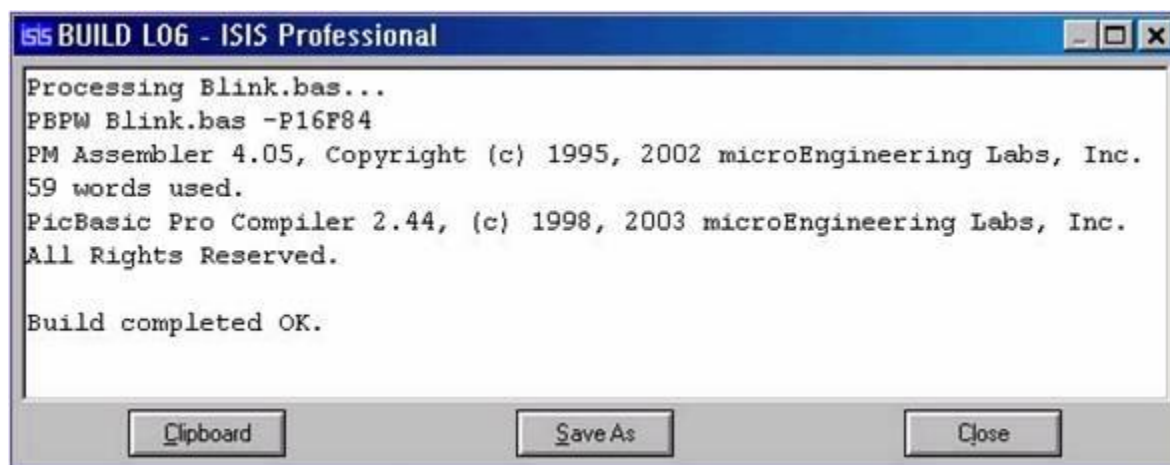
VIII.- Selecionar o arquivo Blink.bas.

IX. Pressionar OK.

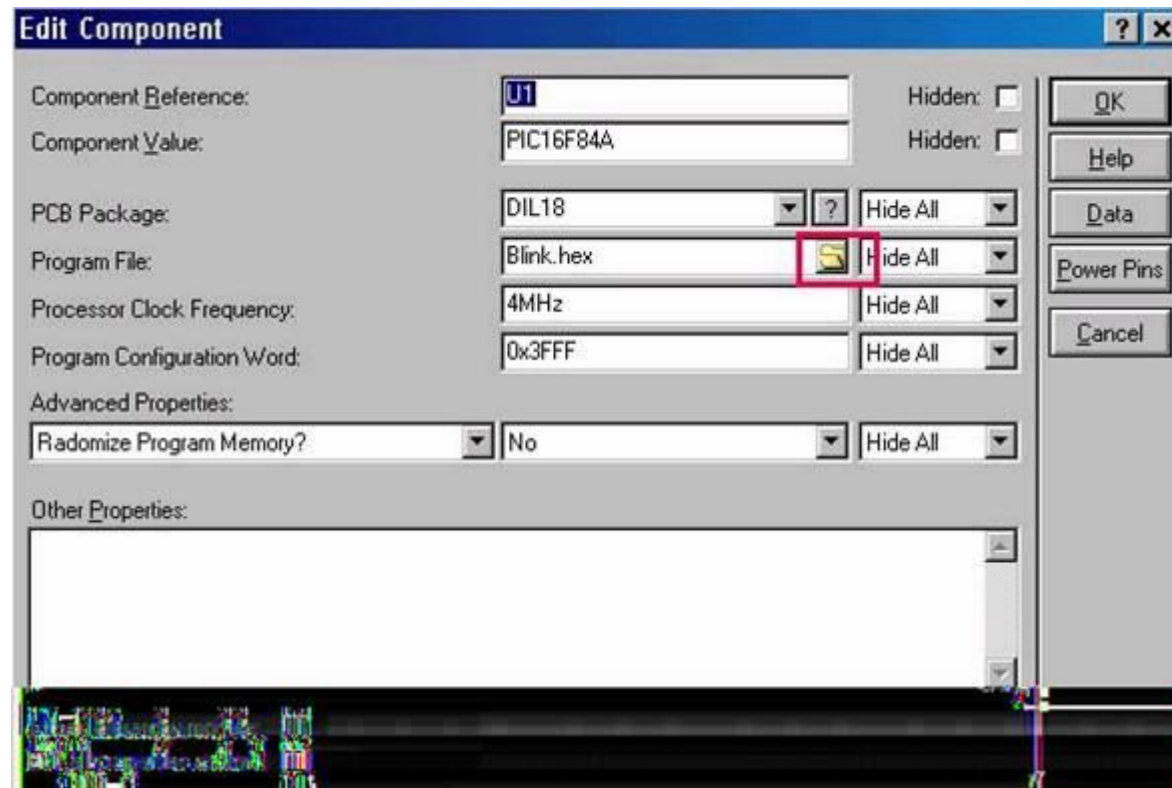
NOTA: Para que funcione todo o anterior deve haver-se incluído no PATH do sistema Windows a rota à pasta C:\PBP244, isto se consegue editando o arquivo AUTOEXEC.BAT, no que se adiciona ao final do PATH existente.

Para o Windows 2000 e XP o procedimento é diferente. tem-se que ingressar na forma System, seleccionar Opções Avançadas e terá que declarar uma nova variável de entorno. Este pendente verificar estes passos.

X.- Não menu Source escolher Build All. Se tudo esta correto deve aparecer a seguinte janela a qual se pode fechar pressionando CLOSE.

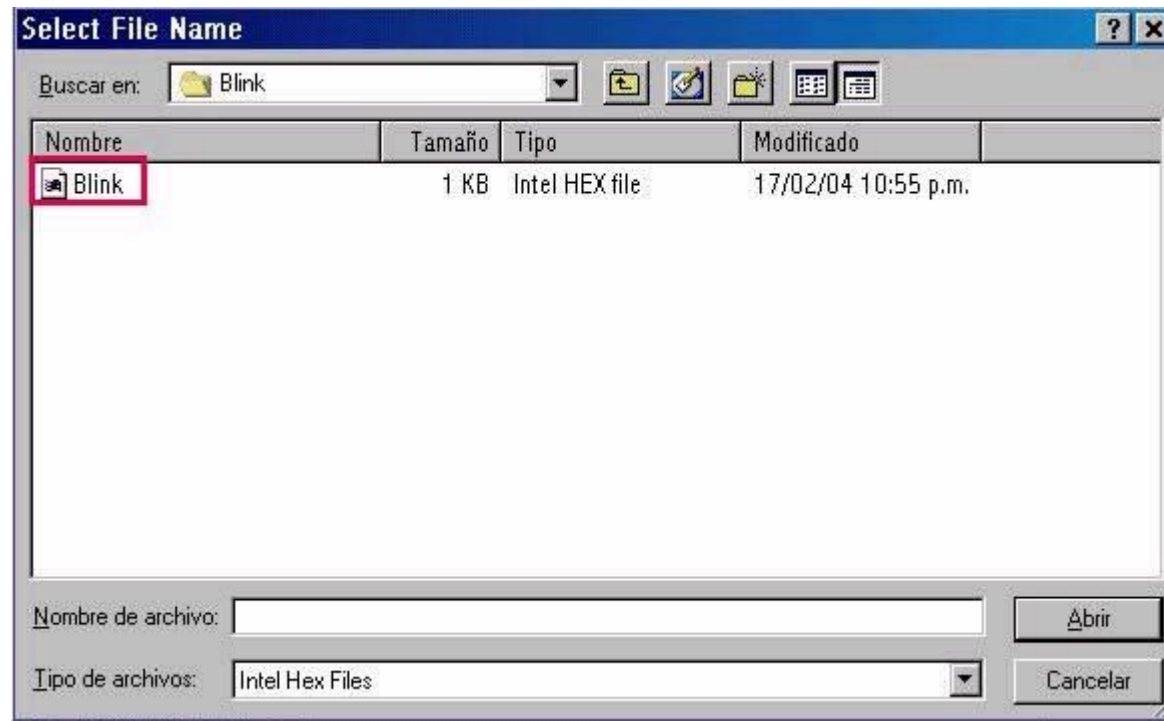


XI.- Dar click com o botão direito sobre o microcontrolador e depois outro com o esquerdo para abrir a forma de configuração.



XII.- Dar um click com o botão esquerdo no ícone que mostra uma pasta aberta (que marca o retângulo vermelho).

XIII.- Selecionar o arquivo Blink.hex.



XIV.- Ajustar a frequência do relógio a 4MHz no campo Processor Clock Frecuency.

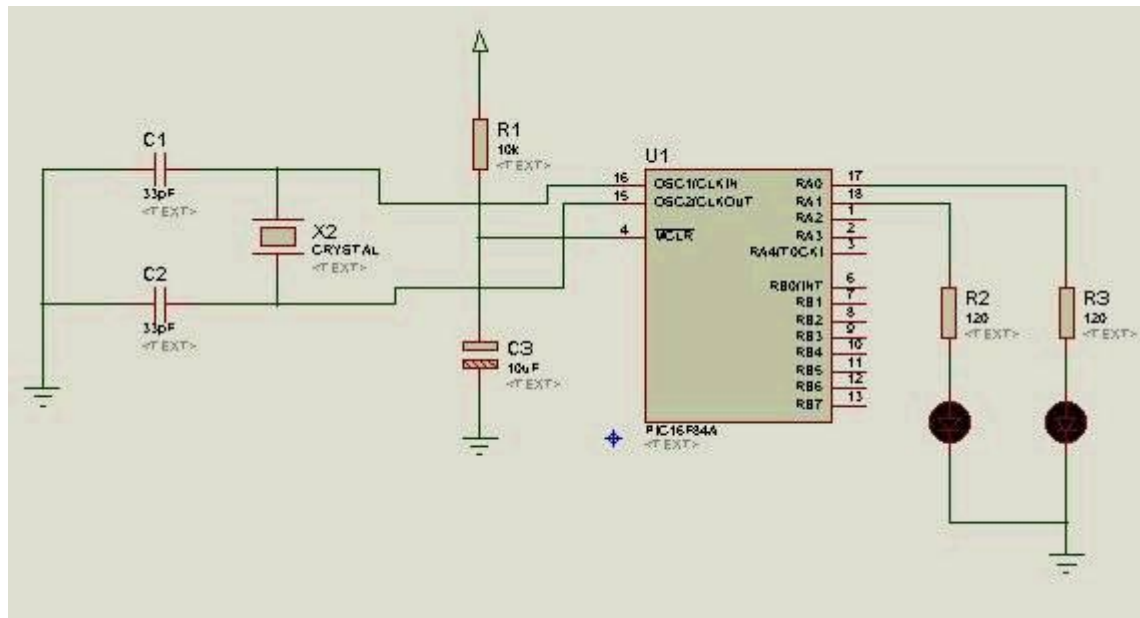
XV.- Pressionar OK.

XVI.- Executar a simulação.

PROTEUS VSM 4

M.- Exemplos com microcontroladores Pic da MICROCHIP.

I.- Microcontrolador PIC16F84 com dois LEDS.

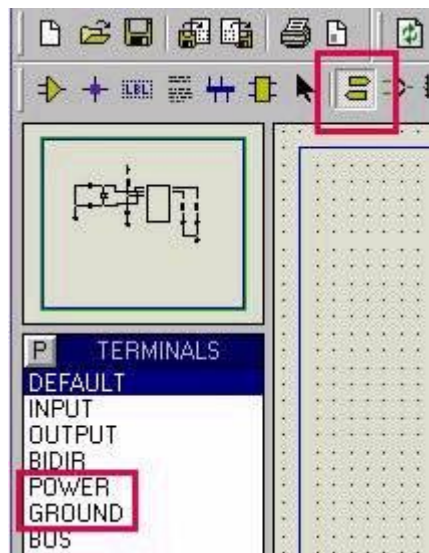


Componentes:

- DEVICE (CAP) = 33pF
- DEVICE (CRYSTAL) = 4MHz
- DEVICE (RES) = 10k, 120, 120

- **DEVICE (CAP-ELEC) = 10uF**
- **MICRO (PIC16F84A)**
- **ACTIVE (LED-GREEN)**
- **ACTIVE (LED-RED)**

A alimentação e o terra para este circuito se encontra dentro do Inter-Sheet Terminal da barra de ferramentas.



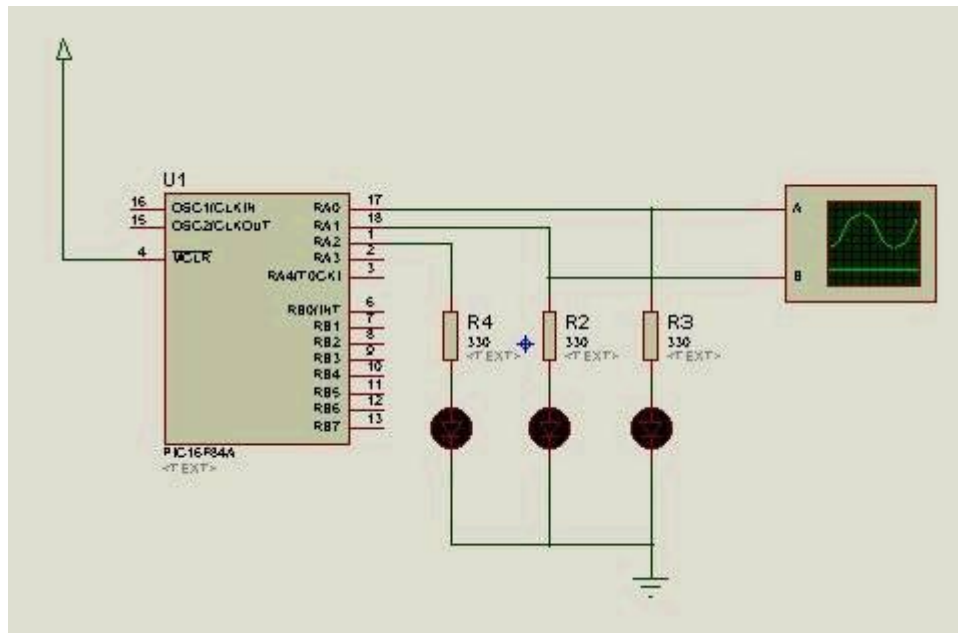
Programa:

```
TRISA = 0  
PORTA.0 = 1  
PORTA.1 = 0
```

ASB:
TOGGLE PORTA.0
TOGGLE PORTA.1
PAUSE 500
GOTO ASB

END

*** Provar o seguinte circuito:



Programa:

```
TRISA = 0  
LOW PORTA.0  
HIGH PORTA.1  
LOW PORTA.2
```

```
ASB:  
TOGGLE PORTA.0  
TOGGLE PORTA.1  
TOGGLE PORTA.2  
PAUSE 1000  
GOTO ASB
```

```
END
```

ii.- Circuito com interruptores de 2 posições.

Programa:

```
TRISB = %11111111  
TRISA = 0
```

```
SWS VAR BYTE
```

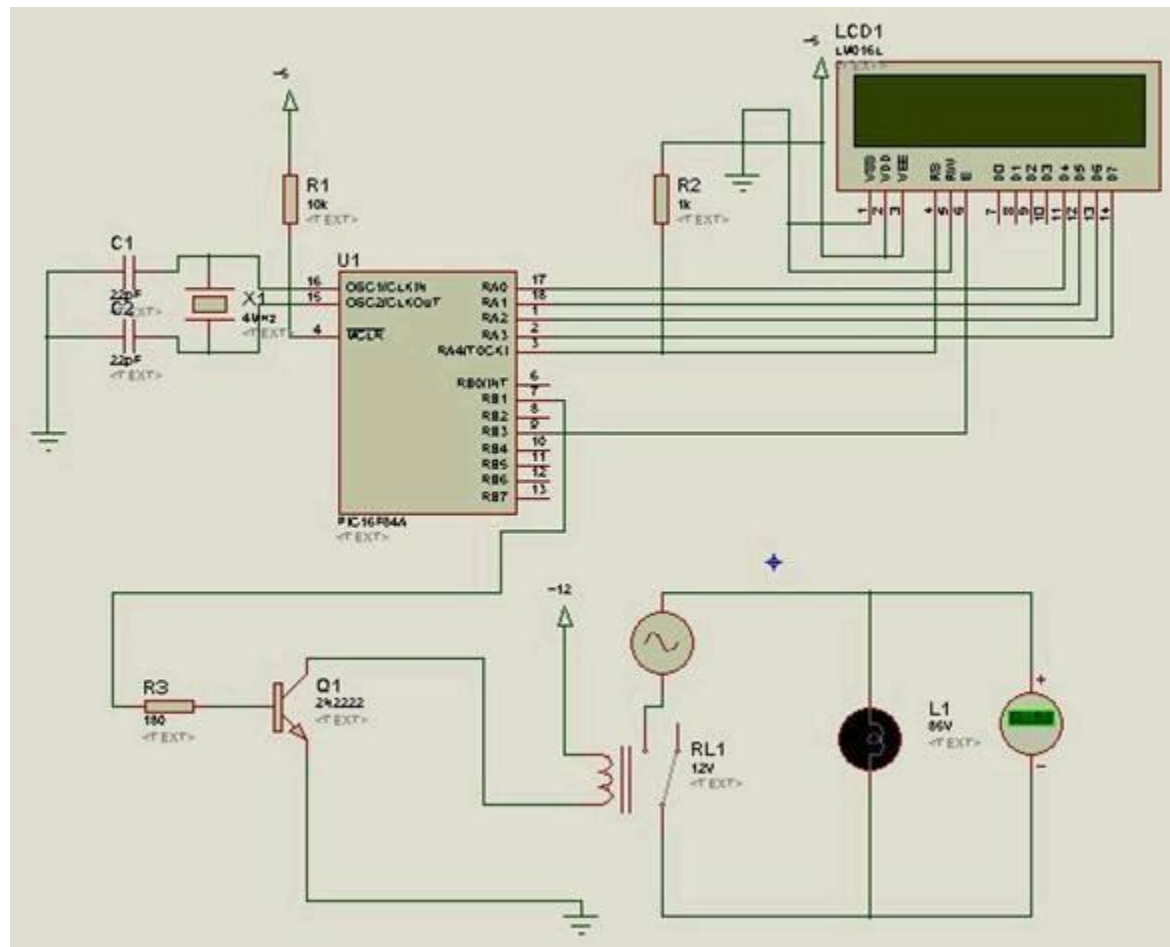
```
OPTION_REG = 0
```

```
ASB:  
SWS = PORTB & %00001110
```

```
SELECT CASE SWS
CASE %1110
LOW PORTA.0
CASE %1100
HIGH PORTA.0
CASE %1010
HIGH PORTA.0
CASE %1000
LOW PORTA.0
CASE %0110
HIGH PORTA.0
CASE %0100
LOW PORTA.0
CASE %0010
LOW PORTA.0
CASE %0000
HIGH PORTA.0
END SELECT
PAUSE 100
GOTO ASB

END
```

iii.- Uso do LCD Alfanumérico e ativação de uma carga de corrente alterna por meio de um transistor e um rele.



Componentes:

- **BIPOLAR (2N2222)**
- **DISPLAY (LM016L)**

Programa:

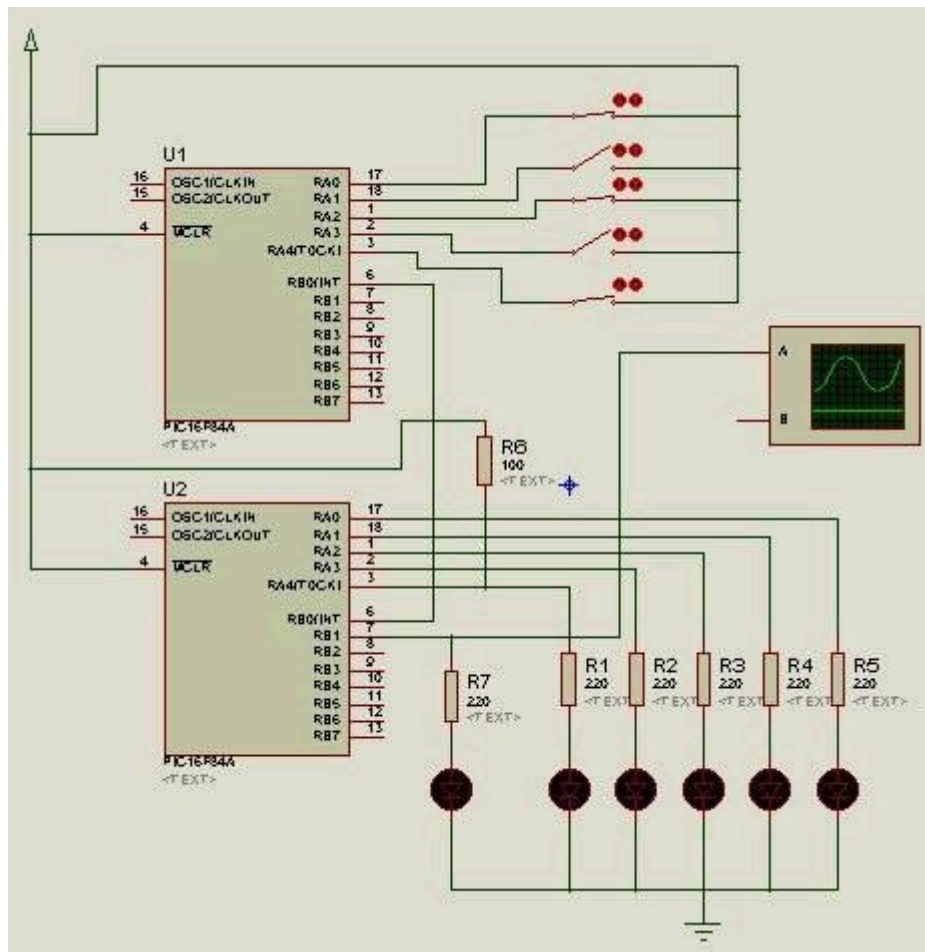
```
' Set LCD Data port
DEFINE LCD_DREG PORTA
    'Set starting Data bit (0 or 4) if 4-bit bus
DEFINE LCD_DBIT 0
    ' Set LCD Register Select port
DEFINE LCD_RSREG PORTA
    ' Set LCD Register Select bit
DEFINE LCD_RSBIT 4
    ' Set LCD Enable port
DEFINE LCD_EREG PORTB
    ' Set LCD Enable bit
DEFINE LCD_EBIT 3
    ' Set LCD bus size (4 or 8 bits)
DEFINE LCD_BITS 4
    ' Set number of lines on LCD
DEFINE LCD_LINES 2
    ' Set command delay time in us
DEFINE LCD_COMMANDUS 2000
    ' Set data delay time in us
DEFINE LCD_DATAUS 50
```

```
TRISB.1 = 0
```

```
ASB: LCDOUT $FE, 1, "HOLA ALUMNO"
PAUSE 1000
```

```
LCDOUT $FE, $C0, "PROTEUS"  
PAUSE 1000  
TOGGLE PORTB.1  
GOTO ASB  
END
```

iV.- Comunicação serial entre dois microcontroladores PIC16F84.



Programas:

[Picuno.bas]

```
INCLUDE "modedefs.bas"
```

```
TRISA = 255
```

```
ASB:
```

```
SEROUT PORTB.0, T9600, ["OKy",PORTA]
```

```
GOTO ASB
```

```
END
```

```
[Picdos.bas]
```

```
INCLUDE "modedefs.bas"
```

```
TRISA = 0
```

```
ASB:
```

```
SERIN PORTB.0, T9600, ["OKy"], PORTA
```

```
TOGGLE PORTB.1
```

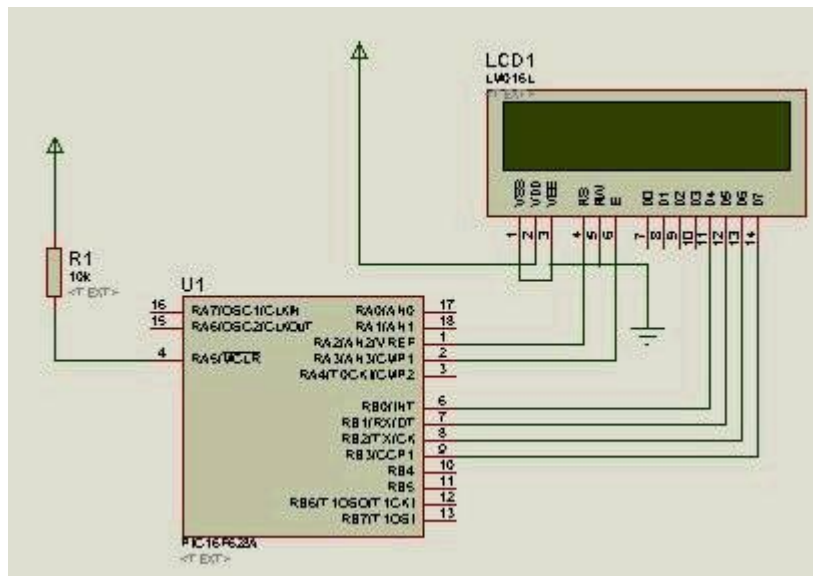
```
GOTO ASB
```

```
END
```

V.- Uso do microcontrolador PIC16F628

NOTA: modificar o micro utilizado em Define Code Generation Tools... dentro do menu Source.

Circuito:



Componentes:

- MICRO (PIC16F628A)

Programa:

```

DEFINE
CMCON          =          7          '          OSC          PortA          =          digital          4
VRCON = 0 ' A/D Voltage reference disabled

DEFINE
'          Set          LCD          Data          port
LCD_DREG          (0          or          4)          if          4-bit          PORTB          bus
'          Set          starting          Data          bit          (0          or          4)          if          4-bit          bus

```

```

DEFINE                                LCD_DBIT                                0
                                '          Set          LCD          Register          Select          port
DEFINE                                LCD_RSREG                                PORTA
                                '          Set          LCD          Register          Select          bit
DEFINE                                LCD_RSBIT                                2
                                '          Set          LCD          Enable          port
DEFINE                                LCD_EREG                                PORTA
                                '          Set          LCD          Enable          bit
DEFINE                                LCD_EBIT                                3
                                '          Set          LCD          bus          size          (4          or          8          bits)
DEFINE                                LCD_BITS                                4
                                '          Set          number          of          lines          on          LCD
DEFINE                                LCD_LINES                                2
                                '          Set          command          delay          time          in          us
DEFINE                                LCD_COMMANDUS                                2000
                                '          Set          data          delay          time          in          us
DEFINE LCD_DATAUS 50

PAUSE 20

LCDOUT $FE, 1, "NOMBRE"

ASB:

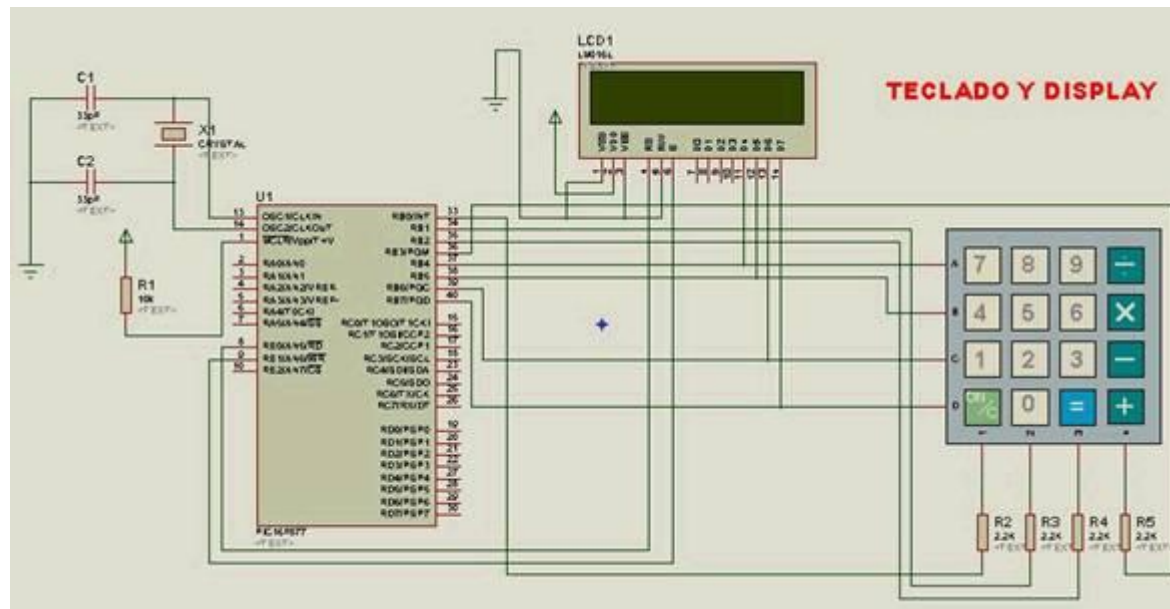
GOTO ASB

END

```


vi.- Uso do Display Alfanumérico e Teclado.

Trocar o tipo de micro da mesma forma que o circuito anterior para -P16F877



Componentes:

ACTIVE (KEYPAD-SMALLCALC)

Programa:

```
DEFINE OSC 20
'DEFINE LOADER_USED 1
ADCON1 = 7
TRISE = 0
TRISB = %00001111
OPTION_REG = %00010101
'INTCON2 = %00000000

DEFINE LCD_DREG PORTB
DEFINE LCD_DBIT 4
DEFINE LCD_RSREG PORTE
DEFINE LCD_RSBIT 0
DEFINE LCD_EREG PORTE
DEFINE LCD_EBIT 1
DEFINE LCD_BITS 4
DEFINE LCD_LINES 2
DEFINE LCD_COMMANDUS 2000
DEFINE LCD_DATAUS 50

Tecla var byte
FLAG VAR BIT

L1 CON %01110000
L2 CON %10110000
L3 CON %11010000
L4 CON %11100000
```

PAUSE 20

LCDOUT \$FE, 1,"Listo" 'Clear display and show “Preparado”

Rastreo:

```
PORTB = L1
SELECT CASE PORTB & $0F
CASE %1110
LCDOUT $FE, 1,"ON/C"
FLAG = 1
CASE %1101
LCDOUT $FE, 1,"0"
FLAG = 1
CASE %1011
LCDOUT $FE, 1,"="
FLAG = 1
CASE %0111
LCDOUT $FE, 1,"+"
FLAG = 1
END SELECT
```

```
PORTB = L2
SELECT CASE PORTB & $0F
CASE %1110
LCDOUT $FE, 1,"1"
FLAG = 1
CASE %1101
LCDOUT $FE, 1,"2"
FLAG = 1
```

```
CASE %1011
LCDOUT $FE, 1,"3"
FLAG = 1
CASE %0111
LCDOUT $FE, 1,"-"
FLAG = 1
END SELECT
```

```
PORTB = L3
SELECT CASE PORTB & $0F
CASE %1110
LCDOUT $FE, 1,"4"
FLAG = 1
CASE %1101
LCDOUT $FE, 1,"5"
FLAG = 1
CASE %1011
LCDOUT $FE, 1,"6"
FLAG = 1
CASE %0111
LCDOUT $FE, 1,"X"
FLAG = 1
END SELECT
```

```
PORTB = L4
SELECT CASE PORTB & $0F
CASE %1110
LCDOUT $FE, 1,"7"
FLAG = 1
CASE %1101
```

```
LCDOUT $FE, 1,"8"  
FLAG = 1  
CASE %1011  
LCDOUT $FE, 1,"9"  
FLAG = 1  
CASE %0111  
LCDOUT $FE, 1,"/"  
FLAG = 1  
END SELECT
```

```
IF FLAG = 1 THEN  
PAUSE 150  
FLAG = 0  
ENDIF
```

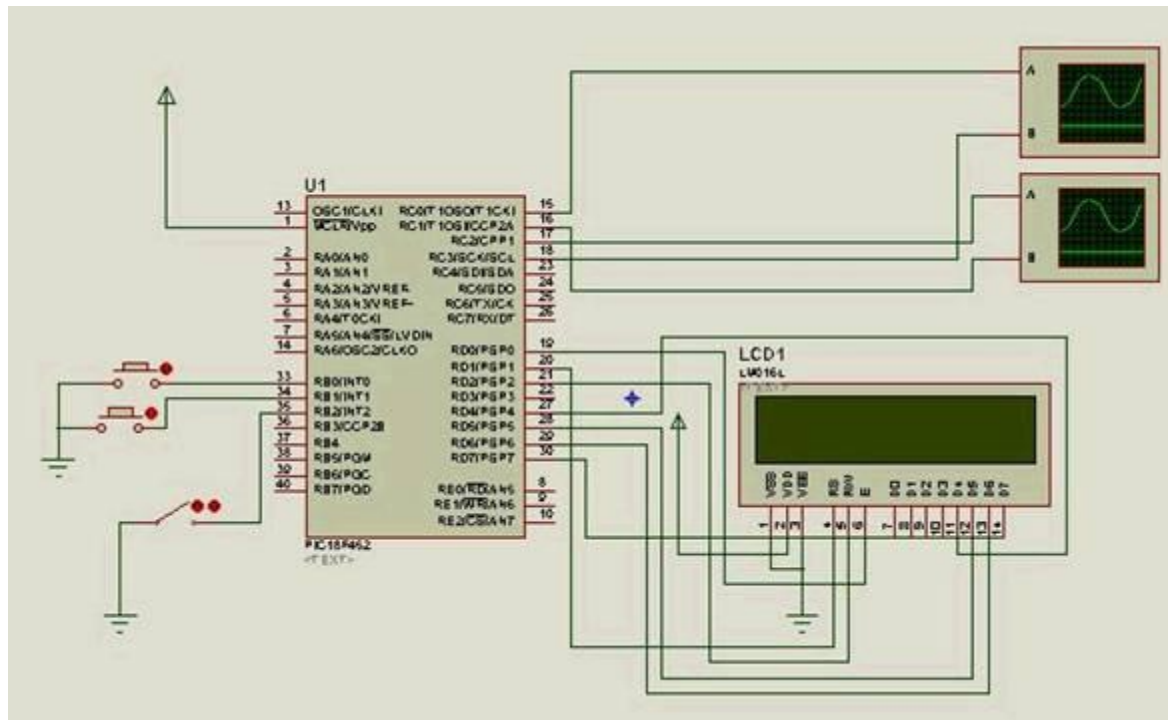
```
GOTO Rastreo
```

PROTEUS VSM 5

N.- Uso do PICC com o Proteus.

- Armar o seguinte circuito:
- Este circuito se utilizou como controle de um inversor monofásico de múltiplos pulsos, com frequência e voltagem variável.

18F452



Componentes:

- MICRO (PIC18F452)
- DISPLAY (LM016L)
- ACTIVE (BUTTON)

- ACTIVE (SWITCH)
- Salvar o circuito em uma nova pasta só para o circuito. Nesta pasta se deve salvar também o código fonte gerado com o PICC.
- **Pegar o seguinte programa em um novo arquivo usando PICC.**

Programa:

```
// Inversor 05 "Final"

#include <18F452.h>
#include <delay.h>
#define delay(clock=20000000)
#define fuses HS,PUT,BROWNOUT,NOWDT,NOPROTECT,NOLVP, BORV20, STVREN
#include <fast_io.h>
#include <fast_io(B)>
#include <lcd.c>

int16 delta = 0, d1 = 0, delta_back, d1_back;
int8 volfreq = 0, cuenta = 0, periodo = 0;
int1 inicio = 0, arranque = 0, flag1 = 0, flag2 = 0;

void cambio();

#INT_EXT
void modo()
{
if (inicio == 0)
{
inicio = 1;
flag1 = 1;
lcd_gotoxy(5,2); lcd_putc("000");
```

```
lcd_gotoxy(14,2); lcd_putc("000");  
}  
else  
{  
if (input(PIN_B2) == 1)  
{  
if (volfreq <= 11)  
{  
volfreq += 1;  
}  
}  
else  
{  
if (volfreq > 0)  
{  
volfreq -= 1;  
}  
}  
cambio();  
}  
}
```

```
#INT_EXT1  
void motor()  
{  
disable_interrupts(INT_EXT1);  
inicio = 0;  
arranque = 1;  
volfreq = 3;
```



```
cambio();  
}
```

```
#INT_TIMER0  
void mot_arr()  
{  
  if (volfreq <= 11)  
  {  
    set_timer0(26473);  
    volfreq += 1;  
    cambio();  
    if (volfreq == 12)  
    {  
      disable_interrupts(INT_TIMER0);  
      setup_timer_0(RTCC_OFF);  
    }  
  }  
}
```

```
#INT_TIMER1  
void carga_delta()  
{  
  SET_TRIS_C(0x00);  
  SET_TRIS_B(0xFF);  
  output_C(0x00);  
  
  disable_interrupts(INT_EXT);  
  disable_interrupts(INT_TIMER1);  
  setup_timer_1(T1_DISABLED|T1_DIV_BY_1);
```

```
set_timer3(d1_back);
setup_timer_3(T3_INTERNAL/T3_DIV_BY_2);
enable_interrupts(INT_TIMER3);
periodo += 1;
```

```
cuenta += 1;
if (cuenta == 5)
{
    cuenta = 0;
    if (flag2 == 0)
    {
        flag2 = 1;
    }
    else
    {
        flag2 = 0;
    }
}
```

```
output_low(PIN_C0);
output_low(PIN_C1);
enable_interrupts(INT_EXT);
}
```

```
#INT_TIMER3
void carga_d1()
{
    disable_interrupts(INT_EXT);
    disable_interrupts(INT_TIMER3);
```

```
setup_timer_3(T3_DISABLED|T3_DIV_BY_2);
set_timer1(delta_back);
setup_timer_1(T1_INTERNAL|T1_DIV_BY_1);
```

```
if (flag2 == 0)
{
    output_high(PIN_C0);
}
else
{
    output_high(PIN_C1);
}
enable_interrupts(INT_TIMER1);
}
```

```
void main()
{
    port_b_pullups(TRUE);
    setup_adc_ports(NO_ANALOGS);
    setup_adc(ADC_OFF);

    setup_psp(PSP_DISABLED);
    setup_spi(FALSE);
    setup_timer_2(T2_DISABLED,0,1);

    disable_interrupts(INT_TIMER1);
    disable_interrupts(INT_TIMER3);
    setup_timer_0(RTCC_OFF);
```

```
ext_int_edge(H_TO_L);
```

```
lcd_init();
```

```
lcd_putc("\f");
```

```
lcd_gotoxy(2,1); lcd_putc("INVERSOR ITA");
```

```
lcd_gotoxy(1,2); lcd_putc("VOL:OFF FREC:OFF");
```

```
setup_timer_1(T1_DISABLED|T1_DIV_BY_1);
```

```
set_timer1(0x0001);
```

```
setup_timer_3(T3_DISABLED|T3_DIV_BY_2);
```

```
set_timer3(0x0001);
```

```
setup_timer_0(RTCC_OFF);
```

```
set_timer0(0x0001);
```

```
enable_interrupts(GLOBAL);
```

```
enable_interrupts(INT_EXT);
```

```
enable_interrupts(INT_EXT1);
```

```
while(TRUE)
```

```
{
```

```
if (inicio == 1)
```

```
{
```

```
if (periodo == 10)
```

```
{
```

```
periodo = 0;
```

```
flag1 = 0;
```

```
}
```

```
if (flag1 == 0)
```

```

{
delta_back = delta;
d1_back = d1;
flag1 = 1;
set_timer3(d1_back);
setup_timer_3(T3_INTERNAL|T3_DIV_BY_2);
enable_interrupts(INT_TIMER3);
}
}
else
{
if (arranque == 1)
{
disable_interrupts(INT_EXT1);
set_timer0(26473);
setup_timer_0(RTCC_INTERNAL|RTCC_DIV_128);
enable_interrupts(INT_TIMER0);
arranque = 0;
inicio = 1;
flag1 = 0;
}
}
}
}

```

```

void cambio()
{
switch (volfreq)
{

```

```
case 0: disable_interrupts(INT_TIMER1);  
disable_interrupts(INT_TIMER3);  
enable_interrupts(INT_EXT1);  
output_low(PIN_C0);  
output_low(PIN_C1);  
flag1 = 1;  
lcd_gotoxy(5,2); lcd_putc("000");  
lcd_gotoxy(14,2); lcd_putc("000");  
break;
```

```
case 1: flag1 = 0;  
disable_interrupts(INT_EXT1);  
delta = 65227;  
d1 = 15690;  
lcd_gotoxy(5,2); lcd_putc("010");  
lcd_gotoxy(14,2); lcd_putc("005");  
break;
```

```
case 2: delta = 64919;  
d1 = 40845;  
lcd_gotoxy(5,2); lcd_putc("020");  
lcd_gotoxy(14,2); lcd_putc("010");  
break;
```

```
case 3: delta = 64610;  
d1 = 49332;  
lcd_gotoxy(5,2); lcd_putc("030");  
lcd_gotoxy(14,2); lcd_putc("015");  
break;
```

```
case 4: delta = 64301;  
d1 = 53653;  
lcd_gotoxy(5,2); lcd_putc("040");  
lcd_gotoxy(14,2); lcd_putc("020");  
break;
```

```
case 5: delta = 63993;  
d1 = 56308;  
lcd_gotoxy(5,2); lcd_putc("050");  
lcd_gotoxy(14,2); lcd_putc("025");  
break;
```

```
case 6: delta = 63684;  
d1 = 58129;  
lcd_gotoxy(5,2); lcd_putc("060");  
lcd_gotoxy(14,2); lcd_putc("030");  
break;
```

```
case 7: delta = 63376;  
d1 = 59473;  
lcd_gotoxy(5,2); lcd_putc("070");  
lcd_gotoxy(14,2); lcd_putc("035");  
break;
```

```
case 8: delta = 63067;  
d1 = 60521;  
lcd_gotoxy(5,2); lcd_putc("080");  
lcd_gotoxy(14,2); lcd_putc("040");  
break;
```

```
case 9: delta = 62758;  
d1 = 61369;  
lcd_gotoxy(5,2); lcd_putc("090");  
lcd_gotoxy(14,2); lcd_putc("045");  
break;
```

```
case 10: delta = 62450;  
d1 = 62079;  
lcd_gotoxy(5,2); lcd_putc("100");  
lcd_gotoxy(14,2); lcd_putc("050");  
break;
```

```
case 11: delta = 62141;  
d1 = 62688;  
lcd_gotoxy(5,2); lcd_putc("110");  
lcd_gotoxy(14,2); lcd_putc("055");  
break;
```

```
case 12: delta = 61832;  
d1 = 63221;  
lcd_gotoxy(5,2); lcd_putc("120");  
lcd_gotoxy(14,2); lcd_putc("060");  
break;  
}  
}
```

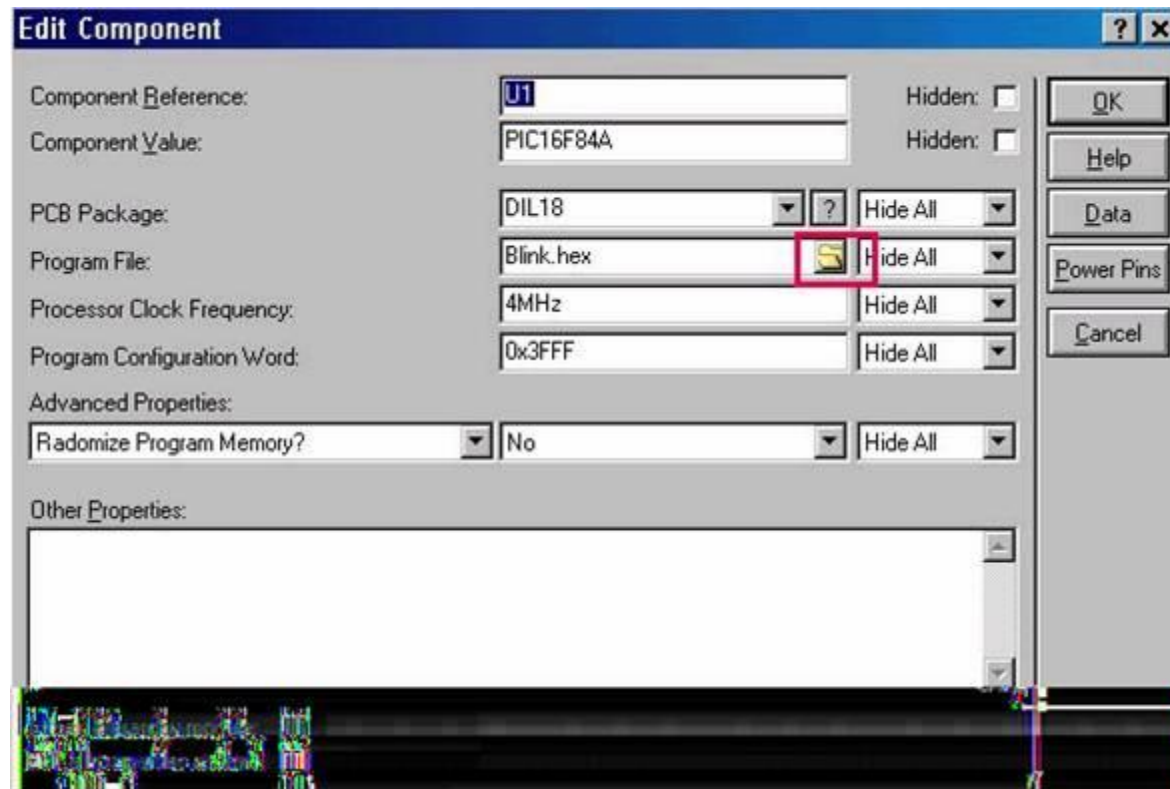

NOTA: recomenda-se tabular o programa anterior para que seja mais fácil sua leitura.

- Salvar o arquivo na mesma pasta em que se salvou o circuito.
- Compilar o arquivo.

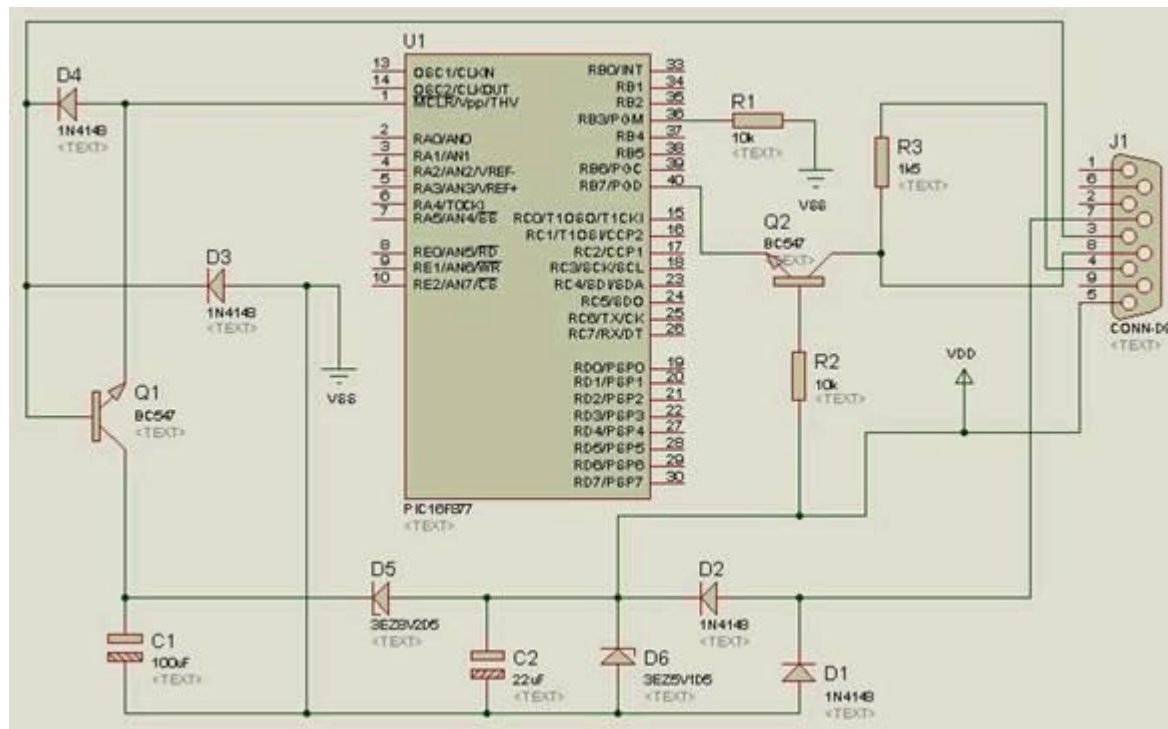
Para que o programa anterior funcione corretamente é necessário editar o arquivo LCD.C que vem com o PICC, substituindo a instrução delay_cycles(1) por delay_us(1). Este arquivo se encontra em:

C:\Archivos de programa\PICC\Drivers

- Carregar o arquivo *.HEX gerado pelo PICC, como se realizou para os microcontroladores PIC16F84 e 877. A frequência do relógio deve ser trocada para 20MHz.



I.- Usando ISIS montar o seguinte circuito:



Componentes:

DIODE (1N4148)

BIPOLAR (BC547)

DEVICE (CAP-ELEC)

DIODE (3EZ8V2D5)

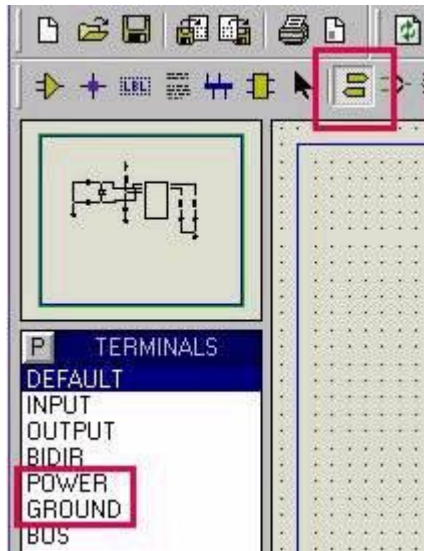
DIODE (3EZ5V1D5)

DEVICE (RES)

DEVICE (CONN-D9)

MICRO (PIC16F877)

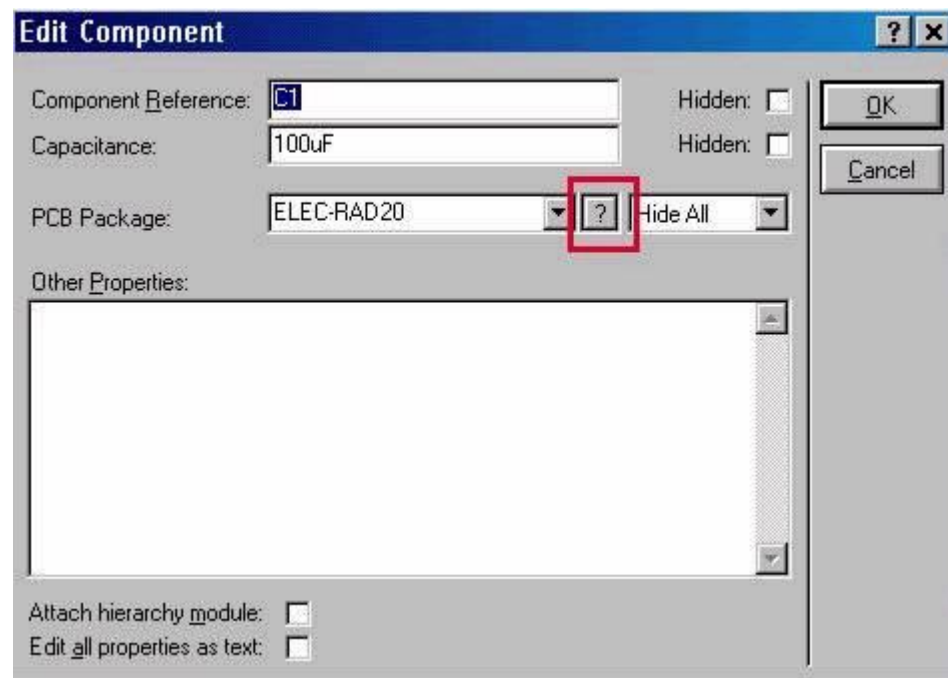
A alimentação e o terra para este circuito se encontram dentro do Inter-Sheet Terminal da barra de ferramentas.



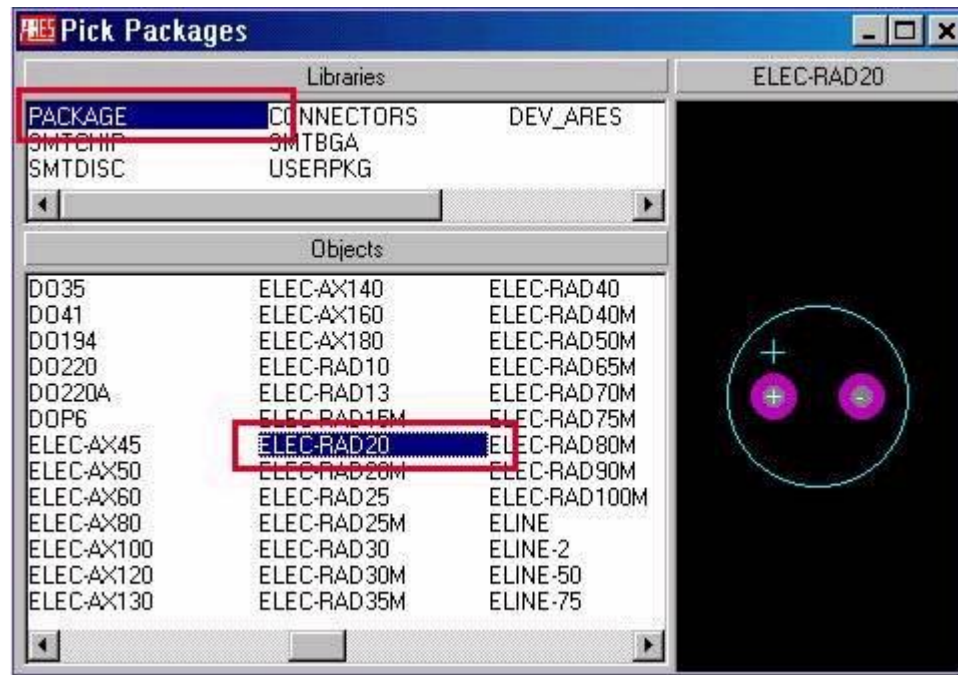
É necessário etiquetar o POWER como VDD e o GROUND como VSS, para que se realize a conexão adequada com o PIC. As duas terras devem ter a mesma etiqueta.

Também se requer modificar o encapsulamento do capacitor de 100uF, o que se realiza da seguinte forma:

Dar click direito sobre o componente CAP-ELEC (100uF) "C1" e depois com o esquerdo para abrir sua forma de configuração.



Dar um click no símbolo de interrogação.



Procurar o Objeto (ELEC-RAD20) da biblioteca PACKAGE dando um duplo click sobre o nome do objeto.

Pressionar OK na forma de edição do componente.

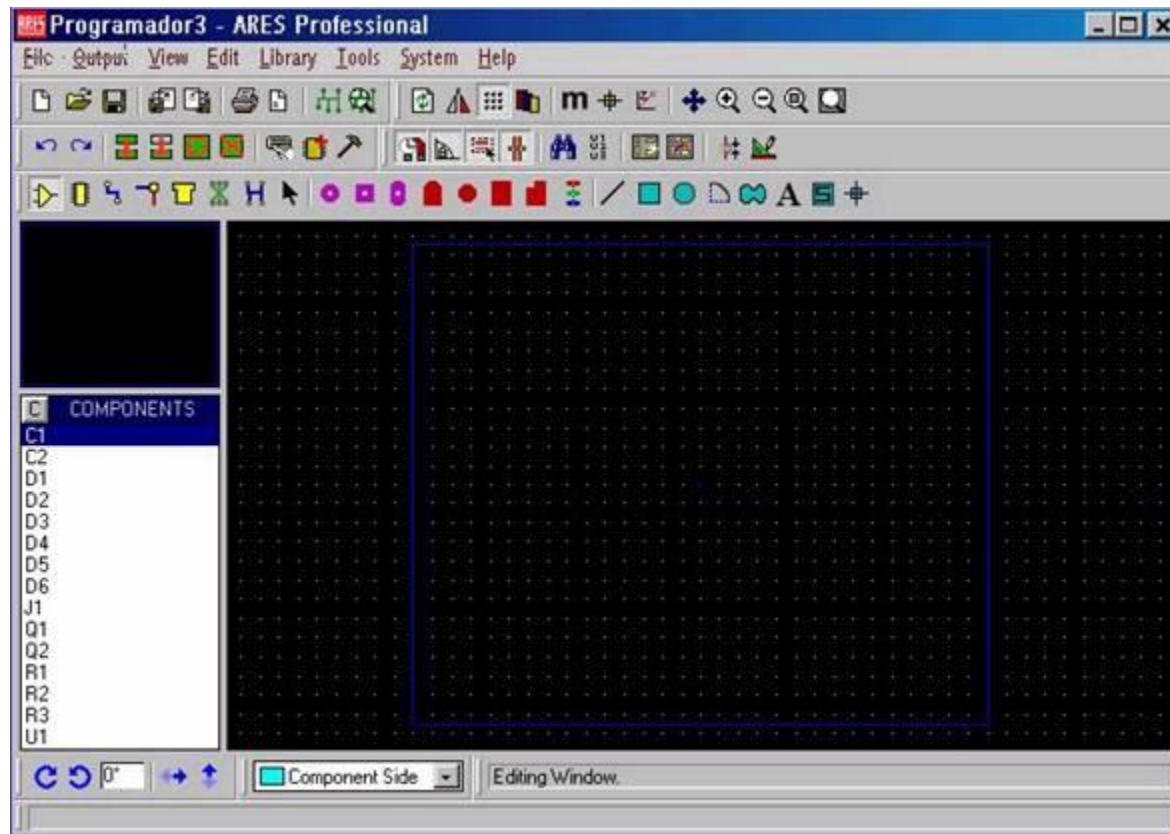
Salvar o circuito.

Ir ao menu Tools e pressionar Netlist Compiler.

No mesmo menu pressionar Netlist to ARES.

Também se pode pressionar o ícone de ARES que se encontra na barra de ferramentas. O qual gera a Netlist e a exporta a ARES.

abre-se a tela de trabalho do ARES.



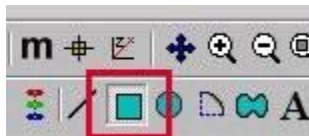
Os componentes aparecem do lado esquerdo da tela, ao parecer não serve de nada o ter conectado os componentes no ISIS, mas o que acontece é que é necessário colocar os componentes dentro de uma área que represente o tamanho da placa que queremos criar.

Ao ir colocando os componentes se vão conectando automaticamente.

O processo de colocação manual só é necessário para aqueles componentes que requerem um colocação especial na placa, já que outros componentes se podem colocar em forma automática usando o Auto Placer, que se verá mais adiante.

realiza-se a criação do borde que representa o tamanho da placa a gerar.

utiliza-se a ferramenta 2D Graphics Box.



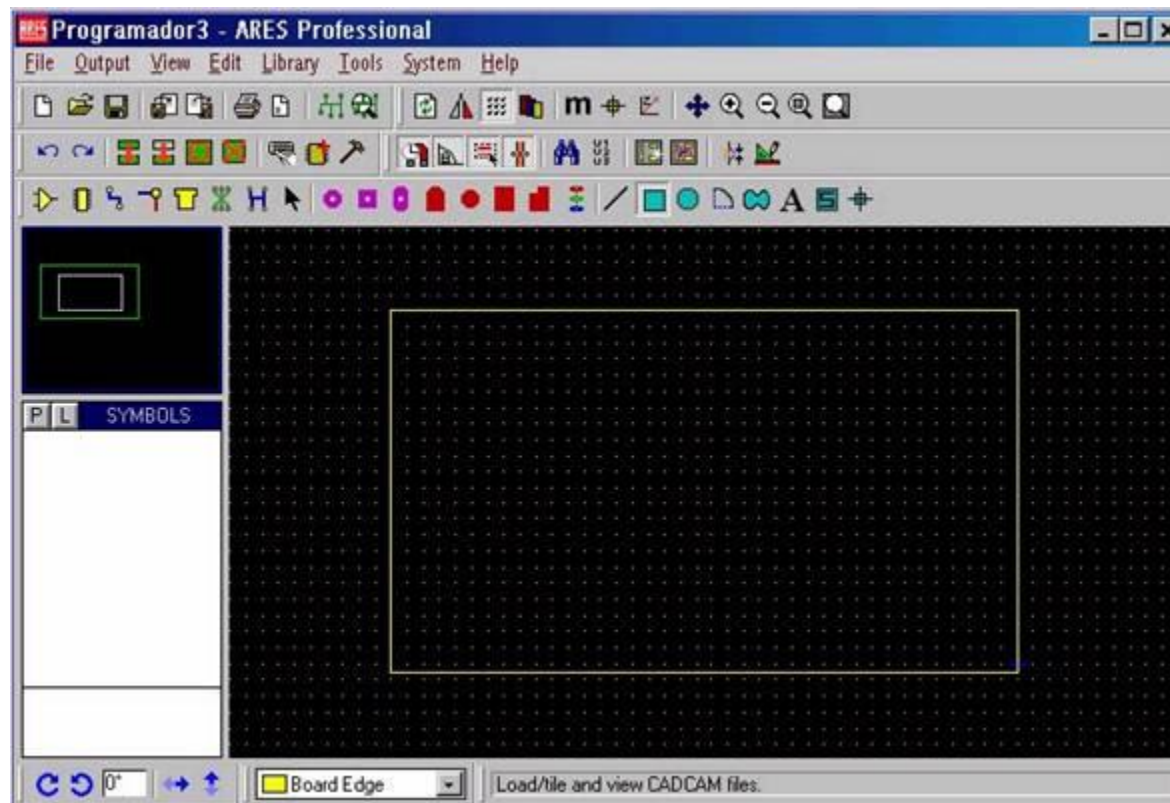
Ao dar click sobre esta ferramenta é possível desenhar uma figura retangular usando o botão esquerdo do mouse dando um click dentro da área de trabalho e sem soltar desenhar o retângulo do tamanho desejado.

antes de realizar o anterior é necessário especificar em que Layer queremos desenhar o retângulo. Isto se especifica na parte inferior da tela e se deve seleccionar "Board Edge".



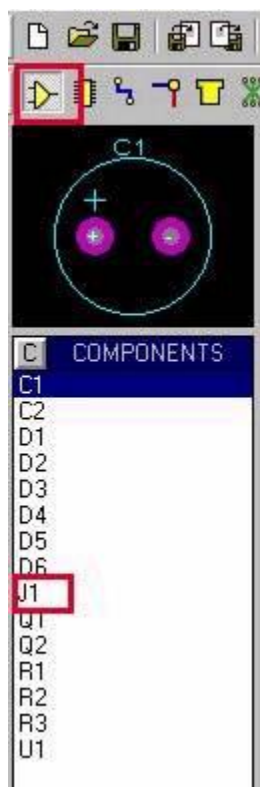
Com isto se obtém que o programa ARES possa identificar a área de trabalho específica em que se têm que colocar os componentes e aonde se tem que levar a cabo o AUTO ROUTER.

A cor deste tipo de retângulo é amarelo.



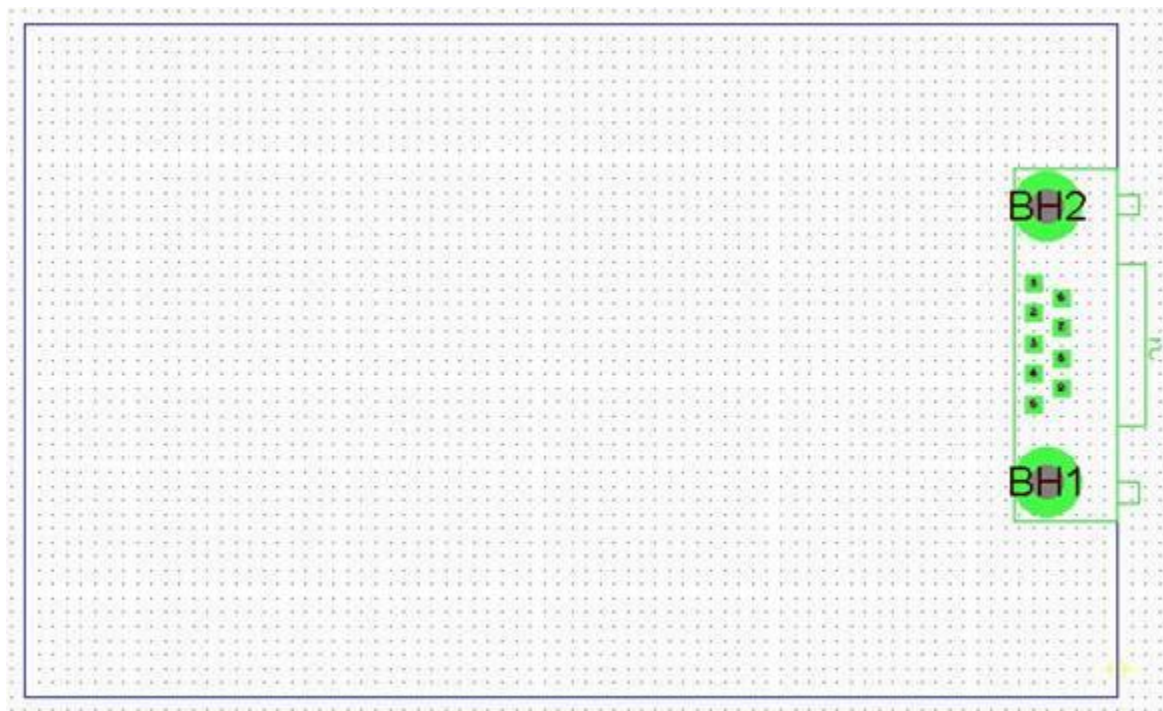
O único componente que é necessário colocar em forma manual é o DB-9 (Conector série), já que este deve estar no bordo da placa e se o fizemos em forma automática será colocado em qualquer parte.

Para seleccioná-lo voltar a habilitar a janela de componentes pressionando Component placement and editing.

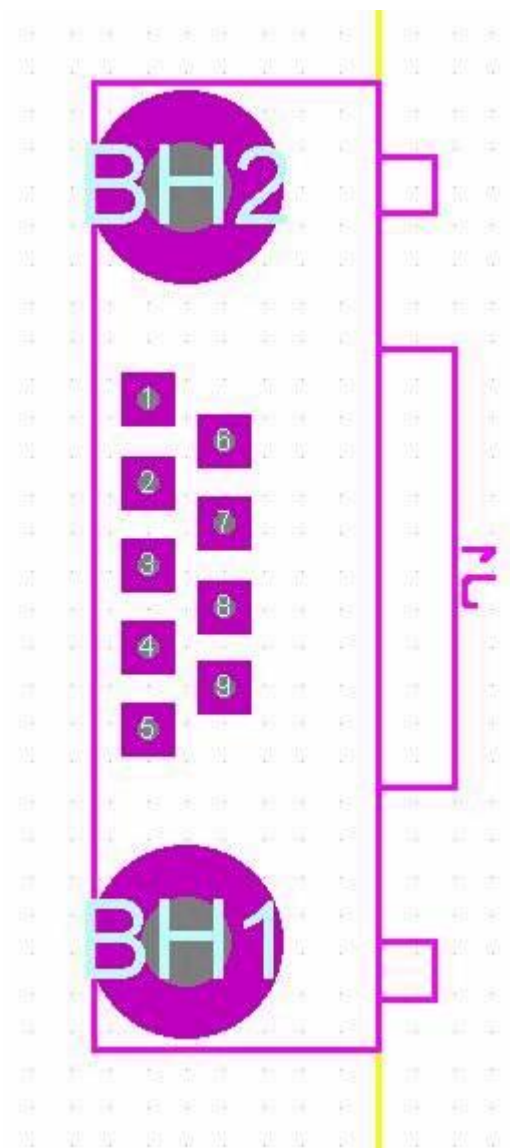


Este componente está identificado como "J1"

Colocar este componente no borde direito, como se mostra a seguir.



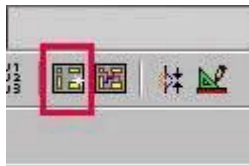
A numeração do componente se pode apreciar melhor na seguinte aproximação:



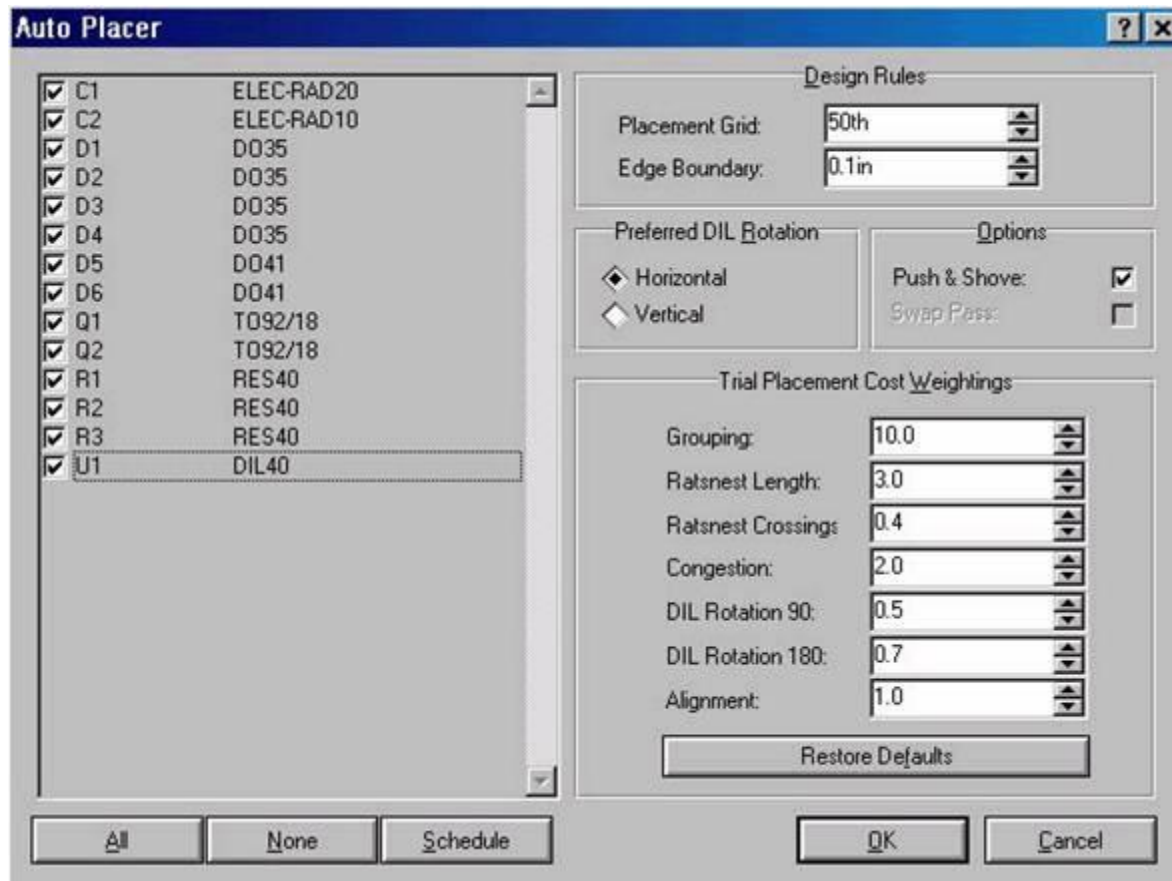
Para a colocação correta do componente é necessário usar as ferramentas de giro e espelho



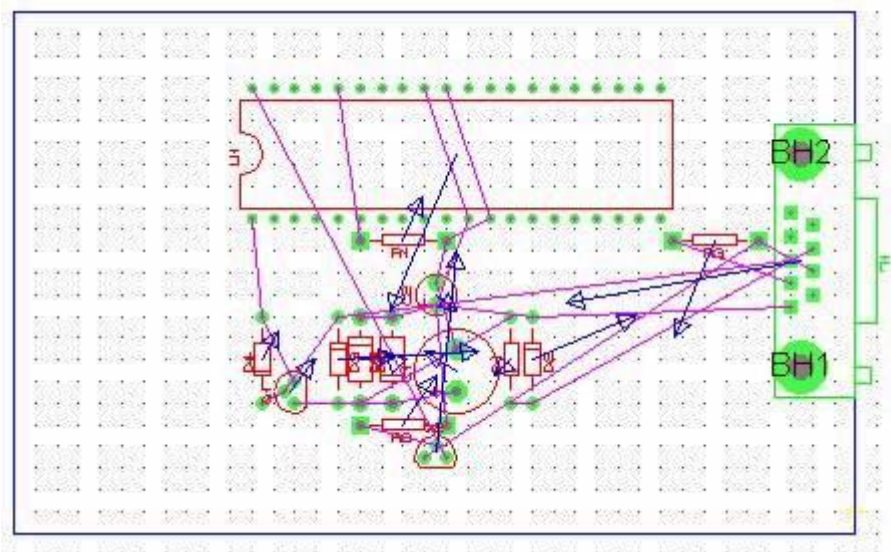
O Auto Placer se ativa ao dar um click sobre o ícone desta ferramenta.



Aparece a seguinte forma em que terá que pressionar OK.



Com o que se obtém o seguinte resultado:



Na figura anterior se pode ver que já se realizaram as conexões, mas ainda não se criaram as pistas correspondentes, só parecem unidas com um cabo direto.

Outra ferramenta que facilita o trabalho é a possibilidade de gerar as pistas de forma automática, utilizando o AUTO RUTER.

antes de poder utilizar esta ferramenta é necessário modificar como vão se criar as pistas, principalmente a grossura e em que lado da placa vão aparecer.

Para trocar estas características se acessa ao menu System e se selecciona Set Strategies...

A forma deve ficar da seguinte forma, para obter um largo de pista aceitável e que só o faça de um lado da placa. A figura seguinte é para a Strategy Power.

Edit Strategies [?] [X]

Strategy: SIGNAL [v] [OK] [Cancel]

Priority: 1

Trace Style: T30 [v]

Via Style: DEFAULT [v]

Neck Style: (None) [v]

Pair 1 (Hoz): Bottom Copper [v]
 (Vert: Bottom Copper [v])

Pair 2 (Hoz): (None) [v]
 (Vert: (None) [v])

Pair 3 (Hoz): (None) [v]
 (Vert: (None) [v])

Pair 4 (Hoz): (None) [v]
 (Vert: (None) [v])

Vias:

☒ Normal

☐ Top Blind

☐ Bottom Blind

☐ Buried

Tactics:

☐ Power

☐ Bus

☒ Signal

Corners:

☒ Optimize?

☒ Diagonal?

Design Rules

Pad - Pad Clearance: 10th [v]

Pad - Trace Clearance: 10th [v]

Trace - Trace Clearance: 10th [v]

Graphics Clearance: 15th [v]

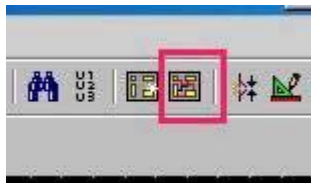
Edge/Slot Clearance: 15th [v]

☐ Hide Ratsnest?

*** Se aparecer uma Strategy POWER se devem pôr os mesmos valores que se observam na forma anterior, com o fim de usar sozinho um lado da placa.

aceitam-se os valores pressionando OK.

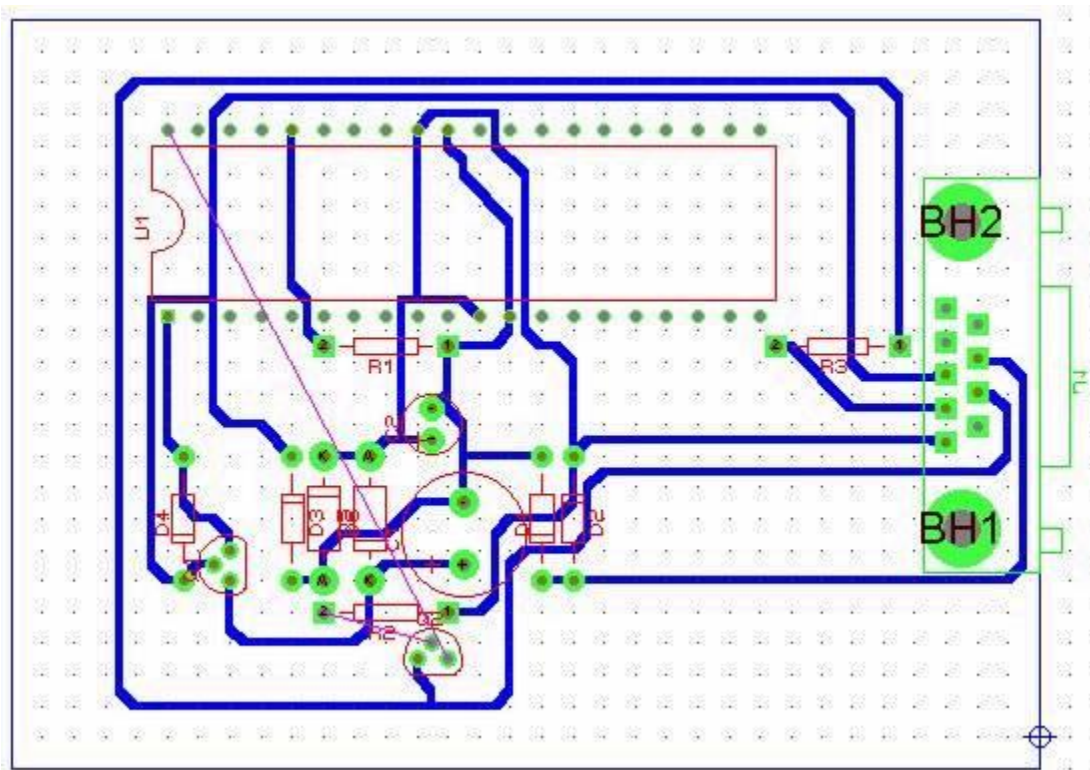
Para executar o AUTO ROUTER se dá um click sobre a seguinte ferramenta.



Aparece a seguinte forma e se pressiona OK.

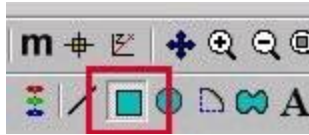


obtém-se o seguinte resultado.



Cabe mencionar que os resultados são variáveis e dependem da grossura das pistas, o tamanho da placa escolhida e se se executar novamente o auto router se pode chegar a obter outro resultado se ainda apresentar pontes (cabos diretos).

Para tratar de reduzir o número de pistas não criadas se pode ampliar o tamanho da placa. Para poder fazê-lo é necessário voltar a selecionar a ferramenta 2D Graphics Box e ter na parte inferior da tela selecionado BOARD EDGE.

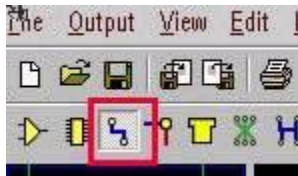


seleciona-se o quadro com o botão direito e ao trocar a cor para branco aparecem uns pequenos quadros que se utilizam para modificar o tamanho da placa, o que se realiza utilizando o botão esquerdo sobre estes quadros pressionando e sem soltar realizar a modificação de tamanho.

depois disto se pode voltar a tentar o AUTO-ROUTER.

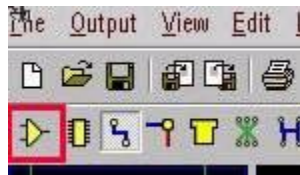
Para desfazer o AUTO-ROUTER se realiza o seguinte:

- 1.- Escolher a ferramenta Track placement and editing dando um click sobre ela.



2.- Depois utilizando o botão direito selecionar toda a placa e por último pressionar SUPR ou DELETE no teclado.

Se queremos tirar todos os componentes se tem que escolher Component placement and editing.



Se não querermos apagar o borde da placa se tem que realizar a seleção dentro desta, já que se seleccionarmos toda a placa, esta também se apagará. depois de realizada a seleção se pressiona SUPR ou DELETE.

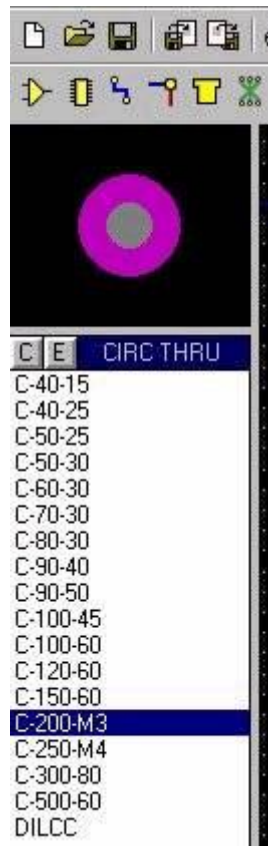
Cabe mencionar que os componente são retornados à janela COMPONENTS do lado esquerdo da tela, com o qual podemos realizar o processo novamente até obter um resultado adequado.

Se queremos adicionar umas perfurações nas esquinas da placa se realiza o seguinte:

1.- Escolher a ferramenta Round through-hole pad placement.

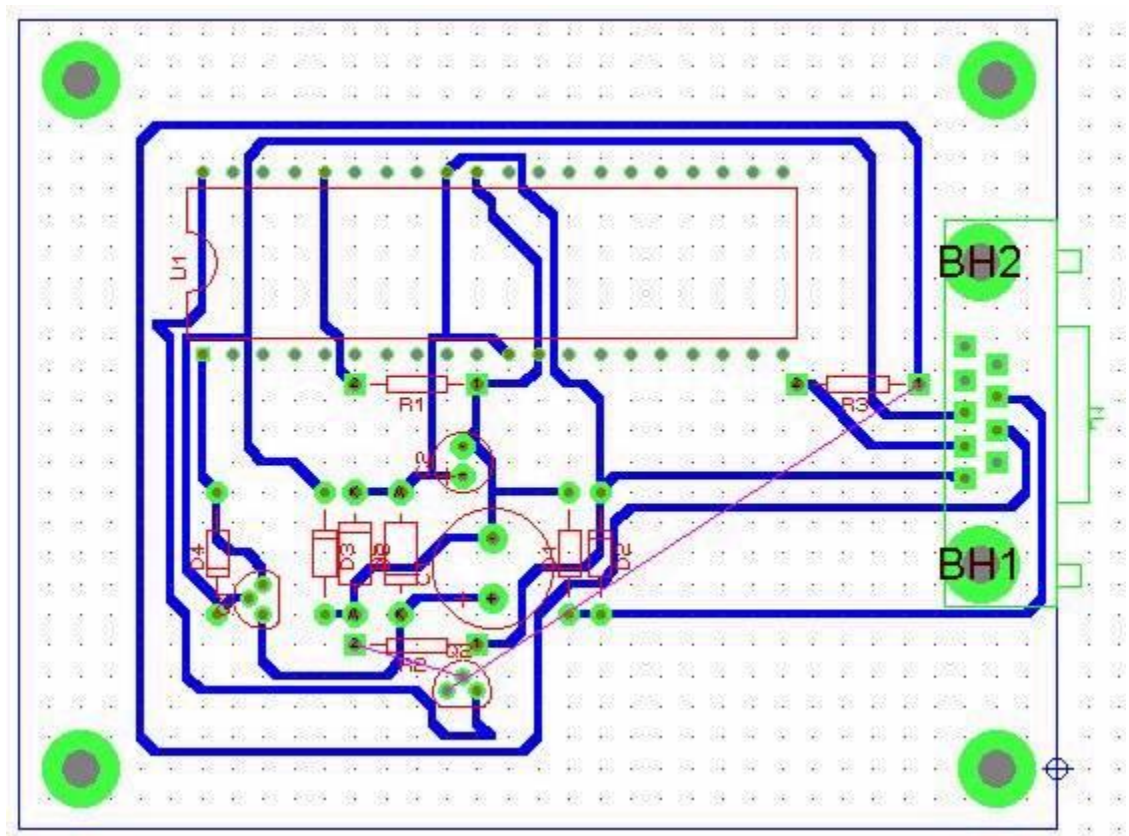


2.- Escolher o pad adequado as nossas necessidades.

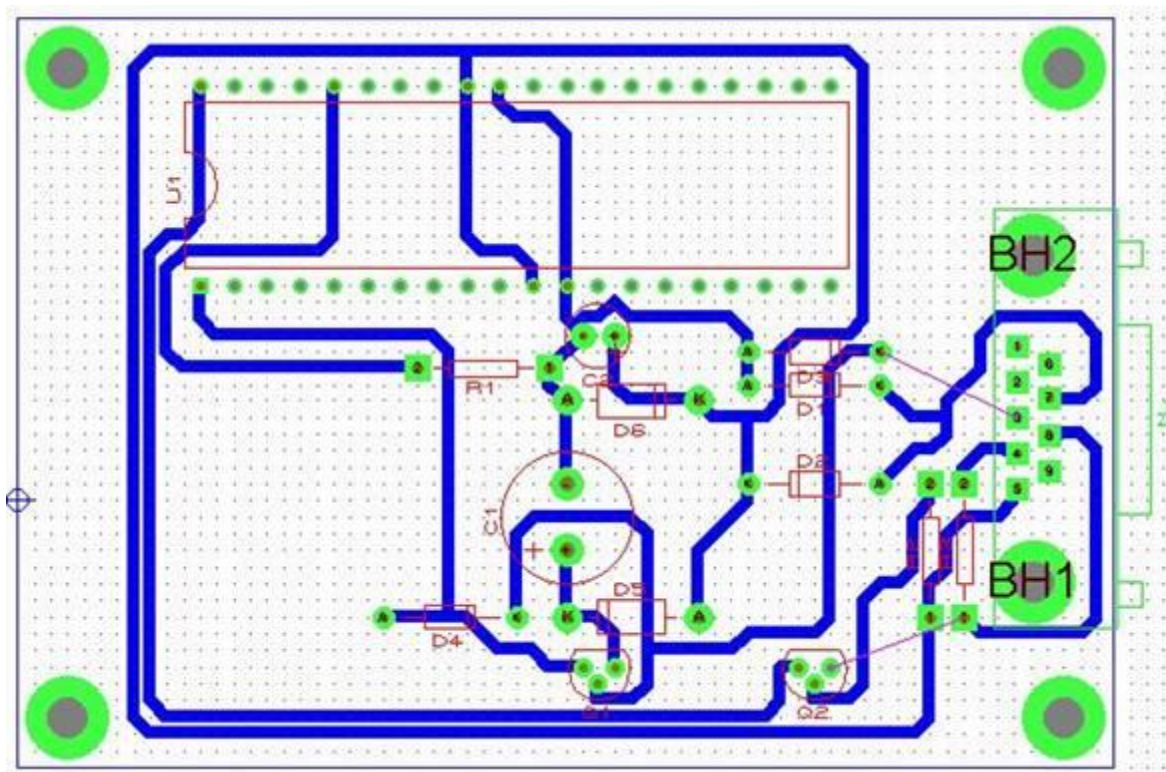


3.- Dar um click no que se necessite, para poder realizar a colocação nas quatro esquinas da placa utilizando o botão esquerdo.

4.- Placa final obtida (Não está otimizada)



*** Placa final obtida (Otimizada)



NOTA: Esta placa não foi fabricada e provada, só se utiliza neste curso como exemplo do funcionamento do programa ARES, por isso se aconselha que seja feito uma revisão antes de levar a cabo sua fabricação.
