

TUTORIAL COMO USAR O PICKIT 3

Professor: Roberto Bairros dos Santos

www.bairrospd.com

Data:27/06/2016

SOU: bairrospd, tutoriais, eletrônica, PicKit 3, microcontrolador, pic, gravando o pic

Sumário

Introdução.....	3
Interligação do PicKit3 com o microcontrolador.	4
O Circuito.	6
O programa.	14
Transferindo o programa para o PIC.	16
Conclusão.....	20
Referências.....	21

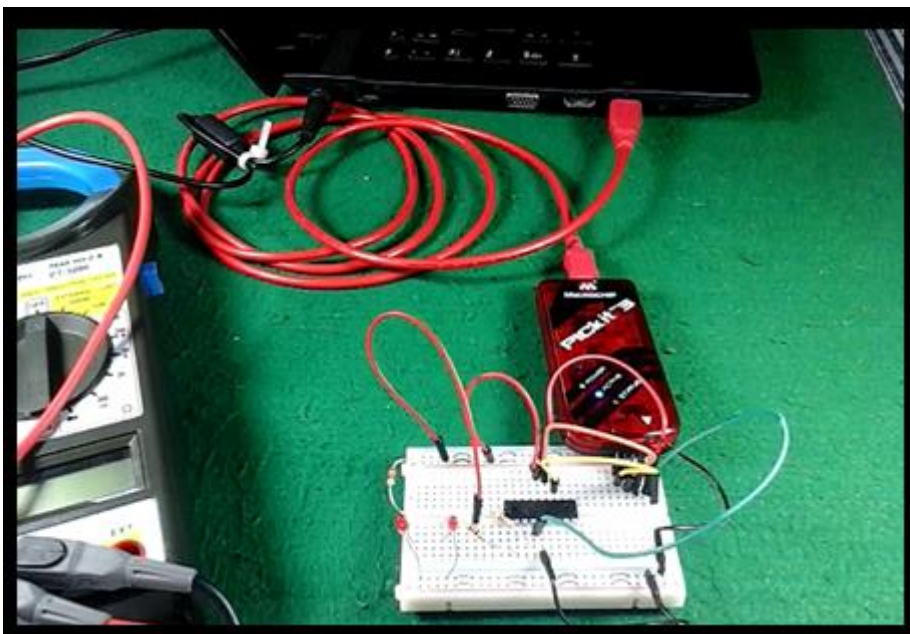
Introdução.

O PicKit3 pode ser usado para programar e depurar um trabalho utilizando os microcontroladores da linha PIC.

Em alguns modelos junto com o PICKIT3 acompanha um módulo de demonstração chamado de 44PinDemoBoard com um PIC do tipo PIC18F45K20, a figura abaixo mostra estes componentes!

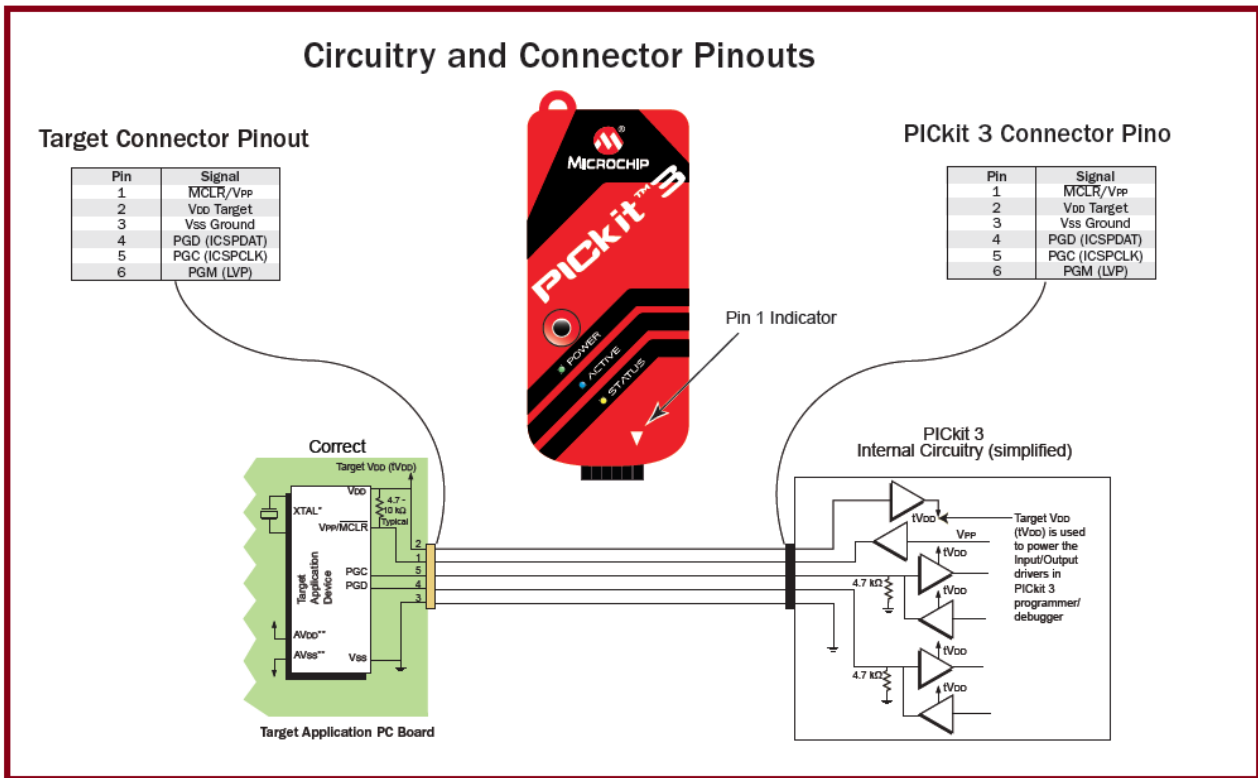


Neste tutorial você vai ver como carregar um programa simples do tipo pisca-pisca para o PIC16F628 montado em sua forma mais simples, sem cristal usando o oscilador interno!

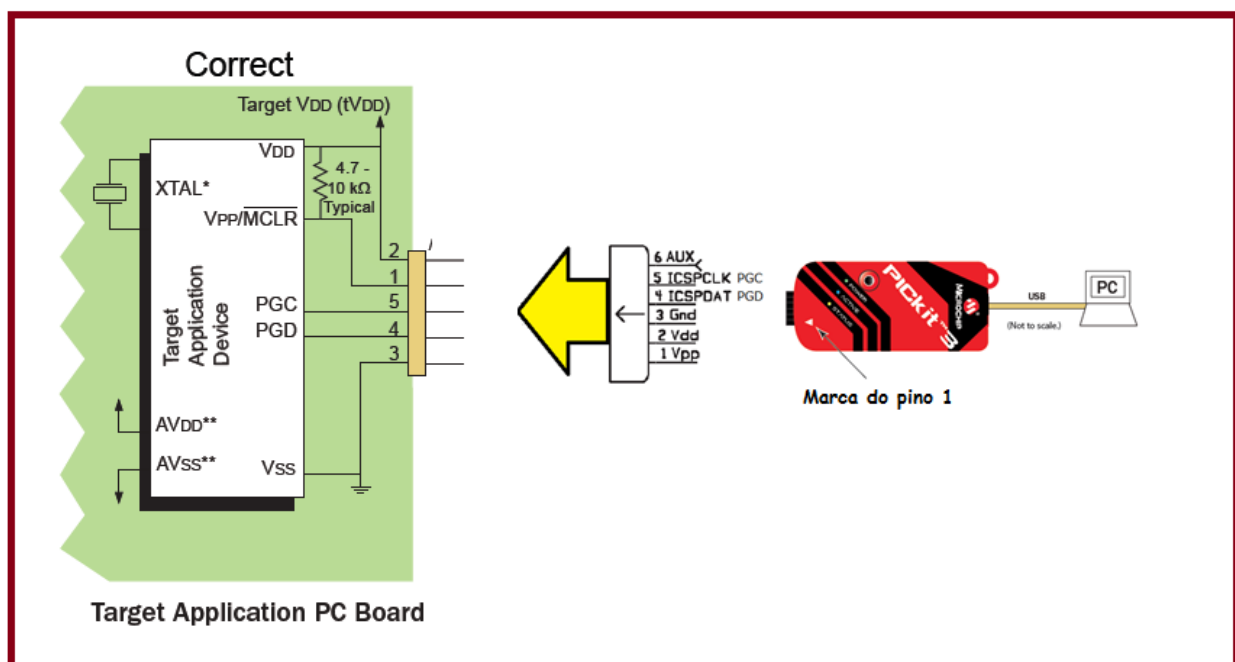


Interligação do PicKit3 com o microcontrolador.

Junto com o PicKit3 vem um pôster com as principais informações sobre como usar o dispositivo.



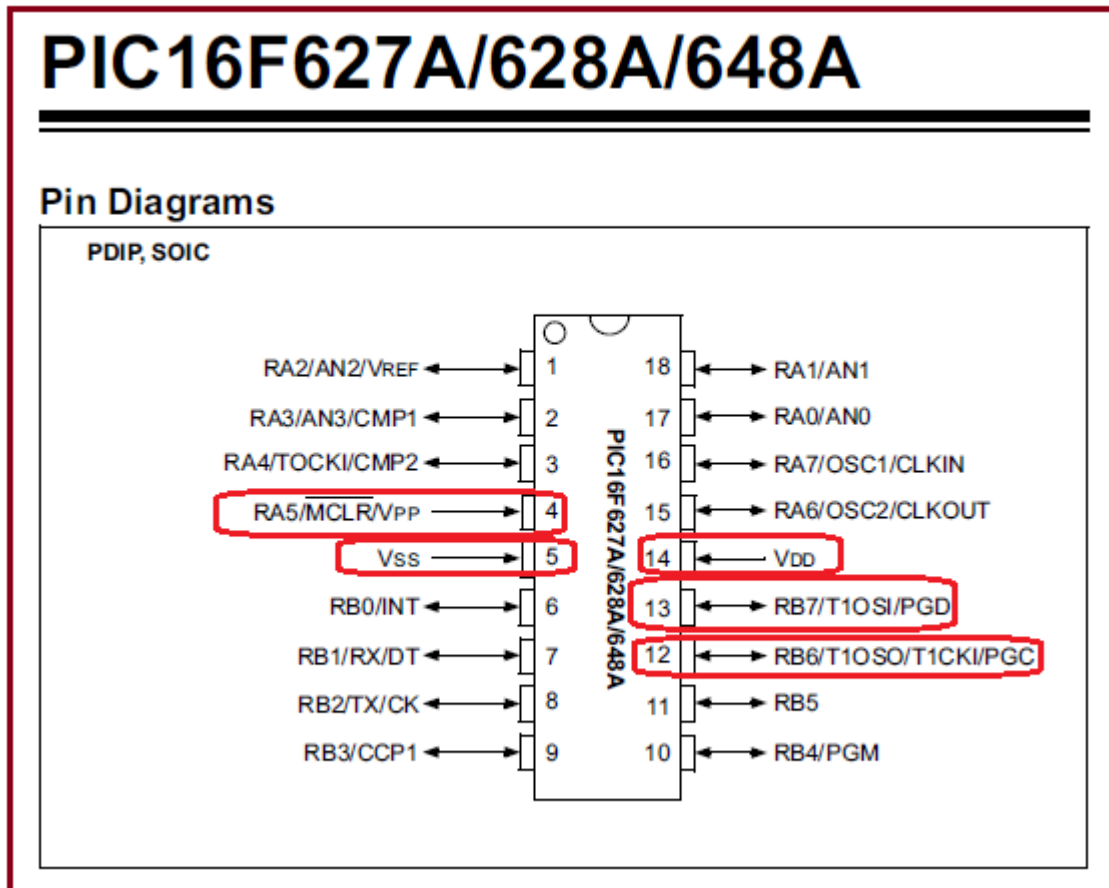
A principal informação é como ligar o PicKit3 no microcontrolador, eu fiz um resumo abaixo usando o PIC 16F628 como target (o PIC que eu quero programar)!



Onde ligar no PIC:

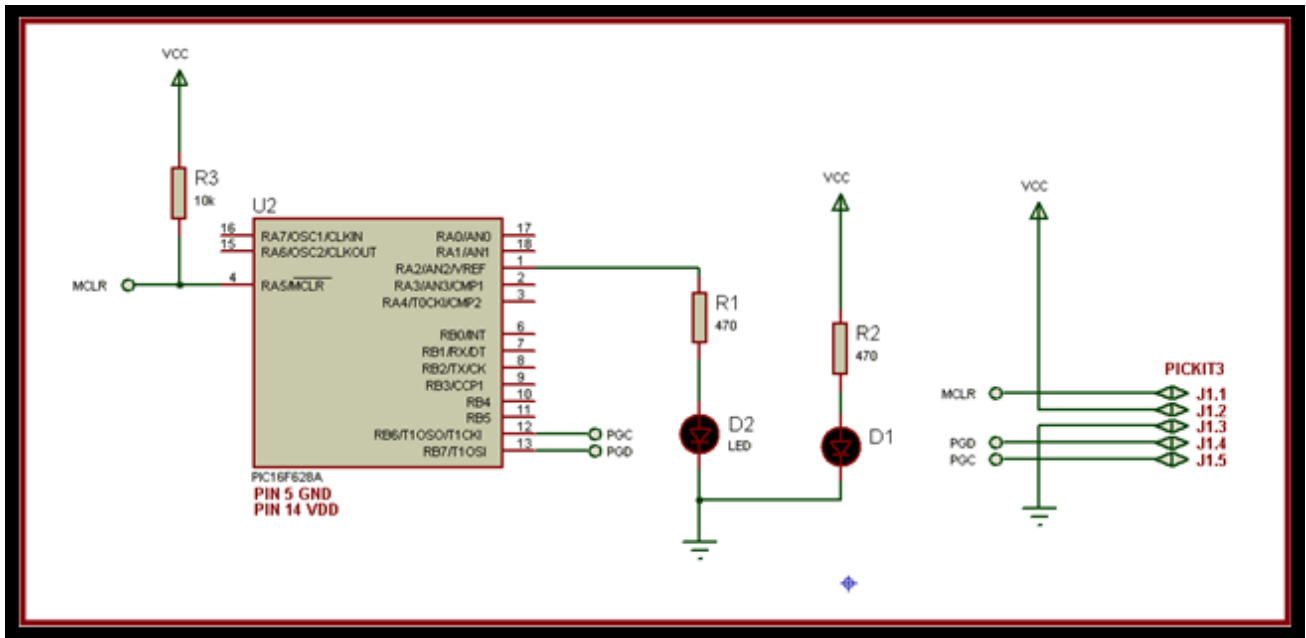
Você só tem que escolher os pinos no datasheet do PIC!

Neste exemplo será usado o PIC16F628A mostrado na figura abaixo salientando os pinos usado com o PicKit3.



O Circuito.

O circuito é simples e pode ser montado em uma protoboard, o diagrama é mostrado abaixo incluindo a interligação com o PicKit3!



Os pinos J1 são os pinos de interligação com o PicKit3.

Observar que foi ligado um resistor (R3) de pull up no pino 4 (MCLR) do microcontrolador, não omita esta resistência!

O LED D1 foi colocado para que você veja quando o circuito está energizado, se você quiser pode deixar sem!

O LED D2 está ligado a saída RA2 pino 1 e será usado para demonstrar o programa funcionando, neste caso o LED irá pisca!

Notar que não é usado cristal neste circuito. Você deverá programar os fusíveis para oscilador interno!

Não usar cristal oscilador simplifica o circuito, mas só deve ser usado em circuitos simples onde rotinas de tempo não são críticas!

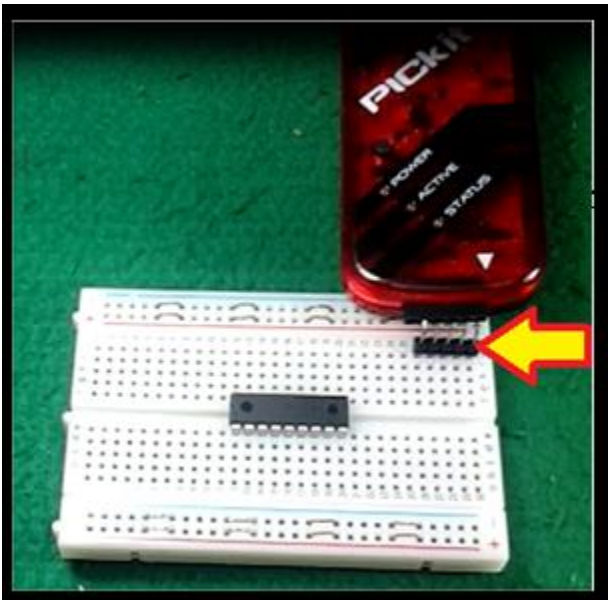
Para este exemplo é devido a simplicidade você pode montar em uma protoboard mesmo!

A montagem:

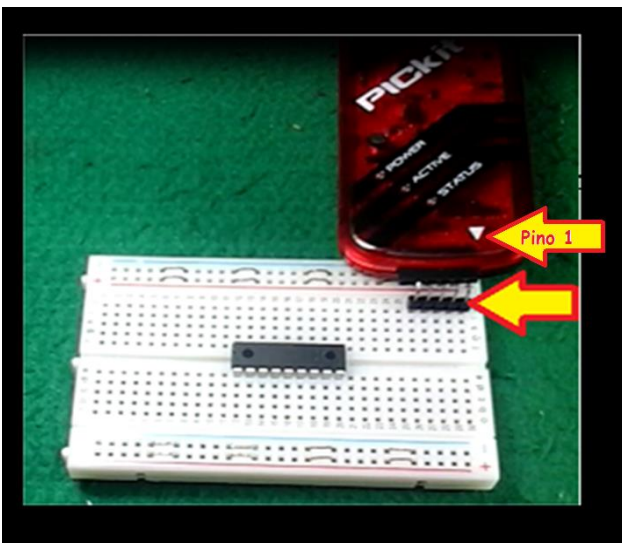
Proceda como descrito a seguir para ligar o PICkit3 no PIC montado em uma protoboard!



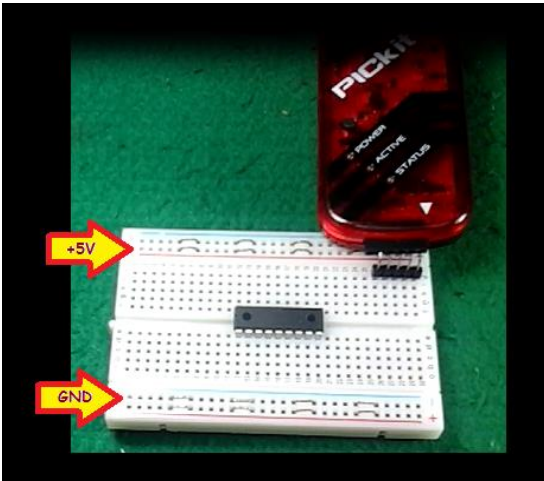
Monte uma barra de ligação na protoboard para conectar ao PICkit3!
Junto com o PICkit3 vem um cabo que também pode ser usado para esta finalidade!



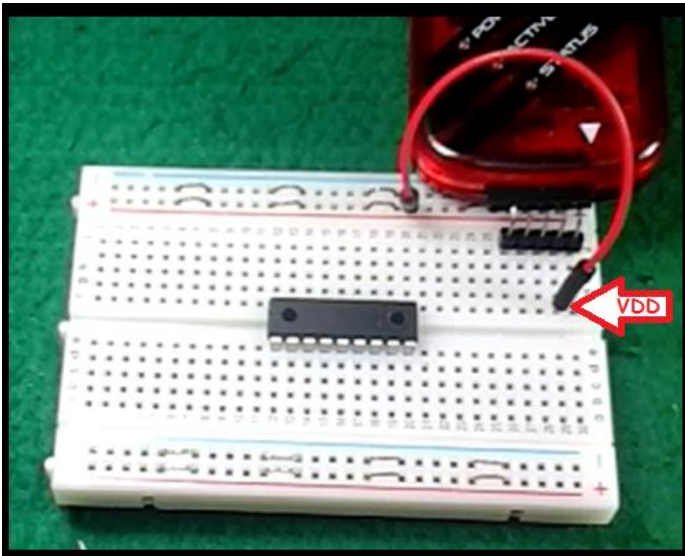
Observe com atenção a marca do pino 1 no PicKit3!



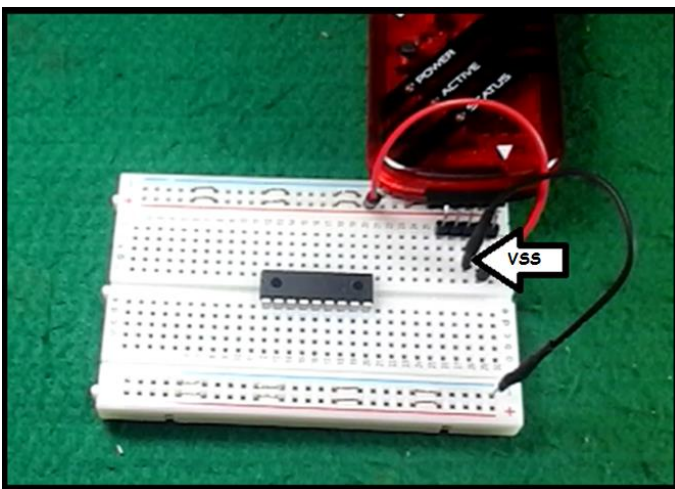
Aconselho a iniciar a montagem ligando a alimentação do circuito usando a trilha vermelha superior da protoboard para o VDD(+5V) e a trilha azul inferior para o terra (GND 0V)!



Você deve ligar o pino 2 do PICkit3 para alimentar todo o circuito (fio vermelho), desta forma o circuito será alimentado pela USB do PC via PicKit3!

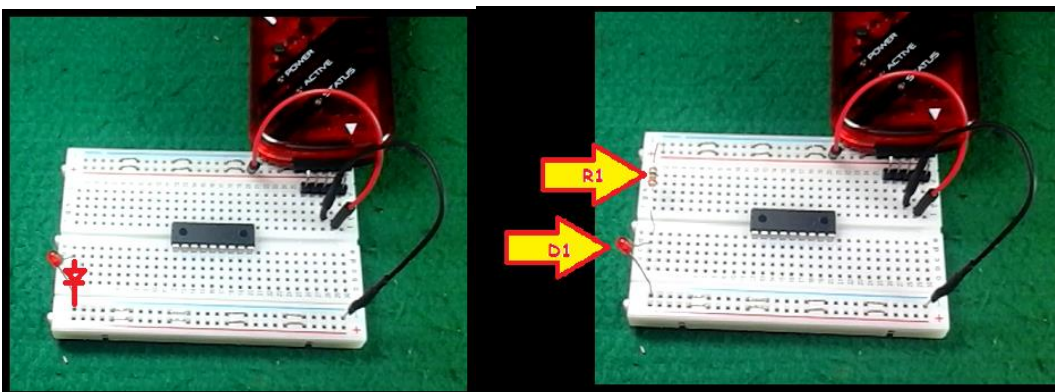


Você deve ligar o pino 3 GND do PICkit3 ao sinal de terra de todo o circuito (fio preto) completando o circuito de alimentação!

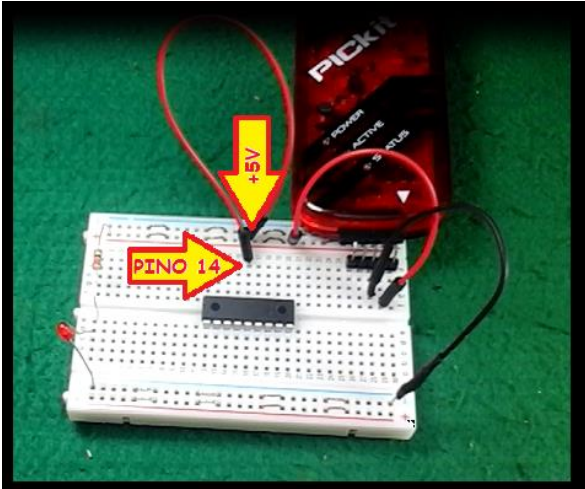


Conecte o LED D1 e o resistor R1 entre o +5V e o terra.

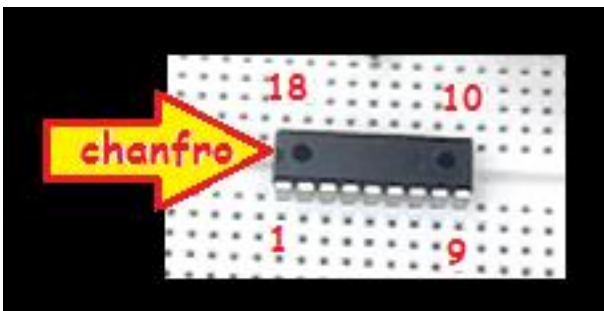
No LED o terminal mais comprido é o anodo e deverá ser ligado no resistor.



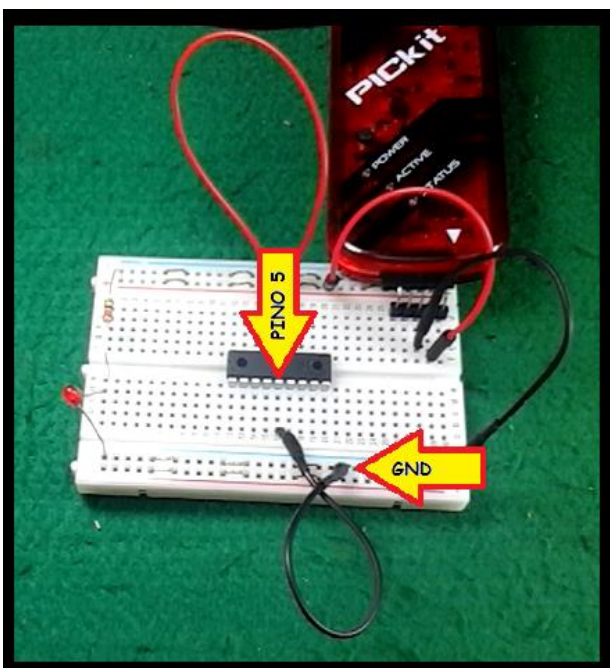
Para completar o circuito de alimentação ligue o VDD(+5V) no pino 14 do PIC (fio vermelho)!



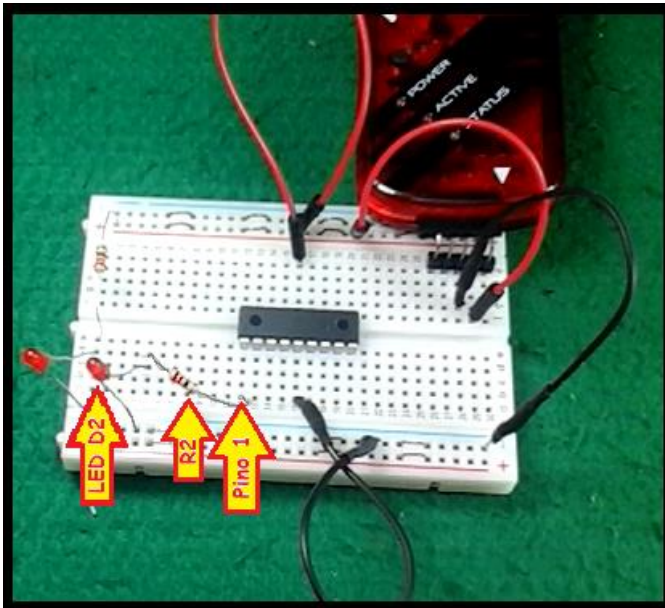
Observar a sequência correta de contagem dos pinos no CI!



E o sinal de terra (GND) ao pino 5 do PIC (fio preto)!

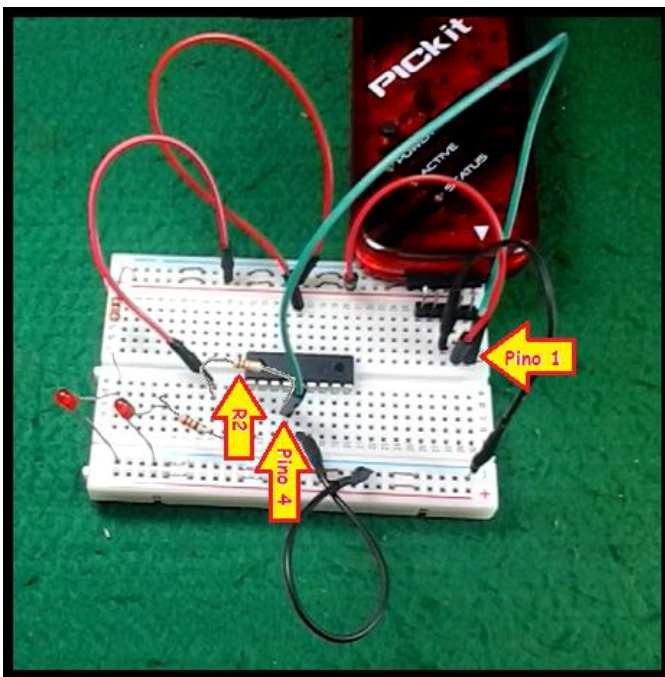


Ligando o LED D2 ao pino 1 do PIC!



Você deve ligar um terminal do resistor de pullup R3 no pino 4 MCRL do PIC, e o outro através do fio vermelho no VDD (+5V)!

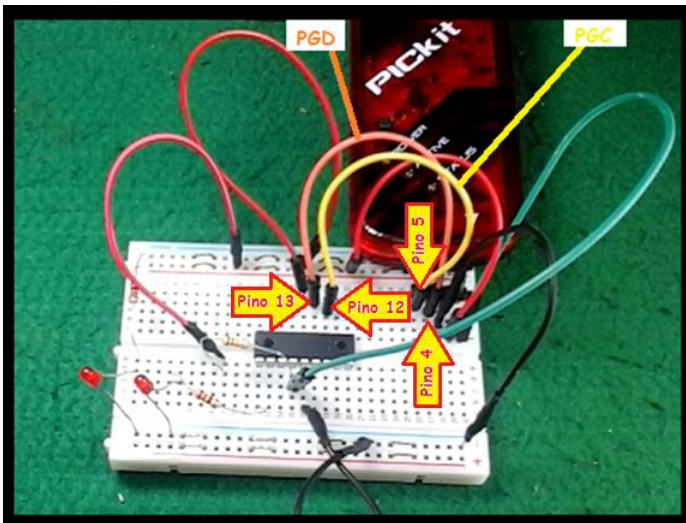
Depois você deverá ligar o pino 1 do PICkit3 ao pino 4 do PIC (fio verde).



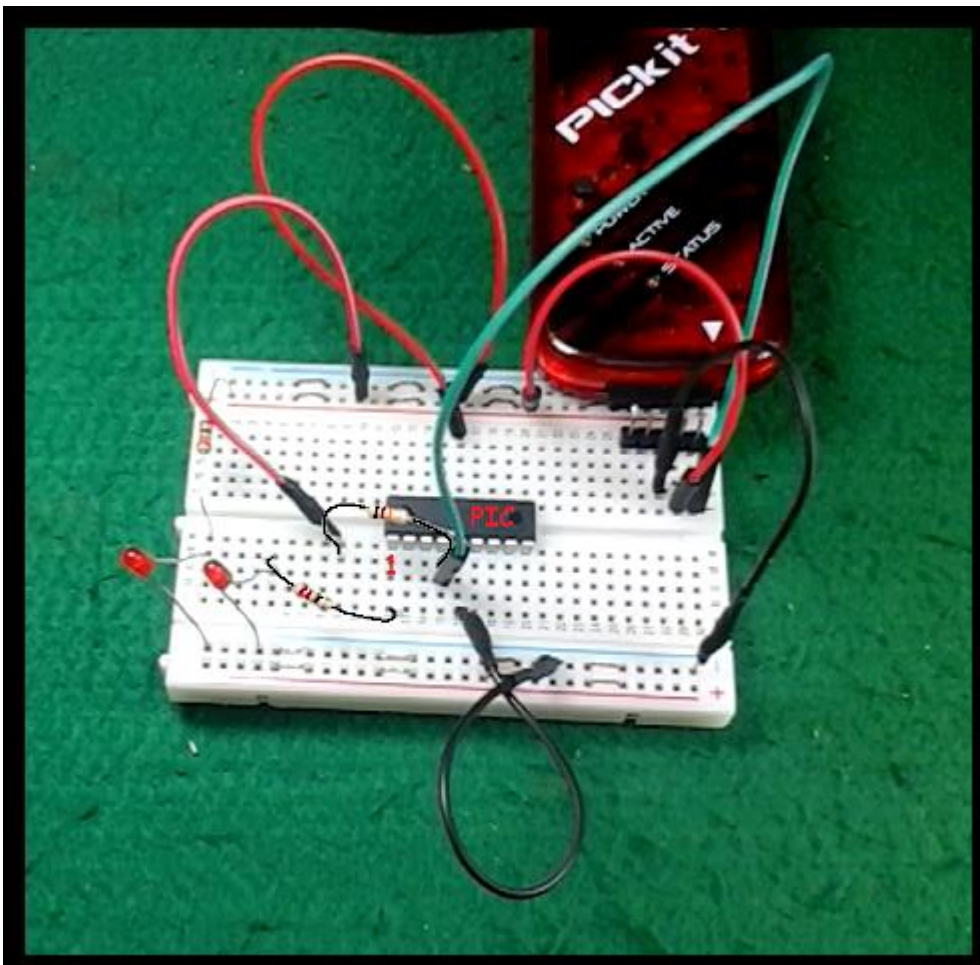
A função deste sinal durante a programação do PIC é fornecer a tensão de programação vpp.

Quando o PIC não está sendo programado a função do pino 4 é de MASTER RESET, por isto você deve ligar o resistor de 10K ao VDD (+5V) para garantir que o PIC não fique ressetado!

O PINO 4 PGD (fio laranja) e O PINO 5 PGC (fio amarelo) do PICKIT3 funcionam como pinos de comunicação para carregar o programa no PIC, deverão ser ligados aos pinos PGC (PINO 12) e PGD do PIC.



A montagem final é mostrada abaixo!

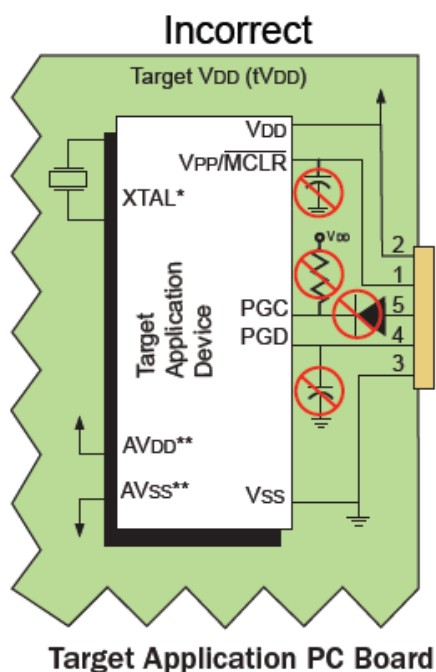


Precauções:

No folheto aparece a observação abaixo descrevendo os cuidados que você deve ter ao ligar o PicKit3 no microcontrolador!

Não use os componentes descritos com o "X" na figura abaixo, isto é, não use capacitores em paralelo nem resistor de pull up nem diodos em série com os pinos de programação PGC e PGD!

Target Circuit Design Precautions



- **Do not use greater than 100 μ F capacitance on VDD** – depending on the overall load, it will prevent the target from powering quickly when PICKit 3 is the source of power.
- **Do not use capacitors on MCLR** – they will prevent fast transitions of VPP.
- **Do not use pull-ups on PGC/PGD** – they will divide the voltage levels since these lines have 4.7 k Ω pull-down resistors in PICKit 3.
- **Do not use multiplexing on PGC/PGD** – they are dedicated for communications to PICKit 3.
- **Do not use capacitors on PGC/PGD** – they will prevent fast transitions on data and clock lines during programming and debug communications.
- **Do not use diodes on PGC/PGD** – they will prevent bidirectional communication between PICKit 3 and the target PIC[®] MCU.

O programa.

Para programar o PIC você pode usar qualquer interface, neste exemplo o projeto foi desenvolvido no MpLab!

O programa completo é mostrado abaixo.

```
//Nome:      Data
//teste com o PIC16F628 sem cristal
// PIC16F628A Configuracao dos bits
#include <xc.h>
#define _XTAL_FREQ 4000000 // Frequencia interna do oscilador 4MHz

#pragma config FOSC = INTOSCIO // Oscillator Selection bits (INTOSC oscillator: I/O function on RA6/OSC2/CLKOUT pin, I/O function on RA7/OSC1/CLKIN)
#pragma config WDTE = OFF // Watchdog Timer Enable bit (WDT disabled)
#pragma config PWRTE = OFF // Power-up Timer Enable bit (PWRT disabled)
#pragma config MCLRE = ON // RA5/MCLR/VPP Pin Function Select bit (RA5/MCLR/VPP pin function is digital input, MCLR internally tied to VDD)
#pragma config BOREN = OFF // Brown-out Detect Enable bit (BOD disabled)
#pragma config LVP = OFF // Low-Voltage Programming Enable bit (RB4/PGM pin has digital I/O function, HV on MCLR must be used for programming)
#pragma config CPD = OFF // Data EE Memory Code Protection bit (Data memory code protection off)
#pragma config CP = OFF // Flash Program Memory Code Protection bit (Code protection off)

int n;//variável de uso geral
void main() {
//Escreva o seu programa aqui
CMCON=0x07;//configura portas RA como digitais
#define led PORTAbits.RA2 //defina a porta RA2 com o nome de led
TRISA2=0;//1=entrada 0=saida programa a porta RA como saída.
while(1){
//pisca_pisca quando a chave estiver aberta
led=1;//liga led
for(n=0;n<10000;n++){//atraso led ligado
led=0;//desliga led
for(n=0;n<10000;n++){//atraso led desligado
} //Fim do while(1)
} //fim do main()
}
```

Nas linhas abaixo os fusíveis internos são programados.

A linha em amarelo mostra a programação para uma aplicação usando o oscilador interno!

O MCLRE deve estar habilitado, por isto o resistor de pull up no pino 4. O restante deve estar desabilitado!

```
//Nome:      Data
//teste com o PIC16F628 sem cristal
// PIC16F628A Configuracao dos bits
#include <xc.h>
#define _XTAL_FREQ 4000000 // Frequencia int

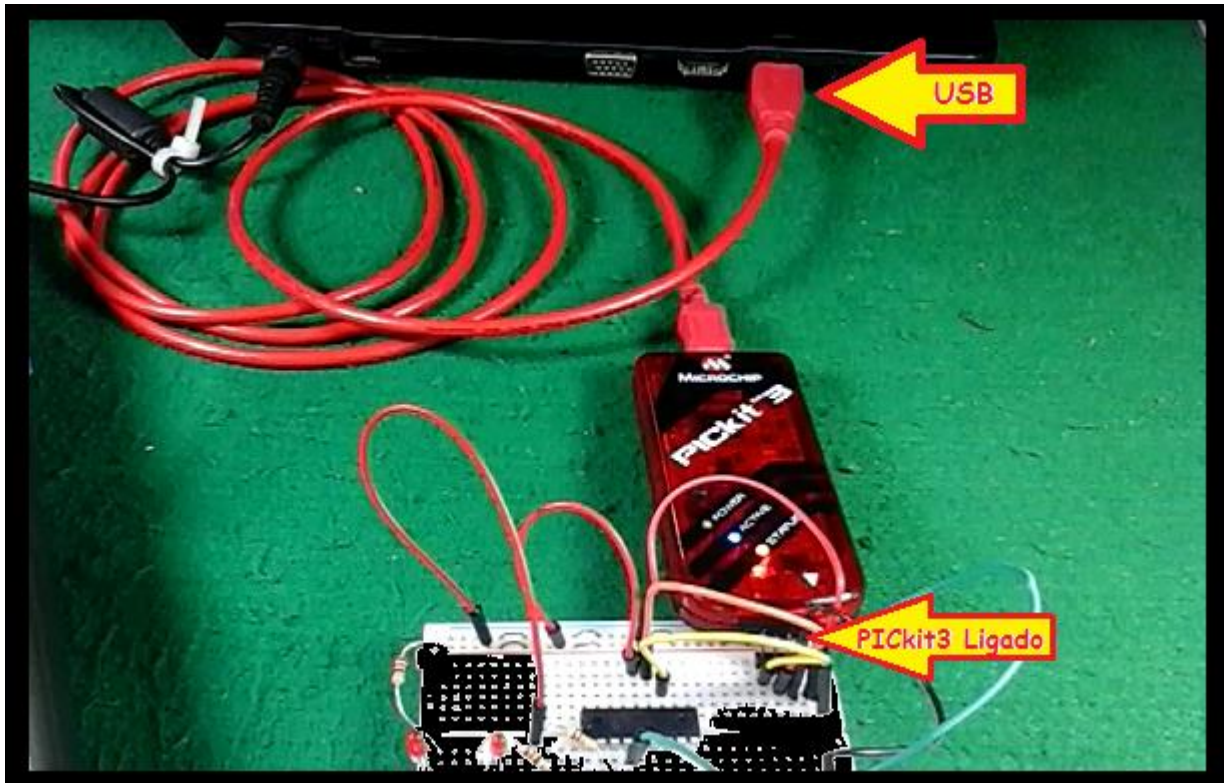
#pragma config FOSC = INTOSCIO // Oscillator Sel
#pragma config WDTE = OFF // Watchdog Timer
#pragma config PWRTE = OFF // Power-up Timer
#pragma config MCLRE = ON // RA5/MCLR/VPP Pi
#pragma config BOREN = OFF // Brown-out Dete
#pragma config LVP = OFF // Low-Voltage Pr
#pragma config CPD = OFF // Data EE Memory
#pragma config CP = OFF // Flash Program
```

A função main() é mostrada abaixo, o programa do pisca-pisca é bem simples!

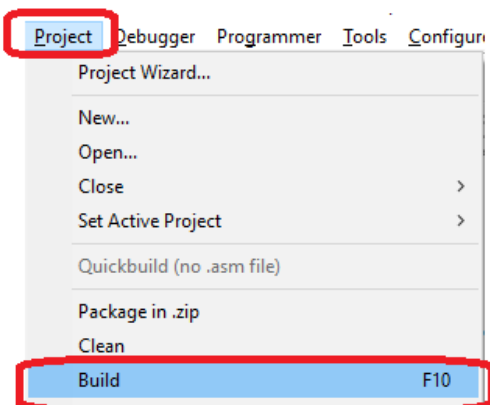
```
int n;//variável de uso geral
void main() {
    //Escreva o seu programa aqui
    CMCON=0x07;//configura portas RA como digitais
    #define led PORTAbits.RA2 //defina a porta RA2 com o nome de led
    TRISA2=0;//1=entrada 0=saida programa a porta RA como saída.
    while(1){
        //pisca_pisca quando a chave estiver aberta
        led=1;//liga led
        for(n=0;n<10000;n++){ }//atraso led ligado
        led=0;//desliga led
        for(n=0;n<10000;n++){ }//atraso led desligado
    } //Fim do while(1)
} //fim do main()
```

Transferindo o programa para o PIC.

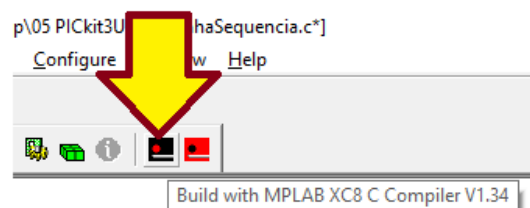
Estes passos deverão ser feitos com o cabo ligado no PicKit3 e este ligado no circuito da protoboard!



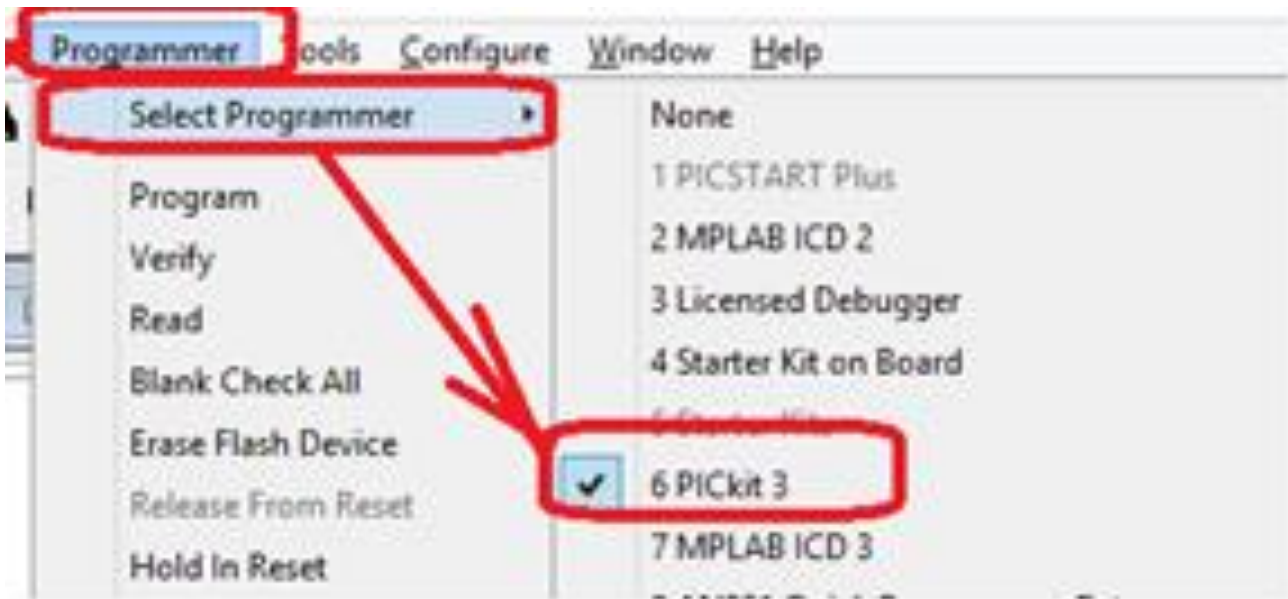
Vá noo MpLab compile o programa!



OU



Vá no menu e selecione o programador para PicKit3 como é mostrado a seguir:
Programmer>>SelectProgrammer>>PICKit3



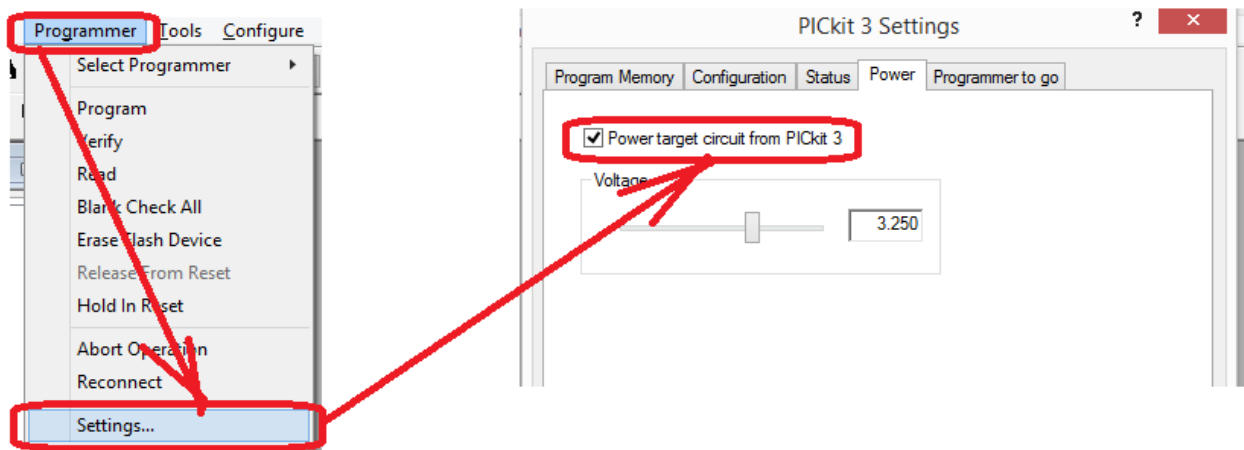
O PICKit irá se comunicar com o PC e em seguida tentará se comunicar com o PIC montado no protoboard.



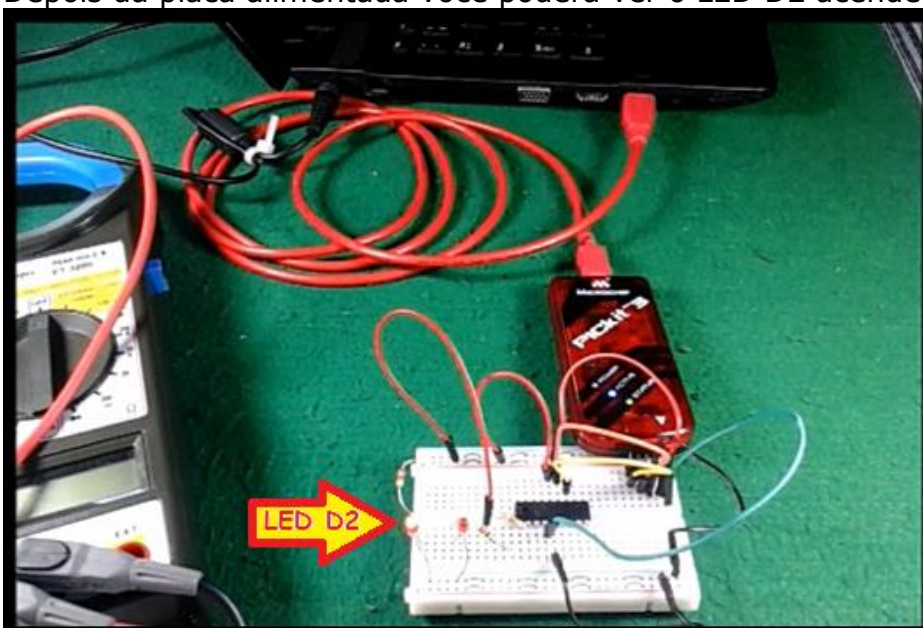
Algumas vezes a comunicação com o PIC montado no circuito não é fechada porque esta placa não está alimentada, neste caso a mensagem abaixo irá ser mostrada.

```
Build | Version Control | Find in Files | PICKit 3 |
PICKit 3 detected
Connecting to PICKit 3...
Running self test...
Self test completed
Firmware Suite Version..... 01.28.90
Firmware type.....PIC18F
PICKit 3 Connected.
PK3Err0045: You must connect to a target device to use PICKit 3.
```

Para ligar a alimentação no circuito você deverá seguir os passos descritos abaixo, isto vai ser necessário somente a primeira vez.



Depois da placa alimentada você poderá ver o LED D2 acender!



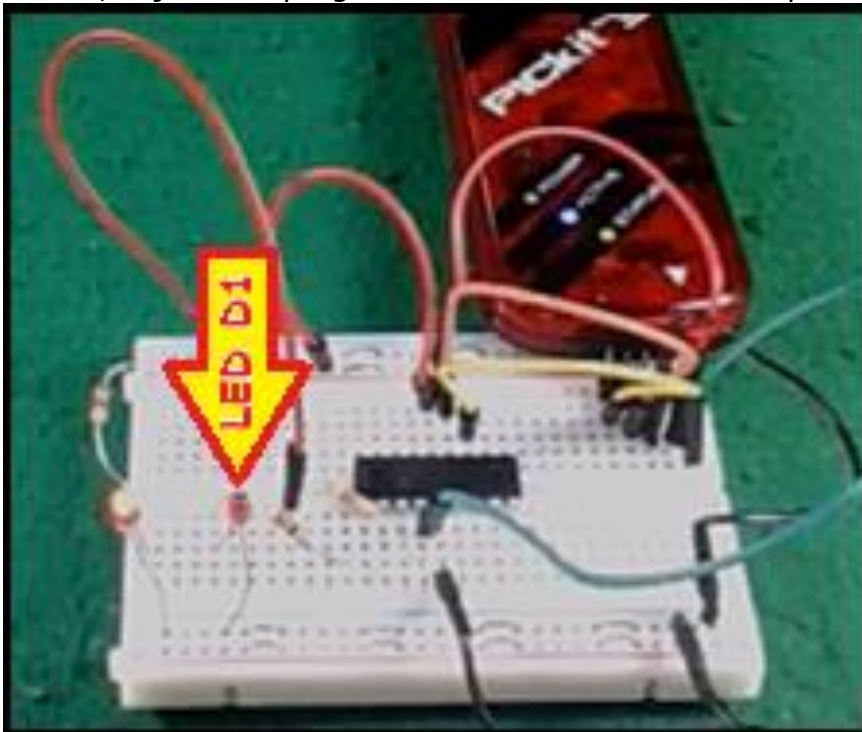
Na tela do MpLab você verá a mensagem no console indicando que do PIC foi detectado!.

```
firmware type.....PIC10f
PICkit 3 Connected.
PK3Err0045: You must connect to a target device to use PICkit
3.
Device ID Revision = 0000001b
```

Selecione o Download para baixar o seu programa.



Pronto, veja o seu programa rodando com o LED D1 piscando!



Conclusão.

Usar o PicKit3 é muito simples e uma vez feito o investimento inicial você poderá programar a maioria de PICs e logo terá o seu investimento de volta!

Referências.

Bibliografia.

Manuais:

Sites: