

Vitor Amadeu Souza

Projetos com

Raspberry Pi Pico

Volume único

© 2022 by Cerne Tecnologia e Treinamento Ltda.

© 2022 by Vitor Amadeu Souza

Nenhuma parte desta publicação poderá ser reproduzida sem autorização prévia e escrita de **Cerne Tecnologia e Treinamento Ltda.** Este livro publica nomes comerciais e marcas registradas de produtos pertencentes a diversas companhias. O editor utiliza as marcas somente para fins editoriais e em benefício dos proprietários das marcas, sem nenhuma intenção de atingir seus direitos.

Outubro de 2022

Direitos reservados por:

Cerne Tecnologia e Treinamento Ltda

Produção: Cerne Tecnologia e Treinamento

E-mail da Empresa: cerne@cerne-tec.com.br

Home Page: www.cerne-tec.com.br.com.br

Atendimento ao Consumidor: sac@cerne-tec.com.br

Contato com o Autor: vitor@cerne-tec.com.br



FEITO NO BRASIL

“Não podemos pensar em duas coisas ao mesmo tempo.”

Blaise Pascal

Kits Didáticos e Gravadores da Cerne Tecnologia

A Cerne tecnologia têm uma linha completa de aprendizado para os RASPBERRYes da família PIC, 8051, Holtek, dsPIC, RASPBERRY, RASPBERRY, etc. Veja os detalhes de um de nossos kits.



Kit Cerne Arduino

- Microcontrolador ATMEGA8;
- Comunicação serial RS232;
- Alimentação de 12V;
- Pinos de I/O;
- Gravação ICSP.

Uma linha completa de componentes para o desenvolvimento de seus projetos eletrônicos como displays, PICs, botões, leds, cristais e etc. Visite a nossa página na Internet, no endereço **www.cerne-tec.com.br** e conheça melhor nossos serviços e produtos.



www.cerne-tec.com.br

Sumário

I Metodologia de desenvolvimento	6
II Controle de potência de motor AC	7
III Varredura de leds RGB	32
IV Comunicação PS2	62
V RFID	75
VI Motor de passo bipolar	87
VII Geração de barras de vídeo	119
VIII Potenciômetro digital	124
IX Recepção IR no protocolo RC5	130
X Comunicação SPI	143
XI Partida de motor AC	158
XII SGA	218
XIII Expansão de saídas com shift register	231
XIV PGA	258
XV Comunicação com DDS AD9835	276
XVI Efeito Peltier	296
XVII Acelerômetro LIS302DL	305
XVIII Comunicando com DAC externo	321
XIX Detecção de DTMF	335
XX Analisador Lógico	350
XXI Horímetro	363
XXII Comunicando com ADC externo	373
XXIII Recepção IR no protocolo NEC	389
XXIV Expandindo o número de entradas com shift register	403
XXV Varredura de displays de 7 segmentos	421

Capítulo I

Metodologia de desenvolvimento

1. Introdução

Esta literatura é uma coletânea de projetos com a Raspberry Pi Pico sendo uma continuação da obra *Aplicações eletrônicas no Raspberry Pi Pico – Programado em Python* (2022) do mesmo autor e editora, onde outros exemplos são explorados de modo que o leitor possa ampliar seu embasamento teórico e prático para desenvolver mais aplicações nesta ferramenta. É importante que o leitor tenha ciência dos assuntos abordados nesta obra citada, para que haja um melhor aproveitamento do conteúdo a ser apresentado.

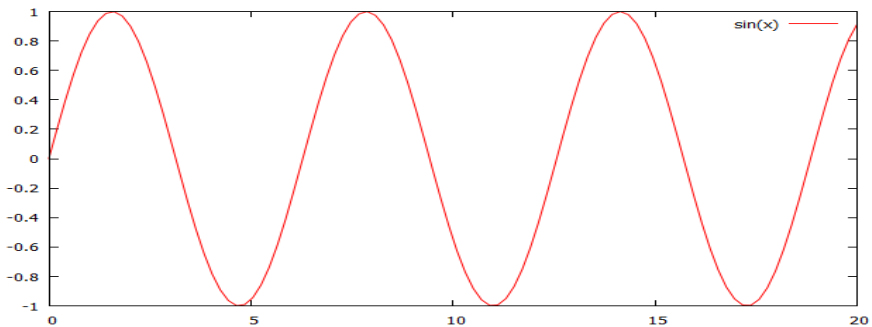
A placa didática utilizada é a Raspberry Pi Pico, onde os recursos para testar os circuitos propostos são conectados através das conexões disponíveis na lateral do kit. Tal kit está à venda no site www.cerne-tec.com.br.

Capítulo II

Controle de potência de motor AC

1. Fundamentos

A corrente alternada ou CA (em inglês AC - Alternating Current) é uma corrente elétrica cuja magnitude e direção da corrente variam ciclicamente, ao contrário da corrente contínua, cuja direção permanece constante e que possui pólos positivo e negativo definidos. A forma de onda usual em um circuito de potência CA é senoidal por ser a forma de transmissão de energia mais eficiente. Entretanto, em certas aplicações, diferentes formas de ondas são utilizadas tais como triangular ou ondas quadradas. A seguir está apresentado um exemplo de onda senoidal.



Este tópico mostra como utilizar o RASPBERRY para realizar o controle de potência do motor AC. Para isso, apresenta o hardware utilizado para disparar o TRIAC assim como verificar o sincronismo com a rede. De modo prático, será elaborado um exemplo para tratar o controle de potência utilizando botões. No decorrer da obra também é apresentado um esquema elétrico onde o leitor poderá montar o hardware básico para simular o funcionamento do controle de motor AC de no máximo 3/4 HP, já que o hardware proposto possui tal característica.

Obs: *Faça todos os testes utilizando um transformador isolador entre a rede elétrica e o esquema apresentado. Use óculos de proteção e nunca deixe de usar os fusíveis de proteção, disjuntor e todos os elementos pertinentes para sua segurança. O motor utilizado deve ser de baixa potência pelo dimensionamento dos TRIACs, no máximo 8 A por fase.*

2. História da corrente alternada

A corrente alternada surgiu quando Nikola Tesla foi contratado por J. Westinghouse para construir uma linha de transmissão entre Niágara e Búfalo, em NY. Thomas Edison fez o possível para

desacreditar Tesla, mas o sistema polifásico de Tesla foi adotado. A Corrente Alternada é a forma mais eficaz de se transmitir uma corrente elétrica por longas distâncias. Nela os elétrons invertem o seu sentido várias vezes por segundo.

Na primeira metade do século XX havia sistemas de Corrente Alternada de 25 Hz no Canadá (Ontário) e no norte dos EUA. Em alguns casos alguns destes sistemas (por exemplo, nas quedas de Niágara) perduram até hoje por conveniência das fabricas industriais que não tinham interesse em trocar o equipamento para que operasse a 60 Hz. As baixas frequências facilitam construção de motores de baixa rotação.

Há também sistemas de 16,67 Hz em ferrovias da Europa (Suíça e Suécia).

Sistemas AC de 400 Hz são usados na indústria têxtil, aviões, navios, espaçonaves e em grandes computadores.

No Brasil a variação (frequência) da rede elétrica é de 60Hz. Na América do Sul, além do Brasil, também usam 60Hz o Equador, Peru, Venezuela e a Colômbia. Em outros países, por exemplo, a Argentina, a Bolívia, o Chile e o Paraguai, bem como na Europa é usada a frequência de 50Hz.

A Corrente Alternada foi adotada para transmissão de energia elétrica a longas distâncias devido à facilidade relativa que esta apresenta para ter o valor de sua tensão alterada por intermédio de transformadores. No entanto as primeiras experiências e transmissões foram feitas com Corrente contínua (CC ou, em inglês, DC).

3. Função da tensão AC

Correntes alternadas são usualmente associadas com tensões alternadas. Uma tensão CA senoidal v pode ser descrita matematicamente como uma função do tempo, pela seguinte equação:

$$v(t) = A \sin(\omega t)$$

Onde:

A é a *amplitude* em volts (também chamada de *tensão de pico*),
 ω é a frequência angular em radianos por segundo, e
 t é o tempo em segundos.

Como frequência angular possui difícil compreensão prática, esta fórmula é comumente reescrita da seguinte forma:

$$v(t) = A \sin(2\pi f t)$$

Onde f é a frequência em hertz.

O valor de pico-a-pico de uma tensão alternada é definida como a diferença entre seu pico positivo e seu pico negativo. Desde o valor máximo de seno (x) que é +1 e o valor mínimo que é -1, uma tensão CA oscila entre + A e - A . A tensão de pico-a-pico, escrita como V_{pp} , é, portanto $(+A) - (-A) = 2 \times A$.

Geralmente a tensão CA é dada quase sempre em seu valor eficaz, que é o valor quadrático médio desse sinal elétrico (em inglês é chamado de *root mean square* ou **rms**), sendo escrita como V_{ef} (ou V_{rms}). Para uma tensão senoidal:

$$V_{ef} = V_{RMS} = \frac{A\sqrt{2}}{2} \text{ ou } V_{ef} = V_{RMS} = A.0,707$$

O valor de V_{ef} ou V_{RMS} é útil no cálculo da potência consumida por uma carga. Se a tensão CC de V_{CC} transfere certa potência P para a carga dada, então uma tensão CA de V_{ef} irá entregar a mesma potência média P para a mesma carga se $V_{ef} = V_{CC}$. Por este motivo, RMS é o modo normal de medição de tensão em sistemas de potência.

Para ilustrar estes conceitos, considere a tensão de 220V AC usada em alguns estados brasileiros. Ela é assim chamada porque seu valor eficaz (RMS) é, em condições normais, de 220V. Isto quer dizer que ela tem o mesmo efeito joule, para uma carga resistiva, que uma tensão de 220V_{CC}. Para encontrar a tensão de pico (amplitude), podemos modificar a equação acima para:

$$V_P = V_{RMS} \sqrt{2} \text{ ou } V_P = V_{EF} \sqrt{2}$$

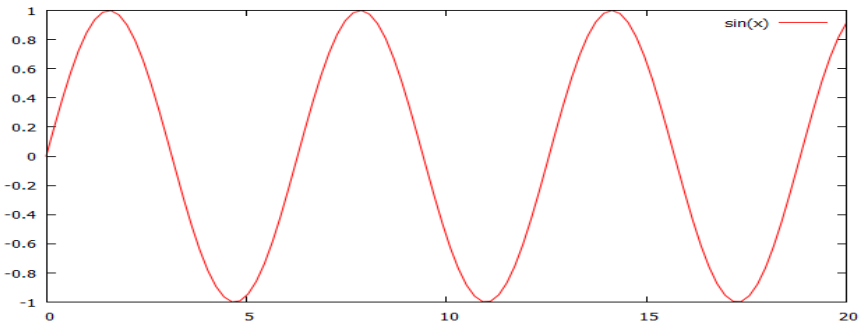
Para 220 V_{CA}, a tensão de pico V_p ou A é, portanto, $V_p = 220 \cdot \sqrt{2} = 311V$ (aprox.). O valor de pico-a-pico $V_{p,p}$ de 220V CA é ainda mais alta: $2 \cdot 220 \cdot \sqrt{2} = 622V$ (aprox.)

Note que para tensões não senoidais, temos diferentes relações entre seu pico de magnitude valor eficaz. Isso é de fundamental importância ao se trabalhar com elementos do circuito não lineares que produzem correntes harmônicas, como retificadores.

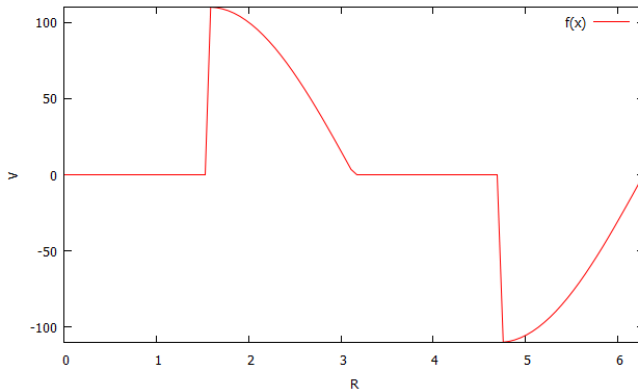
4. Técnica de controle de ângulo de disparo

A técnica de controle de ângulo de disparo consiste em controlar o ponto de início de uma senoide. Por exemplo, suponha

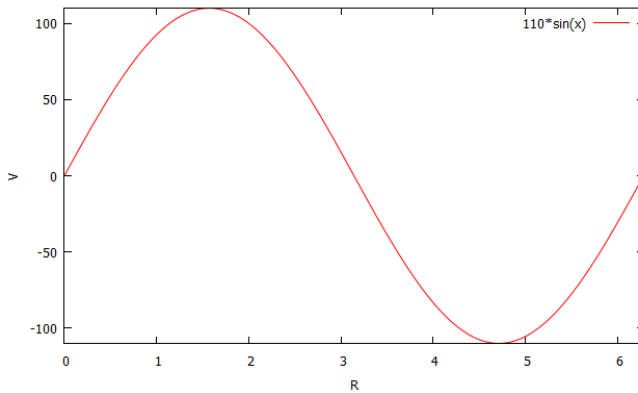
que está sendo utilizado uma rede de 110V_{AC} RMS a 60Hz. Cada período terá o tempo definido por $T=1/F$ e desta forma tem-se o tempo de aproximadamente 16ms sabendo que $1/60$ é igual a este valor. Observe por exemplo, a senóide apresentada abaixo:



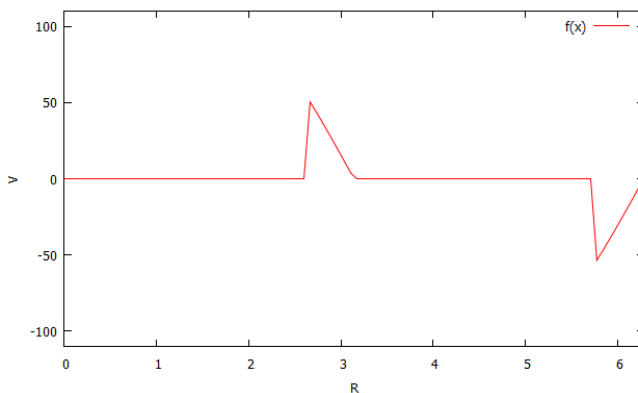
Se, por exemplo, esta fase estiver em uma carga como um resistor, o efeito será como se estivéssemos aplicando uma tensão DC de 110V na mesma. Porém se, por exemplo, controlRaspberrys o início desta onda, poderá alterar assim a potência fornecida a carga, e no caso de um resistor, diminuirá o seu aquecimento, no caso de uma lâmpada a sua luminosidade e no caso de um ventilador, a sua rotação (considere todas as cargas lineares). Observe um exemplo na figura abaixo, onde neste caso a tensão rms já não é mais 110V e sim 55V:



Observe que neste caso, 50% de cada semi-ciclo da senóide fica ativado. Nesta situação do ângulo de 90° até 180° para o primeiro semi-ciclo e de 270° a 360° para o segundo. Note que através desta técnica é possível controlar a potência de uma carga AC através do RASPBERRY. No exemplo a seguir, toda a onda fica ativa no período, onde teremos neste caso potência plena na carga.



Já no próximo exemplo temos uma potência muito pequena sendo aplicada a carga, já que a mesma tem partida no ângulo de 120° e 300° , gerando assim uma pequena tensão RMS.

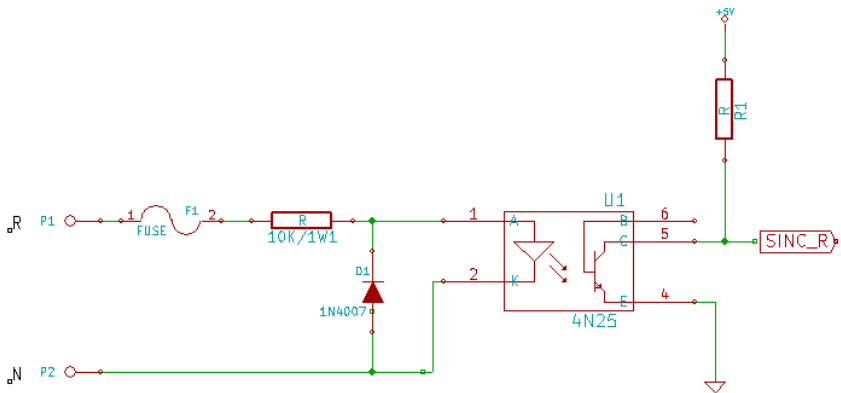


O mesmo conceito apresentado para uma fase será usado para controlar um motor AC monofásico, pois sua potência, ou seja, seu

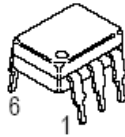
torque e velocidade são proporcionais a tensão RMS aplicada ao mesmo.

5. Hardware de controle de ângulo de disparo

Observe que para fazer este controle, é necessário detectar o início da senóide, de forma a controlar o seu ângulo de disparo. Para isso, utiliza-se o CI 4N25, para que ao iniciar uma senóide, o mesmo seja alimentado e assim possa informar ao RASPBERRY o início e através de temporizações o mesmo possa controlar em que ponto aciona o TRIAC e conseqüentemente controla a potência de uma carga externa. Observe como está projetado este circuito, que é chamado de circuito de sincronismo:

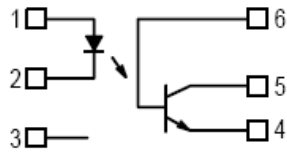


A pinagem do 4N25 está apresentada a seguir:



STANDARD THRU HOLE

SCHEMATIC

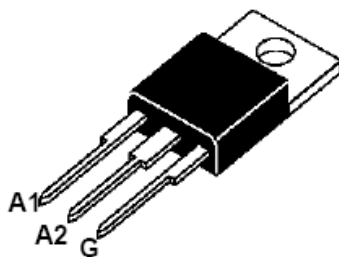
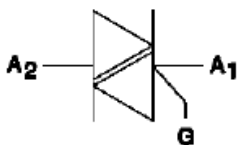


- PIN 1. LED ANODE
- 2. LED CATHODE
- 3. N.C.
- 4. EMITTER
- 5. COLLECTOR
- 6. BASE

A saída de sincronismo (SINC_R) é acionada no início do ciclo da senóide e este pino conectado a um pino de entrada do RASPBERRY, permitindo a este detectar o início da senóide.

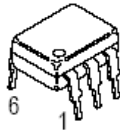
6. Conhecendo o MOC3023 e o BTA08

Para acionar cada uma das saídas é usado um fotodiac chamado MOC3023 e o TRIAC BTA08 que pode alimentar uma carga de até 8A. Observe abaixo a pinagem do TRIAC e do foto diac.



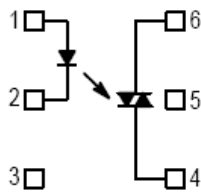
TO220AB
(Plastic)

STYLE 6 PLASTIC



STANDARD THRU HOLE
CASE 730A-04

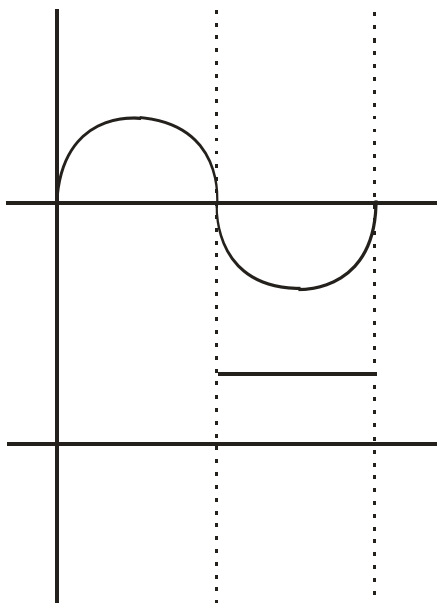
SCHEMATIC



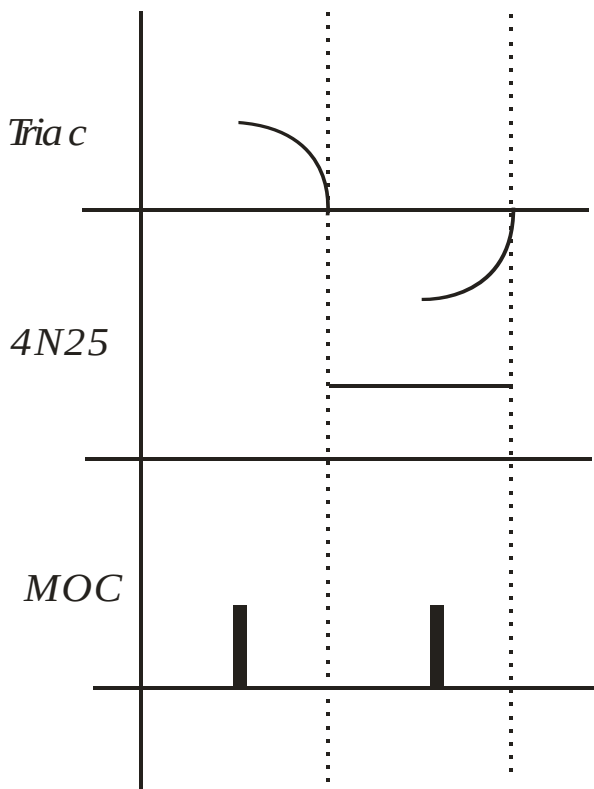
1. ANODE
2. CATHODE
3. NC
4. MAIN TERMINAL
5. SUBSTRATE
DO NOT CONNECT
6. MAIN TERMINAL

A lógica implementada permite ao RASPBERRY detectar sincronismo e assim aguarde um tempo dependendo da potência a ser aplicada a carga.

Suponha, por exemplo, que deseja-se colocar a potência máxima, neste caso o ciclo completo. Imediatamente após receber o pulso de sincronismo o programa deve acionar o MOC, permitindo que o TRIAC venha a conduzir o ciclo completo da senóide. Se, por exemplo, o objetivo fosse deixar a tensão RMS em 50%, então deve-se aguardar 4ms (já que o semiciclo tem 8 ms) e passado estes 4ms, acionar o MOC. Mesma ideia é válida para o semiciclo seguinte. Observe o seguinte resultado na saída do 4N25 de acordo com o semiciclo:



Desta forma, a saída SINC fica em nível baixo quando o semiciclo for positivo (a lógica é invertida mesmo, pois a saída do 4N25 é um coletor aberto e quando o mesmo conduz, a saída fica aterrada) e alto no semiciclo negativo. Se quiser deixar a carga com uma tensão RMS de 50% conforme abordado tem-se então a seguintes cartas de tempo:



Observe que após receber o pulso de sincronismo, o sistema aguarda o tempo de acordo com a potência desejada e logo em seguida, gera um pulso na saída do MOC, de forma a acionar o mesmo (pulso típico de 1ms). A partir deste momento, o TRIAC passa a conduzir até que chegue ao final do semiciclo, onde o mesmo volta a ficar no estado de corte.

7. Controle de potência de motor AC

O objetivo deste capítulo é demonstrar o uso do controle de ângulo de disparo para a fase que alimenta o motor AC, controlando assim sua velocidade, potência e torque. A ideia é usar quatro botões conectados ao RASPBERRY para gerar tempos diferentes de disparo e assim alterar a potência fornecida à carga. A temporização é estipulada de acordo com a tabela a seguir:

Botão	Pino	Tempo
1	GPIO14	4 ms
2	GPIO15	5 ms
3	GPIO16	6 ms
4	GPIO17	7 ms

A contagem de tempo inicia no momento em que for detectado o início da senóide através do pulso de sincronismo e depende de

qual botão estiver pressionado naquele instante. Por exemplo, no momento em que for detectado o início de uma senóide será verificado imediatamente se há algum botão pressionado. Havendo, em seguida aguarda-se um tempo de acordo com o botão e liga-se a saída do MOC de modo que ele ative o TRIAC e assim forneça a esta saída à potência requerida de acordo com o intervalo de tempo configurado.

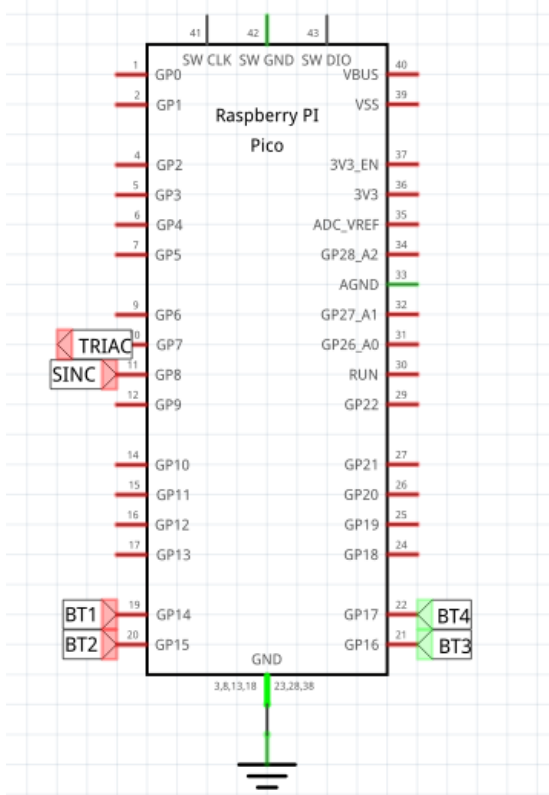
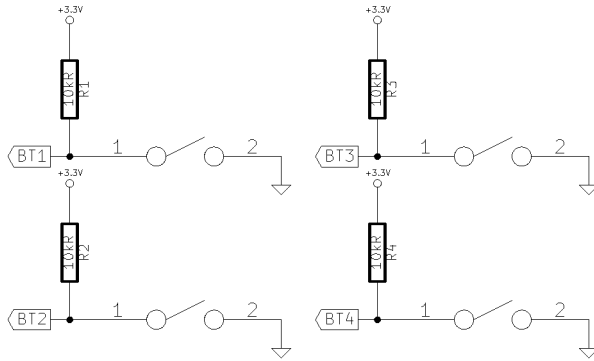
Observe a seguir o esquema elétrico proposto para este exemplo. Inicialmente, recomenda-se que seja testado com uma lâmpada incandescente conectada na saída AC, onde se observa que a intensidade da carga (lâmpada) altera de acordo com o botão. Se nenhum botão estiver pressionado a saída fica desligada. Se o botão do pino GPIO14 estiver pressionado, é observado na saída 50% da tensão RMS, já cada semiciclo é acionado em sua metade, pois o disparo ocorre no instante 4 ms (observe que cada semiciclo possui intervalo de 8 ms). Mesma ideia para os botões conectados as entradas GPIO15, GPIO16 e GPIO17, onde o segundo botão proporciona 37,5% da tensão RMS já que o semiciclo é acionado 5 ms após o seu início. O botão conectado ao pino GPIO16 proporciona 25% da tensão RMS, já que o acionamento do TRIAC é feito 6 ms

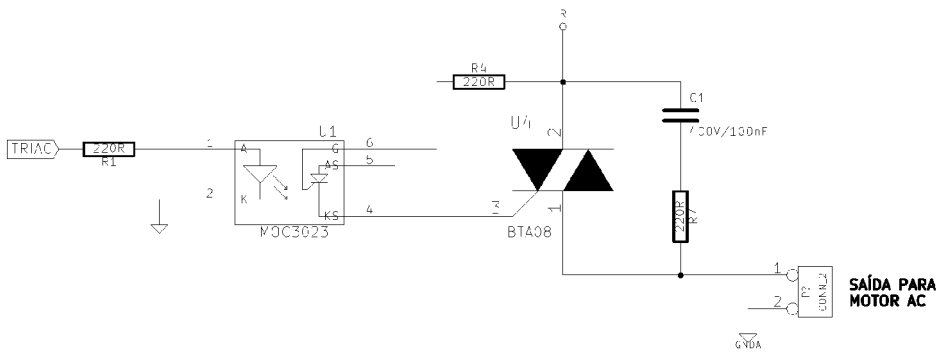
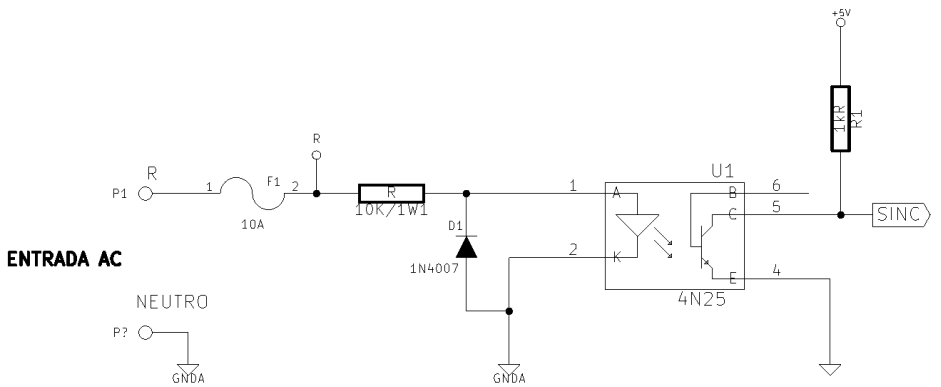
após o início do semiciclo e finalmente o último botão proporciona 12,5 % da tensão RMS.

Note que dependendo do motor, talvez o mesmo não consiga vencer a inércia ou fornecer o torque necessário para partir, sendo necessário neste caso alterar a configuração de tempo. Todavia, usando inicialmente lâmpadas como carga será possível observar este efeito.

8. Esquema elétrico

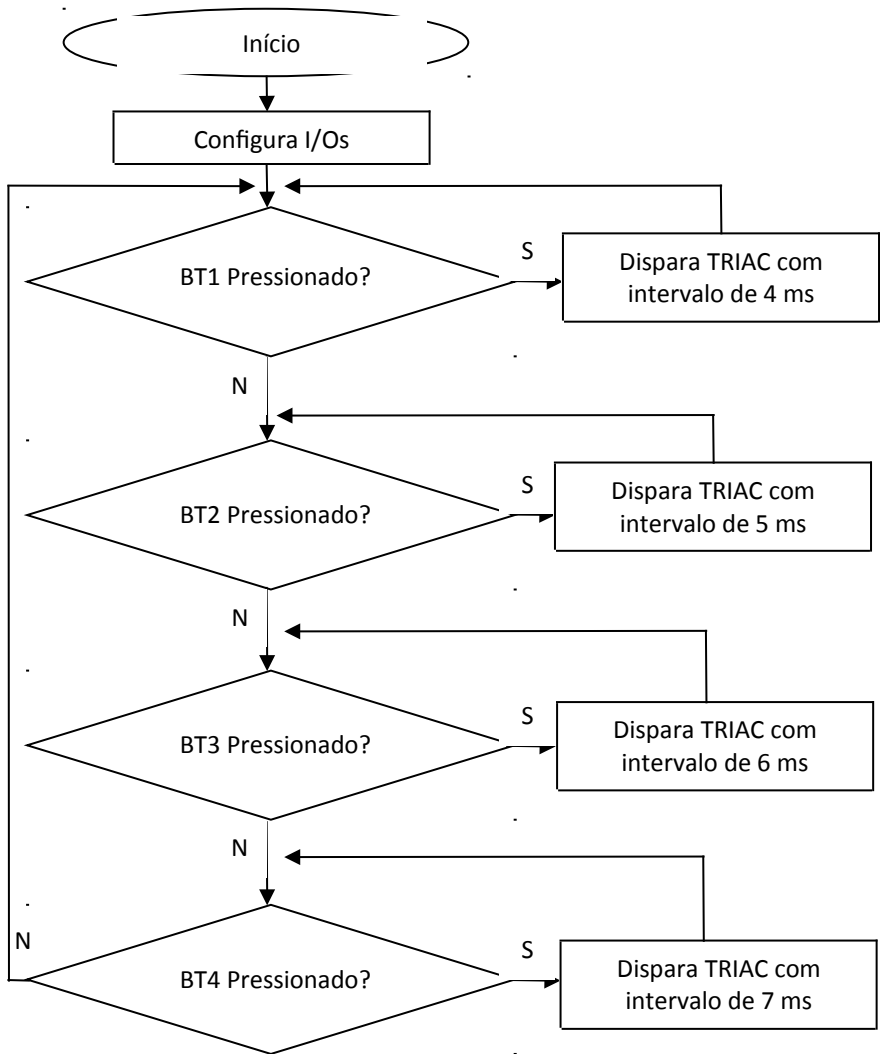
O esquema elétrico para este exemplo está apresentado a seguir. Note que há o circuito de sincronismo na fase que alimenta o motor assim como o TRIAC para seu controle. O TRIAC pode suportar correntes de até 8 A usando dissipadores. Pode ser interessante após o motor atingir pleno funcionamento, chavear a carga através de uma contatora e retirar o TRIAC como passagem de corrente, o que faz que ele aqueça menos.





9. Fluxograma

O fluxograma está apresentado a seguir.



10. Código fonte

O código fonte para este experimento está apresentado a seguir. Note que o mesmo faz uso das estruturas básicas apresentadas no decorrer do estudo do RASPBERRY.

```

#* * * * *
#*      Cerne Tecnologia      *
#*      www.cerne-tec.com.br  *
#*  Projetos em RASPBERRY - MOTOR AC *
#* * * * *
from machine import Pin
import time

#* * * * *
#*      Declaração de I/Os      *
#* * * * *
TRIACP=10
SINCP=11
BT1P=19
BT2P=20
BT3P=21
BT4P=22

#* * * * *
#*      Função Principal      *
#* * * * *
TRIAC = Pin(TRIACP, Pin.OUT)
SINC  = Pin(SINCP, Pin.OUT)
BT1   = Pin(BT1P, Pin.IN,Pin.PULL_UP)
BT2   = Pin(BT2P, Pin.IN,Pin.PULL_UP)
BT3   = Pin(BT3P, Pin.IN,Pin.PULL_UP)
BT4   = Pin(BT4P, Pin.IN,Pin.PULL_UP)

while(1):

    while(SINC.value()==1):
```

```

pass

if BT1.value()==0:
    time.sleep(0.004)
    TRIAC.on()
    time.sleep(0.0001)
    TRIAC.off()

if BT2.value()==0:
    time.sleep(0.005)
    TRIAC.on()
    time.sleep(0.0001)
    TRIAC.off()

if BT3.value()==0:
    time.sleep(0.006)
    TRIAC.on()
    time.sleep(0.0001)
    TRIAC.off()

if BT4.value()==0:
    time.sleep(0.007)
    TRIAC.on()
    time.sleep(0.0001)
    TRIAC.off()

while (GPIO.input(SINC)==0):
    pass

if BT1.value()==0:
    time.sleep(0.004)
    TRIAC.on()
    time.sleep(0.0001)
    TRIAC.off()

if BT2.value()==0:
    time.sleep(0.005)
    TRIAC.on()
    time.sleep(0.0001)
    TRIAC.off()

```

```
if BT3.value()==0:
    time.sleep(0.006)
    TRIAC.on()
    time.sleep(0.0001)
    TRIAC.off()

if BT4.value()==0:
    time.sleep(0.007)
    TRIAC.on()
    time.sleep(0.0001)
    TRIAC.off()
```


Capítulo III

Varredura de leds RGB

1. Led branco

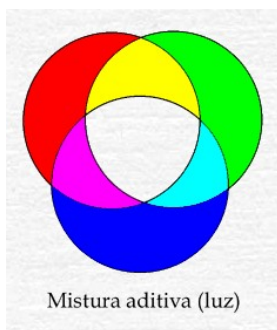
O LED branco foi desenvolvido por engenheiros na GE e é formada a partir da emissão de vários comprimentos de onda. A sua construção é baseada em partículas de fósforo encimado sobre um LED ultravioleta, onde ao receber a incidência de UV faz a conversão desta incidência em luz branca.

2. Led RGB

Cada led é composto de 4 terminais, sendo um catodo comum e o restante o anodo para as cores Vermelha (R), Verde (G) e Azul (B) como mostra a próxima figura.



Para acender a luz vermelha, basta aterrar o catodo e aplicar nível alto no pino R. Mesma ideia é válida para as outras cores, como o azul (B) e o verde (G). Com estas três cores primárias é possível formar outra de acordo com a combinação destas, como mostra a figura a seguir.

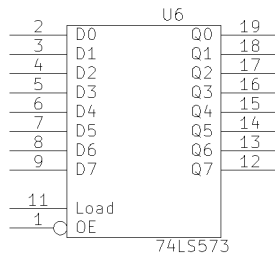


Observe que ao unir as cores vermelha e verde há como resultado o amarelo. Se for o vermelho e azul, o resultado é o violeta. A mistura de todas as cores primárias forma a cor branca. A isso chama-se de mistura aditiva.

3. Esquema elétrico

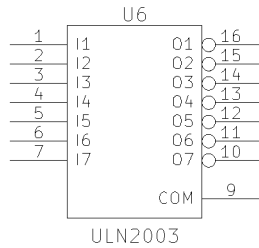
O exemplo proposto consistirá de apresentar um caracter em um painel de leds RGB de 8 linhas e 5 colunas (8x5). Como o número de

pinos de I/O disponíveis no Raspberry é pequeno, faz-se necessário utilizar um latch do tipo 74HC573, para armazenar a palavra a ser carregada nos anodos dos leds R, G e B. A próxima figura mostra a pinagem deste circuito integrado.



A linha OE deve ficar aterrada para deixar as saídas Q0 a Q7 habilitadas. Ao gerar um pulso na linha LOAD, os dados presentes nas entradas D0 a D7 são passadas para as saídas Q0 a Q7, no qual retém esta informação até a próxima atualização. Para este exemplo há 3 latches 74HC573, onde cada um deles fica responsável pelas cores R, G e B, respectivamente.

Cada uma das colunas está interligada, no qual faz-se necessário o uso de um driver de corrente. O CI ULN2003 tem esta incumbência e sua pinagem está apresentada a seguir.

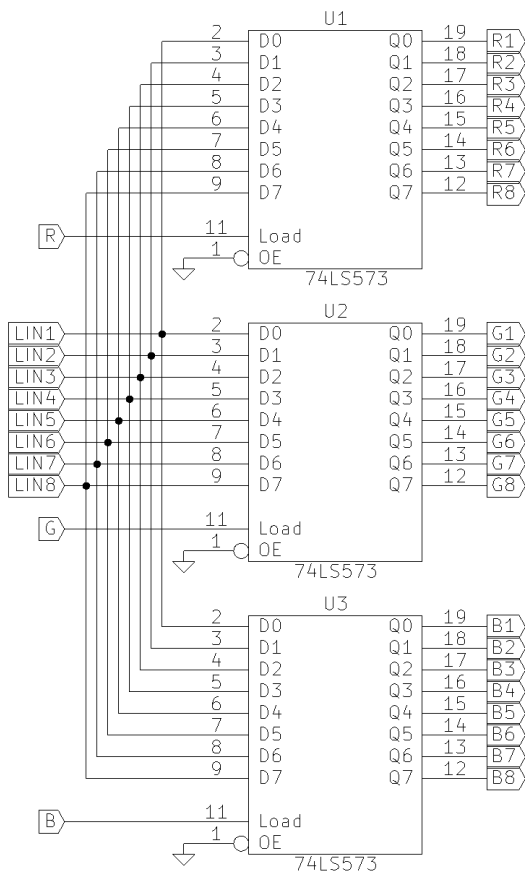


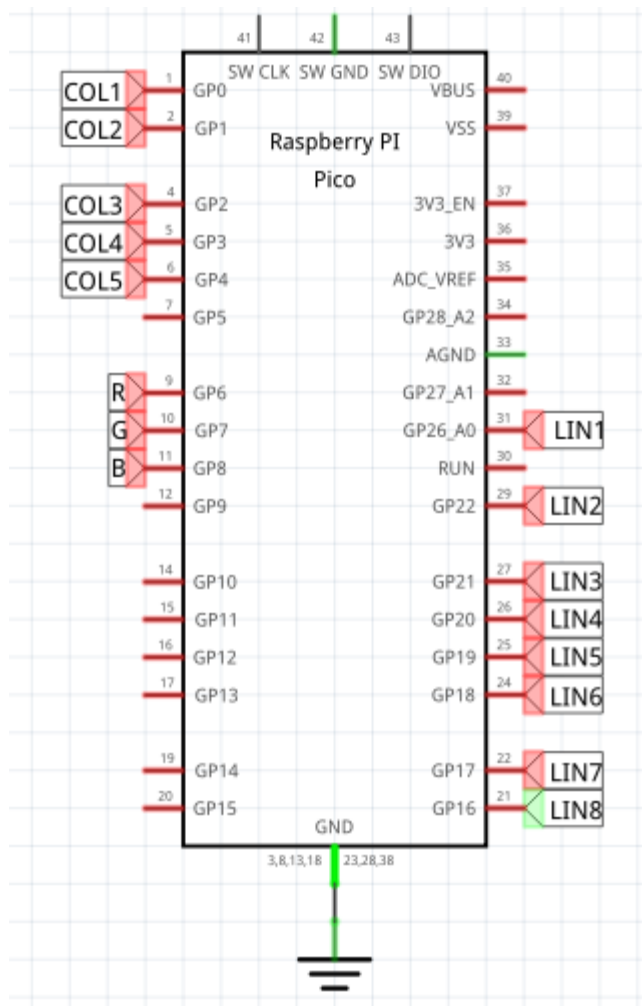
Quando uma das entradas I1 a I7 ficar em nível alto a saída respectiva de Q1 a Q7 fica em nível baixo, suportando uma corrente de até 500 mA. Como são 5 colunas por tratar de uma matriz 8x5, ao todo 5 entradas são utilizadas neste experimento.

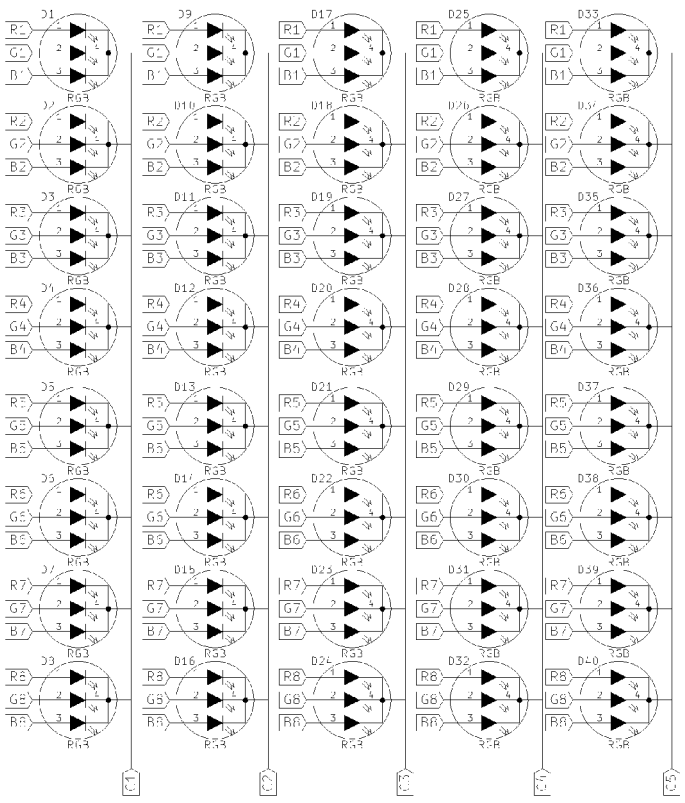
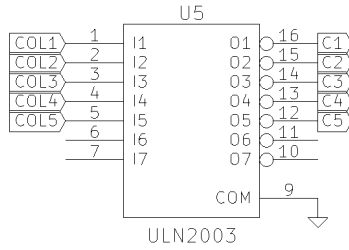
O painel de leds ao ser montado em circuito impresso (PCI) será similar a figura apresentada a seguir.



Apesar de encorajarmos a montagem em PCI, inicialmente este esquema pode ser montado em placa padrão ou protoboard. O esquema elétrico completo do experimento está apresentado a seguir.







4. Testando o barramento de leds

Note que há no esquema 5 colunas cada uma com um conjunto de 8 leds RGB. Todos os leds de uma cor da mesma linha estão ligados no mesmo latch responsáveis por uma das cores do tipo vermelho (R), do tipo azul (B) e do tipo verde (G). Para controlar uma coluna de leds, deve-se primeiramente aterrar a coluna desejada e deixar o restante desabilitado. Em seguida, deve-se controlar algum dos latches 74HC573 de acordo com o conjunto de leds a ficar ligado. Como há ao todo 5 colunas de leds e considerando uma frequência de varredura de 50 Hz, encontra-se o período de 20 ms, onde ao dividir tal valor pelo número de colunas que são 5, constata-se que deve-se deixar cada coluna ligada por 4 ms.

Baseado nestes comentários o próximo programa faz a varredura coluna a coluna no conjunto de leds vermelhos apenas. A funcionalidade deste programa está em confirmar que todos os leds vermelhos do hardware montado estão funcionando. Observe que há uma função chamada `rgb()` que ao ser chamada passa-se como parâmetro o estado das linhas R, G e B. No exemplo, verifique que apenas o parâmetro R está em 0xFF, ou seja, todos os leds ligados enquanto que as linhas de G e B estão em 0x00. Individualmente no

loop do programa cada coluna é ativada por 1 segundo, permitindo visualizar o funcionamento de todos os leds vermelhos do exemplo.

```
#* * * * * * * * * * * * * *
#*           Cerne Tecnologia      *
#*           www.cerne-tec.com.br  *
#*           Projetos em Raspberry - RGB *
#* * * * * * * * * * * * * *
```

```
#* * * * * * * * * * * * * *
#*           Declaração de I/Os    *
#* * * * * * * * * * * * * *
```

```
LIN1P=31
LIN2P=29
LIN3P=27
LIN4P=26
LIN5P=25
LIN6P=24
LIN7P=22
LIN8P=21
RP=9
GP=10
BP=11
COL1P=1
COL2P=2
COL3P=4
COL4P=5
COL5P=6
```

```
#* * * * * * * * * * * * * *
#*           Função de inicialização *
#* * * * * * * * * * * * * *
```

```
from machine import Pin
import time
```

```
LIN1 = Pin(LIN1P, Pin.OUT)
LIN2 = Pin(LIN2P, Pin.OUT)
LIN3 = Pin(LIN3P, Pin.OUT)
LIN4 = Pin(LIN4P, Pin.OUT)
LIN5 = Pin(LIN5P, Pin.OUT)
LIN6 = Pin(LIN6P, Pin.OUT)
LIN7 = Pin(LIN7P, Pin.OUT)
LIN8 = Pin(LIN8P, Pin.OUT)
```

```
R = Pin(RP, Pin.OUT)
G = Pin(GP, Pin.OUT)
B = Pin(BP, Pin.OUT)
```

```
COL1 = Pin(COL1P, Pin.OUT)
COL2 = Pin(COL2P, Pin.OUT)
COL3 = Pin(COL3P, Pin.OUT)
COL4 = Pin(COL4P, Pin.OUT)
COL5 = Pin(COL5P, Pin.OUT)
```

```
##* * * * * * * * * * * * * * *
##*           Função de loop          *
##* * * * * * * * * * * * * * *
```

```
while true:
    rgb(0xFF,0x00,0x00)
```

```
    COL1.on()
    time.sleep(1)
    COL1.off()
```

```
    COL2.on()
    time.sleep(1)
    COL2.off()
```

```
    COL3.on()
    time.sleep(1)
    COL3.off()
```

```
    COL4.on()
    time.sleep(1)
    COL4.off()
```

```

COL5.on()
time.sleep(1)
COL5.off()

#* * * * *
#* Função de carga de latch 74HC573 *
#* * * * *
def rgb(r,g,b)
    LIN1.value(r>>0)
    LIN2.value(r>>0)
    LIN3.value(r>>0)
    LIN4.value(r>>0)
    LIN5.value(r>>0)
    LIN6.value(r>>0)
    LIN7.value(r>>0)
    LIN8.value(r>>0)
    R.on()
    R.off()

    LIN1.value(r>>0)
    LIN2.value(r>>0)
    LIN3.value(r>>0)
    LIN4.value(r>>0)
    LIN5.value(r>>0)
    LIN6.value(r>>0)
    LIN7.value(r>>0)
    LIN8.value(r>>0)
    G.on()
    G.off()

    LIN1.value(r>>0)
    LIN2.value(r>>0)
    LIN3.value(r>>0)
    LIN4.value(r>>0)
    LIN5.value(r>>0)
    LIN6.value(r>>0)
    LIN7.value(r>>0)
    LIN8.value(r>>0)
    B.on()

```

```
B.off()
```

O próximo exemplo parte do mesmo princípio do que fora visto neste exemplo, onde apenas os leds verdes acendem.

```
#* * * * * * * * * * * * * * *
#*          Cerne Tecnologia          *
#*          www.cerne-tec.com.br      *
#*          Projetos em Raspberry -  *
#* * * * * * * * * * * * * * *
```

```
#* * * * * * * * * * * * * * *
#*          Declaração de I/Os       *
#* * * * * * * * * * * * * * *
```

```
LIN1P=31
LIN2P=29
LIN3P=27
LIN4P=26
LIN5P=25
LIN6P=24
LIN7P=22
LIN8P=21
RP=9
GP=10
BP=11
COL1P=1
COL2P=2
COL3P=4
COL4P=5
COL5P=6
```

```
#* * * * * * * * * * * * * * *
#*          Função de inicialização  *
#* * * * * * * * * * * * * * *
```

```
from machine import Pin
import time
```