

# Controle de Fita LED RGB utilizando PIC18F877A

Vitor Bruno de O. Barth

IFMT – DAI

vbob@vbob.com.br

**Resumo** – O número de residências com iluminação LED aumentou nos últimos anos devido a comodidade, durabilidade e baixo custo. Este trabalho apresenta um modo de construir um controlador para uma fita LED RGB capaz de exibir até 16 milhões de tonalidades de cor.

## I. INTRODUÇÃO

Em 2017, 22% dos sistemas de iluminação tradicionais, baseados em lâmpadas incandescentes e fluorescentes já haviam sido substituídos por lâmpadas do tipo LED (*light emitter diode*), e espera-se que este número seja de 63% em 5 anos. [1]

Isso acontece pois, além de serem visualmente mais agradáveis que as lâmpadas tradicionais por oferecerem meios de regular intensidade e cor, LEDs convertem 95% da energia em forma de luz, iluminando melhor enquanto consomem menos energia e emitem menos calor. [2]

Uma das formas de encapsulamento de LEDs é na forma de fita, apresentada na Figura 01: em meio a um pedaço de plástico chato e estreito (1mm x 10mm) são colocadas LEDs em uma densidade de 30 a 60 por metro, sendo adequadas para locais com limitação de espaço. Além disso, apenas um metro de fita LED é suficiente para substituir uma lâmpada fluorescente de 20W, com  $\frac{1}{4}$  do consumo de energia, para densidades de 60LEDs/m. [3]



Figura 1 - Fita LED. Fonte: Seed Studio.

Outra característica marcante em fitas LEDs é a possibilidade de mudanças de cor. Neste tipo de fitas, cada ponto de luz é composto por três LEDs individuais, cada uma de uma cor do espectro RGB (*Red-Green-Blue*), e a combinação de intensidades destas permite que sejam formadas quase 16 milhões e 500 mil cores diferentes. [4] Para que possa ser feita a seleção de cores, é necessário um equipamento controlador, que regula a tensão em cada uma das LEDs conforme a escolha do usuário. O modelo padrão de um sistema de iluminação por fita LED RGB pode ser visto na Figura 02.

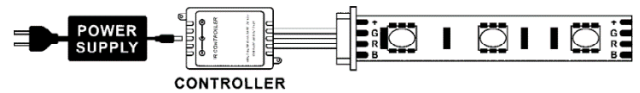


Figura 2 - Sistema de Iluminação por fita LED. Fonte: Lavolta.

Apesar de possuírem custo baixo de produção, a instalação de sistemas de iluminação por LED RGB ainda é caro, pois os sistemas controladores são, em maioria, proprietários e possuem pouco suporte fora das empresas fabricantes.

Observando o crescimento do mercado de LEDs e o baixo número de soluções *open-source*, este trabalho busca trazer como resultado um *software* para um microcontrolador de baixo custo capaz de realizar o controle de uma Fita LED RGB.

O restante do trabalho é dividido como segue: a seção 2 descreve os equipamentos utilizados, enquanto a seção 3 descreve a solução em *software*. A seção 4 mostra como foi feita a montagem do *hardware* própria ao programa feito, e na seção 5 são apresentados as dificuldades e o resultado do trabalho.

## II. EQUIPAMENTOS

Todo o sistema foi planejado de modo a ser possível controlar a cor um segmento de 50cm de uma fita LED RGB 5050 de 12V por meio do microcontrolador PIC16F877A.

Sabendo que a tensão máxima de alimentação e saída de um PIC 16F877A é 5V, é necessário utilizar transistores para realizar o controle de tensão. Neste caso, por ser um segmento pequeno que consome 6mA em pico (valor obtido por experimento), utilizou-se o transistor BJT BC547 [5], contudo, para fitas mais longas com consumo maior que 100mA é necessário utilizar FETs para que a resposta de 16us seja mantida mesmo com correntes maiores.

Para interface com o usuário, optou-se por utilizar um *display* 16x2 HD44780 e um teclado matricial 4x4. Para diminuir a complexidade do circuito, foi adquirido um *display* acoplado a um expensor de *bits* PCF8574, que permitiu a comunicação no formato I<sup>2</sup>C.

O custo aproximado e a lista de componentes utilizada são apresentados no Quadro 01. Não foi avaliado o custo de montagem do circuito e itens de suporte, como placas, matrizes de contato, solda, gravadora para o microcontrolador, fios, etc.

Quadro 1 - Material Utilizado

Equipamento	Qtd.	Valor Un. (R\$)	Total (R\$)
PIC 16F877A	1	15,00	15,00
Display 16x2	1	20,00	20,00
Fita LED RGB	0,5m	6,00	3,00
Resistor 220Ω	3	0,05	0,15
Transistor BC547	3	0,10	0,30
Teclado 4x4	1	12,00	12,00
<b>Total (R\$)</b>			<b>50,45</b>

### III. SOFTWARE

Inúmeros desafios permearam o desenvolvimento de um *software* que antedesse aos requisitos do projeto:

- Para que a gama de cores de 16.581.375 tons fosse mantida, era necessário que cada uma das LEDs do espectro RGB possuísse 255 níveis de intensidade (além estado desligado) [6];
- Taxas de atualização de cor menores que 60Hz são perceptíveis ao ser humano. Deste modo, pulsos de largura maior que 16ms causariam desconforto devido ao efeito de *flickering* (do inglês, ‘piscadas’, ‘tremulação’) [7];
- São necessários três sinais para controlar os tons de cor, cada um deles responsável por um dos tons RGB. Entretanto, sinais variáveis só podem ser emitidos pelo PIC16F877A por meio de PWM, e nele só existem dois canais de PWM nativos [8];
- Não foi encontrada bibliotecas prontas que realizam comunicação I<sup>2</sup>C entre o PIC 16F877A e o expensor de *bits* PCF8574.

Para realizar a comunicação entre o PIC 16F877A e o *display* 16x2 por meio de I<sup>2</sup>C, as funções essenciais de escrita foram transcritas utilizando funções específicas do PIC, baseadas na versão da biblioteca para *Arduino* [9].

Para garantir os 255 níveis de cor em cada um dos espectros RGB, foram construídos três sinais de PWM sincronizados, com largura 16ms, garantindo uma taxa de atualização maior que 60Hz.

A sincronia destes sinais é mantida por meio de interrupções temporais, que ocorrem a cada 63us. Estas interrupções servem para alternar o estado dos LEDs conforme necessidade e manter a contagem dos pulsos. O código executado a cada interrupção consiste em oito operações bit-a-bit, um incremento e uma mudança de estados das portas D, e são passíveis de serem realizadas no tempo previsto de 63us, em um microcontrolador operando a 20MHz. O código da interrupção de tempo é apresentado na Figura 3.

```

242 uint8_t counter = 0;
243
244 #INT_TIMER2
245 void TIMER2_isr(void)
246 {
247     output_d(0b00000000 |
248             ((blue_level>counter)<<2) |
249             ((red_level >counter)<<1) |
250             (green_level>counter));
251     counter++;
252 }

```

Figura 3 - Interrupção por tempo.

A leitura do teclado é controlada através de um loop que é executado na função principal do PIC. Nesta função, é verificado o curto circuito ocasionado pelo pressionamento de um dos botões.

A mudança de cor é feita por meio da digitação do código RGB da cor de interesse. Conforme o usuário digita, o valor é apresentado no display 16x2. O valor pode ser incrementado ou decrementado por meio das teclas # e \* do teclado.

Há a verificação de valores: caso o usuário digite um valor fora do intervalo de 0 a 255, o valor é negado (entretanto, não é exibido erro).

Como não foram implementadas funções de leitura da memória do *display*, por conta da complexidade, o valor das variáveis digitadas é mantida através do salvamento dos valores digitados pelo usuário, tecla a tecla.

### IV. CONSTRUÇÃO

A fita LED RGB utilizada possui alimentação de 12V, entretanto a tensão máxima de operação do PIC 16F877A é de 7.5V, portanto deve ser tomado muito cuidado durante a prototipagem e produção do circuito para as faixas de tensão utilizadas estejam isoladas entre si.

Para garantir este isolamento, optou-se pelo uso de transistores BJT BC547, que possuem tempo de resposta suficientes ( $T_{ON}=35ns$ ;  $T_{OFF}=300ns$ ) e operam em  $V_{CBO}$  de até 50V [5] [10].

Para que não haja flutuações de sinal de leitura, optou-se por utilizar o teclado nas portas B, que possui resistores de *pull-up* internos.

Foi montado em uma *protoboard* o circuito apresentado nas Figuras 4 a 8.

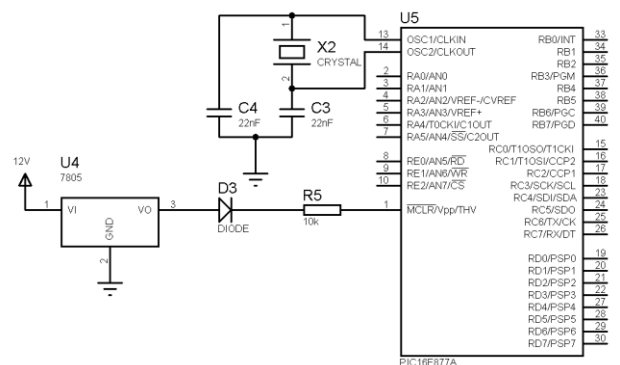


Figura 4 - Circuito de alimentação de frequência.

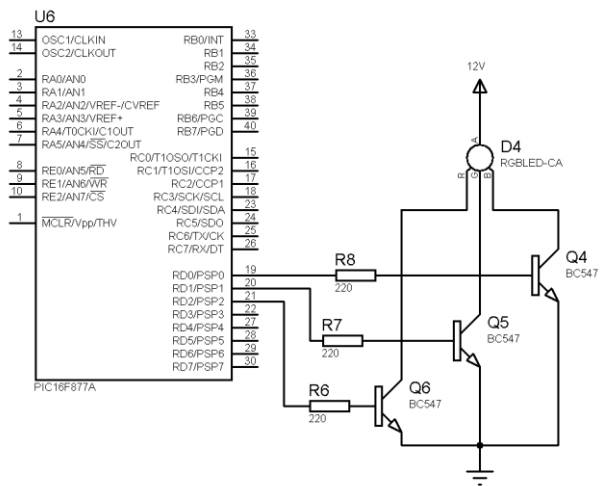


Figura 5 - Circuito de controle da fita LED RGB.

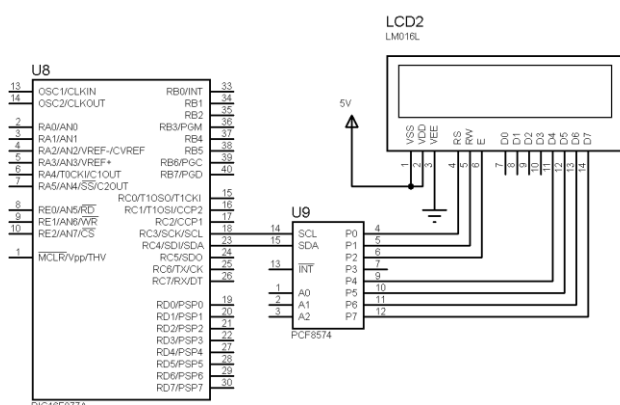


Figura 6 - Circuito de comunicação com display 16x2.

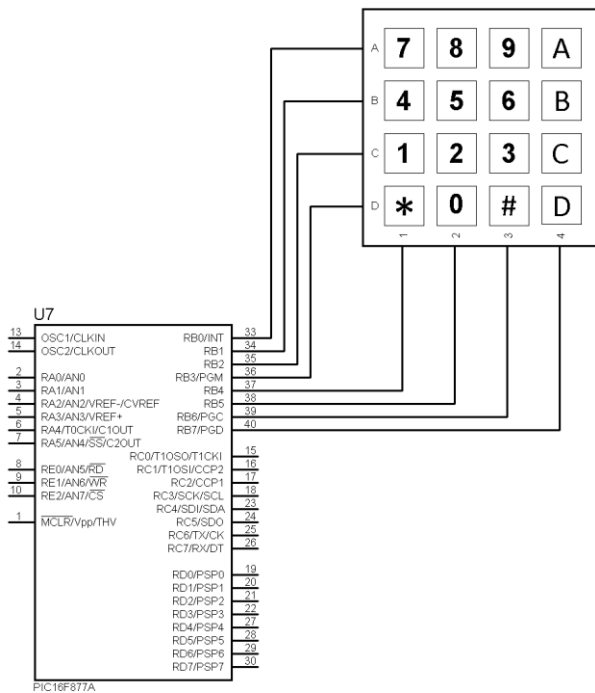


Figura 7 - Circuito de interface com teclado 4x4.



Figura 8 - Mensagem exibida ao usuário.

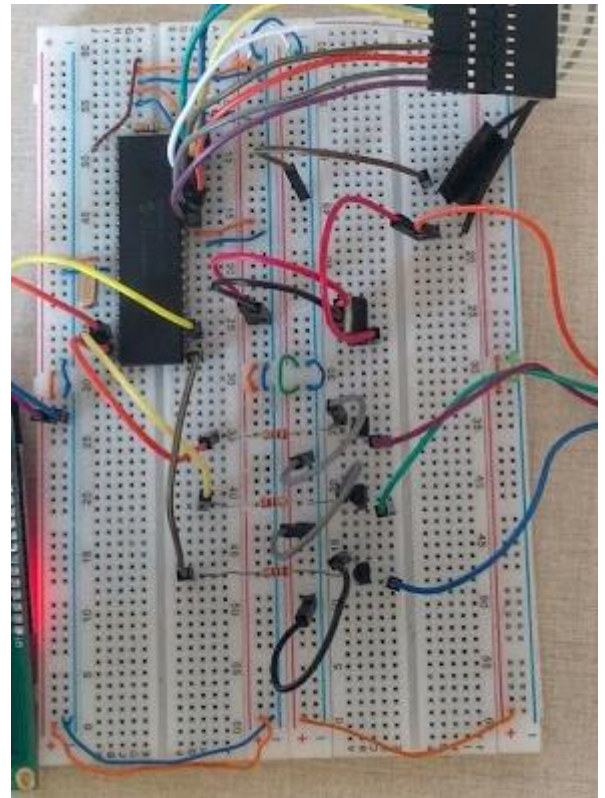


Figura 9 - Circuito completo físico.



Figura 10 - Intensidade de Luz #0000FF vs. #000006

## V. CONSIDERAÇÕES FINAIS

Ainda que o resultado obtido seja simples, a complexidade envolvida no desenvolvimento deste trabalho é alta: muitos conceitos não triviais foram aplicados, e foi necessário desenvolver soluções de modo a unir todas as peças em um único objetivo.

Além disso, o funcionamento do conjunto deve ser em tempo real, para que o usuário não perceba travamentos, latência ou até mesmo ocasionar efeitos nocivos à saúde, como oscilações na luz emitida. Isto é um tanto trabalhoso, dada a natureza síncrona de execução do PIC 18F877A e a necessidade de realizar operações em frequências de 15.5 kHz.

Apesar disso, o resultado obtido foi um tanto quanto satisfatório. Apesar das dificuldades e da complexidade, o produto é estável e está de acordo com as especificações do projeto.

## VI. REFERÊNCIAS

- [1] J. Wu, “Global LED Lighting Market Trend,” 2017. [Online]. Available: <https://goo.gl/yBNqmB>.
- [2] InStyle LED, “Energy Efficiency – how economical is LED strip lighting?,” 2013. [Online]. Available: <https://www.instyleled.co.uk/support/is-led-tape-energy-efficient/>.
- [3] NTE Electronics Inc., “Single Color 3528 Size Flexible LED Strips Datasheet,” pp. 3–4.
- [4] Velleman, “LED Strips Choose Manual,” pp. 1–6, 2010.
- [5] Fairchild, “BC546/547/548/549/550 Datasheet,” pp. 2–5, 2002.
- [6] Wikipédia, “RGB.” [Online]. Available: <https://pt.wikipedia.org/wiki/RGB>.
- [7] Wikipédia, “Flicker (screen).” [Online]. Available: [https://en.wikipedia.org/wiki/Flicker\\_\(screen\)](https://en.wikipedia.org/wiki/Flicker_(screen)).
- [8] Microchip, “Pic 16F87XA,” *Datasheet*, vol. 25, no. 6, p. 5, 1997.
- [9] fdebrabander, “Arduino-LiquidCrystal-I2C-library,” 2011. [Online]. Available: <https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library>.
- [10] Электрика это просто, “Popular Transistors Specifications - BC547.” [Online]. Available: <http://elektrikaetoprosto.ru/ebc547.html>.

## ANEXO I - SOFTWARE DO MICROCONTROLADOR

Disponível em: <<https://goo.gl/4iebE1h>>