

CONTROLE AUTOMÁTICO DE ILUMINAÇÃO COM MSP430

Jennifer Gladys P Cavalcante

Programa de Engenharia Eletrônica
Faculdade Gama - Universidade de Brasília
St. Leste Projeção A - Gama Leste, Brasília-
DF, 72444-240
email: jennifercavalcante.unb@gmail.com

Renato da Costa Motta Jr

Programa de Engenharia Eletrônica
Faculdade Gama - Universidade de Brasília
St. Leste Projeção A - Gama Leste, Brasília-
DF, 72444-240
email: renato.motta.jr@gmail.com

RESUMO

Medidor de luz ambiente, onde será usado um MSP430 que será responsável por monitorar a luz ambiente. Que irá detectar quando o ambiente possui luminosidade acima de determinado limiar, ele apaga a iluminação do cômodo.

1. INTRODUÇÃO

O aumento do consumo de energia elétrica, em razão do consumismo acelerado, tem provocado a construção de mais usinas hidrelétricas. Elas causam enormes impactos ambientais, em virtude da quantidade de água represada a fim de mover as turbinas na produção de energia elétrica. Uma alternativa seria a construção de usinas nucleares, mas esse tipo de usina produz um lixo radioativo que deve ser armazenado em locais remotos, além de ser muito perigosa, podendo causar catástrofes de grandes proporções^[1]. Quando se observa o consumo residencial, a iluminação representa de 15% a 25%^[2] da potência consumida. Parte desse consumo provém do desperdício e mau uso da iluminação. Existem diversos sistemas que realizam o desligamento das lâmpadas, esses porém, geralmente são baseados em movimentos, o que não reflete precisamente a utilização do ambiente, ocorrendo falsos positivos ou negativos de presença. Sendo cada vez mais frequente a necessidade das pessoas por conforto e economia, visto que as chamadas iluminação inteligente custam mais não sendo acessível e não tendo uma economia em seu produto^[4]. O MSP430, onde o protótipo do controle de luminosidade é desenvolvido, é conhecido por seu baixo consumo elétrico^[5].

2. DESENVOLVIMENTO

Com a evolução da chamada iluminação inteligente, onde no Brasil foi visto pela primeira vez em 2010, o modelo foi chamado pelo seu fabricante PIAL LeGrand de Nereya, que consistia em um dispositivo que basicamente acendia, apagava e variava a iluminação por meio do controle remoto^[6].

O projeto com o MSP430G2 consiste em utilizar o microcontrolador, visto que o mesmo tem um baixo consumo de energia, onde foi desenvolvido um sistema de controle de iluminação elétrica. A princípio no protótipo, que foi pensado para um escritório, não é necessário acionar nenhum interruptor. O LDR que será acoplado a fileira de LED que são usados na iluminação, é que o mesmo modifique a iluminação. Em um ambiente claro, propício para o trabalho, não há acionamento da iluminação local. Assim ocorrendo variações de luminosidade, não sendo apenas aceso, mediano e apagado. O LDR tem a funcionalidade de fazer o nivelamento. O código utilizado no software livre Energia, consiste apenas no controle do LDR, nos seus níveis.

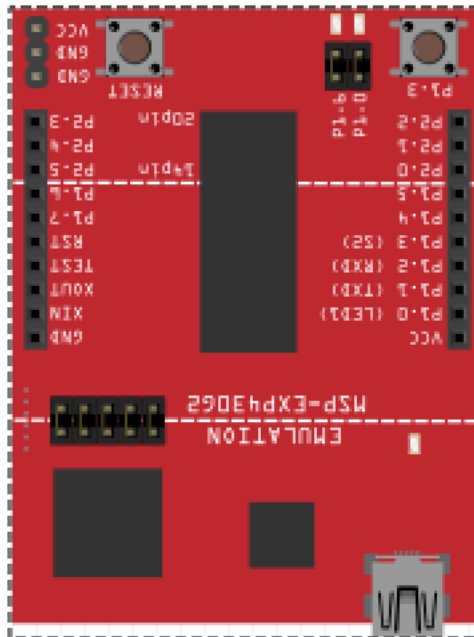
3. REQUISITOS

- Conhecimentos básicos de circuitos eletrônicos;
- Linguagem de programação em C ou Assembly;
- Conhecimento em MSP430;
- Sensores e Relés;
- Amplificadores Operacionais;
- LDR (light dependent resistor).
- Computador

4. DESCRIÇÃO DO HARDWARE

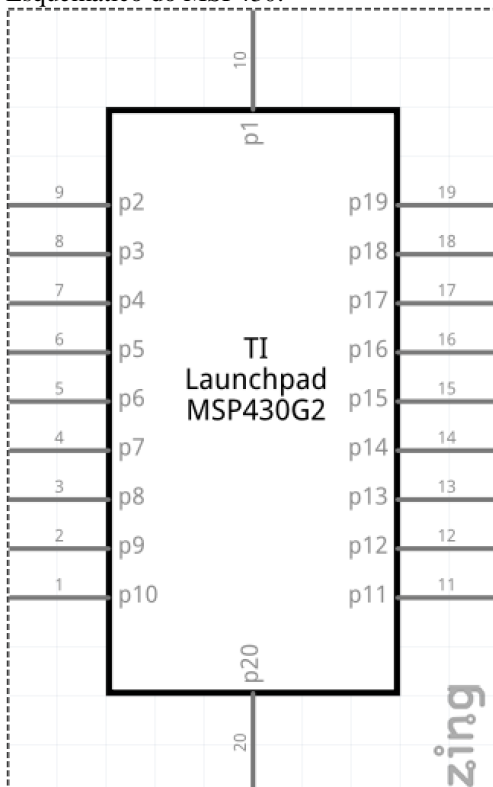
O TI Launchpad MSP430G2 é uma panela de desenvolvimento de microprocessadores da Texas Instruments. É muito parecido com os chips Atmel no Arduino. O MSP430 não é muito popular por conta da sua complexidade de programar. Para o desenvolvimento do projeto, só foi necessário as portas e a lógica de programação para a funcionalidade do LDR com as fitas de LED.

O MSP430G2:



MSP430 1

Esquemático do MSP430:



5. DESCRIÇÃO DO SOFTWARE

- TI LaunchPad
- Protoboard
- Jumpers
- Sensor LDR
- Resistores de 100KΩ
- Arduino uno
- 50 LEDs

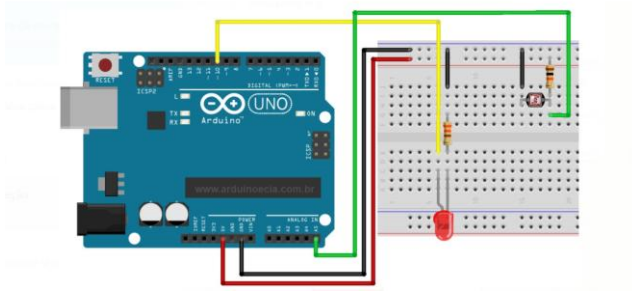
Para finalidade de implementação inicial faremos uso das bibliotecas do Energia, de domínio público. No qual o autor faz uso de um único LED como fonte de luz e utiliza um sensor LDR para medir a intensidade da luz ambiente. Sendo o código para tal apresentado a seguir:

```
1 void setup()
2 {
3   Serial.begin(9600);
4   pinMode(19, OUTPUT);
5 }
6
7 void loop()
8 {
9   int lightLevel = analogRead(2);
10  delay(1);
11  Serial.println(lightLevel);
12
13  lightLevel = map(lightLevel, 0, 4096, 0, 255);
14  lightLevel = constrain(lightLevel, 0, 255);
15  analogWrite(19, lightLevel);
16 }
```

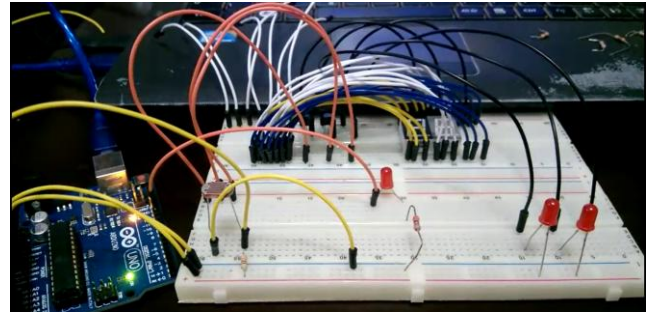
Para nossa aplicação, porém, a reatividade do LED é inversa, sendo que este aumenta sua intensidade quando a pouca luz, funcionando como um regulador e não apenas um indicador, ao contrário do código apresentado. Além desta modificação, se faz necessária a utilização de uma série de foto dispositivos. Assim sendo, o uso de uma matriz de saída se mostra adequado a princípio, ao simular assim a rede de iluminação de um grande ambiente. Nesta fase, entretanto, serão apenas utilizados 4 LEDs individuais, formando um retângulo.

Para uma nova implementação, visto que houveram problemas com a placa de MSP430, foi testado um novo

código em arduino.



Circuito Arduino+LDR

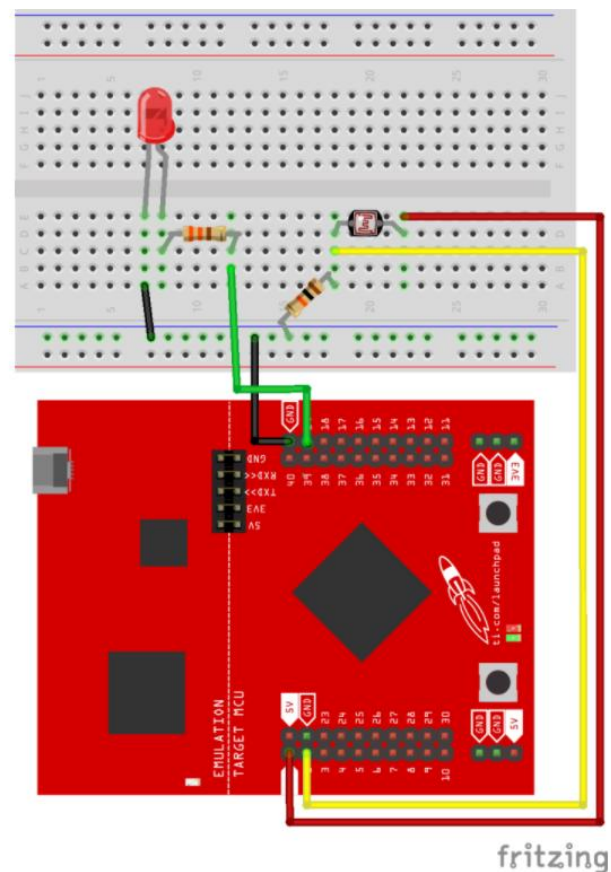


Teste LDR+Arduino

O código para implementação:

```
1 // Projeto : Controle de luminosidade de led com LDR
2
3 int pinLED = 10;      //Pino ligado ao anodo do led
4 int pinLDR = 5;       //Pino ligado ao LDR
5 int valorLDR = 0;     //Armazena valor lido do LDR, entre 0 e 1023
6 int luminosidade = 0; //Valor de luminosidade do led
7 int nivel = 0;        //Nível de intensidade de iluminação
8
9 void setup()
10 {
11   Serial.begin(9600); //Inicializa a serial
12   pinMode(pinLED, OUTPUT); //Define o pino do led como saída
13   pinMode(pinLDR, INPUT); //Define o pino do LDR como entrada
14 }
15
16 void loop()
17 {
18   valorLDR = analogRead(pinLDR); // Le o valor analogico do LDR de 0 a 1024
19
20   if (valorLDR >= 870){ // Luz emitida maior ou igual a 85%
21     luminosidade = 64;
22     nivel = 85;
23   } else if (valorLDR < 870 && valorLDR >= 768){ // Luz emitida entre 85% e 75%
24     luminosidade = 128;
25     nivel = 75;
26   } else if (valorLDR < 768 && valorLDR >= 512){ // Luz emitida entre 75% e 50%
27     luminosidade = 192;
28     nivel = 50;
29   } else if (valorLDR < 512){ // Luz emitida menor que 50%
30     luminosidade = 256;
31     nivel = 25;
32   }
33
34   Serial.print("Valor lido do LDR : ");
35   Serial.print(valorLDR);
36   Serial.print(" = Nivel Luminosidade : ");
37   Serial.println(nivel);
38   Serial.print("%");
39
40   // Acende o led com luminosidade variável
41   analogWrite(pinLED, luminosidade);
42 }
```

A partir dos testes em arduino, foi feita a alteração para que o mesmo fosse implementado em MSP430 através do energia. Sendo a montagem para um único LED em caráter de teste.



Circuito+MSP430

Código para MSP430:

```

1 #include <msp430g2553.h>
2
3 void atraso() {
4
5     volatile unsigned i;
6
7     for(i=0;i<5000;i++);
8
9 }
10
11
12
13 void main (void)
14 {
15 {
16
17     WDTCTL = WDTPW | WDTHOLD;           //Desliga Watchdog
18
19     BCSCTL1 = CALBC1_1MHZ;
20     DCOCTL = CALDCO_1MHZ;               //Configura Clocks
21
22     P1DIR |= BIT6;
23     P1SEL |= BIT6;
24
25     TACCR0 = 1026 - 1;
26     TACCR1 = 0;
27     TACTL1 = OUTMOD_7;
28     TACTL = TASSEL_2 + ID_2 + MC_1 + TACLR + TAIE; //divide o clock por 4, modo up, Timer_A clear,
29                                                     //habilita a interrupcao por timer
30
31     ADC10CTL0 = SREF_0 + ADC10SST_1 + ADC10ON + ADC10IE; //Vcc de ref como o Vcc da map
32                                                         //Vas da ref como o ground da map
33                                                         //habilita o ADC10ON e interrupcao por ADC10
34                                                         //P1.0 como o pino de referencia
35     ADC10CTL1 = INCH_0 + RES_0 + ADC10DIV_0 + ADC10SSSEL_0 + CONSEQ_0; //habilita no bit ADC10SC o valor da tensão da porta
36     ADC10A09 |= BIT0;
37
38     for (;;)
39     {
40         _BIS_SR(LPM0_bits+GIE);
41     }
42
43 }
44
45 #pragma vector = TIMER0_A1_VECTOR
46
47 __interrupt void TA0_ISR(void)
48 {
49
50     atraso();
51     TACTL &= ~TAIFG;
52     ADC10CTL0 |= ENC + ADC10SC;
53 }

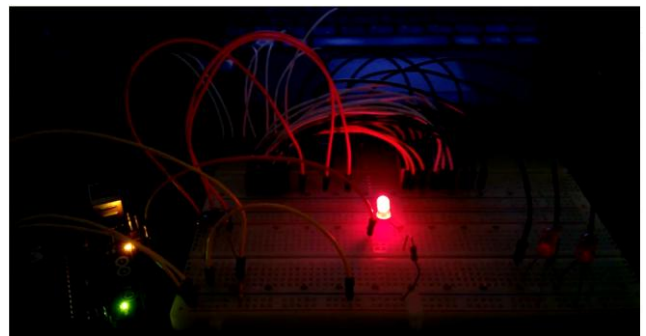
```

```

45 #pragma vector = TIMER0_A1_VECTOR
46
47 __interrupt void TA0_ISR(void)
48 {
49
50     atraso();
51     TACTL &= ~TAIFG;
52     ADC10CTL0 |= ENC + ADC10SC;
53 }
54
55 #pragma vector = ADC10_VECTOR
56
57 __interrupt void ADC_10_ISR(void) {
58
59     volatile float algo = 0;
60
61     algo = ADC10MEM;
62
63     TACCR1 = (1025/(algo + 1)); // Máximo valor do brilho que é possível chegar
64
65     ADC10CTL0 &= ~(ENC) & ~(ADC10SC);
66     ADC10CTL0 &= ~ADC10IFG; //apaga a flag de interrupcao
67 }
68 }
69

```

O código descrito acima foi o utilizado para o funcionamento dos LEDs já no ambiente. Funcionando para os 50 LEDs com o LDR.



Teste LDR

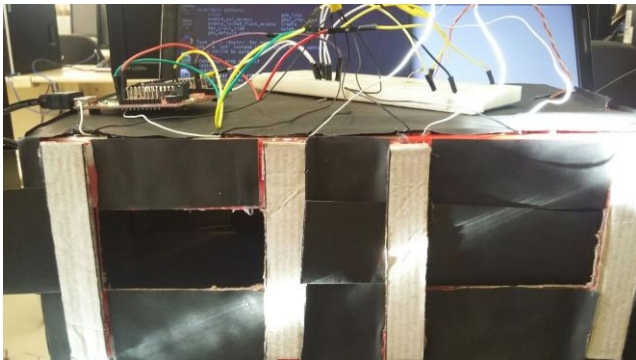
Os testes com LDR foram feitos respondendo a um estímulo de uma lâmpada convencional de uma casa. Mostrando que o código estava de acordo conforme as intensidades de luz que o LDR recebia. Estando completamente escuro o LED acendia e com variações de luz, mudava a intensidade com que o LED ficava aceso. Mostrando que para um ambiente com pouco LUZ, o LED ficava em modo alto, com uma luz ambiente adequada, o LED completamente apagado, e um intermediário, para uma iluminação adequada para que a visão não fosse forçada.



Teste LDR no ambiente simulado

Quando acoplados a parte superior da caixa, que estava sendo usada para simular um escritório, que foi um ambiente previamente pensado para o projeto, fez-se uso de 50 LEDs soldados em cinco fitas, 10 LEDs em cada fita e a um resistor de 100k Ω em sua ponta, para que não houvesse risco de queima de algum LED.

Na caixa haviam buracos onde simulavam janelas e uma porta, para simulasse a entrada de luz e o LDR trabalhando quanto a economia. Avaliando se era necessário que a luz fosse acesa ou não, e um nível intermediário.



Escritório Simulado para o LDR com MSP430

6. RESULTADOS

Para validação do experimento, primeiro foram feitos testes para um único Led e um LDR em um ambiente com e sem iluminação, para que os níveis fossem definidos em baixo, intermediário e alto. Também de início, não foi usado o MSP430, os testes foram feitos usando Arduino Uno, visto que de imediato não havia tanta mudança na linguagem de um para o outro.

Depois foram acrescentados mais LED, ao todo 50 LEDs em 5 fileiras e LDRs e incluído uma caixa para simular o

ambiente do escritório proposto para o protótipo. Houveram dificuldades na utilização da placa, da passagem do arduino para o MSP430, do controle do LDR para que obedecesse o controle necessário para iluminação. Fazendo que o LED começasse apagado e não acesso em contato com LDR. Também houve dificuldade para que cada fileira tivesse uma porta no MSP430, para que não houvesse a tensão dividida, visto que o projeto conta com 50 LEDs acesos. Conseguiu-se ao final ligar os 50 LEDs no ambiente simulado, que foram acoplados á parte superior da caixa, simulando o teto de um escritório, porém não estavam independentes, uma fileira para cada porta.

Há necessidades de melhorias quanto as portas, para cada fileira de LED e necessário para que o projeto possa se desenvolver futuramente.

7. CONCLUSÃO

O sistema atendeu ao escopo proposto visto que ao final obteve-se um ambiente controlado por MSP430 com seu baixo consumo energia onde detectava a iluminação ambiente fazendo controle da iluminação.

Os gastos com esse projeto foram os da MSP430 usado em sala, protoboard, fios, LEDs e LDR, não ultrapassando o valor de R\$100,00. A parte mais cara a princípio é o microcontrolador, mas com a economia proposta, o projeto se paga ao longo do período.

8. REFERÊNCIAS

- [1] Agência Nacional de Energia Elétrica- Disponível em: <<http://www.aneel.gov.br/regulacao-do-setor-eletrico>> acesso em: 02 de abril de 2017.
- [2] CEB - disponível em: <<http://www.ceb/index.php/uso-racional-de-energia/371-uso-racional-de-energia>> acesso em: 02 de abril de 2017.
- [3] OPT3001 Ambient Light Sensor (ALS) - <<http://www.ti.com/lit/ds/symlink/opt3001.pdf>>.
- [4] Iluminação inteligente - <<http://www.ecologiaurbana.com.br/conscientizacao/iluminacao-inteligente-saiba-como-funciona/>>
- [5] MSP430G2X53 - <<http://www.ti.com/lit/ds/symlink/msp430g2553.pdf>>
- [6] Meio Ambiente - Iluminação Inteligente - <<http://meioambiente.culturamix.com/noticias/iluminacao-inteligente-o-que-e-e-como-funciona>>