

# Armazenador e emissor de código infravermelho

Gabriel da Silva Soares

Faculdade do Gama

Universidade de Brasília

Gama - DF, Brasil

Ricardo Vieira Borges

Faculdade do Gama

Universidade de Brasília

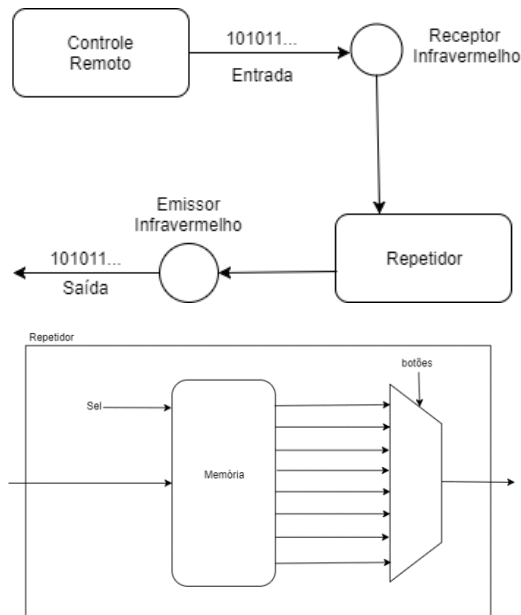
Gama - DF, Brasil

## I. RESUMO

O Projeto proposto consiste em um armazenador de sinal infravermelho através de um receptor. O projeto também será capaz de emitir esse mesmo sinal armazenado em uma memória do microcontrolador.

## II. INTRODUÇÃO

Muitas vezes necessita-se de vários controles (infravermelho) para diversos eletrodomésticos em uma casa. Nem sempre é cômodo manter todos os controles à mão para usufruir de seus dispositivos. Além disso, alguns equipamentos possuem funções exclusivas somente em seu controle remoto.



## III. DESENVOLVIMENTO

Desta forma, pensou-se em uma maneira de diminuir o número de controles ou de garantir

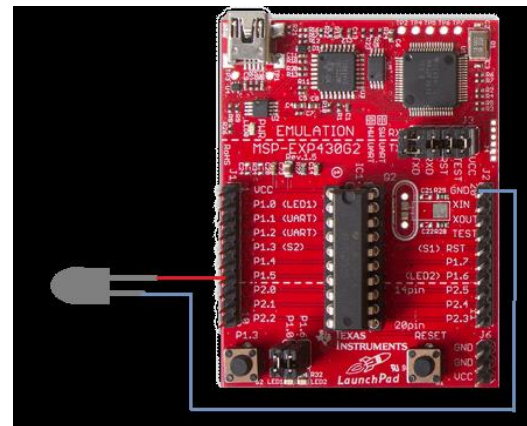
que determinado equipamento não fique inutilizável por causa de um controle. O grupo mantém como objetivo a criação de um controle remoto que não dependa de cadastro de códigos, bastando apontar um controle para ele e associando um botão a essa sequência enviada pelo controle a ser copiado.

Inicialmente foi feito um protótipo funcional do projeto para mostrar a comunicação dos sensores que serão utilizados.

Os materiais utilizados no protótipo foram:

- Duas placas de desenvolvimento com microcontrolador MSP-430;
- Resistor 100Ω;
- LED infravermelho;
- Receptor infravermelho codificado;

Foi montado um pequeno esquema em uma protoboard com a saída P1.5 de um MSP alimentando o LED infravermelho, o outro MSP alimentou o receptor e recebeu como entrada seu pino de dados na porta P1.3



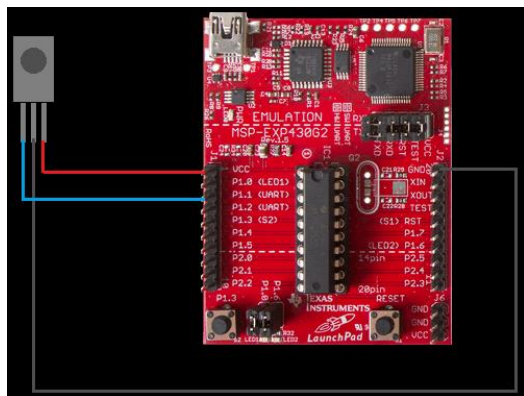
O programa implementado no MSP que controla o LED infravermelho utiliza a biblioteca

msp430g2553.h de registradores e intrinsics.h para interrupções.

Primeiramente é desligado o Watchdog timer, após isso definimos os registradores que ativam os clock's de 1MHz, o pino desejado é definido como saída (P1.5). Registradores de comparação são inicializados para definir uma frequência específica para a saída P1.5.

É utilizada uma interrupção que inverte a saída toda vez que o ciclo detectado pelo comparador se completa.

Para o circuito receptor, foi feito o seguinte circuito:



A única entrada estabelecida foi a porta P1.1 para ler o dado captado pelo receptor. A saída do circuito foi a porta P1.0 que é acoplada com o LED vermelho da placa, dessa forma, quando o sensor detectava algum sinal IR, o LED vermelho era aceso.

#### IV. RESULTADOS

Os circuitos propostos foram testados, começando pelo LED infravermelho, a funcionalidade foi conferida com uma câmera de celular, o LED pisca perfeitamente com a frequência definida. Após isso o funcionamento do receptor foi conferido com um controle remoto infravermelho de um equipamento qualquer.

O LED infravermelho foi apontado para o receptor e seus pulsos foram detectados pelo mesmo, porém com uma intensidade muito baixa (LED que indicava o recebimento do sinal acende com baixa intensidade). Após algumas conferências no circuito e alguns testes para averiguar a causa do problema, foi constatado

que o LED infravermelho recebia pouca potência, o que resultava em uma emissão fraca de luz, refletindo em uma percepção também fraca pelo receptor. Este problema pode ser resolvido com um circuito amplificador de potência para o LED infravermelho.

#### V. REVISÃO BIBLIOGRÁFICA

Funcionamento do controle remoto infravermelho. Disponível em: <http://www.hpspin.com.br/site1/circuitos/ctremoto/>. Acesso em 04/09/2017.

## ANEXO I

Programa para detectar sinal infravermelho.

//PISCA O LED1 QUANDO UM SINAL DE IR É DETECTADO.

```
#include <msp430.h>
#define LED BIT0
#define RECEPT BIT1

int main(void)
{
    WDTCTL = WDTPW | WDTHOLD; // stop watchdog timer

    P1DIR = ~RECEPT;
    P1DIR = LED;
    P1REN = RECEPT;
    P1OUT = RECEPT;

    while(1){

        if ((RECEPT & P1IN) == 0){
            P1OUT = LED;
        }else{
            P1OUT = ~LED;
        }

    }

    return 0;
}
```

Programa para piscar LED infravermelho.

//DEFINE A FREQUÊNCIA DE UMA ONDA QUADRADA PARA UMA PORTA DE SAÍDA

```
#include <msp430g2553.h>
#include <intrinsics.h>
```

```
int main(void)
{
    WDTCTL = WDTPW + WDTHOLD; // stop WDT

    BCSC1L1 = CALBC1_1MHZ; //clk 1MHz
    DCOCTL = CALDCO_1MHZ;
    P1OUT &= ~BIT5;
    P1DIR |= BIT5;
    TA0CCR0 = 3120-1; //FREQ
    TA0CTL = TASSEL_2 + ID_3 + MC_1 + TAIE;
    _BIS_SR(LPM0_bits+GIE);
    return 0;
}

#pragma vector = TIMER0_A1_VECTOR
__interrupt void TIMER0_TA0_ISR(void)
```

```
{  
    P1OUT ^= BIT5;  
    TA0CTL &= ~TAIFG;  
}
```

## ANEXO II

Partes editadas do PC1.

### REQUISITOS

O sistema deve conseguir primeiramente receber um sinal infravermelho através do receptor codificado e armazená-lo em uma memória do microcontrolador MSP430 utilizando seus registradores e comparadores para identificar a sequência recebida o melhor possível. Após isso o código armazenado deverá ser disponibilizado em um botão ou comando e emitir esse sinal por um LED infravermelho a fim de controlar dispositivos sem mais precisar de seu controle remoto. O sistema controlará equipamentos com esse tipo de comando se o controle original do equipamento estiver disponível (ou qualquer outro controle que seja capaz de comandar o mesmo) para ser possível realizar a cópia do código corretamente. O sistema também só funcionará dentro do alcance de emissão e recepção dos equipamentos (LED infravermelho e receptor infravermelho) que ainda serão definidos.